EFFECT OF GAME DEVELOPMENT-BASED LEARNING ON THE ABILITY OF INFORMATION TECHNOLOGY UNDERGRADUATES TO LEARN COMPUTER AND OBJECT-ORIENTED PROGRAMMING

by

ALAA KHALAF AL-MAKHZOOMY

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2018

MAJOR: LEARNING DESIGN AND TECHNOLOGY

Approved By:

Advisor	Date
Co-Advisor	Date

DEDICATION

and say, "O my Lord! advance me in knowledge."

Surah Taha 114 – the Holy Quran

To my loving parents Dr. Khalaf Al-Makhzoomy and Mona Amin Suleiman; the educators who made me fond of teaching and pursuing my Ph.D. degree.

To my beloved wife Samah Mallak for her continued love and care.

To our bundle of joy, our long-awaited baby, our son; Kenan.

Loving son, husband and father,

Alaa

ACKNOWLEDGMENTS

"He has not thanked Allah who has not thanked people."

— Prophet Muhammad

I would like to express my sincere gratitude to all the people who helped me throughout my Ph.D. study.

My sincere gratitude to my Dissertation Chair Professor Ke Zhang for believing in me as a computer scientist who could become a learning designer and an accomplished researcher in the field of Learning Design and Technology, for the continuous support and the encouragement doses she provided at the right times. I do not imagine that I would have had a better advisor. My sincere gratitude goes as well to my Dissertation Co-Chair Dr. Timothy Spannaus, my role model in teaching, for teaching me how to design instruction, for his patience, his thoughts and insights for a better dissertation experiment design.

Besides my wonderful advisors, I would like to thank the rest of my committee; Professor Ingrid Guerra-Lopez and Professor Monica Tracey for their comments, encouragement and challenging questions to expand my thoughts to various perspectives. Also, I would like to thank Dr. Kevin Deegan-Krause for stepping in as an outside committee member, for his great example of a caring teacher he modeled through the first-year courses of the Irvin D. Reid Honors College and for showing how course design would be at its excellence.

My sincere thanks also go to Dr. Qasim Al-Radaideh of Yarmouk University, my professor in my undergraduate studies, who advocated for me to conduct my doctoral research experiment at Yarmouk University and made it happen. Thanks to him for his continuous support and encouragement.

I would like to thank the brilliant statisticians for their insights and help; Dr. D. Carl Freeman for opening my eyes on future research and Dr. Malek Abdel Jaber for generously providing his time and expertise which helped me with the statistical analysis of this thesis.

I would like to thank my direct supervisor at the Irvin D. Reid Honors College; Kevin Rashid who not only taught me to become a better staff member, but also showed some extraordinary patience with me to become a better thinker, better speaker of English and a better scholar. Thanks Kevin for the great coffee talks. My sincere thanks go to the rest my colleagues at the Irvin D. Reid Honors College; especially Dean John Corvino who always supported me as a member of the Honors College, Kevin Piotrowski and Ali Salamey for their friendship, wisdom and encouragement. My sincere thanks go to Emeritus Dean Jerry Herron and the Honors Senior Lecturers who always inspired me with their thoughts and discussions.

Finally, thanks to all my family members for their continuous motivating and reassuring words, for their support and for believing in me as a scholar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

Learning computer programming is always perceived by learners as a steep learning curve due to many reasons. One of them is the advancement of high-level languages that are used in instruction to teach computer programming. In addition, the increase in the number of programming languages that are used in the market makes it harder for students who are majoring in Information Technology (IT) to learn the concepts of computer programming and Object-Oriented Programming (OOP) in the first year of their undergraduate study.

The instruction being used to teach computer programming should be designed as a learner-centered instruction refraining from being an instructor-centered instruction. The more the learner experiences a hands-on computer programming in a challenging manner that moves them from one level of understanding to another in seek to master the concepts and skills, the more the instruction is successful in achieving the goal. Collaboration and scaffolding are main aspects of the learning process. Problem-based learning will intrigue the cognitive skills of the students to tackle the programming tasks and projects. And finally, the approach where the educator sits aside as a facilitator of the learning process, while learners engage in the process of constructing their own knowledge about the subject matter in a social collaborative learning environment, is the approach that the researcher believes that will be more beneficial and fruitful in learning computer programming in a high-level language.

Problem Statement

The undergraduate students in the IT disciplines in Jordanian universities study computer programming in their first semester of their undergraduate programs. Depending on the program they are enrolled in, they study computer programming with either C++ or Java ("JUST IT Undergraduate Courses According to 2015 Curriculum," 2016) ("YU Computer Sciences Courses

| Faculty of IT," 2016) ("Mutah IT Course Descriptions," 2016). They come to the programs from various regions of the country, where there are different lifestyles; urban and rural areas, and a huge socioeconomic gap in between, which has its impact on their levels of learning, especially computer and technology competency (Beck, 2010).

For those who come from less fortunate areas, it becomes harder to cope with their learning at the university level, where everything is different from what they had been through the last twelve years of their lives at school and within their communities. In addition, peer pressure may play a role with their learning at the university level, where these learners will have to compete with those learners who come from affluent urban areas, hence better educated and exposed to computer technology at earlier ages.

Given the fact that programming languages are based on English language, this would add more burden on learning computer programming on learners in Jordan, where students learn English as a second language in their elementary and high schools. In addition, the burden would be more, specifically on those who did not receive education that is comparable to those who were fortunate enough to be more competent in English language in other parts of the country.

Therefore, it would be harder for the students who come from less affluent areas, either from urban or rural areas, to learn how to program computers using a high-level language, such as C++ or Java.

Jordanian Universities

Jordan has eleven public and twenty-five private universities. Most of them have undergraduate-level programs in Information Technology. According to the Ministry of Higher Education of Jordan, in the year 2013, the number of students who were admitted to computer science and mathematics programs - combined - in all Jordanian universities were 5931 students.

Among them, 3015 were females; i.e. more than 50% of them ("The Ministry of Higher Education of Jordan," 2016).

In the year 2015, the number of students who were admitted to computer science programs in all Jordanian universities were 1416 students, 633 of them are females (The Ministry of Higher Education of Jordan, 2017).

Yarmouk University of Jordan

Yarmouk University is an urban public university, located in Irbid city in the northern of Jordan. The university consists of 15 colleges, one of them is the Faculty of IT which consists of five departments; Computer Science, Computer Information Systems, Management Information Systems, Network and Information Security and Software Engineering. Each of the first three departments offer two programs; a Bachelor of IT and a Master's degree program, while the latter two departments offer only Bachelor of IT programs. ("Yarmouk University," 2017).

In the year 2015, the number of students enrolled in undergraduate programs of Yarmouk University was 33334 students, 19848 of them were females (The Ministry of Higher Education of Jordan, 2017). While the number of students who were admitted to all undergraduate programs in the Faculty of IT at Yarmouk University were 259 students, 183 of them were females ("Yarmouk University," 2017). Those who were admitted to the computer science program in the Faculty of IT at Yarmouk University were 128 students, 86 of them were females (The Ministry of Higher Education of Jordan, 2017).

Purpose of the Study

Despite all of the enormous efforts that had been taken by Jordanian institutions at the national level, there still exist barriers to integrating and utilizing ICT in education in Jordan

(Alkhawaldeh & Menchaca, 2014). These barriers play a major role in expanding the competency gap between learners of IT disciplines.

Learners of IT disciplines face difficulties especially when it comes to learn computer programming. That is, according to Ismail, Ngah, & Umar (2010), undergraduate IT students tend to "lack skills in analyzing problems." Also, the "ineffective use of problem representation techniques for problem solving," and the "ineffective use of teaching strategies for problem solving coding" by educators lead to low motivation among learners, therefore learners react passively to learning computer programming (Ismail et al., 2010).

According to other research studies conducted in countries other than Jordan, high school and college learners show higher motivation when game design and development is used as a vehicle to promote learning computer programming (Papastergiou, 2009) (Overmars, 2004).

Thus, the researcher seeks to investigate the effect of Game Development-Based Learning (GDBL) through constructivist learning environment on IT undergraduates in Jordanian universities learning computer and Object-Oriented Programming (OOP) using a high-level language, such as C++.

Research Questions

The study aims to investigate the following questions:

1. What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to define the concepts of computer programming and OOP?
2. What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to program in a high-level language, such as C++?

Theoretical Constructs

Constructivist learning

A theory developed by Piaget (1976) that focuses on problem-solving, where learners construct their own knowledge in an environment where their experiences and new ideas interact to construct that new knowledge. It is believed that the origins of constructivism go back to John Dewey's Pedagogical Creed were he said: "education must be conceived as a continuing reconstruction of experience; that the process and the goal of education are one and the same thing" (Dewey & Small, 1897).

Seymour Papert, a student of Piaget's, built on his constructivist learning theory and introduced the constructionist learning theory. In this theory, Papert & Harel (1991) implies that students learn best by making tangible objects through authentic, real life learning opportunities that allow for a guided, collaborative process which incorporates peer feedback.

The two theories; constructivist learning theory developed by Piaget (1976) and constructionist learning theory developed by Papert & Harel (1991), are sometimes treated as one theory, while they are different. They both work towards similar objectives but via different ways, that is the two theories consider that knowledge is being constructed incrementally, and learners construct the knowledge based on their own experiences. On the other hand, what makes the two theories apart is that Piaget focuses on "the construction of internal stability," more mental and cognitive. While Papert focuses on the connection between learners and the physical world around them.

An obvious distinction between the two theories could be seen in Jong, Shang & Lee (2010), who introduce the idea of using game development in teaching founded on Piaget's constructivist learning theory. They aimed at "discussing what and how the intrinsic traits of games can promote constructivist learning." They also ended with a conclusion that since learning

through gaming is important, thus there is a pedagogical value to include game-based learning in formal education.

The researcher believes that an instruction for teaching computer programming is more likely to be designed based on this theory, not mentioning that GDBL does not go beyond the boundaries of the definition of a constructivist learning theory. The researcher believes that any instruction designed for K-12 or undergraduate level learners of game design and development using GameMaker® is going to be founded on the constructivist learning theory, especially social constructivist learning, which focuses on learning collaboratively in teams or groups.

Zone of Proximal Development (ZPD)

A concept that was identified by Vygotskiĭ & Cole (1978). Vygotsky defines the ZPD as "the distance between the actual developmental level and the level of potential development." The actual development level is the level of development that a learner can achieve independently without any help or support of others, while the level of potential development is the level of development a learner can achieve with help or support of an expert or through "collaboration with more capable peers" (Vygotskiĭ & Cole, 1978).

The ZPD is synonymous with the term; Scaffolding, which was used by other researchers referring to the same concept. However, scaffolding in the instructional literature, refers to the help being provided by the expert or a peer to move across the ZPD (Arts, Gijselaers, & Segers, 2002).

The instruction the researcher designed as a treatment intervention treatment for this research is based on the two aforementioned theoretical constructs; constructivist learning and ZPD. That is students will construct their own knowledge about game development, they will be able to work collaboratively to help and support each other through the learning process and they will learn by accomplishing hands-on tasks that they will need to work with until they master them.

Significance of the Study

The researcher designed the instruction as a treatment intervention in this study depending on a constructivist learning approach bearing in mind the ZPD and scaffolding approach. Through the experiment, learners will construct their own knowledge about game development and computer programming and OOP concepts through the tutorials and projects they accomplish in the instruction and they will work in collaboration as a scaffolding technique to help them cross the ZPD.

Students who are enrolled in the IT undergraduate programs in Jordanian higher educational institutes will benefit from the experiment of this study in bridging the gap between their lack of mathematical analysis and problem-solving skills and their ability to learn new concepts of the discipline of computer programming and to approach problem solving effectively. Learners, educators, educational institutes, learning designers and researcher may benefit of the results of this research, especially those who are experimenting or are designing interventions that implement a constructivist learning approach, a problem-based learning approach, and/or game development-based approach.

Definition of Key Terms

In this research, the following terms and concepts are identified as follows:

Constructivism: a theory about learners construct their own understanding and knowledge of the world, through experiencing things and reflecting on those experiences.

Object-Oriented Programming (OOP): a computer programming realm where the world is seen as "objects" not actions and the process of writing a program concerns with the problem of defining data not defining procedures (Kindler & Krivy, 2011).

GameMaker®: A freeware, that could be installed on Windows or Mac machines, was first developed by (Overmars, 2004), Now it is provided by ("YoYo Games," 2016). GameMaker® is defined by its creator and developer; Mark Overmars, as tool that helps with developing two-dimensional computer games, and helps to support OOP concepts. These objects are triggered by events. The environment is drag-and-drop, which minimizes the need to write code."

Similar to OOP languages, its user can create objects, but unlike OOP, most of these objects have visual representations in the game room. Also, many instances could be created, i.e. copies of the same object, but each one of those instances may have its own properties.

Each one of the objects has its own behavior, and this behavior is governed by events and actions that are attached to them. If an event occurs with an object, the attached action is triggered. As an example, imagine that a learner created a circular-shape object, and associated the Press-Up key with it along with an action of moving upwards attached to this event. Therefore, if the down-arrow key on the keyboard is pressed, the action associated with this action is going to be triggered, and the object moves in the up direction. This could be accomplished either by writing code, or by using command that are in the drag-and-drop lists of the application.

CHAPTER 2 REVIEW OF LITERATURE

Varying between qualitative, quantitative, and mixed mode methods, the literature that had been reviewed by the researcher based their assumptions on the importance of using game development in teaching in general. They all agreed on a recognizable pedagogical value of a Game Development-Based Learning (GDBL) approach.

Constructivist Learning

Baytak, Land, & Smith (2011) founded their research on Papert's theory, not neglecting its origins in Piaget's theory. Founded on Papert's constructionist theory, Majgaard (2014) aligned the characteristics of Papert's constructionist theory with its origins in Piaget's constructivist theory. In addition to that Majgaard (2014) emphasized on the influence of culture on shaping cognitive development. Thus he linked his work to Vygotsky's Zone of Proximal Development (ZPD), which states the idea that playing with peers "creates a zone where learners behave beyond their average age and above their daily behavior" (Vygotskiĭ & Cole, 1978). Thus, he created an environment for his learners that encourages them to work in groups to encourage them to learn.

A study that relies on the constructionist learning theory in designing the instruction that was used in the study, was the study conducted by Marlow (2012). The instruction was designed around GDBL to promote learning Landscape Architecture (LA).

Another study, published by Pretelín-ricárdez & Sacristán (2015), was conducted with the intention to promote the modelling activity in a constructionist learning approach depending on Pappert's theory. However, the authors did not distinguish between the two theories; constructionism and constructivism. That is by the end of their published article, they refer to their approach as being constructivist, while they refer to it in the beginning of the article as being constructionist.

Neither constructivist nor constructionist

The two theories; Piaget's and Papert's are not mentioned in the remaining studies being reviewed in this research, but they followed the same ways to build their instruction design. Without referring to neither of the two theories; constructivist nor constructionist learning, Ernst & Clark (2012) explained their design of a module adjunct to an online course that is taught at North Carolina Virtual Public School. However, it could be said that their instruction is founded on Piaget's constructivist theory. Because it allows learners to construct their knowledge by creating games by their own, while they get some guidance to help them grow their knowledge gradually.

Like Ernst & Clark (2012), not mentioning any of the two theories, Doman Sleigh & Garrison (2015) provided a short description of the instruction they used for their intervention courses. It could be inferred that they focused on having their learners construct their own knowledge around their experiences. They provided them with tutorials that helped them to construct games, and then they gave them the opportunity to design their own games.

Without depending on any of the two theories, the study conducted by Chandler (2013) directly assumes the importance of embedding technology in the instruction of middle schools, and since the students are getting more acquainted with technology, then schools and teachers are ought to use technology in the instruction.

Another study where the authors do not rely on either constructivist nor constructionist is the study conducted by Charlier & De Fraine (2012). On the other hand, they refer to the same concept of GDBL mentioned by Wu & Wang (2012) by referring to the learning approach they are introducing in their study as a Digital Game Based Learning (DGBL) approach, distinguishing it from the well-known GBL approach.

Lastly, Patton (2011), in his thesis, did not describe in detail the nature of the instruction he had used in his experiment. But instead, he illustrated his learners' designs, and provided a description of the different aspects they could incorporate in their games.

Game Development-Based Learning (GDBL)

A distinguishing notion that was introduced by Wu & Wang (2012) in their review of the research that introduce teaching using game design and development in a constructivist learning approach. The authors clearly identify the approach as Game Development-Based Learning (GDBL) clearly distinguishing it from other types of GBL or Gamification. The researcher will be using this notion to identify his research strategy.

Games in Education vs. Gamification. In their chapter, Jong et al., (2010) introduce two different notions. The first is Education in Games where educators adopt the ready-made commercial games into instruction, and this is known as Gamification or Game-Based Learning (GBL) in other research. The latter is founded on the constructivist learning theory, where learners build their learning via designing computer games, specifically educational games. The authors depend on the new term referred to by Wu & Wang (2012); GDBL, hence they recognize a difference between the two concepts; GDBL or Games-in-Education on one hand, and Gamification or GBL on the other hand.

Charlier & De Fraine (2012) focused on preparing the next generation of teachers, emphasizing on "the importance of learning with technology rather than learning about technology." Therefore, they introduced the concepts behind adopting GDBL with a different notion; Digital Game-Based Learning (DGBL). With that, the authors considered the idea of developing games as part of the larger umbrella; Gamification or Game-Based Learning (GBL). But at the same time, they distinguish learning through game design and development from other

types of gamification with the word “digital” referring to computer games not any other types of games.

This concept; GDBL or DGBL, were not referred by the remaining studies in this literature review. That is, they do not investigate into the reasons or the importance of using GDBL or GBL as educational approaches.

Contributors not Native Gamers. When game development is being the subject matter, learners become the designers, creators and producers of their own computer games. They are not those learners who only consume commercial games that are sold in the market. This was referred to by Majgaard (2014), who referred to game designers as “contributors”, and referred to the consumers who played the commercial games as “native gamers”.

Studies Samples

In terms of the samples of participants of the studies; Doman et al., (2015) worked in their first study with a representative sample consisting of three hundred and sixty participants. Then one hundred and nine participants out of the previous three hundred and sixty participants who participated in the first study accepted to participate in the second study, which was a two-year longitudinal qualitative study. Similar to that study in terms of the sample size, is the study conducted by Chandler (2013). The number of participants from 19 different schools was 800 students, 377 of them were from five schools in the first year of the study, while the remaining 423 participants were students of 14 schools in the second year of the study.

