

Islamic University of Gaza
Deanship of Graduate Studies
Faculty of Engineering
Computer Engineering Program



Design and Implementation of a Standard Library for WebCube Components and Events

By:
Saad J. M. Sunoallah

Supervisor:
Prof. Mohammad Mikki

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master in Computer Engineering

2013

Abstract

3D web browsing is a promising trend for interaction with web content. WebCube is a new model for 3D web browsing that capitalizes on features of virtual reality technology with those of conventional browsing in order to provide an enhanced interactive user experience with web content. The new model is based on representing information content elements in 3D perspective and organizing them inside a 3D container. But 3D websites building requires thousands of java script lines to produce acceptable interface and events. This research is a try to dramatically reduce the time and effort spent by the web developers while building 3D websites, by designing and implementing a robust and pretested standard library for WebCube component and events.

Keywords: 3D Web, WebCube, Web Browsing, WebGL, software library.

تصميم وبناء مكتبة برمجية قياسية لمكونات وأحداث مكعب الويب

الملخص:

تصفح الوب ثلاثي الأبعاد يعد بمستقبل أفضل في تحسين تفاعل المستخدم مع مواقع الويب. مكعب الويب هو نموذج جديد لتصفح الوب ثلاثي الأبعاد يستفيد من مزايا تكنولوجيا الواقع الافتراضي ومزايا التصفح التقليدي لتوفير تجربة متطورة للمستخدم في التفاعل ثلاثي الأبعاد مع محتوى الويب. يستند هذا النموذج الجديد إلى تمثيل عناصر محتوى المعلومات بمنظور ثلاثي الأبعاد وتنظيمها داخل حاوية ثلاثية الأبعاد. لكن بناء مواقع ويب ثلاثية الأبعاد يتطلب كتابة آلاف الاسطر بلغة جافا سكريبت لإنتاج واجهة تفاعلية مقبولة. هذا البحث هو محاولة للحد بشكل كبير من الوقت والجهد المبذولين من قبل مطوري الويب في بناء المواقع ثلاثية الأبعاد، من خلال تصميم وبناء مكتبة برمجية قياسية ثابتة وموثوقة لعناصر وأحداث مكعب الويب.

الكلمات الرئيسية: وب ثلاثي الأبعاد، مكعب الويب، تصفح مواقع الإنترنت، ويب جي ال، مكتبة برمجية.

Acknowledgment

It gives me great pleasure in expressing my gratitude to all those people who have supported me and had their contributions in making this thesis possible. First and foremost, I must acknowledge and thank The Almighty Allah for blessing, protecting and guiding me throughout this period. I could never have accomplished this without the faith I have in the Almighty.

I express my profound sense of reverence to my supervisor Prof. Dr. Mohammad Mikki, for his constant guidance, support and motivation.

I must also thank my family for their continuous support and patience throughout my project.

Table of Contents

عنوان البحث:	II
الملخص:	II
Acknowledgment.....	III
Table of Contents	IV
List of Figures.....	VII
Chapter 1: Introduction	1
1.1. Software library	2
1.1.1. Software metrics	3
1.2. 3D WebCube.....	3
1.2.1. WebCube benefits	4
1.2.2. Web Cube Model Prototype.....	4
1.2.3. Web Cube Model and Its Contents	5
1.3. Thesis motivation.....	5
1.4. Statement of the Problem.....	6
1.5. Objectives	8
1.6. Resources and Tools	8
1.7. Thesis Structure	8
Chapter 2: Theoretical and Technical Foundations	11
2.1. HTML5	11
2.2. CSS3	12
2.3. JavaScript.....	12
2.4. XML.....	13
2.5. WebGL.....	13
2.6. Three.js.....	14
2.7. 3D Graphics, A primer.....	15
2.7.1. Meshes, Polygons, and Vertices	15
2.7.2. Materials, Textures, and Lights.....	16
2.7.3. Cameras, Perspective, Viewports, and Projections	16
2.7.4. Understanding Lighting	17
2.8. Software Development Methodologies.....	20
2.8.1. A brief history	20
2.8.2. Agile software development	22
2.8.3. Agile planning.....	22

2.9.	UML Notation.....	23
2.9.1.	Class Diagrams	23
2.9.2.	Relationships.....	23
Chapter 3: Related Work.....		25
3.1.	3D websites.....	25
3.1.1.	3D Environment.....	25
3.1.2.	3D Style	27
3.2.	3D Web Browser.....	28
3.3.	3D WebCube Prototype	29
3.3.1.	WebCube Model and Its Contents	29
Chapter 4: 3D WebCube Library Implementation		35
4.1.	WebCube Class Diagrams	35
4.1.1.	WebCube Class	35
4.1.2.	Side Class.....	37
4.1.3.	WebCube component.....	39
4.1.4.	HTMLFrame Component	40
4.1.5.	Image component.....	42
4.1.6.	Video component	43
4.1.7.	Audio component.....	44
4.1.8.	Gallery Component.....	45
4.1.9.	Calendar Component.....	46
4.1.10.	Marquee Component.....	48
4.1.11.	Hyperlink component.....	49
4.1.12.	3D Object component	50
Chapter 5: Library Evaluation.....		53
5.1.	Software Testing	53
5.1.1.	Test Levels.....	53
5.1.1.1.	Unit testing.....	53
5.1.1.2.	Integration testing	53
5.1.1.3.	System Testing.....	53
5.1.1.4.	Acceptance testing	54
5.1.2.	Types of Testing.....	54
5.1.2.1.	Black-box Testing.....	54
5.1.2.2.	White-box Testing.....	56

5.2.	Software Metrics	56
5.2.1.	Quality.....	57
5.2.2.	Size (LOC).....	57
5.2.3.	Complexity.....	57
5.3.	Experimental Study.....	57
5.3.1.	Personal Website.....	58
5.3.2.	Commercial website.....	62
5.3.3.	Educational / Entertainment Website.....	68
Chapter 6: Conclusion and Future Work.....		75
6.1.	Conclusion	75
6.2.	Future Work	75
References.....		77
Library Source Code		79

List of Figures

<i>Figure</i>	<i>Caption</i>	<i>Page</i>
Figure 1.1:	Virtual Reality 3D web.	1
Figure 1.2:	3D web presentation as a style.	2
Figure 1.3:	WebCube web browsing.	4
Figure 1.4:	Example of Web Cube and its components.	5
Figure 1.5:	The code used to creates the WebCube image component.	7
Figure 2.1:	A 3D coordinate system.	15
Figure 2.2:	Walt Disney head object showing triangle polygons.	16
Figure 2.3:	Concepts of the camera, viewport, and projection.	17
Figure 2.4:	PerspectiveCamera vs OrthographicCamera	17
Figure 2.5:	Lights initializations.	18
Figure 2.6:	Three.js example.	18
Figure 2.7:	A rotated cube	20
Figure 2.8:	Waterfall software process model	20
Figure 2.9:	Types of UML relationships	24
Figure 3.1:	Google Earth 6.2, 3-D street view.	26
Figure 3.2:	World of Warcraft web game.	26
Figure 3.3:	People gathered at SecondLife.	27
Figure 3.4:	3D web presentation as a style.	27
Figure 3.5:	3D Gallery example.	28
Figure 3.6:	3D Web Browser Pro.	28
Figure 3.7:	CubicEye Web Browser	29
Figure 3.8:	Example of WebCube and its components.	30
Figure 3.9:	WebCube image component.	31
Figure 3.10:	WebCube video component.	31
Figure 3.11:	WebCube gallery components.	32
Figure 3.12:	WebCube HTML frame components.	33
Figure 3.13:	WebCube 3-D object model components.	33
Figure 4.1:	WebCube surfaces	35
Figure 4.2:	WebCube and Side class diagrams	36
Figure 4.3:	TopLeftCorner and BottomRightCorner for each side	38
Figure 4.4:	Objects interlacing problem	38
Figure 4.5:	WebCube Component class	39
Figure 4.6:	Sample HTMLFrame component	40
Figure 4.7:	HTMLFrame class	41
Figure 4.8:	Image component	42
Figure 4.9:	ImageObject class	42
Figure 4.10:	Video component	43
Figure 4.11:	VideoObject class	44
Figure 4.12:	Audio component	44
Figure 4.13:	Gallery component	45
Figure 4.14:	GalleryObject class	45
Figure 4.15:	Sample gallery XML file	46
Figure 4.16:	Calendar component	47

Figure 4.17:	CalendarObject class	47
Figure 4.18:	Marquee class	48
Figure 4.19:	Sample Marquee XML file	49
Figure 4.20:	Marquee component	49
Figure 4.21:	Hyperlink class	50
Figure 4.22:	3D object added to the WebCube	51
Figure 4.23:	3D object class	51
Figure 5.1:	Black-Box Test of ImageObjects.	55
Figure 5.2:	Result of Black-Box test of ImageObject	56
Figure 5.3:	Rasheed website	58
Figure 5.4:	The code needed to build Rasheed website	59
Figure 5.5:	Rasheed website built using the library	61
Figure 5.6:	Screenshots of the personal website	61
Figure 5.7:	Auto show website - Lamborghini webpage	62
Figure 5.8:	3D models properties and color definitions	63
Figure 5.9:	Auto show website code.	65
Figure 5.10:	Auto show website- using the library	67
Figure 5.11:	screenshots of auto show website	68
Figure 5.12:	Educational website.	69
Figure 5.13:	The code needed to build the educational website.	70
Figure 5.14:	screenshots of the educational website	73
Figure A.1:	Side class code	79
Figure A.2:	WebCube class code	80
Figure A.3:	HTMLFrame class code	92
Figure A.4:	ImageObject class code	103
Figure A.5:	VideoObject class code	116
Figure A.6:	AudioObject class code	134
Figure A.7:	GalleryObject class code	147
Figure A.8:	CalendarObject class code	162
Figure A.9:	Marquee class code	175
Figure A.10:	Hyperlink class code	183
Figure A.11:	D3Object class code	187

Chapter 1

Introduction

With the increase of Internet speed, it would be possible to include rich content in web sites like videos, images, animations and 3D content that simulates real objects such as marketing products, human body parts and museum collections [35].

In the last few years, 3D web environment was one of the most interesting and evolving research areas, but the effort in this area of research is not on the level of ambition [8]. Nowadays, there are numerous of researches and projects that exploit 3D environment, there is a promised future for this trend of using 3D technology due to an opportunity to improve human computer interaction (HCI). So, users of different domains such as commercial, medical, and etc. interact with 3D graphical interfaces which open more doors for research in this area.



Figure 1.1: Virtual Reality 3D web.

These projects classified into two main types according to deploying 3D environment in the Internet. The first type based on creating a full 3D environment in the Virtual Reality (VR) “3D interaction” while the second uses 3D presentation as a style for organizing search results. In the first, 3D interaction environment consist of 3D buildings, 3D streets, 3D gardens, 3D surfaces, and so on. Users in this environment can tour through it and interact with its content. This type put users in VR environment enables them to tour freely with

unconfined area Figure 1.1 depicts this type of 3D web. Projects of this type have significant impact for simulating the real world locations to improve user knowledge of locations which cannot be reached easily in the real world. The main problem of this type is that abandoned hyperlink page style.

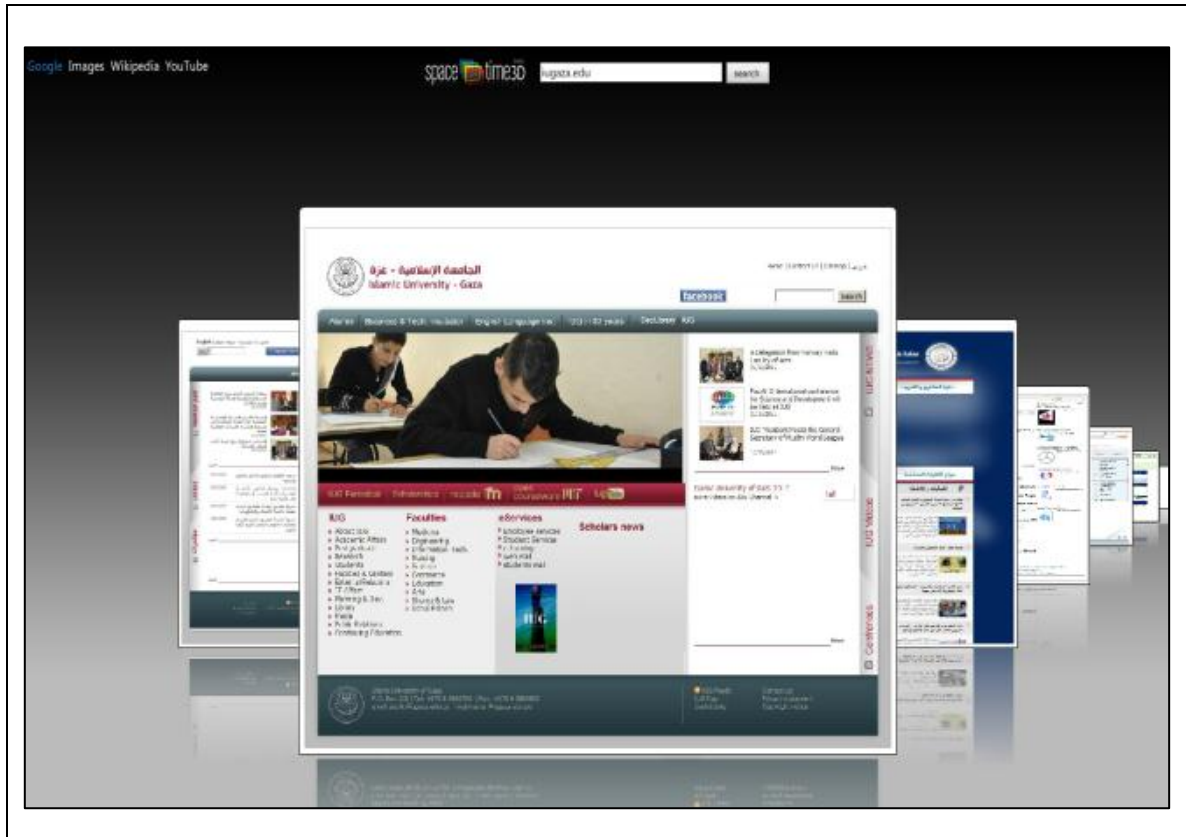


Figure 1.2: 3D web presentation as a style.

In the other hand, the second type of 3D environment still in the beginning and a few attempts were found. Some websites try to employ 3D web environment only for browsing media files such as images and videos in 3D perspective. Figure 1.2 depicts the second type of 3D web. This type of 3D environment used as a style for organizing search results and no 3D interaction taken into account [29].

1.1. Software library

A software library is an organized collection of code with associated tools supporting programming in general or in specific domains, usually united by a specified set of principles and conventions. Most libraries are aimed at the use by several people and in different environments¹.

¹ <http://lcsd05.cs.tamu.edu/>, Retrieved 6 Oct. 2012

The basic definition for a software library can be stated as: “A collection of composable assets that contribute to building a product aggregated within a mechanism that holds the collection for the purpose of promoting reuse by providing greater accessibility, for use by others” [17].

Libraries contain code and data that provide services to independent programs. This encourages the sharing and changing of code and data in a modular fashion, and eases the distribution of the code and data¹.

High performance computing, and scientific computing in particular, relies heavily on software libraries. Libraries provide an easy mechanism for reusing code. Moreover, each library typically encapsulates a particular domain of expertise, such as graphics or linear algebra, allowing programmers to think at a higher level of abstraction [15]. This research will add a library in computer graphics domain, this library is a compound of WebGL 3D graphics and JavaScript programming expertise, making the build of 3D websites is a trivial process.

1.1.1. Software metrics

Software metric is: “a measure of some property of a piece of software or its specifications” [28]. Many useful comparisons involve only the order of magnitude of lines of code in a project. Software projects can vary between 1 to 100,000,000 or more lines of code. Using lines of code to compare a 10,000 line project to a 100,000 line project is far more useful than when comparing a 20,000 line project with a 21,000 line project².

Lines of code (or LOC) is possibly the most widely used metric for project size. It would seem to be easily and precisely definable; however, there are a number of different definitions for the number of lines of code in a particular project. But the most common definition of LOC seems to count any line that is not a blank or comment line, regardless of the number of statements per line [18].

LOC is typically used to estimate the amount of effort that will be required to develop a project and to estimate productivity once the software is produced [28].

1.2. 3D WebCube

Web-based Graphics Library (WebGL) is the new standard for 3D graphics on the Web. It allows developers to harness the full power of the computer’s graphics rendering hardware using only JavaScript, a web browser, and a standard web technology stack. Before WebGL, they had to rely on plugins or native applications and ask their users to download and install custom software in order to deliver a true 3D experience [24].

¹ [http://en.wikipedia.org/wiki/Library_\(computing\)](http://en.wikipedia.org/wiki/Library_(computing)) , Retrieved 6 Oct. 2012

² http://en.wikipedia.org/wiki/Software_metric , Retrieved 6 Oct. 2012

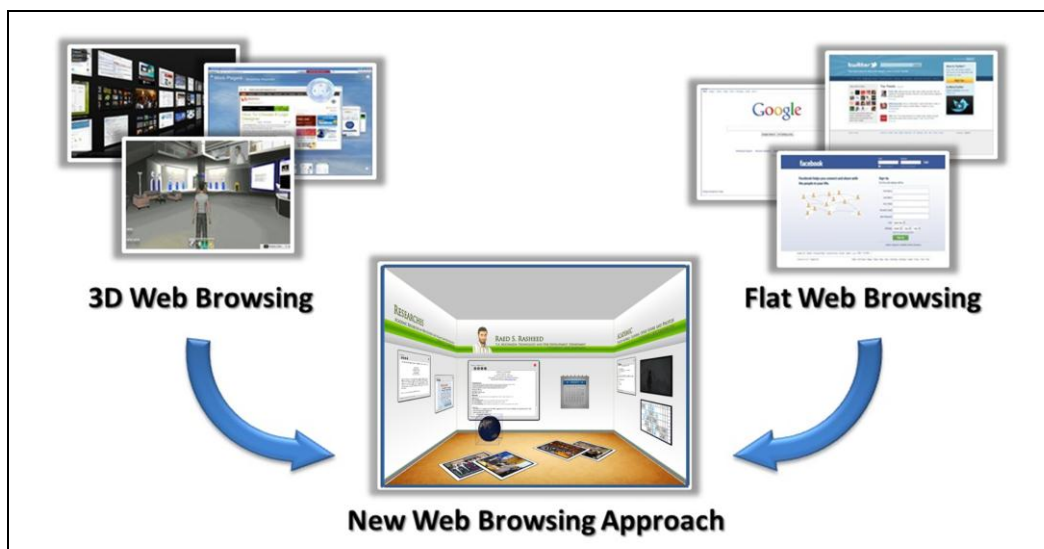


Figure 1.3: WebCube web browsing.

3D WebCube approach is a prototype of 3D browsing of web sites that effectively combines features from both 3D interaction and conventional web browsing; Figure 1.3 depicts the concept of integrating the virtual 3D environment with conventional web browsing. This model is not to replace the 2-D flat web page model. It will rather complement it in areas where 3D presentation and interaction is more suitable. Designers will have a new alternative for designing web content based on what is best for users.

1.2.1. WebCube benefits

1. Integrate features of 3D web browsing technology and conventional web browsing technology.
2. Using of conventional navigation-oriented paradigm within 3D interaction environment.
3. Putting users in a semi-realistic environment, and easily interacting contents of the website as real life.
4. Increase user satisfaction leads to improve browsing experience.
5. Designers will have a new alternative for designing web content based on what is best for users.

1.2.2. Web Cube Model Prototype

3D WebCube is a novel 3D web browsing model that casts the third dimension based on the current web browsing experience. This model will improve the existing web browsing experience at three levels:

1. The container level: container of information, i. e. the web page,
2. The element level: various web content component, and
3. The interaction level: the associated interaction mechanisms.

3D Web Cube is a 3D model that corresponds to the conventional web page. The Web Cube is a limited space in three dimensions that contains 3D web content elements. Users navigate within the Web Cube and interact with its contents, and also navigate among Web Cubes [29].

1.2.3. Web Cube Model and Its Contents

WebGL, JavaScript, HTML5 and CSS3 are used to build the Web Cube environment and its content. The canvas element in HTML5 is used to draw 2-D graphics on the web. Drawing 3D graphics on the canvas element needs the WebGL library.

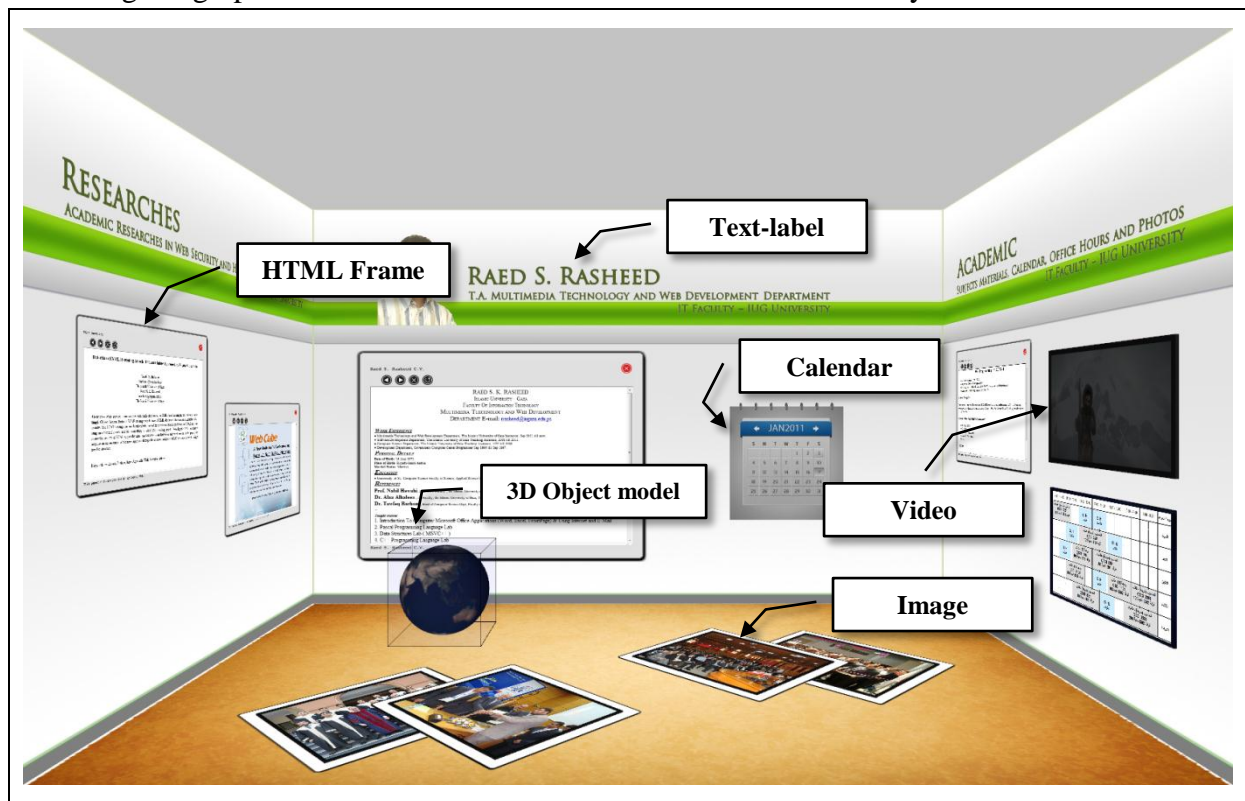


Figure 1.4: Example of Web Cube and its components.

The main Web Cube structure depicted in Figure 1.4 consists of five surfaces: the front wall, two sides walls one to the right and the other to the left, floor and ceil. Each surface can hold part of the web content such as text frames or photos. The user cannot navigate behind these surfaces except navigating to another Web Cube using link for that cube. The main Web Cube components implemented in the model prototype are text-label, image, video, calendar, gallery, link, frame and object-model.

1.3. Thesis motivation

Viewing here the evaluation for WebCube prototype that encourages us to proceed in building a WebCube library, here is the result of evaluating the feasibility and usability of the WebCube model done by Rasheed as a try to evaluate the WebCube model rototype.

The first experiment is to examine the sub-hypothesis “Web Cube content is more visible than webpage content”. Results Show that:

1. The average percentage of web-content visible on screen in the WebCube is 83% and in the conventional webpage is 21.4%.
2. The average time spent to reach goal content in the WebCube is 10.77 sec, and in the conventional webpage is 32.24 sec.

From the questionnaire results and in order of total responses depicts that:

1. More than 94% of the participants will use the Web Cube frequently.
2. More than 80% believe that Web Cube is easy to interact.
3. Less than 8% need technical support to be able to use the Web Cube.
4. Less than 4% detect much inconsistency in the Web Cube.
5. More than 61% believe that most people could learn this Web Cube quickly.
6. Less than 8% find that Web Cube is difficult to use.
7. Less than 10% need to know lot of things before they can use the Web Cube.
8. More than 80% feel very confident using the WebCube.
9. More than 60% believe that they can dispense conventional web interface using the WebCube.
10. More than 78% find that contents of the Web Cube are easy to manipulate.

The previous evaluations show that WebCube was acceptable by majority of web users those use it, and improves the user browsing experience.

1.4. Statement of the Problem

Creating a 3D web contents is now possible with WebGL technology and javascript code, but the use of WebGL require a deep experience in 3D graphis. With three.js library, dealing with 3D component become more simpler and easier, but it still require a large number of javascript lines to build a simple 3D objects.

The current 3D websites separated into two different fields as mentioned in the introduction of this chapter, the first one is the complete 3D environment such as 3D games and the second one is the 2D contents websites with 3D styles. But WebCube prototype was a novel idea to build a 3D websites for general use, like personal or commercial websites. WebCube is backward compatible since it also can merge the traditional 2D contents into a real 3D environment. Webcube is represented by a room or a cube with the ability to website contents inside the boundaries of that cube.

This unique prototype in this field has a lot of benefits:

- The boundaries of the WebCube allowing the user to view all the web site contents without the need to surf in the 3D environment or scrolling the page.
- The five surfaces of the webcube afford additional space to add a lot of components in the cube boundaries.

- The room shape of this prototype making the website look like a real room with objects spreaded on the walls.
- The ability to add 2D contents making it compatible with traditional webcontents.

3D WebCube prototype has a major drawback which is that a very simple WebCube site requires more than five thousand of java script lines to produce acceptable interface and events, this is because the website designer must create each WebCube object from its basic shapes, WebCube objects means frames, calendar, gallery and so on, and the basic shapes are rectangles, lines, text and simple images. And to add the event handler for any object the designer need another tens of lines, for example the following code lines shown in figure 1.5 are required to add an image object to the WebCube.

```
function RSR_Image01(ImageURL, width, height, depth, pX, pY, pZ, rX, rY, rZ,
alpha){
RSR_Image01Plane = new THREE. Object3D();
var RSR_windowTextureMaterial = THREE. ImageUtils. loadTexture("RSRwindow. png");
var RSR_windowMaterial = new THREE. MeshBasicMaterial( { color:0xfffff, map:
RSR_windowTextureMaterial, opacity: 1. 7 ,transparent: true} );
var RSR_windowGeometry = new THREE. PlaneGeometry( width+RSR_webCubeWidth*0. 036,
height+RSR_webCubeHeight*0. 057 );//new THREE. CubeGeometry( 150, 80, 5 );
RSR_window = new THREE. Mesh(RSR_windowGeometry,RSR_windowMaterial);
RSR_window. doubleSided = true;
RSR_Image01Plane. addChild(RSR_window);
var RSR_Image01TextureMaterial = THREE. ImageUtils. loadTexture(ImageURL);
var RSR_Image01Material = new THREE. MeshBasicMaterial( { color:0xfffff, map:
RSR_Image01TextureMaterial, opacity: 1. 0, transparent: false} );
var RSR_Image01Geometry = new THREE. PlaneGeometry( width, height );//new THREE.
CubeGeometry( 150, 80, 5 );
RSR_Image01 = new THREE. Mesh(RSR_Image01Geometry,RSR_Image01Material);
RSR_Image01. position. set(0,0,0. 5);
RSR_Image01Plane. addChild(RSR_Image01);
var RSR_Image01BorderTextureMaterial = THREE. ImageUtils. loadTexture("back05.
jpg");
var RSR_Image01BorderMaterial = new THREE. MeshLambertMaterial( { color:0x202020,
opacity: 1. 0, transparent: false } );
var RSR_Image01TopBorderGeometry = new THREE. CubeGeometry( width+10, 5, depth );
RSR_Image01TopBorder = new THREE.
Mesh(RSR_Image01TopBorderGeometry,RSR_Image01BorderMaterial);
RSR_Image01TopBorder. position. set(0,((height+5)/2),0);
RSR_Image01Plane. addChild(RSR_Image01TopBorder);
RSR_Image01BottomBorder = new THREE.
Mesh(RSR_Image01TopBorderGeometry,RSR_Image01BorderMaterial);
RSR_Image01BottomBorder. position. set(0,-((height+5)/2),0);
RSR_Image01Plane. addChild(RSR_Image01BottomBorder);
RSR_Image01LeftBorder = new THREE.
Mesh(RSR_Image01LeftBorderGeometry,RSR_Image01BorderMaterial);
RSR_Image01LeftBorder. position. set(-((width+5)/2),0,0);
RSR_Image01Plane. addChild(RSR_Image01LeftBorder);
RSR_Image01BottomBorder = new THREE.
Mesh(RSR_Image01LeftBorderGeometry,RSR_Image01BorderMaterial);
RSR_Image01BottomBorder. position. set(((width+5)/2),0,0);
RSR_Image01Plane. addChild(RSR_Image01BottomBorder);
return RSR_Image01Plane;}
```

Figure 1.5: The code used to creates the WebCube image component.

Each simple wall frame (inactive frame) created from independent parts like border, content picture, event buttons, back image and title text. These parts must grouped together when handling an event (move or activate), which will consume web designer time; source lines of code and introduce more bugs for a simple website with few frames.

So this research will introduce for design and implementation of a standard library for WebCube components and events allowing the web developer to build a complete 3D website in a few simple lines of code.

1.5.Objectives

Software library has many benefits:

- Encourages the sharing and changing of code and data in a modular fashion¹.
- Eases the distribution of the code and data.
- Minimize source line of code.
- Save programmer time and effort.
- Minimize the bugs in the software when using a pretested, trusted and robust software library.

Our goal is to produce a standard library for WebCube site design that provides a set of objects needed to build almost any website, this allow the web designer to build a complete website in short time by writing a few simple lines of code, without the need to worry about event handling, object management or object movement and interlacing.

1.6. Resources and Tools

The following resources and methods will be required:

1. HTML5 specification.
2. XML specification.
3. CSS3 specification.
4. JavaScript scripting language.
5. WebGL “Three.js” API.

1.7. Thesis Structure

This thesis consists of six chapters: Introduction, Technical Foundations, Related Works, Library Implementation, Library Evaluation and Conclusion and Future Work. The main points discussed in the chapters are:

1. Chapter 1 “Introduction” gives a short introduction about software libraries, 3D WebCube, thesis problem and objectives.

¹ [http://en.wikipedia.org/wiki/Library_\(computing\)](http://en.wikipedia.org/wiki/Library_(computing)), Retrieved 6 Oct. 2012

2. Chapter 2 “Theoretical and Technical Foundations” describes theoretical foundations needed for thesis such as Agile software development method and technical foundations needed for thesis such as HTML5, CSS3, JavaScript, XML, WebGL.
3. Chapter 3 “Related Works” presents related work to the thesis.
4. Chapter 4 “3D Web Cube Library Implementation” is devoted to the presenting the implementation of the 3D WebCube library.
5. Chapter 5 “Library Evaluation” shows three examples of 3D websites clarifying the advantages of the library.
6. Chapter 6 “Conclusion and Future Work” discusses the final conclusions and presents possible future works.

Chapter 2

Theoretical and Technical Foundations

Technical foundations include the set of technologies based on which this library can be implemented. This chapter will introduce the Web and 3D Web design technologies, these are the tools needed for designing the WebCube and its components; both flat and 3D components. This set includes: HTML5, CSS3, JavaScript, XML, WebGL and Three.js. Later it will introduce theoretical basics of 3D graphics and explain agile software development methodology basics that the library developing depends on.

The following subsections discuss these technologies in further detail, and explain in what way each technology was necessary for the development of the library.

2.1. HTML5

HTML is the primary language for designing web pages and has a strong potential for that. The language of HTML5 is the latest version of HTML and is currently under development.

With the first introduction of HTML 5 in 2008 numerous additions were added to the HTML standard and a few deprecated tags have been removed. The most interesting was the inclusion of the <canvas> element. This element provides programmers with the ability for dynamic, scriptable rendering of 2D shapes and bitmap images. The canvas element is a field within a browser with defined width and height, which can be used to draw on. This field however can be accessed by JavaScript Code with a full set of drawing functions, allowing for dynamically generating images and graphics [21]. Like its predecessors, HTML5 is designed to be cross-platform. You don't need to be running any particular operating system in order to take the advantages of HTML5. A modern web browser is all what you need [25].

New features were added to HTML5 such as videos, sounds and animations which were previously dependent on external plugins such as Flash from the Adobe or Silverlight from Microsoft. HTML5 is a product of cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). They work together on HTML5 since 2004, and it is expected to remain under development for years, but completed parts will be supported in the new versions of major web browsers. The most interesting features in HTML5 are:

- **Better control:** the language includes a distinct set of Application Programming Interfaces (API) which provides access to microphone and camera that allow the programmer to build interactive web applications.
- **Offline browsing:** where it is possible to browse web without a connection to the Internet by storing webpage on the client.

- **Multimedia support:** You can add audio and video without any need to install any external plugins like Flash or Silverlight plugin.
- **Graphical effects:** It includes graphics libraries which provide many tools for drawing and animations.
- **Compatibility:** It is compatible with previous versions of HTML.

No browsers today can claim 100 percent support for the entire HTML5 feature set and Internet Explorer, has only begun HTML5 support with its newest version, 9. Some features like WebGL are not implemented at all, and Microsoft has so far not shown any interest in doing so [32].

2.2. CSS3

Cascading Style Sheets (CSS), is a language to define the presentation and appearance of an HTML document [1], such as colors, fonts and layout. Cascading Style Sheets are growing ever more important and more sophisticated. With CSS, web developers lay out web pages to match their intentions [36]. There are now more than 40 modules in CSS3 at various stages of completion and browser support. The modular system is beneficial in many ways. It makes CSS3 easier to write, and implement by the browser [22].

CSS works by associating rules with HTML elements specifying how the content of specified elements should be displayed. A CSS rule contains two parts: a selector which element the rule applies to, and a declaration which indicate how the elements should be styled. Declarations are split into two parts: a property and a value [10].

CSS3 allows you to create some really beautiful effects to your web designs. But most of the visual effects that CSS3 allows you to create can be accomplished without CSS3, using alternative tools such as images, JavaScript, or Flash. Another issue, using CSS3 decrease not only the time you spend developing and maintaining pages, but also the time spent in loading those pages. Simultaneously, you can increase usability and accessibility, make your pages more adaptable across devices, and even enhance your search engine placement [13].

2.3. JavaScript

The main purpose of JavaScript when first appeared in 1995 was to handle some of the input validation that had previously been left to server-side languages such as Perl. A round trip to the server was needed to validate a required field value. Netscape Navigator sought to change that with the introduction of JavaScript. The capability to handle some basic validation on the client was an exciting new feature at a time of slow internet connection was widespread. Such slow speeds turned every trip to the server into an exercise in patience [38]. JavaScript is now the programming language of the Web. The overwhelming majority of modern websites use JavaScript, making JavaScript the most ubiquitous programming language in history [12]. Despite the name, JavaScript is most similar to the C programming language, with some aspects inherited from Java [37].

JavaScript is an object-oriented, weakly typed, scripting language. Almost all of the variables are objects. An object is a special variable type that can have members, divided into variables, called properties, and functions, called methods. For example the string variable automatically has a property called length, which reflects the number of characters in the string, and has several defined methods, like substring() and toUpperCase(). With object-oriented programming, you'll use object notation extensively to refer to an object's members: someObject.someProperty or someObject.someMethod() [34].

A major development in the history of JavaScript is the creation of frameworks. A framework is just a library of code. In any programming language there are many tasks and processes that get repeated. Instead of just re-create the appropriate code each time, its better to write a library that will easily and quickly replicate that code for you. JavaScript frameworks can simplify form validation, and enhance common Web elements, such as paginating and sorting tables of data [34].

JavaScript normally executed on the client-side and integrated with HTML code. JavaScript can be executed according several events like loading webpage, mouse drag-and-drop and keyboard typing, submitting a form, or leaving the page.

2.4. XML

XML, or eXtensible Markup Language, is a specification for storing information and describing the structure of that information. And while XML is a markup language, XML has no tags of its own. It allows the person writing the XML to create whatever tags they need while adhering the rules of the XML specification [14]. XML usfull to store data needed to be consumed by more than one application. It has gained a reputation for being a candidate where interoperability is important, either between two applications in different businesses or simply those within a company [11].

An XML document is simply a text file and nothing more. It is well-structured, easy to understand, parse, manipulate, and is considered "human-readable". XML was created by the W3C and it is free to anyone who wishes to use it [14].

In the library implementation, XML files used to store a long list parameter passed to an object like image list for the Gallery object or marquee elements for the Marquee object as explained in chapter 4.

2.5. WebGL

WebGL stands for Web-based Graphics Library is an application programming interface (API) for advanced 3D graphics on the web. It is based on OpenGL ES 2.0, and provides similar rendering functionality, but in an HTML and JavaScript context. It uses HTML5 canvas element as a rendering surface. The final WebGL 1.0 specification was frozen in March 2011. WebGL support is implemented in several browsers, including Google Chrome, Mozilla Firefox, and (at the time of this writing) in the development releases of Safari and Opera [2]. With WebGL, developers can harness the full power of the computer's graphics

rendering hardware using only JavaScript, a web browser, and a standard web technology stack. Before WebGL, developers had to rely on plugins or native applications and ask their users to download and install custom software in order to deliver a true 3D experience [24].

WebGL is a JavaScript API binding from the CPU to the GPU of a computer's graphics card allowing for faster rendering and the ability to produce a degree of realism and configurability that is not possible outside of using WebGL [9]. Some uses of WebGL are viewing and manipulating models and designs, virtual tours, mapping, gaming, art, data visualization, creating videos, manipulating and processing of data and images. WebGL enables modern Internet browsers to render 3D scenes in a standard and efficient manner. Rendering: "is the process of generating an image from a scene file by means of computer programs". There are two different ways to produce such images. The software-based rendering, where all the calculations required to render 3D scenes are performed using computer CPU; and the hardware-based rendering where there is a Graphics Processing Unit (GPU) performing the computations in real time. From a technical point of view, hardware-based rendering is much more efficient than software-based rendering because there is dedicated hardware taking care of the operations [7]. A scene object contains objects in a strictly defined language; it would contain geometry, viewpoint, texture, lighting, and shading information as a description of the virtual scene. The data contained in the scene is then passed to a rendering program to be processed and output to a digital image.

With WebGL, you get hardware-accelerated 3D graphics inside the browser. You can create advanced 3D graphics applications, and have all the benefits that a web application has: cheap, easy to distribute, easy maintenance and platform independent. In addition to these benefits, WebGL also has the following attractive characteristics [2]:

- WebGL is an open standard that everyone can implement or use.
- WebGL is really fast since it takes advantage of the graphics hardware to accelerate the rendering.
- No plug-in is needed to support WebGL.
- It is quite easy to learn for many developers who have previous experience with OpenGL API.
- WebGL combines with other web content; the 3D canvas can take up either a portion of the page, or the whole page. And you can develop 3D graphics using WebGL, but all your other elements are built using familiar old HTML [24].

2.6. Three.js

3D web environment development complexity encouraged developers to develop many libraries based on WebGL for easier 3D web development using WebGL. Three.js¹ is one of these libraries. It is a lightweight and low-complexity library. It makes easier development

¹ <http://mrdoob.github.com/three.js/>

way for creating 3D Web environment. It enables the developer to call functions that do many operations those require a great effort using WebGL pure programming.

Three.js provides a set of objects that are commonly found in 3D graphics. It is fast, powerful, and open source, hosted on GitHub, and well maintained [24]. Three.js core object types are [9]:

- **THREE.Renderer:** The object that actually renders the scene.
- **THREE.Scene:** Scene graph that stores the objects and lights contained within a scene.
- **THREE.Camera:** Virtual camera; can be `PerspectiveCamera` or `OrthographicCamera`.
- **THREE.Object3D:** Many object types, including `Mesh`, `Line`, `Particle`, `Bone` and `Sprite`.
- **THREE.Light:** Light model. Types can be `AmbientLight`, `DirectionalLight` or `PointLight`.

2.7. 3D Graphics, A primer

The WebGL coordinate system is arranged as depicted in Figure 2.1, with x axis running from left to right, y axis running from bottom to top, and positive z coming out of the screen.

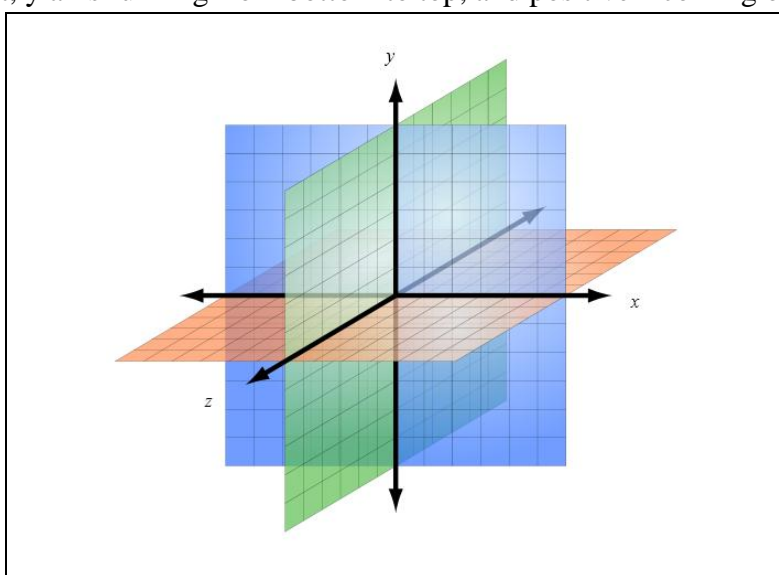


Figure 2.1: A 3D coordinate system
(https://commons.wikimedia.org/wiki/File:3D_coordinate_system.svg)

2.7.1. Meshes, Polygons, and Vertices

The most common way to draw 3D graphics is to use a mesh. A mesh is an object composed of one or more polygonal shapes, constructed out of vertices (x, y, z triples) defining coordinate positions in 3D space. The polygons could be triangles or quads. Figure 2.2 illustrates a 3D mesh. The dark lines outline the triangles that comprise the mesh, defining the shape of the face. The x, y, and z components of the mesh's vertices define the

shape only where color and shading properties of the mesh surface are defined using additional attributes.

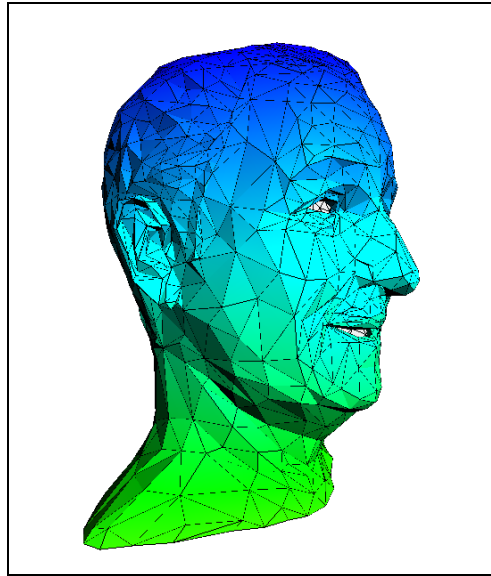


Figure 2.2: Walt Disney head object showing triangle polygons

2.7.2. Materials, Textures, and Lights

Surface attributes can be a single solid color or can be complex. Surface information can also be represented using textures. In most graphics systems, the surface properties of a mesh are referred as materials [24].

2.7.3. Cameras, Perspective, Viewports, and Projections

3D systems typically use a camera as a point of view from which the user will be viewing the scene, camera object defines where the user is positioned and oriented. The camera's properties combine to deliver the final rendered image of a 3D scene into a 2D viewport defined by the window or canvas. Figure 2.3 depicts the core concepts of the camera, viewport, and projection. The icon of the eye represents the location of the camera. The red vector represents the direction in which the camera is pointing. The blue cubes are the objects in the 3D scene. The green and red rectangles are the near and far clipping planes which define the boundaries of a subset of the 3D space, known as the view volume. Only objects within the view volume are actually rendered to the screen. The near clipping plane is where the user will see the final rendered image [24].

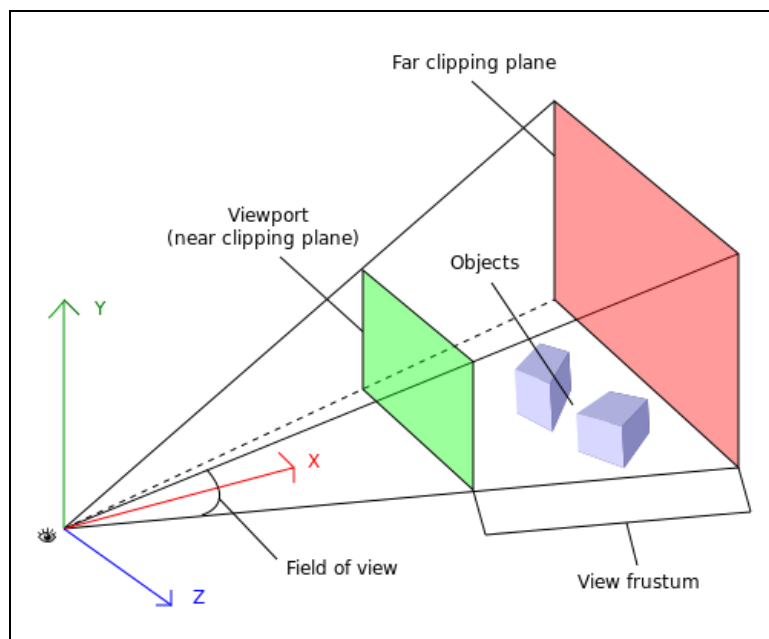


Figure 2.3: Concepts of the camera, viewport, and projection [24].

The type of camera determines how the 3D model will be rendered into a 2D image on-screen. With `PerspectiveCamera`, the farther away objects are, the smaller they appear. Because that's how things look in real life, this produces a reasonably natural-looking image. Where a 3D object of a particular size will always be rendered at exactly the same size with `OrthographicCamera`, regardless of how far away it is. This tends to produce rather unnatural-looking images, but it can occasionally be useful sometimes. Figure 2.4 shows a series of identical columns rendered with a `PerspectiveCamera`, and with an `OrthographicCamera` [33].

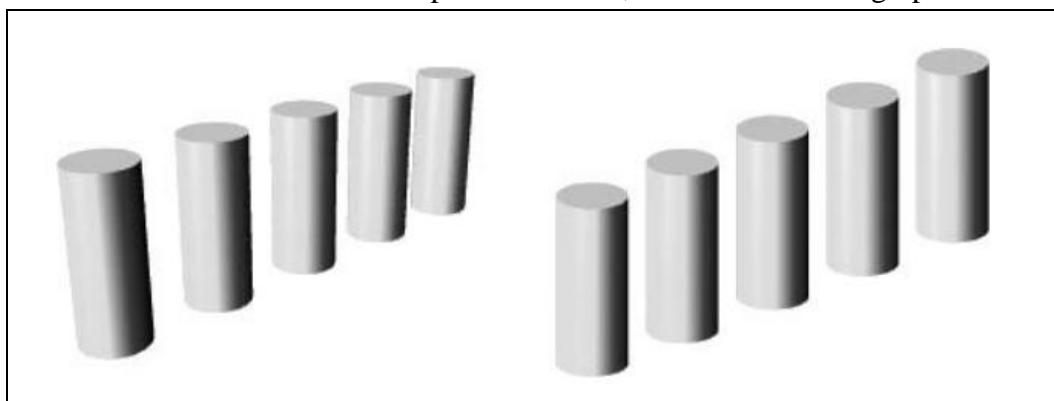


Figure 2.4: `PerspectiveCamera` (left) vs `OrthographicCamera` (right)[33]

2.7.4. Understanding Lighting

Lighting in Three.js isn't that dissimilar to what happens in the real world via the Sun. It has three types of lighting objects [37]:

- **AmbientLight:** Ambient lighting is the average of all the light generated from all light sources in an area. If objects rendered with only ambient lighting, it will appear two-dimensional because all vertices receive the same amount of light.

- **PointLight:** Point lighting is attenuated light coming from a specific location and emitted in all directions and does make objects look more 3D. As an object moves further from the light source, the amount of light that can affect objects is attenuated.
- **DirectionalLight:** Directional lighting can be viewed as similar to shining a bunch of lamps on a subject from the same direction. Instead of attenuation over distance, directional lights deliver the same intensity as they stretch toward the specified maximum distance.

Figure 2.5 shows examples of the three light types, starting with AmbientLight, which has a sole parameter. Intensity corresponds to the brightness of the light rays, and distance refers to the maximum distance of light ray before falloff. CastShadow determines whether or not the object illuminated by DirectionalLight with cast a shadow. Parameters listed in brackets are optional.

```
new THREE.AmbientLight(hexColor);
new THREE.PointLight(hexColor, [intensity], [distance]);
new THREE.DirectionalLight(hexColor, [intensity], [distance], [castShadow]);
```

Figure 2.5: Lights initializations

Figure 2.6 despite a three.js example contains the complete code listing for a rotating cube drawing program using cube geometry, covered with textural material, and the light used is the directional one.

