

Face-to-face and online collaboration: Appreciating rules and adding complexity

Richard Noss, Celia Hoyles, Jean-Luc Gurtner, Ross Adamson and Sarah Lowe

The Playground Project
Institute of Education, London, UK

Abstract

This paper reports how 6-8 year-old children build, play and share video-games in an animated programming environment. Children program their games using rules as creative tools in the construction process. While working both face-to-face and remotely on their games, we describe how they can collaboratively come to *explain* phenomena arising from programmed or 'system' rules. Focusing on one illustrative case study of two children, we propose two conjectures. First, we claim that in face-to-face collaboration, the children centre their attention on narrative, and address the problem of translating the narrative into system rules which can be 'programmed' into the computer. This allowed the children to debug any conflicts between system rules in order to maintain the flow of the game narrative. A second conjecture is that over the Internet children were encouraged to add complexity and innovative elements to their games, not by the addition of socially-constructed or 'player' rules but rather through additional system rules which elaborate the mini-formalism in which they engaged. This shift of attention to system rules occurred at the same time, and perhaps as a result of, a loosening of the game narrative that was a consequence of the remoteness of the interaction.

Keywords

children's programming, rules, computer games, collaborative game construction

'Wherever there is modularity there is the potential for misunderstanding: Hiding information implies a need to check communication'. Perlis, A. (1982).

Introduction

Since the work of Lave and Wenger (1991), views of learning have tended to emphasise the importance of participation in communities of practice. It is also acknowledged that groups or institutions can learn by making visible the work undertaken by each of its components provided this work is open to comment within the community. This sharing of a public or semi-public entity within a community meshes nicely with the constructionist paradigm (see Harel and Papert, 1991), which similarly stresses the importance of building and sharing knowledge within a community, and successfully incorporates this into an effective model of learning and teaching. Within such communities, not only can less experienced members learn the accepted behaviours, codes and rules by progressively taking on more responsibilities, but also new

forms of knowledge can emerge through collaborative exchange that extend even the most advanced members of the community. In other words, the total of the community's knowledge, instantiated in one or more entities, is more than the sum of its constituent components.

An important part of the progress in our understanding of how this paradigm works has come from sociological studies of the way the scientific community is producing knowledge which have been shown that, even more than individual or team efforts, the open process of publishing, critiquing, exchanging, debating ideas significantly contributes to the growth of science (Latour 1987; see also, Roschelle, 1996). Similarly, in an explicitly educational field, there have been attempts to build school settings that are modelled on the paradigm of intellectual communities of practice (see, for example, Brown & Campione 1990, Scardamelia & Bereiter 1996).

Not surprisingly, information and communication technologies have become a crucial component of all these directions, by opening up new facilities to share and discuss knowledge, both synchronously and asynchronously, face-to-face and at a distance. However, much of the work on collaborative learning has focused on groups or communities in which all participants know each other and work on the same site. Technology, and the internet in particular, now allows interactions to take place without direct eye or body contact. Yet even with broadband access (and most internet-based communication in learning environments is *not* broadband) it is not clear what is gained and lost in such interactions, compared with face-to-face interaction. Building a picture of this is clearly a long-term goal, and no single study can do more than add one more piece to an already-complex puzzle. It is, however, relatively clear that the ways in which interaction takes place and its ultimate role in mediating what participants think and say, must surely be highly contingent on the knowledge domain under consideration, the tasks, activity structures and situational specificities of the community and the software and hardware employed that mediates and shapes what can and cannot be exchanged.

There are well-documented claimed benefits for collaborative contexts for learning, but also an acknowledgement that the benefits of peer-presence are not universal and may vary across tasks and with individual students (see Hoyles & Forman, 1995; Howe, 1993), but students in general learn more effectively in collaborative settings when utmost care is taken in the design of the task and activities (see Hoyles, Healy & Pozzi, 1994).

So a fundamental issue for research is to design collaborative environments that encourage the construction and sharing of knowledge effectively using technology (Bell & Winn, 2000; Koschmann 1996). Moreover, we would like to contribute to the literature that analyses how knowledge and learning is collaboratively achieved in learning situations mediated by digital technologies, and seeks to identify the mechanisms that make collaborative settings effective (see, for example, Dillenbourg P., 1999).

We begin by briefly outlining the context of our study, the "Playground Project" and then describe the episodes that form the corpus of data on which we draw for the remainder of the paper.

Background to the study: The Playground Project

Attempting to define how technology can support knowledge building by building a community of learners in schools, Scardamelia and Bereiter (1996) set up a list of important pedagogical attitudes in order to achieve this goal. Among these attitudes are: provide opportunities for peer commentary and notification, allow entry points for all ages and ability levels, and maximise the interplay and value of different communication modes.