Smaller samples were in the studies conducted by Ernst & Clark (2012) and Charlier & De Fraine (2012). The former had a sample that consisted of only 28 participants, while the latter had a sample that consisted of only 32 participants. A qualitative study was conducted by Baytak et al. (2011). Their sample of participants consisted of six male fifth-graders and four female five-

graders. Similar to that was the case-study qualitative research conducted by Pretelín-ricárdez & Sacristán (2015). In their study, they had a small sample of 12 university engineering students who were asked to build videogames. Those who did not mention the numbers of their participants were Majgaard (2014) and Patton (2011). The former described the participants in his study as engineering freshmen who were taking an introductory course in computer science. While the latter; Patton (2011), focused on younger participants. They were students of four classes, of the ages between 8 and 13 years old.

Research Questions, Instruments, Methods, and Conclusions.

According to what the researchers were trying to support and conclude, the research questions varied. However, all the research studies here in this literature review were targeted to examine the possibility of having a pedagogical value in constructivist learning via game development. In addition, they aimed at examining if this type of learning will enhance learning computer programming.

Reflective approach. Considering a retrospective reflective approach, Majgaard (2014) asked the questions: "How can construction, play, and reflection enrich the game design process for engineering students?". The conclusions that Majgaard (2014) could come up with were many. First, he states that engineering freshmen were successful in shifting from being consumers to be game designers. Second, he introduced two different notions; "reflection-in-games", and "reflection-on-games". The former falls in in the world of game design, where learners work on problem-based programming learning. The latter, which "is the subsequent reflection and evaluation on the process that has occurred and its potential consequences", is nothing but a retrospective discussion of the process of development and learning.

Similar to what Majgaard (2014) did, another study was reflective and did not have research questions being described; Patton (2011) who did not have a sample that was defined. He conducted his study to support his reflective development of the instruction. He used game files that were under development. He used his students' and his colleagues' feedback. He also included the course assessments, pre- and post-tests, and interviews with the participants and their parents. All of the previous were used to develop the course. He came into a conclusion that learners who were participants in his study showed a higher understanding of how interconnected systems work. He also concluded that it is important to train educators to create games to help their learners develop their learning.

Quantitative research. Moreover, Ernst & Clark (2012) clearly described two questions:

> "(1) Can the use of gaming as an instructional tool enhance basic computer science competency for distance education CTE programs?
>
> (2) Can the use of gaming as an instructional tool result in attainment of computer science proficiency?"

Without using a pre-test, Ernst & Clark (2012) used a post-test. The test consisted of fifty points. They analyzed the obtained data using the nonparametric Wilcoxon Signed Ranks Test. This test is used when the study is conducted as repeated measures. Another instrument was used; twenty-eight students were surveyed to capture their opinions using an additional survey consisting of eighteen questions.

In his study which was conducted in Australia, Chandler (2013) aimed to investigate the students' knowledge of what the author refers to as "new media," such as software tools that create three-dimensional games or movies. The study was motivated by the changes that had been made to the Australian curriculum by embedding such new media. The distinguishing characteristic of

this study is that it investigates the students' knowledge about, not only GameMaker®, but also the software authoring tools that help in designing three-dimensional computer games and movies.

In the first year of the study, participants came from five schools. In the second year of the study, the participants came from 14 schools. The schools, mainly governmental schools, varied geographically and socioeconomically. A questionnaire was used to gather data about the software with which students had prior experience. This questionnaire became available online, completed in the first third of the year. The data were analyzed with Pearson's Chi-Squared test and the sign test.

The results showed that the participants identified more than one authoring tool. In addition, some were able to identify GameMaker®. The study showed that the majority of students were able to identify at least one software tool. The study also indicates that game development was an activity that seemed to be favored by males. In the schools in which GameMaker® was identified, males were more likely to have had experience with this software than females. This also applies to another game development tool identified in one the schools. The author concludes that students at school present a diverse experience of several software tools. The author also concludes with that implications of embedding the new media in the curriculum and the knowledge that students have about such media will reflect on the way schools provide education. The teacher is essential in this change, in facilitating students' engagement with the media.

Qualitative research. Three different questions were described by Baytak et al. (2011): "(1) what conceptions of nutrition knowledge were used or evident in the game design? (2) What programming strategies did students use to develop their game over time? (3) What was the role of social interaction on students' game design?" These questions were not directly answered in

their conclusion, but they can be inferred. Having their participants being a focus group of ten five-graders, thus their sole instrument was interviewing them after they completed the instruction.

Some important remarks were noted; the authors observed similarities in gender attitudes and their engagement with the instruction. All games developed by all ten participants were completed and they satisfied the requirements. Also, differences where observed, that is the games that were developed by boys aimed at destroying enemies. On the other hand, the games developed by girls were not aimed at destroying enemies or any type of targets, and they were focused on avoidance and dodging. They also concluded that computers themselves are viewed differently by genders; boys view them as toys to play with, while girls view them as tools to get things accomplished with.

A case-study qualitative research, which was published while the project that it illustrated was still ongoing, was conducted by Pretelín-ricárdez & Sacristán (2015). Participants in the study were engineering student who were asked to build videogames, as part of the mathematical modelling process they had to understand its concepts and the elements involved in it.

The authors offered an optional course that they designed the instruction for. The course topic was about videogame design and programming, where students would put into practice the modelling processes. The course was offered at the National Polytechnic Institute in Mexico City, Mexico.

An initial questionnaire was distributed among participants to capture an insight into the participants' conceptions with regard to the modelling process. Then participants were organized into two teams. Similar to the studies annotated in this article, the authors used GameMaker®. But what was distinguishing their approach was that they used a physics engine that exists within the

application called Box2D. This engine uses the mathematical model of water; hence students can simulate their models within a virtual environment governed by basic physical laws.

The course instruction consisted of three activities designed taking into consideration the principles of modelling. Activity 1 involved collaborative design in pairs of students. Activity 2 required that students individually propose a mathematical model. Activity 3 involved collaborative work in pairs to adapt the models into videogame design.

Focusing on the cases of only two students, the authors concluded that through programming the videogame, "students engaged in producing a working model that was meaningful to them and gained deeper understanding of all the elements involved in the modelling process."

They also found that their constructivist approach helped their students "appreciate how in the real world, modelling processes are iterative and perfectible, and often collaborative."

A third qualitative study in this review was conducted by Marlow (2012). A pilot study about game development toward an environmental design education that embraces games as inspiration for better teaching and learning. The author indicates that "digital technologies have significantly enhanced LA education since the early 1980s." In addition, games had rarely been considered to use in learning LA. The author described the new generation of learners as being "better equipped with the digital skills and confidence to make visualization and learning tasks easier and more effective."

The study context was an undergraduate LA elective course in the Ball State University during the Spring 2010 semester. The students who enrolled in the course were 12. Eleven of them were LA majors and one non-major. Three were females and nine were males. The participants were novices to gaming and game design. Therefore, they needed adequate time to get acquainted

with games and design. The first five weeks were spent on defining and analyzing games. Then they learned the necessary knowledge to apply to the end-of-the-semester project. The project was developed in two phases; the first was to design an educational non-digital game about a topic in LA, while the second was to start creating them using game development tools. All done in groups; in part one students were divided in 4 groups; each group consisted of three students. While in Part Two students were divided in two groups of six students.

The author concludes: "this first-time LA game design elective proved successful on multiple levels." This well-designed instruction, explained in-detail in this study, represents "a significant pedagogical shift in a traditional LA curriculum." Collaborative exploration and discovery led to creative and functional project-based problem solving.

Mixed Mode. Following a mixed mode approach, the questions that Charlier & De Fraine (2012) clearly addressed looked for "whether a technological learning experience using the concept of DGBL in pre-service teacher training programs: (i) can help students understand and experience on DGBL can contribute to teaching and learning, (ii) can improve students' self-confidence for technology use, and (iii) motivates student teachers to use DGBL for instructional purposes."

Their strategy was to design a workshop on DGBL for a one-year pre-service teacher training program; a master's in health science education. Their participants were 32 graduate students who had a background in bio-medical, sports, or science. All of them showed a tech-savviness level adequate to the requirements of the program. In addition, some of them were gamers. However, none of them was a game developer.

Collecting data through many research instruments such as; a group interview that included all participants, a questionnaire given to all participants, and a focus group with a smaller sample

withdrawn from the original sample. The quantitative data were analyzed statistically, and the qualitative data were analyzed by identifying themes, including participants concerns and personal interpretations and reflections.

The results of the questionnaire showed that 89% of the participants agreed that all teacher-training programs should use DGBL, and 84% believe that DGBL should be missing in the teacher training programs. Those who disagreed were enrolled in a sports program and pointed that using DGBL may be useful in teaching theoretical concepts on sports. The also provided that the use of games that involve movement, such as those that run-on Nintendo Wii® are useful for training on sports activities. In the questionnaire, 83% of the participants indicated that they would use DGBL in their teaching. A very important result implies that according to the findings of the projects the participants conducted in their authentic teaching environments, 89% of them agreed that their young learners were motivated into learning by using digital games.

Similarly, two hypotheses for the first experiment of Doman et al. (2015) were stated:

> "H1: Students who used GameMaker® in their Computer Science class would show improved attitudes toward computer science compared to students who did not use GameMaker®.
>
> H2: Students would evaluate the instructor more favorably when GameMaker® was used to teach computer programming concepts compared to students whose instructor did not use GameMaker®."

While for their second experiment, they had two other hypotheses:

> "H1: Students who used GameMaker® in their Computer Science class would have more positive attitudes toward computer science than students not exposed to GameMaker® approximately two years following completion of the course.
>
> H2: Students who used GameMaker® in their Computer Science class would evaluate the use of GameMaker®, the course, and the instructor more positively than would students not exposed to GameMaker® approximately two years following completion of the course."

Fifteen sections of an introductory course of computer science were selected to participate in the study. Eight of them included GameMaker® in their instruction, while the remaining seven

had their traditional instruction and did not include GameMaker®. Students were surveyed and their qualitative perceptions and teaching evaluations all were instruments for this part of the research. Less participation in the second study occurred because participants of the first study were invited to participate voluntarily in the second part of the study. The same survey used in the first study was used in the second study. In addition to that, another survey consisting of sixteen questions was developed by the researchers based on the results of the first study.

Adequate clear answers and proofs to their assumptions and conclusions were laid out in the article for all four questions. Though the authors reported there was no evidence to support the first hypothesis of their second experiment. The authors proved that learning game development improves students' attitudes and learning of computer science. In addition to all of that, the authors indicated that additional research is encouraged to take place in the direction of investigating the similarity between genders attitudes.

Summary

Doman et al. (2015) indicated that they "hypothesized that the positive attitudes toward computer science observed in the experimental participants from Experiment 1 would still exist two years after the class, but this hypothesis was not supported". This might be due to many reasons one can think of, but one of them might be that the authors did not take into consideration the importance of the subject matter itself to the students who took the course. It was not obvious if the course was a service course, where all students from all majors enroll in it, or if it was meant to be for students of computer science major. The researcher suggests that it might make a difference if students were coming from IT disciplines, then the subject matter will be related directly to their majors.

With almost the same goals Doman et al. (2015) aimed at, Ernst & Clark (2012) conducted their experiment by adding a supplemental unit to teach GameMaker® to a Career and Technical Education (CTE) course named Computer Applications taught at North Carolina Virtual Public Schools.

The described unit seems to be a well-designed instruction consisting of several learning objects including three tutorials, thus it creates a space for learners to construct their knowledge of game development. However, it had its limitations; that is according to the authors; "the high-end nature of the content being presented was difficult for some students", this means that the unit was distinct from the subjects of the courses they were incorporated in.

The authors also mentioned that students shared their work with teachers by sharing the EXE files, which are supposed to be the final product of the developed game, thus no one can view its contents and what objects and sprites a developer included in it. The researcher infers that there it might be that there was not enough technical guidance provided for students at the front end. That is students should have shared the resulting game file with the extension GM81 or GM6, depending on the GameMaker® application version, either GameMaker® 8.1 or GameMaker® 6. If they do so, the instructors will be able to view the design of the game, otherwise, they will not be able to evaluate students works.

If this unit was expanded to be a whole course delivered in a blended learning or a face-to-face setting, the mentioned obstacles would not be encountered. Likewise what Baytak et al. (2011) accomplished in their study, where all class session were face-to-face and the first and second authors attended all sessions, the first author was there for the technical issues and the second author was the nutrition expert.

Similar to Ernst & Clark (2012), the instruction Baytak et al. (2011) applied was part of a science class in a unit on nutrition and not a standalone class, the main task students accomplished was to build a game that teaches younger students concepts about nutrition, thus sixteen first graders tested the games when they were produced.

Further research and more evidence needed by Baytak et al. (2011) to conclude what they have concluded in their study. In comparison between what Doman et al. (2015) referred to and what Baytak et al. (2011) concluded about gender differences and similarities, the former did not expand on the remark since their study did not bring evidence on this issue. They only recommended further research on this notable remark. While the latter were more comfortable to elaborate on several remarks pertaining gender differences and similarities without hypothesizing on it from the first place.

Linking between complexity thinking and gaming, Patton (2011) suggests that strategic games, such as SimCity imitates the behavior of ants, that is, deterministic order and randomness, rather than chaotic behavior. Thus, he introduced what he refers to as his method of game design that uses the concepts of Move, Avoid, Release, and Contact (MARC), which provides a broad basis for connecting programmable video game units, and therefore, could be used to design, develop, critique, and evaluate games. It seems that this method can be used as a framework to follow by instructional designers and teachers who design game development courses.

Finally, and from what the authors concluded in their studies, the researcher recommends that it would be better to conduct a whole standalone course, than embedding a subject unit in a distinct course, in order to teach game development for any purpose. This could be accomplished through an instruction that is based on constructivist learning theory, where learners construct their

knowledge by experiencing designing and developing games, testing each other's games, and reflecting on their designs.

The lab workshops, where learners are given tutorial handouts to follow to build games, should be held in a face-to-face environment, with the presence of a Subject Matter Expert (SME) and a computer technician in case any technical issues occurred.

A final project is required to be submitted by the end of the course, perhaps to be presented to other students for testing and evaluation and evaluated by SME. In this project, the learner will demonstrate that they acquired the skills and the knowledge that is aimed by the instruction.

This instruction could be oriented to increase the understanding of computer programming and OOP, or it could be oriented to other subjects, such as demonstrating their understanding of different subjects, such as language grammar, by creating games that teach certain topics of the subject.

CHAPTER 3 METHODOLOGY

This chapter describes the research methodology. It will include the participants and context, research design, research instruments, procedures of data collection and techniques of data analysis. This research is going to be conducted to investigate the effect of using Game Development-Based Learning (GDBL) on IT undergraduates in Jordanian universities learning computer programming and OOP concepts. An instruction was designed for an eight-week-long online course to teach the participating students game development using GameMaker®. This instruction is going to be used as the treatment intervention of the experiment. The educational institute in which the study will be conducted in uses an open-source learning platform named Moodle® (www.moodle.org) as their Learning Management System (LMS) ("Moodle - Open-source learning platform | Moodle.org," 2017). This LMS will be used to deliver the instruction in an online course.

The following questions are addressed by the study:

1. What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to define the concepts of computer programming and OOP?
2. What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to program in a high-level language, such as C++?

Quantitative measures will be used in this study. Analyzing the quantitative data will indicate if the designed intervention, an instruction that is based on GDBL to teach game development using GameMaker®, will improve students learning computer programming and OOP concepts using C++ as a programming language.

To gain consent to conduct the study, the researcher contacted a faculty member who forwarded the researcher's petition for approval to conduct the study to the Dean of the Faculty IT

at YU. The Dean forwarded the petition to the Associate-Provost of Academic Affairs who approved the request. The Dean forwarded the decision to the Head of Computer Science department with approval. The Head of Computer Science department sent a letter of approval to the researcher (See Appendix D). Contact between the researcher and the faculty of that educational institute occurred through email messages.

Permissions from Wayne State University's Internal Review Board will be obtained before conducting this study (See Appendix A).

Participants and Context

The study is going to be conducted in a college of IT at an urban public university in the northern of Jordan. The language of instruction at this university is English language (See Appendix E). Students should be able to interpret the instruction material without the need of translation. The researcher anticipates that the majority of the participants are going to be undergraduate first-year students. The students who were enrolled in the computer science program in the first semester of the academic year 2017-2018. While the rest of the participants are going to be undergraduate students coming from other disciplines within the same college, such as Computer Information Systems or Computer Networks and Security, or from the College of Engineering, such as Computer Engineering.

Depending on the data obtained from the Ministry of Higher Education in Jordan (2017) about the numbers of admitted students to the computer science program at Yarmouk University, the researcher anticipates that the number of computer science freshmen of that university will be about 130 students coming from different areas of Jordan. These students will form the majority of the study's population. Adding to them the students who will come from other disciplines, which

will raise the number of students who may participate in the study to be about 250 students. The ages of the students range from 18-65.

It is required in the study plans of the undergraduate programs of the Faculty of IT and the Faculty of Engineering in Yarmouk University that students have to either take two programming courses; CS210: Object-Oriented Programming and its prerequisite CS110: Programming in a Selected Language, or only take CS110. Depending on the program they are enrolled in, the two courses are required by the majority of the targeted undergraduate programs while only CS110 is required by some.

Both courses use C++ as a programming language of topic. CS110 course covers the elementary level of computer programming of the computer science program, while the CS210 course provides students with knowledge and needed skills in order to design and develop object-oriented programs ("Study Plans - Faculty of IT of Yarmouk University," 2017).

The study will be conducted on the second semester 2017/2018 which begins in February 4th and ends in May 16th, 2018. An online course with the topic of Game Development will be open for registration before the beginning of the second semester 2017/2018.

The two courses are taught in a traditional face-to-face classroom setting, with the support of Moodle as a Learning Management System (LMS) for announcements, posting of grades, submission of assignments of the course and to deliver the course content and material. All sections of each course will have the same instruction and textbook to be taught. Each of the two courses are of 3-credit hours in addition to one lab for each course of one credit hour where the students meet a teaching assistant assigned to conduct workshops, so they practice what they learn about programming in C++ and OOP. Therefore, students of each section meet their instructor three times every week on the same campus and they meet a teaching assistant in the lab once a week

on the same campus. Students learning is evaluated formatively through two unified midterm exams and summatively through a unified final exam for each one of the two courses. All exams are computerized and will be administered in computer labs on campus for all sections on specific days of the semester. The first midterm exam is usually held between the 6th and 7th weeks of the semester, the second is held between the 12th and 13th, while the final is held in the 15th week of the semester.

