Building Rules

Ronnie Goldstein[1], Ivan Kalas[2], Richard Noss[3], and Dave Pratt[4]

[1] Open University, Milton Keynes, UK
[2] Comenius University, Bratislava, Slovakia
[3] Institute of Education, University of London, UK
[4] Institute of Education, University of Warwick, UK

Abstract. This paper reports on aspects of the Playground project in which young children (age 6 to 8) are writing and sharing their own computer video-games. We discuss how structures in the kernel language influenced the design of one of the project's playgrounds and in turn children's thinking and use of rules. One feature of the paper is the range of children's responses to the task of translating their ideas for games into formal rules; kernel features, such as object orientation and the use of events, at times support and at other times constrain those responses.

1 Introduction

In the Playground Project[1], a team of researchers and developers from several countries is exploring how young children (6 to 8 years) construct computer-based games. Children build their games in a *playground*, a space containing game-building tools, games already built by children or developers, and sub-elements of a game, such as game objects, rules, parts of rules and scenery. At the broadest level, we are interested in how they think about the rules that underlie such games including how their meanings for such rules evolve and are expressed through the use of the virtual tools that we provide. This perspective of the children's meaning-making recognises the dialectic relationship between the design of the tools themselves and the evolution of thinking about the sorts of rules that might define computer-based games.

Our main theoretical construct is that of *webbing* [1]. The notion of webbing is strongly influenced by the paradigm of situated cognition [2] in which meaning, as constructed in everyday experience, has little to do with generalised theory and much to do with strategy constructed *in situ*, and based heavily on the structuring resources available in that setting. Noss and Hoyles propose a network of links, encompassing not only individual meanings but also resources external to that person.

The idea of webbing emphasises that learners come to construct new knowledge by forging and reforging internal connections through the interaction of internal and external resources during activity and in reflection upon it, and so reinforces the dependence that such knowledge inevitably has upon the particular attributes of those

[1] EU ESE Project #29329: see http://www.ioe.ac.uk/playground.

tools as understood by children [see 3 for a seminal analysis of how new representational forms may contribute fundamentally new kinds of literacies]. Representational systems are constitutive of knowledge construction, and new kinds of systems might at the very least, make available classes of knowledge that are unreachable without it [see 4].

A key facet of design for digitally-based systems is the affordance to inspect and manipulate the mechanisms beneath the surface — this is, incidentally, genuinely unique to digital technologies, and does potentially change the situation with respect to learning. The challenge is to build systems, which support the construction of objects and ideas (together) by the learner (this is a restatement of the 'constructionist' programme eloquently stated by Seymour Papert and some others). Giving learners a sense that they can build things for themselves, and thus think about and reflect on what they are building, has been a central tenet of the learner-programming movement of the last two or three decades.

Yet the idea of children programming – constructing and manipulating executable representations of objects and relationships – has not always been straightforward. (For a history of children's programming, see Noss and Hoyles [1].) One reason for this has been the essential identification between the act of programming, and the manipulation of lines of text. Despite some genuinely novel and powerful attempts to make this situation as palatable as possible (Logo is the most obvious example), these have not always been entirely successful. This situation has now changed, and there is a sufficiently large set of examples of programming systems where the primary mode of interaction is no longer restricted to textual interface. We are trying, therefore, to design a system that is at once open and accessible, that allows the novice to operate within new worlds, yet simultaneously allows her to see what works and how it works. While we want the learner to be fluent *within* the medium, we also want her to reflect on the structures of the medium itself [see 5 for an interesting view of this problem].

The Playground Project integrates three strands of work that might conventionally have been seen as distinct stages of development. One strand (the *kernel*) is the development of programming environments that provide the infrastructure for playgrounds. In the project, there are two kernels, ToonTalk [6] and Imagine [7]. For the purposes of this paper, we shall focus on the latter, a powerful object-oriented member of the Logo family but we wish to acknowledge the influence of elements of ToonTalk on the design of the Imagine playground. Imagine was published in UK by Logotron in January 2001. A second strand (the *playground*) has been the building of an environment, written in Imagine, which young children can use to construct their own games. This new environment is called *Pathways*, an interactive, open, visual, multimedia environment. The third strand (*meaning-making*) has been the observation of children using this environment. Unusually, all three strands have been implemented iteratively and in parallel so that each informs the ongoing development of the others. We see this integrated iterative development as a natural manifestation of our interest on the intertwined relationship between tools and learning.