In the Playground Project (www.ioe.ac.uk/playground) most of these issues are instantiated in the design of our computational environments where children aged 6-8 build, modify and share

games using the formalization of rules as creative tools in the constructive process (see Hoyles, Noss, and Adamson, 1999). We call these environments 'Playgrounds'. Our research is framed by two main questions: How are children's understandings and expressions of rules mediated by a programming language in which the rules are available for inspection and change, and expressed as animated 'code'? How are children's understandings and expressions of rules mediated by collaborative activity whether face-to-face or remote, and what are the transformation processes that occur during children's experience of different rule types during game playing and changing? In this paper, we focus on the second question, although we will inevitably touch on some aspects of the first.

Our project is to design and try out computational worlds — playgrounds — in which the objects in a game *and* the means for expressing them are engaging; where the programming of a game is itself a game. One key design choice was that of a programming platform on which to base the playground elements. We wanted to open access to all layers of the system to children, rather than pre-empting any possibilities by closing off the platform itself; and we wanted to achieve just the right grain size of system elements which would afford straightforward access and functionality as well as being reasonably clear what *mechanisms* lay behind it. We decided to base the design and construction of playground on *ToonTalk* (Kahn, 1999). Its appeal for our purposes is in its "concretisations" of computational abstractions based on the animated metaphors of the computer game. Program sources are animated, tactile, enhanced with sound effects, and based on a pervasive metaphor of programming (training by example) animated robots. Details of how ToonTalk works, its design principles and some applications can be found at www.toontalk.com.

We hope the reader will gain some appreciation of the design by reading the subsequent descriptions, although it is difficult to describe the system in a static medium. Crucially, we have designed sets of *behaviours* of the play objects which inhabit the playground world. These consist of open 'components' which are modular in design, and which contain their mechanism on their 'flipside' (any object in the playground may be flipped over and its workings inspected). In order to make sense of what follows, we will briefly describe how behaviours work, and one of the other mechanisms – message passing – which will become an important part of the collaborative story as it unfolds.

Open components -- Behaviours

At the simplest level, a behaviour is a ready-made piece of programming code that can be (re-) used, inspected, or combined. In principle, this means that an object which is grabbed (from anywhere) and which seems to have an interesting property (it bounces, it makes a noise when it hits something etc.) can be used *as it stands* to copy (or cut) its behaviours to another object, or to share a subset of them (e.g. move horizontally with the mouse, but not vertically). Behaviours are part of the object they control – this object-oriented connection between game component and its code means that programming and debugging can be directed by higher level structures.

Behaviours have multi-modal representations of their functionality in the form of text which speaks when pointed at, and graphical (mostly animated) descriptions. Figure 1 shows a behaviour which makes an object (any object it is placed on) move to the left when the shift key is pressed. It consists of the ToonTalk code (a robot and its input box which has been programmed by example), a speaking text description and an iconic (in this case, not animated) representation of the behaviour.

INSERT FIGURE 1 HERE

In order to achieve a useful level of functionality, we combine behaviours into ‘animagadgets’ and place them in the playground world on suggestively-designed backgrounds. Animagadgets may combine several behaviours, grouped together to suggest elements of common functionality. In Figure 2, a set of three behaviours is illustrated: I bounce up and down, I bounce left and right, I bounce off the edges. When the child switches the gadget on, she can see a dynamic representation of the behaviours involved.

INSERT FIGURE 2 HERE

To see how any behaviour works, the child needs to suck the chosen object off the background (using a special cutting tool consisting of an animated vacuum pump) and flip it over – as we stated above, *all* objects contain their functionalities as programmed descriptions on their flipside. To fix ideas, let us suppose that a child wants to inspect (and then use) the ‘I bounce up and down’ behaviour. For the moment, we will further suppose that the child is working in England (as we will see, this makes a difference!).

All behaviours have text descriptions attached. We have added the further modality of speaking text because such young children do not all know how to read fluently. Our child will see the robot which makes the behaviour ‘work’, a picture which suggests the functionality, and the speaking text (see Figure 3a). In fact, reusing the component or behaviour can be achieved by copying the whole grey box, or – and this sometimes happens when modifications need to be made to the robot – simply copying the robot and its ‘input box’ (just visible as a two-element rectangular box in Figures 3a).