Research Design

The researcher will adopt one of the most popular quasi-experimental designs; the Non-equivalent Control Group design. That is an experimental group and a control group will be asked to take a pre-test and a post-test, while only the experimental group is going to be exposed to the treatment (Creswell, 2013). Similar to any quasi-experimental design, participants in this study are not randomly selected, rather the experimental group members will register themselves in the treatment online course.

The researcher will facilitate an online course about game development for the participants in the study. The course will be hosted on Moodle; the LMS the university uses. This online course will be open for all students who are enrolled in CS110 and CS210. It will run concurrently with these two courses in which the participants will be enrolled. The students who will take the online course are considered the experimental group, while other participants of the study who do not take the online course are considered as the control group. The number of members of the control group will be decided depending on the number of members of the experimental group, i.e. the number of students who enrolled in the online course.

Convenience Sampling is used in this study, where students can select to register in the online course. This will ensure a random selection of participants. Students who register in the

online course will study the intervention instruction for eight weeks from the beginning of the second semester 2017/2018. For the purpose of the study, the intervention instruction will be available only for the students who are enrolled on the second semester of the academic year 2017/2018, either in CS210 or in CS110. This will open the floor for those who did not take CS110 on the first semester 2017/2018 with their cohort or those who did not pass with their cohort on the first semester 2017/2018 and are repeating CS110 to participate in the study. Therefore, the sample population of the study are going to be students from all sections of the two formal courses; CS210 and CS110.

At the beginning of the second semester of the academic year 2017-2018, the participants in the study will be asked to take a pre-test on GameMaker® and C++ Programming Language to capture some information about their knowledge about game development, programming languages and OOP (Two examples of the pre-test are shown in Appendices G and H). They will also be asked to take a demographic information survey (See Appendix F). The analysis of the demographic information will illustrate a better idea about the population sample and will help in determining who in the IT programs had benefited most from the experiment.

During the semester, the experimental group will learn about game development through the instruction that was designed by the researcher. The researcher will be the facilitator of the online course. The other group of learners, considered to be the control group, will not be exposed to the game development course during the conduct of the study, but they will be given the chance to learn about the game development after the conduct of the study in the remaining seven weeks of the second semester. A course of game development will be provided in the summer semester of 2017-2018 to give the opportunity to other students to learn more about game development.

Moreover, the Faculty of IT at Yarmouk University are considering creating a game development course in cooperation with the researcher to be offered in later semesters.

The researcher, in cooperation with the instructors of the CS210 and CS110 courses, will test students' learning of computer programming concepts in a high-level programming language, namely C++. The first midterm exam in both courses is the instrument to be used for this purpose. Where measurement of learning outcomes will indicate students' improved learning. In the first midterm, the questions will measure the students' ability to define the concepts of computer programming and OOP and will also measure the students' ability to program in a high-level programming language, namely C++.

At the end of the 8th week of the semester, members of the two groups will be asked to respond to a post-test (See Appendix I) to investigate the effect of the learning game-development on the students learning of the concepts of OOP and computer programming with C++ (An example could be seen in Appendix G obtained from (Kunkle & Allen, 2016)). The measurement of the learning outcomes of the course will also be used to indicate how the treatment improved students' learning. The control group will have the chance to learn game development after the end of eight weeks of the study. Figure 3-1 shows the research design.

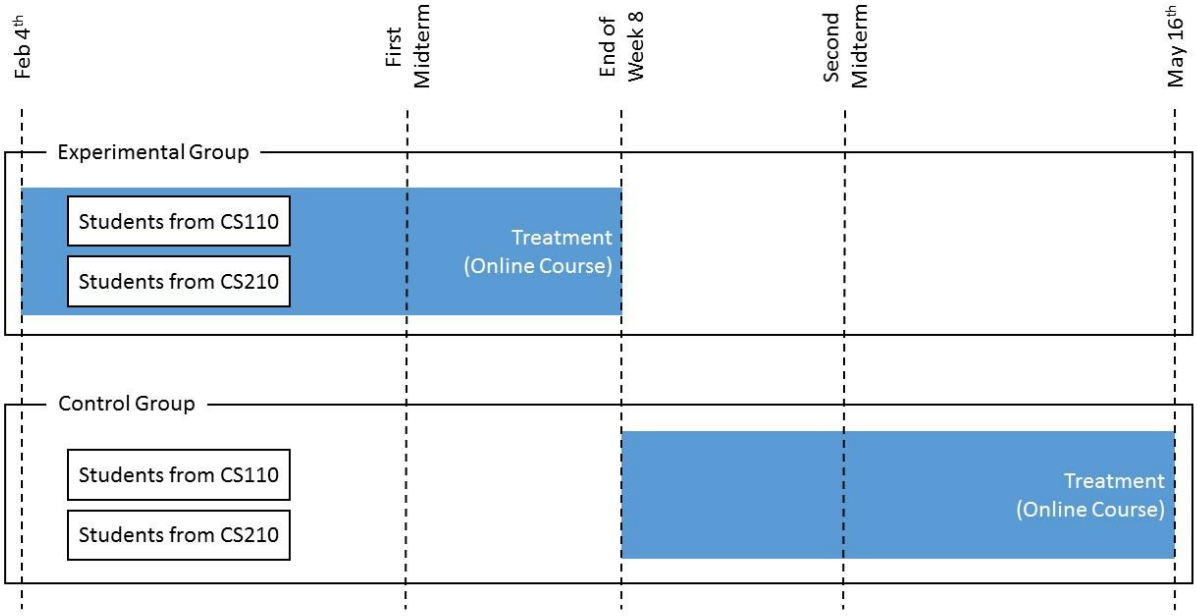


Figure 3-1: Non-Equivalent Pre-test Post-Test Control Group Design

This experimental study will investigate the effect of GDBL; the independent variable, on the ability of students of IT disciplines to define the concepts of computer programming and OOP; one dependent variable, and their ability to program in C++; another dependent variable. Table 3-1 provides a detailed research design outline.

Table 3-1: Research Design Outline

Research Question	Data collection Methods, Resources & Instruments	Dependent Variable	Independent Variable	Sample/ Participants & Contexts	Method(s)	Data Analyses
Q1: What is the effect of GDBL on the	- First midterm exam.	Students' ability to define the	GDBL Instruction	Students enrolled in CS210 and	Quantitative	Statistical Analysis:

ability of IT undergraduates in Jordanian universities to define the concepts of computer programming and OOP?	(Questions aim to measure the students' ability to define the concepts of computer programming and OOP). - Pre-test and Post-test	concepts of computer programming and OOP concepts		CS110 on the second semester 2017/2018 at Yarmouk University; a public university in the northern of Jordan.		Unpaired t-test
Q2: What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to program in a high-level	- First midterm exam. (Questions aim to measure the students' ability to program in a high-level	Students' ability to program in a high-level language (such as C++ or Java)?	GDBL Instruction	Students enrolled in CS210 and CS110 on the second semester 2017/2018 at Yarmouk University;	Quantitative	Statistical Analysis: Unpaired t-test

language, such as C++?	programming language, specifically C++). - Pre-test and Post-test			a public university in the northern of Jordan.		

The Intervention, the Game Development Online Course:

The researcher will facilitate an 8-week online course of the topic Game Development. This course will be hosted on Moodle; the LMS that is used in the educational institute in which the study is going to be conducted. At the beginning of each week, students will view a 20-minutes video created by the researcher that will demonstrate new skills. Then they will get to work on workshop handouts that are going to guide them in a systematic manner in creating games. The earlier in the semester the more detailed a handout becomes, and the later in the semester the less detailed the handout becomes. This will give the opportunity to predict the best approaches to solve new problems and come up with their own solutions to certain problems. Table 3-2 shows the course weekly schedule.

Table 3-2: Game Development Course Weekly Schedule

Week	**Topic**	**Assignments**
1	• Introduction to GameMaker® • Installing and Discovering GameMaker® 8.1	• Description of GameMaker® 8.1 environment
2	• Your First Game: Devilishly Easy	• ClickBall Game Tutorial - V1 • ClickBall Game Tutorial - V2

3	• More Actions • Target the Player	• Space Invaders Game Tutorial – V1 • Space Invaders Game Tutorial – V2 • Space Invaders Game Tutorial – V3
4	• Fixing problems. • Getting it together.	• Normal Distribution Curve Game
5	• Design your own game	• Lumosity Game
6	• Fixing problems. • Getting it together.	• Mind Reader Game
7	• Fixing problems. • Getting it together.	• Memory Game
8	• Become a Programmer (Introduction to programming in GML)	• Reflection on Learning

The "Assignments" column contains the tutorial handouts to be finished in that week. The products that come out of those handouts are to be submitted as homework assignments to be graded. The researcher will be grading the assignments for the students. Students will earn points instead of grades to give them the impression that this course will not affect their grades in their formal education courses; CS110 or CS210.

CHAPTER 4 RESULTS

This chapter outlines the findings from the Game-Based Development Learning (GDBL) Experiment on Yarmouk University's Information Technology undergraduates' performance on C++ and Object-Oriented Programming courses. The structure of the chapter is as follows. First, a descriptive note on the dependent variable and group distributions on such measures is provided. Second, a cursory inspection of whether gender or area type, urban or rural, is related to performance on educational achievements in programming courses, first, second and final exams (See Appendices J and K). Third, a One-Way Analysis of Variance is carried out to evaluate the effectiveness of the game development concurrent course on the performance of students in their regular examinations in C++ and Object-Oriented Programming courses. Further, given the widespread assumption in the educational assessment's literature concluding that most distributions generated from exams' scores are not normal, a Kruskal-Wallis H Test, a non-parametric equivalent to One-Way ANOVA is performed and its findings presented.

This research investigates the effectiveness of an intervention, a Game Development Based Learning on Yarmouk University students' understanding and ability to program. The research design chosen for conducting the study was a quasi-experimental non-equivalent group design. The difference between this design and the classical experimental strategy is the absence of random assignment. This threatens the validity of findings by introducing confounding variables. It was unfeasible for the researcher to conduct a classical experiment given the amount of administrative approvals required. The sampling design to recruit participants was convenience, which was selected for its time and cost efficacies. The researcher only had access to one institution, Yarmouk University, and IT students enrolled at the various departments requiring programming courses were the population for the experiment. A total of 250 students were recruited and distributed into

three groups, one that fully participated in the online GDBL course, one that partially completed the course; 25% or less and a control group who did not participate in the course. To analyze the effectiveness of the intervention a One-Way Analysis of Variance was carried out. Given the increasing confirmation that educational assessment data, final exam scores serving as the post-measurement outcome, are non-normally distributed, a Kruskal-Wallis H Test, a non-parametric ranks-based equivalent test to ANOVA, was carried out. Results in both statistical procedures confirmed that GDBL improves the understanding and ability in programming courses for undergraduate students in Jordan.

Summary of Main Findings

RQ1 *What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to define the concepts of computer programming and OOP?*

The mean of students who fully participated in the GDBL online course was significantly higher compared to the mean of students who did not participate in the experiment. The difference in the C++ course was 5 and in the OOP was 20. It was not feasible to discern means' differences per question, skill, ability or certain knowledge domains within each course, nevertheless, the difference is likely to reflect an improved understanding of the basic concepts of programming in both courses.

RQ2 *What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to program in a high-level language, such as C++?*

The enhanced performance of students who fully participated in the experiment compared to those who did not reflects a statistically significant positive effect on GDBL on students' ability to program. The final examinations in both courses, (see Appendix K), contain a multitude of questions requiring students to recall, put to use and evaluate their ability to program in various

capacities. Therefore, this research confirms earlier findings that GDBL interventions improve students' abilities to program and their overall positive attitudes and study behaviors towards the undergraduate computer science curriculum.

Distribution Characteristics of Programming Exam Scores in Yarmouk University

The main outcome of this research is student's performance on two final exams, C++ and Object-Oriented Programming in the Second Semester of the academic year 2017-2018 at Yarmouk University in Irbid, Jordan. Figures 4-1 and 4-2 present the distribution of scores for the three main groups who participated in the experiment, those who did not participate at all In the online course, those who partially participated, completed the survey and less than 25% of the course and those who completed the course, 50% or more completion rate was included in this group. Notice that all three groups on both exams, the final exam in C++ and the transformed score on the second exam in Object-Oriented Programming course (see Appendix K), do not exhibit normal distributions. Table 4-1 confirms this conclusion by presenting skewness and kurtosis statistics on the distributions noting slight negative skewness for the two variables. While such skewness may be expected from the data generating process, some may argue that programming exams will have lower scores producing negative skewness, this research utilizes Kruskal-Wallis H Test, presented below, to free the analysis from any distributional assumptions since such a test is non-parametric.

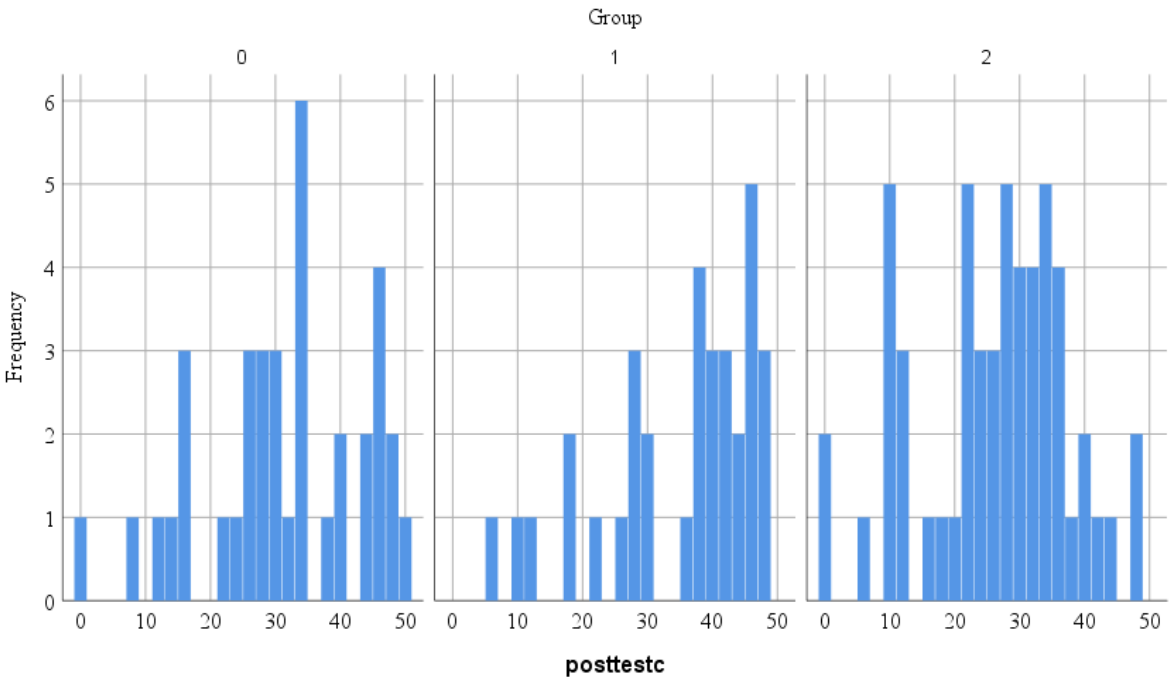


Figure 4-1: Groups Scores on C++ Final Exam

Notes: posttestc is the Final Exam Score, post-test measurement

Groups: 0= Partial Participation, 1= Full Participation and 2= No Participation

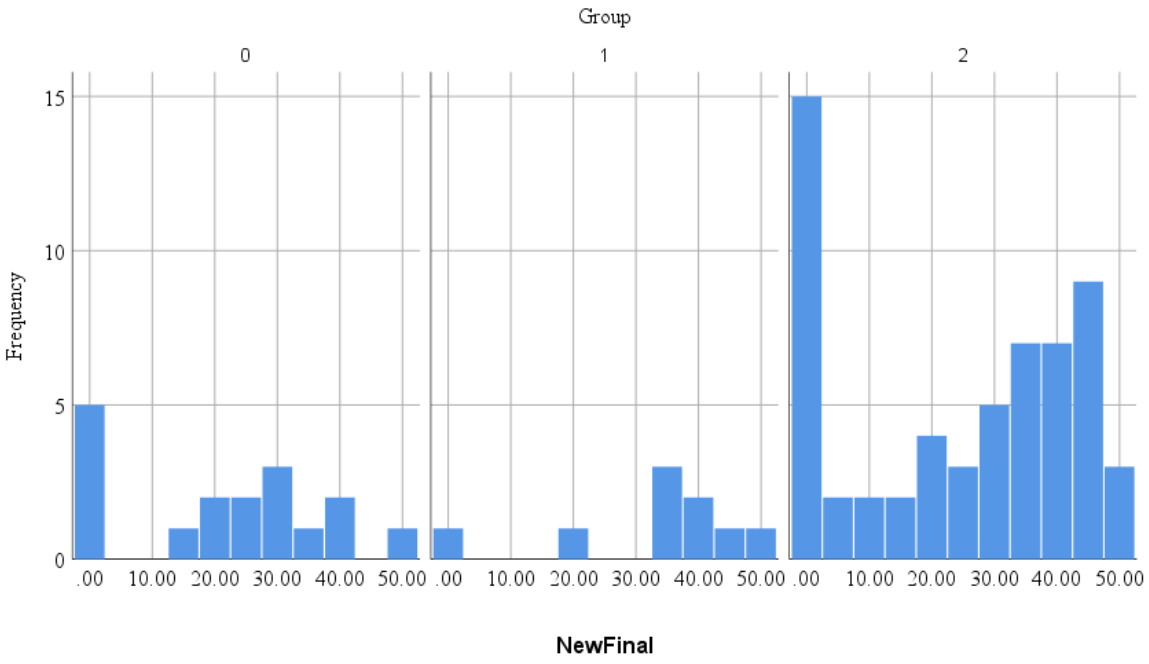


Figure 4-2: Groups Scores on OOP Second Exam * 2.5 (Second Appendix K)

Notes: NewFinal is the Second Exam Score * 2.5, post-test measurement

Groups: 0= Partial Participation, 1= Full Participation and 2= No Participation

Table 4-1: Distributional Characteristics for Pre and Post Measurements for C++ and OOP

Statistics					
		First Exam C++	Final Exam C++	First Exam OOP	Second Exam OOP * 2.5
N	Valid	140	124	104	101
	Missing	2	18	38	41
Skewness/		-.270	-.417	-.485	-.189
Std. Error of Skewness		.205	.217	.237	.240
Kurtosis		-.193	-.521	.230	-1.146
Std. Error of Kurtosis		.407	.431	.469	.476

To better understand the predictors of students' performance in programming courses in Yarmouk University of Jordan, information on high school GPA, area type residence and gender was collected. Figures 4-3 and 4-4 present a matrix scatter-plot conveying the bi-variate relationship between students' performance on the first exam and the final exam in C++ and Object-Oriented Programming (See Appendix J). Notice that High School GPA has a slight positive relationship with both sets of scores in the two courses. Nevertheless, this set of correlations is below 0.3 indicating a weak relationship. Both graphs also indicate a strong positive relationship between the first and final exams in the two courses, correlations are above 0.5. This relationship is stronger in the C++ course compared to the Object-Oriented Programming course. This difference may be explained by the number of students who scored 0 on the second exam, transformed second exam score variable due to a variety of reasons including failing to take the exam, withdrawing from the course or being barred from taking the exam at one point during the exam due to cheating or exam disruption.