In this paper, we continue the theme of interlinked strands (kernel, playground and meaning-making) by discussing how each is contributing to our understanding of three closely connected arenas, the Game Formal, the Game Outside and the Game Inside, each representing a different type of activity as will be described below. It turns out the main issues to be raised about how children think about rules emerge from activity across these three arenas. The aim of this paper is to articulate these issues by describing activity on that interface. Our approach in this paper is to illustrate activity on this interface by reporting on excerpts from sessions involving two researchers (in the transcribed excerpts, R refers to either one of them) and usually two children. The two researchers acted as participant observers in a clinical interviews spread over between 3 and 6 sessions with each pair of children. The children had not previously used the software. All had occasional access to computers in school and used computers at home. The sessions took place outside the classroom. The activity and discussions of the children with each other and with the researchers were video-taped, transcribed and interpreted. The work is ongoing. Here we report on emergent ideas apparent at this relatively early stage of the analysis.

2 The Kernel: The Imagine Programming Language

The kernel for the Pathways playground, Imagine, is an innovative modern programming environment especially designed for children to explore and develop. It is a powerful tool for designing open visual applications for education and entertainment.

Imagine is a new 32-bit version of Logo. [See 7 for more details.] We report here only those features of the kernel, which have directly supported and influenced the process of the design of Pathways.

- Substantially strong direct manipulation tools are provided in Imagine, which supports each direct step of the manipulation by an unambiguous language construct. This approach has turned out to be a productive choice for Pathways because the entire playground is a language of direct manipulations – through simple visual structures (rules) children express the behaviour of each object-actor of the game. Each rule is internally represented by a Logo language construct in a relatively straightforward way.
- The object-oriented metaphor is built into Imagine in a non-intrusive way. Imagine contains a set of primitive classes. The user can create either instances of those classes, i.e. individual objects, or define his or her own classes by further specifying already existing classes and creating instances of new classes. Moreover, the user can create an instance, then develop its behaviour, then use this individual object as a prototype and create instances based on that particular instance. This provides for easy and intuitive multiplication of objects. Later on we will present how this concept is implemented in Pathways through the magic wand tool.
- Event-driven programming considerably simplifies specifying reactions to events such as clicking an object by the mouse (any button), pressing a key, dragging an object, colliding with another object, or controlling an object by a joystick. There is a set of standard events recognised by Imagine. However, creating user-defined events can extend that set.

· Parallel processing is inherent. An Imagine project usually consists of several objects (actors), which follow their own behaviours. Although they may interact (e.g. through events) significantly, each of them may be controlled by its own independent process running in parallel with others.

· Imagine includes the facilities for text to speech and speech to text (if a speech engine is installed in the computer). To realise the voice output, the `say` command with any text input is used. Imagine, in co-operation with the speech engine, can also analyse the voice input, use the user-defined voice-menu to transform it into the associated Logo instruction and run it.

3 The Game Formal: The Pathways Playground

Pathways is a formal iconic language, which aims to make the process of building games accessible to very young children. The rules created by children in Pathways define the game that has been built. We refer to this definition as the Game Formal.

The software allows children to choose from many different background screens and to place a number of objects on the screen (see Figure 1). The objects can be given various shapes and they can be made to move automatically or to be dependent on an input device. An underpinning hypothesis is that young children can create complex behaviours by teaching objects a relatively small number of simple rules expressed in an iconic language. An object can hold several rules. For example, the pathways in Figure 6 apply to an object that has been given the shape of a tiger. The two rules combine so that the tiger is controlled by the joystick and when button 1 of the joystick is pressed, the tiger jumps upwards on the screen (see the translation in the third column of Figure 6). Children move from defining rules to playing the game by clicking a switch that turns the game on but they can return at any time to making rules by switching the game off again.

Fig. 1. This section of the Pathways screen shows two boxes that were opened from the icons in the control bar at the foot of the screen. The Toolbox at the top of the figure contains the following icons: a wand for copying; a bomb for deleting; a star for changing various settings of an object including its shape; a microphone for giving comments to objects and the mouth for speaking comments. The opened Stonesbox (below the Toolbox) is currently revealing some of the when stones. The when stones represent conditions such as "when I am touching". The Stonesbox also contains do stones that represent actions such as "I jump" and behaviours that allow an object to be controlled by the mouse, joystick or keyboard. There are further icons in the control box to open a stored game, change the background, to create a new object in a game and to switch a game on or off.