INSERT FIGURES 3A AND 3B HERE

If, however, the child is sitting in Sweden, flipping *the same* behaviour over will reveal the situation in Figure 3b. This ‘translation’ of mechanism is achieved automatically, as any piece of text has a mechanism attached which senses the ‘nationality’ of the host machine. In other words, one of the *ToonTalk* sensors built into the system detects the language currently being used, and this mechanism will choose the correct language version of the text and display it accordingly. This becomes a rather interesting situation when we come to consider message passing across countries.

Message Passing and ‘long distance birds’

When children are building games, they naturally (and quite quickly) discover a need to pass messages between objects (a ubiquitous example is the need to keep score of some event). *ToonTalk* provides a powerful concrete metaphor for message passing between objects, instantiated as a bird flying to its nest. Birds are hatched from nests, and if a bird is given any object (which, of course, may have one or more robots on its flipside), it will deliver the object to its nest and then return to its original location (Figures 4a and 4b).

INSERT FIGURES 4A AND 4B HERE

As with all mechanisms built into the Playground platform, it is straightforward to expose the workings of any given mechanism, and to exploit the strong metaphorical basis of the underlying ToonTalk language. In the case of message passing and birds, children became used to seeing how messages were passed between objects (for example, to see a bird flying to and from a score, one only has to remove the score from the game or background containing the score's (nest's) bird).

In the course of the Playground project, we made an important extension to the idea of message passing. We simply extended the metaphor so that a bird can 'fly' to its nest, even when the nest is on any other computer on the Internet running *ToonTalk*. Interestingly (but not strictly relevant for this paper) children took for granted that messages could be 'long distance', and used the mechanism extensively to send messages and swap games between sites. It should now be evident what happens when a game is swapped between sites: the translation mechanism comes into play, and the textual description of all behaviours and animagadgets magically appear in the right language. Of course, this only applies to pre-written textual description – automatic language translation is a little beyond the scope of the current project!

The activity structures and task

In early interviews with children, we asked them to tell us about their favourite games (both on and off the computer) and the rules of these games. We found, surprisingly perhaps, that most only described constraints ("you mustn't hit other children") and not the actual rules designed into the game. In what follows, therefore, we make a distinction between two kinds of (computer game) rule :

- *Player rule* – a regulation that can be ignored but is agreed (possibly tacitly) among players should not be transgressed in the interests of the narrative elements of the game.
- *System rule* – a formal expression of object relationships in a programming system, that is a programmed rule that specifies the formal condition and action for the behaviour of game elements.

We now turn to the setting in which our case study is situated. We held some 'international games workshops' at which children who have worked in pairs on their own game for several weekly sessions share their games over the Internet with children in remote sites who have done similar preparatory work.

In the sessions we describe below, we set a pair of English children working in an after-school club the following task: To build a game from scratch using a limited set of eight animagadgets. We allocated a period of four one hour weekly sessions for this task. The sessions were audio and, where appropriate, videotaped. During the final session, they worked in the University lab (international connections within the school were bedevilled by firewalls) to share their work with a pair of Swedish children. Internet video conferencing was set up to facilitate the organization of the workshop and also to give the children a sense of presence of their partners in the remote site. This turned out to have an important effect on the running and experience of the workshop. Dialogue was in English with a researcher/interpreter in Sweden, but in spite of this, most of the children cited this, using the video conferencing to talk to their partners as the most enjoyable aspect of the workshop. Long distance birds were used to send and receive games, evaluations and to communicate ideas for game improvements.

In the description that follows, we provide evidence and discuss conditions under which groups of children working separately on their own tasks can function as a community. We describe how, sharing the same tools and the same goals at different times and under given circumstances, produced outcomes that nicely match the requirements of collaborative work.

Building a game on one site

The children in both sites were given the following eight animagadgets to work with (Figure 5).

INSERT FIGURE 5 HERE

Initially, the children were given worksheets which directed them to explore the functionalities of the eight animagadgets. They turned each animagadget on and watched the behaviours play out. Next they started to map out their game, through talk, drawing and writing. They talked about the various objects and relationships between them that they wanted in their game. Then they were encouraged to start the formalisation process by drawing the game as it would appear with its components, and finally they were asked to write down the rules of the game. We present the work of Safia and Faroq who had an idea of a game with a coffin that:

Faroq: "If it touches you, [the player] you're in the coffin."

After some discussion Safia and Faroq decided on a plan. Safia listed the behaviours she and Faroq would need for their game and described how they would use them. Her verbal description of the game comprised both her imagination and references to the anima gadgets. Some of her ideas would be extremely hard to implement: this marked the start of a long process of transforming between player and system rules.