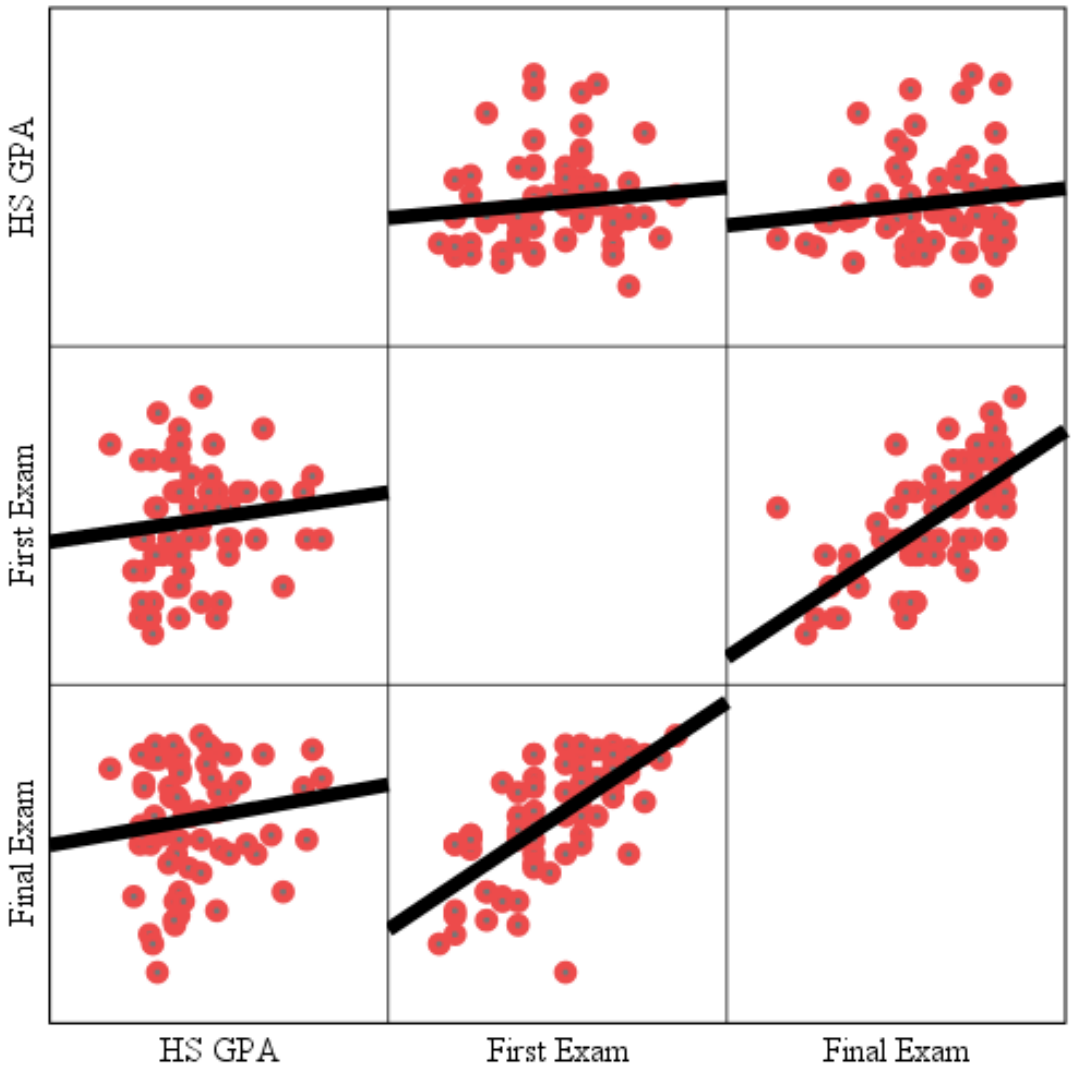


Figure 4-3: Correlations High School GPA and Student's Performance in C++ Exams

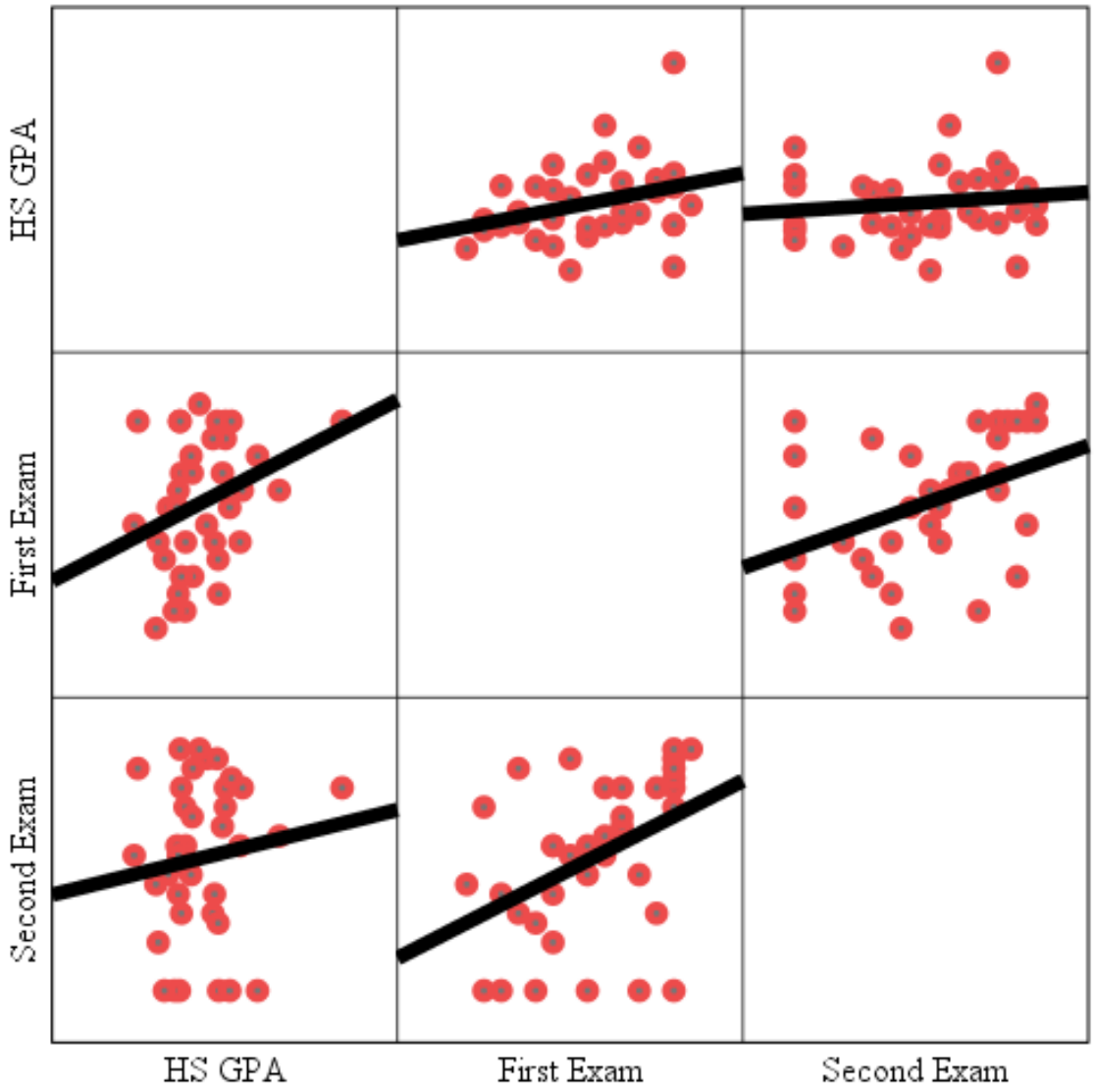


Figure 4-4: Correlations High School GPA and Student's Performance OOP Exams

Figures 4-5 and 4-6 present the means of all groups based on the two main outcomes, final exam scores on C++ and Object-Oriented Programming. The treatment group outperformed both control groups on both tests. This superiority is more accentuated in the Object-Oriented Language course more than the C++ course. This difference may be explained by many factors. First, Object-Oriented Programming is a higher tier programming courses compared to C++, the introductory programming course, which means that the students who made it to Object-Oriented are more receptive to programming training and courses. Further, more advanced students in programming

are likely to be more engaged in the game development course, as it was the case in this experiment gaining better knowledge, skills and abilities to score higher. While such difference may be due to other factors, it is clear from the descriptive evidence that participation in game development course fully improved test scores for students in C++ and Object-Oriented languages.

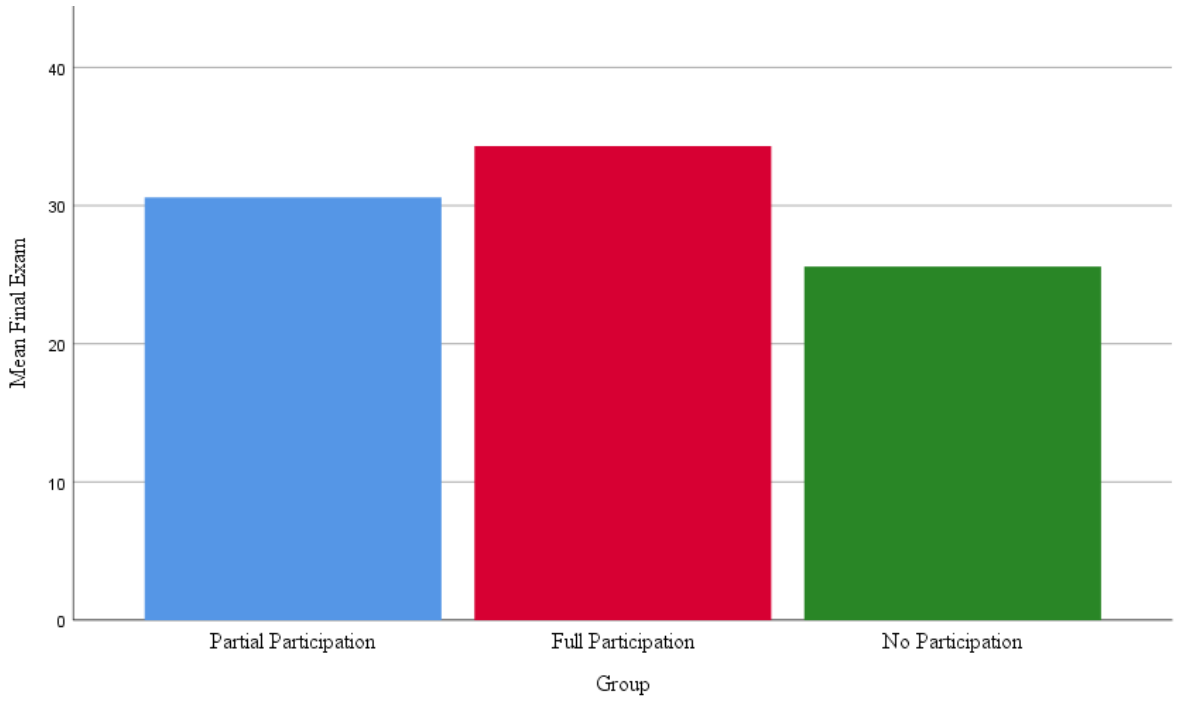


Figure 4-5: Means of Groups on + C++ Final Exam

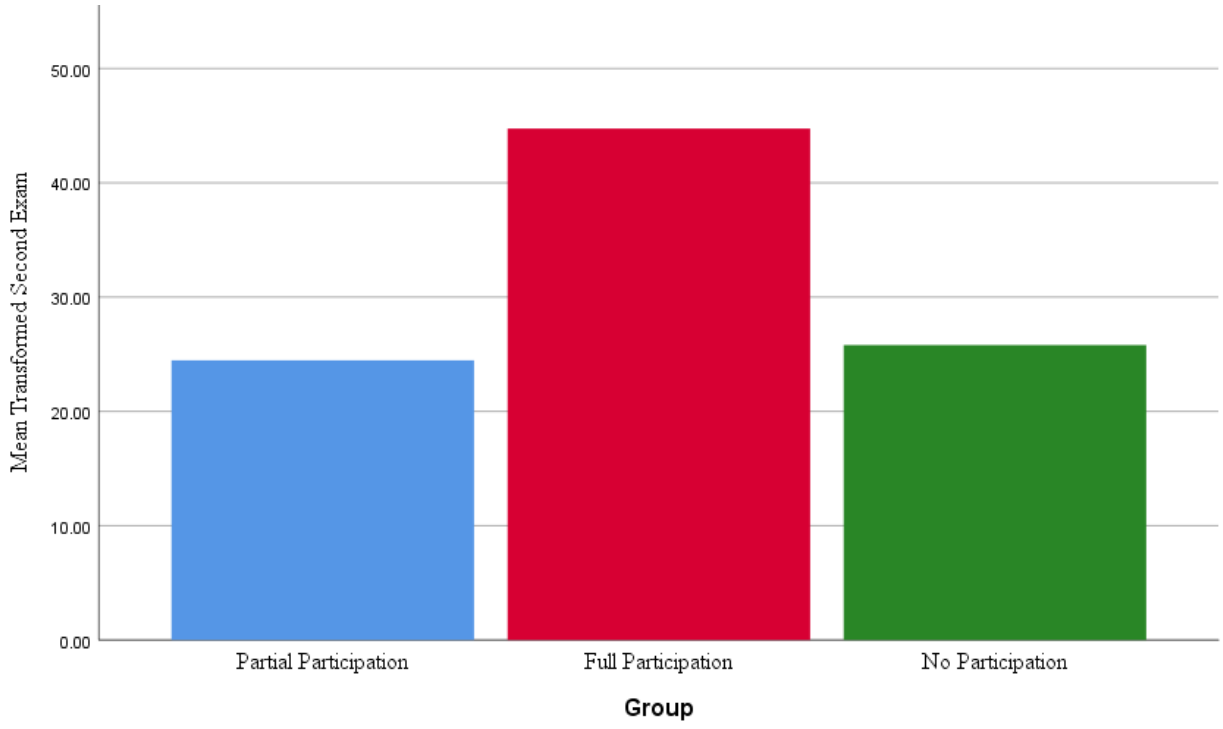


Figure 4-6: Means of all Groups on OOP Transformed Second Exam

Table 4-2 presents the means of pre and post-measurements by gender and residence area type. Notice that females outperformed males on the first, pre-measurement, and post-measurement, final exam, in the C++ course. There was no significant discerning difference between rural and urban students with respect to means of C++ exams. Further, female students outperformed their male counterparts on the transformed second exam score in the Object-Oriented Programming course. There was a slight difference in means between urban and rural students when it came to the means of Object-Oriented exam scores. The females' scores advantage could be explained by a variety of reasons. First, females in Jordan have more restricted socialization contexts compared to males who spend a significantly higher amount of time on social activities. This may translate into more studying time for females compared to males resulting in better performance. Further, much research has confirmed that female students outperform male students in higher education.

Table 4-2: Permeance of Yarmouk University Students in C++ and Object-Oriented Programming Courses in the Second Semester 2017-2018 by Gender and Area Type

Assessment	First Exam C++ (Pre-test)	Final Exam C++ (Post-Test)	First Exam OOP (OOP)	Second Exam OOP (Post-Test)
Male	10.26 (92)	27.10 (78)	12.52 (58)	27.92 (83)
Female	12.13 (48)	33.30 (46)	12.65 (48)	34.75 (18)
Urban	10.79 (42)	29.13 (38)	14.19 (16)	30.87 (17)
Rural	11.17 (98)	30.03 (86)	12.28 (88)	28.75 (84)

N = the number between the parentheses

Note that OOP Second Exam original score was multiplied by 2.5 to make it out of 50 to ease comparisons.

Tables 4-3 and 4-4 present the sample size, mean and standard deviation of all three groups on the pre and post-measurement for the two courses. Notice that the group who fully participated in the online course outperformed both other groups. Further, the group who expressed interest and joined the course, but never completed it, outperformed the group that did not participate in the treatment. This evidence confirms the descriptive results presented above that participation in the game development course improved students' ability in programming courses both in C++ and Object-Oriented programming.

Table 4-3: Performance in Pre-test and Post-test Assessments for C++ Groups

Group		First Exam	Final Exam
Partial Participation	Mean	11.32	30.59
	N	41	37
	Std. Deviation	3.467	12.417
Full Participation	Mean	12.64	34.30
	N	33	33
	Std. Deviation	4.022	11.876
No Participation	Mean	9.77	25.59
	N	66	54
	Std. Deviation	4.285	11.331
Total	Mean	10.90	29.40
	N	140	124
	Std. Deviation	4.140	12.263

Table 4-4: Performance in Pre-test and Post-test Assessments for OOP Groups

Group	First Exam	Second Exam * 2.5

Partial Participation	Mean	12.95	24.47
	N	20	19
	Std. Deviation	5.052	7.664
Full Participation	Mean	15.06	44.72
	N	18	19
	Std. Deviation	3.298	6.341
No Participation	Mean	11.79	25.80
	N	66	63
	Std. Deviation	3.936	7.634
Total	Mean	12.58	29.10
	N	104	101
	Std. Deviation	4.217	7.946

Analysis of Variance Results

A One-Way Analysis of Variance was carried out to assess the effect of participating in the game development online course on students' performance on the C++ programming final exam. Results indicate that participating in the game development course is statistically significant in explaining variance in final exam scores for the C++ course, F= 5.84 (2), P= 0.004. Table 4-5 presents a post-hoc test result confirming the difference hypothesis. Notice that not all groups' differences were statistically significant, however. Fully participating students outperformed those who did not participate at all, and the difference in means was statistically significant. There was no statistical significance between the means of those who partially or fully participated in the online course. All in all, participating fully in the game development course improved students' performance on C++ exam.

