Embodied interaction in authoring environments

Pablo Romero, Judith Good,
Benedict du Boulay, Henry Reid, Katherine Howland
IDEAs Laboratory
University of Sussex, Department of Informatics
Brighton BN1 9QH, UK
+44 (0)1273 877 543
pablor@sussex.ac.uk

Judy Robertson
Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh EH14 4AS, UK
+44 (0)131 451 8223
Judy.Robertson@hw.ac.uk

ABSTRACT
How can we make it more appealing for young people to learn about and understand digital technology and computing concepts? We use both a different kind of interaction and a different kind of content to convey computing ideas such as *subroutines, modules, abstraction, classes, objects* and *debugging*. A crucial notion is embodied interaction, the idea that the child's interaction with the system should, in part, use her whole body, and not just fingers and eyes. In order to explore this approach we use the magic mirror, a device constructed from an interactive whiteboard and a camera such that when the child stands in front of it, the image she sees is her own but changed or augmented. For example it may show her in costume, or as a non-human creature. As she moves, so does her image. This device can be used as part of a lesson in which children create sequences of movements that can be recorded by the system and then manipulated in various ways to create more complex entities. The proposed work cuts across several disciplines, including education, psychology and computing, and will consider: in what ways can embodied authoring in principle (and in practice) be used to explore computing concepts, and, for each way, what advantages and disadvantages might it offer over traditional methods? In this paper we elaborate on the ways in which embodied authoring can support programming, explain the detail of the approach and report on some preliminary prototyping work we have undertaken.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *graphical user interfaces, input devices and strategies, user-centered design.*

General Terms
Design, Human Factors, Languages.

Keywords
Embodied interaction, Programming, Software Authoring.

1. INTRODUCTION
Digital technology plays an increasingly important role in our lives. Given the importance of software and digital products, there is a need for people who can develop these products, both at professional and end-user levels: not only students undertaking Computer Science studies and careers but also other professionals able to use packaged software as well as produce their own computational solutions [4]. However admission and retention rates to computer science university courses are falling [20], enrollment is male dominated [20] and although there is a thriving community of end-user programmers, there are serious concerns about the dependability of the software which they produce [4]. Thus there is a need both to foster the development of computational thinking [23] in young learners and to motivate them to study computing subjects by improving the perception of computing, especially for girls.

We suggest that there are three issues that conspire to make computing concepts difficult to learn i) context: the problems and scenarios to which the concepts are applied are often not very motivating, ii) abstraction: some of the concepts are presented in too abstract a fashion and iii) great attention to detail is needed in order to make something appealing work.

A number of researchers have grappled with the issues above. A prominent strand of this work has sought to make the learning of computing a playful and creative endeavour: for example, Lego Mindstorms (http://mindstorms.lego.com/), and the Scratch programming language (http://scratch.mit.edu/), whose website indicates how young people are appropriating the language to create programs of real interest and meaning to them. With respect to gender issues, there is a growing body of work which acknowledges the role of women in the computing field, e.g. the Grace Hopper Celebration of Women in Computing (http://gracehopper.org/2007/). And finally, there are concerted moves toward extending the notion of computing, and its overall reach: VL/HCC 2007's doctoral consortium, entitled, "Broadening the Audience for Computational Thinking, is a prime example (http://vlhcc07.eecs.wsu.edu/consortium.html).

2. EMBODIED AUTHORING
If the underlying computational concepts were to be introduced in a dramatically different way, the story would be different. A key to this change is modifying the interaction medium. We will employ embodied interaction [6] to make computing concepts more accessible, and increase the appeal and collaborative potential of the scenarios within which these concepts are introduced. Embodied interaction, i.e. using the physical world as a medium for interacting with digital technology, is a new form of naturalistic, multi-modal interface that can support co-located communities of learners through a variety of different interaction devices.

Proceedings of the second Workshop on Physicality
Physicality 2007, 2-3 September 2007, Lancaster University, UK
Devina Ramduny-Ellis, Joanna Hare, Steve Gill & Alan Dix (Editors)

Our approach to introducing computing concepts combines embodied interaction with social constructivist and constructionist theories of learning [17, 21, 5, 18]. Through acting and recording animation scenes instead of programming complex character animation episodes, young people will be involved in an authoring process, where, by turns, they will be both designers and critical consumers of their own and other children's applications. This process will be embedded in scenarios in which there is scope for discussion and collaborative, imaginative play.

This research programme is investigating the ways in which embodied interaction can be used to explore computing concepts and, for each way, the advantages and disadvantages it might offer over traditional methods.