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to WebGL</title>
<script src="../../libs/Three.js"></script>
<script src="../../libs/RequestAnimationFrame.js"></script>
<script>
var renderer = null,
    scene = null,
    camera = null,
    cube = null,
    animating = false;
function onLoad()
{
    // Grab our container div
    var container = document.getElementById("container");

    // Create the Three.js renderer, add it to our div
    renderer = new THREE.WebGLRenderer( { antialias: true } );
    renderer.setSize(container.offsetWidth,
container.offsetHeight);
    container.appendChild( renderer.domElement );

    // Create a new Three.js scene
    scene = new THREE.Scene();

    // Put in a camera
    var aspect = container.offsetWidth / container.offsetHeight;
```

Figure 2.6 (page 1 of 2): Three.js example.

```

    camera = new THREE.PerspectiveCamera( 45,aspect, 1, 4000 );
    camera.position.set( 0, 0, 3 );

    // Create a directional light to show off the object
    var light = new THREE.DirectionalLight( 0xffffff, 1.5);
    light.position.set(0, 0, 1);
    scene.add( light );
    // Now, create the textural material
    var mapUrl = "../images/monster.jpg";
    var texture = THREE.ImageUtils.loadTexture(mapUrl);
    var material = new THREE.MeshBasicMaterial({ map: texture });

    // Create the cube geometry
    var geometry = new THREE.CubeGeometry(1, 1, 1);
    // And put the geometry and material together into a mesh
    cube = new THREE.Mesh(geometry, material);

    // Turn it toward the scene, or we won't see the cube shape!
    cube.rotation.x = Math.PI / 5;
    cube.rotation.y = Math.PI / 5;
    // Add the cube to our scene
    scene.add( cube );
    // Add a mouse up handler to toggle the animation
    var dom = renderer.domElement;
    dom.addEventListener( 'mouseup', onMouseUp, false);
    // Run our render loop
    run();
}
function run()
{
    // Render the scene
    renderer.render( scene, camera );

    // Spin the cube for next frame
    if (animating)
        cube.rotation.y -= 0.01;

    // Ask for another frame
    requestAnimationFrame(run);
}
function onMouseUp    (event)
{
    event.preventDefault();
    animating = !animating;
}
</script>
</head>
<body onLoad="onLoad();" style="">
    <center><h1>Welcome to WebGL!</h1></center>
    <div id="container"
        style="width:95%; height:80%; position:absolute;">
    </div>
</body>
</html>

```

Figure 2.6 (page 2 of 2): Three.js example [24].

Figure 2.7 shows the output of the previous example, a rotating cube geometry covered by monster image material.

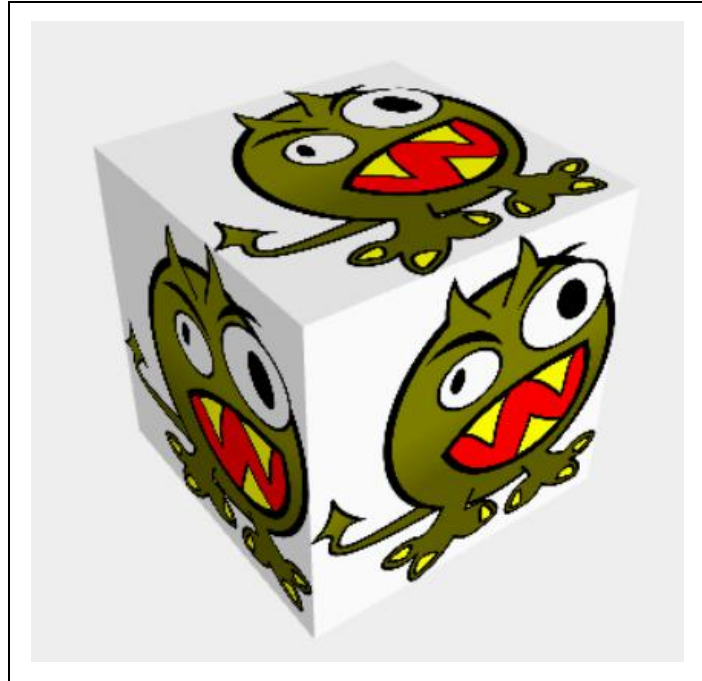


Figure 2.7: A rotating cube

2.8. Software Development Methodologies

A software development methodology “is a collection of procedures, techniques, tools and documentation aids which will help the systems developers in their efforts to implement a new information system” [3]. Waterfall and agile methodologies are focused on in this research because they represent the old and modern methodologies respectively. The contradictions in these methodologies clarify the direction of software development process.

2.8.1. A brief history

After its inception in the 1950s and 1960s, the software industry advanced quickly. As it did so, the need to be able to better predict and control ever larger-scale software project outcomes the sequential, stage-gated “waterfall” software process model, usually typified by a graphic such as Figure 2.8[19].

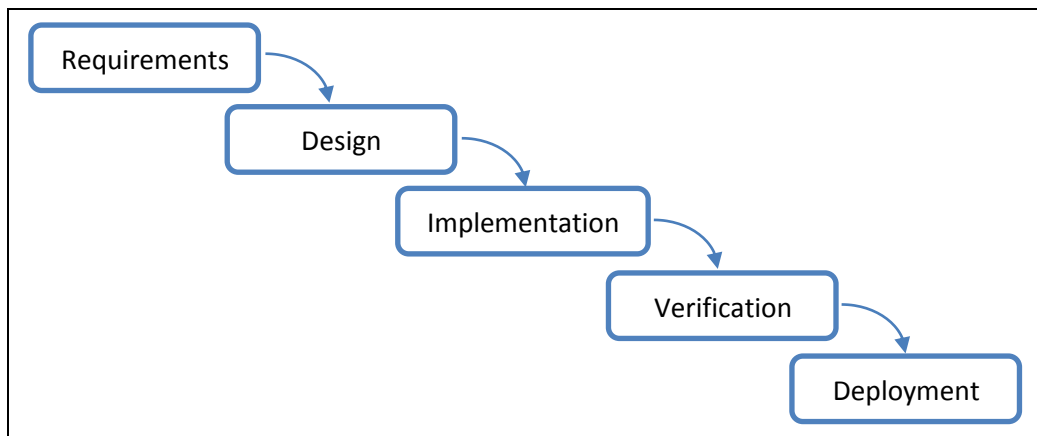


Figure 2.8: Waterfall software process model

The Waterfall approach was developed by the U.S. Navy to enable development of complex software. In this model, the project follows an ordered sequence of independent phases. At the end of each phase the project team completes a review or sign off resulting in high cost of changes especially in the late phases. The Waterfall is document driven. It places a high value on on planning. This up-front planning reduces the need for continuous planning as the project progresses [4]. Table 1 shows the benefits and drawbacks of waterfall model:

Table 2.1: The benefits and drawbacks of the waterfall methodology

Benefit	Drawback
<ul style="list-style-type: none"> • Output is defined up-front • Strict control • Can work well with technically weak or inexperienced staff • Work well when the quality of requirements is high 	<ul style="list-style-type: none"> • Phases are disjointed • Amount of documentation required • No real output until the end • Difficult to fully specify at the start of the project • Reliance on documentation change is difficult

The increasing of time-to-market pressures and advances in software development tools and technologies, led us to the iterative processes of the 1980s and 1990s, for example: spiral, Rapid application development (RAD) and rational Unified Process (RUP). In the late 1990s and through the current decade, a light-weight, more adaptive software process models appeared, assuming that it was simply more cost effective to write the code quickly, have it evaluated by customers in actual use [19].

In 2001, the creators of many of the agile software development methodologies came together and created an Agile Manifesto¹ summarizing their belief that there is a better way to produce software. Even today it does an excellent job of synthesizing and defining the core beliefs underlying the movement:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Customer collaboration over contract negotiation
Working software over comprehensive documentation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

¹ <http://agilemanifesto.org/>

2.8.2. Agile software development

Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams¹. Agile methodologies promote early and continuing interaction with customers to assert that the software is in line with their requirements, through the continued provision of models in short periods of time [31].

Instead of wasting a lot of time working on the requirements, teams focus on the periodic construction of and delivering the software. Early delivery serves to test the requirements, and it drives risk out by proving or disproving assumptions about integration of features and components. The user community will not wait breathlessly for months, hoping the team is building the right thing. At worst, the next checkpoint is only a week or so away, and users may be able to deploy even the earliest iterations in their own working environment [20].

Agile suggest starting look at software delivery from your customer's point of view, then the good things start to happen [30]:

1. You break big problems down into smaller ones.
2. A week (time unit) is a relatively short period of time to do everything. So, you have to break big, scary problems down into smaller and simpler ones.
3. You focus on the really important stuff and forget everything else.
4. Most of what programmers traditionally deliver on software projects is the documentation which has a little or no value to the customer. Sure, you need documentation, but in support of working software.
5. You make sure that what you are delivering works.
6. Delivering something of value every week implies that what you deliver had better work. That means a lot of testing, early and often.
7. You go looking for feedback from your customer.
8. You change course when necessary.

2.8.3. Agile planning

Agile planning depends on three simple truths:

1. It is impossible to gather all the requirements at the beginning of a project.
2. Whatever requirements you do gather are guaranteed to change.
3. There will always be more to do than time and money will allow.

Planning an agile project is like preparing for a busy long weekend. In agile, the master story list is your project to-do list containing all the high-level features your customer will want to see in their software. It's prioritized by the customer and estimated by your development team. The iteration is engine for getting stuff done on an agile project, it is a one- to two-

¹ http://en.wikipedia.org/wiki/Agile_software_development, Retrieved: 14 Oct 2012

week period where you take your customers' most important stories and transform them into running, tested software [30].

Extreme programming (XP) is the most popular of the Agile approaches and is particularly suited to the dynamic world of Internet development. It accepts change as a fact of life. Small- to medium-sized teams work closely with customers, in favor of rapid iterations that provide direct, real results to the customer [4].

Key practices of XP include the following [5]:

- A team of five to ten programmers work at one location with customer representation on-site.
- Development occurs in frequent builds, and delivers incremental functionality.
- Requirements are specified as user stories, each a chunk of new functionality the user requires.
- Programmers work in pairs, follow strict coding standards, and do their own unit testing. Customers participate in acceptance testing.
- Requirements, architecture, and design emerge over the course of the project.

The Waterfall methodology works from the premise that change is fundamentally bad to be avoided, controlled, and managed. It is good to desire to control change as early as possible in the development cycle because costs tend to rise rapidly as the product is implemented. XP uses techniques and practices to effectively flatten the cost of change throughout the project. It allows for change in the project lifecycle and improves quality by resolving defects early and provides constant feedback [4].

2.9.UML Notation

UML distinguishes between the notions of model and diagram. A model contains all the components of the system, and the diagram is a particular visualization of certain types of components from a model, exposing in some cases detailed information [6]. There are several diagram types in the UML definition, but this text uses only the class diagram.

2.9.1. Class Diagrams

The class diagram is the core for a UML model, and it shows the important abstractions in the system and how they relate to each other. The basic components found in class diagrams are class icons and relationship icons. UML represents individual classes as solid rectangles that may be divided into three parts or compartments. The first part contains the name of the class. The second and third parts are optional and may be used to list the attributes and functions of the class.

2.9.2. Relationships

Besides the individual classes with their attributes and functions, class diagrams also represent the relationships that exist between dependent classes. UML identifies several types of relationships with their respective graphic representations as shown in figure 2.9. An

association between two classes is depicted by connecting the classes with a straight line. Aggregation is a special form of association that is used to show that one object is at least partially composed of another. An aggregation with a hollow diamond represents that the whole object maintains a pointer or a reference to its parts. If the diamond is filled, then the diagram shows that the aggregation is by value, i.e., the whole object declare an actual instance of the part object within itself.

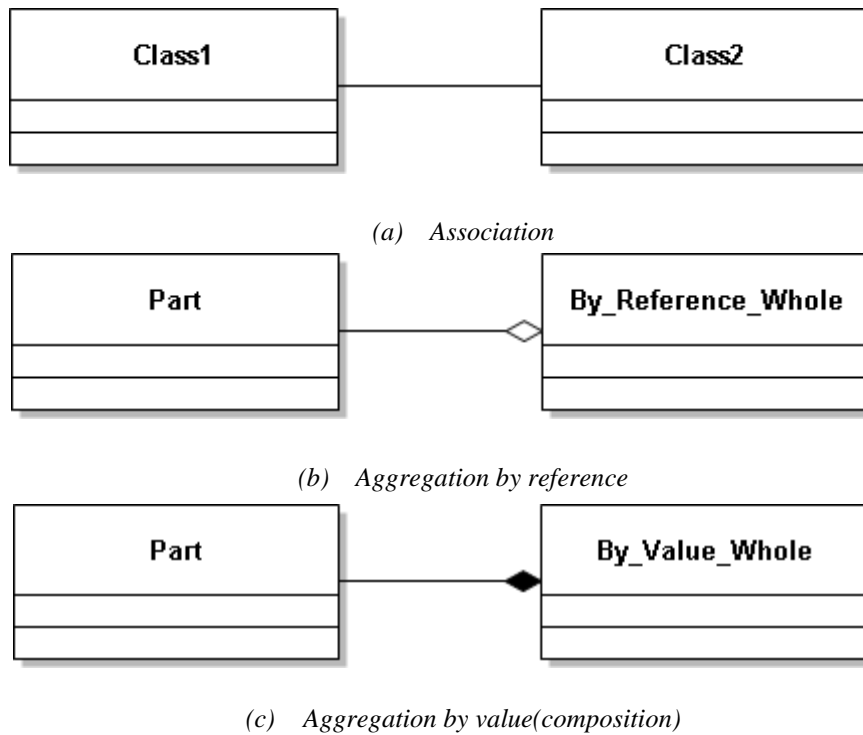


Figure 2.9 : Types of UML relationships

Chapter 3

Related Work

This chapter will discuss various works related to this project. The related work will be divided into three sections: the first section will introduce two types of 3D web sites, and the second will talk about 3D web browser, and the last will explain the 3D WebCube model prototype which is the basic prototype that the library implementation depends on.

The related works is not so close to this study because there is no library - until the writing of this research - has been developed in 3D websites building except three.js and few other weak libraries, which are generic and low-level libraries for WebGL. These libraries are not specialized and could be used in any 3D applications like games, simulation, animations, etc...

As a result the required code to build 3D applications using three.js is less than the code written using WebGL directly but it still very large.

So this research produces the first library specialized in 3D websites development allowing web developer to build a complete 3D websites with few simple lines of code. And because of this specialization, the research focuses on 3D websites and browsers as a related works.

3.1. 3D websites

3D websites has two main categories, the first is the sites with totally 3D environment that allow us to surf in a real or virtual 3D world, and the other is the sites that use 3D styles to add some attraction and new look to the contents.

3.1.1. 3D Environment

The first related work is Google Earth that provides massive information of earth maps captured by satellites. It is one of the most used products in the world. This product enables users to explore cities, building, and streets in 3D environment [16]. With Google Earth version 6.2 new features added such as 3D street touring which enables users to walk through buildings.

Figure 3.1 depicts how Google Earth 6 views 3D buildings and streets.

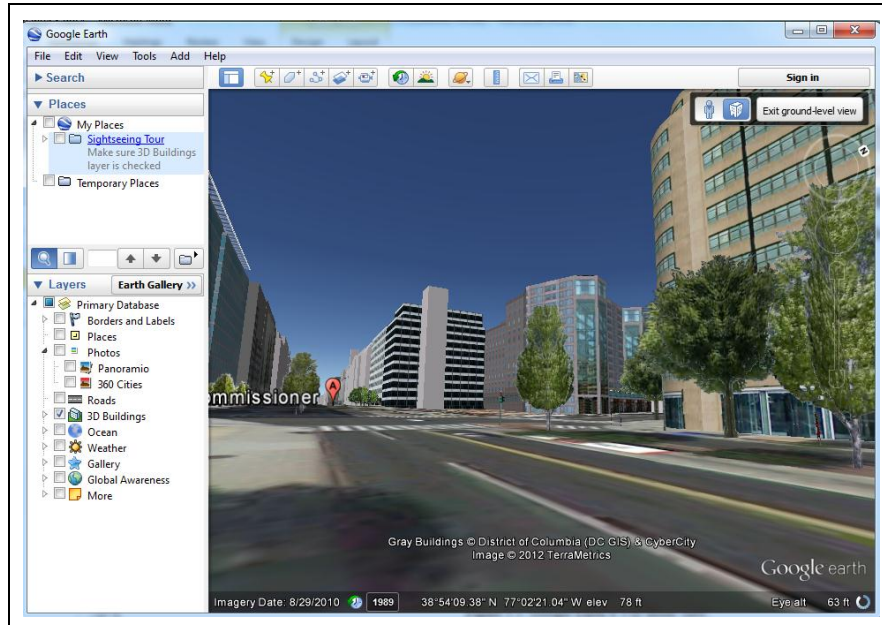


Figure 3.1: Google Earth 6.2, 3-D street view.

Massively Multiplayer Online Games (MMOGs) has grown fast, World of Warcraft (WoW) game shown in Figure 3.2 is one of the most famous Internet games based on MMOGs. WoW allows user to choose a persona and to join fighting adventure with other users in a 3D environment.



Figure 3.2: World of Warcraft web game.

3D interaction environment such as SecondLife shown in Figure 3.3 could be a good educational environment over the Internet [27]. SecondLife was launched in 2003 as other world corresponding real human world. Nowadays, millions of users join and tour this 3D interaction world. Users in this 3D interaction world can buy and sell everything such as houses, cars, lands etc. Moreover, they can contact each other as in the real world.



Figure 3.3: People gathered at SecondLife.

Noting that all previous examples need a pre-installed application to run the 3D word environment other than web browser.

3.1.2. 3D Style

An example of this type is a search website that views the results of user search in 3D frames. Figure 3.4 depicts search.spacetime.com search site browsing the search results in 3D perspective.

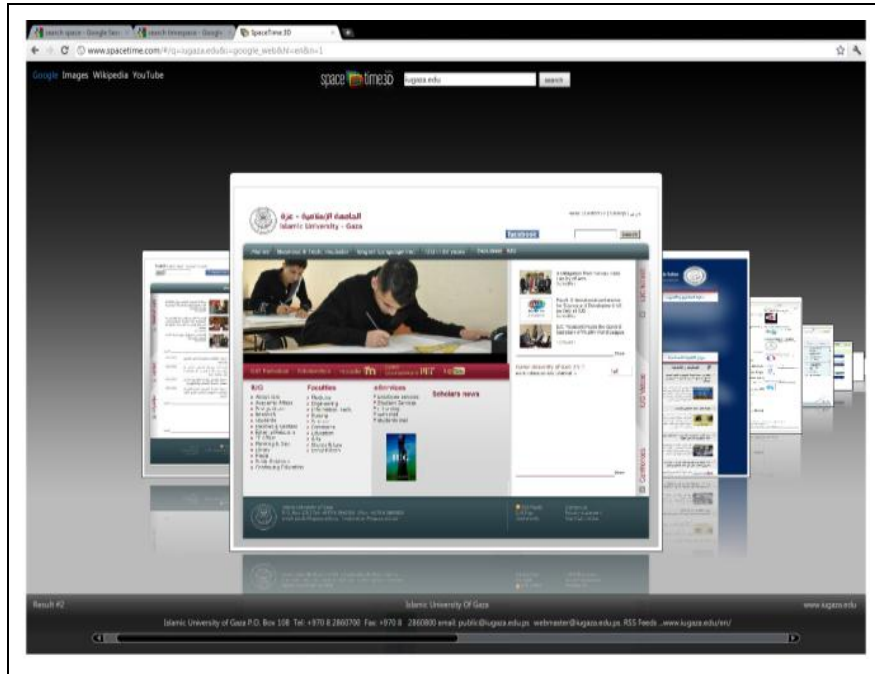


Figure 3.4: 3D web presentation as a style.

Another famous example is the tremendous variety of 3D image galleries; figure 3.5 show a 3D image gallery example from activeden.net.

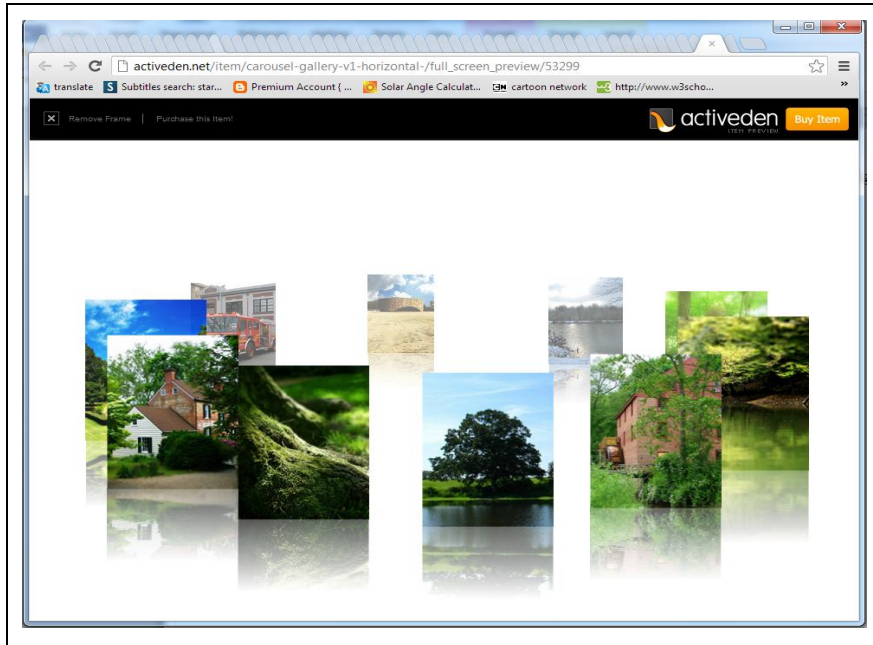


Figure 3.5: 3D Gallery example.

Most of the 3D styles on the web are flash animation so it will require adobe flash player plugin added to the web browser to be able to view the 3D contents.

3.2. 3D Web Browser

This section will talk about some attempts to make web browsing in 3D look, this attempts views the regular 2D flat pages in a 3D look, using special browser without changing the contents origin of the web page.



Figure 3.6: 3D Web Browser Pro.

The first example is 3D Web Browser Pro¹ software which available only for Mac iOS devices, this browser opens multiple web pages at the same time allowing the user to rotate a cube to view another page, figure 3.6 shows the cube in rotation mode.

Another 3D web browser example is the CubicEye², the CubicEye visual interface resembles the inside walls of a box, with each panel being a CRT display. You see a different display in each direction. Since each display could consist of five more displays, it is possible to display hundreds of live web pages, at the same time, on a single monitor. CubicEye enables clients to manage and manipulate multiple screens of data in real time in this unique format. Figure 3.7(a) show the CubicEye web browser, unfortunately this browser stop at its first beta version since 2005, and it seems it didn't find the expected success, may be because of the new tapped browser that allow simultaneous multiple pages in the same window.



Figure 3.7: CubicEye Web Browser.

Another feature in CubicEye prototype browser that it can view 3D Object, figure 3.7 (b), but this feature require a server side components to make it possible – which is not implemented –, so if you want to add a 3D contents to your site, you must host your site in a CubicEye enabled web server and the client must have CubicEye web browser, this may be the main lack in this idea.

3.3. 3D WebCube Prototype

This section explains in through details the 3D WebCube Prototype that the implementation depends on.

3.3.1. WebCube Model and Its Contents

The main WebCube structure consists of five surfaces: floor, ceiling, and three walls (right, left, and back; the front is omitted to make an entrance to the Web Cube). While these

¹ <https://itunes.apple.com/qa/app/3d-web-browser-pro/id443764787?mt=8>, Retrieved 20 Oct. 2012

² <http://cubiceye.net/index.php>, Retrieved 20 Oct. 2012

surfaces are holders of content components also, they limit WebCube 3D space. Just as a webpage is limited by its four boundaries WebCube is limited by these surfaces. Furthermore, users navigate from one WebCube to another through links.

A WebCube may contain any number of content components Figure 3.8 shows an example of WebCube with various such content components. These components will be described individually next.

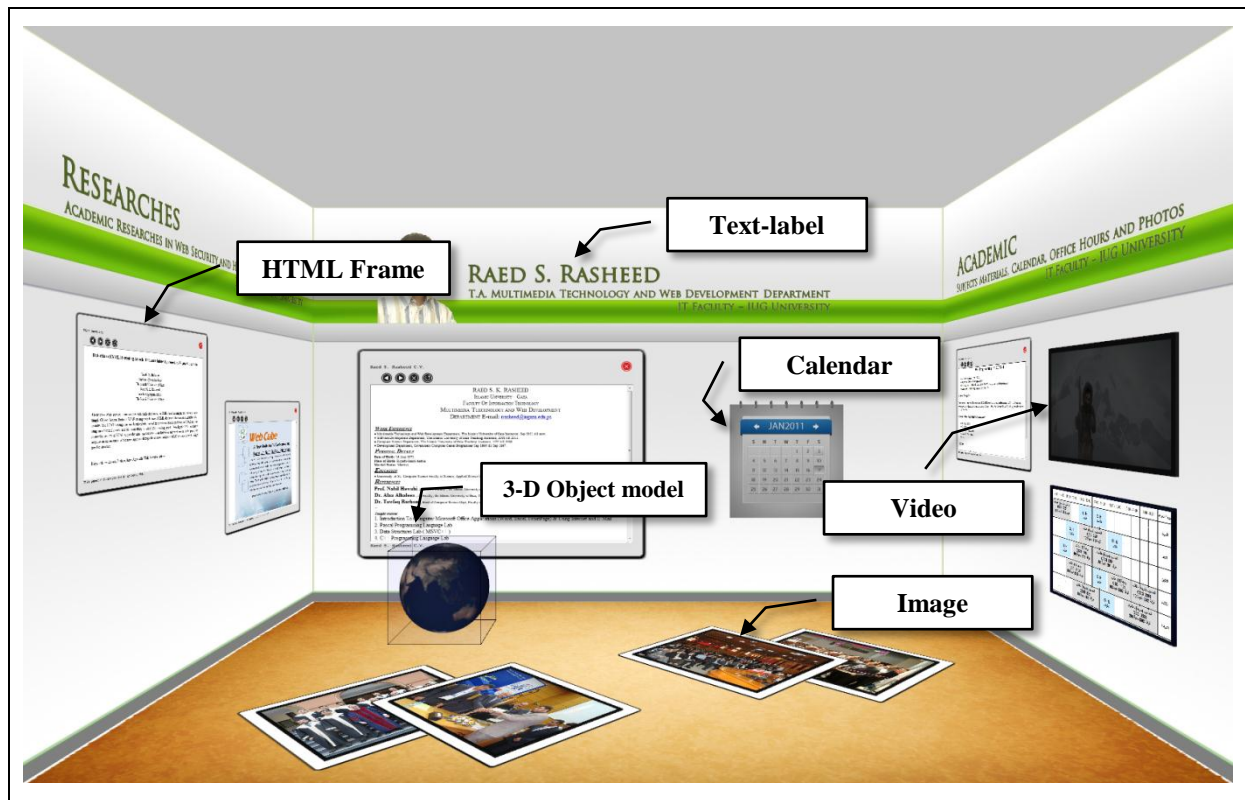


Figure 3.8: Example of WebCube and its components.

WebCube will be a text file on the server side that containing details of the design and content. This file will be loaded via HTTP protocol and rendered at the client side by the Internet browser.

- **Text-label:** is the static text used to label or describe something, as seen in Figure 3.8 the text-label “RAED S. RASHEED” and “T.A. MULTIMEDIA TECHNOLOGY...” on the front wall describes the owner name and his title.
- **Image:** this component consists of two sides the front side which contains the picture or image itself and the back side that contains the image details. Figure 3.9 depicts the front and the back of an image.

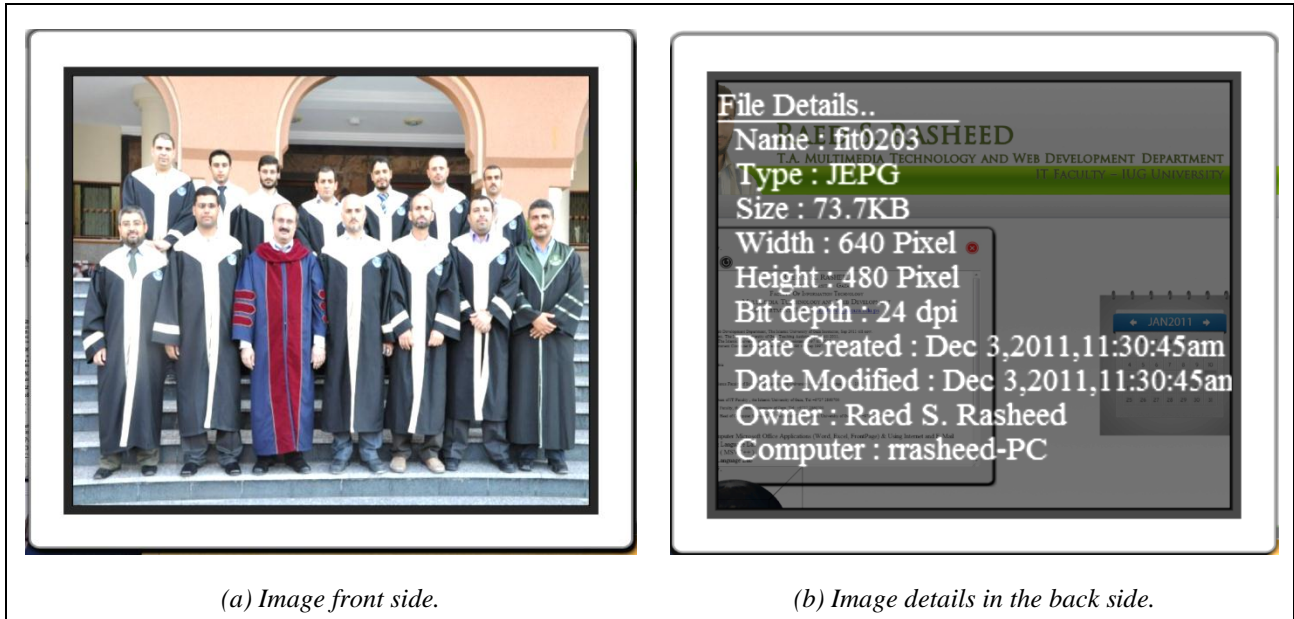


Figure 3.9: WebCube image component.

- **Video:** as the image component, this component consists of two sides the front side which contains the video itself and the back side that contains the video details. Figure 3.10 depicts the front and the back of a video.



Figure 3.10: WebCube video component.

- **Calendar:** a very simple calendar enables users to explore the date. It contains years, months and days Figure 3.8 depicts a simple calendar on the front wall.
- **Link:** a hyperlink to another WebCube used to load another WebCube and its contents. The link can be used also to select another WebCube component in the same Web Cube.
- **Gallery:** a photo album used to arrange photos or pictures together and scroll them. Figure 3.11 depicts the implemented gallery.



Figure 3.11: WebCube gallery components.

- **HTML-Frame:** is a container of textual information or HTML document. This component can be a simple web browser inside the Web Cube. Figure 3.12 depicts the WebCube frame component.
- **3-D Object Model:** a model of any object that can be manipulated in 3-D, the object can be created using any 3-D modeling software and exported as an object file (.obj) then converted to JavaScript file (.js) using converter programs. The JavaScript file can be loaded to the WebCube easily. Figure 3.13 depicts car object created using 3-D Studio Max software and loaded in the WebCube after converting it to JavaScript file.

The WebCube environment and its content developed using WebGL, JavaScript, HTML5 and CSS3. The canvas element in HTML5 is used to draw 2-D graphics on the web. Drawing 3-D graphics on the canvas element needs the WebGL library that was implemented in JavaScript.

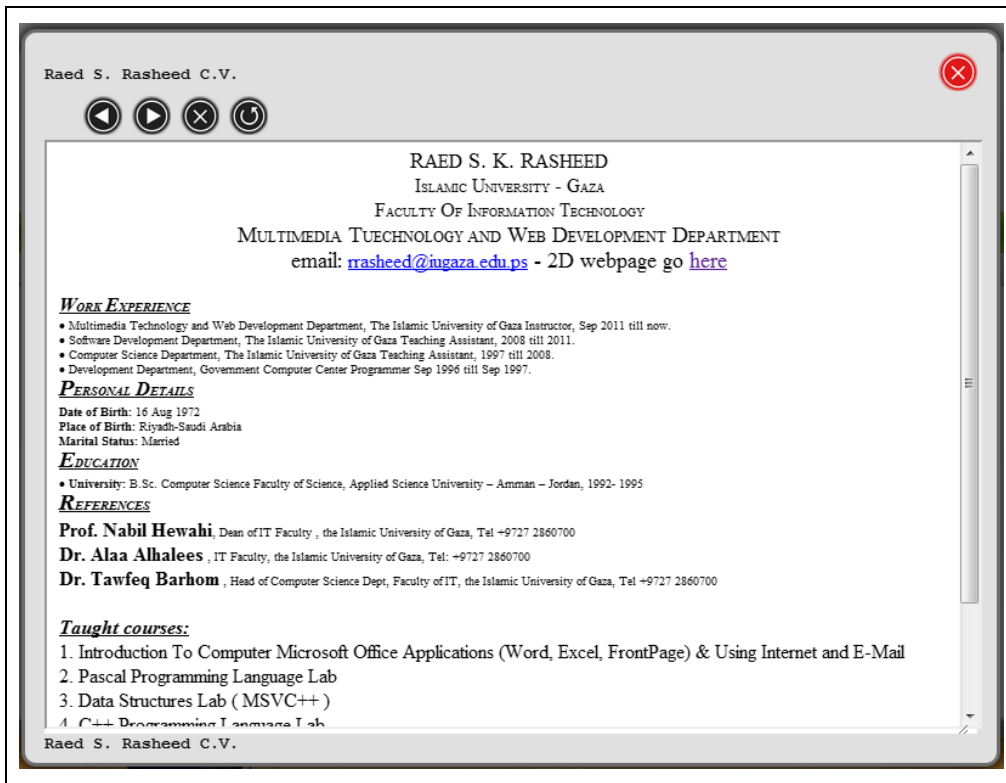


Figure 3.12: WebCube HTML frame components.



Figure 3.13: WebCube 3-D object model components.

Chapter 4

3D WebCube Library Implementation

Creating a well-architected WebCube library requires meeting the needs of the web developer and web client needs. A WebCube library is responsible for object creating, customizing, and events handling. WebCube library must have a coherent architecture that is reliable and tested. This chapter covers the implementation details of the WebCube library.

This chapter uses the Unified Modeling Language (UML) [6] as a software engineering tool that supports the development of the library classes.

4.1. WebCube Class Diagrams

This section will discuss the internal structure of the library classes with detailed explanations for their attributes and functions.

4.1.1. WebCube Class

The WebCube is a limited space in three dimensions that contains 3D web content components. Users navigate the WebCube and interact with its contents, and also navigate among WebCubes. Figure 4.1 shows the structure of WebCube surfaces, any surface can hold zero or more WebCube components in any position and size.

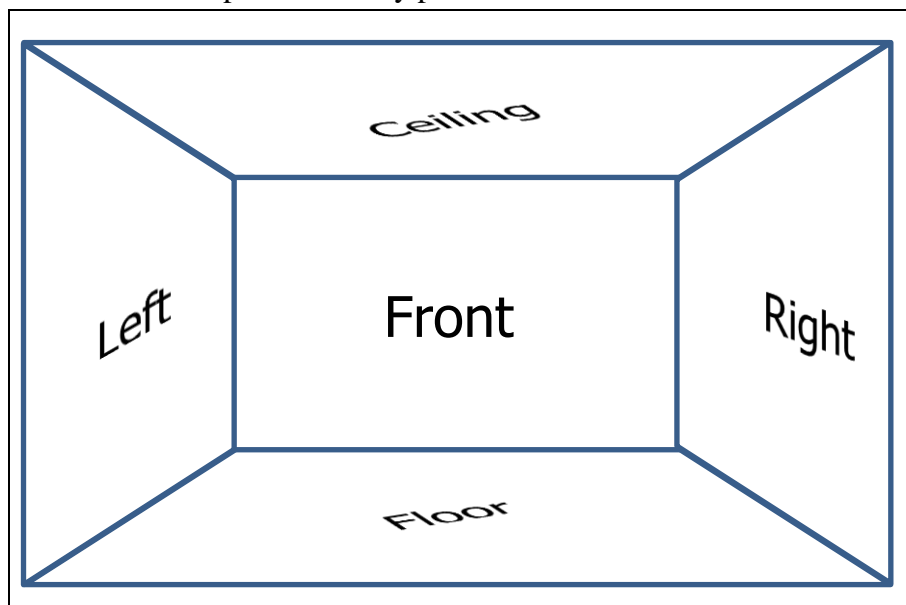


Figure 4.1: WebCube surfaces

WebCube class is the main class in the library which an instance of it must be created as the first step, *WebCube* constructor creates five instances of the *Side* class - front, left, right, ceiling and floor - to build the scene, and then the designer ask it to add WebCube components by calling the creators methods.

The class diagram shown in figure 4.2 contains the main attributes and functions within *WebCube* class which will be discussed.

The *WebCube* boundaries – width, height and depth – are represented in the attributes section. The attributes *screen_w* and *screen_h* represent the dimensions of the browser window which is also the borders of the scene, *screen_d* attribute represent the depth of the cube which is customized by the user in the *WebCube* constructor parameters. The *WebCube* has five sides so it uses the *sides* array to store five references of the *WebCube* sides of the type *Side*. *WebCube* has a reference for each component in the cube, it uses the *objects* array to store a reference for each new *WebCube* component, and this reference will be used to invoke the required function from that object.

To handle mouse events, *WebCube* must know the component that the mouse moves on, so it uses the attribute *selected* to refer to that object.

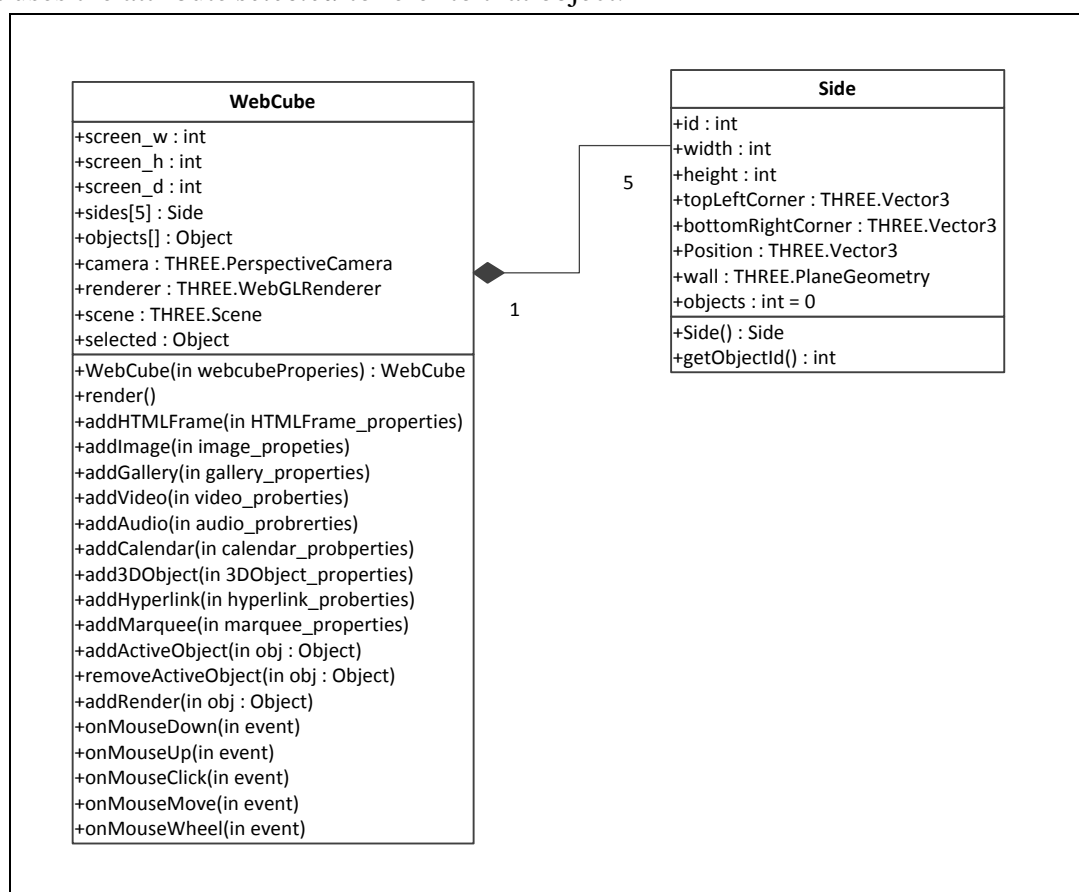


Figure 4.2: *WebCube* and *Side* class diagrams

The last three attributes *camera*, *renderer* and *scene* are classes of the *THREE.js* library which discussed in chapter two.

WebCube is responsible for object creation and events handler, so the functions in *WebCube* class can be grouped into two categories beside the constructor and the renderer, these categories are component creators and event handler functions.

The *constructor* as mentioned above creates the five sides of the cube and compute the initial values of the cube dimensions, its input parameters are the five textures of the side surfaces and the depth of the screen, each side of the cube surface could be a solid color or an image.

The *render* function invoked automatically at the end of the *requestAnimationFrame* function to redraw the scene, noting that the WebCube *render* function invokes the *render* function in all objects in the *objects* array, the *requestAnimationFrame* function is invoked automatically all the time by the browser.

The first category in the *WebCube* functions is the component creators which used by the page designer to add any number and type of WebCube components to the page after specifying the components properties.

The second group is the event handlers which handle mouse move, mouse click, mouse down, mouse up and mouse wheel events, each one of these functions after executing its body invokes the same function in the *selected* object if exist.

To create a WebCube you simply write this line code:

```
var wc = new WebCube(  
0, // use image texture, to use colors we write 1  
"images/front.jpg", //front side texture image path, when using colors  
we "images/left.jpg", //write the RGB formula, for ex.:0xFF0000 for red  
"images/right.jpg",  
"images/ceil.jpg",  
"images/floor.jpg",  
0.95 ); // cube depth relative to scene height
```

4.1.2. Side Class

Figure 4.3 shows the main attributes and functions of the *Side* class. WebCube creates five instances of *Side* Object, each instance represent a specific side and has a unique position in the cube, because of that the *Side* class must has an identity attribute, so it uses the attribute *id* to represent its position in the cube - 0: front, 1: left, 2: right, 3: ceil, 4: floor -. The dimensions of the side depend on its position in the cube, for example; the height of the ceiling side is the same as the width of the left side and so on. The *width* and *height* attributes represent the dimensions of the side plane, these dimensions computed depend on side *id*, *screen_w*, *screen_h* and *screen_d* attributes.

Each component added to the WebCube belongs to a specific side, and its position in the space is relative to that side position. *Side* class contains *TLC* (*topLeftCorner*), *BRC* (*bottomRightCorner*) and *position* attributes shown in figure 4.3, these values will be used by the WebCube class to specify the position for each component in the space. The *wall* attribute represent the visible surface of the side while the *objects* attribute represent the number of objects belongs to this side.

The *Side* has only two functions, the *constructor* which build the surface in the appropriate values and the *getObjectId* which return the *object* attribute value to be the id of the new added WebCube component on this side and then increment it by one, noting that the component uses this value to know its third dimension place to avoid plane interlacing, this is done by reserving three pixels depth for each object, so the third dimension position of any new component is $3 * object$.

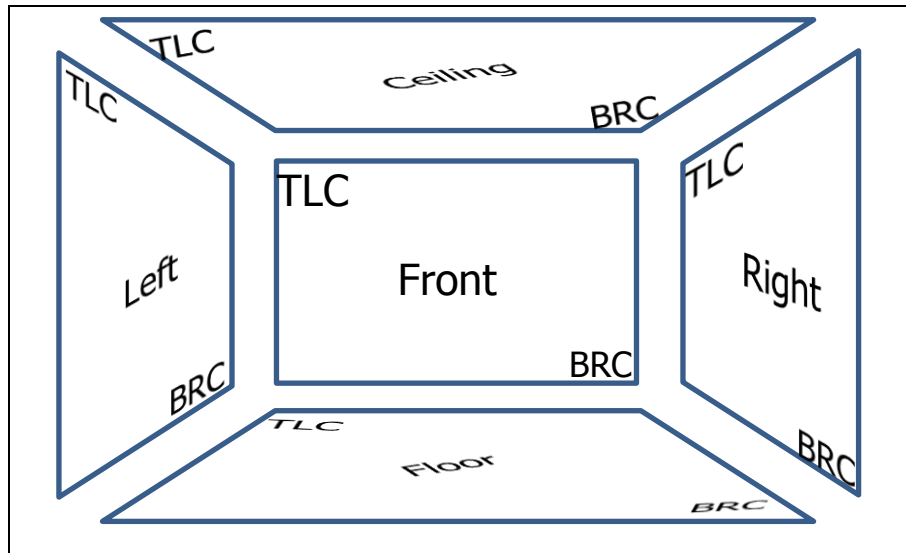


Figure 4.3: TopLeftCorner and BottomRightCorner for each side

Figure 4.4(a) shows two planes in the same third dimension position which will cause destruction in the interlacing area, while in (b) the planes in different third dimension position eliminate this problem, which is done after allocating a free space depending on *getObjectId* return value.

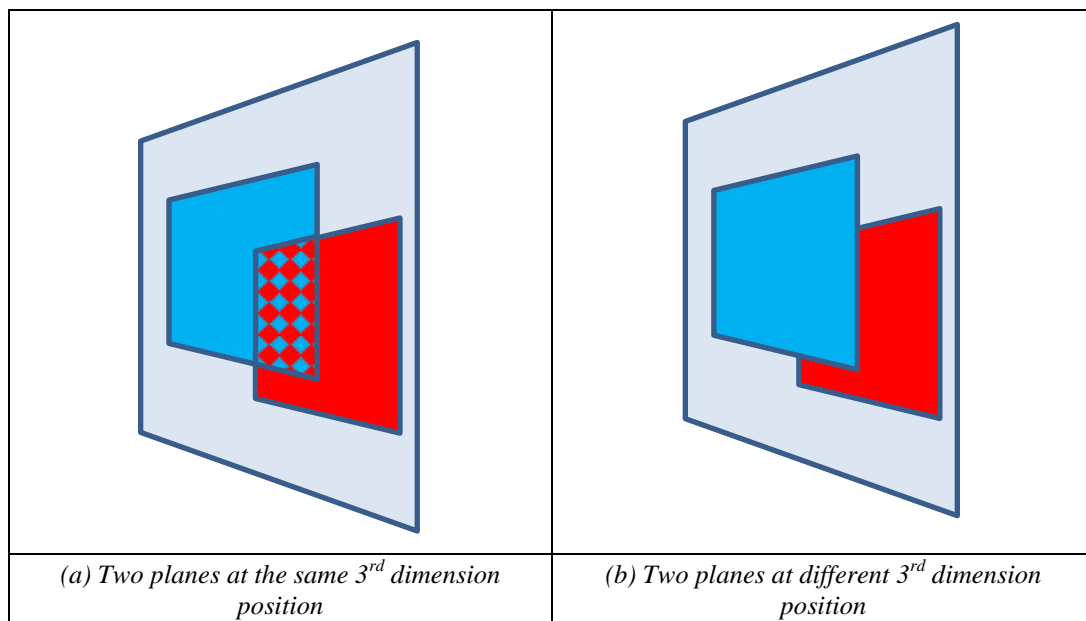


Figure 4.4: Objects interlacing problem

4.1.3. WebCube component

Each WebCube component class contains a common attributes and functions added to its specific attributes and functions.

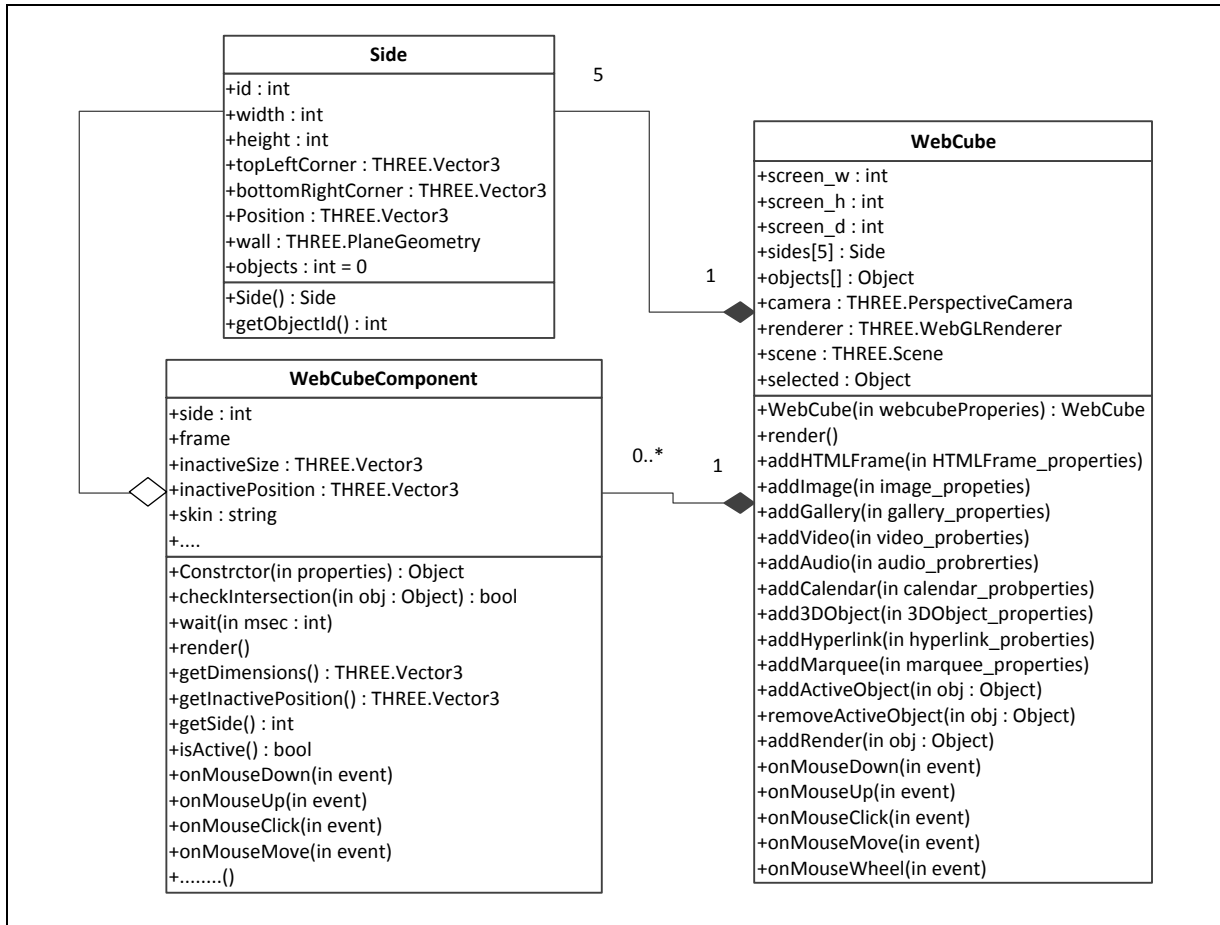


Figure 4.5: WebCube Component class

The first common attributes as shown in figure 4.5 is *side*, which represent the side number that the component belongs to.

Most components of the WebCube have two states, the inactive state, when it reside on a specific side, and the active state, when the user clicks on, this will brings it to the inner space of the cube to interact with the user. For each state, the component needs to store its size and position.

The *sideId* attribute used to refer to the *Side* object which will be used to compute the actual size and position in space for the component. The second one is *frame* which is the geometry that represents the component. Another common attributes is the *inactiveSize* which represents the dimensions of the component in the inactive state, and the *inactivePosition* is the relative component position on that side.

Each component must allow the developer to change its appearance, so it use the *skin* attribute to store the path of an XML file that contains the definitions of color scheme, fonts and images of that component.

In the functions part, there is the *constructor* which takes the component properties as input parameters to create the component and return a reference to the *WebCube* class.

As the mouse moves in the scene the *WebCube* invoke *checkIntersection* function in each component in the *objects* array to check whether the object that the mouse currently moves on is one of these component objects, since each component represented by many small objects, if the return value is true then this component will be the selected one.

The *wait* function is used to execute a command after some milliseconds, this function use the *render* function to know the current time since the *render* is invoked continuously by the *WebCube render* function as mentioned above.

While the component in the inactive state which is known by invoking the *isActive* function the responsibility of its moves rely on the *WebCube* class, so this class uses the *getDimensions* and *getInactivePosition* to move the component within its side borders, the side of the component is known by invoking the *getSide* function.

When a mouse event occurs, say mouse click, the *WebCube* class executes the *onMouseClicked* function and then invokes the *onMouseClicked* in the selected component, and this thing is done for each other mouse events.

4.1.4. HTMLFrame Component

HTMLframe used to add regular html webpages to the *WebCube* site, the designer can customize the colors, buttons, title and size of the frame. This component can be a simple web browser inside the *Web Cube*. Figure 4.6 depicts the *Web Cube* frame component.

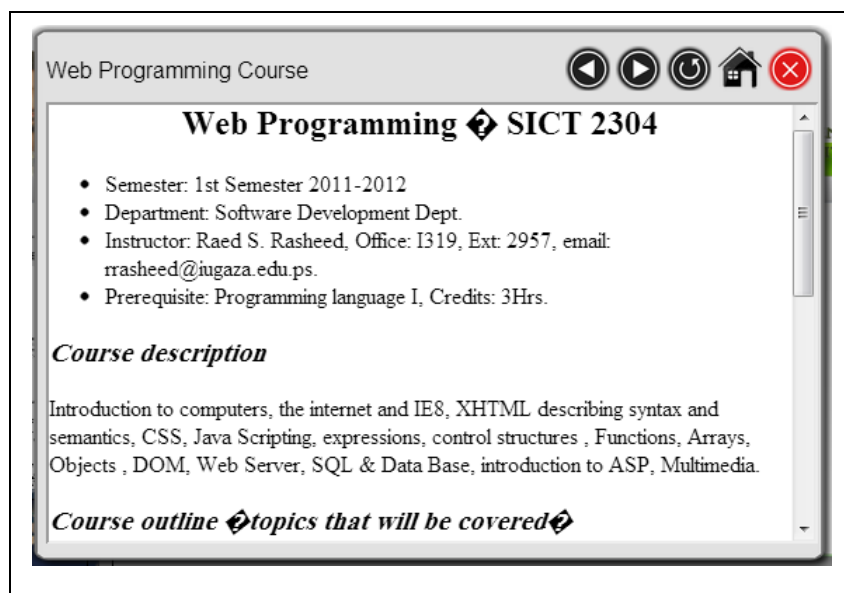


Figure 4.6: Sample *HTMLFrame* component

Figure 4.7 shows the structure of the *HTMLFrame* class, in the attributes section, the *frameID* attribute is unique for every *HTMLFrame* object in the *WebCube* because it is used as the ID of frame DIV container, this ID used to show, hide and move the DIV.

The *frame* attribute is *FrameGeometry* type, which is a collection of planes grouped together to build the inactive *HTMLFrame*.

The *div* attribute is the html code of the frame container div, which represent the tables, icons, title, iframe URL and colors, this code extracted from the skin XML file and from the *activeSize* and *activePosition* attributes.

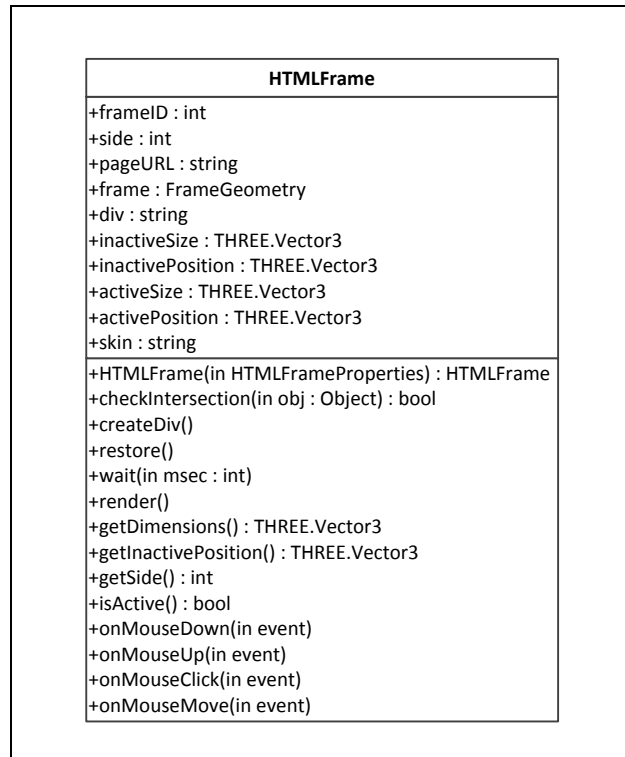


Figure 4.7: *HTMLFrame* class

There is some new added function in this class like *createDIV* which is invoked when the user activate the HTML Frame by clicking it, and the *restore* function return the frame back to its inactive state.

Here is the code line needed to add HTML Frame to the WebCube.