Table 4-5: Analysis of Variance Results for C++ Final Exam by Group

ANOVA					
Final Exam					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1628.913	2	814.457	5.842	.004
Within Groups	16868.926	121	139.413		
Total	18497.839	123			

Table 4-6: ANOVA Means' Comparison for Groups on C++ Final Exam

Multiple Comparisons						
Dependent Variable: Final Exam						
Tukey HSD						
Group	Group	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Partial Participation (0)	1	-3.708	2.827	.391	-10.42	3.00
	2	5.002	2.520	.120	-.98	10.98
Full Participation (1)	0	3.708	2.827	.391	-3.00	10.42
	2	8.710*	2.609	.003	2.52	14.90
No Participation (2)	0	-5.002	2.520	.120	-10.98	.98
	1	-8.710*	2.609	.003	-14.90	-2.52
*. The mean difference is significant at the 0.05 level.						

Table 4-7 presents the results from a One-Way Analysis of Variance with transformed second exam scores for Object-Oriented Programming as the dependent variable and whether

students participated in the online course as the independent variable. Results indicate that participating in the game development course improves students' performance on the exam, F= 8.34 (2), P= 0.001. Table 4-8 presents a post-hoc comparison test to verify whether this statistical significance holds across the three groups. The difference in means holds for students who fully participated in the experiment when compared to the other two groups. Nevertheless, the comparison between those who partially participated and those who did not participate at all is not statistically significant. Results of the Analysis of Variance supports the overarching conclusion that participating in game development courses improve students' abilities, skills and knowledge in programming languages.

Table 4-7: Analysis of Variance Results for OOP Second Exam by Group

ANOVA					
Transformed Second Exam OOP					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5741.063	2	2870.532	8.344	.000
Within Groups	33716.239	98	344.043		
Total	39457.302	100			

Table 4-8: Post-Hoc Comparison Between Groups

Multiple Comparisons						
Dependent Variable: Transformed Second Exam						
Tukey HSD						
Group	Group	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
	1	-20.26316*	6.01790	.003	-34.5848	-5.9415

Partial Participation (0)	2	-1.31997	4.85474	.960	-12.8735	10.2336
Full Participation (1)	0	20.26316*	6.01790	.003	5.9415	34.5848
	2	18.94319*	4.85474	.001	7.3897	30.4967
No Participation (2)	0	1.31997	4.85474	.960	-10.2336	12.8735
	1	-18.94319*	4.85474	.001	-30.4967	-7.3897
*. The mean difference is significant at the 0.05 level.						

Kruskal-Wallis H Test Results

Many have argued that educational data is generated from non-normal data generating processes, therefore a verification of the ANOVA results above is conducted by performing the ranks-based Kruskal-Wallis H Test. Table 4-9 presents results from the test and confirm the findings above. Participating in the online experiment is a statistically significant predictor of performance on the C++ performance exam. The results indicate that means' differences are statistically significant with X^2 (2), $P = 0.003$.

Table 4-9: Kruskal-Wallis H Test C++

Ranks			
	Group	N	Mean Rank
Final Exam C++	Partial Participation	37	65.99
	Full Participation	33	77.70
	No Participation	54	50.82

	Total	124	

Test Statistics	
	Final Exam C++
Kruskal-Wallis H	11.960
df	2
P-Value	.003
a. Kruskal Wallis Test	
b. Grouping Variable: Group	

Table 4-10 presents the Kruskal-Wallis H Test results finding that group differences are statistically significant confirming the ANOVA results above. Findings suggest that participating in the online game development course improves students' performance on the final exam in Object Oriented Programming. The test, however, does not provide details on the differences between groups, $X^2 = 15.67$ (2), P= 0.002. All in all, both the non-parametric ranks-based test and the parametric test concluded that the intervention, game development course, improved students' performance in programming courses at Yarmouk university in Jordan.

Table 4-10: Kruskal-Wallis H Test for Object-Oriented Programming

Ranks			
	Group	N	Mean Rank
Transformed Second Exam Score	Partial Participation	19	43.00
	Full Participation	19	74.71
	No Participation	63	46.26
	Total	101	

Test Statistics	
	Transformed Second Exam
Kruskal-Wallis H	15.673
df	2
P-Value	.000
a. Kruskal Wallis Test	
b. Grouping Variable: Group	

To better assess the degree to which game development interventions influence students' performance in programming courses in Jordan, Cohen *d* effect sizes were calculated using automated tools for effect size calculations available online. Results indicate that the intervention has a medium effect on C++ performance, effect sizes around 0.5. On the other hand, the intervention seems to have a large effect on performance in Object Oriented programming, effect sizes around 0.8. In the following chapter, more detailed discussion on how to utilize game development in future courses to ameliorate students' performance in courses is provided

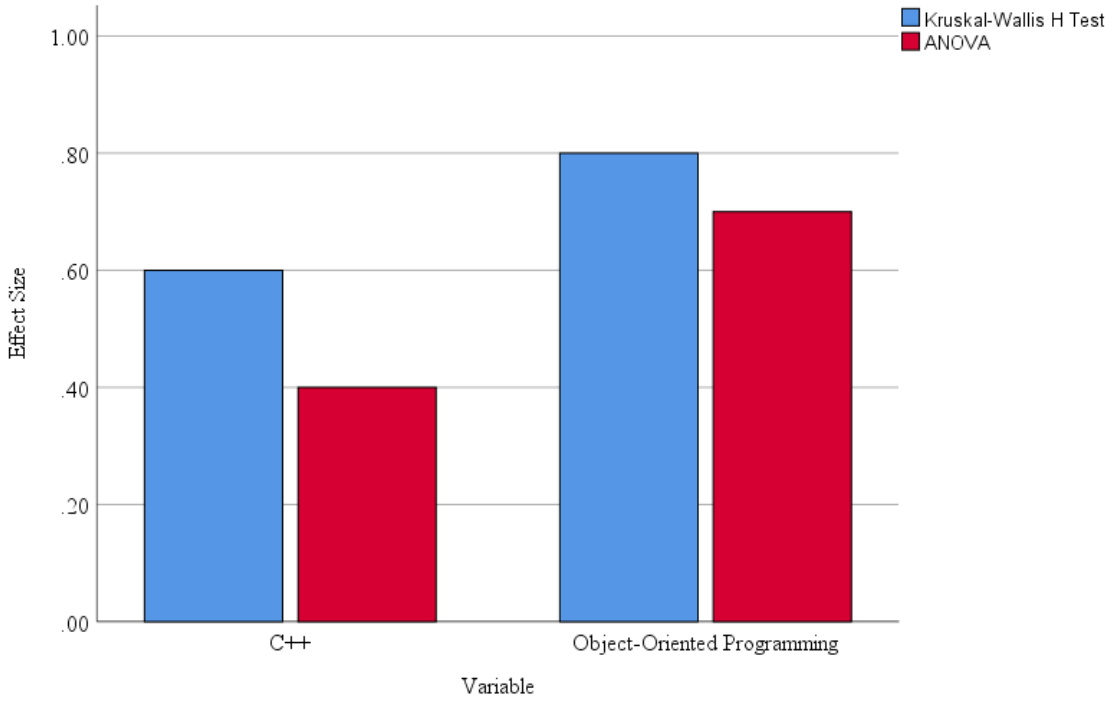


Figure 4-7: Effect Size for ANOVA and Kruskal-Wallis H Test

Figures 4-8 and 4-9 summarize the conclusion of this project suggesting a positive effect of online game development-based learning on programming skills, abilities and knowledge for undergraduates measured in their test performance on C++ and Object-Oriented Programming. The effect is stronger on Object-Oriented as can be seen in Figure 4-8 compared to C++, Figure 4-7. In both courses, students who participated fully clearly indicate an improvement in their scores on the post-measurement. For groups who did not participate at all, control group, the difference in C++ is abysmal. On the other hand, this difference is large in Object-Oriented Programming due to the large number of this group receiving a score of 0 for a list of potential reasons discussed above. Similarly, students who partially participated in the experiment reflects a small insignificant improvement on their C++ performance and many of them received a score of 0 on the post-measurement in Object-Oriented Programming driving the average, whether measured in mean or

median, down. In sum, online game-based learning enhances students' performance in programming courses in underrate information technology programs in Jordan.

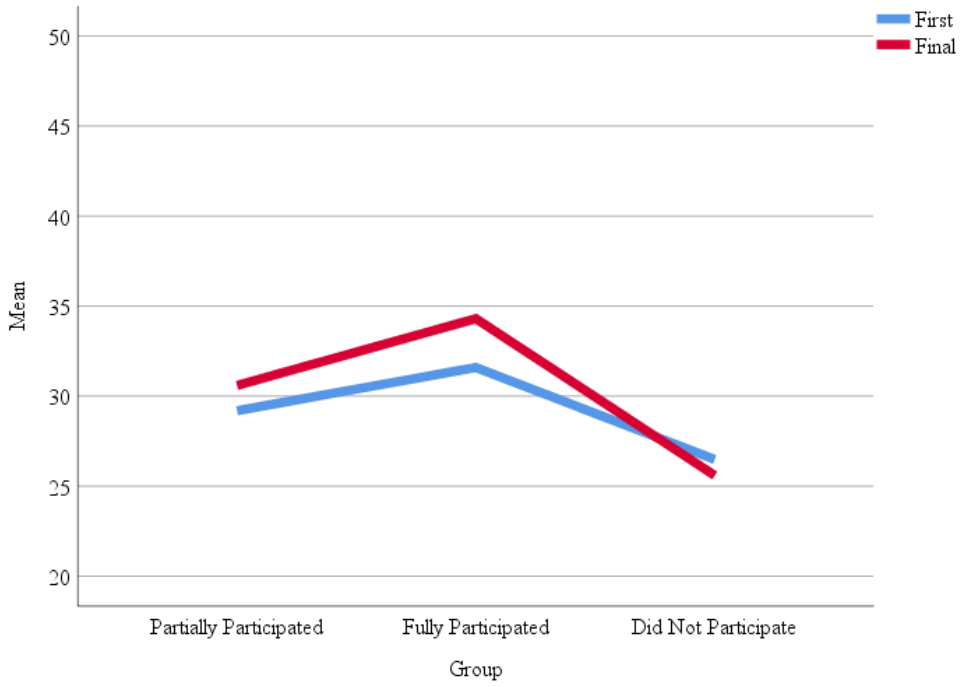


Figure 4-8: Comparison Pre-Test and Post-Test Results for C++ Students

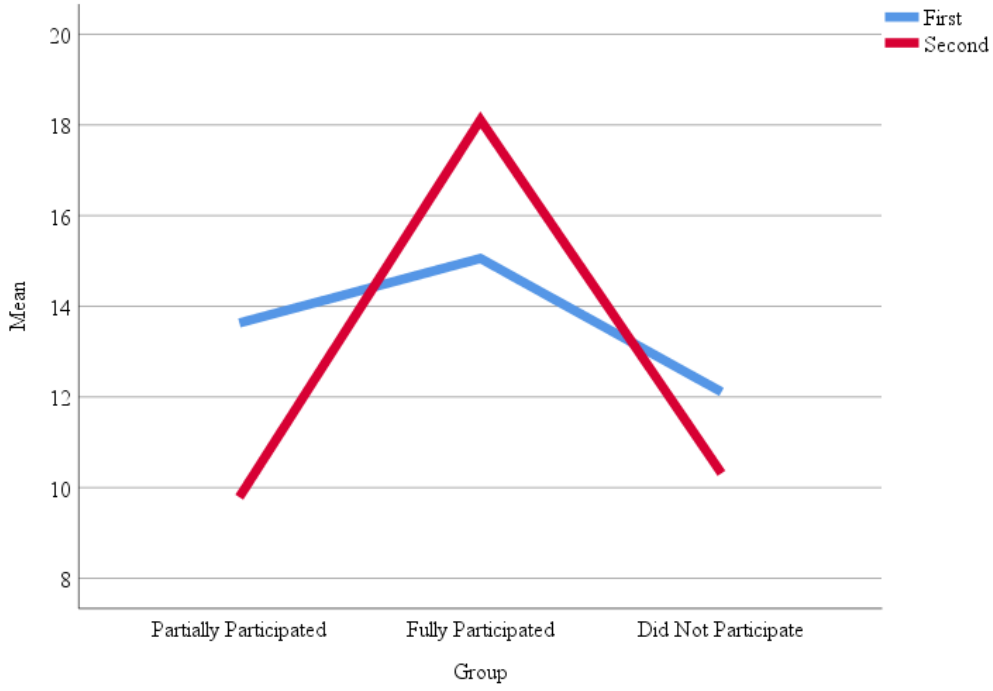


Figure 4-9: Comparison Pre-Test and Post-Test Results for Objected-Oriented Students

Linking Statistical Findings to Research Questions

The effect of GDBL on students' ability to define concepts related to C++ and OOP in Jordanian universities was found to be positive and significant. This effect is stronger in OOP compared to C++. For C++, the mean's difference between those who fully participated in GDBL and those who did not was 5 points on the final examination, which was out of 50. This represents an increase of 10% of the total grade on the final exam. The means' difference between those who fully participated in the GDBL in OOP course and those who did not in the final exam score, transformed second exam score, was 20 points out of a total of 50. This represents a 40% increase. Notice that every question on the final exam in both courses required substantial understanding of programming concepts. This suggests that participation fully in the GDBL intervention improved students' performance across the board concerning programming concepts.

Students' performance on final examinations in C++ and OOP reflects their overall ability in programming. While abilities differ with respect to the specific domain of a programming language, this study was interested in the effect of GDBL on overall programming ability. Questions on the final examinations in both courses incorporated a number of ability dimensions (see Appendix K). Those includes the creation of new codes, evaluation of existing codes and solving problems with presented codes. The effect of GDBL on ability on both courses was found to be significant and positive. This effect is equivalent to that of defining concepts. This is true given the necessary marriage between understanding defining features of programming language and the ability to program in that specific language. If a student understands basic concepts of a language, his ability will be tantamount to his level of understanding. If one is high, the other is likely to be high as well. Therefore, the magnitude observed in GDBL effect referenced above is consistent on both realms: understanding concepts and ability to program.

This effect is robust given the fact that students participating in the experiment exhibits similar demographic and cognitive characteristics. Most students participating in the experiment were between 18 and 20, Jordanian citizens, largely from public high schools and were examined with the government national examinations upon their completion of senior year in high school. Most subjects were placed by the government's matching system for high school graduates based on high school GPA performance and students' major top choices. This indicates that the effect of GDBL is consistent across different conditions and remains significant.

Summary of Chapter Four

This chapter outlined the findings of an experiment testing the effect of a Game Development Based Learning course on students' ability to understand concepts of programming and skill to program in C++ and OOP at Yarmouk University in Irbid, Jordan. Results of an Analysis of Variance and Kruskal-Wallis H Test indicated that the GDBL experience is effective in improving students' understanding and ability in programming courses. This effect was stronger in OOP compared to C++ as it appears on the effect size for both tests. The analysis also indicated that the results are invariant with respect to demographic and cognitive characteristics defining the students' population in Jordan.

CHAPTER 5 Conclusions, Discussion and Recommendations

This chapter presents the main findings of the experiment conducted by the researcher at Yarmouk University investigating the effect of Game Development Based Learning on students' understanding and ability in programming courses. The chapter also outlines several limitations faced by the researcher in accessing the research site, useful information and conducting the experiment. Father, a number of suggested future research directions is made along several recommendations to future researchers and stakeholders on maximizing the experience of Game Development Based Learning to improve students' performance in computer science courses.

This research documents the findings of an experiment testing the effect of GDBL on students' performance in programming courses in Jordan. Three groups were assessed based on two exams for two courses, C++ and OOP, the first and final exams for C++ and the first and second exams for OOP. Note that the first and second exams in both courses were out of 20 and the final exam was out of 50. Transforming the first and second exams by multiplying them by 2.5 facilitates comparative ability and ease the evaluation of the intervention, participating in an eight weeks voluntary course using GameMaker® in both courses. Results from the Analysis of Variance and Kruskal-Wallis H Test concluded that GDBL improves students' understanding and ability in programming courses. This finding is in line with previous research as outlined below.

Conclusions

RQ1 *What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to define the concepts of computer programming and OOP?*

While specific information concerning the breakdown of students' grades in their examinations during the experiment could not be obtained due to administrative and faculty failure to furnish such data for personal reasons despite the university's approval of the experiment, this

research concludes that GDBL enhances understanding, evaluation and application of programming concepts among learners in Jordanian universities. Evidence presented in Figures 4-8 and 4-9 clearly suggests that fully participating in GDBL experiences increase students' final scores significantly signaling to the fact that students' mastery of programming concepts is ameliorated through their GDBL experience.

RQ2 *What is the effect of GDBL on the ability of IT undergraduates in Jordanian universities to program in a high-level language, such as C++?*

Knowing that the final exam in C++ and Object-Oriented Programming feature questions on creating, modifying and evaluating language codes for achieving a desired end, this experiment clearly confirms the evidence that GDBL improves students' ability in programming. Figures 4-8 and 4-9, results from the Analysis of Variance, as well as the Kruskal-Wallis H Test presented in Chapter 4 all indicate an improvement in students' ability due to fully participating in GDBL online experience. While students' grades for direct programming ability-related questions are unavailable, all final exam questions tap into programming ability (see Appendix K). Therefore, an overall improvement in the total exam score is likely to reflect a holistic improvement in student' ability to program in high level languages including C++ and Object-Oriented Programming.