4 The Game Outside

The games to which we refer in this paper were created by children of age 7 with some support as appropriate from the researchers. We refer to such a game as the

Game Outside to distinguish it from its formal definition, the Game Formal, and the rather less precise imagined game that takes place in the child's head (discussed in the next section). In Figures 2, 3 and 4, we briefly describe the games referred to in this paper.

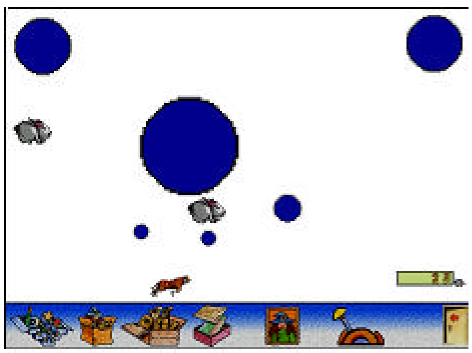


Fig. 2. Matthew and Hannah built this game, Dodgerz. The player controls the tiger with a joystick and he has to catch the rabbits as quickly as possible. The rabbits move automatically around the screen, bouncing off the walls. (Matthew and Hannah referred to the circles as walls.) When the tiger touches a rabbit, the rabbit disappears and a sound is made. If the tiger touches a wall, one of the rabbits that has already been "eaten" by the tiger, may reappear. (We have removed the grassy background in the game in order to improve the clarity of the picture.)

Fig. 3. A section of Harry and Laura's Dome game is shown. One player controls the clown with a joystick while a second player controls the shark with the mouse. The turtles move automatically and randomly around the screen. The players compete to collect the six spikes (only four are shown here) from the roof of the Millennium Dome before the turtles explode them.

5 The Game Inside

When young children work with Pathways, the ideas that they have in their heads about the games they want to create are very important. They are important to the children because their work belongs to them and they are not simply regurgitating what adults have told them to do. They are important to their teachers because of the motivation that this ownership stimulates. And, not insignificantly, they are important to us, the researchers, because we want to learn about the children's thoughts as they engage with the entire process of creating a game.

Before children have started to use the computer we tell them that they are about to design their own games and their imaginations run riot. Of course their thoughts are constrained by their experience and the games they want to design are usually based on recent experiences or on video-games they have played before. When Laura and Harry were about to start Laura said she would like to create a Millennium Dome

Fig. 4. A section of Eleanor's game is shown. This game is more like a story in which the cat pushes the meat (with its bottom!) towards the dog. When the dog and the meat touch, the meat disappears, a sound is made, and the cat returns.

game. It became clear after further discussion that Laura had recently visited the Dome. She could imagine a game in which the aim was to collect the spikes from the top of the Dome and transfer them to certain zones. The imagined game went further into fantasy as they suggested that their characters (i.e. the ones they controlled) might be a dog for Laura and a fish for Harry.

As they progressed, Laura and Harry obviously learned many things about Pathways and what they might be able to achieve using the software. We had written a simple game to start them off and they had begun by changing the rules for the dog. This bottom-up approach is not inappropriate but, even after working with some of the rules for more than one session, their fantasised game still had a lot of meaning. In the excerpt below Harry had been immersed in a discussion with Laura about certain rules and which objects should own those rules. Suddenly he surprises us:

1. H: Because that big spike … I wonder what would happen if we get all the spikes. I wonder if the game will actually finish.

In fact Harry repeated that the game might have automatically stopped when all the spikes had exploded despite no rule having being written to that effect. The imagined game often seemed to interpose with the game being built. An appropriate ending for the Dome game could have been when all the spikes were exploded and Harry talked as if this might have happened automatically. Was Harry just articulating this imagined ending, was he assuming that the initial game he and Laura had been given contained this ending or did he really think this could have happened without his instruction?

At a still later stage their thinking about the game seemed to be entirely governed by the rules that were to be given to the objects of the game. Harry and Laura were discussing how their game might progress and the phrases they were using were close in their form to the rules as they would have been expressed by the mouth in Pathways. They seemed to have a very clear appreciation of the relationship between the Game Inside and the Game Formal.

2. H: It could be a two-player game. When, whoever gets more, the most, spikes in there, wins the game.
3. H: The turtle could be one and the shark could be the other.
4. L wants to use the dog and the shark from last time.
5. L: When the shark touches the spikes, it could explode. We could tell the shark or dog to hide when it gets touched and then to show again.
6. R(esearcher): Say that again slowly.
7. L: When the shark touches the spikes, it could explode.