```

wc.addHTMLFrame (
"acadimic", // frame ID
"rrasheedWPC.html", // page URL
"./images/rrasheedWPC.jpg", // inactive image represent the contents
"Web Programming Course", // page title
"./skin/", // skin.xml Path
2, // side number (right side in this case)
new THREE.Vector3(0.5,0.3,1), // inactive size relative to the side
new THREE.Vector2(0.1,0.35), // inactive position relative to side TLC
true, // inactive movable ??
new THREE.Vector3(0.4,0.5,1), //active size relative to scene dimensions
new THREE.Vector3(0.3,0.25,0), // active position
0); // the rotation angle in the inactive state
  
```

4.1.5. Image component

This component consists of two sides, the front side which contains the image itself and the back side contains the image details. Figure 4.8 (a) depicts the front side and (b) the back side of an image, the image could be resized by the user using the resize button in the bottom right corner or even rotated using the rotate button in the top left corner.

As in all WebCube components the user can use skin file to customize the colors, size, and position noting that image comment, backside and buttons are all optional and customizable.



Figure 4.8: Image component

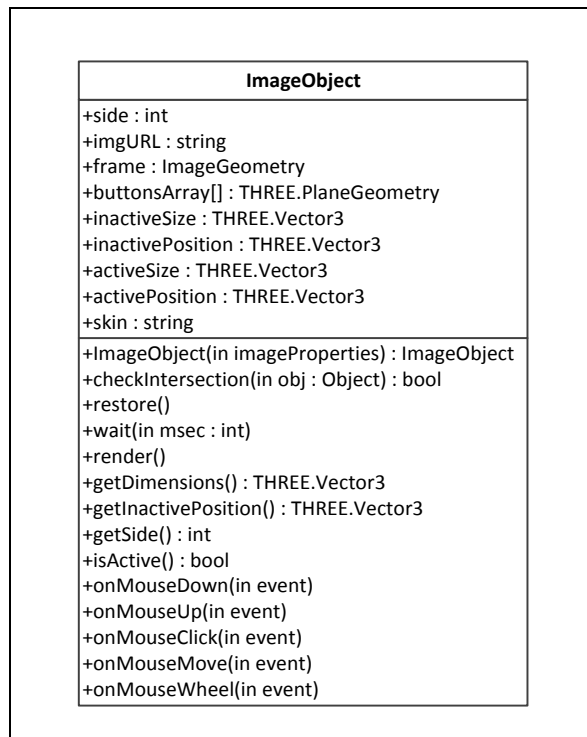


Figure 4.9: ImageObject class

The class structure shown in figure 4.9 shows the added attributes and function for this component, the new attribute is the *buttonsArray* which represent the buttons on the image component, these buttons are: restore, zoomBack, flip, resize and rotate. The restore button used to return the image to the inactive state on the cube side, while zoomBack button which is optional used to return the image zoom into 100% since the user can zoom in or out using mouse wheel, and the flip button which is also optional used to view the back side of the image that contains image details written by the designer.

And the code line needed to add an image component to the WebCube like:

```

wc.addImage (
"./images/my_image.jpg" ,// image URL
"", // image title, write nothing to remove title
section
"./skin/", // skin.xml Path
1, // side number
new THREE.Vector3(0.4,0.2,1) ,// inactive size relative to the side
new THREE.Vector2(0.55,0.67), // inactive position relative to side TLC
true, // inactive movable ??
new THREE.Vector3(0.3,0.3,1), // active size relative to scene
dimensions
new THREE.Vector3(0.2,0.4,0), // active position
3, // wanted buttons (011):flip=no, zoomBack=yes,
restore=yes
30, // the rotation angle in the inactive state
"Some Info"); // image back side info

```

4.1.6. Video component

As the image component, this component consists of two sides the front side which contains the video itself and the back side that contains the video details. Figure 4.10 depicts the front and the back of the video component. The video component has new added controls such as play, pause, time bar and volume bar. Figure 4.11 shows the *VideoObject* class.

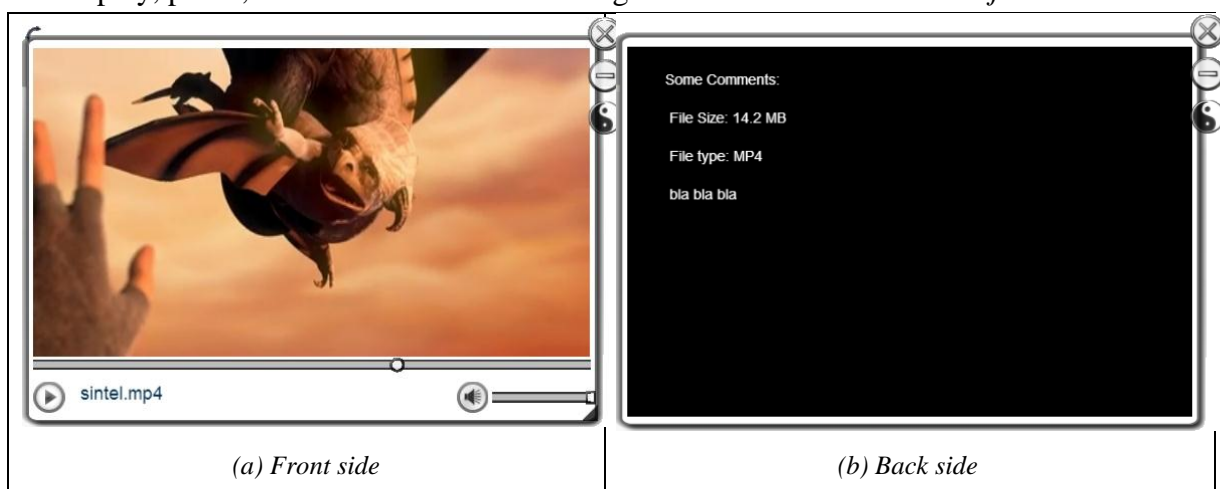


Figure 4.10: Video component

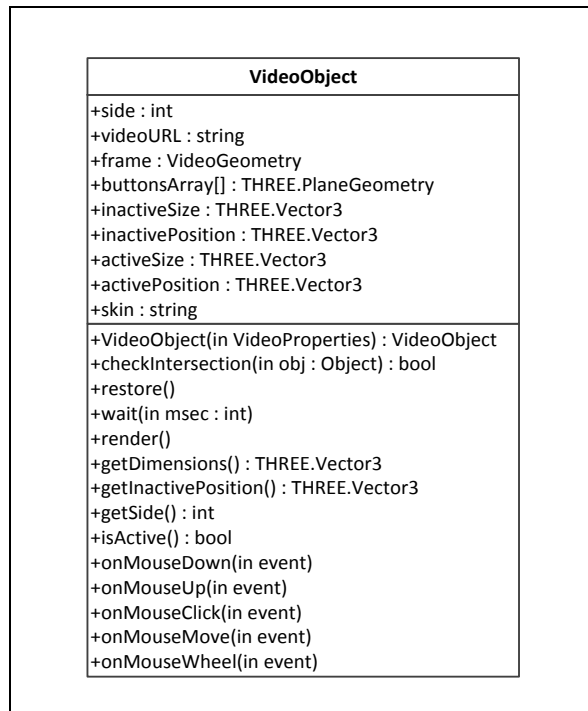


Figure 4.11: VideoObject class

And here is the code line needed to add a video component to the WebCube.

```

wc.addVideo (
"videol",           // video ID
"./sintel.mp4",    // video URL
"sintel",          // video title
"./skin/",         // skin.xml Path
2,                // side number
new THREE.Vector3(0.35,0.28,1), //inactive size
new THREE.Vector2(0.62,0.35), // inactive position
true,             // inactive movable ??
new THREE.Vector3(0.4,0.5,1), // active size
new THREE.Vector3(0.1,0.1,0), // active position
7,               // wanted buttons
0,               // the rotation angle
"Some info ..... ", // video back side info
false,          // Auto start ??
true);          // Loop ??
  
```

4.1.7. Audio component

This component is the same as video component but without the video area, and the class has the same attributes and functions with replacing the video URL by Audio URL, figure 4.12 shows the audio component.

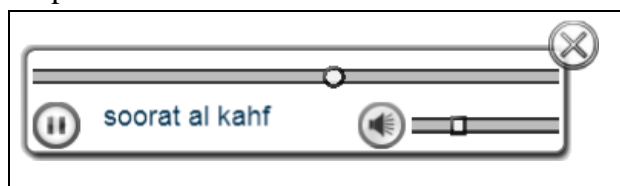


Figure 4.12: Audio component

4.1.8. Gallery Component

A photo album used to arrange photos together and scrolling them. Figure 4.13 depicts the implemented gallery, each gallery has a title, “Birds” in this figure, and each image in gallery could have an optional title, “American eagle” in the figure. The user can scroll it left or right using the appearing buttons or using the keyboard arrows, and by clicking the “x” he returns the gallery to the inactive state.

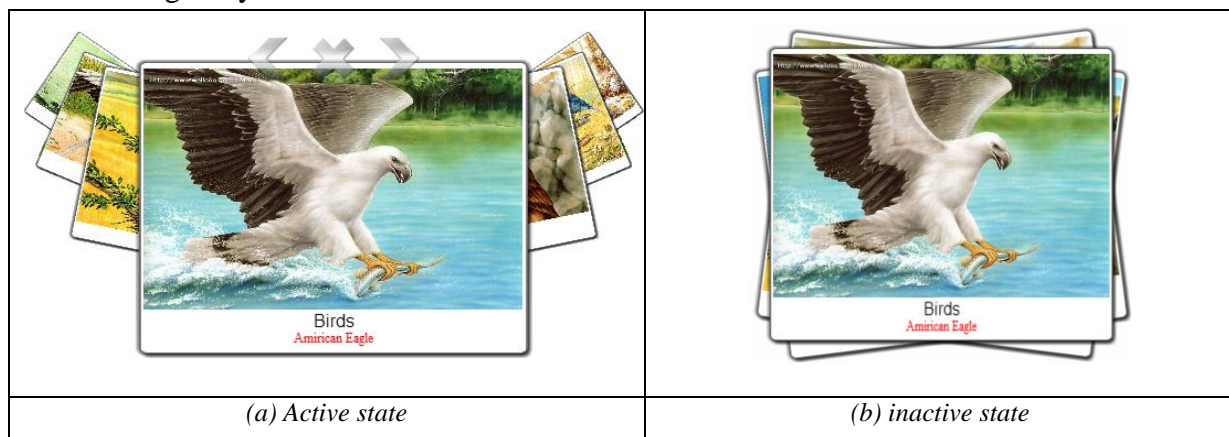


Figure 4.13: Gallery component

Figure 4.14 shows the *GalleryObject* class, where new added attributes and functions, in the attributes section there is *viewedImages* which represent the number of images displayed at once which is seven in the figure 4.13 (a), and *imagesListXML* is the path of the XML file that contains the images URL that and images titles.

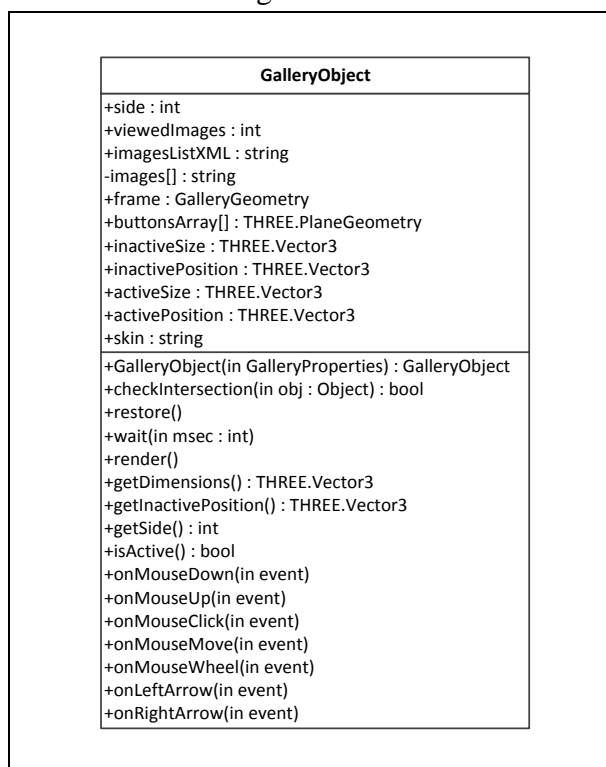


Figure 4.14: GalleryObject class

A Sample of the gallery xml file shown in figure 4.15.

```
<gallery>
  <image>
    <URL>./images/tex/1.jpg</URL>
    <title>Amirican Eagle</title>
  </image>
  <image>
    <URL>./images/tex/2.jpg</URL>
    <title>null</title>
  </image>
  <image>
    <URL>./images/tex/3.jpg</URL>
    <title>null</title>
  </image>
  .....
  .....
</gallery>
```

Figure 4.15: Sample gallery XML file

And here is the code line needed to add a gallery component to the WebCube.

```
wc.addGallery (
  "./gallery.xml",           // XML file path
  "Birds",                  // Gallery main title
  "./skin/",                // skin.xml Path
  0,                        // side number
  new THREE.Vector3(0.22,0.35,1), // inactive size
  new THREE.Vector2(0.7,0.1), // inactive position
  true,                     // inactive movable ??
  new THREE.Vector3(0.25,0.35,1), // active size of the main picture
  new THREE.Vector3(0.3,0.3,0), // active position
  0,                         // the rotation angle
  7);                        //number of viewed images at the same time
```

4.1.9. Calendar Component

A simple calendar enables users to explore the date. Figure 4.16 depicts the calendar showing the year, month and weekdays, the current day is selected - 25th Nov in the figure - it has four buttons to restore, zoom back, next month and previous month.

The CalendarObject class shown in figure 4.17 has an attributes and functions the same as the basic class which is explained above.



Figure 4.16: Calendar component

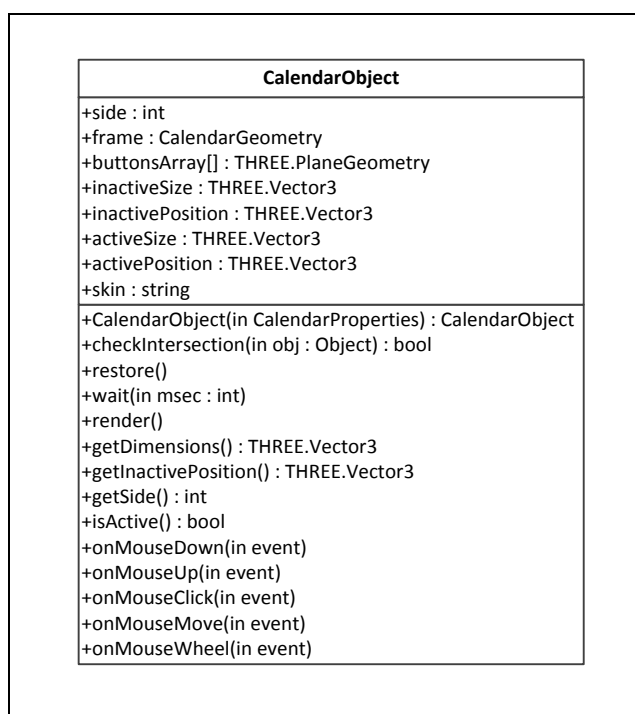


Figure 4.17: CalendarObject class

And to add a Calendar component to the WebCube, write:

```

wc.addCalendar (
  "./skin/",           // skin.xml Path
  0,                   // side number
  new THREE.Vector3(0.2,0.3,1), // inactive size
  new THREE.Vector2(0.65,0.5), // inactive position
  true,                // inactive movable ??
  new THREE.Vector3(0.2,0.3,1), // active size relative
  new THREE.Vector3(0.5,0.4,0), // active position
  0);                  // the rotation angle

```


4.1.10. Marquee Component

A customizable marquee component where the designer controls the colors, font and speed, the designer adds the marquee elements and its destination addresses in an XML file which stored in the *marqueeComponents* array. The *Marquee* class constructor shown in figure 4.18 creates a basic canvas and then in the *render* function scrolls the marquee elements depending on the speed, and between each two elements there is a customizable spacer.

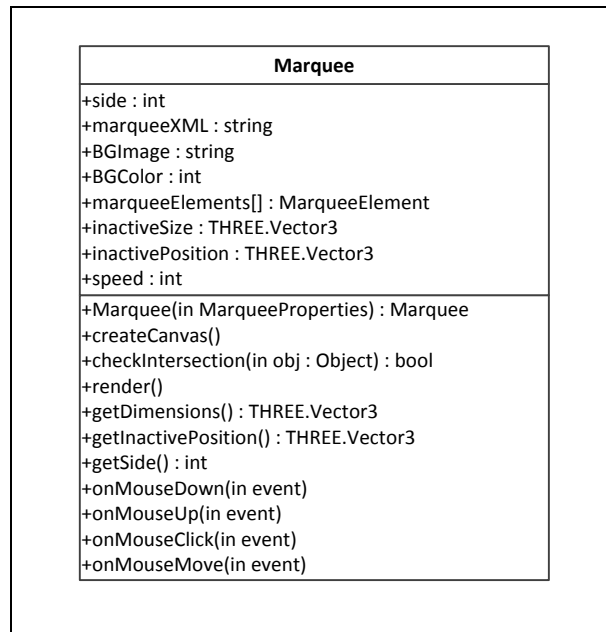


Figure 4.18: Marquee class

And here is the code line needed to add a Marquee component to the WebCube.

```
wc.addMarquee (
0, // side number
"marqueeImage.jpg", // optional background image
0x208000, // optional background color
"./marquee.xml", // marquee texts and destination XLM
path
0x002000, // text font color
0xFFFFFFFF, // spacer color
"Arial", // text font
12, //text font size
false, // underlined?
true, // bold?
false, // italic?
0.5, // marquee opacity
new THREE.Vector3(0.3,0.06,1), // marquee size
new THREE.Vector3(0.3,0.235,1), // marquee position
false, // movable?
0, // rotation angle
4); // scrolling speed
```

A sample of marquee XML file shown in figure 4.19 producing the marquee component shown in figure 4.20, this component has no background image or color so it will be transparent in this case.

```
<marquee>
  <marqueeComponent>
    <URL>http://www.hotmail.com</URL>
    <title>Hotmail mail server</title>
  </marqueeComponent>
  <marqueeComponent>
    <URL>http://www.google.com</URL>
    <title>Google Search engine</title>
  </marqueeComponent>
  .....
  <spacer> - </spacer>
</marquee>
```

Figure 4.19: Sample Marquee XML file

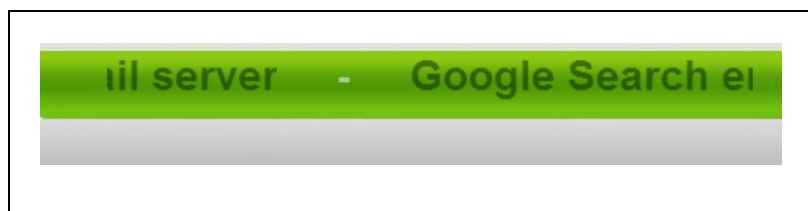


Figure 4.20: Marquee component

4.1.11. Hyperlink component

A hyperlink is a link to the outside world, another WebCube or a regular 2D webpages. In the WebCube Hyperlink component the designer can customize the colors, fonts, background image, background color and opacity, noting that the destination part is optional, so this component could be used as a Label or static image or color box.

The Hyperlink class shown in figure 4.21 is nearly the same as the Marquee class without the marqueeComponents array which becomes only one component represented with title and destination attributes and without the render function.

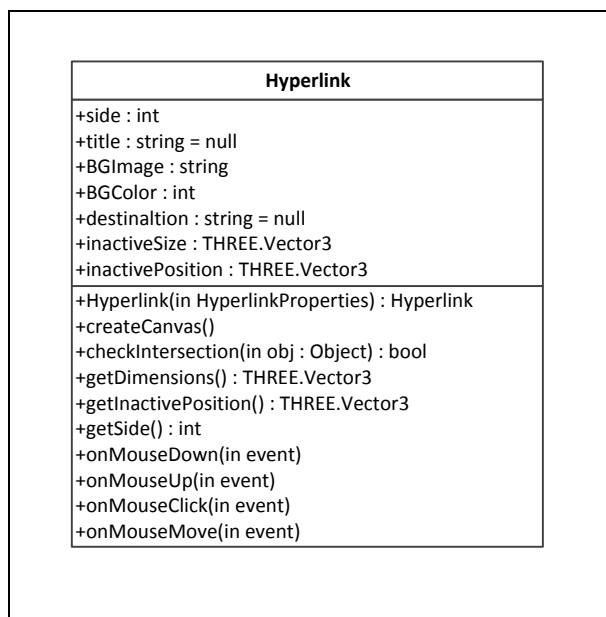


Figure 4.21: Hyperlink class

And the code line needed to add a Hyperlink component to the WebCube likes:

```

wc.addHyperlink(
0, // side number
"./images/button.png", // optional background image
null, // optional background color
"Goto Google", // hyperlink title
0xFF00FF, // text color
"Arial", // text font
10, // text size
true, // underlined?
true, // bold?
true, // italic?
0.5, // opacity
new THREE.Vector3(0.2,0.1,1), // marquee size
new THREE.Vector3(0.6,0.35,1), // marquee position
1, // movable?
0, // rotation angle
"http://www.google.com"); // optional destination
  
```

4.1.12. 3D Object component

Any object that can be manipulated in 3D, a simple 3D objects like cube, sphere , cylinder, torus and many more other can be created directly using Three.js library, but other complex object can be created using a 3D modeling software and exported as an object file (.obj) then converted to JavaScript file (.js) using Three.js exporter utility. The JavaScript file can be loaded to the WebCube easily. Figure 4.22 (a) shows an earth globe inside a cube and (b) shows Walt Disney head object loaded in the WebCube after converting it to JavaScript file.

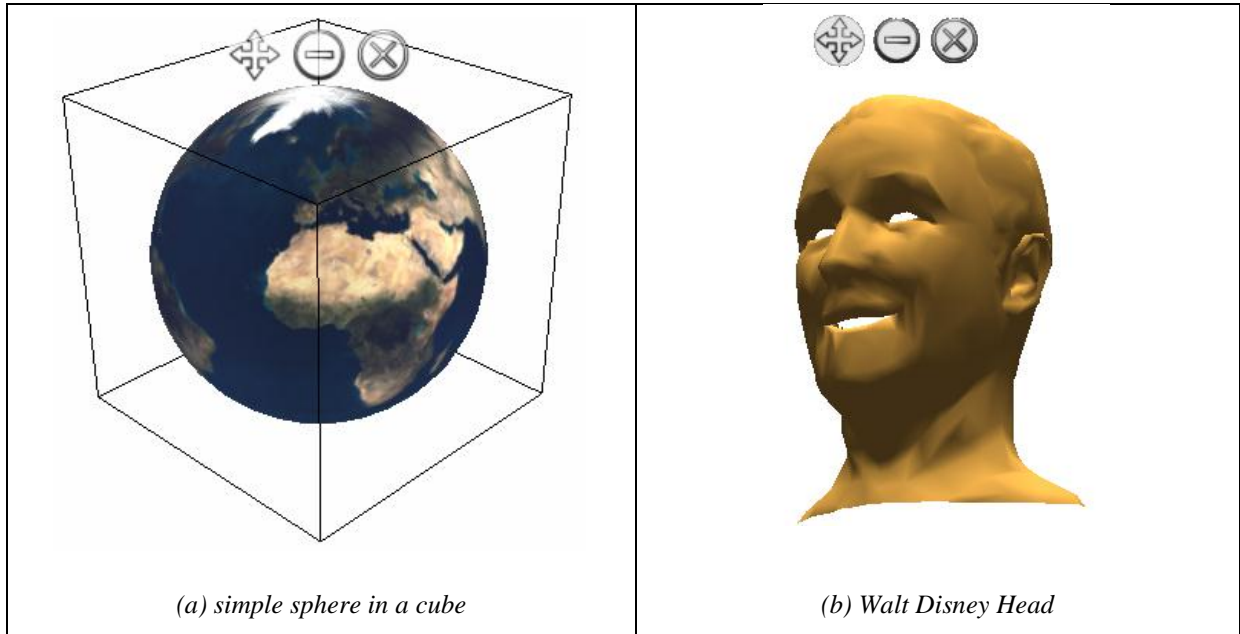


Figure 4.22: 3D object added to the WebCube

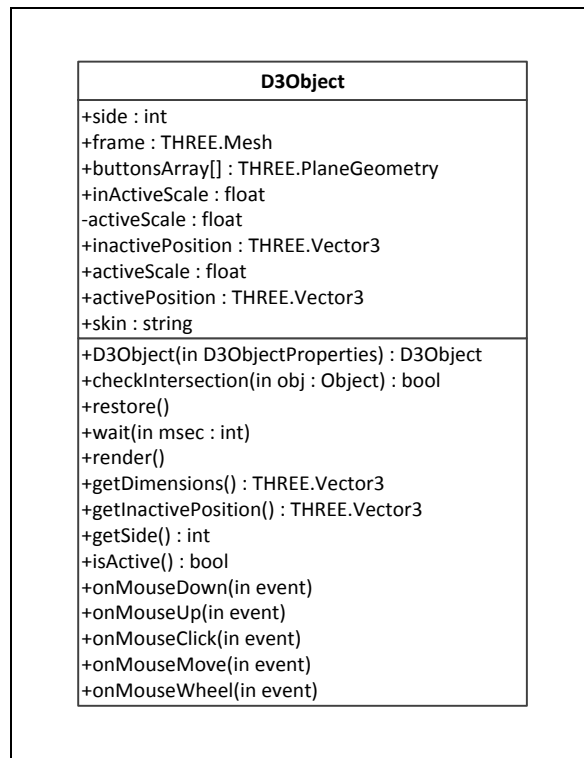


Figure 4.23: 3D object class

The frame attributes of the 3D object class shown in figure 4.23 is of type mesh, so the user must define a geometry and material to create the mesh then pass it to the class, the user may also customize the needed active and inactive scales that will be applied on the mesh. The 3D object functions are the same as the basic class.

And here is the code line needed to add a simple 3D object to the WebCube.

```
var earthGeometry = new THREE.SphereGeometry( screen_w*0.05, 30, 30 );
var texture = THREE.ImageUtils.loadTexture( './images/earth.jpg' );
var earthMaterial = new THREE.MeshLambertMaterial({ map:texture } );
var earthMesh = new THREE.Mesh(earthGeometry, earthMaterial);

wc.add3DObject(
earthMesh,                                // mesh
4,                                        // side
1,                                        // inactive scale
"./skin/",                                // skin
new THREE.Vector3(0.2,0.2,0.01), // inactive position
true,                                    // inactive movable?
1.5,                                    // active scale
new THREE.Vector2(0.4,0.4,0), // active position
7,                                        // buttons array
new THREE.Vector3(90,45,0) ); // inactive rotation
```

To add a complex 3D object to the WebCube you must have the js file of that object and then write:

```
loader = new THREE.JSONLoader();
texture = THREE.ImageUtils.loadTexture( 'textures/metal.jpg' );
loader.load( 'WaltHeadLo.js', function ( geometry ) {
    geometry.computeVertexNormals();
    material = new THREE.MeshBasicMaterial( { map:texture} );
    mesh = new THREE.Mesh( geometry, material);
} );
wc.add3DObject(mesh,4,1,"./skin/",new THREE.Vector3(0.2,0.2,0.01),true,
1.5,new THREE.Vector2(0.4,0.4,0),7,new THREE.Vector3(90,45,0) );
```

Chapter 5

Library Evaluation

This chapter will discuss the methods used for software testing and testing types, and it will introduce for software metrics. After that it will discuss three websites: a personal website, a commercial website and an educational website as examples of using the library, and use them to evaluate the library.

5.1. Software Testing

Software testing: “is the process used to help identify the correctness and completeness of developed computer software”. So the aim of testing is to demonstrate that a program works by showing that it has no errors. Because of that, testing can be defined as: “testing is the process of executing a program with the intent of finding errors” [23].

5.1.1. Test Levels

Tests are grouped by where they are added in the software development process, or by the level of specificity of the test. The main levels during the development process are unit, integration, system and acceptance testing.

5.1.1.1. Unit testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct which allow the programmer to find the problems early.

5.1.1.2. Integration testing

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates [26]. It seeks to verify the interfaces between components which may be integrated in an iterative way to allow interface issues to be localized more quickly and fixed.

5.1.1.3. System Testing

System testing tests an integrated system to verify that it meets its requirements. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and it seeks to detect defects within the system as a whole.

5.1.1.4. Acceptance testing

Acceptance testing allows the end-user or customer to decide whether or not to accept the product and if the product meets the requirements or not.

In the case of this library, there is an interlacing in the unit, integration and system testing since the only unit that can run and tested alone is the WebCube class and every other units can produce an output if they added to the WebCube, so unit test is done through the integration test, and since the system could be a WebCube and one other component so it interlaced somehow with integration test. For acceptance test, it done in the WebCube prototype evaluation mentioned in section 1.3 which result in a high acceptance percentage.

5.1.2. Types of Testing

Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

5.1.2.1. Black-box Testing

Black-box testing examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It checks for all expected inputs and tests the item based on its interfaces and functional requirements by varying input parameters and compare the result with independently calculated results, but since the result in this library is a scene in the browser canvas so the result will be compared with the expected scene by the tester's eye. About twenty instances are created of every component in the library during the black-box testing to ensure that the component behave as expected in any environment, figure 5.1 shows the code written to do a black-box test of the ImageObject component, this code will create twenty instances of ImageObject with different input parameters as a try to test all possible inputs and check if the component behave as expected in each case.


```

<script>
function v3(vx,vy,vz){return new THREE.Vector3(vx,vy,vz);}
function v2(vx,vy){return new THREE.Vector2(vx,vy);}
function init() {
  wc = new WebCube(0,"t.jpg","t.jpg","t.jpg","t.jpg","t.jpg",1);
  wc.addImage ( "test/1.jpg","title v1","./skin/", 0,v3(0.2,0.2,1)
    ,v2(0.1,0.1),false,v3(0.4,0.5,1),v3(0.1,0.1,0),7,0,"comment");
  wc.addImage ( "test/2.jpg","title v2","./skin/", 0,v3(0.2,0.3,1)
    ,v2(0.5,0.5),true,v3(0.4,0.4,1),v3(0.25,0.1,0),5,-10,"");
  wc.addImage ( " test/3.jpg","title v3","./skin/", 0,v3(0.3,0.4,1)
    ,v2(0.1,0.6),false,v3(0.3,0.5,1),v3(0.1,0.15,0),1,0,"");
  wc.addImage ( "test/4.jpg","title v4","./skin/", 0,v3(0.25,0.2,1)
    ,v2(0.6,0.4),true,v3(0.5,0.3,1),v3(0.14,0.1,0),5,180,"");
  wc.addImage ( "test/5.jpg","title v5","./skin/", 1,v3(0.2,0.2,1)
    ,v2(0.1,0.1),false,v3(0.4,0.5,1),v3(0.1,0.1,0),7,0,"comment");
  wc.addImage ( "test/6.jpg","title v6","./skin/", 1,v3(0.2,0.3,1)
    ,v2(0.5,0.5),true,v3(0.4,0.5,1),v3(0.1,0.4,0),5,-10,"");
  wc.addImage ( "test/7.jpg","","./skin/", 1,v3(0.3,0.4,1)
    ,v2(0.1,0.6),false,v3(0.4,0.5,1),v3(0.2,0.2,0),1,0," comment ");
  wc.addImage ( "test/8.jpg","title v8","./skin/", 1,v3(0.25,0.2,1)
    ,v2(0.6,0.4),true,v3(0.4,0.5,1),v3(0.3,0.1,0),5,180,"");
  wc.addImage ( "test/9.jpg","","./skin/", 2,v3(0.2,0.2,1)
    ,v2(0.1,0.1),false,v3(0.4,0.5,1),v3(0.3,0.5,0),7,0,"");
  wc.addImage ( "test/10.jpg","title v10","./skin/", 2,v3(0.2,0.3,1)
    ,v2(0.5,0.5),true,v3(0.2,0.5,1),v3(0.3,0.3,0),5,-10," comment ");
  wc.addImage ( "test/11.jpg","title v11","./skin/", 2,v3(0.3,0.4,1)
    ,v2(0.5,0.6),false,v3(0.4,0.5,1),v3(0.1,0.2,0),1,0,"");
  wc.addImage ( "test/12.jpg","title v12","./skin/", 2,v3(0.25,0.2,1)
    ,v2(0.6,0.4),true,v3(0.4,0.5,1),v3(0.2,0.1,0),5,180,"");
  wc.addImage ( "test/13.jpg","title v13","./skin/", 3,v3(0.2,0.2,1)
    ,v2(0.1,0.1),false,v3(0.4,0.6,1),v3(0.1,0.5,0),7,0,"");
  wc.addImage ( "test/14.jpg","","./skin/", 3,v3(0.2,0.3,1)
    ,v2(0.5,0.1),true,v3(0.4,0.5,1),v3(0.3,0.2,0),5,-10,"");
  wc.addImage ( "test/15.jpg","title v15","./skin/", 3,v3(0.3,0.4,1)
    ,v2(0.5,0.6),false,v3(0.2,0.5,1),v3(0.2,0.1,0),1,0,"");
  wc.addImage ( "test/16.jpg","title v16","./skin/", 3,v3(0.25,0.2,1)
    ,v2(0.6,0.4),true,v3(0.3,0.5,1),v3(0.1,0.4,0),5,180," comment ");
  wc.addImage ( "test/17.jpg","title v17","./skin/", 4,v3(0.2,0.2,1)
    ,v2(0.1,0.1),false,v3(0.14,0.15,1),v3(0.2,0.1,0),7,0,"");
  wc.addImage ( "test/18.jpg","title v18","./skin/", 4,v3(0.2,0.3,1)
    ,v2(0.5,0.5),true,v3(0.23,0.25,1),v3(0.1,0.3,0),5,-10,"");
  wc.addImage ( "test/19.jpg","title v19","./skin/", 4,v3(0.3,0.4,1)
    ,v2(0.5,0.6),false,v3(0.45,0.5,1),v3(0.1,0.1,0),1,0,"");
  wc.addImage ( "test/20.jpg","","./skin/", 4,v3(0.25,0.2,1)
    ,v2(0.6,0.4),true,v3(0.4,0.5,1),v3(0.18,0.18,0),5,180,"");
  animate();
}
function animate() {
  requestAnimationFrame( animate );
  wc.animate();
}
</script>

```

Figure 5.1: Black-Box Test of ImageObjects.

The resulting WebCube contains twenty ImageObjects as shown in figure 5.2.



Figure 5.2: Result of Black-Box test of ImageObject

5.1.2.2. White-box Testing

White-box testing examines internal structures of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system is used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system level test. It ensures that all combinations of paths through the code are executed.

White-box testing is done for every component in the library by adding a number of components with a different parameters to ensure that every path in the constructor are executed at least once and apply the needed mouse or keyboard events to enforce the component to invoke every method and execute every path in all methods.

5.2. Software Metrics

A comprehensive list of metrics is available for software engineering use, ranging from high-level effort and software size measures to detailed requirements measures. A typical set of metrics include: Quality, Size, effort and Complexity which will discussed here.

5.2.1. Quality

Quality can be defined as the degree of excellence that is measurable in the product. The IEEE definition for software quality is the degree to which a system, component, or process meets the specified requirements and user needs.

Section 5.3 shows three examples of websites built using the library; these examples provide the solution of the problem defined in section 1.4 and meet the objectives listed in section 1.5.

5.2.2. Size (LOC)

Lines of code (LOC) is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code. LOC is typically used to predict the amount of time and effort that will be required to develop a program¹.

As mentioned in section 1.1, the major benefits of a software library is to decrease the lines of code of a software and allow the programmer to think at higher level. The examples shown in section 5.3 explain the percentage of LOC for a websites with the same functionality written with and without this library, this percentage is less than 0.05% in some cases, which mean 99.5% reduction in programmer time, effort and software bugs.

5.2.3. Complexity

Complexity measures focus on designs and actual code. They assume there is a direct correlation between design complexity and design errors, and code complexity and latent defects. Complexity correlates to software size, interfaces among modules, and structure (the number of paths within a module).

The code written using the library almost has no branches at all, it is only statements to add components, and the library classes has a maximum of three levels of branches in any path, and the interfaces found only between the component and the WebCube class.

5.3. Experimental Study

Trying to evaluate this library, three website of different categories was designed as examples and will be used to evaluate the library. The first website will be an example of personal website which is an enhanced copy of Rasheed website. The second one is an auto-show website as an example of a commercial website; this website will take the advantages of 3D graphics and modeling in the WebGL. The third site will be an attractive educational and entertainment website showing the ease of designing with WebCube library.

¹ http://en.wikipedia.org/wiki/Source_lines_of_code , Retrieved 6 Feb. 2012

5.3.1. Personal Website

All examples will not focus on the HTML code of the flat webpages component in the site because it is not the point, it will explain the statements written to generate the 3D look of the websites.

The first website is a copy of Rasheed prototype is shown in figure 5.3 which built using about five thousand lines of WebGL code, this prototype has incomplete events and object properties, this site constructed of fourteen components beside the WebCube itself; these components are four html frames, four images, a video, calendar, label, image gallery and two 3D object models.

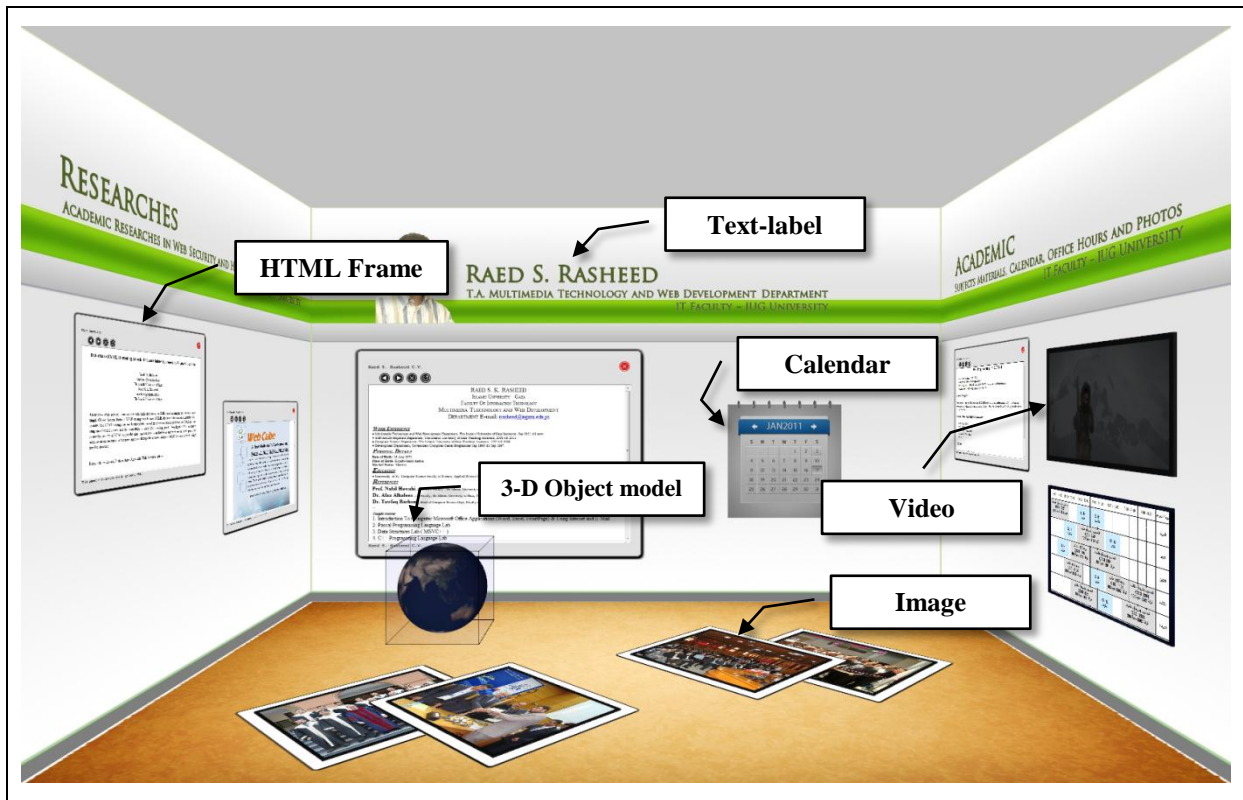


Figure 5.3: Rasheed website

Most objects in the WebCube require only a single line with the required configuration (clarified in chapter 4), so to build this website you need the code shown in figure 5.4 which constructed of about 25 LOC.

The first line is the WebCube container line, WebCube has some added features upon the prototype like: automatic screen resolution detection and fitting, ability to use colored background as another option and the ability to control the depth of the cube.

Lines 2 to 5 add four HTML frames with different properties including source frame, title, skin, side, size, position and rotation.