Discussion

The findings of this research confirm previous evidence of positive gains on programming skills in undergraduate curriculum using GameMaker®. Studies presented in Table 5-1 indicates that the use of GameMaker® in a variety of settings and across countries have consistently improved students' motivation, engagement, understanding and ability with respect to programming and computer science courses. This study contributes to the existing literature by

confirming the robust positive relationship between the use of GameMaker® and students' performance in programming courses in a new context, the Jordanian Information Technology setting.

Table 5-1: Literature Review Main Findings

Author	**Country**	**Main Findings**
Papastergiou (2009).	Greece	Gaming approach is better than non-gaming approach in improving students' understanding of computer-memory concepts. Gender was not a statistically significant variable in the experiment.
Hamari, Shernoff, Rowe, Coller, Asbell-Clarke & Edwards (2016).	The United States	Gaming was found to a good predictor of learning outcomes. The challenge of the game positively improved learning outcomes for participants across different types of games.
Eagle & Barnes (2009)	The United States	Assignments conducted through gaming environments improve the attitudes and ability of students in introductory computer science courses. Students who perform programming assignments in a gaming environment spend less time and master the material more than students who program without a gaming environment. All students preferred to program in a gaming environment.

Doman, Sleigh & Garrison (2015)	The United States	The use of GameMaker® in computer science courses improve students' attitudes towards programming. It also improves their perceptions of the instructor on the short-term and the long-term.
Jenson & Droumeva (2015)	Canada	GameMaker® use in classrooms improved students' learning outcomes. This finding generalizes over high-school, as well as undergraduate students in multiple disciplines including mathematics and computer science.
Johnson (2017)	The United Kingdom	The use of GameMaker® in undergraduate computer science curriculum improved students' programming skills. Further, the incorporation of GameMaker® into programming courses improved students' engagement and enjoyment of programming-oriented courses.

The findings of this research celebrate the constructionist paradigm of education promoted by Papert and his advocates. Students learn better if they could translate theory into practice, textual instructions, manuals and notes, into programs fulfilling specified functions. This pushes instructors of mathematics and computer science, and to that matter all higher education disciplines, to utilize a constructionist pedagogy when teaching. For instance, a university professor teaching instructional design at a given college should have her students design course webpages, interactive syllabi, course shells on Learning Management Systems or any tangible hoped to be made by the professional of the specific field. This call is likely to increase the motivation, interest, engagement and probability of mastering desired course outcomes given any subject being taught in higher education.

This research also confirmed that the use of instructional of computer-based scaffolding in computer science courses, and to that matter in Science, Technology, Engineering and Mathematics STEM courses, has a significant positive effect on students' performance. A meta-analysis summarizing the effect of computer-based scaffolding found that "indicate that computer-based scaffolding showed a consistently positive ($\bar{g}$ = 0.46) effect on cognitive outcomes across various contexts of use, scaffolding characteristics, and levels of assessment" (Belland, Walker, Kim, & Lefler, 2017). Many programming courses, assignments and tasks require the execution of a strict rigid sequence of steps. If a student fails to perform one of those, the program fails to execute the assigned functions. Therefore, the instructor needs to aid students at each step throughout the process. One of the best ways is the use of pre-recorded videos, dynamic interactive feedback offered while students are taking the assignment, the individual support the instructor awards to students with more needs and the use of all such strategies in teaching. All such scaffolding techniques mediated by computers are capable of increasing students' performance.

This research has also confirmed a long-standing finding in the e-learning literature in higher education. High levels of instructor's responsiveness, interactivity and attention have been linked to better performance and satisfaction scores with courses. The instructor of the GameMaker® experiment, the researcher, has offered exceptional levels of accessibility, amicability and support for students making them desire to complete the course. Low levels of engagement with students, the use of traditional paper-pencil, catch board and only textual notes provision have been proven to be inferior to the use of multitude of instructional methods and instructor's interactivity levels throughout the course.

This research further confirms earlier studies concluding that stand-alone learning by doing methods are inferior to blended learning approaches. Structured instruction mediated by computers

aided by reliable and valid assessments facilitated by peers, as well as an interactive instructor outperform traditional, as well as exploratory ways of learning. Students need feedback, encouragement, guidance and direction. They are novice learners and hungry for learning by doing only if such process is linked to reliable, valid and timely feedback accompanied by constructive feedback.

The foregoing discussion alludes to a significant finding in the literature of teaching programming language using gaming tools. Students are better served with exploratory methods of learning elementary programming supported by an interactive, dynamic, motivational, transformative, knowledgeable and scaffolding advocate instructor. The adage concluding that an instructor "can make or break a class" applies to any higher education course especially programming. The existence of a confident instructor whose students' learning needs constitute the core of his teaching pedagogy significantly improves undergraduate understanding, ability and skill in any programming language taught.

Implications on the field of Learning Design and Technology

The study clearly distinguishes the notion of Game-Development-Based Learning (GDBL) first proposed by Wu & Wang (2012) from its larger umbrella the Game-Based Learning (GBL). Further research is needed to validate the subordinary of GDBL to GBL or probably define it as a standalone paradigm. In addition, the study advances further to identify GDBL as a new learning paradigm for learning designers to follow and implement when they design courses and learning experiences. These learning experiences not only could be for the field of computer programming and OOP, but for other fields in general. When this paradigm is implemented, learners become authors of the games they build, not only consumers of ready-made commercial games as Majgaard (2014) emphasized on in his study. One of the examples of implementing this paradigm

in non-programming courses is what Baytak et al. (2011) did in their study. They implemented GDBL by requiring learners to design educational games using GameMaker® to help younger learners learn better the material of a course about nutrition. Another example is what the researcher, who is a computer science educator, designed in one of his non-programming courses. The researcher included an extra credit assignment in a non-programming computer science course he taught at Wayne State University between August 2016 and December 2018; Algorithms Design and Analysis. The course only involved designing algorithms and analyzing them mathematically as a prior step to implementing code in any programming language. Therefore, coding was rarely required throughout the course. In this extra-credit assignment, students were asked to create an educational game using GameMaker® to help other learners learn one of the algorithms covered in the course. Not only they had to deeply understand the algorithm they are designing a game for, but they had to find ways to design and create a game with incentives and penalties that will eventually lead players to the goal of the game, which is the desired output or behavior of the algorithm itself.

As computer programming requires scaffolding to help learners get through learning experiences, GDBL facilitates a suitable learning environment with adequate scaffolding opportunities. If a learning experience embraces GDBL, learners will unknowingly acquire the necessary fundamental knowledge and skills to program before even they admit to a computer programming learning experience using any high-level language.

Implementing this paradigm in fields other than computer programming, supports targeting the metacognitive level of knowledge. That is, when game development is used to help learners become aware of not only what they have learned in their learning experience, but what they have

mastered to the level that they are able to design and develop games that will help others learn the same knowledge they have acquired in those learning experience.

Limitations

One of the most obvious limitations for the study was the inability of the researcher to obtain refined measurements for student's performance on C++ and Object-Oriented Programming examinations. While the researcher secured the final grades for experimental groups, the instructors of participating courses did not provide the researcher with the breakdown of grades based on assessments. Therefore, the researcher was unable to determine the effect of GameMaker® use on the different skills, abilities and knowledge associated with programming. The researcher originally aspired to determine the effect of GameMaker® use on three levels of students' performance: mastery of concepts, evaluation of codes and codes creation. The only available information provided by instructors was the final test scores allowing the researcher to test the effect of GameMaker® use on the overall score in both courses.

Another limitation of the study was the lack of supportive environment within the housing department where the experiment took place. The instructor was compelled to leave his home institution in the United States to fly to Irbid, Jordan and conduct the demographic information survey from there due to the low support levels exhibited by the department at Yarmouk University. Once the researcher landed in Jordan, a number of physical visits to the university were made and the experiment took place according to the design proposed in Chapter three. Nevertheless, the first day of the experiment scheduled was to be the first day of the Second Semester 2017-2018 and the actual start day was delayed to the third week of the semester.

Third, the attrition rate of the experiment was high. Many students agreed to participate in the online course, however, they failed to complete it. This can be explained by several factors

including the voluntary nature of the course, the absence of continued channels of communications between the researcher and the teaching instructors, the late start date of the course and students' busy schedules with 5 or more courses. This decreased the number of valid cases for experimental groups threatening the findings of the experiment.

Another noticeable limitation of the experiment was the researcher's observation of high performing students' bias. Many students who completed the experiment, by fully participating in the GameMaker® designed course, were those who performed well in their courses, especially in the Object-Oriented Programming course. This may have introduced a bias accentuating the positive gains observed by participating in the intervention on students' performance in both the C++ and Object-Oriented Programming courses.

Future Research

One of the missing elements in the literature examining computer-based learning in undergraduate education is the absence of action-based research. Most studies focus on student's outcomes and how a set of predictors, theory or practice based, influence the outcome of interest. This misses a great information resource, the researchers' and students' potential contribution. Researchers may adopt instruments or tools ready for implementation in research settings without modifying or altering them in ways that could be conducive to teaching. For instance, researchers could enhance GameMaker® by introducing theoretical and structured lesson plans along the inquiry-based assignments for students. Students also could inform researchers on the viability, ease and challenge levels of assigned work or technology. The rich qualitative information provided by all participants in research refine the course of experiments and observational studies to generate more confident results. Future research on computer-based technology in curriculum

development and assessment for STEM undergraduate education should utilize action-based research.

Future research on the use of technology in STEM education should concentrate on the instructor as an important source of explanatory power to educational outcomes. Many experiments in the literature focus on the type of technology, content, duration and perceptions of students towards all of such variables and neglect the importance of the instructor. More experimental work is needed on the effect of different interactivity levels exhibited by instructors. This goes hand in hand with the larger focus on educational leadership where teachers' leadership styles in classrooms has been linked to educational outcomes. This helps schools and administrators better design learning environments where students benefit not only from the constructionist educational methods, but also proven educational leadership practice to produce an augmented combined effect capable of enhancing students' performance.

While non-equivalent experimental research designs offer great insight into the effectiveness or instructional techniques, they fail to generate comparable results to classical experiments. The absence of random assignment into groups provides more evidence to the existence of confounding variables such as home environment, income, type of education received prior to post-secondary education and stress levels. While all students ended up in the same classroom, regardless of where they came from, contextual factors still could potentially threaten the internal validity of experiments. Therefore, future experimental work should utilize random assignment under the umbrella of classical experiments to generate more confident findings.

Future research should also utilize more refined measurements of educational outcomes. While overall test scores have been proven to provide a good measure of general performance, more fine-grained assessments such as codes' creation, codes' evaluation and codes' validation

exercises supply researchers with specific relationships that could supplement the use of good instruction in undergraduate classrooms. Moderation analysis between participating in specific instructional techniques performance on several tasks composing a general assessment provides researchers with more detailed descriptions of the magnitude the new instructional method has on the specific task in question.

Future research should consider mediation analysis of the relationship between computer-based instruction and performance on STEM assessments in higher education. It has been proven that students are likely to perform better on computer science and mathematics courses if they utilize constructionist methods of instruction such as the use of GameMaker®. Nevertheless, why such a relationship exists is still subject to much introspection. More research on how the two variables are related is needed. Knowledge of paths connecting instructional methods and performance measures are likely to improve instructor's ability, the content depth, students' learning outcomes and overall performance delivering the long-awaited goals of graduating students who are capable of securing meaningful positions providing them with economic autonomy in life.

Recommendations

To better generate invariant findings, educational institutions interested in enhancing programming skills among students should implement GameMaker® on a multitude of programming courses. This includes C++, Object-Oriented Programming and other courses. This helps researchers and stakeholders determine whether GameMaker® or for that matter any technology tool for the same or similar purpose carries tantamount effects on students' performance. This helps in the creation of more robust, valid and reliable curriculum and changes to existing curricula exhibiting best-practice research.

Furthermore, experiments on the use of computer-aided design should be carried out by parties other than the researcher. While, the researcher in this study is a computer scientist lecturer at several institutions, it would have been better to assess the effectiveness of GameMaker® by class instructors themselves rather than the researcher. This reduces internal validity biases to the experiment and increases objectivity by making the class mimic real-world arrangements.

One of the recommendations of this research is the assignment of GameMaker® as a 1 credit required course taught in a lab format for information technology majors. Jordanian universities require students to take C++ as the fits programming course in their curriculum. This course is difficult by not only students' accounts, but also faculty expert feedback. For instance, in the United States, C++ programming courses required for meeting the general requirement credits or introducing students to programming in the computer science-oriented fields was replaced by Python in many universities. This logic stems from the fact that many students left computer science fearing from C++ programming. Taking a one credit course in the first semester using GameMaker® provides a cushion for many students who feel less confident about their courses.

APPENDIX A: IRB CONCERRENCE NOTICE

WAYNE STATE UNIVERSITY

IRB Administration Office
87 East Canfield, Second Floor
Detroit, Michigan 48201
Phone: (313) 577-1628
FAX: (313) 993-7122
http://irb.wayne.edu

CONCURRENCE OF EXEMPTION

To: Alaa Al-Makhzoomy
Honors College

From: For Dr. Deborah Ellis M. Tancer, MD/BC
Chairperson, Behavioral Institutional Review Board (B3)

Date: January 18, 2018

RE: IRB #: 012118B3X

Protocol Title: Effect of Game Development-Based Learning on the ability of Information Technology undergraduates to learn Computer and Object-Oriented Programming

Sponsor:

Protocol #: 1801001112

The above-referenced protocol has been reviewed and found to qualify for **Exemption** according to paragraph #2 of the Department of Health and Human Services Code of Federal Regulations [45 CFR 46.101(b)].

- Social/Behavioral/Education Exempt Protocol Summary Form (revision received in IRB Office 01/16/2018)
- Research Protocol (dated 2017, received in the IRB Office 01/02/2018)
- Medical records are not being accessed therefore HIPAA does not apply.
- Research Information Sheet (revision dated 01/16/2018)
- Data collection tools (2): (I) Demographic Information and (II) Learner's Knowledge Pre-test

This proposal has not been evaluated for scientific merit, except to weigh the risk to the human subjects in relation to the potential benefits.

- Exempt protocols do not require annual review by the IRB.
- All changes or amendments to the above-referenced protocol require review and approval by the IRB **BEFORE** implementation.
- Adverse Reactions/Unexpected Events (AR/UE) must be submitted on the appropriate form within the timeframe specified in the IRB Administration Office Policy (http://irb.wayne.edu/policies-human-research.php).

NOTE: Forms should be downloaded from the IRB Administration Office website http://irb.wayne.edu at each use.

Notify the IRB of any changes to the funding status of the above-referenced protocol.

APPENDIX B: RESEARCH INFORMATION SHEET

Title of Study: Effect of Game Development-Based Learning on the ability of Information Technology undergraduates to learn Computer and Object-Oriented Programming

Principal Investigator (PI): Alaa Al-Makhzoomy
Learning Design and Technology
(313) 212 0165

Purpose:
You are being asked to be in a research study of exploring the effects of Game Development-Based Learning (GDBL) on the ability of Information Technology students to learn computer and Object-Oriented Programming (OOP) because you are registered in a computer-programming course in the faculty of Information Technology at Yarmouk University. This study is being conducted at Wayne State University and Yarmouk University.

Study Procedures
If you take part in the study, you will be asked to respond to a demographic information survey in which you will provide information about you and your undergraduate study. Responding to all questions in this survey is optional. In the survey, you will be asked if you agree to enroll in an online course during the semester. If you agree, you will be asked to do the following throughout the conduct of the study:

1) Complete an hour-long pre-test during the first week of the semester and an hour-long post-test in the middle of the semester.
2) Enroll in an eight-week long online course about game development using GameMaker® designed by the PI.

Benefits
The possible benefits to you for taking part in this research study are acquiring new knowledge about game development and computer programming. Additionally, information from this study may benefit Information Technology students and instructors in the future.

Risks
There are no known risks at this time to participation in this study.

Costs
There will be no costs to you for participation in this research study.

Compensation
You will not be paid for taking part in this study.

Confidentiality:

- All information collected about you during the course of this study will be kept without any identifiers.
- You will be identified in the research records by a code name or number. There will be no list that links your identity with this code.

Voluntary Participation /Withdrawal:
Taking part in this study is voluntary. You are free to not answer any questions or withdraw at any time. Your decision will not change any present or future relationships with Wayne State University or its affiliates or Yarmouk University or its affiliates.

The PI may stop your participation in this study without your consent. The PI will make the decision and let you know if it is not possible for you to continue. The decision that is made is to protect your health and safety, or because you did not follow the instructions to take part in the study.

Participation in the study or withdrawing from it will not affect your grades in your study at Yarmouk University.

Questions
If you have any questions about this study now or in the future, you may contact Alaa Al-Makhzoomy at the following phone number +1 (313) 212 0165. If you have questions or concerns about your rights as a research participant, the Chair of the Institutional Review Board can be contacted at +1 (313) 577-1628. If you are unable to contact the research staff, or if you want to talk to someone other than the research staff, you may also call the Wayne State Research Subject Advocate at +1 (313) 577-1628 to discuss problems, obtain information, or offer input.

Participation
By completing the demographic information survey, you are agreeing to participate in this study.

The data that you provide may be collected and used by Qualtrics at Wayne State University as per its privacy agreement.

APPENDIX C: IRB APPROVED INFORMATION SHEET

Research Information Sheet

Title of Study: Effect of Game Development-Based Learning on the ability of Information Technology undergraduates to learn Computer and Object-Oriented Programming

Principal Investigator (PI): Alaa Al-Makhzoomy
Learning Design and Technology
313-212-0165

Purpose:

You are being asked to be in a research study of exploring the effects of Game Development-Based Learning (GDBL) on the ability of Information Technology students to learn computer and Object-Oriented Programming (OOP) because you are registered in a computer-programming course in the faculty of Information Technology at Yarmouk University. This study is being conducted at Wayne State University and Yarmouk University.

Study Procedures

If you take part in the study, you will be asked to respond to a demographic information survey in which you will provide information about you and your undergraduate study. Responding to all questions in this survey is optional. In the survey, you will be asked if you agree to enroll in an online course during the semester. If you agree, you will be asked to do the following throughout the conduct of the study:

1) Complete an hour-long pre-test during the first week of the semester and an hour-long post-test in the middle of the semester.
2) Enroll in an eight-week long online course about game development using GameMaker® designed by the PI.

Benefits

The possible benefits to you for taking part in this research study are acquiring new knowledge about game development and computer programming. Additionally, information from this study may benefit Information Technology students and instructors in the future.