Safia: "bouncing shooting and bombing ... bombing so when you touch the coffin you're gonna die, shooting for when you are in the coffin you can shoot the ghost that took your place and bouncing for the coffin to bounce up and down ... if the person touches the coffin it dies and it makes a noise and a girl takes the place of the man. It gonna make a noise when the girl comes out and they are just gonna play. Man or girl has to get some treasure. Zombie, there's gonna be a zombie guarding the treasure."

The basic idea of their game was that the player must reach the treasure (a ring) by avoiding the coffin that 'guards' it. The coffin is a moving obstacle that simply bounced vertically off two lines across the top and bottom of the screen. They made the background for the game and added a white coffin (actually just a white rectangle), and two barriers with the bouncing behaviour on the flipside of the coffin (see Figure 6a and b.)

INSERT FIGURE 6 HERE

Next they added the player character that could be moved about, as it had the behaviour "I move in the direction of the arrow key you press" on its flipside, and finally a picture of a ring. At this point, their game is represented in Figure 7.

INSERT FIGURE 7 HERE

Safia then transformed Faroq's narrative rule *"the player going inside the coffin"* into one of the available animagadgets "I blow something up when it touches me". She decided to reinterpret the idea of "the player going inside the coffin" as simply "the player disappearing". The coffin would destroy any object that it touched, including the player, hence making it disappear.

Safia: we need the bombing behaviour on the back of the coffin so because when he touches it it's going to bomb up. The person touches the coffin right, he's gonna go inside the coffin but we're gonna make him bomb up.

Safia and Faroq added this behaviour to the back of the coffin, put the coffin back on the game and tried it out. The coffin started moving but promptly blew up the barriers when it touched them, and flew out of sight as there was nothing to keep it on the screen! Neither Safia nor Faroq predicted this would happen: The coffin blew up the barriers, the very things that were needed to make the coffin bounce. They immediately tried to work out why. Slowly, Safia started to realise that the coffin would blow up *anything* although she had not fully articulated a solution to their conflict of behaviours:

Interviewer: What happened?

Safia: It touches anything! Any object!

Faroq came up with an alternative solution. He suggested using another behaviour "I bounce off edges" to make the coffin bounce, thus removing the need for the barriers and hence circumventing the problem.

Faroq: I know what we should do, get coffin and put on stop at edge [meaning bounce off edges].

Why did they not realise the implications of adding their 'guarding' behaviour to the coffin? It may be that through their joint discourse they had already specialised the behaviour to "I make the player disappear when it touches me" because that was the focus and endpoint of their discussion and construction. In other words, relying solely on their narrative account for constructive purposes may have jeopardised their ability to determine the implications of adding behaviours (system rules).

To overcome this impasse, the children came up with different, but equally reasonable solutions. Faroq proposed changing the way the coffin moved by removing the barriers. If this were done, the coffin would bounce off the edges of the game and if anything touched the coffin, the coffin would blow it up. As an alternative, Safia eventually proposed altering the robot so the generalized behaviour "I blow things up when they touch me" would *only* blow up the player. Her solution was to go down a level to look at the robots (programming code) which she edited to specify and restrict *what* should be blown up. The coffin would now only blow up the player if they touched each other.

Safia made a further change to the game. She still wanted to preserve the idea of the player 'going into' the coffin: So she made the coffin change into the picture of a girl when it hit something by adding a behaviour "when I hit something my picture changes" to the coffin and editing the robot so the only pictures shown were of the girl (incidentally, this is a great discovery: cycling through a sequence of identical pictures looks like nothing is happening!). Her narrative interpretation of this system rule was "When the player touches the coffin it goes inside and a girl comes out", (see Figures 8 and 9 for the full game).

INSERT FIGURES 8 AND 9 HERE

Further changes were made in readiness for the games workshop by both children. First they slowed down the coffin so that it did not turn into the girl so quickly: To do this, they had to go into the robot's "thought bubble" and carefully consider (and modify) its inputs. But Safia noticed that this made the game too easy. After discussion, Safia and Faroq decided to add a "zombie" (in a spaceship) that also bounced off barriers to guard the ring. This zombie had the same behaviours as the coffin (except it did not change pictures) but moved more quickly.

Safia: ... but no that's not good because when the man goes he can easily get the treasure, and we want it hard. A zombie can be guarding the lady, the coffin.

Interviewer: maybe there should be another object moving about as well

Faroq: we wanted a zombie bouncing up and down

Safia: the coffin bouncing, the zombie guarding a larger one that would stay still like this.

Faroq made his input to Safia's changes and said he did not want the coffin changing into a girl, so they replaced the pictures of the girl with picture of aliens. The coffin now changed into different aliens when it touched something rather than turning into a girl. Their final game is shown in Figure 10.