```

<script>
function v3(vx,vy,vz){return new THREE.Vector3(vx,vy,vz);}
function v2(vx,vy){return new THREE.Vector2(vx,vy);}
function init() {
1: wc = new WebCube(0,"images/front.jpg","images/left.jpg",
   "images/right.jpg","images/ceil.jpg","images/floor.jpg",0.95);

2: wc.addHTMLFrame("RaedCV","rrasheedCV.html", "./images/rrasheedCV.jpg",
   "Raed S. Rasheed C.V.", "./skin/",0,v3(0.4,0.5,1),v2(0.1,0.4),true,
   v3(0.4,0.5,1),v3(0.3,0.25,0),"",0);

3: wc.addHTMLFrame("research","rrasheedSOP.html",
   "./images/rrasheedSOP.jpg", "SOAP
Security","./skin/",1,v3(0.4,0.27,1),v2(0.05,0.35),true,
   v3(0.4,0.5,1),v3(0.3,0.25,0),"",0);

4:
wc.addHTMLFrame("webcube","rrasheedWCHCI.html","./images/rrasheedW.jpg",
   "3D WebCube Prototype","./skin/",1,v3(0.4,0.3,1),v2(0.5,0.5),true,
   v3(0.6,0.6,1),v3(0.3,0.25,0),"",0);

5: wc.addHTMLFrame("acadimic","rrasheedWPC.html",
   "./images/rrasheedWPC.jpg", "Web Programming
Course","./skin/",2,v3(0.5,0.3,1),v2(0.1,0.35),true,
   v3(0.4,0.5,1),v3(0.3,0.25,0),"",0);

6: wc.addCalendar ( "./skin/", 0, v3(0.2,0.3,1) ,v2(0.65,0.5),true,
   v3(0.2,0.3,1),v3(0.5,0.4,0),3,0);

7: wc.addVideo ( "video1", "./sintel.mp4","sintel.mp4","./skin/", 2,
   v3(0.35,0.28,1) ,v2(0.62,0.35),true,v3(0.4,0.5,1),v3(0.1,0.1,0),7,0,
   "Some Comments:\n File Size:14.2 MB\n File type:MP4\n bla bla bla ",
   false,false);

8: wc.addImage ( "./images/rrasheedOH.jpg" ,"" ,"./skin/", 2,v3(0.4,0.2,1) ,
   v2(0.55,0.67),true,v3(0.35,0.35,1),v3(0.2,0.4,0),3,0, " ");

9: wc.addImage ( "./images/1.jpg" ,"" ,"./skin/", 4,v3(0.25,0.4,1) ,
   v2(0.5,0.3),true, v3(0.3,0.35,1),v3(0.3,0.4,0),7,-45,
   "File Name: image4 \n\n File Size: 484 KB \n\n File Type: jpg");

10:wc.addImage ( "./images/2.jpg" ,"" ,"./skin/", 4, v3(0.2,0.3,1) ,
   v2(0.1,0.6),true,v3(0.3,0.35,1),v3(0.2,0.4,0),1,45, " ");

11:wc.addImage ( "./images/3.jpg" ,"" ,"./skin/", 4,v3(0.22,0.3,1) ,
   v2(0.25,0.6),true,v3(0.3,0.35,1),v3(0.5,0.4,0),1,-45, " ");

12:wc.addGallery ( "./gallery.xml" ,"قسم تكنولوجيا المعلومات", "./skin/",
4,
   v3(0.25,0.4,1),v2(0.6,0.3),true,v3(0.2,0.3,1),v3(0.3,0.3,0),-
45,4,3);

13:wc.addHyperlink(0,null,null, "Raed S. Rasheed",0x55632E,"times", 0.8,
false,true,false,1,v3(0.6,0.2,1),v3(0.17,0.07,1),1,0, "");

```

Figure 5.4 (page 1 of 2) : The code needed to build Rasheed website


```

14:wc.addMarquee(0,""," ", "./marquee.xml",0x002000,0xFFFF, "Arial",
    1.2,false,true,false,0.5,v3(0.3,0.06,1),v3(0.3,0.235,1),false,0,4);

15:var earth =new THREE.Mesh(new THREE.SphereGeometry(screen_w*0.05,30,30),
    new THREE.MeshLambertMaterial( {map:THREE.ImageUtils.loadTexture(
    './images/earth.jpg' )} ) );

16:var object1=wc.add3DObject(earth, 4,1 ,"./skin/",v3(0.2,0.2,0.01),
    true,1.5,v2(0.4,0.4,0),7,v3(90,45,0) );
17:var earthCube = new THREE.Mesh( new
    THREE.CubeGeometry(screen_w*0.1,screen_w*0.1,screen_w*0.1 ),
    [new THREE.MeshBasicMaterial({color: 0x0000FF, opacity: 0,
    transparent: true}),
    new THREE.MeshBasicMaterial({color:0x000,wireframe: true})]);
18:object1.addMesh(earthCube,1);
19:object1.show();
20:animate();
}
function animate() {
    requestAnimationFrame( animate );
    wc.animate();
}
</script>

```

Figure 5.4 (page 2 of 2) : The code needed to build Rasheed website

Line 6 adds a calendar which automatically draws the current month and selects the current day, where line 7 adds a video component to the WebCube, video component like all other component require the same properties.

Lines 8 to 11 add four images on different sides with different properties, image and video component have a unique property which is the ability to flip and showing the component information added by the designer.

Line 12 adds a gallery component which requires the images list in an XML file and a title with the common properties.

Line 13 adds a hyperlink component with no destination, which in this case used as a label, the designer could use this object as an static image which will not be activated.

Line 14 adds a marquee component which also requires an XML file having the marquee elements and their destinations, this component is not found in the prototype but it is added to enrich the site.

Line 15 build a mesh which is sphere geometry covered with earth image as a material, this mesh will represent the globe, and line 16 passes this mesh to the 3D object component specifying its properties. Line 17 builds another mesh represent the cube and add it to the object in line 18, line 19 just shows this object.

Line 20 calls the animate function which calls the requestAnimationFrame function that renders the scene and repeatedly calls itself.

The resulting website is shown in figure 5.5 and screen shots are shown in figure 5.6.



Figure 5.5: Rasheed website built using the library

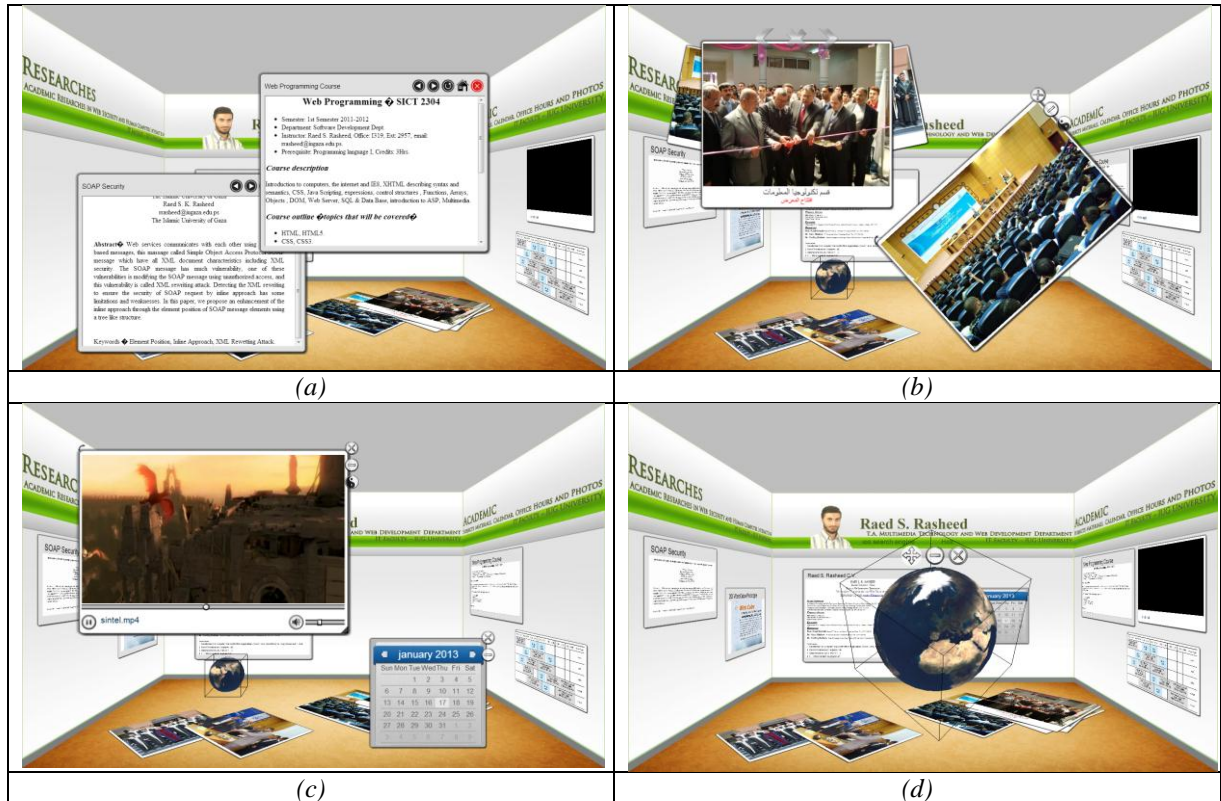


Figure 5.6: screenshots of the personal website showing: (a) two active HTML frames, (b) rotated image and image gallery, (c) running video and calendar (d) manipulating 3D object.

Comparing the source line of code used to build the site using the library with the original code show that using the library reduces LOC to 25 lines from more than 5000 lines, this save about 99.5% from developer time and effort. Another important issue is that to build a 3D website you must be an expert in the 3D graphics and WebGL coding but using this library you need only the programming basics and even the basics may be not required for simple website, it is only copy and paste and then change the configuration.

5.3.2. Commercial website

This example simulate a commercial website represent in this case an auto show, this web site will show an exact 3D representation of different models of cars, the user will choose a car then a rotated 3D model of that car will appear, and he can view the full specification of that model. This web site is an example of many companies wants to show an exact 3D model of their products to the customers.

There was a try to build the auto show website using three.js library, but it was too complex so by separating the website, it requires about 5400 line of code for each car page. The Lamborghini car page is shown in figure 5.7.

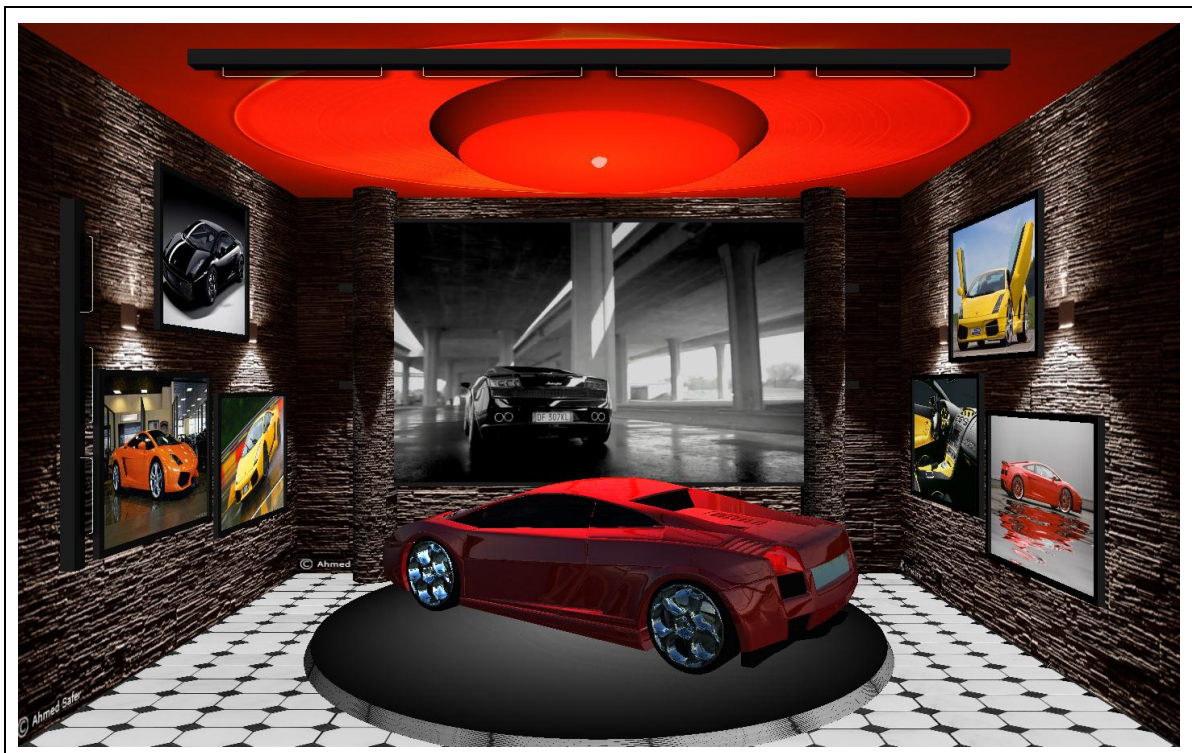


Figure 5.7: Auto show website - Lamborghini webpage .

A similar site was built using the library contains four different models of cars and each car has many different colors, for each car type there is a specification HTML frame ,contact frame video file and a gallery so there is eight frames, four galleries and four video files displayed on the same video component.

Figure 5.8 show a part of code representing the 3D models definitions and properties and color definitions, most of this par built by the 3D designer:

```

D1:var CARS = {
  "veyron":{
    name: "Bugatti Veyron",
    url: "obj/veyron/VeyronNoUv_bin.js",
    mesh: null,
    object: null,
    materials: null,
    video:"./videos/veyron.mp4",
    specs:null,
    contact:null
  },
  "gallardo": {
    name: "Lamborghini Gallardo",
    url: "obj/gallardo/GallardoNoUv_bin.js",
    .....
    .....
    .....
  }
};
D2:var mlib = {
  "Orange":new
THREE.MeshLambertMaterial({color:0xff6600,reflectivity:0.3}),
  "Blue": new
THREE.MeshLambertMaterial({color:0x001133,reflectivity:0.3}),
  .....
  .....
}
D3:CARS[ "veyron" ].materials = {
  [mlib[ "Orange metal" ] , "images/orange_metal.png"],
  [mlib[ "Blue metal" ] , "images/blue_metal.png"],
  [mlib[ "Green metal" ] , "images/green_metal.png"],
  .....
  .....
}

```

Figure 5.8: 3D models properties and color definitions

The script shown in figure 5.9 require definitions for the 3D models and the colors used as materials, the CARS variable at D1 segment of the script is an array represent the four cars and their properties like model name, Object URL, reference to the object mesh initiated with null until the object loaded in the createScene function, object variable refer to the WebCube object that represent the car, materials represent the allowed colors used for this car and initiated in segment D3, video represent the video URL that will displayed along with this car, specs is a reference to the specifications HTML frame and finally contact points to contact HTML frame. Where segment D2 defines the colors used in these cars as materials.

The createScene() function in the script called by the binary loader when completely loads the car model, this function create a mesh from the geometry and adds it to the WebCube. The switchCar() function called by the hyperlinks when the user change the car model, this function hide the current car and show the selected one, and at the same time it switches the video to the new car video and calls addColor() function which will remove the previous colors list and add a new list from the car colors specifications defined in D3 segment as mentioned above.

```

function v3(vx,vy,vz){return new THREE.Vector3(vx,vy,vz);}
function v2(vx,vy){return new THREE.Vector2(vx,vy);}
f1: function switchCar( car ) {
    selectedCar.object.hide();
    selectedCar = CARS[car];
    selectedCar.object.show();
    carVideo.changeSource(selectedCar.video);
    addColors();
}
f2: function createScene( geometry, car ) {
    CARS[ car ].mesh=new THREE.Mesh(geometry,new
    THREE.MeshFaceMaterial());
    CARS[ car ].object = wc.add3DObject(CARS[ car ].mesh, 5,1 ,
    "./skin/",v3(0.38,0.63,0.3),false,1,v2(0.4,0.5,0),7,v3(0,90,0),false
    );
}
var colorArray = new Array();
f3: function addColors()
{
    for (var j=0; j<colorArray.length; j++)
        colorArray[j].remove();

    for (var i=0; i<selectedCar.materials.length; i++)
    {
        colorArray[i] = wc.addSlidingHyperlink(3,
        selectedCar.materials[i][1], null, "",0x55632E,"times",
        0.8,false,true,false,1,
        v3(0.05,0.08,1),v3(0.06*i,0,1),1,0,"",
        "selectedCar.mesh.geometry.material=selectedCar.materials.[\"+i+\"][0]" );
    }
}
var decor2, carVideo;
function init() {
00:wc = new WebCube(0,"w.jpg","w.jpg","w.jpg","c.jpg","f.jpg",0.94);
01:var Pillar1 = new THREE.Mesh(
    new THREE.CylinderGeometry(screen_h*0.06,screen_h*0.06,screen_h,20),
    new THREE.MeshLambertMaterial({map:THREE.loadTexture('w.jpg')}));
02:var Pillar2 = new THREE.Mesh(
    new THREE.CylinderGeometry(screen_h*0.06,screen_h*0.06, screen_h,20),
    new THREE.MeshLambertMaterial({map: THREE.loadTexture('w.jpg')}));
03:wc.add3DObject(Pillar1,0,1,"./skin/",v3(0.15,0,0.05),false,1,
    v2(0,0,0),7,v3(0,180,0),false).show();
04:wc.add3DObject(Pillar2,0,1,"./skin/",v3(0.85,0,0.05),false,1,
    v2(0,0,0),7,v3(0,180,0),false).show();
05:var decor1 = new THREE.Mesh(
    new THREE.CylinderGeometry(screen_h*0.35,screen_h*0.31, screen_h/25,40),
    new THREE.MeshLambertMaterial( { color: 0x800000} ) );
06:wc.add3DObject(decor1, 0,1,"./skin/",v3(0.5,0,0.1),false,1,
    v2(0,0,0),7,v3(0,0,0),false).show();

```

Figure 5.9 (page 1 of 3): Auto show website code.

```

07:decor2 = new THREE.Mesh(
    new THREE.CylinderGeometry(screen_h*0.48,screen_h*0.49, screen_h/50,50),
    new THREE.MeshLambertMaterial({map:THREE.loadTexture('f.jpg')}));

08:wc.add3DObject(decor2, 0,1 , "./skin/",v3(0.5,0.9,0.2),false,1,
    v2(0,0,0),7,v3(0,0,0),false ).show();

09:loader.load({model: CARS[ "gallardo" ].url,
    callback:function(geometry){createScene(geometry,"gallardo")}});

10:loader.load({model: CARS[ "veyron" ].url,
    callback:function(geometry){createScene(geometry,"veyron")}});

11:loader.load({model: CARS[ "f50" ].url,
    callback:function(geometry){createScene(geometry,"f50")}});

12:loader.load( { model: CARS[ "camaro" ].url,
    callback:function(geometry){createScene(geometry,"camaro")}});

13:selectedCar = CARS['gallardo'];
14:addColors();

15:wc.addSlidingHyperlink(1,"
lamborghini.png",null,"",null,"",0,false,true,
    false,1,v3(0.25,0.08,1),v3(0,0.25,1),1,0,"", "switchCar('gallardo');");

16:wc.addSlidingHyperlink(1," bugatti1.png",null,"",null,"",0,false,true,
    false,1,v3(0.25,0.08,1),v3(0,0.35,1),1,0,"", "switchCar('veyron');");

17:wc.addSlidingHyperlink(1," ferrari.png",null,"",null,"",0,false,true,
    false,1,v3(0.25,0.08,1),v3(0,0.45,1),1,0,"", "switchCar('f50');");

18:wc.addSlidingHyperlink(1,"chevrolet.png",null,"",null,"", 0,false,true,
    false,1,v3(0.25,0.08,1),v3(0,0.55,1),1,0,"", "switchCar('camaro');");

19:CARS["camaro"].contact = wc.addHTMLFrame("cc","camarocontact.htm",
    "cc.jpg","Chevrolet Contact","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

20:CARS["f50"].contact = wc.addHTMLFrame("fc","f50contact.htm",
    "fc.jpg","Ferrari Contact","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

21:CARS["gallardo"].contact = wc.addHTMLFrame("gc","gallardocontact.htm",
    "lc.jpg","Lamborghini
Contact","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

22:CARS["veyron"].contact = wc.addHTMLFrame("vc","veyroncontact.htm",
    "bc.jpg","Bugatti Contact","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

23:CARS["camaro"].specs = wc.addHTMLFrame("cs","camarospecs.html",
    "cs.jpg","Chevrolet Specs","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

```

Figure 5.9 (page 2 of 3): Auto show website code.

```

24:CARS["f50"].specs = wc.addHTMLFrame("fs","f50specs.html",
    "fs.jpg","Ferrari Specs","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

25:CARS["gallardo"].specs = wc.addHTMLFrame("gs","gallardospecs.html",
    "ls.jpg","Lamborghini Specs","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

26:CARS["veyron"].specs = wc.addHTMLFrame("veyronspecs","veyronspecs.html",
    "bs.jpg","Bugatti Specs","./skin/",0,v3(0.5,0.6,1),v2(0.2,0.1),true,
    v3(0.6,0.7,1),v3(0.2,0.2,0),"",0);

27:carVideo=wc.addVideo("video","gallardo.mp4","","./skin/", 0,v3(0.7,0.8,1),
    v2(0.15,0),false,v3(0.4,0.5,1),v3(0.1,0.1,0),7,0,"",true,true,false);

28:wc.addHyperlink(1,"specs.png",null,"",null,"",0,false,true,false,1,
    v3(0.4,0.4,1),v3(0.1,0.5,1),1,0,"","selectedCar.specs.onMouseClicked()");

29:wc.addHyperlink(1,"contact.png",null,"",null,"",0,false,true,false,1,
    v3(0.4,0.4,1),v3(0.5,0.1,1),1,0,"","selectedCar.contact.onMouseClicked()");

30:wc.addGallery ( "./lg.xml" ,"Lamborghini Gallardo","./skin/", 2,
    v3(0.35,0.4,1),v2(0.05,0.05),false,v3(0.4,0.5,1),v3(0.3,0.05,0),-15,18,7);

31:wc.addGallery ( "./fg.xml" ,"Ferrari F50","./skin/", 2,
    v3(0.35,0.4,1) ,v2(0.6,0.05),false,v3(0.4,0.5,1),v3(0.3,0.05,0),15,17,7);

32:wc.addGallery ( "./cg.xml" ,"Chevrolet Camaro","./skin/", 2,
    v3(0.35,0.4,1) ,v2(0.2,0.55),false,v3(0.4,0.5,1),v3(0.3,0.05,0),15,13,7);

33:wc.addGallery ( "./bg.xml" ,"Bugatti Veyron","./skin/", 2,
    v3(0.35,0.4,1),v2(0.65,0.55),false,v3(0.4,0.5,1),v3(0.3,0.05,0),-15,20,7);

    animate();
}
function animate() {
    requestAnimationFrame( animate );
    wc.animate();
    selectedCar.mesh.rotation.y+=0.01;
    decor2.rotation.y+=0.01;
}

```

Figure 5.9 (page 3 of 3): Auto show website code.

Moving to the `init()` function, line 0 creates the WebCube, and trying to add some decoration, two pillars were added as cylinders covered with wall material, another 3D object is the shape under the ceiling and the other one on the floor under the cars which will refer to it as `decor2`, after creating the WebCube scene, loading the cars objects using the binary loader done in lines 9 to 12, line 13 chooses the Gallardo car to be the selected one and adds its colors menu in line 14.

Lines 15 to 18 adds a `slidingHyperlink` component which the same as normal hyperlink except it will slide-in on mouse move and slide-out on mouse out, these links will call the `swichCar()` function on mouse click.

Lines 19 to 22 adds the contact HTML frames and link these frames to the car contact reference in segment D1, and lines 23 to 26 will add the specifications frames and link them to specs reference where line 27 adds a video component which will hide the eight HTML frames behind.

Line 28 adds a hyperlink component that will call the contact frame of the selected car by triggering the onMouseClick event in that component and line 29 is the same but dealing with specification frame.

Lines 30 to 33 add four galleries to the WebCube, one for each car.

The animate() function which called all the time rotate the selected car and the stage in y direction by a little value, this will result an animated car showing it in 360o view.

The resulting website in figure 5.10 showing the Lamborghini Gallardo car in chrome color, the SlidingHyperlink menu shown in the left with Lamborghini hyperlink slides-in, and there are two normal hyperlink for contact and specifications of the selected car, the right wall have the four galleries, and in the ceiling the allowed colors of that car as slidingHyperlink also.

Figure 5.11 shows four screen shots of the website appearing different car models and colors.



Figure 5.10: Auto show website

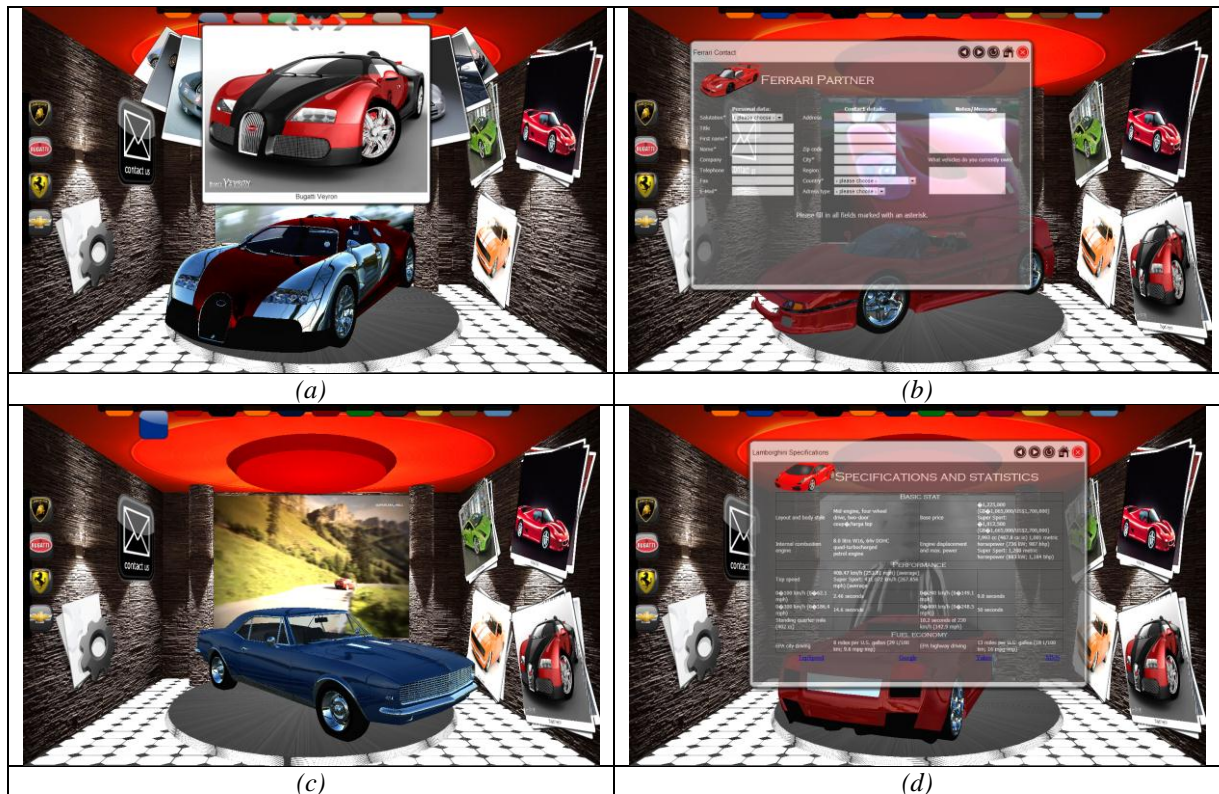


Figure 5.11: screenshots of auto show website showing: (a) Bugatti gallery, (b) Ferrari partner contact frame, (c) changing Chevrolet color and (d) Lamborghini specifications.

The auto show website requires more advanced level of programming as seen before, but still the comparison is unfair, since the original incomplete website of a single car require about 5300 lines of code with a lot of hanging and unexpected errors.

5.3.3. Educational / Entertainment Website

This section introduces to an education and entertainment website, this site designed for children in age of about five years old and the contents could be updated upon the targeted child ages.

This site trying to learn the basics like numbers and colors, and has a painting application allowing the child to paint many lovely photos and train the child on controlling his hand motion, the site also contains things names like animals and fruits as an example with a clear photo for every element, and also the voice of some animals activated when the child clicks on the chosen animal, and for the mothers; the site contains some purposeful stories to read it for their children, the website contents in Arabic language but it include some English letters allowing the child to arrange them as he like.



Figure 5.12: Educational website.

Figure 5.12 shows the educational website represented a child's room, it contains in the left side the photos in black and white ready for painting by clicking any one of them, and in the right side the stories, the animals and fruits names shown in the front side, while the animals distributed on the front side and floor can play the animal voice file, the cube on the floor for numbers and the ball for colors, at last on the table the English letters cubes.

WebCube idea allowing all different elements found in the site appears clearly and requires no scrolling or searching and the library make the design of that site requires no time, discarding the time needed for the graphics design.

To design the site you need seven 3D objects represent the colors ball, the numbers cube and five cubes for letters. The table object built using 3 cylinders for legs and on cylinder for the surface and the chairs are the same as the table but in different scale.

All the painting objects and animals for voice is a hyperlinks, animals and fruits books are gallery component, and at last the stories is a book component.

The code shown in figure 5.13 contains about forty lines and few functions, the large number of lines because of its large number of objects not because of complexity as shown later.

The function f1 used by the coloring image hyperlinks to send the selected image to the paint application, while f2 used as macro to build a table or chair depends on the parameters send by the main program, and the same f3 used to build a cube of letters.

Line 2,3,4 create the numbers cube from six different images, where lines 5 and 6 create the colors ball.

Lines 7 to 10 create the table and three small chairs using the buildTable function.

Line 22 adds animals gallery, where line 23 adds fruits gallery. Lines 13 to 16 add four stories to the site using BookObject which is not discussed before, this object is the same as the gallery but displays two pages at the same time instead of one, trying to simulate a real book.

```

<script>
function v3(vx,vy,vz){return new THREE.Vector3(vx,vy,vz);}
function v2(vx,vy){return new THREE.Vector2(vx,vy);}
function texture (img){
    return new THREE.MeshBasicMaterial(
        {map:THREE.ImageUtils.loadTexture (img)});
}
f1:function coloring(image){
    if (!paintApp.is_Active())
    {
        paintApp.changeURL('paintApp/paint.html?img=images/'+image);
        paintApp.onMouseClicked();
    }
}
f2:function buildTable(radius,legRadius,height,position,color1,color2)
{
    var surface = new THREE.Mesh(
        new THREE.CylinderGeometry(radius, radius, thick,30,true ),
        new THREE.MeshLambertMaterial( { color: color1} ) );
    var table=wc.add3DObject(surface,4,1 ,"/skin/",position,false,1,
        v2(0,0,0),7,v3(90,0,0),false );
    var tableLeg1=new THREE.Mesh( new THREE.CylinderGeometry( legRadius,
        legRadius, height,10,true ),
        new THREE.MeshLambertMaterial({color:color2}));
    tableLeg1.position.set(radius/2,-height/2,radius/2);
    var tableLeg2=new THREE.Mesh( new THREE.CylinderGeometry( legRadius,
        legRadius, height,10,true ),
        new THREE.MeshLambertMaterial({color:color2}));
    tableLeg2.position.set(radius/1.5,-height/2,radius/3);
    var tableLeg3=new THREE.Mesh( new THREE.CylinderGeometry( legRadius,
        legRadius, height,10,true ),
        new THREE.MeshLambertMaterial({color:color2}));
    tableLeg3.position.set(0,-height/2,-radius/3);
    table.addMesh(tableLeg1,1);
    table.addMesh(tableLeg2,1);
    table.addMesh(tableLeg3,1);
    table.show();
}
f3:function addLettersCube(basePath, inactivePosition, activePosition)
{
    var cw1 = screen_h*0.075;
    var mat= [ texture(basePath +'1.jpg'), texture (basePath +'2.jpg'),
        texture(basePath +'3.jpg'), texture(basePath +'4.jpg'),
        texture (basePath +'5.jpg'), texture(basePath +'6.jpg')];
    var mesh=new THREE.Mesh(new THREE.CubeGeometry(cw1,cw1,cw1,1,1,1,mat),
        new THREE.MeshFaceMaterial());
}

```

Figure 5.13(page 1 of 3): The code needed to build the educational website.

```

        wc.add3DObject(mesh,4,1,"./skin/", inactivePosition ,false,
                        1.5, activePosition,5,v3(0,0,30),true ).show();
    }
    var paintApp,sound;
    function init() {
01:wc = new WebCube(0,"f.jpg","l.jpg","r.jpg","c.jpg","fl.jpg",1);
    var diceMaterials = [];
02:for ( var i = 1; i <= 6; i ++ )
        diceMaterials[i-1]=texture('d'+i+'.jpg');
        var cw = screen_h*0.15;
03:diceMesh = new THREE.Mesh( new THREE.CubeGeometry(cw,cw,cw,diceMaterials),
                               new THREE.MeshFaceMaterial());
04:var dice=wc.add3DObject(diceMesh,4,1,"./skin/",v3(0.05,0.15,0),false,1.5,
                           v3(0.4,0.4,0),7,v3(0,0,30),true ).show();

05:var ballMesh=new THREE.Mesh(new THREE.SphereGeometry(screen_w*0.05,30,30),
                               new THREE.MeshLambertMaterial({ map:
                                   THREE.ImageUtils.loadTexture ('ball.jpg')}));
06:ball=wc.add3DObject(ballMesh, 4,1 ,"./skin/",v3(0.43,0.7,0),false,1.5,
                       v3(0.55,0.6,0),7,v3(90,90,0),true ).show();

        var r1=screen_h*0.3, r2=screen_h*0.01, h1=screen_h*0.2, r3=screen_h*0.09;
07:buildTable(r1,r2,h1 ,v3(0.6,0.3,-0.15),0xFFCE4D,0x698BE4);
08:buildTable(r3,r2,h1/2,v3(0.55,0.35,-0.03),0x698BE4,0xFFCE4D);
09:buildTable(r3,r2,h1/2,v3(0.65,0.57,-0.03),0x698BE4,0xFFCE4D);
10:buildTable(r3,r2,h1/2,v3(0.77,0.55,-0.03),0x698BE4,0xFFCE4D);

11:wc.addGallery ( "animals.xml" , "كتاب الحيوانات", "./skin/",0,v3(0.3,0.5,1),
                  v2(0.58,0.25),false,v3(0.4,0.6,1),v3(0.3,0.05,0),-15,19,5);
12:wc.addGallery ( "fruit.xml" , "كتاب الفواكه", "./skin/", 0,v3(0.25,0.4,1),
                  v2(0,0.1),false,v3(0.4,0.6,1),v3(0.3,0.05,0),10,19,5);

13:wc.addBook ( "./book1.xml" , "./skin/", 2,v3(0.1,0.2,1) ,v2(0.87,0.55),
                false,v3(0.3,0.7,1),v3(0.2,0.05,0),10,12,0x003300);
14:wc.addBook ( "./book2.xml" , "./skin/", 2,v3(0.1,0.2,1) ,v2(0.6,0.6),
                false,v3(0.3,0.7,1),v3(0.2,0.05,0),-10,12,0x000000);
15:wc.addBook ( "./book3.xml" , "./skin/", 2,v3(0.1,0.2,1) ,v2(0.17,0.75),
                false,v3(0.3,0.7,1),v3(0.2,0.05,0),10,12,0x000000);
16:wc.addBook ( "./book4.xml" , "./skin/", 2,v3(0.1,0.2,1) ,v2(0.2,0.5),
                false,v3(0.3,0.7,1),v3(0.2,0.05,0),10,8,0x000000);

17:paintApp = wc.addHTMLFrame("paint","paint.html","p.jpg","Paint","./skin/",
                              0,v3(0.15,0.35,1),v2(0.05,0.1),true,v3(0.6,0.9,1),v3(0.2,0,0),"",0);

18:wc.addHyperlink(1,"1.png",null, "",null,"",0.8,false,true,false,1,
                  v3(0.2,0.2,1),v3(0,0.5,1),false,0,"","coloring('1_.png');");
19:wc.addHyperlink(1,"2.png",null, "",null,"", 0.8,false,true,false,1,
                  v3(0.2,0.2,1),v3(0,0.8,1),false,0,"","coloring('2_.png');");
20:wc.addHyperlink(1,"3.png",null, "",null,"",0.8,false,true,false,1,
                  v3(0.35,0.4,1),v3(0.37,0.15,1),false,0,"","coloring('3_.png');");
21:wc.addHyperlink(1,"4.png",null, "",null,"",0.8,false,true,false,1,
                  v3(0.3,0.3,1),v3(0.5,0.7,1),false,0,"","coloring('4_.png');");
22:wc.addHyperlink(1,"5.png",null, "",null,"", 0.8,false,true,false,1,
                  v3(0.3,0.3,1),v3(0.05,0.2,1),false,0,"","coloring('5_.png');");
23:wc.addHyperlink(1,"6.png",null, "",null,"", 0.8,false,true,false,1,
                  v3(0.3,0.3,1),v3(0.7,0.45,1),false,0,"","coloring('6_.png');");

```

Figure 5.13(page 2 of 3): The code needed to build the educational website.

```

24:sound = wc.addAudio ( "sound","","", "./skin/",0,v3(0.01,0.01,1)
    ,v2(0.8,0.9), false,v3(0.4,0.5,1),v3(0.1,0.1,0),7,0,"",true,false,false);

25:wc.addHyperlink(4,"cat.png",null, "",null,"", 0.8,false,true,false,1,
    v3(0.2,0.4,1),v3(0,0.6,1), false,0,"",
    "sound.changeSource('sound/cat.mp3');");
26:wc.addHyperlink(4,"dog.png",null, "",null,"", 0.8,false,true,false,1,
    v3(0.2,0.8,1),v3(0.17,0,1), false,0,"",
    "sound.changeSource('sound/dog.mp3');");
27:wc.addHyperlink(4,"sheep.png",null, "",null,"",0.8,false,true,false,1,
    v3(0.2,0.45,1),v3(0.8,0.53,1), false,0,"",
    "sound.changeSource('sound/sheep.mp3');");
28:wc.addHyperlink(4,"frog.png",null, "",null,"", 0.8,false,true,false,1,
    v3(0.1,0.2,1),v3(0.6,0.75,1), false,0,"",
    "sound.changeSource('sound/frog.mp3');");
29:wc.addHyperlink(0,"horse.png",null, "",null,"", 0.8,false,true,false,1,
    v3(0.1,0.2,1),v3(0.5,0.45,1), false,0,"",
    "sound.changeSource('sound/horse.mp3');");
30:wc.addHyperlink(0,"elephant.png",null,
    "",null,"",0.8,false,true,false,1,
    v3(0.2,0.3,1),v3(0.25,0.3,1), false,0,"",
    "sound.changeSource('sound/elephant.mp3');");
31:wc.addHyperlink(0,"lion.png",null, "",null,"", 0.8,false,true,false,1,
    v3(0.15,0.35,1),v3(0.1,0.5,1), false,0,"",
    "sound.changeSource('sound/lion.mp3');");
32:wc.addHyperlink(0,"goat.png",null, "",null,"", 0.8,false,true,false,1,
    v3(0.1,0.15,1),v3(0.5,0.75,1), false,0,"",
    "sound.changeSource('sound/goat.mp3');");
33:wc.addHyperlink(0,"cow.png",null, "",null,"", 0.8,false,true,false,1,
    v3(0.15,0.25,1),v3(0.3,0.6,1), false,0,"",
    "sound.changeSource('sound/cow.mp3');");

34:addLettersCube('./images/cube1/', v3(0.68,0.25,0.17), v3(0.3,0.1,0));
35:addLettersCube('./images/cube2/', v3(0.73,0.25,0.17), v3(0.38,0.1,0));
36:addLettersCube('./images/cube3/', v3(0.78,0.28,0.17), v3(0.46,0.1,0));
37:addLettersCube('./images/cube4/', v3(0.73,0.45,0.17), v3(0.54,0.1,0));
38:addLettersCube('./images/cube5/', v3(0.80,0.45,0.17), v3(0.62,0.1,0));

    animate();
}
function animate() {
    requestAnimationFrame( animate );
    wc.animate();
}
</script>

```

Figure 5.13(page 3 of 3): The code needed to build the educational website.

Line 17 adds the paint application in an HTML frame, this application is a simple JavaScript code used to add colors to a transparent PNG file. Where lines 18 to 23 add the black/white images that lie on the left side, these images is hyperlinks which invoke the coloring function and pass the image URL when the user clicks on it.

Line 24 adds a sound component that will be used to play the animal sound files, where lines 25 to 33 add nine images of animals as hyperlinks to play the sound of that animal when the user clicks on it.

Lines 34 to 38 add five cubes on the table; each cube face has a letter from A to Z.

Screen shots of the educational website shown in figure 5.14 depicting different active objects in the site.

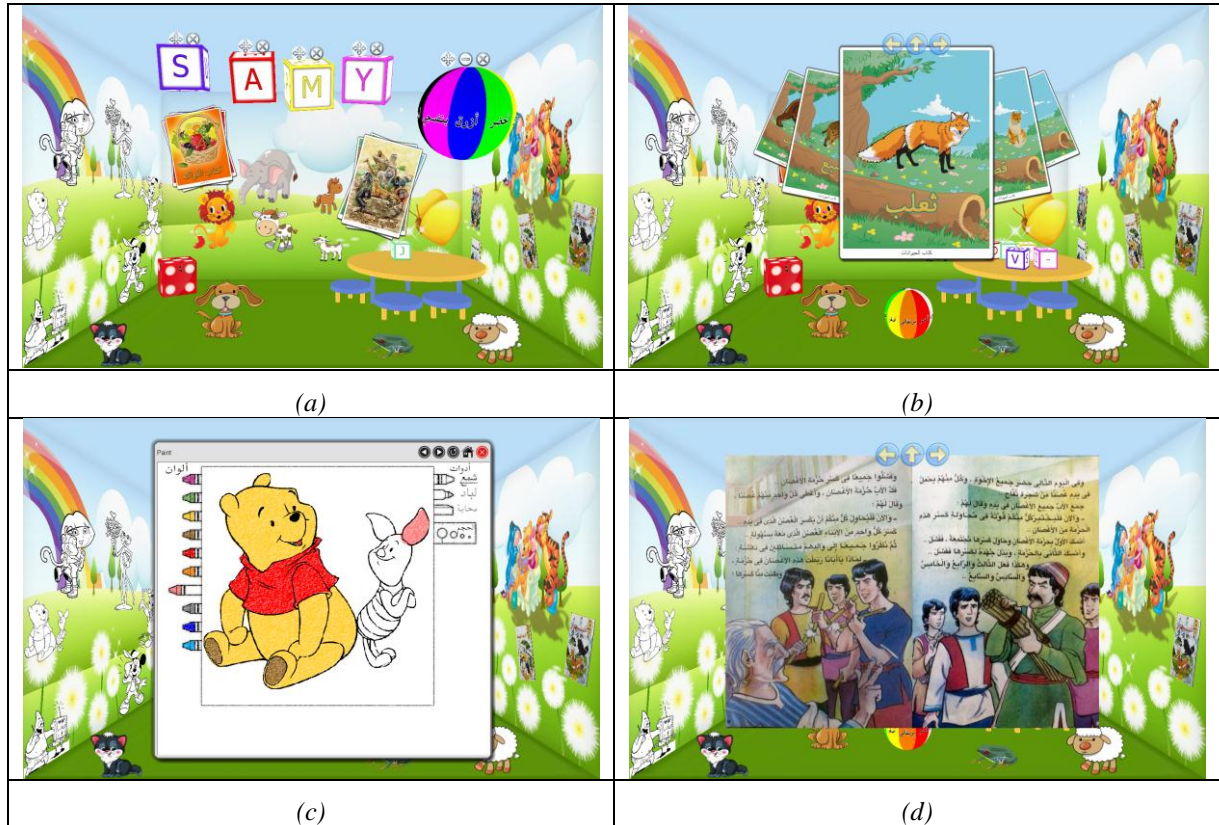


Figure 5.14: screenshots of the educational website showing: (a)manipulating 3D objects (letters and color ball), (b) animals Gallery, (c) paint application and (d) story as BookObject.

This web site built by about fifty simple lines; nearly a single line for each component. This fifty lines execute more than 5400 lines of code from the library without counting the repeated items code, this save at least 99% of web developer time and effort.

Chapter 6

Conclusion and Future Work

6.1. Conclusion

Software libraries are one of the oldest, most used approaches for software reuse. Despite their long past and interesting future, there has been almost no research that relates to libraries as a product domain. In this research, a standard library for WebCube component and events has been introduced. A discussion of the 3D environment in the web and its application aspects has been covered. Then, the WebCube idea and its benefits has been explained. Thereafter, an introduction to 3D graphics basics, WebGL and Threejs has been given. Then, some related works have been explained. Then, a detailed discussion of the WebCube prototype components has been raised. Later, a component analysis and classes design has been explained, and finally showing how to build a website in WebCube style and adding each component to it using the library.

In the evaluation phase, three website -as examples of using the library- have been built, showing how easy and fast it is, and explaining the differences between the code with and without the library.

As a result, using the library will almost save more than 99% of website developer time and effort leading to a dramatically decrease in the code bugs because of using a pretested and robust code, since each component in the library passes all testing levels and types.

6.2. Future Work

WebCube library is a work in progress. In this research, a trial to develop a standard library for WebCube component and events has been given. Yet, the WebCube library can be upgraded. The newly spread of tablet computers and touch interface must be taken into account. Future direction in this research could be summarized as:

- Add touch event listeners to support new tablet computers.
- Develop a web wizard able to build and customize a WebCube website.
- Adding more customization options for each component.
- Update WebCube library to accept newly added objects without the need to update the WebCube class.
- Standardize the WebCube model.
- Extend HTML to include the Web Cube model.

References

- [1] Anthes G., HTML5 Leads a Web Revolution, Communications of the ACM Magazine, Vol. 55 No. 7, 2012
- [2] Anyuru A., Professional WebGL Programming: Developing 3D Graphics for the Web, USA: Wrox, 2012.
- [3] Avison D. and Fitzgerald G., Information Systems Development: Methodologies, Techniques & Tools, 4th Edition, McGraw-Hill, 2006
- [4] Baird S., Sams Teach Yourself Extreme Programming in 24 Hours, USA: Sams Publishing, 2002.
- [5] Beck K. and Cynthia A., Extreme Programming Explained: Embrace Change, 2nd Edition, USA: Addison-Wesley, 2005.
- [6] Booch G., Object-Oriented Analysis and Design with Applications, Addison-Wesley, Menlo Park, CA, 1994
- [7] Cantor D. and Jones B., WebGL Beginner's Guide, United Kingdom: Packt Publishing, 2012.
- [8] Chittaro L. and Ranon R., Dynamic generation of personalized VRML content: a general approach and its application to 3-D e-commerce, Proceedings of the seventh international conference on 3-D Web technology, Tempe, Arizona, USA, 2002.
- [9] Danchilla B., Beginning WebGL for HTML5, USA: Apress , 2012.
- [10] Duckett J., HTML & CSS: Design and Build Websites, USA: John Wiley & Sons, 2011.
- [11] Fawcett J., Quin L. and Ayers D., Beginning XML, 5th edition, USA: John Wiley & Sons, 2012.
- [12] Flanagan D., JavaScript: The Definitive Guide, 6th edition, USA: O'Reilly, 2011.
- [13] Gillenwater Z. M., Stunning CSS3: A project-based guide to the latest in CSS, USA: New Riders, 2011.
- [14] Goldberg K. H., Visual QuickStart Guide: XML, 2nd Edition, USA: Peachpit Press, 2009.
- [15] Guyer S. Z. and Lin C., Optimizing the Use of High Performance Software Libraries, Proceedings of the 13th International Workshop on Languages and Compilers for Parallel Computing, New York, USA, 2000
- [16] Hongyu Z., Google Earth Software Application in Urban Design of China Proceedings of fourth International Conference on Computer Sciences and Convergence Information Technology, Seoul, Kore, 2009a.
- [17] Hunt J. M. and McGregor J. D., A Model for Software Libraries, Proceedings of fifth International Conference on Library-Centric Software Design, San Diego, California, 2005.
- [18] Jones T. C., Programming Productivity. New York: McGraw-Hill, 1986.
- [19] Leffingwell D., Agile Software Requirements, USA: Addison-Wesley, 2010.
- [20] Leffingwell D., Scaling Software Agility: Best Practices for Large Enterprises. USA: Addison-Wesley, 2007.

- [21] Lubbers P., Albers B., Salim F. and Smith R., Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development, USA: Apress, 2010.
- [22] Mills C., Practical CSS3: Develop and Design, USA: Peachpit Press, 2012.
- [23] Myers G. J., The Art of Software Testing, 1st Edition, USA: John Wiley and Sons, 1979.
- [24] Parisi T., WebGL Up and Running, O'REILLY, 2012
- [25] Pilgrim M., HTML5: Up and Running, USA: O'Reilly, 2010.
- [26] Prasad L., Bhadauria S. S., Design Integration Based Testing Using Test Case Generation Technique, International Journal of Scientific & Engineering Research Volume 3, Issue 5, May-2012. ISSN 2229-5518.
- [27] Quy P., Lee J., Kim J., Kim J. and Kim H., Collaborative Experiment and Education based on Networked Virtual Reality. Proceedings of fourth International Conference on Computer Sciences and Convergence Information Technology, Seoul, Korea, 2009.
- [28] Rajib M., Fundamentals of Software Engineering, 2nd edition, Prentice Hall of India, 2003.
- [29] Rasheed R., Web Cube: A New Model for 3-D Web Browsing Based on Hand Gesture Interaction, Master's Thesis, Faculty of Information Technology, Islamic University of Gaza, 2012.
- [30] Rasmusson J., The Agile Samurai: How Agile Masters Deliver Great Software, USA: The Pragmatic Bookshelf, 2010.
- [31] Rivero J. M., Grigera J., Rossi G., Luna E. R. and Koch N., Improving Agility in Model-Driven Web Engineering, 23rd International Conference on Advanced Information Systems Engineering, London, UK, CEUR WS 734, 163-170, June 2011.
- [32] Seidelin J., HTML5 Games: Creating Fun with HTML5, CSS3, and WebGL, United Kingdom: John Wiley & Sons, 2012.
- [33] Sells C. and Griffiths I., Programming Windows Presentation Foundation, 2nd Edition, USA: O'Reilly, 2007.
- [34] Ullman L., Modern JavaScript: Develop and Design, USA: Peachpit Press, 2012
- [35] Vani V. and Mohan S., Interactive 3-D Class Room – A Framework for Web3-D using J3-D and JMF, Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India, India, 2010.
- [36] Weyl E., What's New in CSS3, USA: O'Reilly, 2012.
- [37] Williams J. L., Learning HTML5 Game Programming, USA: Addison-Wesley, 2011.
- [38] Zakas N. C., Professional JavaScript for Web Developers, USA: John Wiley & Sons, 2005.

Appendix A

Library Source Code