Risks

There are no known risks at this time to participation in this study.

Costs

There will be no costs to you for participation in this research study.

Compensation

You will not be paid for taking part in this study.

Confidentiality:

- All information collected about you during the course of this study will be kept without any identifiers.
- You will be identified in the research records by a code name or number. There will be no list that links your identity with this code.

Voluntary Participation /Withdrawal:
Taking part in this study is voluntary. You are free to not answer any questions or withdraw at any time. Your decision will not change any present or future relationships with Wayne State University or its affiliates or Yarmouk University or its affiliates.

The PI may stop your participation in this study without your consent. The PI will make the decision and let you know if it is not possible for you to continue. The decision that is made is to protect your health and safety, or because you did not follow the instructions to take part in the study.

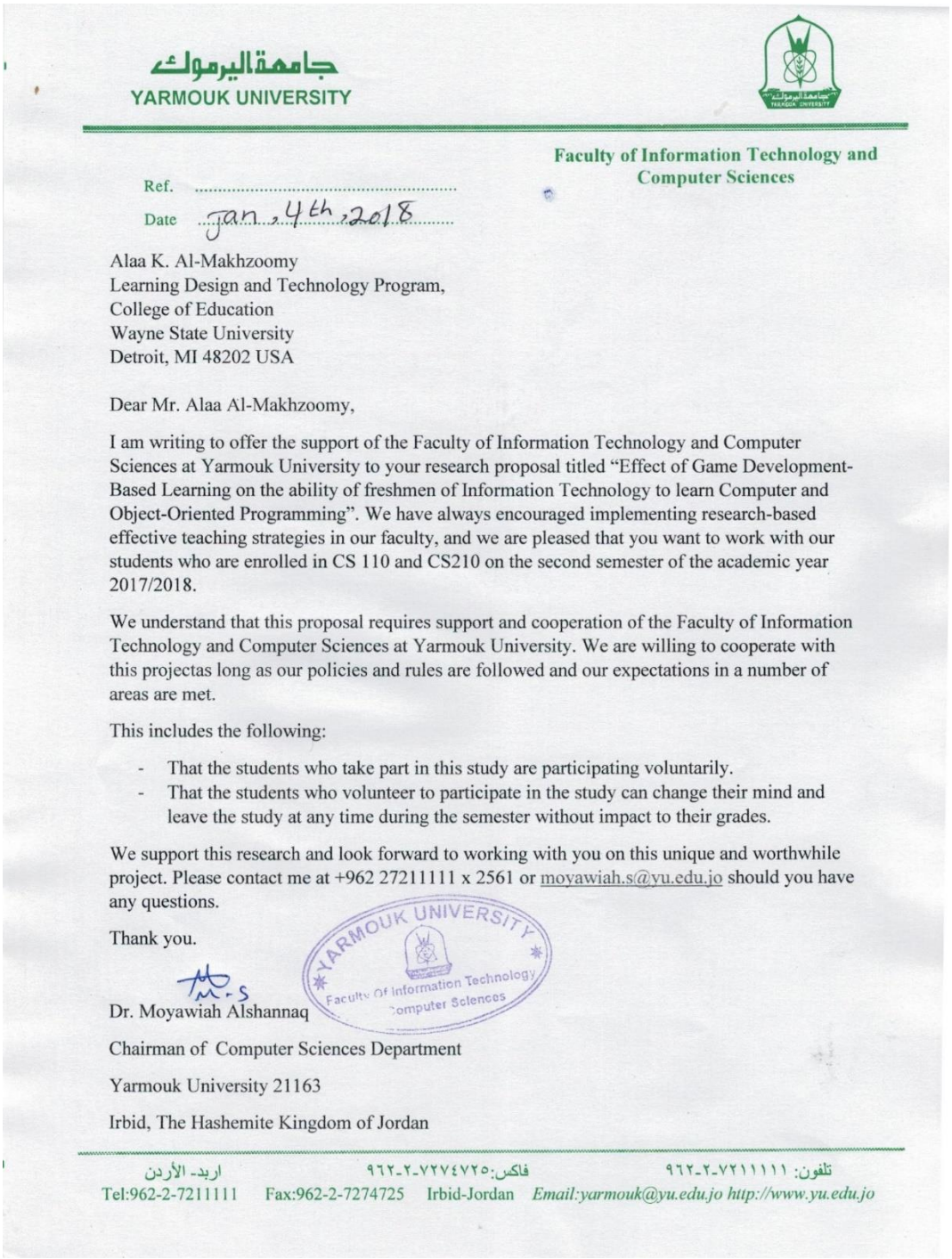
Participation in the study or withdrawing from it will not affect your grades in your study at Yarmouk University.

Questions
If you have any questions about this study now or in the future, you may contact Alaa Al-Makhzoomy at the following phone number +1 (313) 212 0165. If you have questions or concerns about your rights as a research participant, the Chair of the Institutional Review Board can be contacted at +1 (313) 577-1628. If you are unable to contact the research staff, or if you want to talk to someone other than the research staff, you may also call the Wayne State Research Subject Advocate at +1 (313) 577-1628 to discuss problems, obtain information, or offer input.

Participation
By completing the demographic information survey, you are agreeing to participate in this study.

The data that you provide may be collected and used by Qualtrics at Wayne State University as per its privacy agreement.

APPENDIX D: LETTER OF CONSENT TO CONDUCT THE STUDY IN THE FACULTY OF INFORMATION AND TECHNOLOGY AT YARMOUK UNIVERSITY.



جامعة اليرموك
YARMOUK UNIVERSITY

Faculty of Information Technology and Computer Sciences

Ref. .....
Date Jan, 4th, 2018

Alaa K. Al-Makhzoomy
Learning Design and Technology Program,
College of Education
Wayne State University
Detroit, MI 48202 USA

Dear Mr. Alaa Al-Makhzoomy,

I am writing to offer the support of the Faculty of Information Technology and Computer Sciences at Yarmouk University to your research proposal titled "Effect of Game Development-Based Learning on the ability of freshmen of Information Technology to learn Computer and Object-Oriented Programming". We have always encouraged implementing research-based effective teaching strategies in our faculty, and we are pleased that you want to work with our students who are enrolled in CS 110 and CS210 on the second semester of the academic year 2017/2018.

We understand that this proposal requires support and cooperation of the Faculty of Information Technology and Computer Sciences at Yarmouk University. We are willing to cooperate with this projectas long as our policies and rules are followed and our expectations in a number of areas are met.

This includes the following:

- That the students who take part in this study are participating voluntarily.
- That the students who volunteer to participate in the study can change their mind and leave the study at any time during the semester without impact to their grades.

We support this research and look forward to working with you on this unique and worthwhile project. Please contact me at +962 27211111 x 2561 or moyawiah.s@yu.edu.jo should you have any questions.

Thank you.

YARMOUK UNIVERSITY
Faculty Of Information Technology
Computer Sciences

Dr. Moyawiah Alshannaq

Chairman of Computer Sciences Department

Yarmouk University 21163

Irbid, The Hashemite Kingdom of Jordan

اربد- الأردن فاكس:٩٦٢-٢-٧٢٧٤٧٢٥ تلفون: ٩٦٢-٢-٧٢١١١١١
Tel:962-2-7211111 Fax:962-2-7274725 Irbid-Jordan *Email:yarmouk@yu.edu.jo http://www.yu.edu.jo*

APPENDIX E: LANGUAGE OF INSTRUCTION LETTER

جامعة اليرموك
YARMOUK UNIVERSITY

Faculty of Information Technology and Computer Sciences

Ref. .....
Date Jan. 4th, 2018

TO WHOM IT MAY CONCERN

This is to certify that the English Language is the official language of instruction utilized by the faculty of Information Technology and Computer Sciences at Yarmouk University. Curriculum and course work ranging from lectures, textbooks, reading, writing, assignments, presentations and discussions are held in English.

Students of the Faculty of Information Technology and Computer Sciences are expected to possess the skills and capabilities to comprehend and communicate effectively in the English language and should not face challenges or difficulties in courses that adhere to the English language as the medium of instruction.

B. Abul Huda

Prof. Dr. Bilal Abul-Huda
Dean, The Faculty of Information Technology and Computer Sciences
Yarmouk University 21163
Irbid, The Hashemite Kingdom of Jordan

YARMOUK UNIVERSITY
Faculty Of Information Technology And Computer Sciences

اربد- الأردن فاكس:٩٦٢-٢-٧٢٧٤٧٢٥ تلفون: ٩٦٢-٢-٧٢١١١١١
Tel:962-2-7211111 Fax:962-2-7274725 Irbid-Jordan Email:yarmouk@yu.edu.jo http://www.yu.edu.jo

APPENDIX F: DEMOGRAPHIC INFORMATION

There are nine questions in this survey. Choose one answer for each question that best explains or represents you.

1. What is your age? (Choose One)
   - 18
   - 19
   - 20
   - 21
   - 22+
2. What is your gender? (Choose One)
   - Male
   - Female
3. What undergraduate program are you enrolled in at Yarmouk University?
   - Computer Science (CS)
   - Computer Information Systems (CIS)
   - Management Information Systems (MIS)
   - Network and Information Security (NIS)
   - Software Engineering (SE)
   - Computer Engineering (CE)
4. What is your Minor study (if any)? _____.
5. Which of the following two courses you are registered in on the second semester 2017-2018?
   - CS110: Programming with a Selected Language
   - CS210: Object-Oriented Programming
   - Both courses
   - None of the two courses.
6. How many total classes are you enrolled in this semester? _____.
7. What was the name of your school? _____.
8. Where is your school located? _____.
9. What was your high school / Second Secondary (Tawjihi) GPA? _____.
10. Are you currently working? (Choose One)
    - Yes
    - No (You may return the survey now)
11. If Yes, How many hours per week? (Choose One)
    - Full-time 40+ hours a week
    - Part-time <40 hours per week
    - Not working
12. Is your current job part of the work study program at Yarmouk University? (Choose One)
    - Yes
    - No

APPENDIX G: LEARNER'S KNOWLEDGE PRE-TEST (EXAMPLE)

Effect of Game Development-Based Learning (GDBL) on the ability of
IT undergraduates to learn Computer and Object-Oriented Programming

Principal Investigator (PI): Alaa Al-Makhzoomy
Wayne State University
College of Education
Learning Design and Technology
(313) 212 0165

Topic 1: GameMaker® 8.1

1) A Sprite is:
   a) A movable image/animation
   b) An image
   c) A video
   d) All of the above are correct.
2) A feature that prevents objects from overlapping with objects of the same feature is named:
   a) Solid
   b) Static
   c) Stop
   d) Prevent
3) An Event is:
   a) the same as a sprite.
   b) the same as an object.
   c) the same as an action.
   d) Something has happened like a collision or some has pressed a key.
4) **Check Collision** action performs actions when there is always a collision.
   a) True
   b) False
5) Events and Actions specify how objects should _____.
   a) look like
   b) behave
   c) move
   d) create rooms
6) It is not necessary for a game to have a room to run
   a) True
   b) False
7) Only one object could be created out of a sprite.
   a) True
   b) False

8) Sprites do not do anything on their own, objects are the parts of the game that control how the game elements move and react to each other.
   a) True
   b) False
9) Test Chance action acts like throwing a die with many sides, where only one side will trigger an event.
   a) True
   b) False
10) Which of the following is best used as a name for a sprite?
   a) SPR Dragon
   b) spr.dragon
   c) spr_dragon
   d) Sprite->Dragon
11) _____ indicate the things that happen in a game created with GameMaker®. They are placed in events of objects. They are performed whenever the event takes place, resulting in certain behavior for the instances of the object.
   a) Actions
   b) Programmer
   c) Fade
   d) Paths
12) Large images that are used as backgrounds for the rooms in which the game takes place.
   a) Backgrounds
   b) Collision Detection
   c) Debug
   d) Actions
13) A process of determining if two objects have collided by testing their bounds or a spatial overlap.
   a) Backgrounds
   b) Actions
   c) Collision Detection
   d) Debug
14) The process of tracking and eliminating errors or bugs from your source code.
   a) Debug
   b) Collision Detection
   c) Backgrounds
   d) Actions
15) _____ sense whenever something happens in the game, thus the instances of the objects act. For example, if an object hits a wall, these trigger the action associated to stop the object's movement.
   a) Backgrounds
   b) Events
   c) Debug
   d) Actions

16) Refers to the process of adding additional pixels around the border of an object in order to blend it into its background more smoothly, and to reduce the appearance of jagged edges.
    a) Anti-aliasing
    b) Actions
    c) Events
    d) Debug
17) A transition effect where an image slowly appears out of a solid color, such as black or white
    a) Fade
    b) Anti-aliasing
    c) Events
    d) Actions
18) A single image in a series of images to be animated.
    a) Anti-aliasing
    b) Fade
    c) Frame
    d) Actions
19) A program that's free for commercial use that has a simple user interface for developing computer games.
    a) Frame
    b) Fade
    c) GameMaker®
    d) Actions
20) A copy of an object. An object can have multiple _____ that all have different actions.
    a) Genre
    b) Frame
    c) Actions
    d) Instances
21) To exit the game by clicking on the red X button, the following should be added to objExitButton:
    a) Event Left Button, then action "End the game".
    b) Event Left Pressed, then action "End the game".
    c) Event Left Released, then action "End the game".
    d) All the above are correct.
22) The person who writes the software.
    a) Frame
    b) Instances
    c) Genre
    d) Programmer
23) The collection of images, sounds, backgrounds, scripts, etc. that can be used by the video game designer.

a) Resources
b) Frame
c) Programmer
d) Instances

24) The graphical environment for the game, often programmed in games as levels.
a) Programmer
b) Rooms
c) Resources
d) Instances

APPENDIX H: LEARNER'S KNOWLEDGE PRE-TEST (EXAMPLE)

Effect of Game Development-Based Learning (GDBL) on the ability of
IT undergraduates to learn Computer and Object-Oriented Programming

Principal Investigator (PI): Alaa Al-Makhzoomy
Wayne State University
College of Education
Learning Design and Technology
(313) 212 0165

Topic 2: C++ Programming Language

1) Which of the following statements is false?
   a) Object-oriented programming is today's key programming methodology.
   b) C++ is one of today's most popular software development languages.
   c) Software commands computer hardware to perform tasks.
   d) In use today are more than a trillion general-purpose computers and trillions more cellphones, smartphones and other handheld devices.
2) Which of the following statements is false?
   a) Object-oriented programming is today's key programming methodology.
   b) C++ is standardized worldwide through the International Organization for Standardization
   c) Hardware controls software.
   d) In use today are more than a billion general-purpose computers and billions more cell phones, smartphones, and handheld devices (such as tablet computers).
3) Which of the following statements is false?
   a) Cloud computing allows you to use software, hardware and information stored on remote computers via the Internet and available on demand rather than having it stored on your personal computer.
   b) Electronic health records enable health care providers to share patients' information across a secure network, improving patient care, reducing the probability of error and increasing overall efficiency of the health care system.
   c) Global Positioning System (GPS) devices a signal satellite to retrieve location-based information.
   d) The Human Genome Project was founded to identify and analyze the 20,000+ genes in human DNA.
4) Which of the following statements is false?
   a) Cloud computing allows you to use software, hardware, and information stored in the cloud i.e., accessed on remote computers via the Internet and available on demand rather than having it stored on your personal computer.
   b) Cloud computing services allow you to increase or decrease resources to meet your needs at any given time, so they can be more cost effective than purchasing expensive

hardware to ensure that you have enough storage and processing power to meet your needs at their peak levels.

c) **Businesses using cloud computing services must still manage the applications, which can be costly.**

d) Both (a) and (c).

5) This creates object code and stores it on disk.
   a) Interpreter
   b) Compiler
   c) Preprocessor
   d) Loader

6) Today, virtually all new major operating systems are written in:
   a) Objective-C
   b) C or C++
   c) Visual C#
   d) Ada

7) Which of the following languages is used primarily for scientific and engineering applications?
   a) Fortran
   b) COBOL
   c) Pascal
   d) Basic

8) Which of the following is most closely associated with Moore's Law?
   a) Every year or two, the price of computers has approximately doubled.
   b) Object-oriented programming uses less memory than previous software-development methodologies.
   c) Demand for communications bandwidth is decreasing dramatically each year.
   d) Every year or two, the capacities of computers have approximately doubled without any increase in price.

9) Which of the following is not one of the six logical units of a computer?
   a) Input unit.
   b) Output unit.
   c) Central Processing Unit (CPU).
   d) Printer.

10) Which of the following statements is false?
   a) Speaking to your computer is a form of input.
   b) Playing a video is an example of output.
   c) A multi-core processor implements several processors on a single integrated-circuit chip
   d) Information in the memory unit is persistent it is retained when the computer's power is turned off.

11) Which of the following statements is false?
   a) The impressive functions performed by computers involve only the simplest manipulations of 1s and 2s.

b) ASCII is a popular subset of Unicode.
c) Fields are composed of characters or bytes.
d) On some operating systems, a file is viewed simply as a sequence of bytes.

12) _____ is a graphical language that allows people who design software systems to use an industry stan-dard notation to represent them.
a) The Unified Graphical Language
b) The Unified Design Language
c) The Unified Modelling Language
d) None of the above.

APPENDIX I: LEARNER'S KNOWLEDGE POST-TEST (EXAMPLE)

Effect of Game Development-Based Learning (GDBL) on the ability of IT undergraduates to learn Computer and Object-Oriented Programming

Principal Investigator (PI): Alaa Al-Makhzoomy
Wayne State University
College of Education
Learning Design and Technology
(313) 212 0165

Topic 2: C++ Programming Language

Question 1 *(tests students' ability to code simple algebraic expressions)*

In mathematics, the general form of a second-degree polynomial is $y = ax^2 + bx + c$. Which of the following assignment statements correctly represent(s) the general form?

i. y = a * x * x + b * x + c

ii. y = x * (a * x + b) + c

iii. y = a * (x * x) + b * (x + c)

(a) iii only
(b) i and ii only
(c) i and iii only
(d) i, ii, and iii

Question 2 *(tests students' ability to interpret logical expressions)*

Given the expression: (num1 < num2) AND (num2 < num3)
Which of the following statements must always be true?

(a) The expression returns a value that represents true or false.
(b) The expression is equivalent to (num1 >= num2) OR (num3 >= num2).
(c) The expression is only false if both parenthesized expressions are false.
(d) The expression is basically the same as an algebraic expression (e.g., 1 < x and x< 5, or 1 < x < 5), so it can alternately be coded as (num1 < num2 < num3).