There are at least four ways in which embodiment, or more generally any interaction modality, can be used in programming[1]:

- By supporting non-coding activities. A non-coding activity is a task necessary for the execution of the program but that does not involve programming or scripting. Capturing data or creating a graphical element that the program will use in its execution are examples of such auxiliary activities. The auxiliary activity can be performed in a different interaction modality to the rest of the programming activity. For example, a textual programming language might use graphic elements created in a graphic modality or data inputted as an audio recording. Technological innovations have allowed some programming activities to become auxiliary, for example GUI front ends can now be produced in a graphical environment (by drawing them) instead of programming them.
- As the interface medium. The interface medium is the way in which the programmer interacts with the program. Usually the interface medium is keyboard and mouse but programmers could also interact with the program via speech recognition for example.
- As the programming environment. The programming environment provides a set of tools to support the programming activity. Some examples of programming environment tools are coding editors, output windows, visualisations of the program, automated testing facilities, etc. Programming environments nowadays are usually graphical, even for textual programming languages. Logo and its tangible turtle [17] are an example of a textual programming language combined with a programming environment employing tangible elements.
- As the programming language itself. In this case the lexical elements of the language are expressed in a particular modality. Traditionally programming languages are textual but there are also visual and even tangible programming languages. One could think of a tangible programming language within a graphical programming environment, or of a textual language within a tangible environment, although this might not make much sense in practice.

So far we have identified one way in which embodiment can be used to explore computing concepts, the acting and recording of animation scenes. This can be considered as a non-coding activity which can replace a programming task (the coding of animation scenes). We believe there are several benefits of employing embodied interaction in this way for educational applications: it can make sophisticated authoring tasks more accessible to a wider audience, it can establish a link between computing and other parts of the curriculum such as drama, English and the performing arts, it could help to improve interest in computing degrees and careers and could also enable a more active interaction in the physical sense.

3. ACTING AND RECORDING ANIMATION SCENES

The main idea behind the approach is the possibility of acting and recording movements. This is done through a combination of small paper markers (similar to barcode labels) placed on different parts of a person's clothing, a webcam and a large screen. This arrangement works as a magic mirror where the user is able to see a reflection, however this reflection can be in the form of a character of her choosing (see Figure 1).

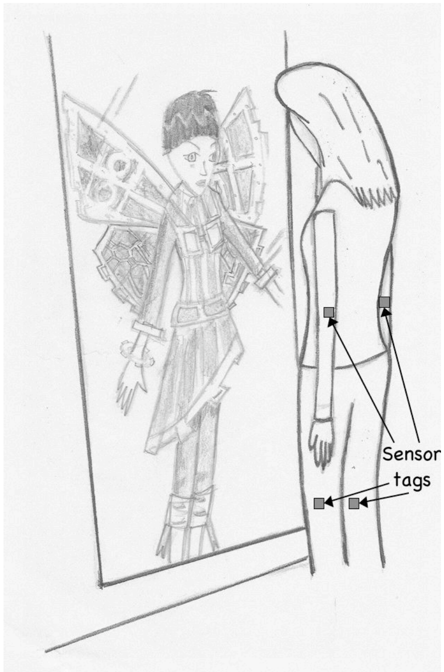

Figure 1. The Magic Mirror.

Additionally, users will be able to record their movements as if they were recording a film with a video camera. However there is an important difference: when users employ a video camera they record concrete scenes, when they use the magic mirror they record movements. Movements are more abstract than concrete scenes because they do not need to be associated with a particular character, background, place in a scene or size, among other characteristics. In this sense, movements can be considered as animation libraries that can be used for authoring purposes. Movements can be instantiated to different characters, duplicated, speeded up, played backwards, connected to create composite movements, etc. The authored applications can be populated with several instances of movements and each one of these instances can be associated with specific behaviours when users interact with them. Young people will be able to manipulate them to build their own applications and in so doing, they will have to familiarise themselves with important computational thinking skills such as *abstraction, modularisation*, identifying and working with *abstract*

[1] In the discussion that follows, we will consider the concept of interaction modality in its widest sense, so we can talk of a textual modality, a graphical modality or a tangible modality.

entities (classes) and their instantiations (*objects*, etc.).

The manipulation of movements will be performed with an editor that follows conventions similar to those of commercial authoring environments such as Flash [7] and will employ an embodied-style interface medium such as the one used for the Eye-Toy [11] or the Nintendo Wii [19]. This form of interface will be compatible with the magic mirror and will enable a concurrent interaction mainly via body movements or a remote control. Concurrency and the public aspect of large screen-based applications, which are shared characteristics with the magic mirror, give this approach strong collaborative potential [8, 1].