But the facility to talk in terms of the formal rules of the software does not develop so fast in all young children and there does not always seem to be a clear-cut distinction between what they want to happen and what will happen. At some points the children's thinking is dominated by their fantasies. At other times they are constrained by the computer and what they can achieve using the tools of Pathways. It may be that some children who are not so comfortable with formal rules prefer to spend more of their time in their fantasies. We have seen children who have given a few rules to some objects, play their 'computer games' with many rules that are still only inside their heads. We do, however, expect a transition so that, as time progresses, the computer becomes more dominant and there are many more Game Formal episodes. In this respect, however, as in most other respects, learning is not a simple linear process.

6 On The Interface

From our point of view, the most interesting activities concern the movements between the Game Inside, Game Formal and Game Outside. In this section we focus on the negotiation that takes place when children try to implement their ideas and when they find that the game they have built is not quite what they had imagined. We wish to focus on two structures in the kernel that were exploited when designing Pathways and which, it turns out, have implications for the activities that take place.

6.1 Working with Objects

When children write instructions in Pathways they choose an object on the screen and put the appropriate stones into one of the pathways for that object. Rules are not written in some general rule-factory area and given to the computer; they are written for particular objects. This is emphasised by the mouth tool in the software which, when placed on a stone or a rule, will speak in the first person singular: 'I move forwards' or 'when I am touching the dog, I hide'. The software's object orientation and the parallel processing that goes along with it have many consequences for children who are thinking about their game and learning to write rules formally.

Where is the rule? We have instances in our data where children have written a rule that they subsequently decide to alter but they are unable to locate it without opening the pathways for each of their objects and inspecting them. In the following excerpt Harry and Laura decided to create more zones in their game by copying the one they already had. They covered the spikes with the zones to hide them. They played the game and found that when the shark picked up a spike and carried it to the zone, the spike blew up. They began to look for a rule that caused the explosion. They looked in one of the zones. There was no rule.

8. L: Maybe it was that one. Go to the other one.
9. H open up the box of the next zone. There are no rules again.
10. H: It hasn't got any rules.
11. H opens up another box belonging to a different zone. There are still no rules.
12. H: I don't think they have got any.
13. They check the remaining zone. Still no rule.
14. R: Where is this rule that you're thinking of that makes the spike go off?
15. There is no response.

Harry and Laura wanted to remove the offending rule but failed to find its location. They had forgotten, or were not able to reconstruct, that the rule belonged to the spike itself. This issue is a clear example of how sometimes the object orientation of the software can be problematic.

Where does the rule belong? We have seen several instances of children having to work hard at where a new rule should be written. Often it is clear and straight-forward which of the objects in the game needs to be given a certain rule. For instance, the rule may only concern one object. But rules may also involve more than one object and this makes it more difficult to decide where the rule belongs.

The excerpt below took place shortly after the previous one. Harry and Laura were discussing how to insert their own rule to explode the shark. The current rule, "when I am touching the shark, I blow up", belonged to the spike.

16. H: When the shark touches the spike it blows up
17. L: When the shark touches the …
18. H: touches the spike
19. L: Yeah, when the shark touches the spike. (L picks up the "when I am touching stone", presumably in preparation to give it to the shark.)
20. L: When I touch…the shark isn't it. So I don't particularly need that (L puts the stone back into the stones box). I just need to change that (pointing to the input to the stone in the spike's pathway).
21. L clicks on the stone and changes the input to a shark.
22. L: When I touch the shark …
23. H: I blow up. The shark blows up.
24. L: When I touch the shark, I bomb.
25. H: Yes, but that will mean that bombs, because we're on the …
26. R: It will mean what bombs?
27. H: The spike.
28. R: Because the spike is "I" at the moment.
29. L: The spike would get bombed. The shark wouldn't. I think we need those rules in the shark.

Harry and Laura stated their rule in many different ways. Their language alternated between the spike being the object that owned the rule and the shark being the object. This oscillation reflected a lack of clarity about where the rule should go. They started by putting the rule into the spike's rule box but in the end Laura expressed the thought that it would have been better to have given it to the shark since it was the shark that they wanted to explode.

The same rule for several objects or several different rules for one object. There are occasions when there is a choice to be made: whether to give an identical rule to each of several objects or to give a number of similar rules to one object. Matthew and Hannah had been working together building a maze game but in the excerpt below Matthew was on his own. He had several walls on the screen and each of them was a different object. He wanted the game to be stopped when the tiger (controlled by the joystick) touched any of the walls.