Question 3 *(A question that tests students' ability to code logical expressions.)*

The following algorithm prompts a user to re-enter a student's grade that is not valid (outside the range of 0 to 100):

While the grade entered by the user is outside the range of 0 to 100,
Display an error message to the user
Prompt the user to re-enter the grade
Input the grade

Which code fragment correctly implements the condition in the incomplete loop shown next?

```
WHILE (condition) DO
      WRITE("Invalid grade!")
      WRITE("Please enter a grade in the range 0 to 100:")
      READ(grade)
END WHILE
```

(a) *condition* = (grade < 0) AND (grade > 100)

(b) *condition* = (grade >= 0) AND (grade <= 100)

(c) *condition* = (grade >= 0) OR (grade <= 100)

(d) *condition* = (grade < 0) OR (grade > 100)

Question 4 ***(****tests students' ability to interpret conditionals.)***

Given the code:

```
IF (actualEnrollment < maxEnrollment) THEN
      WRITE("Seats available!")
ELSE
      WRITE("Sorry! NO seats available!")
END IF
WRITE("Please try again next term.")
```

Which of the following statements must always be true?

(a) "Please try again next term" will only display when the value in the actualEnrollment variable is greater than the value in the maxEnrollment variable.
(b) "Sorry! NO seats available!" will never display because there is no condition following the ELSE.
(c) "Please try again next term" will display no matter what values the actualEnrollment and maxEnrollment variables hold.
(d) "Sorry! NO seats available!" will only display when the value in the actualEnrollment variable is greater than the value in the maxEnrollment variable.

Question 5 ***(****tests students' understanding of classes and objects.)***

Assume that we have a class student with private attributes name, major, and GPA, and public methods getName, getMajor, and getGPA. In addition, assume that we have a separate class professor with private attributes name and subject.

Which of the following statements must always be true?

(a) Since student has a method getName, professor cannot have a method getName.
(b) Since student and professor both have an attribute name, the value of name for a student object must always be different than the value of name for a professor object.
(c) A professor object can access a student object's attributes.
(d) student objects and professor objects have different attributes and different methods.

APPENDIX J: FIRST EXAM OF CS210 - OBJECT-ORIENTED PROGRAMMING ON THE SECOND SEMESTER 2017-2018 AT YARMOUK UNIVERSITY

Yarmouk University
Faculty of information Technology and Computer Sciences
Department of Computer Sciences

Course Name: Object-oriented Programming (CS 210) Semester: Second
First Exam Date: 21/3/2018
Student Name: Student Number:
Serial Name:

Question1 [6 PTS]

From the class implementation below, provide the class's interface?

```
#include<iostream>
#include "Building.h"

Building:: Building (float i, int
j, string n):Height (i)
{Name=n;
      Apt = j;     }

float Building::getHeight() const{
            return Height;
       }
int    Building::getapt() const{
            return Apt;
       }
string Building::getName() const
{return Name;}

int main ()

{
Building w;
w.print();
return 0;
}
```

Question2 [8 PTS]

From the following code, find all errors and specify why it's an error? **DO NOT CORRECT THE ERROR**

Team.h

```
1. #ifndef ABC_h
2. #define ABC_H
3. #include<iostream>
4. using namespace std;
5. class Team
6. {
7. const int A;
```

Team.cpp

```
21.
22.     Team:: Team (int 1)
23.     {C=i)
24.
25.     Void Team::setB(int j)
26.     { B=j; }
27.
```

```
8. const int B;
9. int c=10;
10.     void print() const;

11.     public:
12.     Team (int(:A(0),
   B(0);
13.     Team ():A(0),B(0)
14.     {C= 0}
15.     int  getA() const
16.     { return A;}
17.     int  getB() const;
18.     void setB(int);
19.     };
20.     endif
```

```
28.     int Team::getB(){
29.     Return B;
30.     }
31.     void Team::print() const{
32.     cout<<"A:"<<getA()<<end];
33.     cout<<"B:"<<getB()<<end];
34.     cout<<"C:"<<getC()<<end];
35.
36.     }
37.
```

```
38.
39.     int main ()
40.     {
41.     Team Barca;
42.     const Team ATH;
43.     Barca.print();
44.     ATH.setB(20);
45.     Return 0;
46.     }
```

- _ 1 pt for each error

Error Line Number	Why it's an Error?

Question3 [6 PTS]

What is the output of the following code?

```
#include<iostream>
Using namespace std;
Close BMI {
public{
BMI( double i-0, double j-0
{ setData(i,j);
  If (height i-0)
        cout<<”C zero”<<endl;
  Else
        cout<<”C zero”<<endl;
}
    ~BMI ()
{ cout<<”d “<<weight/height<<endl; }

void setData (double i, double j){
             height - i;
            weight - j;
      }
void change() {
             height += 50;
            weight += 200;       }
private:
  double height;
    double weight;
};

void change () {
obj9.change();
BMI obj17(120, 60);
static BMI obj5(20, 60);
cout << “Change Finished\n”;
}

BMI obj9;
 int main(){
const BMI obj1(100,150);
BMI &obj4=obj0;
BMI *obj5=obj0;
change ();
static BMI obj2(100,100);
change();
obj5->change();
cout << “The End\n”;
      return 0;}
```

Output

APPENDIX K: SECOND EXAM OF CS210 – OBJECT-ORIENTED PROGRAMMING ON THE SECOND SEMESTER 2017-2018 AT YARMOUK UNIVERSITY

Yarmouk University
Faculty of information Technology and Computer Sciences
Department of Computer Sciences

Course Name: Object-oriented Programming (CS 210) Semester: Second
Second Exam Date: 3/5/2018
Student Name: Student Number:
Serial Name:

Question 1 [8 PTS]

For each of the following lines of code, provide the expression that the computer executes.

A) Circle A, B, C;(The function that you must use are MEMBER FUNCTIONS)
l{{A+B) <{C*A} }

B) Date X,Y;(;(The function that you must use are ALL GLOBAL FUNCTIONS)
X++=5+-Y;

Question 2 [9 PTS]

From the following code, find all errors and specify why it's an error? **DO NOT CORRECT THE ERROR**

Team.h

```
47.      #include<iostream>
48.      using namespace std;
49.      class Exam
50.      {
51.      Private:
52.           int first, second, final;
53.          static float average;
54.      puplic:
55.               Exam& setfirst(int a){first=a;   return *this;}
56.  Exam& setsecond(int b){second=b;   return *this;}
57.        Exam& setfinal(int c){final=c;   return *this;}
58.        int  getfirst() {return first;}
59.        int  getsecond() {return second;}
60.      friend const ostream& operator<<( const ostream&,const Exam&);
61.
62.      static float GetAverage()
63.      {  return getfirst()/2;}
64.      };
65.      Float Exam::average=50.0;

66.      Void main()
67.      {  Exam *CS210=new Exam;
68.         Exam *courses=new Exam[10];
69.        (*CS210).setfirst(10).setsecond(10).getfirst().setfinal(30);
70.         Cout<<Exam::average<<endl;
71.         Cout<<Cs210->setfinal(40)<<endl;
72.             delete CS210;
73.             delete Courses;   }
```

Error Line Number	Why it's an Error?

Question3 [6 PTS]

What is the output of the following code?

```
#include<iostream>
using namespace std;

class Equation {
private
       float c1, c0;
public
       Equation(float a=1, float b=0)
       {c1=a, c0=b;
        cout<<"Constructor called:";
       Print Equation();}

       Equation(const Equation& cpy)
       {      c1=cpy.c1;
              c0=cpy.c0;
              Cout<<"Copy Constructor Called:';
                   PrintEquation();}

       void PrintEquation()
       {          if (c0>0)
                   cout<<c1<<"X+"<<c0<<endl;
                 else if (c0<0)
                      cout<<c1<<X-"<<c0<<endl;
                 else
                      cout<<c1<<"X" endl;
        }
        ~Equation()
        (cout<<"Deleted;";'
         PrintEquation();)
};

Class LinearSystem
{private
      Equation A,B;
public
     LinearSystem(Equation X, Equation Y):B(Y),A(X)
     {}
};

void main ()
{
     Equation N(3, 4), M(5, 6);
     LinearSystem (N, M);
}
```

Output

Q4: Writing code [6 marks]:

The following class represents a computer number. Any complex number has two parts, real and imaginary part. For example, 6.1+3i is a complex number with 6.1 as the real part and 3 as the imaginary part.

```
Class complex
{
Private:
    float real;
    float image;

public:
     Complex (float a=0.0, float b=0.0)
     {real =a; image=b;}
}
```

For this question, you will right the following functions

A) //overload the ! operator as a **member function** to conjugate the complex number.
Note: the conjugate of the complex number is -1 multiplied by its imaginary part.

B) //overload <<operator to output the complex number. (if the real and imaginary parts are 3.2 and 1.7 then the number is printed as 3.2 81.7).

REFERENCES

Alkhawaldeh, N., & Menchaca, M. (2014). Barriers to Utilizing ICT in Education in Jordan. *International Journal on E-Learning*, *13*(2), 127–155.

Baytak, A., Land, S. M., & Smith, B. K. (2011). Children as Educational Computer Game Designers: An Exploratory Study. *TOJET: The Turkish Online Journal of Educational Technology*, *10*(4).

Beck, M. (2010, June 3). Civic Participation and Local Development: Building Effective Communities in Jordan, Publications, Jordan Office. Retrieved December 1, 2016, from http://www.kas.de/jordanien/en/publications/19795/

Belland, B. R., Walker, A. E., Kim, N. J., & Lefler, M. (2017). Synthesizing results from empirical research on computer-based scaffolding in STEM education: A meta-analysis. *Review of Educational Research*, *87*(2), 309–344.

Chandler, P. D. (2013). Middle years students' experience with new media. *Australian Journal of Education*, *57*(3), 256–269.

Charlier, N., & De Fraine, B. (2012). Game-Based Learning in Teacher Education: A Strategy to Integrate Digital Games into Secondary Schools. *International Journal of Game-Based Learning (IJGBL)*, *2*(2), 1–12.

Creswell, J. W. (2013). *Qualitative inquiry and research design: choosing among five approaches* (3rd ed.). Los Angeles: SAGE Publications.

Dewey, J., & Small, A. W. (1897). *My pedagogic creed*. EL Kellogg & Company.

Doman, M., Sleigh, M., & Garrison, C. (2015). Effect of GameMaker® on Student Attitudes and Perceptions of Instructors. *International Journal of Modern Education and Computer Science*, *7*(9), 1.

Eagle, M., & Barnes, T. (2009). Experimental evaluation of an educational game for improved learning in introductory computing. In *ACM SIGCSE Bulletin* (Vol. 41, pp. 321–325). ACM.

Ernst, J. V., & Clark, A. C. (2012). Fundamental Computer Science Conceptual Understandings for High School Students Using Original Computer Game Design. *Journal of STEM Education: Innovations and Research*, *13*(5), 40.

Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., & Edwards, T. (2016). Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior*, *54*, 170–179.

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional Strategy in the Teaching of Computer Programming: A Need Assessment Analyses. *TOJET: The Turkish Online Journal of Educational Technology*, *9*(2).

Jenson, J., & Droumeva, M. (2015). Making games with Game Maker: A computational thinking curriculum case study. In *ECGBL2015-9th European Conference on Games Based Learning: ECGBL2015* (p. 260). Academic Conferences and publishing limited.

Johnson, C. (2017). Learning to Program with Game Maker. *Online Submission*, *1*(2).

Jong, M. S., Shang, J., & Lee, F. (2010). Constructivist Learning Through Computer Gaming. In *M. Syed (Ed.), Technologies Shaping Instruction and Distance Education: New Studies and Utilizations* (pp. 207–222). Hershey, PA.

JUST IT Undergraduate Courses According to 2015 Curriculum. (2016, December 1). [Educational]. Retrieved December 1, 2016, from http://www.just.edu.jo/FacultiesandDepartments/it/Departments/cs/Pages/Courses_2015.aspx

Kindler, E., & Krivy, I. (2011). Object-oriented simulation of systems with sophisticated control. *International Journal of General Systems*, *40*(3), 313–343. https://doi.org/10.1080/03081079.2010.539975

Kunkle, W. M., & Allen, R. B. (2016). The Impact of Different Teaching Approaches and Languages on Student Learning of Introductory Programming Concepts. *Trans. Comput. Educ.*, *16*(1), 3:1–3:26. https://doi.org/10.1145/2785807

Majgaard, G. (2014a). The Playful and Reflective Game Designer. *Electronic Journal of E-Learning*, *12*(3), 271–280.

Marlow, C. M. (2012). Making Games for Environmental Design Education: Revealing Landscape Architecture. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*, *4*(2), 60–83.

Moodle - Open-source learning platform | Moodle.org. (2017, December 3). Retrieved December 3, 2017, from https://moodle.org/

Mutah IT Course Descriptions. (2016, December 1). Retrieved December 1, 2016, from https://view.officeapps.live.com/op/view.aspx?src=https://www.mutah.edu.jo/studyplane/science/ITPlans/coursedescriptionEnglish.doc

Overmars, M. (2004). Game design in education. *Internet: Http://Www. Cs. Uu. Nl/Research/Techreps/Repo/CS-2004/2004-056. Pdf [May, 2008]*.

Papastergiou, M. (2009). Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers & Education*, *52*(1), 1–12.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, *36*, 1–11.

Patton, R. M. (2011). *Games as artistic medium: Interfacing complexity theory in game-based art pedagogy*.

Piaget, J. (1976). Piaget's Theory. In B. Inhelder, H. H. Chipman, & C. Zwingmann (Eds.), *Piaget and His School: A Reader in Developmental Psychology* (pp. 11–23). Berlin, Heidelberg: Springer Berlin Heidelberg.

Pretelín-ricárdez, A., & Sacristán, A. I. (2015). Videogame Construction by Engineering Students for Understanding Modelling Processes: The Case of Simulating Water Behaviour. *Informatics in Education - An International Journal*, *14*(Vol14_2), 265–277.

Study Plans - Faculty of IT of Yarmouk University. (2017, February 10). Retrieved February 10, 2017, from http://admreg.yu.edu.jo/index.php?option=com_docman&task=cat_view&gid=154&Itemid=159

The Ministry of Higher Education of Jordan. (2016, December 6). Retrieved December 14, 2016, from http://www.mohe.gov.jo/en/pages/Statistics.aspx

The Ministry of Higher Education of Jordan. (2017, April 11). The Ministry of Higher Education of Jordan. Retrieved April 11, 2017, from http://www.mohe.gov.jo/en/pages/Statistics.aspx

Vygotskiĭ, L. S., & Cole, M. (1978). *Mind in society: the development of higher psychological processes*. Cambridge: Harvard University Press.

Wu, B., & Wang, A. I. (2012). A Guideline for Game Development-Based Learning: A Literature Review. *International Journal of Computer Games Technology, 2012*.

Yarmouk University. (2017, April 21). Retrieved April 22, 2017, from http://www.yu.edu.jo/index.php

YoYo Games. (2016, April 10). Retrieved April 10, 2016, from http://www.yoyogames.com

YU Computer Sciences Courses | Faculty of IT. (2016, December 1). [Educational]. Retrieved December 1, 2016, from http://it.yu.edu.jo/en/computer-sciences-courses

ABSTRACT

EFFECT OF GAME DEVELOPMENT-BASED LEARNING ON THE ABILITY OF INFORMATION TECHNOLOGY UNDERGRADUATES TO LEARN COMPUTER AND OBJECT-ORIENTED PROGRAMMING

by

ALAA KHALAF AL-MAKHZOOMY

December 2018

Advisor: Dr. Ke Zhang

Co-Advisor: Dr. Timothy Spannaus

Major: Learning Design and Technology

Degree: Doctor of Philosophy

The study investigated the effect of Game Development-Based Learning (GDBL) on the ability of IT undergraduates in Jordanian universities to define the concepts of computer programming and Object-Oriented Programming (OOP) and their ability to program in C++. Participants of this research were undergraduate students enrolled in Information Technology majors at Yarmouk University who took one of the two programming courses CS110: Computer Programming with C++ and CS210: Object-Oriented Programming at the second semester 2017-2018. An instruction was designed for an eight-week-long online course to allow the participants learn game development using GameMaker®. It consists of YouTube videos, tutorial handouts and assignments. The online course was concurrent with the two programming courses. The Non-equivalent Control Group design was adopted in this research. Three groups of participants were observed; those who fully-participated in the online course, those who partially-participated in the online course and those who did not participate at all who were considered the control group. Students of the two courses responded to a pre-test and a post-test at the beginning and the end of the second semester 2017-2018. Results of an Analysis of Variance and Kruskal-Wallis H Test of

all three groups indicated that GDBL experience is effective in improving students' understanding of computer programming and ability to program in C++.

AUTOBIOGRAPHICAL STATEMENT

Alaa K. Al-Makhzoomy was raised in Irbid city, north of Jordan. Received his elementary learning at Yarmouk University Model School (YUMS). He earned his Second-Secondary certificate (equivalent to High School) from Irbid Secondary School for Boys. He earned his first bachelor's degree in Computer Science from Yarmouk University in June 2004. He began working as a computer teacher for the Ministry of Education of Jordan. He received his second bachelor's degree in Computer Information Systems in August 2008. Later that summer, he traveled to the US to pursue his graduate studies.

Alaa graduated from Cleveland State University in 2010 earning his first master's degree in Computer and Information Science. He then went back to Jordan to work as a part-time computer science instructor in two different universities; Jordan University for Science and Technology (JUST) and the Hashemite University (HU). One semester later, he was hired as a lecturer in two different academic departments; Computer Science and Software Engineering at the World Islamic Science and Education University (WISE). He worked there for two years.

Alaa was nominated in October 2012 for a scholarship from Yarmouk University of Jordan to pursue a Ph.D. degree in Educational Technology at a highly-ranked university in the US. He received a Master of Education with a major in Learning Design and Technology from Wayne State University in Winter 2017 working towards a Ph.D. in Learning Design and Technology which he earned in December 2018.

Alaa has been working as an adjunct faculty teaching many disciplinary subfields of computer science at various institutes in the U.S, especially Wayne State. Currently, He works as an Instructional Technology Designer at the Irvin D. Reid Honors College at Wayne State University.