```
Side = function
(id,width,height,topCorner_x,topCorner_y,topCorner_z,botCorner_x,botCorner_
y,botCorner_z,pos_x,pos_y,pos_z){
  this.id      = id;
  this.width   = width;
  this.height  = height;
  this.topCorner_x= topCorner_x;
  this.topCorner_y= topCorner_y;
  this.topCorner_z= topCorner_z;
  this.botCorner_x= botCorner_x;
  this.botCorner_y= botCorner_y;
  this.botCorner_z= botCorner_z;
  this.pos_x    = pos_x;
  this.pos_y    = pos_y;
  this.pos_z    = pos_z;
  this.wall =null;
  this.objects =0;
  return this;
};
Side.prototype = {
  constructor: Side,
  getObjectID: function ()
  {
    this.objects ++;
    return this.objects ;
  }
};
```

Figure A.1: Side class code

```

var camera, scene, projector, renderer, stats, wc;
var screen_w, screen_h, screen_d;
var frames_count=0, objects_count = 0,
activeObjectsNumber=0,renderObjectsCount=0;
frames = new Array();
objects = new Array();
renderObjects = new Array();
sides = new Array();
activeDIVs = new Array();
activeObjects= new Array();
directionalLights = new Array();
DIVsCount=0;
var mousedown = false;
var mouseXOnMouseDown, mouseYOnMouseDown, mouseZOnMouseDown;
var SELECTED = null;
WebCube = function (
images_colors, front_tex, left_tex, right_tex, ceil_tex, floor_tex, depth) {
    this.images_colors = images_colors; //0: images, 1: colors
    this.front_tex      = front_tex;
    this.left_tex       = left_tex;
    this.right_tex      = right_tex;
    this.ceil_tex       = ceil_tex;
    this.floor_tex      = floor_tex;
    this.depth          = depth;
    screen_w = window.innerWidth;
    screen_h = window.innerHeight;
    screen_d = window.innerHeight;
    this.counter=0;
    container = document.createElement( 'div' );
    document.body.appendChild( container );
    scene = new THREE.Scene();
    projector = new THREE.Projector();
    renderer = new THREE.WebGLRenderer({'antialias':false});
    renderer.setSize( window.innerWidth, window.innerHeight );
    container.appendChild(renderer.domElement);
    var ambient = new THREE.AmbientLight( 0xe0e0e0 );
    scene.add( ambient );
    directionalLights[1].position.set( screen_w/-2, 0, screen_d/6 );
    directionalLights[2].position.set( screen_w/2, 0, screen_d/6 );
    directionalLights[0].position.set( 0, screen_h/-2, screen_d/6 );
    var cameraPosition = screen_h/depth;
    var vFOV =2*Math.atan(screen_h/( 2 * cameraPosition ) )*180/Math.PI;
    camera = new THREE.PerspectiveCamera(vFOV, window.innerWidth /
window.innerHeight, 1, 10000 );
    camera.position.set ( 0,0,cameraPosition) ;
    sides[0] = new Side(0,screen_w,screen_h,screen_w/-
2,screen_h/2,screen_d,screen_w/2,screen_h/-2,-screen_d,0,0,-screen_d);

```

Figure A.2 (page 1 of 12): WebCube class code

```

sides[1] = new Side(1,screen_d,screen_h,screen_w/-
2,screen_h/2,0,screen_w/-2,screen_h/-2,-screen_d,screen_w/-2,0,screen_d/-
2);
sides[2] = new Side(2,screen_d,screen_h,screen_w/2,screen_h/2,-
screen_d,screen_w/2,screen_h/-2,0,screen_w/2,0,screen_d/-2);
sides[3] = new Side(3,screen_w,screen_d,screen_w/-
2,screen_h/2,0,screen_w/2,screen_h/-2,-screen_d,0,screen_h/2,screen_d/-2);
sides[4] = new Side(4,screen_w,screen_d,screen_w/-2,screen_h/-2,-
screen_d,screen_w/2,screen_h/-2,0,0,screen_h/-2,screen_d/-2);
sides[5] = new Side(0,screen_w,screen_h,screen_w/-2,screen_h/2
,screen_d,screen_w/2,screen_h/-2,-screen_d,0,0,-screen_d);
sides[6] = new Side(1,screen_h,screen_h,screen_w/-2,screen_h/2
,0,screen_w/-2,screen_h/-2,-screen_d,screen_w/-2,0,screen_d/-2);
sides[7] = new Side(2,screen_h,screen_h,screen_w/ 2,screen_h/2 ,-
screen_d,screen_w/2,screen_h/-2,0,screen_w/2,0,screen_d/-2);
sides[8] = new Side(3,screen_w,screen_d,screen_w/-2,screen_h/2
,0,screen_w/2,screen_h/-2,-screen_d,0,screen_h/2,screen_d/-2);
sides[9] = new Side(4,screen_w,screen_d,screen_w/-2,screen_h/-2,-
screen_d,screen_w/2,screen_h/-2,0,0,screen_h/-2,screen_d/-2);

if (images_colors==0)//images
{

this.createTextureRoom(front_tex,left_tex,right_tex,ceil_tex,floor_tex);
}
else
{

this.createSolidRoom(front_tex,left_tex,right_tex,ceil_tex,floor_tex);
}
this.timeReach= 0;
return this;
};
WebCube.prototype = {
constructor: WebCube,
createTextureRoom: function
(front_texture,left_texture,right_texture,ceil_texture,floor_texture)
{
tex = new Array(5);
tex[0] = THREE.ImageUtils.loadTexture(front_texture);
tex[1] = THREE.ImageUtils.loadTexture(left_texture);
tex[2] = THREE.ImageUtils.loadTexture(right_texture);
tex[3] = THREE.ImageUtils.loadTexture(ceil_texture);
tex[4] = THREE.ImageUtils.loadTexture(floor_texture);
for (i=0; i<5;i++)
{
w = new THREE.Mesh( new THREE.PlaneGeometry(
sides[i].width,sides[i].height, 8, 8),
new THREE.MeshLambertMaterial({map:tex[i]}) );

```

Figure A.2 (page 2 of 12): WebCube class code

```

        w.position = new THREE.Vector3(
sides[i].pos_x,sides[i].pos_y,sides[i].pos_z);
        scene.add( w );
        sides[i].wall = w;
        w = new THREE.Mesh( new THREE.PlaneGeometry(
sides[i].width*4,sides[i].height*4, 20, 20),new THREE.MeshBasicMaterial( {
color: 0x000000, opacity: 0, transparent: true, wireframe: true } ));
        w.position = new THREE.Vector3(
sides[i].pos_x,sides[i].pos_y,sides[i].pos_z+1);
        scene.add( w );
        sides[i+5].wall = w;
    }
    sides[1].wall.rotation.y = Math.PI/2;
    sides[2].wall.rotation.y = Math.PI/-2;
    sides[3].wall.rotation.x = Math.PI/2;
    sides[4].wall.rotation.x = Math.PI/-2;
    sides[6].wall.rotation.y = Math.PI/2;
    sides[7].wall.rotation.y = Math.PI/-2;
    sides[8].wall.rotation.x = Math.PI/2;
    sides[9].wall.rotation.x = Math.PI/-2;
},
    createSolidRoom: function
(front_color,left_color,right_color,ceil_color,floor_color)
    {
        colors = new Array(5);

        colors[0] = front_color;
        colors[1] = left_color;
        colors[2] = right_color;
        colors[3] = ceil_color;
        colors[4] = floor_color;
        for (i=0; i<5;i++)
        {
            w = new THREE.Mesh( new THREE.PlaneGeometry(
sides[i].width,sides[i].height, 8, 8),
            new THREE.MeshLambertMaterial( { color: colors[i]} ));
            w.position = new
THREE.Vector3(sides[i].pos_x,sides[i].pos_y,sides[i].pos_z);
            scene.add( w );
            sides[i].wall = w;

            w = new THREE.Mesh( new THREE.PlaneGeometry(
sides[i].width*2,sides[i].height*2, 16, 16),
            new THREE.MeshBasicMaterial( { color: 0x000000, opacity: 0,
transparent: true, wireframe: true } ));
            w.position = new THREE.Vector3(
sides[i].pos_x,sides[i].pos_y,sides[i].pos_z+1);
            scene.add( w );
        }
    }
}

```

Figure A.2 (page 3 of 12): WebCube class code


```

        sides[6].wall.rotation.y = Math.PI/2;
        sides[7].wall.rotation.y = Math.PI/-2;
        sides[8].wall.rotation.x = Math.PI/2;
        sides[9].wall.rotation.x = Math.PI/-2;
    },
    animate: function() {
        this.render();
    },
    render: function () {
        TWEEN.update();
        renderer.render( scene, camera );
        for (var z=0; z<renderObjectsCount;z++)
        {
            renderObjects[z].render();
        }
    },

    addRender: function (obj){
        renderObjects[renderObjectsCount++] = obj;
    },
    removeRender: function ( obj ) {
        for (var z=0; z<renderObjectsCount;z++)
        {
            if ( renderObjects[i] == obj)
            {
                for (var y=z; y<renderObjectsCount-1; y++)
                {
                    renderObjects[y] = renderObjects[y+1];
                }
                renderObjectsCount--;
                break;
            }
        }
    },
    addHTMLFrame: function ( frameID, pageURL, thumbnailURL,title,skin,
        wall,inactiveSizeVector3
, inactivePositionVector2, inactiveMovable,
activeSizeVector3, activePositionVector3, activeActionButtonArray, rotate) {
        frames[frames_count] = new HTMLFrame (frameID, pageURL,
        thumbnailURL,title,skin, sides[wall], inactiveSizeVector3
, inactivePositionVector2, inactiveMovable,
activeSizeVector3, activePositionVector3, activeActionButtonArray, rotate);
        frames_count++;
        return frames[frames_count-1];
    },

    addImage: function ( imgURL ,title,skin, ss,inactiveSizeVector3
, inactivePositionVector2, inactiveMovable,

```

Figure A.2 (page 4 of 12): WebCube class code

```

        return objects[objects_count-1];
    },
    addGallery: function ( imgListXMLURL ,title,skin, ss,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,activeSizeVector3,activePositionVe
ctor3, rotate,totalImages,viewedImages) {
        objects[objects_count]=new GalleryObject(imgListXMLURL,title,skin,
            sides[ss],
inactiveSizeVector3,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,rotate,totalImages,viewedImages);
        objects_count++;
        return objects[objects_count-1];
    },
    addBook: function ( imgListXMLURL ,skin, ss,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,activeSizeVector3,
activePositionVector3, rotate,totalImages,color) {
        objects[objects_count]=new BookObject(imgListXMLURL,skin,sides[ss],
inactiveSizeVector3,inactivePositionVector2, inactiveMovable,
activeSizeVector3,activePositionVector3,rotate,totalImages,color);
        objects_count++;
        return objects[objects_count-1];
    },
    addCalendar: function ( skin, ss,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,activeActionButtonArray,rotate) {
        objects[objects_count]=new CalendarObject(skin,
sides[ss],inactiveSizeVector3 ,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,activeActionButtonArray,rotate);
        objects_count++;
        return objects[objects_count-1];
    },
    addVideo: function (videoID, videoURL ,title,skin,
ss,inactiveSizeVector3 ,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,activeActionButtonArray,rotate,info
,loop,autoplay,clickable) {
objects[objects_count]=new VideoObject(videoID,
videoURL,title,skin,sides[ss],inactiveSizeVector3,inactivePositionVector2,
inactiveMovable,activeSizeVector3,activePositionVector3,activeActionButtonA
rray,rotate,info,loop,autoplay,clickable);
        objects_count++;
        return objects[objects_count-1];
    },
    addAudio: function (audioID, audioURL ,title,skin,
ss,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,activeSizeVector3,
activePositionVector3,rotate,loop,autoplay) {
        objects[objects_count]=new AudioObject(audioID, audioURL,title,
skin,sides[ss],inactiveSizeVector3,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,rotate,loop,autoplay);

```

Figure A.2 (page 5 of 12): WebCube class code

```

objects_count++;
    return objects[objects_count-1];
},
    addHyperlink:function(side,BGImg,BGcolor, title,titleColor,titleFont,
titleFontSize,underlined,bold,italic,opacity,
    sizeVector3,positionVector2,movable,rotate,destination,jscode){
    objects[objects_count]= new Hyperlink (sides[side],BGImg,BGcolor,
title,titleColor,titleFont, titleFontSize,underlined,bold,italic,opacity,
    sizeVector3,positionVector2,movable,rotate,destination,jscode);
    objects_count++;
    return objects[objects_count-1];
},
    addSlidingHyperlink:function(side,BGImg,BGcolor,
title,titleColor,titleFont, titleFontSize,underlined,bold,italic,opacity,
    sizeVector3,positionVector2,movable,rotate,destination,jscode){
    objects[objects_count]= new SlidingHyperlink (sides[side],BGImg,BGcolor,
title,titleColor,titleFont, titleFontSize,underlined,bold,italic,opacity,
    sizeVector3,positionVector2,movable,rotate,destination,jscode);
    objects_count++;
    return objects[objects_count-1];
},
    addMarquee:function(side,BGImg,BGcolor, marqueeXML,titleColor,
spacerColor,titleFont, titleFontSize,underlined,bold,italic,opacity,
    sizeVector3,positionVector2,movable,rotate,speed){
    objects[objects_count]= new Marquee (sides[side],BGImg,BGcolor,
marqueeXML,titleColor,spacerColor,titleFont, titleFontSize,underlined,
bold,italic,opacity,sizeVector3,positionVector2,movable,rotate,speed);
    objects_count++;
    return objects[objects_count-1];
},
    removeObject:function(object){
    for (var i=0; i<objects.length;i++)
    {
        if ( objects[i] == object)
        {
            for (var j=i; j<objects.length-1; j++)
            {
                objects[j]= objects[j+1];
            }
            objects_count--;
            break;
        }
    }
},
    onMouseDown: function ( event ) {
this.mousedown=true;
    var vector = new THREE.Vector3((event.clientX/window.innerWidth)*2-1,-
(event.clientY/window.innerHeight) * 2 + 1, 0.5 );
    projector.unprojectVector( vector, camera );
mouseYOnMouseDown=event.clientY;

```

Figure A.2 (page 6 of 12): WebCube class code

```

if ( intersects.length > 0 ) {
    for (i=0; i<frames_count;i++)
        if ( frames[i].checkIntersection(intersects[0].object)==true)
            {
                SELECTED = frames[i];
                break;
            }
    for (i=0; i<objects_count;i++)
        if ( objects[i].checkIntersection(intersects[0].object)==true)
            {
                SELECTED = objects[i];
                break;
            }
    if (SELECTED !=null)
        {
            if (SELECTED.is_Active()== false)
                {
                    var intersects = ray.intersectObject(
sides[SELECTED.inactiveSide.id+5].wall );
                    sides[SELECTED.inactiveSide.id+5].wall.position.copy(
SELECTED.getCurrentInactivePosition());
                    if (SELECTED.inactiveSide.id==0)
                        {
                            sides[SELECTED.inactiveSide.id+5].wall.position.z = -screen_d;
                        }
                    else if (SELECTED.inactiveSide.id==1)
                        {
                            sides[SELECTED.inactiveSide.id+5].wall.position.x=screen_w/-2;
                        }
                    else if (SELECTED.inactiveSide.id==2)
                        {
                            sides[SELECTED.inactiveSide.id+5].wall.position.x = screen_w/2;
                        }
                    else if (SELECTED.inactiveSide.id==3)
                        {
                            sides[SELECTED.inactiveSide.id+5].wall.position.y = screen_h/2;
                        }
                    else if (SELECTED.inactiveSide.id==4)
                        {
                            sides[SELECTED.inactiveSide.id+5].wall.position.y = screen_h/-2;
                        }
                    SELECTED.offset.copy( intersects[ 0 ].point.subSelf(
sides[SELECTED.inactiveSide.id+5].wall.position ));
                }
            SELECTED.onMouseDown(event, ray);
        }
    }
}

```

Figure A.2 (page 7 of 12): WebCube class code

```

var ray=new THREE.Ray(camera.position,vector.subSelf(camera.position
).normalize());
intersects = ray.intersectScene( scene );
if ( intersects.length > 0 ) {
    for (i=0; i<frames_count;i++)
        if ( frames[i].checkIntersection(intersects[0].object)==true)
            {
                SELECTED = frames[i];
                break;
            }
    for (i=0; i<objects_count;i++)
        if ( objects[i].checkIntersection(intersects[0].object)==true)
            {
                SELECTED = objects[i];
                break;
            }
    if (SELECTED)
        {
            if ((mouseXOnMouseDown==event.clientX) && (mouseYOnMouseDown
==event.clientY))
                {
                    SELECTED.onMouseClicked(event);
                }
            SELECTED.onMouseUp(event);
        }
}
for (var z=0; z<renderObjectsCount;z++)
    {
        renderObjects[z].onMouseUp(event);
    }
SELECTED = null;
},
onMouseWheel: function ( event ) {
    if (SELECTED)
        {
            SELECTED.onMouseWheel(event);
        }
},
onMouseMove: function ( event ) {
    if (this.mousedown)
        {
            intersects = ray.intersectScene( scene );
            if (SELECTED)
                {
                    if (SELECTED.is_Active()==false)
                        {
                            if (SELECTED.inactiveMovable)//&&(SELECTED.isActive == false))
                                {
                                    var intersects=ray.intersectObject(sides[SELECTED.inactiveSide.id+5].wall);

```

Figure A.2 (page 8 of 12): WebCube class code

```

        SELECTED.changePosition( intersects[ 0 ].point.subSelf(
SELECTED.offset ));
        originalPsition = SELECTED.getInactivePosition();
        currentPosition = SELECTED.getCurrentInactivePosition();
        selectedSide = SELECTED.inactiveSide;
        frameSize = SELECTED.getDimentions();
        if (SELECTED.inactiveSide.id==0)
        {
            currentPosition.z = originalPsition.z;
            if (currentPosition.x <(selectedSide.topCorner_x+frameSize.x/2))
                currentPosition.x = selectedSide.topCorner_x+frameSize.x/2;
            if (currentPosition.x >selectedSide.botCorner_x-frameSize.x/2)
                currentPosition.x = selectedSide.botCorner_x-frameSize.x/2;
            if (currentPosition.y >selectedSide.topCorner_y-frameSize.y/2)
                currentPosition.y = selectedSide.topCorner_y-frameSize.y/2;
            if (currentPosition.y <selectedSide.botCorner_y+frameSize.y/2)
                currentPosition.y = selectedSide.botCorner_y+frameSize.y/2;
        }
        else if (selectedSide.id==1)
        {
            currentPosition.x = SELECTED.inactivePositionP.x;
            if (currentPosition.z >(selectedSide.topCorner_z-frameSize.x/2))
                currentPosition.z = selectedSide.topCorner_z-frameSize.x/2;
            if (currentPosition.z <selectedSide.botCorner_z+frameSize.x/2)
                currentPosition.z = selectedSide.botCorner_z+frameSize.x/2;
            if (currentPosition.y >selectedSide.topCorner_y-frameSize.y/2)
                currentPosition.y = selectedSide.topCorner_y-frameSize.y/2;
            if (currentPosition.y <selectedSide.botCorner_y+frameSize.y/2)
                currentPosition.y = selectedSide.botCorner_y+frameSize.y/2;
        }
        else if (selectedSide.id==2)
        {
            currentPosition.x = SELECTED.inactivePositionP.x;
            if (currentPosition.z <(selectedSide.topCorner_z+frameSize.x/2))
                currentPosition.z = selectedSide.topCorner_z+frameSize.x/2;
            if (currentPosition.z >selectedSide.botCorner_z-frameSize.x/2)
                currentPosition.z = selectedSide.botCorner_z-frameSize.x/2;
            if (currentPosition.y >selectedSide.topCorner_y-frameSize.y/2)
                currentPosition.y = selectedSide.topCorner_y-frameSize.y/2;
            if (currentPosition.y <selectedSide.botCorner_y+frameSize.y/2)
                currentPosition.y = selectedSide.botCorner_y+frameSize.y/2;
        }
        else if (selectedSide.id==3)
        {
            currentPosition.y = SELECTED.inactivePositionP.y;
            if (currentPosition.x <(selectedSide.topCorner_x+frameSize.x/2))
                currentPosition.x = selectedSide.topCorner_x+frameSize.x/2;
            if (currentPosition.x >selectedSide.botCorner_x-frameSize.x/2)
                currentPosition.x = selectedSide.botCorner_x-frameSize.x/2;
        }
    }
}

```

Figure A.2 (page 9 of 12): WebCube class code


```

else if (selectedSide.id==4)
{
    currentPosition.y = SELECTED.inactivePositionP.y;
    if (currentPosition.x <(selectedSide.topCorner_x+frameSize.x/2))
currentPosition.x = selectedSide.topCorner_x+frameSize.x/2;
    if (currentPosition.x >selectedSide.botCorner_x-frameSize.x/2)
        currentPosition.x = selectedSide.botCorner_x-frameSize.x/2;
    if (currentPosition.z <(selectedSide.topCorner_z+frameSize.y/2))
        currentPosition.z = selectedSide.topCorner_z+frameSize.y/2;
    if (currentPosition.z >selectedSide.botCorner_z-frameSize.y/2)
        currentPosition.z = selectedSide.botCorner_z-frameSize.y/2;
    }
    originalPsition.copy(currentPosition);
}
}
SELECTED.onMouseMove(event, ray);
}
}
else
{
    var vector = new THREE.Vector3((event.clientX/window.innerWidth)*2-1,-
(event.clientY/window.innerHeight) * 2 + 1, 0.5 );
    projector.unprojectVector( vector, camera );
    var ray = new THREE.Ray( camera.position, vector.subSelf(
camera.position ).normalize() );
    var intersects = ray.intersectScene( scene );
    SELECTED = null;
    if ( intersects.length > 0 ) {
        for (i=0; i<frames_count;i++)
            if ( frames[i].checkIntersection(intersects[0].object)==true)
                {
                    SELECTED = frames[i];
                    break;
                }
        for (i=0; i<objects_count;i++)
            if (
objects[i].checkIntersection(intersects[0].object)==true)
                {
                    SELECTED = objects[i];

                    break;
                }

            if (SELECTED!=null)
                SELECTED.onMouseMove(event, ray);
        }
    }
},

```

Figure A.2 (page 10 of 12): WebCube class code

```

restore: function ( frameid ) {
    for (i=0; i<frames_count;i++)
    {    if ( frames[i].frameID == frameid)
        {
            frames[i].restore();
            break;
        }
    }
},
addDIV: function ( DIV ) {
    activeDIVs[DIVsCount++]= DIV;
},
removeDIV: function ( DIV ) {
    for (i=0; i<DIVsCount;i++)
    {    if ( activeDIVs[i] == DIV)
        {
            for (j=i; j<DIVsCount-1; j++)
            {
                activeDIVs[j]= activeDIVs[j+1];
            }
            DIVsCount--;
            break;
        }
    }
    if (DIVsCount>0)
    {
        //activeDIVs[DIVsCount-1].style.opacity=1;
        activeDIVs[DIVsCount-1].style.zIndex=100;
    }
},
activateDIV: function ( DIV , frameID) {

    this.removeDIV ( DIV );

    this.addDIV( DIV );
    for (i=0; i<DIVsCount;i++)
    {
        if (activeDIVs[i]==DIV)
        {
            activeDIVs[i].style.zIndex=100;
        }
    }
    for (i=0; i<frames_count;i++)

```

Figure A.2 (page 11 of 12): WebCube class code

```

        {
            if (frames[i].frameID == frameID)
            {
                frames[i].changeZIndex(0);
            }
            else
            {
                frames[i].changeZIndex(-1-i);
            }
        }
    },
    addActiveObject: function ( obj) {

        activeObjects[activeObjectsNumber++]=obj;

        return (activeObjectsNumber-1)*5;
    },
    removeActiveObject: function ( obj) {
        for (var ii=0; ii<activeObjectsNumber;ii++)
        {
            if (activeObjects==obj)
            {
                for (jj=ii; jj<activeObjectsNumber;jj++)
                {
                    activeObjects[jj] = activeObjects[jj+1];
                    activeObjects[jj].changePositionZ (jj*5);
                }
                break;
            }
        }
    }
};

```

Figure A.2 (page 12 of 12): WebCube class code

```

FrameGeometry = function ( width, height, depth,dim, imagel,image2,color
,title,title_color,title_font,title_font_Size) {

    THREE.Geometry.call( this );
    this.edge_width = screen_h/50;
    this.title_height = screen_h/22;
    this.total_width = width;
    this.total_height = height;

    this.center_width = width-2*this.edge_width;
    this.center_height = height-2*this.edge_width-this.title_height;
    var scope = this,
    width_half = width / 2,
    height_half = height / 2,
    depth_half = depth / 2;

    var textHolder = document.createElement( 'canvas' );
    var ctext = textHolder.getContext('2d');
    textHolder.width = this.center_width;
    textHolder.height = this.title_height;
    var bg_mat;

    var center_mat=new
THREE.MeshLambertMaterial({map:THREE.ImageUtils.loadTexture(image1)/*, opaci
ty:opacity,transparent:true*/});

    bg_mat=new THREE.MeshBasicMaterial(
{map:THREE.ImageUtils.loadTexture(image2),opacity:1,transparent:true} );
    ctext.fillStyle = color.getContextStyle();;
    ctext.fillRect(0, 0, this.center_width, this.title_height);
    ctext.font = title_font_Size+"px "+title_font;//"Arial";
    ctext.textAlign = "left";
    ctext.fillStyle = title_color.getContextStyle();
    ctext.fillText(title, 10, title_font_Size*1);
    var title_material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(textHolder), opacity:1, transparent:false});
    title_material.map.needsUpdate = true;

    var matt=new THREE.MeshBasicMaterial( { color: 0x00ff00 } );
    buildPlane( 'x', 'y', 1, - 1, this.center_width, this.center_height,
depth_half+3, center_mat,
0, (this.title_height-this.edge_width)/2 );

    buildPlane( 'x', 'y', 1, - 1, width, height, depth_half,
bg_mat,0,0); buildPlane( 'x', 'y', 1, - 1, this.center_width,
this.title_height, depth_half+1, title_material,
function buildPlane( u, v, udir, vdir, width, height, depth, material
,shiftx,shifty) {

```

Figure A.3 (page 1 of 11): HTMLFrame class code

```

    gridY = 1,
    width_half = width / 2,
    height_half = height / 2,
    offset = scope.vertices.length;
    w = 'z';
    var gridX1 = gridX + 1,
        gridY1 = gridY + 1,
        segment_width = width / gridX,
        segment_height = height / gridY;

    for( iy = 0; iy < gridY1; iy++ ) {
        for( ix = 0; ix < gridX1; ix++ ) {
            var vector = new THREE.Vector3();
            vector[ u ] = ( ix * segment_width - width_half ) *
udir+shiftx;
            vector[ v ]=(iy * segment_height - height_half ) * vdir-shifty;
            vector[ w ] = depth;
            scope.vertices.push( new THREE.Vertex( vector ) );
        }
    }
    for( iy = 0; iy < gridY; iy++ ) {
        for( ix = 0; ix < gridX; ix++ ) {
            var a = ix + gridX1 * iy;
            var b = ix + gridX1 * ( iy + 1 );
            var c = ( ix + 1 ) + gridX1 * ( iy + 1 );
            var d = ( ix + 1 ) + gridX1 * iy;
            scope.faces.push( new THREE.Face4( a + offset, b + offset, c +
offset, d + offset, null, null, material ) );
            scope.faceVertexUvs[ 0 ].push( [
                new THREE.UV( ix / gridX, iy / gridY ),
                new THREE.UV( ix / gridX, ( iy + 1 ) / gridY ),
                new THREE.UV( ( ix + 1 ) / gridX, ( iy + 1 ) / gridY ),
                new THREE.UV( ( ix + 1 ) / gridX, iy / gridY )
            ] );
        }
    }
};
FrameGeometry.prototype = new THREE.Geometry();
FrameGeometry.prototype.constructor = FrameGeometry;
HTMLFrame = function ( frameID ,pageURL, thumbnailURL,title,skin,
                        side,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,activeActionButtonArray,rotate) {
    this.frameID = frameID;
    this.inactiveThumbnailURL = thumbnailURL;

```

Figure A.3 (page 2 of 11): HTMLFrame class code

```

this.inactivePosition      = inactivePositionVector2;
this.inactiveMovable      = inactiveMovable;
  this.activePageURL      = pageURL;
this.activeSize           = activeSizeVector3;
this.activePosition       = activePositionVector3;
this.activeActionbuttons  = activeActionButtonArray;
this.isActive = false;
  this.inactiveSizeP = null;
  this.activeSizeP = null;
  this.inactivePositionP=null;
  this.activePositionP=null;
  this.frame = null;
  this.mouseXOnMouseDown=0;
  this.mouseYOnMouseDown=0;
  this.mouseZOnMouseDown=0;
  this.mousePositionOnMouseDownActive=null;
  this.lastActivePositionP=null;
  this.intersectPointMouseDown=null;
  this.reflectivePlane = null;
  this.mousedown = false;
  this.offset=new THREE.Vector3();
  this.rotation=new THREE.Vector3(0,0,0);
  this.frameNumber=side.getObjectID();
  this.timeReach=0;
  this.afterWaitOp = 0;

var xmlhttp;
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
xmlhttp.open("GET",this.skin+"skin.xml",false)
this.prevButton
=(x[0].getElementsByTagName("prevButton")[0].childNodes[0].nodeValue);
  this.nextButton
=(x[0].getElementsByTagName("nextButton")[0].childNodes[0].nodeValue);
  this.refreshButton
=(x[0].getElementsByTagName("refreshButton")[0].childNodes[0].nodeValue);
  this.homeButton
=(x[0].getElementsByTagName("homeButton")[0].childNodes[0].nodeValue);
  this.restoreButton
=(x[0].getElementsByTagName("restoreButton")[0].childNodes[0].nodeValue);
  this.BGColor
=(x[0].getElementsByTagName("BGColor")[0].childNodes[0].nodeValue);
  this.titleColor
=(x[0].getElementsByTagName("titleColor")[0].childNodes[0].nodeValue);

```

Figure A.3 (page 3 of 11): HTMLFrame class code


```

this.titleColor          = new THREE.Color(this.titleColor);

    this.BGColor = new THREE.Color(this.BGColor);

this.BGimg=(x[0].getElementsByTagName("BGimg")[0].childNodes[0].nodeValue);
    this.inactiveSizeP = new THREE.Vector3(this.inactiveSize.x*side.width,
this.inactiveSize.y*side.height, this.inactiveSize.z);
    this.activeSizeP = new THREE.Vector3(this.activeSize.x*screen_w,
this.activeSize.y*screen_h, this.activeSize.z);
    this.activePositionP = new THREE.Vector3(screen_w
*this.activePosition.x+this.activeSizeP.x/2-screen_w/2,
screen_h/2-
(screen_h*this.activePosition.y+this.activeSizeP.y/2),
5 );

    this.lastActivePositionP.z =-1;

    fg = new FrameGeometry( this.inactiveSizeP.x, this.inactiveSizeP.y,
this.inactiveSizeP.z, 80,this.inactiveThumbnailURL,

this.skin+this.BGimg,this.BGColor,this.title,this.titleColor,"Arial",screen
_h/40,this.inactiveOpacity);

    this.frame = new THREE.Mesh(fg,new THREE.MeshFaceMaterial() );

    if (this.inactiveSide.id==0)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
            3*this.frameNumber-screen_d );
        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==1)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+3*this.frameNumber,
            this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
            this.inactiveSide.topCorner_z-
this.inactiveSizeP.x/2-(this.inactivePosition.x*side.width) );
        this.rotation.y = Math.PI/2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==2)
    {

```

Figure A.3 (page 4 of 11): HTMLFrame class code

```

        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x -3*this.frameNumber,
            this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),
this.inactiveSide.topCorner_z+this.inactiveSizeP.x/2+(this.inactivePosition
.x*side.width) );

        this.rotation.y = Math.PI/-2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==3)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y -
3*this.frameNumber,
            this.inactiveSide.topCorner_z -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height) );

        this.rotation.x = Math.PI/2;
        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==4)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y +3*this.frameNumber,
            this.inactiveSide.topCorner_z
+this.inactiveSizeP.y/2+(this.inactivePosition.y*side.height) );

        this.rotation.x = Math.PI/-2;
        this.rotation.z = rotate*Math.PI/180;
    }
    this.inactivePrevPositionP = this.inactivePositionP;
    this.frame.position = new THREE.Vector3(
this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z)
;
    scene.add( this.frame );
    this.myDIV = this.createDiv();
    wc.addRender(this);
    return this;
};

```

Figure A.3 (page 5 of 11): HTMLFrame class code

```

HTMLFrame.prototype = {

  constructor: HTMLFrame,

  checkIntersection: function ( object ) {
    if (this.isActive )
    {
      if ( this.reflectivePlane == object)
        return true;
    }
    else
    {
      if ( this.frame == object)
        return true;
    }
    return false;
  },

  onMouseDown: function ( event,ray ) {

    this.mousedown = true;
    if (this.isActive ==true)
    {
      this.mousePositionOnMouseDownActive=new
THREE.Vector2(event.clientX,event.clientY);
      this.reflectivePlane.materials[ 0 ].opacity = 0.7;
      wc.activateDIV(this.myDIV,this.frameID);
    }
  },

  onMouseUp: function ( event ) {
    this.mousedown=false;
    if (this.isActive)
    {
      this.lastActivePositionP.x=this.reflectivePlane.position.x;
      this.lastActivePositionP.y=this.reflectivePlane.position.y;

      this.reflectivePlane.materials[ 0 ].opacity = 0;
      this.myDIV.style.visibility="visible";
    }
  },

  onClick: function ( event ) {
    if (this.isActive)
    {
    }
    else
    {
      new TWEEN.Tween( this.frame.position ).to(
this.lastActivePositionP,
500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
    }
  }
};

```

Figure A.3 (page 6 of 11): HTMLFrame class code

```

        new TWEEN.Tween( this.frame.scale ).to( {
x:this.activeSizeP.x/this.inactiveSizeP.x,
y:this.activeSizeP.y/this.inactiveSizeP.y, z:1}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        new TWEEN.Tween( this.frame.rotation ).to( {x:0, y: 0, z:0},
500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        this.wait(400);
        this.afterWaitOp = 1;
    }
},
onMouseMove: function ( event,ray ) {
    if (this.mousedown)
    {
        if (this.isActive)
        {
            this.reflectivePlane.position.x =
this.lastActivePositionP.x +(event.clientX-
this.mousePositionOnMouseDownActive.x);
            this.reflectivePlane.position.y =
this.lastActivePositionP.y -(event.clientY-
this.mousePositionOnMouseDownActive.y);
            this.myDIV.style.visibility="hidden";
        }
    }
},
createDiv:function( )
{
    var edge_width = screen_h/60;
    var newdiv = document.createElement('div2');
    //document.getElementById
    newdiv.setAttribute('id',this.frameID);
    newdiv.style.width = (this.activeSizeP.x)+"px";
    newdiv.style.height =( this.activeSizeP.y)+"px";
    newdiv.style.left = (screen_w/2+this.activePositionP.x-
this.activeSizeP.x/2)+"px";
    newdiv.style.top = (screen_h/2-this.activePositionP.y-
this.activeSizeP.y/2)+"px";
    newdiv.style.visibility="hidden";
    newdiv.style.position = "absolute";
    var bgcol =
(this.BGColor==null)?"#000000":this.BGColor.getContextStyle();
    if (this.BGimg!=null)
    {
        newdiv.style.backgroundColor=bgcol;
    }
}

```

Figure A.3 (page 7 of 11): HTMLFrame class code

```

        newdiv.innerHTML = '<div style="width:
'+(this.activeSizeP.x)+'px; height: '+
(this.activeSizeP.y)+'px;"><table
border="0" width="100%" height="100%"
cellpadding="0" style="
color:#000000;border-collapse: collapse;
border:'+edge_width+'px solid
transparent;" align="center" >
<tr><td><table><tr><td width="100%"
style="font:16px sans-serif; color:
'+this.titleColor.getContextStyle()+
'">'+this.title+'</td>
<td align="right" width="1%">
</td>
<td align="right"
width="1%">
</td>
<td align="right"
width="1%">
</td>
<td align="right" width="1%">
</td>
<td align="right"></td></tr></table></td></tr>
<tr height = "100%"><td><table width="100%"
height="100%" ><tr
height="100%"><td width="100%"><iframe id="i'+this.frameID+'
"
src="'+this.activePageURL+'
" width="100%" height = "100%"
marginheight="0"
marginwidth="0" frameborder="1"
style="background-color:
#ffffff;color:#000000"
noresize><p>Your browser does not support
iframes.</p></iframe></td></tr></table></td></tr></table></div>';

        this.reflectivePlane = new THREE.Mesh(
new THREE.PlaneGeometry(
this.activeSizeP.x,this.activeSizeP.y, 1, 1),
new THREE.MeshBasicMaterial( {
color: this.BGColor.getHex(), opacity: 0,
transparent: true, wireframe:
false} ));

        this.reflectivePlane.position = new THREE.Vector3(
this.activePositionP.x,this.activePositionP.y,
this.activePositionP.z+5);
        return newdiv;
    },
    changeURL:function( newURL)
    {
        var edge_width = screen_h/60;
        var newdiv = document.createElement('div2');
        newdiv.style.visibility="hidden";
        newdiv.style.position = "absolute";
        var bgcol =
(this.BGColor==null)?"#000000":this.BGColor.getContextStyle();
        if (this.BGimg!=null)
        {
            newdiv.style.backgroundImage
="url("+this.skin+this.BGimg+")";
            newdiv.style.backgroundSize=this.activeSizeP.x+"px "+
this.activeSizeP.y+"px";

```

Figure A.3 (page 8 of 11): HTMLFrame class code

```

    }
    else
    {
        newdiv.style.backgroundColor=bgcol;
    }
    this.activePageURL = newURL;
    newdiv.innerHTML = '<div style="width:
'+(this.activeSizeP.x)+'px; height: '+ (this.activeSizeP.y)+'px;"><table
border="0" width="100%" height="100%" cellpadding="0" style="
color:#000000;border-collapse: collapse; border:'+edge_width+'px solid
transparent;" align="center" > <tr><td><table><tr><td width="100%"
style="font:16px sans-serif; color: '+this.titleColor.getContextStyle()+';
">'+this.title+'</td> <td align="right" width="1%"> </td> <td align="right"
width="1%"> </td> <td align="right"
width="1%"> </td><td align="right" width="1%"> </td> <td align="right"></td></tr></table></td></tr>
<tr height = "100%"><td><table width="100%" height="100%" ><tr
height="100%"><td width="100%"><iframe id="i'+this.frameID+'
src="'+this.activePageURL+'" width="100%" height = "100%" marginheight="0"
marginwidth="0" frameborder="1" style="background-color:
#ffffff;color:#000000" noresize><p>Your browser does not support
iframes.</p></iframe></td></tr></table></td></tr></table></div>';
    this.reflectivePlane = new THREE.Mesh( new THREE.PlaneGeometry(
this.activeSizeP.x,this.activeSizeP.y, 1, 1),new THREE.MeshBasicMaterial( {
color: this.BGColor.getHex(), opacity: 0, transparent: true, wireframe:
false} ));
    this.reflectivePlane.position = new THREE.Vector3(
this.lastActivePositionP.x,this.lastActivePositionP.y,this.lastActivePositi
onP.z+5);
    this.myDIV = newdiv;
},
restore:function( )
{
    this.myDIV.style.visibility="hidden";
    this.reflectivePlane.visible = false;
    scene.remove( this.reflectivePlane );
    wc.removeDIV(this.myDIV);
    new TWEEN.Tween( this.frame.scale ).to( { x:1, y:1, z:1}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();

```

Figure A.3 (page 9 of 11): HTMLFrame class code


```

        new TWEEN.Tween( this.frame.rotation ).to(
{x:this.rotation.x,y:this.rotation.y,z:this.rotation.z}, 500).
        easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        this.isActive = false;
        this.mousedown=false;
    },
    changeZIndex:function( z_Index)
    {
        if (this.isActive)
        {
            this.reflectivePlane.position.z = z_Index;
        }
    },
    wait: function(msec)
    {
        this.timeReach=new Date().getTime()+msec;
    },
    afterWait:function()
    {
        document.body.appendChild(this.myDIV);
        this.myDIV.style.visibility="visible";
        wc.addDIV(this.myDIV);
        scene.add( this.reflectivePlane );
        this.reflectivePlane.visible=true;
        scene.remove(this.frame);

        this.isActive = true;
        wc.activateDIV(this.myDIV,this.frameID);
    },
    render:function()
    {
        if (this.afterWaitOp!=0)
        {
            var t =new Date().getTime();
        }
    },
    getDimentions:function()
    {
        return this.inactiveSizeP;
    },
    getInactivePosition:function()
    {
        return this.inactivePositionP;
    },
    getCurrentInactivePosition:function()
    {
        return this.frame.position;
    },

```

Figure A.3 (page 10 of 11): HTMLFrame class code

```
getRotation:function()  
{  
    return this.rotation;  
},  
getSide:function()  
{  
    return this.inactiveSide;  
},  
is_Active:function()  
{  
    return this.isActive;  
},  
changePosition:function(inactivePositionVector)  
{  
    this.frame.position = inactivePositionVector;  
}  
};
```

Figure A.3 (page 11 of 11): HTMLFrame class code

```

ImageGeometry = function ( dim, image1,image2,color,title,title_color,
title_font,title_font_Size,info) {
    THREE.Geometry.call( this );
    this.titleExist = 1;
    if ((title==null)|| (title==""))
        this.titleExist = 0;

    this.edge_width = dim.x/40;
    this.title_height = screen_h/25;///titleWidth/2;
    this.total_width = dim.x;
    this.total_height = dim.y;

    this.center_width = dim.x-2*this.edge_width;
    this.center_height = dim.y-2*this.edge_width-
    this.titleExist*this.title_height;

    var scope = this,
    width_half = dim.x / 2,
    height_half = dim.y / 2,
    depth_half = dim.z / 2;

    var bg_mat;

    var center_mat=new
THREE.MeshLambertMaterial({map:THREE.ImageUtils.loadTexture(image1)});

    if (image2==null)
    {
        bg_mat=new THREE.MeshBasicMaterial( { color:color.getHex() } );
    }
    else
    {
        bg_mat=new THREE.MeshBasicMaterial(
{map:THREE.ImageUtils.loadTexture(image2),opacity:1,transparent:true} );
    }
    buildPlane( 'x', 'y', - 1, -1, dim.x, dim.y, depth_half, bg_mat,0,0
);

    if (info!=null)
    {
        var th = document.createElement( 'canvas' );
        var ctext = th.getContext('2d');
        th.width = this.center_width;
        th.height = dim.y-2*this.edge_width;
        var lineheight = this.title_height/1.5;
        var lines = info.split('\n');
        for (var i = 0; i<lines.length; i++)
            ctext.fillText(lines[i], lineheight, lineheight +
(i*lineheight) );
    }
}

```

Figure A.4(page 1 of 13): ImageObject class code

```

        var back_material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(th), opacity:1, transparent:false});
        back_material.map.needsUpdate = true;

        buildPlane( 'x', 'y', - 1, - 1, this.center_width, dim.y-
2*this.edge_width , depth_half-3, back_material,0,0 );

    }
    else
        buildPlane( 'x', 'y', - 1, - 1, this.center_width, dim.y-
2*this.edge_width , depth_half-3, center_mat,0,0 );
        buildPlane( 'x', 'y', 1, - 1, this.center_width, this.center_height,
depth_half+3, center_mat,0,

this.titleExist*(this.title_height)/-2 );
    buildPlane( 'x', 'y', 1, - 1, dim.x, dim.y, depth_half,
bg_mat,0,0);
    if (this.titleExist==1)
    {

        var textHolder = document.createElement( 'canvas' );
        var ctext = textHolder.getContext('2d');
        textHolder.width = this.center_width;
        textHolder.height = this.title_height;

        ctext.fillStyle = color.getContextStyle();
        ctext.fillRect(0, 0, this.center_width, this.title_height);
        this.title_font_Size = this.title_height/2;
        ctext.font = this.title_font_Size+"px "+title_font;//"Arial";
        ctext.textAlign = "center";
        ctext.fillStyle = title_color.getContextStyle();
        ctext.fillText(title, this.center_width/2,this.title_height/2);
        var title_material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(textHolder), opacity:1, transparent:false});
        title_material.map.needsUpdate = true;
        buildPlane( 'x', 'y', 1, - 1, this.center_width,
this.title_height, depth_half+3, title_material,
0,height_half-this.title_height/2-this.edge_width);
    }
    function buildPlane( u, v, udir, vdir, width, height, depth, material
,shiftx,shifty) {

        var gridX1 = gridX + 1,
            gridY1 = gridY + 1,
            segment_width = width / gridX,
            segment_height = height / gridY;
        for( iy = 0; iy < gridY1; iy++ ) {
            for( ix = 0; ix < gridX1; ix++ ) {

```

Figure A.4(page 2 of 13): ImageObject class code

```

        var vector = new THREE.Vector3();
        vector[ u ] = ( ix * segment_width - width_half ) * udir+shiftx;
        vector[ v ] = ( iy*segment_height-height_half )*vdir-shifty;
        vector[ w ] = depth;
        scope.vertices.push( new THREE.Vertex( vector ) );
    }
}
for( iy = 0; iy < gridY; iy++ ) {
    for( ix = 0; ix < gridX; ix++ ) {
        var a = ix + gridX1 * iy;
        var b = ix + gridX1 * ( iy + 1 );
        var c = ( ix + 1 ) + gridX1 * ( iy + 1 );
        var d = ( ix + 1 ) + gridX1 * iy;
        scope.faces.push( new THREE.Face4( a + offset, b + offset, c +
offset, d + offset, null, null, material ) );
        scope.faceVertexUvs[ 0 ].push( [
            new THREE.UV( ix / gridX, iy / gridY ),
            new THREE.UV( ix / gridX, ( iy + 1 ) / gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, ( iy + 1 ) / gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, iy / gridY )
        ] );
    }
}
}
this.computeCentroids();
this.computeFaceNormals();
};

ImageGeometry.prototype = new THREE.Geometry();
ImageGeometry.prototype.constructor = ImageGeometry;
ImageObject = function ( imgURL, title,skin,
                        side,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,activeActionButtonArray,rotate,info
) {
    this.imgURL = imgURL;
    this.title= title;
    this.inactiveMovable      = inactiveMovable;
    this.info = info;
    this.activeSize           = activeSizeVector3;
    this.activePosition       = activePositionVector3;
    this.activeActionbuttons  = activeActionButtonArray;
    this.isActive = false;
    this.inactiveSizeP = null;
    this.activeSizeP = null;
    this.inactivePositionP=null;
    this.activePositionP=null;
}

```

Figure A.4(page 3 of 13): ImageObject class code

```

this.frame = null;
this.mouseXOnMouseDown=0;
this.mouseYOnMouseDown=0;
this.mouseZOnMouseDown=0;

this.mousePositionOnMouseDownActive=null;
this.lastActivePositionP=null;
this.intersectPointMouseDown=null;
this.mousedown = false;
this.offset=new THREE.Vector3();
this.rotation=new THREE.Vector3(0,0,0);
this.imgNumber=side.getObjectID();
this.buttonArray=new Array();
this.mediaButtons=new Array();
this.timeReach=0;
this.afterWaitOp = 0;
var xmlhttp;
////////////////////
if (window.XMLHttpRequest)
{// code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.open("GET",this.skin+"skin.xml",false)
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
x=xmlDoc.getElementsByTagName("image");

this.BGColor
=(x[0].getElementsByTagName("BGColor")[0].childNodes[0].nodeValue);
this.BGColor = new THREE.Color(this.BGColor);
this.titleColor
=
(x[0].getElementsByTagName("titleColor")[0].childNodes[0].nodeValue);
this.titleColor
= new THREE.Color(this.titleColor);

this.BGimg=(x[0].getElementsByTagName("BGimg")[0].childNodes[0].nodeValue);
if (this.BGimg=="null")
    this.BGimg = "imgwindow.png";
this.inactiveSizeP = new THREE.Vector3(this.inactiveSize.x*side.width,
this.inactiveSize.y*side.height, this.inactiveSize.z);
this.activeSizeP = new THREE.Vector3(this.activeSize.x*screen_w,
this.activeSize.y*screen_h, this.activeSize.z);
this.newActiveSizeP = new THREE.Vector3(0,0,0);
this.activePositionP = new THREE.Vector3(screen_w
*this.activePosition.x+this.activeSizeP.x/2-screen_w/2,screen_h/2 -
(screen_h*this.activePosition.y+this.activeSizeP.y/2), 5 );

```

Figure A.4(page 4 of 13): ImageObject class code

```

    image1, image2, color, title, title_color, title_font, title_font_Size) {
    this.ig = new ImageGeometry(this.activeSizeP,
this.imgURL, this.skin+this.BGimg, this.BGColor,
                                this.title, this.titleColor, "Arial", 30, info);
    this.frame = new THREE.Mesh(this.ig, new THREE.MeshFaceMaterial() );
    this.frame.scale.x =
this.inactiveSizeP.x/this.activeSizeP.x;
    this.frame.scale.y =
this.inactiveSizeP.y/this.activeSizeP.y;
    this.frame.scale.z = 1;

    if (this.inactiveSide.id==0)
    {
        this.inactivePositionP = new THREE.Vector3(
                                this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
                                this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
                                3*this.imgNumber-screen_d );
        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==1)
    {
        this.inactivePositionP = new THREE.Vector3(
                                this.inactiveSide.topCorner_x
+3*this.imgNumber,
                                this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
                                this.inactiveSide.topCorner_z-
this.inactiveSizeP.x/2-(this.inactivePosition.x*side.width) );
        this.rotation.y = Math.PI/2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==2)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x -3*this.imgNumber,
            this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),
            this.inactiveSide.topCorner_z+this.inactiveSizeP.x/2+
(this.inactivePosition.x*side.width) );

        this.rotation.y = Math.PI/-2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==3)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.rotation.z = rotate*Math.PI/180;

```

Figure A.4(page 5 of 13): ImageObject class code