INSERT FIGURE 10 HERE

The games workshop: collaborating online

We now describe the online games workshop in which Safia and Faroq collaborated with a pair of children in Sweden. After introducing themselves through video conferencing, Safia and Faroq sent their game to their opposite numbers with the accompanying message "why don't you find out how to get the magic ring". Shortly afterwards they received a message back via a bird: "we want to make the game more difficult. We will copy the slow character and put it in the beginning". Faroq replied with the message "when you have made it more difficult send it to us!" The changed game arrived with the message "we have added 2 slow creatures." (Figure 11). In fact two coffins bouncing rather slowly (and changing their pictures) had been added. Note that the mechanisms of the game would be Swedish in Sweden but English in England.

INSERT FIGURE 11 HERE

Faroq played first and was quickly blown up by the third bouncing object. He said 'it's hard, and it was indeed quite difficult to avoid all the lethal objects! Another message then arrived from Sweden "we are making your game more difficult and interesting." Soon they received a second version of the game, again with a message describing the changes "we have added some men, if you hit them you get blown up" (Figure 12).

This final version of the game had a picture of a minotaur which bounced off the edges of the game rather randomly and again blew up anything in its path! The minotaur had been taken from another game constructed in Sweden and simply added with all its behaviours. They gave their opinions of the game:

Safia: it's good, it's better, it's harder

Interviewer: what did they add to it?

Safia: they added a minotaur that kills anything that touches it ... it moves diagonally

Finally the English pair sent the message "you have made our game more better and harder because the minotaur kills everything but it can kill the player as well."

INSERT FIGURE 12 HERE

Safia and Faroq recognised how their game had been changed, could describe the rules underpinning the objects and appreciated how hard the game had become.

Discussion

We can contrast the type of interaction in the two parts of this brief case study. In the face-to-face interaction during game building, a strong narrative line was generated, which had to be 'translated' into system rules instantiated in behaviours. Clearly the focus of attention was in maintaining this narrative. We conjecture more generally that children are unlikely to predict rule conflicts when collaborating face-to-face, but certainly can develop them in diverse ways that maintain the sense of the story.

In the workshop, the richness of collaboration between the pair of children in London and another in Stockholm was minimal in many respects. Virtually no direct communication nor eye-to-eye contact took place and messages and spontaneous text always had to be translated. Yet it is striking that a community of children using a common representational system may, in some sense, have compensated for the somewhat impoverished interaction. Although each pair of children were given the same general task (to design a game with a fixed set of tools), they chose to interpret the task rather differently (we have not reported the Swedish children's game here). Nevertheless, the commonalities of the tools at their disposal seem to have provided enough of a shared structure that the English children at least, could straightforwardly interpret what had been done, and make sense not only of how to play the new game, but how it was built, as evidenced by their rather sophisticated critique of and response to the returned game.

The expressive power of the animagadgets as a rule-based mini-system foregrounded the necessity to consider system rules not unlike the construction of explanations in face-to-face collaboration. Yet it may be as the system rules became more complex, the narrative of the story became lost between the remote sites.

As the quotation from Perlis stated at the outset illustrates, the modularity of the added behaviours is two-edged. On the one hand, it allowed very straightforward collaboration to take place, and afforded the incorporation of 'new' developments easily into each team's work. On the other hand, the very modularity and independence of the components does not necessarily encourage deconstruction of those components, and we have (unfortunately) little evidence that the Swedish children needed to do more than acknowledge the existence of behaviours rather than engaging with them at any deeper level of mechanism.

Remoteness gives rise to an objectivity but status also plays an important part. The player has had no role in the construction therefore views the player/system rule distinction much more clearly. But the player can also be a builder. Remoteness denies collaboration in the face-to-face mode but fosters a mode of iterative feedback between pairs alternating between player and builder.

The Swedish boys increased the complexity of the game by making it more difficult to win. Although the changes were simple in terms of object manipulation both Safia and Faroq agreed

that it was more fun to play. The complexity was also increased but the narrative was perhaps lost, through the addition of the minotaur whose function in the story was at best unclear.

Concluding remarks

In this paper, we have concentrated on one short case study. Nevertheless, we have a modest set of corroborative data that leads us to the general conclusion that online collaboration of the kind we have outlined catalysed some unique outcomes. The shift from narrative to system/formal rules does, in fact, seem to be a direct result of the necessity to formalise in the absence of all the normal richness of interaction that characterises face-to-face collaboration. Moreover, the contrast is all the more vivid when we compare the children's later work with their initial constructions, in which the narrative was clearly foregrounded, and the focus of attention was necessarily the translation of the narrative into a form that the computer could accept. This initial form of engagement made it possible to debug the system rule conflict that occurred.