The proposed approach will therefore comprise two parts, a platform for developing applications (the Stage Platform), and the actual applications that will run on the platform. The platform will include the magic mirror to record movements and an editor to manipulate them. The applications developed for the platform could be environments to create simulations, video games, authoring environments for creating free-form playful applications or game authoring environments, among others. We are particularly interested in the latter two environments as they allow plenty of scope for imaginative play within the authoring (construction) process.

The Stage Platform will be similar to other generic platforms and tools like the DART Toolkit [15], the standard toolkit for programming augmented reality applications, in that it will be open (and we will actively encourage) for other research groups to use it. We believe that there is a need for a platform to enable developers to build applications with an embodied style of interaction and the Stage Platform can respond to that need similarly to the way the DART Toolkit provided developers with a tool to build Augmented Reality applications.

As part of our research programme, we intend to build a pair of authoring environments related to subjects such as drama, English or dance, for example. We believe the latter is important as performance art subjects usually involve role playing, social interaction and narrative, and these activities have been found to be particularly appealing to girls [3, 10].

Both the platform and the authoring environments will be developed using a learner-centred design process [14, 9] that will involve users, stakeholders and potential beneficiaries.

The authoring environments to be developed will be fairly specific so that, from the outset, young people focus on the environment's application area, making authoring activities implicit aspects of the task rather than the main, explicit part of it. This approach has proven to be effective for recent authoring and scripting environments [13, 2] and has also the added value that a good deal of basic functionality and libraries specific to the environment's application area can be built in to the authoring environment, providing enough support for young people to create more powerful applications but at the same time leaving them enough room for a sense of challenge and the chance to be creative.

3.1 A Sample Scenario

The sample scenario that follows is about young people creating their own dance mat-like games [12] collaboratively and through embodied interaction with Dance Along, a game authoring environment that could be built for the platform.

Dance Along: The students from year 7 are looking at the Maori people in history and the dance teacher has linked the class activities to this topic by practicing the Haka dance. A small group of students in this class (Caitlin, Camila and Jon), who are looking at design abstraction and modularisation in ICT, are interested in creating a dance mat-like game to help the rest of the children learn the Haka. They use the Dance Along game authoring environment to recreate one of these dances and then dance along to it. First, they act and record a basic sequence of dance moves using the magic mirror. Then they edit these dance moves (figuring out the global design of the choreography, creating and assigning characters to dance moves, connecting and duplicating the dance moves), interacting with the moves editor concurrently, each using a wii remote control to create a complex dance with 40 characters. Finally they invoke a built-in functionality for giving feedback and producing a score for players (this works by comparing the trajectories of players dancing along with those of the programmed characters). Then they invite Chloe, who is also in year 7, to see the movie of the dance and then dance along to it. Chloe chooses the character she wants to be and then dances in front of the magic mirror as if she were that character. When the dance finishes she looks at her score and watches the movie of her dancing as the character and realises that she didn't do that badly in the Haka. Chloe is so impressed by the game that she asks Caitlin, Camila and Jon to teach her how to use the Dance Along environment to create her own dance games. They show her how to record scenes and how to design a choreography by manipulating dance moves, stressing that some of the key concepts are designing the dance in a modular way, understanding the difference between the abstract library of moves and their concrete instantiations and assigning parameters to moves.

The scenario illustrates the main characteristics of our approach, an embodied style of interaction [6], fostering computational thinking skills by an active process of construction (not only of a piece of software but also of a plan of action and strategies) [17, 18] and the mediating aspect of the environment to provoke collaborative encounters [5].

The scenario also exemplifies the sorts of authoring environment that can be developed for the platform, the types of application that young people can create with the authoring environments, and the ways these applications can be played. In this case it illustrates a game-authoring environment, but the platform can also be used to implement authoring environments that can be used to build more free-form playful applications. This versatility will be helpful in evaluating the approach in a variety of contexts. Also, in this case, both the authoring process and the game play are performed with an embodied style of interaction. This is the ideal situation, but it could also be the case that the constructed software is played as a conventional desktop application.

3.2 A Proof-of-Concept Prototype

The STAGE system includes an input device, the "magic wand", and an interface in the form of a "magic mirror". After testing and evaluating the many options for the elements of the STAGE system, the best current candidates were chosen and implemented.

The Magic Wand uses the Nintendo Wii Remote and Belkin Bluetooth 2.0/EDR USB Adapter, with WiinRemote, GlovePIE and Widcomm Bluetooth as software interfaces. The Magic Mirror uses the Trust Megapixel USB2 Webcam with ARTag as the C++/OpenGL marker tag detection library. The user interface is written in Java 6.0 using the standard Swing widgets.