```

    }
    else if (this.inactiveSide.id==4)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y +4*this.imgNumber,
            this.inactiveSide.topCorner_z
+this.inactiveSizeP.y/2+(this.inactivePosition.y*side.height) );
        this.rotation.x = Math.PI/-2;
        this.rotation.z = rotate*Math.PI/180;
    }
    this.frame.rotation.x = this.rotation.x;
    this.frame.rotation.y = this.rotation.y;
    this.frame.rotation.z = this.rotation.z;

    this.frame.position = new THREE.Vector3(
this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z)
;
    this.needUpdateSize=false;
    this.imgObject = new THREE.Object3D();

    this.imgObject.add(this.frame);
    scene.add( this.imgObject );

    this.restoreButtonEnabled=false;
    this.restoreSizeButtonEnabled=false;
    this.flipButtonEnabled=false;

    var ii=0;
    this.b_w=screen_w/35;
    this.b_h=screen_w/35;
    var temp =
(x[0].getElementsByTagName("restoreButton")[0].childNodes[0].nodeValue);
        this.restoreButton = imgButton(skin+temp, this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
this.lastActivePositionP.y+this.activeSizeP.y/2,this.lastActivePositionP.z+
4,0.8);
        this.restoreButtonEnabled=true;
        this.buttonArray[ii++] = this.restoreButton;
        if ((activeActionButtonArray&2 ) ==2)
        {
            var temp =
(x[0].getElementsByTagName("restoreSizeButton")[0].childNodes[0].nodeValue)
;
            this.restoreSizeButton = imgButton(skin+temp,this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,

```

Figure A.4(page 6 of 13): ImageObject class code

```

        var temp =
(x[0].getElementsByTagName("flipButton")[0].childNodes[0].nodeValue);
        this.flipButton = imgButton(skin+temp,this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
this.lastActivePositionP.y+this.activeSizeP.y/2-
ii*this.b_h,this.lastActivePositionP.z+4,0.8);
        this.flipButtonEnabled=true;
        this.buttonArray[ii++] = this.flipButton;
    }

    this.media_b_w=screen_h/25;
    this.media_b_h=screen_h/25;
    this.mediaButtonsHidden=false;

    var jj=0
    temp =
(x[0].getElementsByTagName("resizeButton")[0].childNodes[0].nodeValue);
    this.resizeButton = imgButton(skin+temp,this.media_b_w/2,
this.media_b_h/2,0,0,this.lastActivePositionP.z+3,0.8);
    this.mediaButtons[jj++] = this.resizeButton;

    temp =
(x[0].getElementsByTagName("rotateButton")[0].childNodes[0].nodeValue);
    this.rotateButton = imgButton(skin+temp,this.media_b_w/2,
this.media_b_h/2,0,0,this.lastActivePositionP.z+3,0.8);
    this.mediaButtons[jj++] = this.rotateButton;

    this.selectedButton = null;
    wc.addRender(this);
    return this;
};

ImageObject.prototype = {

    constructor: ImageObject,

    checkIntersection: function ( object ) {
        if (this.isActive )
        {
            for (var ii=0; ii<this.buttonArray.length; ii++)
            {
                if (this.buttonArray[ii] == object)
                {
                    this.selectedButton = this.buttonArray[ii];
                    return true;
                }
            }
        }
        for (var ii=0; ii<this.mediaButtons.length; ii++)

```

Figure A.4(page 7 of 13): ImageObject class code

```

        {
            if (this.mediaButtons[ii] == object)
            {
                this.selectedButton =this.mediaButtons[ii];
                return true;
            }
        }
    }
    else
    {
        if ( this.frame == object)
            return true;
    }
    return false;
},
onMouseDown: function ( event,ray ) {
    this.mousedown = true;
    if (this.isActive ==true)
    {
        this.mousePositionOnMouseDownActive=new
THREE.Vector2(event.clientX,event.clientY);
        this.mouseRotatePosition=new THREE.Vector2(event.clientX,event.clientY);
    }
},
onMouseUp: function ( event ) {
    this.mousedown=false;
    if (this.isActive==true)
    {
        this.lastActivePositionP.x=this.frame.position.x;
        this.lastActivePositionP.y=this.frame.position.y;
        if (this.needUpdateSize==true)
        {
            this.needUpdateSize=false;
            this.activeSizeP.x = this.newActiveSizeP.x;
            this.activeSizeP.y = this.newActiveSizeP.y;
            var position = new
THREE.Vector3(this.frame.position.x,this.frame.position.y,this.frame.position.z);
            this.frame.position = new
THREE.Vector3(position.x,position.y,position.z);
            this.imgObject.add( this.frame );
            this.mediaButtonsHidden = false;
            for (var x=0;x<this.buttonArray.length;x++)
            {
                this.imgObject.add( this.buttonArray[x] );
            }
            for (var x=0;x<this.mediaButtons.length;x++)
            {
                this.imgObject.add( this.mediaButtons[x] );
            }
        }
    }
}

```

Figure A.4(page 8 of 13): ImageObject class code

```

        this.repositionButtons();
    }
},
onClick: function ( event ) {
if (this.isActive)
{
    if (this.selectedButton!=null)
    if (this.selectedButton == this.restoreButton)
    {
        this.restore();
    }
    else if (this.selectedButton == this.flipButton)
    {
        var tt = Math.PI;
        if (this.frame.rotation.y >3)
        {
            tt=0;
            for (var x=0;x<this.mediaButtons.length;x++)
            this.imgObject.add( this.mediaButtons[x] );
        }
        else
            for (var x=0;x<this.mediaButtons.length;x++)
            this.imgObject.remove( this.mediaButtons[x] );
            new
TWEEN.Tween(this.frame.rotation).to({x:0,y:tt,z:0},300).easing(TWEEN.Easing
.Sinusoidal.EaseOut).start();
            }
            else if (this.selectedButton == this.restoreSizeButton)
            {
                new TWEEN.Tween( this.imgObject.rotation ).to( {x:0,y:0,z:0},
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
                new TWEEN.Tween( this.frame.position ).to(
this.lastActivePositionP,
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
                for (var i = 0; i<this.buttonArray.length;i++)
                new TWEEN.Tween( this.buttonArray[i].position ).to(
{x:this.buttonArray[i].position.x,y:this.buttonArray[i].position.y,z:4},
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            }}
        }
        else
        { this.lastActivePositionP.z = wc.addActiveObject(this);
            new TWEEN.Tween( this.frame.position ).to( this.lastActivePositionP,
500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            new TWEEN.Tween( this.frame.scale ).to( { x:1, y:1, z:1}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            new TWEEN.Tween( this.frame.rotation ).to( {x:0, y: 0, z:0},
500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();

```

Figure A.4(page 9 of 13): ImageObject class code

```

        this.wait(500);
        this.afterWaitOp =2;
    }

    },
    onMouseMove: function ( event,ray ) {
        if (this.mousedown)
        {
            if (this.isActive)
            {
                if (this.selectedButton == null)
                {
                    this.frame.position.x = this.lastActivePositionP.x +(event.clientX-
                    this.mousePositionOnMouseDownActive.x)*Math.cos(this.imgObject.rotation.z)-
                    (event.clientY-
                    this.mousePositionOnMouseDownActive.y)*Math.sin(this.imgObject.rotation.z);
                    this.frame.position.y = this.lastActivePositionP.y -(event.clientX-
                    this.mousePositionOnMouseDownActive.x)*Math.sin(this.imgObject.rotation.z)-
                    (event.clientY-
                    this.mousePositionOnMouseDownActive.y)*Math.cos(this.imgObject.rotation.z);
                    this.repositionButtons();
                }
                else if (this.selectedButton == this.resizeButton)
                {
                    if (this.mediaButtonsHidden == false)
                    {
                        for (var x=0;x<this.buttonArray.length;x++)
                        {
                            this.imgObject.remove( this.buttonArray[x] );
                        }
                    }
                    this.newActiveSizeP.x = this.activeSizeP.x+
                    ( event.clientX - this.mousePositionOnMouseDownActive.x)*2*Math.cos(-
                    this.imgObject.rotation.z)-(event.clientY-
                    this.mousePositionOnMouseDownActive.y)*2*Math.sin(this.imgObject.rotation.z
                    );
                    this.newActiveSizeP.y = this.activeSizeP.y+(event.clientX-
                    this.mousePositionOnMouseDownActive.x)*2*Math.sin(this.imgObject.rotation.z
                    )+
                    (event.clientY-this.mousePositionOnMouseDownActive.y)*2*Math.cos(-
                    this.imgObject.rotation.z);
                    new TWEEN.Tween( this.frame.scale ).to(
                    { x:this.newActiveSizeP.x/this.activeSizeP.x,
                    y:this.newActiveSizeP.y/this.activeSizeP.y,
                    z:1}, 1 ).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
                    this.needUpdateSize=true;
                }
            }
        }
    }

```

Figure A.4(page 10 of 13): ImageObject class code

```

else if (this.selectedButton == this.rotateButton)
{
    this.imgObject.rotation.z+= (
        (event.clientY-this.mouseRotatePosition.y)*Math.cos(-
this.imgObject.rotation.z)+(event.clientX-
this.mouseRotatePosition.x)*Math.sin(this.imgObject.rotation.z))*3/this.act
iveSizeP.x;

        this.mouseRotatePosition.x=event.clientX;
        this.mouseRotatePosition.y=event.clientY;
    }
}
},
onMouseWheel: function ( event ) {
    if (this.isActive)
    {
        // WebKit
        var pos=new
THREE.Vector3(this.frame.position.x,this.frame.position.y,this.frame.positi
on.z);

        if ( event.wheelDeltaY ) {

            pos.z+= event.wheelDeltaY ;

            // Opera / Explorer 9

        } else if ( event.wheelDelta ) {

            pos.z += event.wheelDelta ;

            // Firefox

        } else if ( event.detail ) {

            pos.z -= event.detail *20;

        }
        if (pos.z>screen_h)
            pos.z = screen_h;
        else if (pos.z<(screen_d/-1+50))
            pos.z = screen_d/-1+50;
    }
},
repositionButtons: function()
{
    for (ii=0; ii<this.buttonArray.length; ii++)
    {
        this.buttonArray[ii].position.x =
this.frame.position.x+this.activeSizeP.x/2;

```

Figure A.4(page 11 of 13): ImageObject class code

```

        this.buttonArray[ii].position.y =
this.frame.position.y+this.activeSizeP.y/2-ii*this.b_h;
        this.buttonArray[ii].position.z = this.frame.position.z+4;
    }
    this.resizeButton.position.x = this.frame.position.x+
this.ig.center_width/2;
    this.resizeButton.position.y = this.frame.position.y-
this.activeSizeP.y/2+this.ig.edge_width;
    this.resizeButton.position.z = this.frame.position.z+4;
    this.rotateButton.position.x = this.frame.position.x-
this.ig.center_width/2;
    this.rotateButton.position.y = this.frame.position.y+
this.activeSizeP.y/2;
    this.rotateButton.position.z = this.frame.position.z+4;
},
wait: function(msec)
{
    this.timeReach=new Date().getTime()+msec;
},
afterWait: function()
{
    this.isActive = true;
    if (this.afterWaitOp ==2)
    {
        for (var x=0;x<this.buttonArray.length;x++)
        {
            this.imgObject.add( this.buttonArray[x] );
        }

        this.repositionButtons();
    }
    else if (this.afterWaitOp ==1)
    {
        this.repositionButtons();
    }
},
render:function()
{
    if (this.afterWaitOp!=0)
    {
        var t =new Date().getTime();
        if (t>=this.timeReach)
        {
            this.afterWait();
            this.afterWaitOp=0;
        }
    }
},
restore:function( )

```

Figure A.4(page 12 of 13): ImageObject class code


```

{
    this.isActive = false;
    for (var x=0;x<this.buttonArray.length;x++)
    {
        this.imgObject.remove( this.buttonArray[x] );
    }
    for (var x=0;x<this.mediaButtons.length;x++)
    {
        this.imgObject.remove( this.mediaButtons[x] );
    }
    wc.removeActiveObject(this);
    new
TWEEN.Tween(this.imgObject.rotation).to({x:0,y:0,z:0},500).easing(TWEEN.Eas
ing.Sinusoidal.EaseIn).start();
    new TWEEN.Tween( this.frame.position ).to(
this.inactivePositionP,500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
    new TWEEN.Tween( this.frame.scale ).to(
{ x:this.inactiveSizeP.x/this.activeSizeP.x,
y:this.inactiveSizeP.y/this.activeSizeP.y,
z:1}, 500 ).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
    new TWEEN.Tween( this.frame.rotation ).to( this.rotation, 500).
easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
    this.isActive = false;
    this.mousedown=false;
},
getDimentions:function()
{
    return this.inactiveSizeP;
},
getInactivePosition:function()
{
    return this.inactivePositionP;
},
getCurrentInactivePosition:function()
{
    return this.frame.position;
},
getRotation:function()
{
    return this.rotation;
},
is_Active:function()
{
    return this.isActive;
},
changePosition:function(inactivePositionVector)
{
    this.frame.position = inactivePositionVector;
}
};

```

Figure A.4(page 13 of 13): ImageObject class code

```

VideoGeometry = function ( dim,
video,image2,color,title,title_color,title_font,title_font_Size,info) {
  THREE.Geometry.call( this );
  this.titleExist = 1;
  if((title==null)|| (title==""))
  {
    this.title = "";
    this.titleExist = 0;
  }

  this.edge_width = dim.x/40;
  this.title_height = screen_h/14;///titleWidth/2;
  this.total_width = dim.x;
  this.total_height = dim.y;

  this.center_width = dim.x-2*this.edge_width;
  this.center_height = dim.y-2*this.edge_width-
this.titleExist*this.title_height;

  var scope = this,
  width_half = dim.x / 2,
  height_half = dim.y / 2,
  depth_half = dim.z / 2;

  var bg_mat;

  var center_mat=video;//new
THREE.MeshLambertMaterial({map:THREE.ImageUtils.loadTexture(video)/*,opacity:
y:opacity,transparent:true*/});

  if (image2==null)
  {
    bg_mat=new THREE.MeshBasicMaterial( { color:color.getHex() } );
  }
  else
  {
    bg_mat=new THREE.MeshBasicMaterial(
{map:THREE.ImageUtils.loadTexture(image2),opacity:1,transparent:true} );
  }
  buildPlane( 'x', 'y', - 1, -1, dim.x, dim.y, depth_half, bg_mat,0,0
);

  if (info!=null)
  {
    var th = document.createElement( 'canvas' );
    var ctext = th.getContext('2d');
    ctext.fillStyle = "#FFFFFF";
    var lineheight = this.title_height/1.5;
    var lines = info.split('\n');

```

Figure A.5(page 1 of 18): VideoObject class code

```

        for (var i = 0; i<lines.length; i++)
            ctext.fillText(lines[i], lineheight, lineheight +
(i*lineheight) );

        //ctext.fillText(info, this.title_height/2,this.title_height/2);
        var back_material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(th), opacity:1, transparent:false});
        back_material.map.needsUpdate = true;

        buildPlane( 'x', 'y', - 1, - 1, this.center_width, dim.y-
2*this.edge_width , depth_half-3, back_material,0,0 );

    }
    else
        buildPlane( 'x', 'y', - 1, - 1, this.center_width, dim.y-
2*this.edge_width , depth_half-3, center_mat,0,0 );

        buildPlane( 'x', 'y', 1, - 1, this.center_width, this.center_height,
depth_half+3, center_mat,0, this.titleExist*(this.title_height)/-2 );
        buildPlane( 'x', 'y', 1, - 1, dim.x, dim.y, depth_half,
bg_mat,0,0);

    if (this.titleExist==1)
    {

        var textHolder = document.createElement( 'canvas' );
        var ctext = textHolder.getContext('2d');

        ctext.textAlign = "left";
        ctext.fillStyle = title_color.getContextStyle();
        ctext.fillText(title, screen_w/40+10,this.title_height*3/4);
        var title_material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(textHolder), opacity:1, transparent:false});
        title_material.map.needsUpdate = true;
        buildPlane( 'x', 'y', 1, - 1, this.center_width,
this.title_height, depth_half+3, title_material,
0,height_half-this.title_height/2-this.edge_width);
    }
    function buildPlane( u, v, udir, vdir, width, height, depth, material
,shiftx,shifty) {

        var w, ix, iy,
        gridX = 1,
        gridY = 1,
        width_half = width / 2,
        height_half = height / 2,
        offset = scope.vertices.length;
        w = 'z';

```

Figure A.5(page 2 of 18): VideoObject class code

```

var gridX1 = gridX + 1,
    gridY1 = gridY + 1,
    segment_width = width / gridX,
    segment_height = height / gridY;
for( iy = 0; iy < gridY1; iy++ ) {

    for( ix = 0; ix < gridX1; ix++ ) {

        var vector = new THREE.Vector3();
        vector[ u ] = ( ix * segment_width - width_half ) *
udir+shiftx;
        vector[ v ] = ( iy * segment_height - height_half ) *
vdir-shifty;
        vector[ w ] = depth;
        scope.vertices.push( new THREE.Vertex( vector ) );
    }
}
for( iy = 0; iy < gridY; iy++ ) {
    for( ix = 0; ix < gridX; ix++ ) {
        var a = ix + gridX1 * iy;
        var b = ix + gridX1 * ( iy + 1 );
        var c = ( ix + 1 ) + gridX1 * ( iy + 1 );
        var d = ( ix + 1 ) + gridX1 * iy;
        scope.faces.push( new THREE.Face4( a + offset, b +
offset, c + offset, d + offset, null, null, material ) );
        scope.faceVertexUvs[ 0 ].push( [
            ] );
    }
}
}
this.computeCentroids();
this.computeFaceNormals();
};
VideoGeometry.prototype = new THREE.Geometry();
VideoGeometry.prototype.constructor = VideoGeometry;

VideoObject = function (videoID, videoURL, title,skin,
                        side,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,activeActionButtonArray,rotate,info
,autoplay,loop,clickable) {
    this.videoID = videoID;
    this.videoURL = videoURL;
    this.title= title;
    this.skin= skin;
    this.inactiveSide          = side;
    this.inactiveSize          = inactiveSizeVector3;
    this.inactivePosition      = inactivePositionVector2;

```

Figure A.5(page 3 of 18): VideoObject class code

```

this.inactiveMovable      = inactiveMovable;
this.info = info;
this.clickable = clickable;
//this.activevideoURL      = videoURL;
this.activeSize           = activeSizeVector3;
this.activePosition       = activePositionVector3;
this.activeActionbuttons  = activeActionButtonArray;
this.loop = loop;
this.autoplay=autoplay;
this.isActive = false;
this.inactiveSizeP = null;
this.activeSizeP = null;
this.inactivePositionP=null;
this.activePositionP=null;
this.frame = null;
this.mouseXOnMouseDown=0;
this.mouseYOnMouseDown=0;
this.mouseZOnMouseDown=0;

this.mousePositionOnMouseDownActive=null;
this.lastActivePositionP=null;
this.intersectPointMouseDown=null;
this.mousedown = false;
this.offset=new THREE.Vector3();
this.rotation=new THREE.Vector3(0,0,0);
this.vidNumber=side.getObjectID();
this.buttonArray=new Array();
this.mediaButtons=new Array();
this.timeReach=0;
this.afterWaitOp = 0;
var xmlhttp;

////////////////////
if (window.XMLHttpRequest)
{ // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
}
else
{ // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}

xmlhttp.open("GET",this.skin+"skin.xml",false)

xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
this.titleColor      = new THREE.Color(this.titleColor);

```

Figure A.5(page 4 of 18): VideoObject class code

```

this.BGimg=(x[0].getElementsByTagName("BGimg")[0].childNodes[0].nodeValue);
if (this.BGimg=="null")
    this.BGimg = "imgwindow.png";

    this.inactiveSizeP = new THREE.Vector3(this.inactiveSize.x*side.width,
this.inactiveSize.y*side.height, this.inactiveSize.z);
    this.activeSizeP = new THREE.Vector3(this.activeSize.x*screen_w,
this.activeSize.y*screen_h, this.activeSize.z);
    this.newActiveSizeP = new THREE.Vector3(0,0,0);
    this.activePositionP = new THREE.Vector3(screen_w
*this.activePosition.x+this.activeSizeP.x/2-screen_w/2,
        screen_h/2-
(screen_h*this.activePosition.y+this.activeSizeP.y/2),
        5 );

    this.needUpdateSize=false;
    this.lastActivePositionP = new THREE.Vector3();
    this.lastActivePositionP.x =this.activePositionP.x;
    this.lastActivePositionP.y =this.activePositionP.y;
    this.lastActivePositionP.z =-1;

    var vid = document.createElement('video');
    var txt = '<video id="'+this.videoID+'"'

    if (loop)
        txt += ' loop';
    if (autoplay)
        txt += ' autoplay';
    txt+=' style="display:none"><source src="'+this.videoURL+'"'
    ></video>';
    vid.innerHTML=txt;
    document.body.appendChild(vid);

    this.video = document.getElementById(this.videoID );
    var parameters = { color: 0xfffffff, map: this.texture };
    this.material = new THREE.MeshLambertMaterial( parameters );

    this.vg = new VideoGeometry(this.activeSizeP, this.material,
this.skin+this.BGimg,this.BGColor,
        this.title,this.titleColor,"Arial",30,info);
    this.frame = new THREE.Mesh(this.vg,new THREE.MeshFaceMaterial() );

    this.frame.scale.x = this.inactiveSizeP.x/this.activeSizeP.x;
    this.frame.scale.y = this.inactiveSizeP.y/this.activeSizeP.y;
    this.frame.scale.z = 1;

    if (this.inactiveSide.id==0)
    {

```

Figure A.5(page 5 of 18): VideoObject class code

```

        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height), 3*this.vidNumber-screen_d );
            this.rotation.z = rotate*Math.PI/180;
        }
        else if (this.inactiveSide.id==1)
        {
            this.inactivePositionP = new THREE.Vector3(
                this.inactiveSide.topCorner_x
+3*this.vidNumber,
                this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
                this.inactiveSide.topCorner_z-
this.inactiveSizeP.x/2-(this.inactivePosition.x*side.width) );
            this.rotation.y = Math.PI/2;
            this.rotation.x = rotate*Math.PI/180;
        }
        else if (this.inactiveSide.id==2)
        {
            this.inactivePositionP = new THREE.Vector3(
                this.inactiveSide.topCorner_x -3*this.vidNumber,
                this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),
this.inactiveSide.topCorner_z+this.inactiveSizeP.x/2+(this.inactivePosition
.x*side.width) );
        }
        else if (this.inactiveSide.id==3)
        {
            this.inactivePositionP = new THREE.Vector3(
                this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
                this.inactiveSide.topCorner_y -3*this.vidNumber,
                this.inactiveSide.topCorner_z -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height) );
        }
        else if (this.inactiveSide.id==4)
        {
            this.inactivePositionP = new THREE.Vector3(
                this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
                this.inactiveSide.topCorner_y +3*this.vidNumber,
                this.inactiveSide.topCorner_z
+this.inactiveSizeP.y/2+(this.inactivePosition.y*side.height) );
        }
    }
}

```

Figure A.5(page 6 of 18): VideoObject class code


```

        this.rotation.x = Math.PI/-2;
        this.rotation.z = rotate*Math.PI/180;
    }
    this.frame.rotation.x = this.rotation.x;
    this.frame.rotation.y = this.rotation.y;
    this.frame.rotation.z = this.rotation.z;
    this.frame.position = new THREE.Vector3(
this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z)
;
    this.videoObject = new THREE.Object3D();
    this.videoObject.add(this.frame);
    scene.add( this.videoObject );
    this.restoreButtonEnabled=false;
    this.restoreSizeButtonEnabled=false;
    this.flipButtonEnabled=false;
    var ii=0,jj=0,temp;
    this.b_w=screen_w/35;
    this.b_h=screen_w/35;
    temp =
(x[0].getElementsByTagName("restoreButton")[0].childNodes[0].nodeValue);
    this.restoreButton = imgButton(skin+temp, this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
this.lastActivePositionP.y+this.activeSizeP.y/2,this.lastActivePositionP.z+
4,0.8);
    this.restoreButtonEnabled=true;
    this.buttonArray[ii++] = this.restoreButton;
    if ((activeActionButtonArray&2 ) ==2)
    {
        temp =
(x[0].getElementsByTagName("restoreSizeButton")[0].childNodes[0].nodeValue)
;
        this.restoreSizeButton = imgButton(skin+temp,this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
this.lastActivePositionP.y+this.activeSizeP.y/2-
ii*this.b_h,this.lastActivePositionP.z+4,0.8);
        this.restoreSizeButtonEnabled=true;
        this.buttonArray[ii++] = this.restoreSizeButton;
    }
    if ((activeActionButtonArray&4 ) ==4)
    {
        this.flipButtonEnabled=true;
        this.buttonArray[ii++] = this.flipButton;
    }
    this.media_b_w=screen_h/25;
    this.media_b_h=screen_h/25;
    this.mediaButtonsHidden=false;

```

Figure A.5(page 7 of 18): VideoObject class code

```

temp =
(x[0].getElementsByTagName("playButton")[0].childNodes[0].nodeValue);
    this.playButton = imgButton(skin+temp,this.media_b_w,
this.media_b_h,0,0,this.lastActivePositionP.z+4,0.8);
    this.mediaButtons[jjj++] = this.playButton;

temp =
(x[0].getElementsByTagName("pauseButton")[0].childNodes[0].nodeValue);
    this.pauseButton = imgButton(skin+temp,this.media_b_w,
this.media_b_h,0,0,this.lastActivePositionP.z+4,0.8);
    this.mediaButtons[jjj++] = this.pauseButton;

temp =
(x[0].getElementsByTagName("volumeButton")[0].childNodes[0].nodeValue);
    this.volumeButton = imgButton(skin+temp,this.media_b_w,
this.media_b_h,0,0,this.lastActivePositionP.z+4,0.8);
    this.mediaButtons[jjj++] = this.volumeButton;

temp =
(x[0].getElementsByTagName("muteButton")[0].childNodes[0].nodeValue);
    this.muteButton = imgButton(skin+temp,this.media_b_w,
this.media_b_h,0,0,this.lastActivePositionP.z+3,0.8);
    this.mediaButtons[jjj++] = this.muteButton;

temp =
(x[0].getElementsByTagName("resizeButton")[0].childNodes[0].nodeValue);
    this.resizeButton = imgButton(skin+temp,this.media_b_w/2,
this.media_b_h/2,0,0,this.lastActivePositionP.z+3,0.8);
    this.mediaButtons[jjj++] = this.resizeButton;

temp =
(x[0].getElementsByTagName("rotateButton")[0].childNodes[0].nodeValue);
    this.rotateButton = imgButton(skin+temp,this.media_b_w/2,
this.media_b_h/2,0,0,this.lastActivePositionP.z+3,0.8);
    this.mediaButtons[jjj++] = this.rotateButton;
    this.timeBarImg =
(x[0].getElementsByTagName("timeBar")[0].childNodes[0].nodeValue);
    this.timeBar = imgButton(this.skin+this.timeBarImg,this.timeBarWidth,
this.timeBarHeight,0,0,this.lastActivePositionP.z+4,0.8);
    this.mediaButtons[jjj++] = this.timeBar;
temp =
(x[0].getElementsByTagName("volumeBar")[0].childNodes[0].nodeValue);
    this.volumeBar = imgButton(skin+temp,this.volumeBarWidth,
this.volumeBarHeight,0,0,this.lastActivePositionP.z+4,0.8);
    this.mediaButtons[jjj++] = this.volumeBar;

```

Figure A.5(page 8 of 18): VideoObject class code

```

temp =
(x[0].getElementsByTagName("timePointer")[0].childNodes[0].nodeValue);
    this.timePointer = imgButton(skin+temp,this.timePointerWidth,
this.timePointerHeight,0,0,this.lastActivePositionP.z+4,0.8);
    this.mediaButtons[jj++] = this.timePointer;

temp =
(x[0].getElementsByTagName("volumePointer")[0].childNodes[0].nodeValue);
    this.volumePointer = imgButton(skin+temp,this.volumePointerWidth,
this.volumePointerHeight,0,0,this.lastActivePositionP.z+4,0.8);
    this.mediaButtons[jj++] = this.volumePointer;
    this.selectedButton = null;
    wc.addRender(this);

    return this;
};
VideoObject.prototype = {

    constructor: VideoObject,

    checkIntersection: function ( object ) {
        if (this.isActive )
        {
            if ( this.frame == object)
            {
                this.selectedButton = null;
                return true;
            }
            for (var ii=0; ii<this.buttonArray.length; ii++)
            {
                if (this.buttonArray[ii] == object)
                {
                    this.selectedButton =this.buttonArray[ii];
                    return true;
                }
            }
        }
        else
        {
            if ( this.frame == object)
                return true;
        }
        return false;
    },
    onMouseDown: function ( event,ray ) {

```

Figure A.5(page 9of 18): VideoObject class code

```

        this.mousedown = true;
        if (this.isActive ==true)
        {
            this.mousePositionOnMouseDownActive=new
THREE.Vector2(event.clientX,event.clientY);
            this.mouseRotatePosition=new
THREE.Vector2(event.clientX,event.clientY);
        }
    },
    onMouseUp: function ( event ) {
        this.mousedown=false;
        if (this.isActive==true)
        {
            this.lastActivePositionP.x=this.frame.position.x;
            this.lastActivePositionP.y=this.frame.position.y;
            if (this.needUpdateSize==true)
            {
                this.needUpdateSize=false;
                this.videoObject.remove(this.frame);

                this.vg = new VideoGeometry(this.activeSizeP,
this.material, this.skin+this.BGimg,this.BGColor,
                this.title,this.titleColor,"Arial",30,this.info);
                this.frame = new THREE.Mesh(this.vg,new
THREE.MeshFaceMaterial() );

                this.frame.position = new THREE.Vector3(
position.x,position.y,position.z);
                this.videoObject.add( this.frame );
                var j=0;
                for (j=0; this.mediaButtons[j] != this.timeBar; j++);

                this.timeBarWidth = this.vg.center_width;
                this.timeBar =
imgButton(this.skin+this.timeBarImg,this.timeBarWidth,
this.timeBarHeight,0,0,this.lastActivePositionP.z+4,0.8);
                this.mediaButtons[j]=this.timeBar;
                if (this.mediaButtonsHidden == true)
                {
                    this.mediaButtonsHidden = false;
                    for (var x=0;x<this.buttonArray.length;x++)
                    {
                        this.videoObject.add( this.buttonArray[x] );
                    }
                    for (var x=0;x<this.mediaButtons.length;x++)
                    {
                        this.videoObject.add( this.mediaButtons[x] );
                    }
                }
            }
        }
    }
}

```

Figure A.5(page 10 of 18): VideoObject class code

```

        }
        this.repositionButtons();
    }
},
onClick: function ( event ) {
    if (this.isActive)
    {
        if (this.selectedButton!=null)
        {
            if (this.selectedButton == this.restoreButton)
            {
                this.restore();
            }
            else if (this.selectedButton == this.flipButton)
            {
                var tt = Math.PI;
                if (this.frame.rotation.y >3)
                {
                    tt=0;
                    for (var x=0;x<this.mediaButtons.length;x++)
                        this.videoObject.add(this.mediaButtons[x] );
                }
            }
            else
                for (var x=0;x<this.mediaButtons.length;x++)

this.videoObject.remove(this.mediaButtons[x]);
            new
TWEEN.Tween(this.frame.rotation).to({x:0,y:tt,z:0},300).easing(TWEEN.Easing
.Sinusoidal.EaseOut).start();
        }
        else if (this.selectedButton == this.restoreSizeButton)
        {
            new TWEEN.Tween( this.videoObject.rotation ).to(
{x:0,y:0,z:0}, 200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            new TWEEN.Tween( this.frame.position ).to(
this.lastActivePositionP,
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            for (var i = 0; i<this.buttonArray.length;i++)
                new TWEEN.Tween( this.buttonArray[i].position ).to(
{x:this.buttonArray[i].position.x,y:this.buttonArray[i].position.y,z:4},
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            this.afterWaitOp = 1;
        }
        else if (this.selectedButton == this.playButton)
        {
            this.pauseButton.position.z = this.playButton.position.z +1;
        }
        else if (this.selectedButton == this.pauseButton)
        {

```

Figure A.5(page 11 of 18): VideoObject class code

```

        this.video.pause();
        this.pauseButton.position.z = this.playButton.position.z -1;
    }
    else if (this.selectedButton == this.timeBar)
    {
        this.timePointer.position.x = (event.clientX/screen_w-
0.5)*screen_w*Math.cos(-this.videoObject.rotation.z)-
        (event.clientY/window.innerHeight-
0.5)*window.innerHeight*Math.sin(this.videoObject.rotation.z);
        if (this.timePointer.position.x < (this.timeBar.position.x-
this.timeBarWidth/2))
            this.timePointer.position.x = this.timeBar.position.x-
this.timeBarWidth/2;
        else if (this.timePointer.position.x >
(this.timeBar.position.x+this.timeBarWidth/2))
            this.timePointer.position.x =
this.timeBar.position.x+this.timeBarWidth/2;
            this.video.currentTime = ( (this.timePointer.position.x-
this.timeBar.position.x)/this.timeBarWidth+0.5 ) *this.video.duration;

    }
    else if (this.selectedButton == this.volumeBar)
    {
        this.volumePointer.position.x =
(event.clientX/screen_w-0.5)*screen_w*Math.cos(-
this.videoObject.rotation.z)-

        (event.clientY/window.innerHeight-
0.5)*window.innerHeight*Math.sin(this.videoObject.rotation.z);

        if (this.volumePointer.position.x < (this.volumeBar.position.x-
this.volumeBarWidth/2))
            this.volumePointer.position.x =
this.volumeBar.position.x-this.volumeBarWidth/2;
            else if (this.volumePointer.position.x >
(this.volumeBar.position.x+this.volumeBarWidth/2))
                this.volumePointer.position.x =
this.volumeBar.position.x+this.volumeBarWidth/2;
                this.video.volume = (this.volumePointer.position.x-
this.volumeBar.position.x)/this.volumeBarWidth+0.5 ;
            }
        else if (this.selectedButton == this.muteButton)
        {
            this.video.muted=false;
            this.muteButton.position.z = this.volumeButton.position.z-
1;
        }
    }
    else
    {

```

Figure A.5(page 12 of 18): VideoObject class code

```

        if (this.clickable)
        {
            this.lastActivePositionP.z = wc.addActiveObject(this);
            new TWEEN.Tween( this.frame.position ).to(
this.lastActivePositionP,500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start
());
            new TWEEN.Tween( this.frame.scale ).to( { x:1, y:1, z:1},
500 ).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            new TWEEN.Tween( this.frame.rotation ).to( {x:0, y: 0,
z:0}, 500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            this.wait(500);
            this.afterWaitOp =2;
        }
    },
    onMouseMove: function ( event,ray ) {
        if (this.mousedown)
        {
            if (this.isActive)
            {
                if (this.selectedButton == null)
                {
                    this.frame.position.x = this.lastActivePositionP.x
+ (event.clientX-this.mousePositionOnMouseDownActive.x) *
Math.cos(this.videoObject.rotation.z) - (event.clientY-
this.mousePositionOnMouseDownActive.y) * Math.sin(this.videoObject.rotation.z
);
                    this.frame.position.y = this.lastActivePositionP.y -
(event.clientX-this.mousePositionOnMouseDownActive.x) *
Math.sin(this.videoObject.rotation.z) - (event.clientY-
this.mousePositionOnMouseDownActive.y) * Math.cos(this.videoObject.rotation.z
);
                    this.repositionButtons();
                }
                else if (this.selectedButton == this.timePointer)
                {
                    if (this.timePointer.position.x >
(this.timeBar.position.x+this.timeBarWidth/2))
                    this.timePointer.position.x =
this.timeBar.position.x+this.timeBarWidth/2;
                    this.video.currentTime = ( (this.timePointer.position.x-
this.timeBar.position.x)/this.timeBarWidth+0.5 ) * this.video.duration;
                }
                else if (this.selectedButton == this.volumePointer)
                {
                    this.volumePointer.position.x = (event.clientX/screen_w-
0.5) * screen_w * Math.cos(-this.videoObject.rotation.z)
(event.clientY/window.innerHeight-
0.5) * window.innerHeight * Math.sin(this.videoObject.rotation.z);
                }
            }
        }
    }
}

```

Figure A.5(page 13 of 18): VideoObject class code

```

        if (this.volumePointer.position.x < (this.volumeBar.position.x-
this.volumeBarWidth/2))
            this.volumePointer.position.x = this.volumeBar.position.x-
this.volumeBarWidth/2;
        else if (this.volumePointer.position.x >
(this.volumeBar.position.x+this.volumeBarWidth/2))
            this.volumePointer.position.x =
this.volumeBar.position.x+this.volumeBarWidth/2;
            this.video.volume = (this.volumePointer.position.x-
this.volumeBar.position.x)/this.volumeBarWidth+0.5 ;
    }
    else if (this.selectedButton == this.resizeButton)
    {
        if (this.mediaButtonsHidden == false)
        {
            for (var x=0;x<this.buttonArray.length;x++)
            {
                this.videoObject.remove( this.buttonArray[x] );
            }
            for (var x=0;x<this.mediaButtons.length;x++)
            {
                this.videoObject.remove( this.mediaButtons[x] );
            }
            this.mediaButtonsHidden = true;
        }
        this.newActiveSizeP.y = this.activeSizeP.y+(event.clientX-
this.mousePositionOnMouseDownActive.x)*2*Math.sin(this.videoObject.rotation
.z)+ (event.clientY-this.mousePositionOnMouseDownActive.y)*2*Math.cos(-
this.videoObject.rotation.z);
        if (this.newActiveSizeP.x <200)
            this.newActiveSizeP.x = 200;
        if (this.newActiveSizeP.y <150)
            this.newActiveSizeP.y = 150;
        new TWEEN.Tween( this.frame.scale ).to(
            { x:this.newActiveSizeP.x/this.activeSizeP.x,
            y:this.newActiveSizeP.y/this.activeSizeP.y,
            z:1}, 1 ).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        this.needUpdateSize=true;
    }
    else if (this.selectedButton == this.rotateButton)
    {
        this.videoObject.rotation.z+= (
            (event.clientY-this.mouseRotatePosition.y)*Math.cos(-
this.videoObject.rotation.z)+ (event.clientX-
this.mouseRotatePosition.x)*Math.sin(this.videoObject.rotation.z)
            ) *3/this.activeSizeP.x;
        this.mouseRotatePosition.x=event.clientX;
        this.mouseRotatePosition.y=event.clientY;
    }
}

```

Figure A.5(page 14 of 18): VideoObject class code


```

    }
  },
  onMouseWheel: function ( event ) {
    if (this.isActive)
    {
      // WebKit
      var pos=new
THREE.Vector3(this.frame.position.x,this.frame.position.y,this.frame.position.z);

      if ( event.wheelDeltaY ) {
        pos.z+= event.wheelDeltaY ;
      // Opera / Explorer 9
      } else if ( event.wheelDelta ) {
        pos.z += event.wheelDelta ;
      // Firefox
      } else if ( event.detail ) {
        pos.z -= event.detail *20;
      }
      if (pos.z>screen_h)
        pos.z = screen_h;
      else if (pos.z<(screen_d/-1+50))
        pos.z = screen_d/-1+50;
    }
  },
  repositionButtons: function()
  {
    var ii=0,jj=this.media_b_w*2;
    for (ii=0; ii<this.buttonArray.length; ii++)
    {
      this.buttonArray[ii].position.x =
this.frame.position.x+this.activeSizeP.x/2;
      this.buttonArray[ii].position.y =
this.frame.position.y+this.activeSizeP.y/2-ii*this.b_h;
      this.buttonArray[ii].position.z = this.frame.position.z+4;
    }
    this.playButton.position.x = this.frame.position.x-
this.vg.center_width/2+this.media_b_w/2;
    this.playButton.position.y = this.frame.position.y-
this.activeSizeP.y/2+this.media_b_h/2+this.vg.edge_width;
    this.playButton.position.z = this.frame.position.z+4;

    this.pauseButton.position.x = this.playButton.position.x;
    this.pauseButton.position.y = this.playButton.position.y;
    if (this.video.paused===true)
      this.pauseButton.position.z = this.playButton.position.z -1;
    else
      this.pauseButton.position.z = this.playButton.position.z +1;
  }
}

```

Figure A.5(page 15 of 18): VideoObject class code

```

this.volumeButton.position.x = this.frame.position.x+
this.vg.center_width/2-this.media_b_w/2-this.volumeBarWidth-4;
    this.volumeButton.position.y = this.playButton.position.y;
    this.volumeButton.position.z = this.playButton.position.z;

    this.muteButton.position.x = this.volumeButton.position.x;
    this.muteButton.position.y = this.volumeButton.position.y;
    this.muteButton.position.z = this.volumeButton.position.z;
    if (this.video.muted==true)
        this.muteButton.position.z = this.volumeButton.position.z +1;
    else
        this.muteButton.position.z = this.volumeButton.position.z -1;

    this.timeBar.position.x = this.frame.position.x;
    this.timeBar.position.y = this.frame.position.y-
this.activeSizeP.y/2+this.vg.title_height+this.vg.edge_width-
this.volumePointerHeight/2;
    this.timeBar.position.z = this.frame.position.z+4;

    this.volumeBar.position.x = this.frame.position.x+
this.vg.center_width/2-this.volumeBarWidth/2;
    this.volumeBar.position.y = this.volumeButton.position.y ;
    this.volumeBar.position.z = this.frame.position.z+4;

    this.timePointer.position.x =
this.timeBar.position.x+this.timeBarWidth*(this.video.currentTime
/this.video.duration -0.5);
    this.timePointer.position.y = this.timeBar.position.y ;
    this.timePointer.position.z = this.timeBar.position.z+1;
    this.resizeButton.position.x = this.frame.position.x+
this.vg.center_width/2;
    this.resizeButton.position.y = this.frame.position.y-
this.activeSizeP.y/2+this.vg.edge_width;
    this.resizeButton.position.z = this.frame.position.z+4;

    this.rotateButton.position.x = this.frame.position.x-
this.vg.center_width/2;
    this.rotateButton.position.y = this.frame.position.y+
this.activeSizeP.y/2;//-this.vg.edge_width;
    this.rotateButton.position.z = this.frame.position.z+4;

    },
    wait: function(msec)
    {
        this.timeReach=new Date().getTime()+msec;
    },
    afterWait: function()
    {

```

Figure A.5(page 16 of 18): VideoObject class code

```

        if (this.afterWaitOp ==2)
        {
            for (var x=0;x<this.buttonArray.length;x++)
            {
                this.videoObject.add( this.buttonArray[x] );
            }
            for (var x=0;x<this.mediaButtons.length;x++)
            {
                this.videoObject.add( this.mediaButtons[x] );
            }
            this.repositionButtons();
        }
        else if (this.afterWaitOp ==1)
        {
            this.isActive = true;
            this.repositionButtons();
        }
    },
    restore:function( )
    {
        this.isActive = false;
        for (var x=0;x<this.buttonArray.length;x++)
        {
            this.videoObject.remove( this.buttonArray[x] );
        }
        new
TWEEN.Tween(this.videoObject.rotation).to({x:0,y:0,z:0},500).easing(TWEEN.E
asing.Sinusoidal.EaseIn).start();
        new TWEEN.Tween( this.frame.position ).to(
this.inactivePositionP,500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        new TWEEN.Tween( this.frame.scale ).to(
        { x:this.inactiveSizeP.x/this.activeSizeP.x,
          y:this.inactiveSizeP.y/this.activeSizeP.y,
          z:1}, 500 ).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        new TWEEN.Tween( this.frame.rotation ).to( this.rotation, 500).
easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        this.isActive = false;
        this.mousedown=false;
    },
    render:function()
    {
        if (this.afterWaitOp!=0)
        {
            var t =new Date().getTime();
            if (t>=this.timeReach)
            {
                this.afterWait();
                this.afterWaitOp=0;
            }
        }
    }
}

```

Figure A.5(page 17 of 18): VideoObject class code

```

    }
    if ( (this.video.readyState == this.video.HAVE_ENOUGH_DATA) ) {
        if ( this.texture ) this.texture.needsUpdate = true;
    }
    if (this.isActive)
    {
        this.timePointer.position.x =
this.timeBar.position.x+this.timeBarWidth*(this.video.currentTime
/this.video.duration -0.5);
        this.volumePointer.position.x =
this.volumeBar.position.x+this.volumeBarWidth*( this.video.volume -0.5);
        if ((this.video.currentTime ==
this.video.duration)|| (this.video.paused == true))
            this.pauseButton.position.z = this.playButton.position.z -1;
        else
            this.pauseButton.position.z = this.playButton.position.z +1;
    }
},
changeSource:function(source,info)
{
    this.videoURL = source;
    var mp4 = document.getElementById(this.videoID);
    mp4.setAttribute("src",source);
    mp4.load();
},
getDimentions:function()
{
    return this.inactiveSizeP;
},
getInactivePosition:function()
{
    return this.inactivePositionP;
},
getCurrentInactivePosition:function()
{
    return this.frame.position;
},
getRotation:function()
{
    return this.rotation;
},
is_Active:function()
{
    return this.isActive;
},
changePosition:function(inactivePositionVector)
{
    this.frame.position = inactivePositionVector;
}
};

```

Figure A.5(page 18 of 18): VideoObject class code

```

AudioGeometry = function ( dim, audio,image2/*border*/,color/*title bg
color*/ ,title,title_color,title_font,title_font_Size) {
    THREE.Geometry.call( this );
    this.titleExist = 1;
    if(title==null)
        title = "";
    this.edge_width = dim.x/40;
    this.title_height = screen_h/14;
    this.title_width = dim.x-2*this.edge_width;
    this.total_width = dim.x;
    this.total_height = dim.y;
    var scope = this,
    width_half = dim.x / 2,
    height_half = this.total_height / 2,
    depth_half = dim.z / 2;

    var bg_mat;
    if (image2==null)
    {
        bg_mat=new THREE.MeshBasicMaterial( { color:color.getHex()} );
    }
    else
    {
        bg_mat=new THREE.MeshBasicMaterial(
{map:THREE.ImageUtils.loadTexture(image2),opacity:1,transparent:true} );
    }
    buildPlane( 'x', 'y', 1, - 1, this.total_width,
this.total_height, depth_half, bg_mat,0,0); // nx
    var textHolder = document.createElement( 'canvas' );
    var ctext = textHolder.getContext('2d');
    textHolder.width = this.title_width;
    textHolder.height = this.title_height;
    ctext.fillStyle = color.getContextStyle();
    ctext.fillRect(0, 0, this.title_width, this.title_height);
    this.title_font_Size = this.title_height/3;
    ctext.font = this.title_font_Size+"px "+title_font;//"Arial";
    ctext.textAlign = "left";
    ctext.fillStyle = title_color.getContextStyle();
    ctext.fillText(title, screen_w/40+10,this.title_height*3/4);
    var title_material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(textHolder), opacity:1, transparent:false});
    title_material.map.needsUpdate = true;
    buildPlane( 'x', 'y', 1, - 1, this.title_width,
this.title_height, depth_half+3, title_material,0,0); // nx
    function buildPlane( u, v, udir, vdir, width, height, depth, material
,shiftx,shifty) {
        w = 'z';

```

Figure A.6(page 1 of 13): AudioObject class code

```

var gridX1 = gridX + 1,
    gridY1 = gridY + 1,
    segment_width = width / gridX,
    segment_height = height / gridY;
for( iy = 0; iy < gridY1; iy++ ) {
    for( ix = 0; ix < gridX1; ix++ ) {
        var vector = new THREE.Vector3();
        vector[ u ] = ( ix * segment_width - width_half ) *
udir+shiftx;
        vector[ v ] = ( iy * segment_height - height_half ) *
vdir-shifty;
        vector[ w ] = depth;
        scope.vertices.push( new THREE.Vertex( vector ) );
    }
}
for( iy = 0; iy < gridY; iy++ ) {
    for( ix = 0; ix < gridX; ix++ ) {
        var a = ix + gridX1 * iy;
        var b = ix + gridX1 * ( iy + 1 );
        var c = ( ix + 1 ) + gridX1 * ( iy + 1 );
        var d = ( ix + 1 ) + gridX1 * iy;
        scope.faces.push( new THREE.Face4( a + offset, b +
offset, c + offset, d + offset, null, null, material ) );
        scope.faceVertexUvs[ 0 ].push( [
            new THREE.UV( ix / gridX, iy / gridY ),
            new THREE.UV( ix / gridX, ( iy + 1 ) /
gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, ( iy
+ 1 ) / gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, iy /
gridY )
        ] );
    }
}
this.computeCentroids();
this.computeFaceNormals();
};
AudioGeometry.prototype = new THREE.Geometry();
AudioGeometry.prototype.constructor = AudioGeometry;
AudioObject = function (audioID, audioURL, title, skin,
    side, inactiveSizeVector3
, inactivePositionVector2, inactiveMovable, activeSizeVector3, activePositionVe
ctor3,
    rotate, autoplay, loop) {
    this.audioID = audioID;
    this.audioURL = audioURL;
    this.activePosition = activePositionVector3;

```

Figure A.6(page 2 of 13): AudioObject class code

