

Tutorial: Game Design for Education

Introduction: Why Game Design?

There is increasing agreement that computational literacy [2] for children encompasses the need to move beyond just being consumers to being producers of computational artifacts. Educational activities of this nature are not limited to programming. For instance, the fluency with information technology (IT) framework by the National Academies of Science¹ includes notions such as the ability to collaborate, model, create abstractions, and create image-based expressions of ideas. Unfortunately, most IT programs in K-12 education seem to fall into one of two extremes. On the one hand, we find low-end courses based on multimedia tools such as PowerPoint in which student create animated presentations. On the other hand, we find IT courses that are miniature versions of undergraduate hard-core computer science programs. With a strong focus on algorithmic thinking and traditional programming, these courses often do not include “fun” applications and ultimately only attract a small number of students.

Game design provides a necessary middle ground between these two extremes. The design and implementation of a game is a highly motivational and collaborative activity that requires a high degree of computational literacy. However, game creation is complex. A lot of research on end-user programming has explored the intricacies of programming for non-professional programmers. The choice of an authoring tool is, of course, important, but we find it even more important to scaffold the design process. Many of the design methodologies that emerged from software engineering are not well suited for applications in K-12 education. To address this, we have developed a new design methodology aimed at the design process of simple games in education called *Gamelet Design*.

Objectives

- learn how to create simple Web-based games called Gamelets;
- learn how to analyze existing arcade games;
- build and publish two Gamelets by applying the Gamelet Design methodology; and
- learn how to use the AgentSheets authoring tool for Gamelet implementation.

Justification for tutorial inclusion in the IDC2005 program

The Gamelet design methodology is a means of facilitating computational literacy. In the sense that we are sharing frameworks that help students and teachers to collaboratively build computational artifacts, we are providing methods and techniques for designing *with* children.

Over many years, we have worked on authoring tools for kids to make simulations, games and program LEGO robots, such as AgentSheets [8-11] and LEGOsheets. To create the tools, we have designed authoring mechanisms by applying user-centered and participatory design approaches. We have tested these authoring tools in different contexts ranging from elementary schools to universities. Some of the projects we have worked on include:

- ***EcoWorlds***: elementary school students used AgentSheets and a domain-oriented version of its programming language to explore life science concepts such as characteristics of

¹ <http://www.nap.edu/openbook/030906399X/html/15.html#pagetop>

organisms, structure and function in living systems, populations and ecosystems. Students worked in small groups to create computer simulations of ecosystems in different environments such as the arctic or a desert [1, 4, 6, 7].

- ***Social Studies projects:*** High school students used AgentSheets to create simulations and embed them in Web pages with related content as part of a history classes [1, 4, 5, 12]. For instance, students studying the 20th century social history created simulations about the impact of the automobile on society and the environment.
- ***LEGOsheets:*** LEGOsheets [3] is a programming, simulation and manipulation environment created using AgentSheets for controlling the MIT Programmable Brick (a research prototype of what is now known as LEGO Mindstorms). Students in middle schools helped us design the authoring environment and used it to create interesting LEGO artifacts such as voice controlled vehicles and program cars for racing competitions.

As useful, motivating, and educationally effective creating computational artifacts is, our experience in these school contexts has been that without the necessary support, programming can quickly turn excitement to frustration. We have developed the Gamelet design methodology to scaffold the process of creating games. We will be presenting this methodology in this tutorial and provide hands-on practice in applying it using the AgentSheets authoring tool.

Material to be covered & participation requirements

Participants in this tutorial will learn about:

- the Gamelet design process
- Game analysis
- the AgentSheets authoring tool

The AgentSheets authoring tool will be provided to tutorial participants. A programming or a game design background is not required for participation. Familiarity with classic arcade games such as Pacman, Space Invaders, Frogger, Centripede etc. is helpful, but not required. However, enjoyment in playing or creating games is required!