30. R: What other penalties could you have when you touch the wall?
31. M: I've just thought of the worst one ever. The game stops… Oh no. Which would I have to change the rule on. The tiger or …(Long pause) It's these (pointing to the wall). I need to do it there. So it's only with one thing… I need

to do it to each of the walls because if the tiger will only go into one wall it will happen. If it goes into a different one, nothing will happen.

32. R: So if the tiger's rule was "When I go into a wall", say again what you think the problem would be.
33. M: Uhm, if it was the tiger that said if I touch the wall, the game stops, it can only do one wall. So I need to do it with each of them (the walls) the tiger.
34. Matthew opens up a piece of wall, erases the bounce stone, and places the 'stop the game' stone after the 'when' stone (see Figure 5).

He then copied the rule from the one piece of wall to all the others.

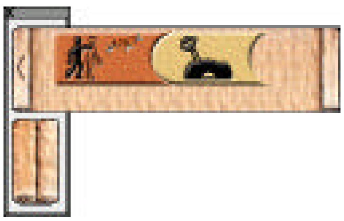


Fig. 5. The rule for each of the walls

This excerpt is quite extraordinary. Matthew appreciated that he could give the rule to either the tiger or to all of the walls. He determined that giving the rules to the walls was more efficient since he could copy the same rule for each wall; if the tiger had been given all the rules, each of them would have needed to be slightly different (one for each section of wall).

In the specification for a future version of Pathways it will be possible to identify objects with a particular shape as the input to stones like "when I am touching". A great advantage then will be that in situations like the one described above, only one rule will be required for the tiger: "When I am touching any wall stop the game".

Confounded rules.

When children are writing a number of different rules for several different objects some of them may be contradictory or have some unexpected influence on other rules in their program. At this stage in our research we cannot definitely say that inconsistencies are encouraged but we do have examples to suggest that this may be the case. Hannah and Matthew built the following rules for the tiger in their game (see Figure 6). The two pathways in the tiger's rules are, in a sense, contradictory: pressing the button of the joystick has no visible effect on the screen since the path of the tiger is, in any case, controlled by the joystick.

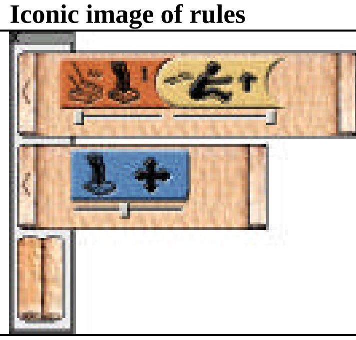
Object	Iconic image of rules	Translation
 Tiger		When joystick button 1 is pressed, I jump up 30. Follow the joystick in all directions

Fig. 6. Matthew and Hannah taught the tiger to jump whenever the joystick button was pressed and to follow the movement of the joystick.

6.2 The Language of Rules

We have discussed how the kernel provides structures for identifying the occurrence of certain events such as when a joystick button is pressed. The availability of events in the kernel was one of the inspirations for the particular implementation of the way rules are expressed in the Pathways language. Consider in more detail the example above. The command, `OnJoystickButton1Down [setycor` ycor + 30`]` in Imagine is represented in Pathways as in the first rule in Figure 6.

Although Pathways does not contain a large collection of when stones, we realised as we worked with children that the sort they needed for their video-games did not match the events provided within Imagine. In some cases it was possible to create new events but often it was necessary to simulate events in Pathways. For example, "When I am near" has no corresponding event in the kernel and so it has been simulated. In order to achieve this it was necessary to create separate modes for playing the game and for editing it. Thus the child now needs to signal that she wishes to leave the editing mode and enter the play mode of Pathways. When she switches the game on, the rules, including the simulated events, are executed. In fact, it turned out that there were benefits in separating edit and play modes since entering the play mode signals an intention to move from the Game Formal to the Game Outside. This is a step that is usually associated with a desire to test the changes made or it might simply be to have a break from the demanding aspects of focussing on the rules. Below we discuss two main issues relating to the language of rules.

Modelling Natural Language. The design of Pathways was intended to model natural language, though obviously the latter is infinitely more flexible. It was felt that by using an iconic language that corresponded to one way of phrasing rules in natural language, children might find it easier to talk to each other about the rules and the transition from Game Inside to Game Formal might be facilitated.