When the game arrived back from Sweden with various additional complexities, we saw how the children were able to interpret what had been done; in this case, the focus of attention shifted markedly onto system rules, as a means (perhaps the only means) by which they could make sense of the complexity in the absence of face-to-face conversations about narrative. (We do not, of course, know what would have happened had there been a fast video link where both sides shared a common language). This shift of attention to system rules occurred at the same time, and perhaps as a result of, a loosening of the game narrative that was a consequence of the remoteness of the interaction.

Acknowledgement

The Playground Project is funded by the EU Esprit/Experimental Schools Environments program, Grant #29329

References

- Brown, A.L. & Campione, J.C. (1990). Communities of learning and thinking, or a context by any other name. *Contribution to Human Development*, 21, 108-126.
- Cypher, A. (1993) *Watch What I do. Programming by Demonstration* Cambridge, Mass.: MIT Press
- Dillenbourg P. (1999) *Collaborative-learning: Cognitive and Computational Approaches*. Oxford: Elsevier.
- Harel, I., & Papert, S. (Eds.). (1991). *Constructionism*. Norwood, NJ: Ablex Publishing Corporation.
- Hoyles, C & Forman, E. A (eds) 'Process and Products of Collaborative Problem Solving : some Interdisciplinary Perspectives' Special Issue, *Cognition and Instruction*, 13, 4
- Hoyles, C, Pozzi, S & Healy, L. (1994) Groupview with Computers : an Overview of Findings *Journal of Computer Assisted Learning* 10, 202-215
- Hoyles, C. & Noss, R. (1999) Playing with (and without) words. Proceedings of the Fifth Eurologo Conference, University of Sofia. pp. 18-29.
- Hoyles, C., Noss, R. & Adamson, R. (in press). Rethinking the Microworld Idea. *Journal of Educational Computing Research*.
- Howe, C (ed) 'Peer Interaction and Knowledge Acquisition' Special Issue, *Social Development*, 2, 3

- Kahn, K. (1996) ToonTalk - An Animated Programming Environment for Children *Journal of Visual Languages and Computing* 7, 2, 197-217.
- Koschmann, T. (Ed.) (1996) *CSCL: Theory and Practice* Mahwah, N.J.: Lawrence Erlbaum
- Latour, B. (1987). *Science in action*. Milton Keynes, UK: Open University Press.
- Lave, J. & Wenger, E. (1991) *Situated learning: Legitimate peripheral participation*. Cambridge, UK: Cambridge University Press
- Lave, J. & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge, England: Cambridge University Press.
- Noss, R., Hoyles, C. & Adamson, R. (in preparation). *Designing for developing a sense of mechanism*.
- Perlis, A. (1982) Epigrams in Programming. ACM's SIGPLAN publication, September.
- Piaget, Jean. (1932) *The Moral Judgment of the Child*. International Library of Psychology Philosophy and Scientific Method, Kegan Paul, Trench, Trubner & Co. Ltd., London.
- Scardamalia, M & Bereiter
- Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, Massachusetts: Harvard University Press.
- Wertsch, J. (1998). *Mind as Action*. New York: Oxford University Press

FIGURES

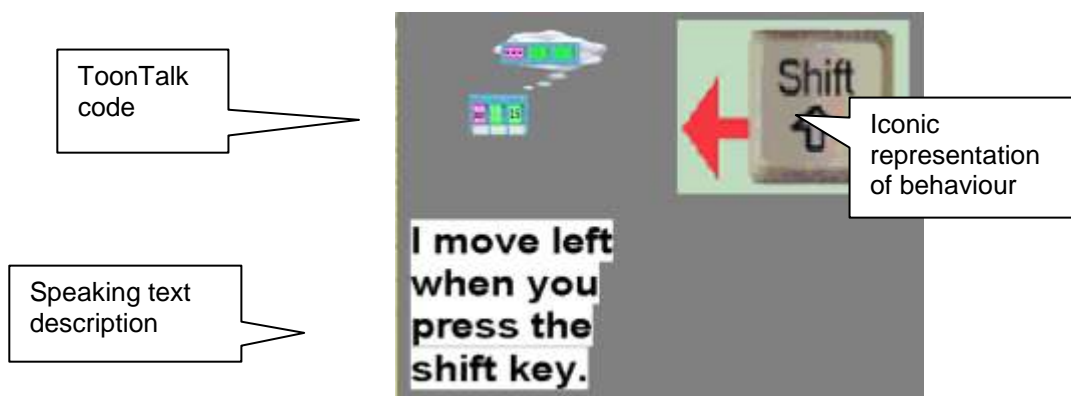


Figure 1 The multi-modal representations of behaviour functionality placed on the reverse of some object (not shown). When the object is flipped over, it will move to the left if (and only if) the shift key is pressed.