Schedule of events

This is a 6-hour tutorial. Assuming the tutorial starts at 9am and has a lunch break of one hour, this is the schedule of events:

9-9:30	General Introduction
9:30-10:00	Introduction to the AgentSheets authoring tool: Example projects Hands on: play with one game
10:00-12:00	Tutorial #1 The makings of a simple arcade game: Frogger Design: Project analysis: agents, methods and states/depictions Implementation: Step-by-step following the Gamelet design process
12:00-1:00	Lunch: form teams, pick a project (or choice of classic arcade games such as Space Invaders and Sokoban) and discuss design
1:00-2:30	Tutorial #2: work in teams to build selected project
2:30-3:00	Show and Tell
3:00 -4:00	Discussion

Presenters

Dr. Alexander Repenning

Dr. Alexander Repenning is the CTO of AgentSheets Inc. and a professor of Computer Science at the University of Colorado. Repenning's research interests include education, end-user programmable agents, and artificial intelligence. He has worked in research and development at Asea Brown Boveri, Xerox PARC, Apple Computer, and Hewlett Packard. Repenning is the creator of the AgentSheets simulation and game-authoring tool. His work has received numerous awards including the Gold Medal from the mayor of Paris for "most innovative application in education of the World Wide Web" and "best of the best innovators" by ACM.

Dr. Andri Ioannidou

Dr. Andri Ioannidou is the Senior Project Manager at AgentSheets Inc. Her research interests and expertise include end-user development and end-user programming, agent-based simulations, educational technology, and human-computer interaction. She engages in outreach activities in local elementary, middle and high schools, where she works with teachers and students to introduce simulation-based activities in math, science, and social studies courses, as well as extra-curricular activities.

Presenters' requirements such as technology support or attendance limits

- Video projector
- Optional: network
- Computers: participants will work in teams of size 2. Participants could use their own hardware
- White board or large stickies

References

1. Cherry, G., Ioannidou, A., Rader, C., Brand, C. and Repenning, A., Simulations for Lifelong Learning. In Proceedings of National Educational Computing Conference (NECC) (Atlantic City, NJ, 1999).
2. diSessa, A. Changing Minds: Computers, Learning, and Literacy. The MIT Press, Cambridge, MA, 2000.
3. Gindling, J., Ioannidou, A., Loh, J., Lokkebo, O. and Repenning, A., LEGOsheets: A Rule-Based Programming, Simulation and Manipulation Environment for the LEGO Programmable Brick. In Proceeding of Visual Languages (Darmstadt, Germany, 1995), IEEE Computer Society Press, 172-179.
4. Ioannidou, A., Rader, C., Repenning, A., Lewis, C. and Cherry, G. Making Constructionism Work in the Classroom. *International Journal of Computers for Mathematical Learning*, 8, 63-108, 2003.
5. Ioannidou, A., Repenning, A. and Zola, J., Posterboards or Java Applets? In International Conference of the Learning Sciences 1998 (Atlanta, GA, 1998), Association of the Advancement of Computing in Education, 152-159.
6. Rader, C., Cherry, G., Brand, C., Repenning, A. and Lewis, C., Principles to Scaffold Mixed Textual and Iconic End-User Programming Languages. In Proceedings of the 1998 IEEE Symposium of Visual Languages (Nova Scotia, Canada, 1998), Computer Society, 187-194.

7. Rader, C.A. Using students' expressive models to facilitate conceptual development. Ph.D. Thesis, Department of Computer Science, University of Colorado, Boulder, 2000.
8. Repenning, A. and Ambach, J., Tactile Programming: A Unified Manipulation Paradigm Supporting Program Comprehension, Composition and Sharing. In Proceedings of the 1996 IEEE Symposium of Visual Languages (Boulder, CO, 1996), Computer Society, 102-109.
9. Repenning, A. and Ioannidou, A., Behavior Processors: Layers between End-Users and Java Virtual Machines. In Proceedings of the 1997 IEEE Symposium of Visual Languages (Capri, Italy, 1997), Computer Society, 402-409.
10. Repenning, A., Ioannidou, A. and Ambach, J. Learn to Communicate and Communicate to Learn. Journal of Interactive Media in Education, <http://www-jime.open.ac.uk/98/7>, 98 (7), <http://www-jime.open.ac.uk/98/7>, 1998.
11. Repenning, A. and Sumner, T. Agentsheets: A Medium for Creating Domain-Oriented Visual Languages. IEEE Computer, 28 (3). 17-25, 1995.
12. Zola, J. and Ioannidou, A. Learning and Teaching with Interactive Simulations. Social Education: the Official Journal of National Council for the Social Studies, 64 (3). 142-145, 2000.