As with natural language there is a grammar in the language of Pathways. There are certain features that were included in the design of Pathways to highlight particular aspects of the grammar. For instance, conditionals and actions are separated into when stones and do stones, and this separation is marked by differing colours and shapes. The shape of the stones is such that a do stone can fit to the right of a when stone or to the right of another do stone; a when stones does not fit to the right of a do stone. As we suggest below, these helpful features are not sufficient for all children to appreciate the grammar of the rules in Pathways – indeed it seems that there are much deeper psychological pro-cesses involved – but we have evidence of children using this grammar to help them formalise their ideas.

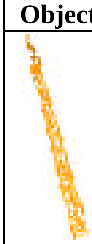
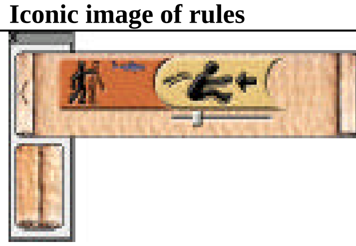
Object	Iconic image of rules	Translation
 Spike		When I am touching the shark, jump left

Fig. 7. The children wanted the spike to jump whenever it touched the shark.

In the following excerpt, Harry and Laura were building their Dome game. The player's character was a shark. At this stage in their activities, they wanted their shark to jump whenever it touched a particular spike and they were considering a rule to be given to that spike.

35. L: When I touch the shark, I jump.
36. Harry begins to put the jump stone in first. Laura stops him.
37. L: No, don't do that first, because we …
38. Harry reacts by putting the jump stone back and looking for the "when I am touching" stone. They place the stone in the blank pathway for the new spike. They alter the input so that their when stone refers to the shark.
39. R: What does that bit of the rule say so far?
40. L: When I touch the shark, I …jump.
41. R: You want it to jump.
42. They find the jump stone and place it alongside the conditional when stone (see Figure 7).

Laura prevented Harry from inserting the do stone (line 37) because she recognised that the grammar of a rule had to be of the form 'when' followed by 'do'. Harry seemed to share this way of thinking about rules and he appreciated the reason for Laura's intervention without it having been articulated.

The next excerpt shows how Matthew was aware of the grammar of a rule. Matthew and Hannah were working on the rules for each piece of wall in their maze. They wanted the wall to send a message to the score. Later they realised that this should happen when the player's character, a tiger, touched the wall but at this stage they were focussing on the message rather than the touching.

43. M finds the when stone with the envelope on it.
44. H: No, not that one. That's receive a message, silly.
45. H tells M to go back to the do stone with the envelope on it.
46. M: So we are having … what stones are they? Aren't they do stones?
47. R: Yes.
48. M: So we are having a do stone before a when stone?

Hannah understood that the wall should send a message rather than receive it (line 44). Matthew on the other hand knew that he should insert a when stone first (lines 46 and 48). The important issue for us is that Matthew, though using the wrong stone, was appreciating the grammar of the rules in Pathways.

The kernel includes the facility for changing text to speech (as well as speech input) and this was exploited in Pathways through the provision of a mouth tool. This tool can be picked up by the child and clicked over any rule or stone so that a message is spoken by that rule or stone in the first person singular. This facility was used by the children on frequent occasions and we conjecture that, along with the above support, the mouth tool provides a means for children to model the language they use on that used by the system.

Given the iconic nature of the Pathways language and its closeness to natural language, it would be easy to assume that children find the experience of working with this language quite comfortable. In fact, our experience of working with children has raised a particularly important issue about the transition from Game Inside/Outside to Game Formal.

Moving from Narrative to Conditional. The When/Do structure that Pathways uses for rules is a particular type of conditional statement. Suppose you are observing a hammer moving across a screen. You notice that it moves on top of a gong and a sound is played. As an observer the above account might be a reasonable story of what happened. Even as a player where you control the hammer with a mouse, a similar account seems valid. You might relate the story as "I moved the hammer across the screen then it touched the gong and then there was a sound." We describe this type of account as narrative. Now suppose you are programming the computer to display this effect in Pathways. The simple linear narrative of how the hammer moved towards, then over the gong and a sound was made could easily have been written in a conventional language such as Logo. However, when the story becomes a game, a narrative approach to programming the game will not suffice. To implement a player controlled game, you need to use events. In Pathways, you need to teach the hammer a rule such as "When I am touching the gong, I make a sound." The perspective of the programmer of games is quite different from that of the observer or player of games.