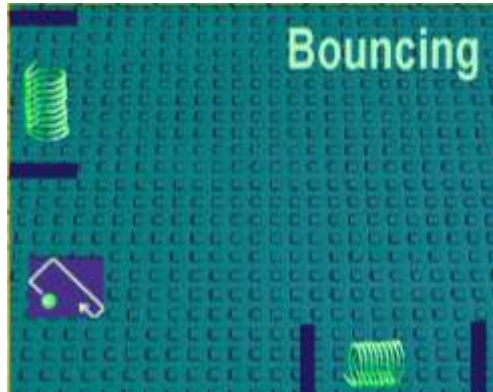


Figure 2 An anima-gadget showing the bouncing Behaviours; I bounce up and down, I bounce left and right, I bounce off the edges.

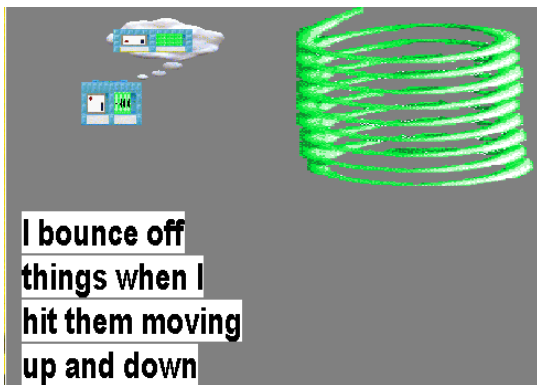


Figure 3a the vertical bounce behaviour
(English version)

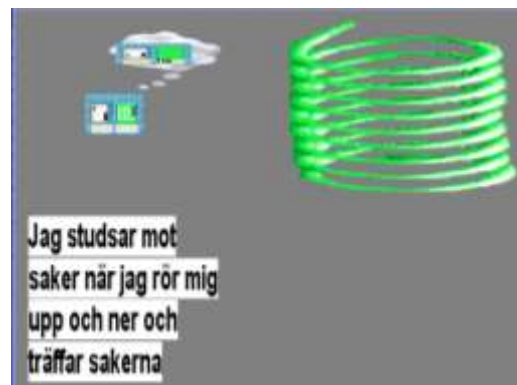


Figure 3b the vertical bounce behaviour
(Swedish version)



Figure 4a A message is given to a bird ...



Figure 4b...she delivers it to her nest (labelled 'test') and returns to her original location. The label 'test' has no semantic function.

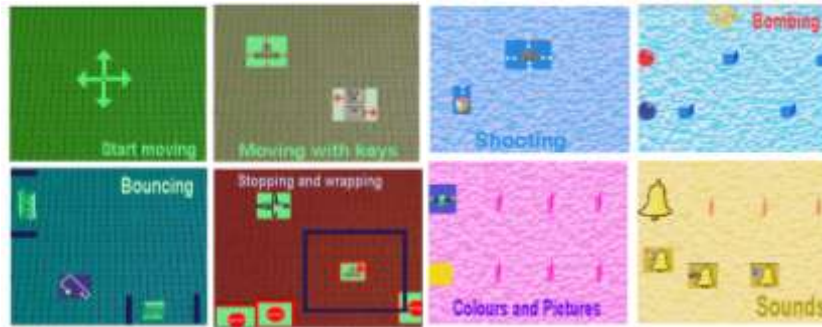


Figure 5 The eight anima-gadgets the children had access to in the game building activity.

Start moving (eg. I start moving left); Moving with keys (eg. I move in the directions of the arrow key you press); Shooting (eg. I fire white balls when you press the mouse button); Bombing (eg. I bomb things when they touch me); Sounds (eg. I make a sound when something touches me); Colours and pictures (eg. When I hit something, my picture changes); Bouncing (eg. When I am moving, I bounce off top and bottom of objects); Stopping and wrapping (eg. I stop at the edge of the scene)



Figure 6a Black background of the game with the white coffin that bounces between two barriers

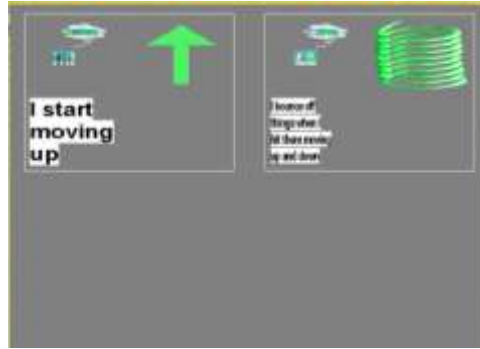


Figure 6b Flipside of the coffin with the bounce behaviour which is comprised of: I start moving up and I bounce up and down



Figure 7 The game with the bouncing coffin and two barriers, the player character, and the ring.