```

this.isActive = false;
this.inactiveSizeP = null;
this.activeSizeP = null;
this.inactivePositionP=null;
this.frame = null;
this.mouseXOnMouseDown=0;
this.mouseYOnMouseDown=0;
this.mouseZOnMouseDown=0;

this.mousePositionOnMouseDownActive=null;
this.lastActivePositionP=null;
this.buttonArray=new Array();
this.mediaButtons=new Array();

this.timeReach=0;
this.afterWaitOp = 0;

var xmlhttp;
////////////////////
if (window.XMLHttpRequest)
{// code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}

xmlhttp.open("GET",this.skin+"skin.xml",false)

xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

x=xmlDoc.getElementsByTagName("audio");

this.BGColor
=(x[0].getElementsByTagName("BGColor")[0].childNodes[0].nodeValue);
this.BGColor          = new THREE.Color(this.BGColor);
this.titleColor      =
(x[0].getElementsByTagName("titleColor")[0].childNodes[0].nodeValue);
this.titleColor      = new THREE.Color(this.titleColor);
this.BGimg=(x[0].getElementsByTagName("BGimg")[0].childNodes[0].nodeValue);
if (this.BGimg=="null")
    this.BGimg = "imgwindow.png";

this.inactiveSizeP = new THREE.Vector3(this.inactiveSize.x*side.width,
this.inactiveSize.y*side.height, this.inactiveSize.z);
this.activeSizeP = new THREE.Vector3(this.activeSize.x*screen_w,
this.activeSize.y*screen_h, this.activeSize.z);
this.newActiveSizeP = new THREE.Vector3(0,0,0);

```

Figure A.6(page 3 of 13): AudioObject class code

```

this.activePositionP = new THREE.Vector3(screen_w
*this.activePosition.x+this.activeSizeP.x/2-screen_w/2,
screen_h/2-
(screen_h*this.activePosition.y+this.activeSizeP.y/2),
5 );

this.needUpdateSize=false;
this.lastActivePositionP = new THREE.Vector3();
this.lastActivePositionP.x =this.activePositionP.x;
this.lastActivePositionP.y =this.activePositionP.y;
this.lastActivePositionP.z =-1;
var aud = document.createElement('audio');
var txt = '<audio id="'+this.audioID+'"'

if (loop)
    txt += ' loop';
if (autoplay)
    txt += ' autoplay';
txt+=' style="display:none"><source src="'+this.audioURL+'"'
></audio>';
aud.innerHTML=txt;
document.body.appendChild(aud);

this.audio = document.getElementById(this.audioID );
this.ag = new AudioGeometry(this.activeSizeP, this.material,
this.skin+this.BGimg,this.BGColor,this.title,this.titleColor,"Arial",30);
this.frame = new THREE.Mesh(this.ag,new THREE.MeshFaceMaterial() );
if (this.inactiveSide.id==0)
{
    this.inactivePositionP = new THREE.Vector3(
this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
3*this.audNumber-screen_d );
    this.rotation.z = rotate*Math.PI/180;
}
else if (this.inactiveSide.id==1)
{
    this.inactivePositionP = new THREE.Vector3(
this.rotation.x = rotate*Math.PI/180;
}
else if (this.inactiveSide.id==2)
{
    this.inactivePositionP = new THREE.Vector3(
this.inactiveSide.topCorner_x -3*this.audNumber,
this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),
this.inactiveSide.topCorner_z+this.inactiveSizeP.x/2+(this.inactivePosition
.x*side.width) );

```

Figure A.6(page 4 of 13): AudioObject class code


```

        this.rotation.y = Math.PI/-2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==3)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y -
3*this.audNumber,
            this.inactiveSide.topCorner_z -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height) );

        this.rotation.x = Math.PI/2;
        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==4)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y +3*this.audNumber,
            this.inactiveSide.topCorner_z
+this.inactiveSizeP.y/2+(this.inactivePosition.y*side.height) );

        this.rotation.x = Math.PI/-2;
        this.rotation.z = rotate*Math.PI/180;
    }
    this.audioObject = new THREE.Object3D();

    this.audioObject.position.set(this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z);
    this.audioObject.rotation.set(this.rotation.x,this.rotation.y,this.rotation.z);
    this.audioObject.scale.set(this.inactiveSizeP.x/this.activeSizeP.x,this.inactiveSizeP.y/this.activeSizeP.y,1);
    this.audioObject.add(this.frame);
    scene.add( this.audioObject );
    this.media_b_w=screen_h/25;
    this.media_b_h=screen_h/25;
    this.mediaButtonsHidden=false;
    this.volumeBarWidth = screen_w/15;
    this.volumeBarHeight = screen_h/75;
    this.timeBarHeight = screen_h/75;
    this.timeBarWidth = this.ag.title_width;
    this.volumePointerWidth = this.media_b_w/2;
    this.volumePointerHeight = this.volumePointerWidth;

```

Figure A.6(page 5 of 13): AudioObject class code

```

this.timePointerWidth = this.media_b_w/2;
    this.timePointerHeight = this.timePointerWidth;

    temp =
(x[0].getElementsByTagName("restoreButton")[0].childNodes[0].nodeValue);
    this.restoreButton = imgButton(skin+temp, this.b_w,
this.b_h, this.lastActivePositionP.x+this.activeSizeP.x/2,

    this.lastActivePositionP.y+this.activeSizeP.y/2, this.lastActivePositionP
.z+4, 0.8);
    this.restoreButtonEnabled=true;

    this.buttonArray[ii++] = this.restoreButton;

    temp =
(x[0].getElementsByTagName("playButton")[0].childNodes[0].nodeValue);
    this.playButton = imgButton(skin+temp, this.media_b_w, this.media_b_h,
-this.ag.title_width/2+this.media_b_w/2,
-this.inactiveSizeP.y/2+this.media_b_h/2+this.ag.edge_width,
    5, 0.8);
    this.mediaButtons[jj++] = this.playButton;
    this.audioObject.add(this.playButton);
    temp =
(x[0].getElementsByTagName("pauseButton")[0].childNodes[0].nodeValue);
    this.pauseButton = imgButton(skin+temp, this.media_b_w, this.media_b_h,
this.playButton.position.x,
this.playButton.position.y,
this.playButton.position.z-1, 0.8);
    this.mediaButtons[jj++] = this.pauseButton;
    this.audioObject.add(this.pauseButton);

    temp =
(x[0].getElementsByTagName("volumeButton")[0].childNodes[0].nodeValue);
    this.volumeButton = imgButton(skin+temp, this.media_b_w, this.media_b_h,

    this.ag.title_width/2-this.media_b_w/2-this.volumeBarWidth-4,

    this.playButton.position.y, this.playButton.position.z, 0.8);
    this.mediaButtons[jj++] = this.volumeButton;
    this.audioObject.add(this.volumeButton);

    temp =
(x[0].getElementsByTagName("muteButton")[0].childNodes[0].nodeValue);
    this.muteButton = imgButton(skin+temp, this.media_b_w,
this.media_b_h, this.volumeButton.position.x,
    this.volumeButton.position.y, this.volumeButton.position.z-1, 0.8);

```

Figure A.6(page 6 of 13): AudioObject class code

```

        this.mediaButtons[jj++] = this.muteButton;
        this.audioObject.add(this.muteButton);

        this.timeBarImg =
(x[0].getElementsByTagName("timeBar")[0].childNodes[0].nodeValue);
        this.timeBar = imgButton(this.skin+this.timeBarImg,this.timeBarWidth,
this.timeBarHeight,0,

this.inactiveSizeP.y/2+this.ag.title_height+this.ag.edge_width-
this.volumePointerHeight/2,

        this.playButton.position.z,0.8);
        this.mediaButtons[jj++] = this.timeBar;
        this.audioObject.add(this.timeBar);

        temp =
(x[0].getElementsByTagName("volumeBar")[0].childNodes[0].nodeValue);
        this.volumeBar = imgButton(skin+temp,this.volumeBarWidth,
this.volumeBarHeight,this.ag.title_width/2-this.volumeBarWidth/2,

                this.volumeButton.position.y,this.volumeButton.position.z,0.8);
        this.mediaButtons[jj++] = this.volumeBar;
        this.audioObject.add(this.volumeBar);
        temp =
(x[0].getElementsByTagName("timePointer")[0].childNodes[0].nodeValue);
        this.timePointer = imgButton(skin+temp,this.timePointerWidth,
this.timePointerHeight,
        this.timeBar.position.x-this.timeBarWidth/2,
        this.timeBar.position.y,this.timeBar.position.z+1,0.8);
        this.mediaButtons[jj++] = this.timePointer;
        this.audioObject.add(this.timePointer);
        temp =
(x[0].getElementsByTagName("volumePointer")[0].childNodes[0].nodeValue);
        this.volumePointer = imgButton(skin+temp,this.volumePointerWidth,
this.volumePointerHeight,
        this.volumeBar.position.x+this.volumeBarWidth/2,

                this.volumeBar.position.y,this.volumeBar.position.z+1,0.8);
        this.mediaButtons[jj++] = this.volumePointer;
        this.audioObject.add(this.volumePointer);

        this.selectedButton = null;
        wc.addRender(this);
        return this;
};
AudioObject.prototype = {
    constructor: AudioObject,
    checkIntersection: function ( object ) {
        if ( this.frame == object)

```

Figure A.6(page 7 of 13): AudioObject class code

```

        {
            this.selectedButton = null;
            return true;
        }
        for (var ii=0; ii<this.buttonArray.length; ii++)
        {
            if (this.buttonArray[ii] == object)
            {
                this.selectedButton =this.buttonArray[ii];
                return true;
            }
        }
        for (var ii=0; ii<this.mediaButtons.length; ii++)
        {
            if (this.mediaButtons[ii] == object)
            {
                this.selectedButton =this.mediaButtons[ii];
                return true;
            }
        }
        return false;
    },
    onMouseDown: function ( event,ray ) {

        this.mousedown = true;
        if (this.isActive ==true)
        {
            this.mousePositionOnMouseDownActive=new
THREE.Vector2(event.clientX,event.clientY);
            this.mouseRotatePosition=new
THREE.Vector2(event.clientX,event.clientY);
        }
    },
    onMouseUp: function ( event ) {
        this.mousedown=false;
        if (this.isActive==true)
        {
            this.lastActivePositionP.x=this.audioObject.position.x;
            this.lastActivePositionP.y=this.audioObject.position.y;
        }
    },
    onMouseClick: function ( event ) {
        if (this.isActive)
        {
            if (this.selectedButton!=null)
            {
                if (this.selectedButton == this.playButton)
                {
                    this.audio.play();
                    this.pauseButton.position.z=this.playButton.position.z +1;

```

Figure A.6(page 8 of 13): AudioObject class code

```

    }
    else if (this.selectedButton == this.pauseButton)
    {
        this.audio.pause();
        this.pauseButton.position.z = this.playButton.position.z -1;
    }

    else if (this.selectedButton == this.timeBar)
    {

        this.timePointer.position.x = (event.clientX/screen_w-
0.5)*screen_w-this.audioObject.position.x;
        if (this.timePointer.position.x <
(this.timeBar.position.x-this.timeBarWidth/2))
            this.timePointer.position.x = this.timeBar.position.x-
this.timeBarWidth/2;
        else if (this.timePointer.position.x >
(this.timeBar.position.x+this.timeBarWidth/2))
            this.timePointer.position.x =
this.timeBar.position.x+this.timeBarWidth/2;
        this.audio.currentTime = ( (this.timePointer.position.x-
this.timeBar.position.x)/this.timeBarWidth+0.5 ) *this.audio.duration;
    }
    else if (this.selectedButton == this.volumeBar)
    {
        this.volumePointer.position.x =
(event.clientX/screen_w-0.5)*screen_w-this.audioObject.position.x;
        if (this.volumePointer.position.x <
(this.volumeBar.position.x-this.volumeBarWidth/2))
            this.volumePointer.position.x =
this.volumeBar.position.x-this.volumeBarWidth/2;

    }
    else if (this.selectedButton == this.volumeButton)
    {
        this.audio.muted=true;
        this.muteButton.position.z =
this.volumeButton.position.z +1;
    }
    else if (this.selectedButton == this.muteButton)
    {
        this.audio.muted=false;
        this.muteButton.position.z =
this.volumeButton.position.z -1;
    }
    }
}
else

```

Figure A.6(page 9 of 13): AudioObject class code

```

        {
            this.lastActivePositionP.z = wc.addActiveObject(this);
            new TWEEN.Tween( this.audioObject.position ).to(
this.lastActivePositionP,500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start
());
            new TWEEN.Tween( this.audioObject.scale ).to( { x:1, y:1, z:1},
500 ).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            new TWEEN.Tween( this.audioObject.rotation ).to( {x:0, y: 0,
z:0}, 500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            this.wait(500);
            this.afterWaitOp =2;
        }
    },
    onMouseMove: function ( event,ray ) {

        if (this.mousedown)
        {

            if (this.isActive)
            {
                if (this.selectedButton == null)
                {
                    this.audioObject.position.x =
this.lastActivePositionP.x +(event.clientX-
this.mousePositionOnMouseDownActive.x);
                    this.audioObject.position.y =
this.lastActivePositionP.y -(event.clientY-
this.mousePositionOnMouseDownActive.y);
                    this.repositionButtons();
                }
                else if (this.selectedButton == this.timePointer)
                {
                    this.timePointer.position.x =
(event.clientX/screen_w-0.5)*screen_w-this.audioObject.position.x;
                    if (this.timePointer.position.x <
(this.timeBar.position.x-this.timeBarWidth/2))
                        this.timePointer.position.x =
this.timeBar.position.x-this.timeBarWidth/2;
                }
                else if (this.selectedButton == this.volumePointer)
                {
                    this.volumePointer.position.x =
(event.clientX/screen_w-0.5)*screen_w-this.audioObject.position.x;

                    if (this.volumePointer.position.x <
(this.volumeBar.position.x-this.volumeBarWidth/2))
                        this.volumePointer.position.x =
this.volumeBar.position.x-this.volumeBarWidth/2;
                }
            }
        }
    }
}

```

Figure A.6(page 10 of 13): AudioObject class code

```

        else if (this.volumePointer.position.x >
(this.volumeBar.position.x+this.volumeBarWidth/2))
            this.volumePointer.position.x =
this.volumeBar.position.x+this.volumeBarWidth/2;
            this.audio.volume = (this.volumePointer.position.x-
this.volumeBar.position.x)/this.volumeBarWidth+0.5 ;
        }
    }
},
onMouseWheel: function ( event ) {
},
repositionButtons: function()
{
    var ii=0,jj=this.media_b_w*2;
    for (ii=0; ii<this.buttonArray.length; ii++)
    {
        this.buttonArray[ii].position.x =
this.frame.position.x+this.activeSizeP.x/2;
        this.buttonArray[ii].position.y =
this.frame.position.y+this.activeSizeP.y/2-ii*this.b_h;
        this.buttonArray[ii].position.z = this.frame.position.z+4;
    }
    this.playButton.position.x = this.frame.position.x-
this.ag.title_width/2+this.media_b_w/2;
    this.playButton.position.y = this.frame.position.y-
this.activeSizeP.y/2+this.media_b_h/2+this.ag.edge_width;
    this.playButton.position.z = this.frame.position.z+4;

    this.pauseButton.position.x = this.playButton.position.x;
    this.pauseButton.position.y = this.playButton.position.y;
    if (this.audio.paused==true)
        this.pauseButton.position.z = this.playButton.position.z -1;
    else
        this.pauseButton.position.z = this.playButton.position.z +1;
    this.volumeButton.position.x = this.frame.position.x+
this.ag.title_width/2-this.media_b_w/2-this.volumeBarWidth-4;
    this.volumeButton.position.y = this.playButton.position.y;
    this.volumeButton.position.z = this.playButton.position.z;

    this.muteButton.position.x = this.volumeButton.position.x;
    this.muteButton.position.y = this.volumeButton.position.y;
    this.muteButton.position.z = this.volumeButton.position.z;
    if (this.audio.muted==true)
        this.muteButton.position.z = this.volumeButton.position.z +1;
}
}
}

```

Figure A.6(page 11 of 13): AudioObject class code

```

        this.volumeBar.position.x = this.frame.position.x+
this.ag.title_width/2-this.volumeBarWidth/2;
        this.volumeBar.position.y = this.volumeButton.position.y ;
        this.volumeBar.position.z = this.frame.position.z+4;
        this.timePointer.position.x =
this.timeBar.position.x+this.timeBarWidth*(this.audio.currentTime
/this.audio.duration -0.5);
        this.timePointer.position.y = this.timeBar.position.y ;
        this.timePointer.position.z = this.timeBar.position.z+1;
        this.volumePointer.position.x =
this.volumeBar.position.x+this.volumeBarWidth*( this.audio.volume -0.5);
        this.volumePointer.position.y = this.volumeBar.position.y ;
        this.volumePointer.position.z = this.volumeBar.position.z+1;
    },
    wait: function(msec)
    {
        this.timeReach=new Date().getTime()+msec;
    },
    afterWait: function()
    {
        this.isActive = true;
        if (this.afterWaitOp ==2)
        {
            for (var x=0;x<this.buttonArray.length;x++)
            {
                this.audioObject.add( this.buttonArray[x] );
            }
            this.repositionButtons();
        }
        else if (this.afterWaitOp ==1)
        {
            this.repositionButtons();
        }
    },
    restore:function( )
    {
        this.isActive = false;
        wc.removeActiveObject(this);
        new
TWEEN.Tween(this.audioObject.rotation).to({x:0,y:0,z:0},500).easing(TWEEN.E
asing.Sinusoidal.EaseIn).start();

        new TWEEN.Tween( this.audioObject.position ).to(
this.inactivePositionP,500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        new TWEEN.Tween( this.audioObject.scale ).to(
            {
x:this.inactiveSizeP.x/this.activeSizeP.x,y:this.inactiveSizeP.y/this.activ
eSizeP.y, z:1}, 500 ).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        new TWEEN.Tween( this.audioObject.rotation ).to( this.rotation,
500). easing(TWEEN.Easing.Sinusoidal.EaseIn).start();

```

Figure A.6(page 12 of 13): AudioObject class code


```

this.isActive = false;
    this.mousedown=false;
},
render:function()
{
    if (this.afterWaitOp!=0)
    {
        var t =new Date().getTime();
        if (t>=this.timeReach)
        {
            this.afterWait();
            this.afterWaitOp=0;
        }
    }
    if ( this.audio.readyState == this.audio.HAVE_ENOUGH_DATA ) {
        if ( this.texture ) this.texture.needsUpdate = true;
    }
    this.timePointer.position.x = this.timeBar.position.x+
this.timeBarWidth*(this.audio.currentTime /this.audio.duration -0.5);
    this.volumePointer.position.x = this.volumeBar.position.x+
this.volumeBarWidth*( this.audio.volume -0.5);
    if ((this.audio.currentTime == this.audio.duration)||
(this.audio.paused == true))
        this.pauseButton.position.z = this.playButton.position.z -1;
    else
        this.pauseButton.position.z = this.playButton.position.z +1;
},
getDimentions:function()
{
    return this.inactiveSizeP;
},
getInactivePosition:function()
{
    return this.inactivePositionP;
},
getCurrentInactivePosition:function()
{
    return this.audioObject.position;
},
getRotation:function()
{
    return this.rotation;
},
changePosition:function(inactivePositionVector)
{
    this.audioObject.position = inactivePositionVector;
}};

```

Figure A.6(page 13 of 13): AudioObject class code

```

GalleryImageGeometry = function ( dim,
imagel,image2,color,title,title_color, title_font,
title_font_Size,info,info_color,info_font,info_font_Size,resize) {
    THREE.Geometry.call( this );
    var texture = imagel;
    this.resize=true;
    if ((resize==null)|| (resize==false))
        this.resize=false;
    this.infoExist = 1;
    if ((info==null)|| (info=="") || (info=="null"))
        this.infoExist = 0;
    this.edge_width = dim.x/40;
    this.title_height = 1*title_font_Size+this.infoExist*info_font_Size;
    this.dimx=dim.x;
    this.dimy=dim.y;
    if((this.resize==1)&&(texture.image.height>0))
    {
        var aa=(this.dimx-2*this.edge_width)/texture.image.width;
        var bb=(this.dimy-2*this.edge_width-
this.title_height)/texture.image.height;
        if (aa<bb)
        {
            this.dimy =
aa*texture.image.height+2*this.edge_width+this.title_height;
        }
        else
        {
            this.dimx = bb*texture.image.width+2*this.edge_width;
        }
    }
    this.total_width = this.dimx;
    this.total_height = this.dimy;
    this.center_width = this.dimx-2*this.edge_width;
    this.center_height = this.dimy-2*this.edge_width-this.title_height;
    var scope = this,
width_half = this.dimx / 2,
height_half = this.dimy / 2,
depth_half = dim.z / 2;
    var bg_mat;
    var center_mat=new THREE.MeshLambertMaterial({map:texture});
    buildPlane( 'x', 'y', 1, - 1, this.center_width, this.center_height,
depth_half+3, center_mat,0,
                                this.title_height/-2 );
    buildPlane( 'x', 'y', 1, - 1, this.dimx, this.dimy, depth_half,
bg_mat,0,0);
    var textHolder = document.createElement( 'canvas' );

```

Figure A.7(page 1 of 15): GalleryObject class code

```

var ctext = textHolder.getContext('2d');
textHolder.width = this.center_width;
textHolder.height = this.title_height;
ctext.fillStyle = color.getContextStyle();
ctext.fillRect(0, 0, this.center_width, this.title_height);
ctext.font = title_font_Size*0.8+"px "+title_font;
ctext.textAlign = "center";
ctext.fillStyle = "#000000";
ctext.fillText(title, this.center_width/2,title_font_Size*0.8);
if (this.infoExist==1)
{
    ctext.font = info_font_Size*0.8+"px "+info_font;//"Arial";
    ctext.textAlign = "center";
    ctext.fillStyle = info_color.getContextStyle();
    ctext.fillText(info,
this.center_width/2,info_font_Size*2/3+title_font_Size*1);
}
var title_material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(textHolder), opacity:1, transparent:false});
title_material.map.needsUpdate = true;
buildPlane( 'x', 'y', 1, - 1, this.center_width, this.title_height,
depth_half+3, title_material,
0,height_half-this.title_height/2-this.edge_width); //
nx
function buildPlane( u, v, udir, vdir, width, height, depth, material
,shiftx,shifty) {
    var w, ix, iy,
    gridX = 1,
    gridY = 1,
    width_half = width / 2,
    height_half = height / 2,
    offset = scope.vertices.length;
    w = 'z';
    var gridX1 = gridX + 1,
    gridY1 = gridY + 1,
    segment_width = width / gridX,
    segment_height = height / gridY;
    for( iy = 0; iy < gridY1; iy++ ) {
        for( ix = 0; ix < gridX1; ix++ ) {
            scope.vertices.push( new THREE.Vertex( vector ) );
        }
    }
    for( iy = 0; iy < gridY; iy++ ) {
        for( ix = 0; ix < gridX; ix++ ) {
            var a = ix + gridX1 * iy;
            var b = ix + gridX1 * ( iy + 1 );
            var c = ( ix + 1 ) + gridX1 * ( iy + 1 );
            var d = ( ix + 1 ) + gridX1 * iy;
            scope.faces.push( new THREE.Face4( a + offset, b +
offset, c + offset, d + offset, null, null, material ) );

```

Figure A.7(page 2 of 15): GalleryObject class code

```

        scope.faceVertexUvs[ 0 ].push( [
            new THREE.UV( ix / gridX, iy / gridY ),
            new THREE.UV( ix / gridX, ( iy + 1 ) / gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, ( iy + 1 ) / gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, iy / gridY )
        ] );
    }
}
}
this.computeCentroids();
this.computeFaceNormals();
};
GalleryImageGeometry.prototype = new THREE.Geometry();
GalleryImageGeometry.prototype.constructor = GalleryImageGeometry;
GalleryObject = function ( imgListXMLURL, title,skin,
                        side,inactiveSizeVector3
,inactivePositionVector2,inactiveMovable,
activeSizeVector3,activePositionVector3,rotate,totalImages,viewedImages) {
    this.imgListXMLURL = imgListXMLURL;
    this.title= title;
    this.skin= skin;
    this.inactiveSide      = side;
    this.inactiveSize      = inactiveSizeVector3;
    this.inactivePosition  = inactivePositionVector2;
    this.inactiveMovable   = inactiveMovable;
    this.activeSize        = activeSizeVector3;
    this.activePosition    = activePositionVector3;
    this.imagesCount = totalImages;
    this.viewedImages = viewedImages;
    this.isActive = false;
    this.inactiveSizeP = null;
    this.activeSizeP = null;
    this.inactivePositionP=null;
    this.activePositionP=null;
    this.intersectPointMouseDown=null;
    this.mousedown = false;
    this.offset=new THREE.Vector3();
    this.rotation=new THREE.Vector3(0,0,0);
    this.baseIndex=side.getObjectID();
    side.getObjectID();
    side.getObjectID();
    this.buttonArray=new Array();
    this.images=new Array();
    this.imageTitle=new Array();
    this.textures=new Array();
    this.timeReach=0;
    this.afterWaitOp = 0;

```

Figure A.7(page 3 of 15): GalleryObject class code

```

var xmlhttp;
////////////////////
if (window.XMLHttpRequest)
{ // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
}
else
{ // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}

xmlhttp.open("GET",this.skin+"skin.xml",false)

xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
x=xmlDoc.getElementsByTagName("gallery");
this.BGColor
=(x[0].getElementsByTagName("BGColor")[0].childNodes[0].nodeValue);
this.BGColor = new THREE.Color(this.BGColor);
this.titleColor
=(x[0].getElementsByTagName("titleColor")[0].childNodes[0].nodeValue);
this.titleColor = new THREE.Color(this.titleColor);
this.titleFont
=(x[0].getElementsByTagName("titleFont")[0].childNodes[0].nodeValue);
this.titleFontSize
=(x[0].getElementsByTagName("titleFontSize")[0].childNodes[0].nodeValue);
this.infoFont
=(x[0].getElementsByTagName("infoFont")[0].childNodes[0].nodeValue);
this.infoFontSize
=(x[0].getElementsByTagName("infoFontSize")[0].childNodes[0].nodeValue);

this.BGimg=(x[0].getElementsByTagName("BGimg")[0].childNodes[0].nodeValue);
if (this.BGimg=="null")
  this.BGimg = "imgwindow.png";
this.inactiveSizeP = new THREE.Vector3(this.inactiveSize.x*side.width,
this.inactiveSize.y*side.height, this.inactiveSize.z);
this.activeSizeP = new THREE.Vector3(this.activeSize.x*screen_w,
this.activeSize.y*screen_h, this.activeSize.z);
this.newActiveSizeP = new THREE.Vector3(0,0,0);
this.activePositionP = new THREE.Vector3(screen_w
*this.activePosition.x+this.activeSizeP.x/2-screen_w/2,
screen_h/2-
(screen_h*this.activePosition.y+this.activeSizeP.y/2),
5 );
this.lastActivePositionP = new THREE.Vector3();
this.lastActivePositionP.x =this.activePositionP.x;
this.lastActivePositionP.y =this.activePositionP.y;
this.lastActivePositionP.z =-1;
if (this.inactiveSide.id==0)
{

```

Figure A.7(page 4 of 15): GalleryObject class code

```

        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
            3*this.baseIndex-screen_d );
        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==1)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+3*this.baseIndex,
            this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
            this.inactiveSide.topCorner_z-
this.inactiveSizeP.x/2-(this.inactivePosition.x*side.width) );
        this.rotation.y = Math.PI/2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==2)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x -3*this.baseIndex,
            this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),
this.inactiveSide.topCorner_z+this.inactiveSizeP.x/2+(this.inactivePosition
.x*side.width) );
        this.rotation.y = Math.PI/-2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==3)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x +this.inactiveSizeP.x/2+
(this.inactivePosition.x*side.width),this.inactiveSide.topCorner_y -
3*this.baseIndex,this.inactiveSide.topCorner_z -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height) );
    }
    else if (this.inactiveSide.id==4)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y
+4*this.baseIndex,
            this.inactiveSide.topCorner_z
+this.inactiveSizeP.y/2+(this.inactivePosition.y*side.height) );
        this.rotation.x = Math.PI/-2;
    }

```

Figure A.7(page 5 of 15): GalleryObject class code

```

this.rotation.z = rotate*Math.PI/180;
    }
    this.igs = new Array();
    this.frames = new Array();
    this.galleryObject = new THREE.Object3D();
    var ii=0;
    this.b_w=screen_w/25;
    this.b_h=screen_w/25;
    var temp =
(x[0].getElementsByTagName("restoreButton")[0].childNodes[0].nodeValue);
    this.restoreButton = imgButton(skin+temp, this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,

this.lastActivePositionP.y+this.activeSizeP.y/2,this.lastActivePositionP.z+
4,0.6);
    var temp =
(x[0].getElementsByTagName("prevButton")[0].childNodes[0].nodeValue);
    this.prevButton = imgButton(skin+temp, this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,

this.lastActivePositionP.y+this.activeSizeP.y/2,this.lastActivePositionP.z+
4,0.6);
    var temp =
(x[0].getElementsByTagName("nextButton")[0].childNodes[0].nodeValue);
    this.nextButton = imgButton(skin+temp, this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
this.lastActivePositionP.y+this.activeSizeP.y/2,this.lastActivePositionP.z+
4,0.6);
    this.buttonArray[ii++] = this.restoreButton;
    this.buttonArray[ii++] = this.prevButton;
    this.buttonArray[ii++] = this.nextButton;
    xmlhttp.open("GET",this.imgListXMLURL,false)
    xmlhttp.send();
    xmlDoc=xmlhttp.responseXML;
    x=xmlDoc.getElementsByTagName("image");
    this.imagesCount = x.length;
    for (var y=0; y<this.imagesCount; y++)
    {
    this.images[y] =
(x[y].getElementsByTagName("URL")[0].childNodes[0].nodeValue);
    this.imageTitle[y] =
(x[y].getElementsByTagName("title")[0].childNodes[0].nodeValue);
    this.textures[y] = THREE.ImageUtils.loadTexture(this.images[y]);
    this.igs[y] = new GalleryImageGeometry(this.activeSizeP,
this.textures[y],this.skin+this.BGimg,
    this.BGColor,this.title,this.titleColor,this.titleFont,this.titleFontSiz
e, this.imageTitle[y],this.infoColor,this.infoFont,this.infoFontSize,true);
    }
    for (var xx=0; xx<3;xx++)
    {

```

Figure A.7(page 6 of 15): GalleryObject class code

```

this.frames[xx] = new THREE.Mesh(this.igs[(xx-
1+this.imagesCount)%this.imagesCount],new THREE.MeshFaceMaterial() );
    this.frames[xx].scale.x = this.inactiveSizeP.x/this.activeSizeP.x;
    this.frames[xx].scale.y = this.inactiveSizeP.y/this.activeSizeP.y;
    this.frames[xx].scale.z = 1;
    var r=0;
    if (xx==0)
        r=-1;
    else if (xx==2)
        r= 1;

    this.frames[xx].rotation.z = r/8;
    this.galleryObject.add(this.frames[xx]);
}
this.frames[0].position = new THREE.Vector3( 0,0,0);
this.frames[1].position = new THREE.Vector3( 0,0,6);
this.frames[2].position = new THREE.Vector3( 0,0,3);
this.frame = this.frames[1];
this.CurrentImageNumber = 0;
this.selectedButton = null;
wc.addRender(this);
return this;
};
GalleryObject.prototype = {
    constructor: GalleryObject,
    checkIntersection: function ( object ) {
        if (this.isActive )
        {
            for (xx=0; xx<this.viewedImages; xx++)
            {
                if ( this.frames[xx] == object)
                {
                    this.selectedButton = null;
                    return true;
                }
            }
            for (var ii=0; ii<this.buttonArray.length; ii++)
            {
                if (this.buttonArray[ii] == object)
                {
                    this.selectedButton = this.buttonArray[ii];
                    return true;
                }
            }
        }
        else
        {
            for (xx=0; xx<3;xx++)

```

Figure A.7(page 7 of 15): GalleryObject class code


```

        {   if ( this.frames[xx] == object)
            {
                return true;
            }
        }
    }
    return false;
},
onMouseDown: function ( event,ray ) {
    this.mousedown = true;
    if (this.isActive ==true)
    {
        this.mousePositionOnMouseDownActive=new
THREE.Vector2(event.clientX,event.clientY);
        this.mouseRotatePosition=new
THREE.Vector2(event.clientX,event.clientY);
    }
},
onMouseUp: function ( event ) {
    this.mousedown=false;
    if (this.isActive==true)
    {
        this.lastActivePositionP.x=this.galleryObject.position.x;
        this.lastActivePositionP.y=this.galleryObject.position.y;
    }
},
onMouseClicked: function ( event ) {
    if (this.isActive)
    {
        if (this.selectedButton!=null)
        {
            if (this.selectedButton == this.restoreButton)
            {
                this.restore();
            }
            else if (this.selectedButton == this.prevButton)
            {
                this.onLeftArrow(null);
            }
            else if (this.selectedButton == this.nextButton)
            {
                this.onRightArrow(null);
            }
        }
    }
    else
    {
        var imgNumber =0;
        for (var xx=0; xx<this.viewedImages;xx++)

```

Figure A.7(page 8 of 15): GalleryObject class code

```

{
    imgNumber = (xx-
this.middleImage+this.CurrentImageNumber);//%this.imagesCount;
    while (imgNumber<0)
        imgNumber = imgNumber + this.imagesCount;
    imgNumber = imgNumber % this.imagesCount;
    this.frames[xx] = new THREE.Mesh(this.igs[imgNumber],new
THREE.MeshFaceMaterial() );
    this.galleryObject.add(this.frames[xx]);
}
    new TWEEN.Tween( this.galleryObject.position ).to(
{x:this.lastActivePositionP.x,y:this.lastActivePositionP.y,z:this.lastActiv
ePositionP.z}, 700).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();

    new TWEEN.Tween( this.galleryObject.rotation ).to( {x:0, y: 0,
z:0}, 700).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
    this.repositionImages();
    this.lastActivePositionP.z = wc.addActiveObject(this);
    this.wait(500);
    this.afterWaitOp =2;
}
},
onMouseMove: function ( event,ray ) {

    if (this.mousedown)
    {
        if (this.isActive)
        {
            if (this.selectedButton == null)
            {
                this.galleryObject.position.x =
this.lastActivePositionP.x + (event.clientX-
this.mousePositionOnMouseDownActive.x);
this.galleryObject.position.y = this.lastActivePositionP.y -
(event.clientY-this.mousePositionOnMouseDownActive.y);
            }
        }
    }
},
onMouseWheel: function ( event ) {
    if (this.isActive)
    {
        // WebKit
        var pos=new
THREE.Vector3(this.galleryObject.position.x,this.galleryObject.position.y,t
his.galleryObject.position.z);
        if ( event.wheelDeltaY ) {
            pos.z+= event.wheelDeltaY ;
        }
        // Opera / Explorer 9
    }
}
}

```

Figure A.7(page 9of 15): GalleryObject class code

```

        } else if ( event.wheelDelta ) {
            pos.z += event.wheelDelta ;
        // Firefox

        } else if ( event.detail ) {

            pos.z -= event.detail *20;

        }
        if (pos.z>screen_h)
            pos.z = screen_h;
        else if (pos.z<(screen_d/-1+50))
            pos.z = screen_d/-1+50;

        this.galleryObject.position= pos;
    }
},
repositionImages: function()
{
    this.middleImage = Math.floor(this.viewedImages/2);
    for (var xx=0; xx<this.viewedImages;xx++)
    {
        var shift = 1;
        if (xx<this.middleImage)
            shift = -1;
        var order = Math.abs(xx-this.middleImage);

        new TWEEN.Tween(this.frames[xx].scale).to ({x:
1/(1+order/1.5),y:1/(1+order/1.5),z:1}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        new TWEEN.Tween(this.frames[xx].rotation).to ({x:
0,y:0,z:order*shift/5}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        if (order==0)
            new TWEEN.Tween(this.frames[xx].position).to ({x:
0,y:0,z:0}, 500 ).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        else
            new TWEEN.Tween(this.frames[xx].position).to ({x:
((order-
1)*this.activeSizeP.x/1.2+this.activeSizeP.x/1.5)*shift/(1+order),y:order*s
creen_h/25,z:order*-3}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
    }
},
onLeftArrow: function(event)
{
    if ((this.isActive)&&(this.afterWaitOp==0))
    {
        middle = middle%this.imagesCount;
        for (var xx=this.viewedImages-1; xx>0;xx--)

```

Figure A.7(page 10 of 15): GalleryObject class code

```

{
    if (xx== (this.middleImage))
    {
        var rot = this.frames[xx-1].rotation.z;
        this.galleryObject.remove(this.frames[xx-1]);
        this.frames[xx] = new THREE.Mesh(this.igs[middle],new
THREE.MeshFaceMaterial() );
        this.frames[xx].position.x = this.frames[xx-1].position.x;
        this.frames[xx].position.y = this.frames[xx-1].position.y;
        this.frames[xx].position.z = this.frames[xx-1].position.z-3;
        this.frames[xx].scale.x = 0.75;
        this.frames[xx].scale.y = 0.75;
        this.frames[xx].scale.z = 1;
        this.frames[xx].rotation.z = rot;
        this.galleryObject.add(this.frames[xx]);
    }
    else
    {
        this.frames[xx] = this.frames[xx-1];
    }
}

    var imgNumber =(this.CurrentImageNumber-this.middleImage);
    while(imgNumber<0)
        imgNumber+=this.imagesCount;
    imgNumber = imgNumber%this.imagesCount;
    this.frames[0] = new THREE.Mesh(this.igs[imgNumber],new
THREE.MeshFaceMaterial() );
    this.frames[0].position.x = this.frames[1].position.x;
    this.frames[0].scale.z = 1;
    this.frames[0].rotation.z = this.frames[1].rotation.z;
    this.galleryObject.add(this.frames[0]);
    this.repositionImages();
    this.wait(500);
    this.afterWaitOp =3;
}
},
onRightArrow: function(event)
{
    if ((this.isActive)&&(this.afterWaitOp==0))
    {
        this.galleryObject.remove(this.frames[0]);
        this.CurrentImageNumber++;
        var middle =this.CurrentImageNumber;
        while(middle<0)
            middle+=this.imagesCount;
        middle = middle%this.imagesCount;
        for (var xx=0; xx<this.viewedImages-1;xx++)
        {

```

Figure A.7(page 11 of 15): GalleryObject class code

```

if (xx== (this.middleImage))
{
    var rot = this.frames[xx+1].rotation.z;
    this.galleryObject.remove(this.frames[xx+1]);
    this.frames[xx] = new THREE.Mesh(this.igs[middle],new
THREE.MeshFaceMaterial() );
    this.frames[xx].position.x = this.frames[xx+1].position.x;
    this.frames[xx].position.y = this.frames[xx+1].position.y;
    this.frames[xx].position.z = this.frames[xx+1].position.z-3;
    this.frames[xx].scale.x = 0.75;
        this.frames[xx].scale.y = 0.75;
        this.frames[xx].scale.z = 1;
        this.frames[xx].rotation.z = rot;
    this.galleryObject.add(this.frames[xx]);
}
else
{
    this.frames[xx] = this.frames[xx+1];
}
}
var imgNumber =(this.CurrentImageNumber+this.middleImage);
while(imgNumber<0)
    imgNumber+=this.imagesCount;
imgNumber = imgNumber%this.imagesCount;
this.frames[this.viewedImages-1] = new
THREE.Mesh(this.igs[imgNumber],new THREE.MeshFaceMaterial() );
    this.frames[this.viewedImages-1].position.x =
this.frames[this.viewedImages-2].position.x;
    this.frames[this.viewedImages-1].position.y =
this.frames[this.viewedImages-2].position.y;
    this.frames[this.viewedImages-1].position.z =
this.frames[this.viewedImages-2].position.z-3;
    this.frames[this.viewedImages-1].scale.x =
1/(1+this.middleImage/1.5);
    this.frames[this.viewedImages-1].scale.y =
1/(1+this.middleImage/1.5);
    this.frames[this.viewedImages-1].scale.z = 1;
    this.frames[this.viewedImages-1].rotation.z =
this.frames[this.viewedImages-2].rotation.z;
    this.galleryObject.add(this.frames[this.viewedImages-1]);
    this.repositionImages();
    this.wait(500);
    this.afterWaitOp =3;
}
},
repositionButtons: function()
{
    this.prevButton.position.x =
this.frames[this.middleImage].position.x-this.b_w;

```

Figure A.7(page 12 of 15): GalleryObject class code

```

this.prevButton.position.y =
this.frames[this.middleImage].position.y+this.activeSizeP.y/2;
    this.prevButton.position.z =
this.frames[this.middleImage].position.z+4;

    this.nextButton.position.x =
this.frames[this.middleImage].position.x+this.b_w;
    this.nextButton.position.y =
this.frames[this.middleImage].position.y+this.activeSizeP.y/2;
    this.nextButton.position.z =
this.frames[this.middleImage].position.z+4;
    },
    wait: function(msec)
    {
        this.timeReach=new Date().getTime()+msec;
    },
    afterWait: function()
    {
        this.isActive = true;
        if (this.afterWaitOp ==2)
        {
            for (var x=0;x<this.buttonArray.length;x++)
            {
                this.galleryObject.add( this.buttonArray[x] );
            }
            this.repositionButtons();
        }
        else if (this.afterWaitOp ==1)
        {
            this.repositionButtons();
        }
    },
    render:function()
    {
        if (this.afterWaitOp!=0)
        {
            {
                var t =new Date().getTime();
                if (t>=this.timeReach)
                {
                    this.afterWait();
                    this.afterWaitOp=0;
                }
            }
        }
    },
    restore:function( )
    {
        this.isActive = false;
        wc.removeActiveObject(this);
        for (var x=0;x<this.buttonArray.length;x++)

```

Figure A.7(page 13 of 15): GalleryObject class code

```

{
    this.galleryObject.remove( this.buttonArray[x] );
}

for (var x=0; x<this.viewedImages;x++)
{
    if (!(x>(this.middleImage-2))&&(x<(this.middleImage+2)))
        this.galleryObject.remove( this.frames[x] );
}
for (var x=-1;x<3;x++)
{
    this.frames[x+1] = this.frames[this.middleImage+x] ;
}

for (var xx=0; xx<3;xx++)
{
    new
TWEEN.Tween(this.frames[xx].scale).to({x:this.inactiveSizeP.x/this.activeSi
zeP.x,y:this.inactiveSizeP.y/this.activeSizeP.y,z:1},500).easing(TWEEN.Easi
ng.Sinusoidal.EaseIn).start();
    var r=0,pos=6;
    if (xx==0)
    {
        r=-1;
        pos = 0;
    }
    new TWEEN.Tween(this.frames[xx].rotation).to({x:0,y:0,z:r/8},
500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
    new TWEEN.Tween(this.frames[xx].position).to({x:0,y:0,z:pos}
,500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
    new TWEEN.Tween(this.galleryObject.position).to(
{x:this.inactivePositionP.x,y:this.inactivePositionP.y,z:this.inactivePosit
ionP.z},500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
    new TWEEN.Tween(this.galleryObject.rotation).to(
{x:this.rotation.x,y:this.rotation.y,z:this.rotation.z},500
).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
}
    this.isActive = false;
    this.mousedown=false;
},
getDimentions:function()
{
    return this.inactiveSizeP;
},
getInactivePosition:function()
{
    return this.inactivePositionP;
},

```

Figure A.7(page 14 of 15): GalleryObject class code

```
    getCurrentInactivePosition:function ()
    {
        return this.galleryObject.position;
    },
    getRotation:function ()
    {
        return this.rotation;
    },
    getSide:function ()
    {
        return this.inactiveSide;
    },
    is_Active:function ()
    {
        return this.isActive;
    },
    changePosition:function (inactivePositionVector)
    {
        this.galleryObject.position= inactivePositionVector;
    }
};
```

Figure A.7(page 15 of 15): GalleryObject class code


```

CalendarGeometry = function ( dim, borderImage,BGimage,titleFont,
    titleColor,daysNameColor,daysNumberColor,nextMonthDaysNumberColor,linesC
olor ,selectedDayColor,year,month) {
    THREE.Geometry.call( this );
    this.monthNames=["january","February","March","April","May","June","July
","August","September","October","November","December"];
    this.weekDays = ["Sun","Mon","Tue","Wed","Thu","Fri","Sat"];
    this.maxDays = [31,28,31,30,31,30,31,31,30,31,30,31];
    this.edge_width = dim.x/30;
    this.total_width = dim.x;
    this.total_height = dim.y;
    this.center_width = dim.x-2*this.edge_width;
    this.linesColor = linesColor;
    this.selectedDayColor = selectedDayColor;
    this.BGimage = BGimage;
    var scope = this,
        width_half = dim.x / 2,
        height_half = dim.y / 2,
        depth_half = dim.z / 2;
    var borderMat=new THREE.MeshBasicMaterial(
{map:THREE.ImageUtils.loadTexture(borderImage),opacity:1,transparent:true}
);
    buildPlane( 'x', 'y', 1, - 1, dim.x, dim.y, depth_half,
borderMat,0,0);
    var canv = this.getMonthCanvas(year,month);
    this.monthMaterial = new THREE.MeshBasicMaterial({map: new
THREE.Texture(canv), opacity:1, transparent:false});
    this.monthMaterial.map.needsUpdate = true;
    buildPlane( 'x', 'y', 1, - 1, canv.width, canv.height,
depth_half+3, this.monthMaterial, 0,0);
    function buildPlane( u, v, udir, vdir, width, height, depth, material
,shiftx,shifty) {
        var w, ix, iy,
            gridX = 1,
            gridY = 1,
            width_half = width / 2,
            height_half = height / 2,
            offset = scope.vertices.length;
        w = 'z';
        var gridX1 = gridX + 1,
            gridY1 = gridY + 1,
            segment_width = width / gridX,
            segment_height = height / gridY;
        for( iy = 0; iy < gridY1; iy++ ) {
            for( ix = 0; ix < gridX1; ix++ ) {
                var vector = new THREE.Vector3();
                vector[ u ] = ( ix * segment_width - width_half ) *
udir+shiftx;
                vector[ v ] = ( iy * segment_height-height_half) * vdir-shifty;
            }
        }
    }
}

```

Figure A.8(page 1 of 13): CalendarObject class code

```

        vector[ w ] = depth;
        scope.vertices.push( new THREE.Vertex( vector ) );
    }
}
for( iy = 0; iy < gridY; iy++ ) {
    for( ix = 0; ix < gridX; ix++ ) {
        var a = ix + gridX1 * iy;
        var b = ix + gridX1 * ( iy + 1 );
        var c = ( ix + 1 ) + gridX1 * ( iy + 1 );
        var d = ( ix + 1 ) + gridX1 * iy;
        scope.faces.push( new THREE.Face4( a + offset, b +
offset, c + offset, d + offset, null, null, material ) );
        scope.faceVertexUvs[ 0 ].push( [
            new THREE.UV( ix / gridX, iy / gridY ),
            new THREE.UV( ix / gridX, ( iy + 1 ) / gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, ( iy + 1 ) / gridY ),
            new THREE.UV( ( ix + 1 ) / gridX, iy / gridY )
        ] );
    }
}
}
this.computeCentroids();
this.computeFaceNormals();
};
getMonthCanvas=function(year,month)
{
    var holder = document.createElement( 'canvas' );
    var ctext = holder.getContext('2d');
    holder.width = this.center_width;
    holder.height = this.center_height;
    var line_height = holder.height/9;
    ctext.drawImage(this.BGimage, 0,0,holder.width, holder.height);
    var font_Size = line_height*0.9;
    ctext.font = font_Size+"px "+this.titleFont;//"Arial";
    ctext.textAlign = "center";
    ctext.fillStyle = this.titleColor.getContextStyle();
    ctext.fillText( this.monthNames[month]+" "+year,
holder.width/2,line_height*1.4);
    var i;
    var x = this.edge_width*2;
    ctext.strokeStyle = this.linesColor.getContextStyle();
    ctext.stroke();
    font_Size = line_height*0.6;
    ctext.font = font_Size+"px "+this.titleFont;//"Arial";
    ctext.textAlign = "center";
    ctext.fillStyle = this.daysNameColor.getContextStyle();
    var gap = holder.width/8;
    for ( i=0; i<7; i++)

```

Figure A.8(page 2 of 13): CalendarObject class code

```

{
    ctext.fillText(this.weekDays[i], i*gap+gap,line_height*2.5);
}
ctext.fillStyle = this.daysNumberColor.getContextStyle();
var days=this.getMonthDays(year,month);
i=0;
var flag=1;
tempDate = new Date();
for (var row = 1; row<7; row++)
{
    for (var column = 0; column<7; column++)
    {
        if
((flag==1) && (days[i]==tempDate.getDate()) && (year==tempDate.getFullYear()) &&
((month)==tempDate.getMonth()))
        {
            ctext.fillStyle =
this.selectedDayColor.getContextStyle();
ctext.fillRect(column*gap+gap/2,line_height*(row+1.8)+2,gap,line_height-4);
            ctext.fillStyle = this.daysNumberColor.getContextStyle();
        }
        if (days[i]!=-1)
        {
            ctext.fillText(days[i],
column*gap+gap,line_height*(row+2.5));
        }
        i++;
    }
}
return holder;
};
getMonthDays=function (year,month)
{
    var days = new Array();
    var maxDay;
    var tempDate = new Date(year ,month,1);
    var weekday= tempDate.getDay();
    if ((month==1) && (year%4==0))
    {
        maxDay = 29;
    }
    else
    {
        maxDay = this.maxDays[month];
    }
    for (var i=0; i< weekday; i++)
    {
        days[i] = -1;
    }
    for (var i=1; i<= maxDay; i++)

```

Figure A.8(page 3 of 13): CalendarObject class code