Children's experience of games is usually as an observer or player. Thus one focus of our research is to observe whether such young children are able to take the programmer's perspective and manipulate When/Do type structures. What do we mean by 'manipulate' here? An essential ability for the programming of games is to transform ideas belonging to the Game Inside into the formal rules of the Game Formal that instantiate the Game Outside. However, this transformation works in both directions, so as children work, they are inspired, perhaps by the discovery of new stones, or by the inevitable constraints of the Pathways language, to renegotiate the Game Inside.

It turns out that some children adopt the programmer's perspective remarkably easily. This excerpt took place at the beginning of Matthew and Hannah's second session with Pathways. They had decided that their character, a tiger, should jump around their maze.

49. H: How do you make the thing jump?
50. M: That's what we need to think about.
51. H: When you press …
52. M: If you press …
53. H: When you press a button, it jumps or something.
54. M: When you press left button, I jump.

Matthew and Hannah discussed the rule needed to make the tiger jump in a language closely corresponding to that of Pathways, even though they had not at that point instantiated the rule using the icons. Matthew and Hannah are able children who appear to think in terms of the formal language they need to use to write rules. Any transition that might be happening between Game Inside and Game Formal is very fluent and it is not easy to find times when a conditional account is appropriate and they make any use of narrative language.

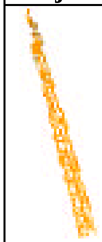
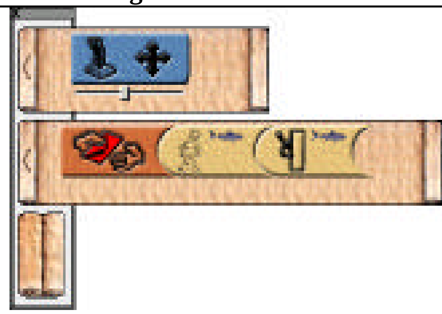
Object	Iconic image of rules	Translation
 Spike		Follow the joystick in all directions When I receive the red message, hide the shark and show the shark.

Fig. 8. Harry and Laura wanted the shark to flash.

In the following excerpt, Harry and Laura demonstrate how they were able to distinguish between the use of narrative when in the role of player and conditional language when acting as programmers. At this point in the development of their Dome game, they had just programmed one of the spikes to make the shark flash when it received a message (see Figure 8).

55. L: But it goes (she makes a funny on and off sound here) doesn't it?
56. H: It keeps on flashing.
57. R: Can someone explain to me, because I really haven't a clue, what this second rule means.
58. H: When the spike sends the message, I hide and then I …
59. L: It hides the shark then it shows the shark.
60. R: In the game what will it do?
61. L: It will hide, then it will show it, then it will hide, then it will show it (L makes the funny on off sound).

In line 58, Harry responded in conditional language to the researcher's question about the rule by taking on the programmer's perspective, whereas in line 61 Laura used the narrative to describe what happened in the game.

Other children found the adoption of the conditional language problematic even though they understood that the expectation was for them to build a game of their own. Eleanor is a child who needed much more guidance in order to begin using the conditional language. In our third session with Eleanor, we instigated a teaching experiment in which we tried to help Eleanor to appreciate the need for when stones in rules. In the fourth session she had begun to implement an idea in which a cat would carry meat to a dog. In Eleanor's Game Inside, the dog should eat the meat. We expected that after the teaching experiment, Eleanor would recognise a solution such as giving the dog the rule: "When I touch the meat, I hide the meat". We found, however, that Eleanor continued to use a narrative account.

62. R: Tell me what you want the rule to say, in your own words.
63. E: I want it to say that the cat goes over to the meat before the dog, and the cat picks up the meat and pushes it along to the dog and then dog eats it and then they just run round the street.

Later, Eleanor was still working on the same problem.

64. E: I'm looking for a stone in here that I can fit inside that one, that will tell it to touch something.

65. R: What's this rule doing, this stone? Do you want another one or do you want the same one as this now?

66. E: I want a different one to actually tell it to touch the meat and then make it go back … It's not bringing back the meat because it's supposed to, well what I was thinking it would do was touch the meat, turn around and bring it back.