Figure 8a Details of the game showing the coffin changed into a girl after it has touched something

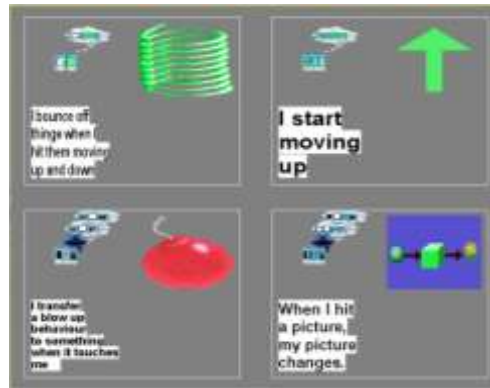


Figure 8b The flipside of the coffin with four behaviours. The behaviours are: I bounce off of things when I am moving up and down; I start moving up; I bomb a player when it touches me; When I hit something, I change my picture

Coffin changes into aliens when touched by something. It will also bomb the player

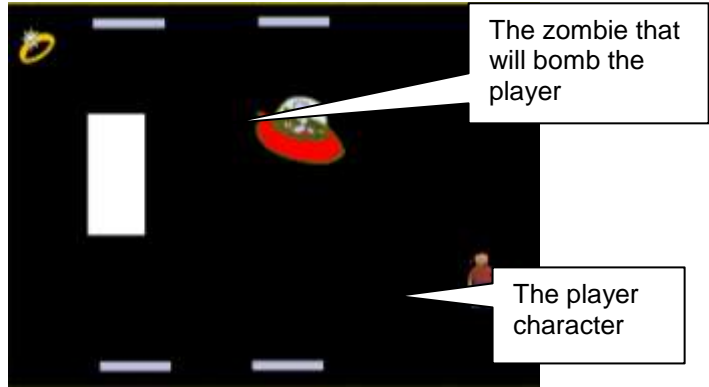


Figure 9 Safia and Faroq's finish game.

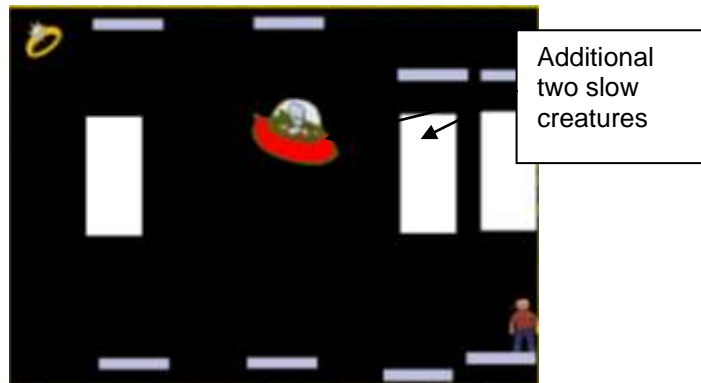


Figure 10 The game after the changes by the Swedish boys. Two slower coffins that blew up the player (and changed their pictures) were added.

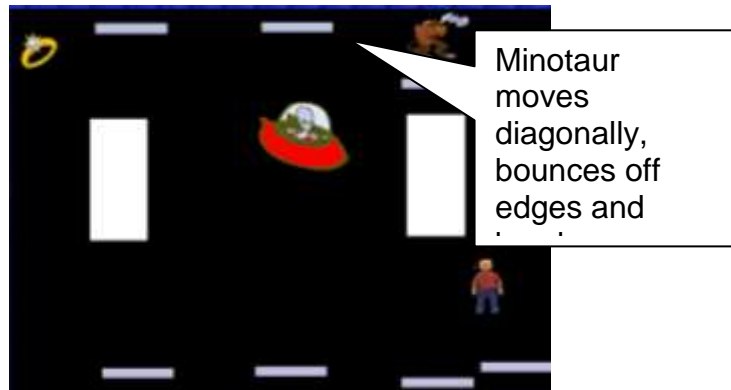


Figure 11 The minotaur has been taken from the Swedes' game and added to the English children's game.