```

{
    days[weekday+i-1] = i;
}
for (var i=1; i<= (42-weekday-maxDay); i++)
{
    days[i+weekday+maxDay-1] = i;
}
return days;
};
CalendarGeometry.prototype = new THREE.Geometry();
CalendarGeometry.prototype.constructor = CalendarGeometry;
CalendarGeometry.prototype.getMonthCanvas = getMonthCanvas;
CalendarGeometry.prototype.getMonthDays = getMonthDays;
CalendarObject = function ( skin,side,inactiveSizeVector3
,inactivePositionVector2, inactiveMovable,activeSizeVector3,
activePositionVector3,activeActionButtonArray,rotate) {
    this.skin= skin;
    this.inactiveSide          = side;
    this.inactiveSize          = inactiveSizeVector3;
    this.inactivePosition      = inactivePositionVector2;
    this.inactiveMovable       = inactiveMovable;
    this.activeSize            = activeSizeVector3;
    this.activePosition        = activePositionVector3;
    this.activeActionButtons   = activeActionButtonArray;
    this.isActive = false;
    this.inactiveSizeP = null;
    this.activeSizeP = null;
    this.inactivePositionP=null;
    this.activePositionP=null;
    this.lastActivePositionP=null;
    this.intersectPointMouseDown=null;
    this.mousedown = false;
    this.offset=new THREE.Vector3();
    this.rotation=new THREE.Vector3(0,0,0);
    this.imgNumber=side.getObjectID();
    this.buttonArray=new Array();
    this.timeReach=0;
    this.afterWaitOp = 0;
    var xmlhttp;
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.open("GET",this.skin+"skin.xml",false)
    xmlhttp.send();
    xmlDoc=xmlhttp.responseXML;

```

Figure A.8(page 4 of 13): CalendarObject class code

```

x=xmlDoc.getElementsByTagName ("calendar");
this.titleColor =
(x[0].getElementsByTagName ("titleColor") [0].childNodes [0].nodeValue);
this.titleColor = new THREE.Color (this.titleColor);
this.daysNameColor =
(x[0].getElementsByTagName ("daysNameColor") [0].childNodes [0].nodeValue);
this.daysNameColor = new THREE.Color (this.daysNameColor);
this.daysNumberColor =
(x[0].getElementsByTagName ("daysNumberColor") [0].childNodes [0].nodeValue);
this.daysNumberColor = new THREE.Color (this.daysNumberColor);
this.nextMonthDaysNumberColor =
(x[0].getElementsByTagName ("nextMonthDaysNumberColor") [0].childNodes [0].nodeValue);
this.nextMonthDaysNumberColor = new
THREE.Color (this.nextMonthDaysNumberColor);
this.linesColor =
(x[0].getElementsByTagName ("linesColor") [0].childNodes [0].nodeValue);
this.linesColor = new THREE.Color (this.linesColor);
this.selectedDayColor =
(x[0].getElementsByTagName ("selectedDayColor") [0].childNodes [0].nodeValue);
this.selectedDayColor = new THREE.Color (this.selectedDayColor);
this.BGimg=(x[0].getElementsByTagName ("BGimg") [0].childNodes [0].nodeValue);
this.borderImg=(x[0].getElementsByTagName ("borderImg") [0].childNodes [0].nodeValue);

this.inactiveSizeP = new THREE.Vector3 (this.inactiveSize.x*side.width,
this.inactiveSize.y*side.height, this.inactiveSize.z);
this.activeSizeP = new THREE.Vector3 (this.activeSize.x*screen_w,
this.activeSize.y*screen_h, this.activeSize.z);
this.activePositionP = new THREE.Vector3 (screen_w
*this.activePosition.x+this.activeSizeP.x/2-screen_w/2,
screen_h/2- (screen_h*this.activePosition.y+this.activeSizeP.y/2), 5 );
this.lastActivePositionP = new THREE.Vector3 ();
this.lastActivePositionP.x =this.activePositionP.x;
this.lastActivePositionP.y =this.activePositionP.y;
this.lastActivePositionP.z =-1;
this.BGImage = new Image ();
this.BGImage.src = this.skin+this.BGimg;
var parent = this;
this.BGImage.onload = function () {
parent.updateGeometry ();
};
this.cg = new CalendarGeometry (this.activeSizeP,
this.skin+this.borderImg, this.BGImage, "Arial",
this.titleColor, this.daysNameColor, this.daysNumberColor, this.nextMonthDaysNumberColor,
this.linesColor
, this.selectedDayColor, this.year, this.month);
this.frame = new THREE.Mesh (this.cg, new THREE.MeshFaceMaterial () );
new TWEEN.Tween ( this.frame.scale ).to (
{

```

Figure A.8(page 5 of 13): CalendarObject class code

```

x:this.inactiveSizeP.x/this.activeSizeP.x,
y:this.inactiveSizeP.y/this.activeSizeP.y,
z:1}, 0 ).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
    if (this.inactiveSide.id==0)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),3*this.imgNumber-screen_d);
            this.rotation.z = rotate*Math.PI/180;
        }
    else if (this.inactiveSide.id==1)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x +3*this.imgNumber,
            this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),
            this.inactiveSide.topCorner_z-this.inactiveSizeP.x/2-
(this.inactivePosition.x*side.width) );
            this.rotation.y = Math.PI/2;
            this.rotation.x = rotate*Math.PI/180;
        }
    else if (this.inactiveSide.id==2)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x -3*this.imgNumber,
            this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),this.inactiveSide.topCorner_z+
this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width) );
            this.rotation.y = Math.PI/-2;
            this.rotation.x = rotate*Math.PI/180;
        }
    else if (this.inactiveSide.id==3)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
            this.inactiveSide.topCorner_y -3*this.imgNumber,
            this.inactiveSide.topCorner_z -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height) );
            this.rotation.x = Math.PI/2;
            this.rotation.z = rotate*Math.PI/180;
        }
    else if (this.inactiveSide.id==4)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x +this.inactiveSizeP.x/2+
(this.inactivePosition.x*side.width), this.inactiveSide.topCorner_y +
3*this.imgNumber, this.inactiveSide.topCorner_z
+this.inactiveSizeP.y/2+(this.inactivePosition.y*side.height) );
    }

```

Figure A.8(page 6 of 13): CalendarObject class code

```

        this.rotation.x = Math.PI/-2;
        this.rotation.z = rotate*Math.PI/180;
    }
    this.frame.rotation.x = this.rotation.x;
    this.frame.rotation.y = this.rotation.y;
    this.frame.rotation.z = this.rotation.z;
    this.frame.position = new THREE.Vector3(
this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z)
;
    scene.add( this.frame );
    this.restoreButtonEnabled=false;
    this.restoreSizeButtonEnabled=false;
    var ii=0;
    this.b_w=screen_w/35;
    this.b_h=screen_w/35;
    var temp =
(x[0].getElementsByTagName("restoreButton")[0].childNodes[0].nodeValue);
        this.restoreButton = imgButton(skin+temp, this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
this.lastActivePositionP.y+this.activeSizeP.y/2,this.lastActivePositionP.z+
4,0.8);
        this.restoreButtonEnabled=true;
        this.buttonArray[ii++] = this.restoreButton;
        if ((activeActionButtonArray&2 ) ==2)
        {
            var temp =
(x[0].getElementsByTagName("restoreSizeButton")[0].childNodes[0].nodeValue)
;
            this.restoreSizeButton = imgButton(skin+temp,this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
this.lastActivePositionP.y+this.activeSizeP.y/2-
ii*this.b_h,this.lastActivePositionP.z+4,0.8);
            this.restoreSizeButtonEnabled=true;
            this.buttonArray[ii++] = this.restoreSizeButton;
        }
        this.selectedButton = -1;
        this.np_b_h=this.activeSizeP.x/10;
        temp =
(x[0].getElementsByTagName("prevButton")[0].childNodes[0].nodeValue);
        this.prevButton = imgButton(skin+temp,this.np_b_w,
this.np_b_h,0,0,this.lastActivePositionP.z+4,0.8);
        this.buttonArray[ii++] = this.prevButton;
        temp =
(x[0].getElementsByTagName("nextButton")[0].childNodes[0].nodeValue);
        this.nextButton = imgButton(skin+temp,this.np_b_w,
this.np_b_h,0,0,this.lastActivePositionP.z+4,0.8);
        this.buttonArray[ii++] = this.nextButton;
        return this;
};
CalendarObject.prototype = {

```

Figure A.8(page 7 of 13): CalendarObject class code

```

constructor: CalendarObject,
checkIntersection: function ( object ) {
    if (this.isActive )
    {
for (var ii=0; ii<this.buttonArray.length; ii++)
    {
        if (this.buttonArray[ii] == object)
        {
            this.selectedButton = ii;
            return true;
        }
    }
    }
else
    {
        if ( this.frame == object)
            return true;
    }
    return false;
},
onMouseDown: function ( event,ray ) {
    this.mousedown = true;
    if (this.isActive ==true)
    {
        this.mousePositionOnMouseDownActive=new
THREE.Vector2(event.clientX,event.clientY);
    }
},
onMouseUp: function ( event ) {
    this.mousedown=false;
},
onClick: function ( event ) {
    if (this.isActive)
    {
        if (this.selectedButton>=0)
        {
            if (this.buttonArray[this.selectedButton] ==
this.restoreButton)
            {
                this.restore();
            }
        }
        else if (this.buttonArray[this.selectedButton] ==
this.restoreSizeButton)
        {
            new TWEEN.Tween( this.frame.position ).to(
this.lastActivePositionP,200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start
();

```

Figure A.8(page 8 of 13): CalendarObject class code


```

for (var i = 0; i<this.buttonArray.length;i++)
    new TWEEN.Tween( this.buttonArray[i].position ).to(
{x:this.buttonArray[i].position.x,y:this.buttonArray[i].position.y,z:4},
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        this.wait(210);
        this.afterWaitOp = 1;
    }
    else if (this.buttonArray[this.selectedButton] == this.prevButton)
    {
        if (this.afterWaitOp == 0)
        {
            if (this.month==0)
            {
                this.month = 11;
                this.year-=1;
            }
            else
                this.month-=1;
            this.newMonthCanvas = this.cg.getMonthCanvas(this.year,this.month);
            var mat = new THREE.MeshBasicMaterial({map: new
THREE.Texture(this.newMonthCanvas), opacity:1, transparent:false});
            mat.map.needsUpdate = true;
            this.tempPlane = new THREE.Mesh(new
THREE.PlaneGeometry(this.newMonthCanvas.width,this.newMonthCanvas.height,8,
8),mat);
            this.tempPlane.position.x = this.frame.position.x-
this.newMonthCanvas.width/2;
            this.tempPlane.position.y = this.frame.position.y;
            this.tempPlane.position.z = this.frame.position.z+5;
            new TWEEN.Tween( this.tempPlane.scale ).to( { x:0.001, y:1,z:1}, 0
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            scene.add( this.tempPlane );
            this.afterWaitOp = 3;
            this.wait(10);
        }
    }
    else if (this.buttonArray[this.selectedButton] == this.nextButton)
    {
        if (this.afterWaitOp == 0)
        {
            if (this.month==11)
            {
                this.month = 0;
                this.year+=1;
            }
            else
                this.month+=1;
            this.newMonthCanvas = this.cg.getMonthCanvas(this.year,this.month);

```

Figure A.8(page 9of 13): CalendarObject class code

```

    this.tempPlane.position.x =
this.frame.position.x+this.newMonthCanvas.width/2;
    this.tempPlane.position.y = this.frame.position.y;
    this.tempPlane.position.z = this.frame.position.z+5;
    new TWEEN.Tween( this.tempPlane.scale ).to( { x:0.001, y:1,z:1}, 0
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
    scene.add( this.tempPlane );
    this.afterWaitOp = 3;
    this.wait(10);
    }
    }
    }
}
else
{
    this.lastActivePositionP.z = wc.addActiveObject(this);

    new TWEEN.Tween( this.frame.scale ).to( { x:1, y:1, z:1}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
    new TWEEN.Tween( this.frame.rotation ).to( {x:0, y: 0, z:0},
500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
    this.wait(500);
    this.afterWaitOp =2;
    }
},
onMouseMove: function ( event,ray ) {
    if ((this.mousedown==true)&&(this.selectedButton==-1))
    {
        if (this.isActive)
        {
            this.repositionButtons();
        }
    }
},
onMouseWheel: function ( event ) {
    if (this.isActive)
    {
        // WebKit
        var pos=new
THREE.Vector3(this.frame.position.x,this.frame.position.y,this.frame.positi
on.z);
        if ( event.wheelDeltaY ) {
            pos.z+= event.wheelDeltaY ;
            // Opera / Explorer 9
        } else if ( event.wheelDelta ) {
            pos.z += event.wheelDelta ;
            // Firefox
        } else if ( event.detail ) {
            pos.z -= event.detail *20;
        }
    }
}

```

Figure A.8(page 10 of 13): CalendarObject class code

```

if (pos.z>screen_h)
    pos.z = screen_h;
else if (pos.z<(screen_d/-1+50))
    pos.z = screen_d/-1+50;
    this.frame.position= pos;
    this.wait(1);
    this.afterWaitOp = 1;
}
},
updateGeometry: function()
{
    this.newMonthCanvas = this.cg.getMonthCanvas(this.year,this.month);
    var mat = new THREE.MeshBasicMaterial({map: new
THREE.Texture(this.newMonthCanvas), opacity:1, transparent:false});
    mat.map.needsUpdate = true;
    this.cg = new CalendarGeometry(this.activeSizeP,
this.skin+this.borderImg,this.BGImage,"Arial",
    this.titleColor,this.daysNameColor,this.daysNumberColor,this.nextMonthDa
ysNumberColor,
                                this.linesColor
,this.selectedDayColor,this.year,this.month);
    var position = new
THREE.Vector3(this.frame.position.x,this.frame.position.y,this.frame.positi
on.z);
    scene.remove(this.frame);
    this.frame = new THREE.Mesh(this.cg,new THREE.MeshFaceMaterial() );
    this.frame.position = position;
    this.frame.rotation.x = this.rotation.x;
    this.frame.rotation.y = this.rotation.y;
    this.frame.rotation.z = this.rotation.z;
    scene.add(this.frame);
},
repositionButtons: function()
{
    var ii=0;
    for (ii=0; ii<this.buttonArray.length; ii++)
    {
        this.buttonArray[ii].position.x =
this.frame.position.x+this.activeSizeP.x/2;
        this.buttonArray[ii].position.y =
this.frame.position.y+this.activeSizeP.y/2-ii*this.b_h;
        this.buttonArray[ii].position.z = this.frame.position.z+5;
    }
    ii=this.buttonArray.length-1;
    this.buttonArray[ii].position.x =
this.frame.position.x+this.activeSizeP.x/2.7;
    this.buttonArray[ii].position.y =
this.frame.position.y+this.activeSizeP.y/2.85;
    this.buttonArray[ii].position.z = this.frame.position.z+5;
}
}

```

Figure A.8(page 11 of 13): CalendarObject class code

```

this.buttonArray[iii].position.z = this.frame.position.z+5;
},
wait: function(msec)
{
    this.timeReach=new Date().getTime()+msec;
},
afterWait: function()
{
    this.isActive = true;
    if (this.afterWaitOp ==2)
    {
        for (var x=0;x<this.buttonArray.length;x++)
        {
            scene.add( this.buttonArray[x] );
        }
        this.repositionButtons();
        this.afterWaitOp=0;
    }
    else if (this.afterWaitOp ==1)
    {
        this.repositionButtons();
        this.afterWaitOp=0;
    }
    else if (this.afterWaitOp ==3)
    {
        new TWEEN.Tween( this.tempPlane.position ).to(
{x:this.frame.position.x,y:this.frame.position.y,z:this.frame.position.z+4}
, 500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        new TWEEN.Tween( this.tempPlane.scale ).to( { x:1, y:1,z:1},
500 ).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        this.wait(500);
        this.afterWaitOp = 4;
    }
    else if (this.afterWaitOp ==4)
    {
        this.cg = new CalendarGeometry(this.activeSizeP,
this.skin+this.borderImg,this.BGImage,"Arial",
        this.titleColor,this.daysNameColor,this.daysNumberColor,this.nextMonthDa
ysNumberColor,this.linesColor ,this.selectedDayColor,this.year,this.month);
        scene.remove(this.frame);
        this.frame = new THREE.Mesh(this.cg,new THREE.MeshFaceMaterial() );
        this.afterWaitOp=0;
    }
},
render:function()
{
    if (this.afterWaitOp!=0)
    {
        var t =new Date().getTime();

```

Figure A.8(page 12 of 13): CalendarObject class code

```

        if (t>=this.timeReach)
        {
            this.afterWait();
        }
    },
    restore:function( )
    {
        this.isActive = false;
        for (var x=0;x<this.buttonArray.length;x++)
        {
            scene.remove( this.buttonArray[x] );
        }
        wc.removeActiveObject(this);
        new TWEEN.Tween( this.frame.position ).to(
this.inactivePositionP,500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        new TWEEN.Tween( this.frame.scale ).to( {
x:this.inactiveSizeP.x/this.activeSizeP.x,
y:this.inactiveSizeP.y/this.activeSizeP.y, z:1}, 500
).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        new TWEEN.Tween( this.frame.rotation ).to( this.rotation, 500).
easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        this.isActive = false;
        this.mousedown=false;
    },
    getDimentions:function()
    {
        return this.inactiveSizeP;
    },
    getInactivePosition:function()
    {
        return this.inactivePositionP;
    },
    getCurrentInactivePosition:function()
    {
        return this.frame.position;
    },
    getRotation:function()
    {
        return this.rotation;
    },
    is_Active:function()
    {
        return this.isActive;
    },
    changePosition:function(inactivePositionVector)
    {
        this.frame.position = inactivePositionVector;
    }
};

```

Figure A.8(page 13 of 13): CalendarObject class code

```

MarqueeElement = function (canv, ctext, destination, opacity, xpos, ypos, zpos) {
  this.canv          = canv;
  this.width         = 500;
  this.height        = canv.height;
  this.ctext         = ctext;
  this.destination   = destination;
  this.tempCanv      = document.createElement( 'canvas' );
  this.tempctx       = this.tempCanv.getContext('2d');
  this.tempCanv.width = 1;
  this.tempCanv.height = canv.height;
  this.plane = new THREE.Mesh(new THREE.PlaneGeometry(
this.width, canv.height, 1, 1), new THREE.MeshBasicMaterial({map: new
THREE.Texture(this.tempCanv), opacity:opacity,
needsUpdate:true, transparent:true})) );
  this.plane.position.x = xpos;
  this.plane.position.y = ypos;
  this.plane.position.z = zpos;
  this.plane.scale.x = 0.0001;
  this.imageData = null;
  this.appliedCanvas = this.tempCanv;
  return this;
};
MarqueeElement.prototype = {
  constructor: MarqueeElement,
  changeWidth: function (width, index)
  {
    if ((this.canv.width-index) > width)
    {
      this.width = width;
      this.tempCanv.width = width;
      this.imageData =
this.ctext.getImageData(index, 0, width, this.canv.height);
      this.tempctx = this.tempCanv.getContext('2d');
      this.tempctx.putImageData(this.imageData, 0, 0);
      this.plane.scale.x = width/500;
      this.plane.materials[0].map.needsUpdate = true;
    }
    else if ( (this.canv.width-index) > 0 )
    {
      this.width = this.canv.width - index;
      this.plane.scale.x = this.width/500;
      this.plane.materials[0].map.needsUpdate = true;
    }
    else
    {
      this.width = 0;
      this.plane.scale.x = 0.0001;
    }
    return this.width;
  },
};

```

Figure A.9(page 1 of 8): Marquee class code

```

    changePosition: function (xpos,ypos)
    {
        this.plane.position.x = xpos;
        this.plane.position.y = ypos;
    }
};

Marquee= function ( side,BGImg,BGcolor,marqueeXML,titleColor,spacerColor,
titleFont, titleFontSize,underlined,bold,italic,opacity,sizeVector3,
positionVector2,movable,rotate,speed) {
    this.marqueeXML = marqueeXML;
    this.inactiveSide = side;
    this.BGImgSrc = BGImg;
    this.title = "ss";
    this.titleFont = titleFont;
    this.titleFontSize = titleFontSize;
    this.opacity = opacity;
    this.size = sizeVector3;
    this.position = positionVector2;
    this.inactiveMovable = movable;
    this.underlined = underlined;
    this.bold= bold;
    this.italic = italic;
    this.mouseXOnMouseDown=0;
    this.mouseYOnMouseDown=0;
    this.mouseZOnMouseDown=0;
    if (speed<1)
        this.timeSlot = 150;
    else if (speed >10)
        this.timeSlot = 15;
    else
        this.timeSlot = 150/speed;
    this.mousedown = false;
    this.offset=new THREE.Vector3();
    this.rotation=new THREE.Vector3(0,0,0);
    this.BGColor = null;
    if ((BGcolor !=null)&&(BGcolor!=""))
        this.BGColor = new THREE.Color(BGcolor);
    this.titleColor = new THREE.Color(titleColor);
    this.spacerColor = new THREE.Color(spacerColor);
    this.marqueeElements = new Array();
    this.sizeP = new THREE.Vector3(this.size.x*side.width,
this.size.y*side.height, this.size.z);
    this.stop =false;
    if (this.inactiveSide.id==0)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x +this.sizeP.x/2+(this.position.x*
side.width),this.inactiveSide.topCorner_y -this.sizeP.y/2-
(this.position.y*side.height), 2-screen_d );
    }

```

Figure A.9(page 2 of 8): Marquee class code

```

        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==1)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x +2, this.inactiveSide.topCorner_y -
            this.sizeP.y/2-(this.position.y*side.height),this.inactiveSide.topCorner_z-
            this.sizeP.x/2-(this.position.x*side.width) );
        this.rotation.y = Math.PI/2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==2)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x -2, this.inactiveSide.topCorner_y -
            this.sizeP.y/2-(this.position.y*side.height),
            this.inactiveSide.topCorner_z+this.sizeP.x/2+(this.position.x*side.width)
        );
        this.rotation.y = Math.PI/-2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==3)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
            this.sizeP.x/2+(this.position.x*side.width),this.inactiveSide.topCorner_y -
            2,
            this.inactiveSide.topCorner_z -this.sizeP.y/2-
            (this.position.y*side.height) );
        this.rotation.x = Math.PI/2;
        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==4)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
            +this.sizeP.x/2+(this.position.x*side.width),
            this.inactiveSide.topCorner_y +2,
            this.inactiveSide.topCorner_z
            +this.sizeP.y/2+(this.position.y*side.height) );
        this.rotation.x = Math.PI/-2;
        this.rotation.z = rotate*Math.PI/180;
    }
    this.marqueeObject = new THREE.Object3D();
    this.createCanvas();
    this.marqueeObject.position = new THREE.Vector3(
    this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z)
    ;    return this;
};

```

Figure A.9(page 3 of 8): Marquee class code


```

Marquee.prototype = {
  constructor: Marquee,
  createCanvas: function()
  {
    this.mainCanvas = document.createElement( 'canvas' );
    this.mainCanvasContext = this.mainCanvas.getContext('2d');
    this.mainCanvas.width = this.sizeP.x;
    this.mainCanvas.height = this.sizeP.y;
    this.title_font_Size = this.sizeP.y*this.titleFontSize/2;
    var a = "";
    if (this.italic)
      a += "italic ";
    if (this.bold)
      a += "bold ";
    this.mainCanvasContext.font = a+this.title_font_Size+"px
    "+this.titleFont;
    this.mainCanvasContext.textAlign = "left";
    if ((this.BGImgSrc != null)&&(this.BGImgSrc!=""))
    {
      this.frame = new THREE.Mesh(new THREE.PlaneGeometry(
this.mainCanvas.width+20,this.mainCanvas.height, 1, 1),
      new THREE.MeshBasicMaterial({map:
THREE.ImageUtils.loadTexture(this.BGImgSrc), opacity:this.opacity,
needsUpdate:true,transparent:true}) );
    }
    else if (this.BGColor != null)
    {
this.material.map.needsUpdate = true;
this.frame = new THREE.Mesh(new THREE.PlaneGeometry(
this.mainCanvas.width+20,this.mainCanvas.height, 1, 1),this.material );
    }
    else
    {
this.frame = new THREE.Mesh(new THREE.PlaneGeometry(
this.mainCanvas.width+20,this.mainCanvas.height, 1, 1),null );
    }
    this.marqueeObject.add(this.frame);
    this.totalWidth = 0 ;
    var metrics,spacerMetrics;
    var canv,spacerCanv;
    var xmlhttp;
    ////////////////////////////////////
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
      xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
      xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
  }
}

```

Figure A.9(page 4 of 8): Marquee class code

```

xmlhttp.open("GET",this.marqueeXML,false)
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;
this.spacer =
xmlDoc.getElementsByTagName("spacer")[0].childNodes[0].nodeValue;
var x =xmlDoc.getElementsByTagName("marqueeElement");
this.length = x.length*2-1;
var title, destination;
for (var xx=0; xx<this.length; xx+=2)
{
    title =
x[xx/2].getElementsByTagName("title")[0].childNodes[0].nodeValue;
    destination =
x[xx/2].getElementsByTagName("URL")[0].childNodes[0].nodeValue;
    canv = document.createElement( 'canvas' );
    ctext = canv.getContext('2d');
    metrics = this.mainCanvasContext.measureText(title);
    canv.width = metrics.width+((xx==0)?this.sizeP.x:0);
    canv.height = this.sizeP.y;
    this.totalWidth += canv.width ;
    ctext.font = this.mainCanvasContext.font;
    ctext.textAlign = this.mainCanvasContext.textAlign;
    ctext.fillStyle = this.titleColor.getContextStyle();
    ctext.fillText(title , (xx==0)?this.sizeP.x:0,canv.height*0.6);
    if (this.underlined)
    {
        ctext.moveTo((xx==0)?this.sizeP.x:0,canv.height*0.65);
        ctext.lineTo(((xx==0)?this.sizeP.x:0)+
canv.width,canv.height*0.65);
        ctext.lineWidth = this.title_font_Size/20;
        ctext.strokeStyle = this.titleColor.getContextStyle();
        ctext.stroke();
    }
    this.marqueeElements[xx]= new MarqueeElement( canv, ctext,
destination,this.opacity,this.mainCanvas.width/2,0,3);
    this.marqueeObject.add(this.marqueeElements[xx].plane);
    if (xx != this.length-1)
    {
        spacerCanv = document.createElement( 'canvas' );
        spacerCtext.font = this.mainCanvasContext.font;
        spacerCtext.textAlign = this.mainCanvasContext.textAlign;
        spacerCtext.fillStyle =
this.spacerColor.getContextStyle();
        spacerCtext.fillText(this.spacer , 0,canv.height*0.6);
        this.marqueeElements[xx+1]= new MarqueeElement (
spacerCanv, spacerCtext, "",this.opacity,this.mainCanvas.width/2,0,3);
        this.marqueeObject.add(this.marqueeElements[xx+1].plane);
    }
}
this.index = 0;

```

Figure A.9(page 5 of 8): Marquee class code

```

        this.selectedElement = null;
this.prevTime = new Date().getTime();
    },
    checkIntersection: function ( object ) {
        if ( this.frame == object)
        {
            document.body.style.cursor = 'pointer';
            this.stop=true;
            return true;
        }
        for (var xx=0; xx<this.length; xx++)
        {
            if (object == this.marqueeElements[xx].plane)
            {
                document.body.style.cursor = 'pointer';
                this.selectedElement = this.marqueeElements[xx];
                this.stop=true;
                return true;
            }
        }
        this.stop=false;
        return false;
    },
    onMouseDown: function ( event,ray ) {
        this.mousedown = true;
    },
    onMouseClick: function ( event ) {
        this.selectedElement.destination;
        if ((this.selectedElement.destination!=null)&&
(this.selectedElement.destination !=""))
            document.location.assign(this.selectedElement.destination) ;
    },
    onMouseMove: function ( event,ray ) {
        document.body.style.cursor = 'pointer';
    },
    onMouseWheel: function ( event ) {
    },
    render:function()
    {
        this.currentTime = new Date().getTime();
        if (((this.currentTime-this.prevTime)>this.timeSlot)&&(this.stop
==false))
        {
            this.index+=screen_w/200;
            this.requiredWidth = this.mainCanvas.width;
            for ( xx=0; ((xx<this.length)&&(this.requiredWidth >0)); xx++)
            {
                if (this.tempIndex >=
this.marqueeElements[xx].canv.width)

```

Figure A.9(page 6 of 8): Marquee class code

```

    {
        this.marqueeElements[xx].changeWidth(1,0);
        this.marqueeElements[xx].changePosition(this.mainCanvas.width/2,0);
        this.tempIndex-=this.marqueeElements[xx].canv.width;
    }
    else if (this.tempIndex >= 0)
    {
        this.tempWidth =
this.marqueeElements[xx].changeWidth(this.requiredWidth,this.tempIndex);
        if (this.tempWidth == this.requiredWidth)
        {
            this.marqueeElements[xx].changePosition(this.mainCanvas.width/2-
this.requiredWidth/2,0);
            this.tempIndex=-1;
            this.requiredWidth = 0;
        }
        else
        {
            this.marqueeElements[xx].changePosition(this.mainCanvas.width/2 -
this.requiredWidth + this.tempWidth/2,0);
            this.tempIndex=0;
            this.requiredWidth -= this.tempWidth;
        }
    }
    for (var y=xx; y<this.length; y++)
    {
        this.marqueeElements[y].changeWidth(1,0);
        this.marqueeElements[y].changePosition(this.mainCanvas.width/2,0);
    }
    this.prevTime = this.currentTime;
}
},
getInactivePosition:function()
{
    return this.inactivePositionP;
},
getCurrentInactivePosition:function()
{
    return this.marqueeObject.position;
},
getRotation:function()
{
    return this.rotation;
},
getSide:function()
{
    return this.inactiveSide;
},

```

Figure A.9(page 7 of 8): Marquee class code

```
is_Active:function ()
{
  return false;
},
changePosition:function(positionVector)
{
  this.marqueeObject.position = positionVector;
}
};
```

Figure A.9(page 8 of 8): Marquee class code

```

Hyperlink = function ( side,BGImg,BGcolor, title,titleColor,titleFont,
titleFontSize,underlined,bold,italic,opacity,
sizeVector3,positionVector2,movable,rotate,destination,jrcode) {
this.inactiveSide = side;
this.BGImgSrc = BGImg;
this.title = title;
this.titleFont = titleFont;
this.titleFontSize = titleFontSize;
this.opacity = opacity;
this.size = sizeVector3;
this.position = positionVector2;
this.inactiveMovable = movable;
this.underlined = underlined;
this.bold= bold;
this.italic = italic;
this.destination = destination;
this.jrcode =jrcode;
this.sizeP = null;
this.inactivePositionP=null;
this.activeinactivePositionP=null;
this.frame = null;
this.rotation=new THREE.Vector3(0,0,0);
this.BGcolor = BGcolor;
this.BGColor = new THREE.Color(BGcolor);
this.titleColor = new THREE.Color(titleColor);
this.sizeP = new THREE.Vector3(this.size.x*side.width,
this.size.y*side.height, this.size.z);
if (this.inactiveSide.id==0)
{
this.inactivePositionP = new THREE.Vector3(
this.inactiveSide.topCorner_x
+this.sizeP.x/2+(this.position.x*side.width),
this.inactiveSide.topCorner_y -this.sizeP.y/2-
(this.position.y*side.height), 2-screen_d );
this.rotation.z = rotate*Math.PI/180;
}
else if (this.inactiveSide.id==1)
{
this.inactivePositionP = new THREE.Vector3(
this.inactiveSide.topCorner_x +2,
this.inactiveSide.topCorner_y -this.sizeP.y/2-
(this.position.y*side.height),
this.inactiveSide.topCorner_z-this.sizeP.x/2-(this.position.x*side.width) );
this.rotation.y = Math.PI/2;
this.rotation.x = rotate*Math.PI/180;
}
else if (this.inactiveSide.id==2)
{

```

Figure A.10(page 1 of 4): Hyperlink class code

```

        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x -2,
            this.inactiveSide.topCorner_y -this.sizeP.y/2-
(this.position.y*side.height),
this.inactiveSide.topCorner_z+this.sizeP.x/2+(this.position.x*side.width) );
        this.rotation.y = Math.PI/-2;
        this.rotation.x = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==3)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.sizeP.x/2+(this.position.x*side.width),
            this.inactiveSide.topCorner_y -2,
            this.inactiveSide.topCorner_z -this.sizeP.y/2-
(this.position.y*side.height) );
        this.rotation.x = Math.PI/2;
        this.rotation.z = rotate*Math.PI/180;
    }
    else if (this.inactiveSide.id==4)
    {
        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.sizeP.x/2+(this.position.x*side.width),
            this.inactiveSide.topCorner_y +2,
            this.inactiveSide.topCorner_z
+this.sizeP.y/2+(this.position.y*side.height) );
        this.rotation.x = Math.PI/-2;
        this.rotation.z = rotate*Math.PI/180;
    }
    this.createCanvas();
    this.hg = new THREE.PlaneGeometry( this.sizeP.x,this.sizeP.y, 1, 1);
    this.frame = new THREE.Mesh(this.hg,this.material );
    this.frame.rotation.x = this.rotation.x;
    this.frame.rotation.y = this.rotation.y;
    this.frame.rotation.z = this.rotation.z;
    this.frame.position = new THREE.Vector3(
this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z);
    scene.add( this.frame );
    if ((this.BGImgSrc != null)&&(this.BGImgSrc!=""))
    {
        this.BGImage = new Image();
        this.BGImage.src = this.BGImgSrc;
        var parent = this;
        this.BGImage.onload = function() {
            parent.createCanvas();
            scene.remove( parent.frame );
            scene.add( parent.frame );
        };
    };

```

Figure A.10(page 2 of 4): Hyperlink class code

```

    }
    return this;
};
Hyperlink.prototype = {
    constructor: Hyperlink,
    createCanvas: function()
    {
        var textHolder = document.createElement( 'canvas' );
        var ctext = textHolder.getContext('2d');
        textHolder.width = this.sizeP.x;
        textHolder.height = this.sizeP.y;
        if (this.BGcolor != null)
        {
            ctext.fillStyle = this.BGcolor.getContextStyle();
            ctext.fillRect(0, 0, this.sizeP.x, this.sizeP.y);
        }
        if ((this.BGImage != null)&&(this.BGImage!=""))
        {
            ctext.drawImage(this.BGImage, 0,0,textHolder.width,
textHolder.height);
        }
        if ((this.title != null)&&(this.title!=""))
        {
            this.title_font_Size = this.sizeP.y*this.titleFontSize/2;
            var a = "";
            if (this.italic)
                a += "italic ";
            if (this.bold)
                a += "bold ";
            ctext.font = a+this.title_font_Size+"px
"+this.titleFont;//"Arial";
            ctext.textAlign = "center";
            ctext.fillStyle = this.titleColor.getContextStyle();
            ctext.fillText(this.title, this.sizeP.x/2,this.sizeP.y*0.6);

            if (this.underlined)
            {
                var metrics = ctext.measureText(this.title);
                ctext.strokeStyle = this.titleColor.getContextStyle();
                ctext.stroke();
            }
        }
        this.material = new THREE.MeshBasicMaterial({map: new
THREE.Texture(textHolder), opacity:this.opacity, transparent:true});
        this.material.map.needsUpdate = true;
    },
    checkIntersection: function ( object ) {
        if ( this.frame == object)
        {
            return true;
        }
    }
};

```

Figure A.10(page 3 of 4): Hyperlink class code


```

}
},
onMouseDown: function ( event,ray ) {
    this.mousedown = true;
},
onMouseUp: function ( event ) {
    this.mousedown=false;
},
onClick: function ( event ) {
    if ((this.destination!=null)&&(this.destination !=""))
        document.location.assign(this.destination) ;
    else if ((this.jscode!=null)&&(this.jscode !=""))
        eval(this.jscode);
},
onMouseMove: function ( event,ray ) {
    if (((this.destination!=null)&&(this.destination
!=""))||((this.jscode!=null)&&(this.jscode !="")))
        document.body.style.cursor = 'pointer';
},
onMouseWheel: function ( event ) {
},
getDimentions:function()
{
    return this.sizeP;
},
getInactivePosition:function()
{
    return this.inactivePositionP;
},
getCurrentInactivePosition:function()
{
    return this.frame.position;
},
getRotation:function()
{
    return this.rotation;
},
getSide:function()
{
    return this.inactiveSide;
},
is_Active:function()
{
    return false;
},
changePosition:function(positionVector)
{
    this.frame.position = positionVector;
}
};

```

Figure A.10(page 4 of 4): Hyperlink class code

```

D3Object = function (mesh, side,inactiveScale
,skin,inactivePositionVector3,inactiveMovable,
activeScale,activePositionVector3,activeActionButtonArray,rotate,clickable) {
  this.frame = mesh;
  this.inactiveSide          = side;
  this.inactiveScale        = inactiveScale;
  his.inactivePosition      = inactivePositionVector3;
  this.inactiveMovable      = inactiveMovable;
  this.skin = skin;
  this.clickable = clickable;
  this.activeScale          = activeScale;
  this.activePosition       = activePositionVector3;
  this.activeActionbuttons = activeActionButtonArray;
  this.isActive = false;
  this.inactiveSizeP = null;
  this.activeSizeP = null;
  this.inactivePositionP=null;
  this.activePositionP=null;
  this.lastActivePositionP=null;
  this.intersectPointMouseDown=null;
  this.mousedown = false;
  this.offset=new THREE.Vector3();
  this.rotation=new
THREE.Vector3(rotate.x*Math.PI/180,rotate.y*Math.PI/180,rotate.z*Math.PI/180);
  this.objectNumber=side.getObjectID();
  this.buttonArray=new Array();
  this.lights=new Array();
  this.particle=new Array();
  this.timeReach=0;
  this.afterWaitOp = 0;
  var xmlhttp;
  if (this.clickable)
  {
    ///////////////////////////////////////////////////////////////////
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
      xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
      xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    //alert(this.skin+"skin.xml");
    xmlhttp.open("GET",this.skin+"skin.xml",false)
    //alert(xmlhttp);
    xmlhttp.send();
    xmlDoc=xmlhttp.responseXML;
    //alert(xmlDoc);
    x=xmlDoc.getElementsByTagName("D3Object");
  }
}

```

Figure A.11(page 1 of 8): D3Object class code

```

mesh.geometry.computeBoundingBox();
var xx,yy,zz;
xx = mesh.geometry.boundingBox.x[1]-mesh.geometry.boundingBox.x[0];
yy = mesh.geometry.boundingBox.y[1]-mesh.geometry.boundingBox.y[0];
zz = mesh.geometry.boundingBox.z[1]-mesh.geometry.boundingBox.z[0];
this.inactiveSizeP = new THREE.Vector3(inactiveScale*xx,
inactiveScale*yy, inactiveScale*zz);
this.activeSizeP=new THREE.Vector3(activeScale*xx, activeScale*yy,
activeScale*zz);
this.activePositionP = new THREE.Vector3(screen_w
*this.activePosition.x+this.activeSizeP.x/2-screen_w/2,
screen_h/2-
(screen_h*this.activePosition.y+this.activeSizeP.y/2), 5);
this.lastActivePositionP = new THREE.Vector3();
this.lastActivePositionP.x =this.activePositionP.x;
this.lastActivePositionP.y =this.activePositionP.y;
this.lastActivePositionP.z =-1;
if (this.inactiveSide.id==0)
{
this.inactivePositionP = new THREE.Vector3(
this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
this.inactiveSide.topCorner_y -
this.inactiveSizeP.y/2-(this.inactivePosition.y*side.height),
(this.inactivePosition.z*screen_d
+this.inactiveSizeP.z/2) -screen_d);
}
else if (this.inactiveSide.id==1)
{
this.inactivePositionP = new THREE.Vector3( -
screen_w/2+(this.inactivePosition.z*screen_w +this.inactiveSizeP.z/2) ,
this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),this.inactiveSide.topCorner_z-
this.inactiveSizeP.x/2-(this.inactivePosition.x*side.width) );
this.rotation.y += Math.PI/2;
}
else if (this.inactiveSide.id==2)
{
this.inactivePositionP = new THREE.Vector3(
screen_w/2-
(this.inactivePosition.z*screen_w +this.inactiveSizeP.z/2),
this.inactiveSide.topCorner_y -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height),
this.inactiveSide.topCorner_z+this.inactiveSizeP.x/2+(this.inactivePosition
.x*side.width) );
this.rotation.y += Math.PI/-2;
}
else if (this.inactiveSide.id==3)
{

```

Figure A.11(page 2 of 8): D3Object class code

```

        this.inactivePositionP = new THREE.Vector3(
            this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
screen_h/2-(this.inactivePosition.z*screen_h+this.inactiveSizeP.z/2),
            this.inactiveSide.topCorner_z -this.inactiveSizeP.y/2-
(this.inactivePosition.y*side.height) );
            this.rotation.x += Math.PI/2;
        }
        else if (this.inactiveSide.id==4)
        {
            this.inactivePositionP = new THREE.Vector3(
                this.inactiveSide.topCorner_x
+this.inactiveSizeP.x/2+(this.inactivePosition.x*side.width),
-screen_h/2+(this.inactivePosition.z*screen_h+this.inactiveSizeP.z/2),
                this.inactiveSide.topCorner_z
+this.inactiveSizeP.y/2+(this.inactivePosition.y*side.height) );
                this.rotation.x += Math.PI/-2;
            }
            this.object3D.add( this.frame );
            this.restoreButtonEnabled=false;
            this.restoreSizeButtonEnabled=false;
            this.moveButtonEnabled=false;
            var ii=0,temp;
            this.b_w=screen_w/35;
            this.b_h=screen_w/35;
            if (this.clickable)
            {
                temp =
(x[0].getElementsByTagName("restoreButton")[0].childNodes[0].nodeValue);
                this.restoreButton = imgButton(skin+temp, this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
                this.lastActivePositionP.y+this.activeSizeP.y/2,this.lastActivePositionP
.z+4,0.8);
                this.restoreButtonEnabled=true;
                this.buttonArray[ii++] = this.restoreButton;
                if ((activeActionButtonArray&2 ) ==2)
                {
                    temp =
(x[0].getElementsByTagName("restoreSizeButton")[0].childNodes[0].nodeValue)
;
                    this.restoreSizeButton = imgButton(skin+temp,this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
                    this.lastActivePositionP.y+this.activeSizeP.y/2-
ii*this.b_h,this.lastActivePositionP.z+4,0.8);
                    this.restoreSizeButtonEnabled=true;
                    this.buttonArray[ii++] = this.restoreSizeButton;
                }
            }
            if ((activeActionButtonArray&4 ) ==4)
            {

```

Figure A.11(page 3 of 8): D3Object class code

```

        temp =
(x[0].getElementsByTagName("moveButton")[0].childNodes[0].nodeValue);
        this.moveButton = imgButton(skin+temp,this.b_w,
this.b_h,this.lastActivePositionP.x+this.activeSizeP.x/2,
        this.lastActivePositionP.y+this.activeSizeP.y/2-
ii*this.b_h,this.lastActivePositionP.z+4,0.8);
    }
}
wc.addRender(this);
return this;
};
D3Object.prototype = {
constructor: D3Object,
checkIntersection: function ( object ) {
    if (this.isActive )
    {
        for (var ii=0; ii<this.buttonArray.length; ii++)
        {
            if (this.buttonArray[ii] == object)
            {
                this.selectedButton = ii;
                return true;
            }
        }
    }
    for ( var i = 0; i< this.object3D.children.length; i ++ )
    {
        if (this.object3D.children[ i ]==object)
        {
            this.selectedButton = -1;
            return true;
        }
    }
    return false;
},
show:function()
{
    this.object3D.position = new THREE.Vector3(
this.inactivePositionP.x,this.inactivePositionP.y,this.inactivePositionP.z)
;
    scene.add(this.object3D);
},
centroidPosition:function(x,y,z)
{
    this.inactivePositionP.x = x;
    this.inactivePositionP.y = y;
    this.inactivePositionP.z = z;
},
onMouseDown: function ( event,ray ) {
    this.mousedown = true;

```

Figure A.11(page 4 of 8): D3Object class code

```

        if (this.isActive ==true)
        {
            this.mousePositionOnMouseDownActive=new
THREE.Vector2(event.clientX,event.clientY);
            this.mouseMovePosition=new
THREE.Vector2(event.clientX,event.clientY);
        }
    },
    onMouseUp: function ( event ) {
        this.mousedown=false;
        if (this.isActive==true)
        {
            this.lastActivePositionP.x=this.object3D.position.x;
            this.lastActivePositionP.y=this.object3D.position.y;
        }
    },
    onClick: function ( event ) {
        if (this.isActive)
        {
            if (this.selectedButton>=0)
            {
                if (this.buttonArray[this.selectedButton] ==
this.restoreButton)
                {
                    this.restore();
                }
            }
            else if
(this.buttonArray[this.selectedButton]==this.restoreSizeButton)
            {
                new TWEEN.Tween( this.object3D.rotation ).to( {x:0,y:0,z:0},
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
                for (var i = 0; i<this.buttonArray.length;i++)
                    new TWEEN.Tween( this.buttonArray[i].position ).to(
{x:this.buttonArray[i].position.x,y:this.buttonArray[i].position.y,z:4},
200).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
                this.wait(205);
            }
        }
    }
    else
    {
        if (this.clickable)
        {
            this.lastActivePositionP.z = wc.addActiveObject(this);
            new TWEEN.Tween( this.object3D.position ).to(
{x:this.lastActivePositionP.x,y:this.lastActivePositionP.y,z:this.lastActiv
ePositionP.z}, 500).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
            var sc = this.activeScale;
            new TWEEN.Tween( this.object3D.scale ).to( { x:sc, y:sc,
z:sc}, 500 ).easing(TWEEN.Easing.Sinusoidal.EaseOut).start();
        }
    }
}

```

Figure A.11(page 5 of 8): D3Object class code

```

        this.wait(500);
        this.afterWaitOp =2;
    }
}
},
onMouseMove: function ( event,ray ) {
    if (this.mousedown)
    {
        if (this.isActive)
        {
            if
((this.selectedButton>=0)&&(this.buttonArray[this.selectedButton] ==
this.moveButton))
            {
                this.object3D.position.x = this.lastActivePositionP.x
+(event.clientX-this.mousePositionOnMouseDownActive.x);
                this.object3D.position.y = this.lastActivePositionP.y -
(event.clientY-this.mousePositionOnMouseDownActive.y);
                if (this.object3D.position.x>(screen_w/2))
                    this.object3D.position.x = screen_w/2;
                if (this.object3D.position.x<(screen_w/-2))
                    this.object3D.position.x = screen_w/-2;
                this.repositionButtons();
            }
            else if (this.selectedButton < 0)//frame
            {
                this.object3D.rotation.y+=((event.clientX-this.mouseMovePosition.x))/100;
                this.object3D.rotation.x+= ((event.clientY-this.mouseMovePosition.y))/100;
                this.mouseMovePosition.x=event.clientX;
                this.mouseMovePosition.y=event.clientY;
            }
        }
    }
},
onMouseWheel: function ( event ) {
    if (this.isActive)
    {
        // WebKit
        var pos=new
THREE.Vector3(this.object3D.position.x,this.object3D.position.y,this.object
3D.position.z);
        if ( event.wheelDeltaY ) {
            pos.z+= event.wheelDeltaY ;
            // Opera / Explorer 9
        } else if ( event.wheelDelta ) {
            pos.z += event.wheelDelta ;
            // Firefox
        } else if ( event.detail ) {
            pos.z -= event.detail *20;
        }
    }
}
}

```

Figure A.11(page 6 of 8): D3Object class code

```

        if (pos.z>(screen_h-this.activeSizeP.z/2))
            pos.z = (screen_h-this.activeSizeP.z/2);
        else if (pos.z<(screen_d/-1+100))
            pos.z = screen_d/-1+100;
        this.object3D.position= pos;
        this.wait(1);
        this.afterWaitOp = 1;
    }
},
repositionButtons: function()
{
    var ii=0, xxx = this.object3D.position.x+(this.buttonArray.length-
1)*this.b_w/2;
    for (ii=0; ii<this.buttonArray.length; ii++)
    {
        this.buttonArray[ii].position.x = xxx-ii*this.b_w;
        this.buttonArray[ii].position.z = this.object3D.position.z+4;
    }
},
wait: function(msec)
{
    this.timeReach=new Date().getTime()+msec;
},
afterWait: function()
{
    if (this.afterWaitOp ==2)
    {
        this.isActive = true;
        for (var x=0;x<this.buttonArray.length;x++)
        {
            scene.add( this.buttonArray[x] );
        }
        this.repositionButtons();
    }
    else if (this.afterWaitOp ==1)
    {
        this.repositionButtons();
    }
},
restore:function( )
{
    this.isActive = false;
    for (var x=0;x<this.buttonArray.length;x++)
    {
        scene.remove( this.buttonArray[x] );
    }
    new TWEEN.Tween( this.object3D.position ).to(
this.inactivePositionP,500).easing(TWEEN.Easing.Sinusoidal.EaseIn).start();

```

Figure A.11(page 7 of 8): D3Object class code


```

        new TWEEN.Tween( this.object3D.rotation ).to( {x:0,y:0,z:0}, 500).
easing(TWEEN.Easing.Sinusoidal.EaseIn).start();
        this.isActive = false;
        this.mousedown=false;
    },
    render:function()
    {
        if (this.afterWaitOp!=0)
        {
            var t =new Date().getTime();
            if (t>=this.timeReach)
            {
                this.afterWait();
                this.afterWaitOp=0;
            }
        }
    },
    addMesh: function (mesh,scale)
    {
        mesh.scale.x=mesh.scale.y=mesh.scale.z = scale;
        this.object3D.add(mesh);
    },
    getDimentions:function()
    {
        return this.inactiveSizeP;
    },
    getInactivePosition:function()
    {
        return this.inactivePositionP;
    },
    getCurrentInactivePosition:function()
    {
        return this.object3D.position;
    },
    getRotation:function()
    {
        return this.rotation;
    },
    getSide:function()
    {
        return this.inactiveSide;
    },
    is_Active:function()
    {
        return this.isActive;
    },
};

```

Figure A.11(page 8 of 8): D3Object class code