Current functionality includes changing Magic Mirror characters and backgrounds, and limited possibilities for Story Grid manipulation. The recording of scenes does not provide significant challenges but presenting fluid and anatomically plausible movements may require motion capture techniques [16].

As part of our ongoing learner-centred design process, we have conducted workshops with school teachers and have obtained very positive feedback on the potential of the system.

4. CONCLUSION

This paper sketches a research programme for investigating the ways in which embodied interaction can be used to teach and learn computing concepts and for each way what advantages and disadvantages might it offer over traditional methods. It presents a tentative analysis of the ways in which embodied interaction can be used in authoring and programming, explores the potential of a specific approach, the acting and recording of movements, and briefly describes a prototype built as a proof of concept.

Future work will consider a full implementation of the platform as well as educational applications similar to the one described in the scenario of Section 3.1. Additionally, the research will explore some of the other ways (described in Section 2) in which embodied interaction can be used in programming.

5. ACKNOWLEDGEMENTS

This work was supported by a UK ESRC/TLRP Technology Enhanced Learning development grant (RES-139-25-0338) (.

6. REFERENCES

[1] G. Beauchamp and J. Parkinson. Beyond the ‘wow’ factor: developing interactivity with the interactive whiteboard. *School Science Review*, 86(316):97–103, 2005.

[2] A. Begel and E. Klopfer. Starlogo TNG: An introduction to game development. *Journal of E-Learning*, 2007.

[3] C. Brunner, D. Bennett, and M. Honey. Girl games and technological desire. In J. Cassell and H. Jenkins, editors, *From Barbie to Mortal Kombat: Gender and computer games*. MIT Press, Cambridge, 1998.

[4] M. Burnett, C. Cook, and G. Rothermel. End-user software engineering. *Communications of the ACM*, 47(9):53–58, 2004.

[5] C. Crook. *Computers and the collaborative experience of learning*. Routledge, London, 1994.

[6] P. Dourish. *Where the action is: the foundations of embodied interaction*. MIT Press, London, England, 2001.

[7] J. English. *Macromedia Flash 8:Training from the Source*. Macromedia Press, Berkely, CA, 2005.

[8] S. Godwin and R. Sutherland. Whole-class technology for learning mathematics: the case of functions and graphs. *Education, Communication & Information*, 4(1):131–152, 2004.

[9] J. Good and J. Robertson. CARSS: A framework for learner centred design with children. *International Journal of Artificial Intelligence in Education*, 16(4): 381-413, 2006.

[10] C. M. Gorriz and C. Medina. Engaging girls with computers through software games. *Communications of the ACM*, 43(1):42–49, 2000.

[11] J. Hill. The rise and rise of playstation 2. http://www.smh.com.au/articles/2004/10/27/1098667824629.html?from=storylhs, 2004 (accessed 31 May 2007).

[12] J. Hoysniemi. International survey on the dance dance revolution game. *Computers in Entertainment*, 4(2):8, 2006.

[13] C. Kelleher and R. Pausch. Lessons learned from designing a programming system to support middle school girls creating animated stories. In *VLHCC ’06: Proceedings of the Visual Languages and Human-Centric Computing*, pages 165–172, Washington, DC, USA, 2006. IEEE Computer Society.

[14] R. Luckin, J. Underwood, B. du Boulay, J. Holmberg, L. Kerawalla, J. O’Connor, H. Smith, and H. Tunley. Designing educational systems fit for use: A case study in the application of human centred design for AIED. *International Journal of Artificial Intelligence in Education*, 16:353–380, 2006.

[15] B. MacIntyre, M. Gandy, S. Dow, and J. D. Bolter. DART: A toolkit for rapid design exploration of augmented reality experiences. In S. K. Feiner and J. A. Landay, editors, *UIST ’04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 197–206, New York, NY, USA, 2004. ACM Press.

[16] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104:90–126, 2006.

[17] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York, 1980.

[18] S. Papert. *The Connected Family: Bridging the Digital Generation Gap*. Longstreet Press, Atlanta, Georgia, 1996.

[19] A. Sliwinski. Nintendo wii marketing to exceed 200 million. http://www.joystiq.com/2006/11/12/nintendo-wii-marketing-to-exceed-200-million/, 2006 (accessed 31 May 2007).

[20] J. Vegso. Interest in CS as a major drops among incoming freshmen. *Computing Research News*, 17(1):17–18, 2005.

[21] L. Vygotsky. *Thought and Language*. MIT Press, Cambridge Ma, 1986.

[22] D. Wagner. The artoolkitplus 2.1.1. http://studierstube.icg.tu-graz.ac.at/handheld ar/artoolkitplus.php, 2007 (accessed 26 June 2007).

[23] J. M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.