Eleanor is using narrative language without any conditional statements even though she recognises that the meat must touch the dog before there is any further action. A few minutes later, she does pick up the appropriate when stone from the Stonesbox and she solves her problem. We can not claim that Eleanor henceforth was able to move smoothly between narrative and conditional modes, as could the previous children, but there was evidence of increasing flexibility. Thus, towards the end of her fifth session, Eleanor wanted her cat to move and she suggested:

67. E: Maybe we could have When I am touching an object, I move forward 30.

Although we have only worked with a small group of children so far, it is already clear that the modes of articulation about rules vary dramatically. Matthew appeared to be firmly rooted in the type of conditional language used by programmers and he was so at ease with this mode of articulation that he rarely expressed any narrative thinking. Harry and Laura were sufficiently flexible to take on either the programmer's perspective using conditional language, or the observer/player's perspective using a narrative account. Eleanor tended to adopt the narrative account whatever the context but with sufficient guidance and support appeared able to begin using the conditional.

7 Discussion

First we summarise the main issues raised in the reporting of activity on the interfaces between Game Formal, Game Inside and Game Outside.

- Children do not always find it easy to recognise which object from their Game Inside should own a particular rule emerging in the Game Formal. Children also have occasional problems locating a rule since the program is distributed across many objects and it is often not obvious to which object certain rules should be given. It is an issue sometimes whether to give one rule to several objects or to give several different rules to one object. We have also noticed incidences of rules that (unintentionally) impact on the execution of other rules.

- The use of events is essential in the programming of games and this has led to the when/do structure of rules in Pathways. Children react very differently to the demands of translating their Game Inside into a Game Formal, which must be defined in terms of when/do rules. Some children appear to move smoothly between a narrative account of their game given from the player's perspective and a conditional account that a programmer has to utilise. Other children of this age find the conditional account much less natural than a narrative account. Such children need much more support than is given in the software alone. We are still exploring what type of extra support might help children with the transition from narrative to conditional language.

The story of how Eleanor, Hannah, Harry, Laura and Matthew were able to create such interesting games is essentially a story of webbing, in which the combination of the external resources in Pathways and the inputs by the researchers are juxtaposed with the internal resources of the children. The stories highlight the contrasting nature of those internal resources. In Matthew's case they are powerful and they allow him to tune [REF] very quickly into the external resources available in Pathways. Harry and Laura's internal resources also enabled them to employ the formal language of Pathways in a creative way. In contrast, Eleanor's resources seemed to limit her to narrative accounts. Eleanor was eventually successful in beginning to use conditional language effectively. In webbing terms this indicates a reforging of connections. We expect, given the nature of our theoretical framework, that this new knowledge about conditionals is situated within the Pathways context. We also recognise that Eleanor's new situated knowledge remains potentially available as an internal resource for future meaning-making. However, for this potential to be released so that she might draw on these experiences in non-Pathways contexts, further support would need to be offered through the external resources of that new situation. We see the role of the researcher or teacher as critical in the webbing both within and beyond Pathways. Our focus with children such as Eleanor is to gain a better understanding of how their thinking about rules might be reforged in the light of such support.

The insights gained from children's activities sometimes cause us to reconsider the design of the playground and, to accommodate such changes, the kernel itself may need to be modified occasionally. The management of such deeply embedded iterative design has been difficult and at times painful. Nevertheless, there have been and continue to be tremendous gains in terms of the creation of tools that are cognitive in nature, in the sense that they are designed to support the activity that takes place on the interface between the Game Inside and the Game Formal.

References

1. Noss, R. and Hoyles, C.: Windows on Mathematical Meanings: Learning Cultures and Computers. Kluwer Academic Publishers, London (1996)
2. Lave, J.: Cognition in Practice. Cambridge University Press, Cambridge (1988)
3. diSessa, A.: Changing Minds: Computers, Learning, and Literacy. MIT Press, (2000)
4. Kaput, J., Hoyles, C., and Noss, R.: Developing New Notations for a Learnable Mathematics in the Computational Era. In L. English (Ed.), Handbook of International Research in Mathematics Education Lawrence Erlbaum(in press)
5. Hancock, C.. The medium and the curriculum: reflections on transparent tools and tacit mathematics. In A. diSessa, C. Hoyles, and R. Noss (Eds.), Computers for Exploratory Learning Springer-Verlag. Berlin. (1995) 221-240
6. Kahn, K.: ToonTalk - An Animated Programming Environment for Children. Journal of Visual Languages and Computing. June (1996)
7. Kalas, I. and Blaho, A.: Imagine… New Generation of Logo: Programmable Pictures. In Proceedings of WCC2000. Beijing. (2000)