Teaching Computer Game Design to Middle School Students

Jill Denner & Jacob Martinez, ETR Associates
Linda Werner, University of California, Santa Cruz

BACKGROUND

- There are numerous programming environments that can be used to teach children to build computer games.
- Each environment has pros and cons, but few comparative analyses exist.
- The goal of this poster is to provide educators with information to consider when choosing a programming environment.
- Since 2003, we have taught over 300 middle school students to program computer games.
- In this poster, we describe our experience using Macromedia Flash MX, Storytelling Alice, and Stagecast Creator.

Flash MX

How We Taught it:
Length and frequency of instruction: 60 hours (2 hours/day for 23 weeks)
Total number of students: 126
Age range of students: 11-14
Gender of students: 100% female
Types of games created: Choose your own adventure

Pros
- Appearance: Full color animation is very appealing.
- 1F workforce skills: Learning to use this software builds real world job skills.
- Allows for creativity: Students can draw/animate their own images or download them from the Internet.
- Students can learn new programming concepts: Variables, loops, goto.
- Instant reinforcement and debugging: Can test program and instantly see results of each step.
- Playability: Games can be played by anyone with a Flash player, which is widely available.

Cons
- Very high threshold for entry: Actionscripting (which involves programming) is complex and takes students with limited experience at least 30 hours to build a simple yet interactive game.
- Variety of games: High levels of teacher support are needed, which limits the types of games that can be created by less experienced students.
- Instructional materials: Geared toward adults rather than children.
- Cost: The software is expensive (over $1,000).

Storytelling Alice

How We Taught it:
Length and frequency of instruction: 20 hours (2 hours/day for 2 weeks)
Total number of students: 33
Age range of students: 10-14
Gender of students: 42% female
Types of games created: Racing, Trivia, Fighting, Adventure, Puzzle

Pros
- Very low threshold for entry: Students with limited computer experience can make a story in 4 hours.
- Frustration is reduced by prevention of syntax errors: This is done via use of programming via drag and drop.
- Appearance: 3-D color images and animation are appealing and modern.
- Variety of games: Students can choose among a wide range of game types.
- Students can learn key programming concepts (e.g., loops, parallelism, methods, variables, parameters, conditional execution) that are directly transferable to other programming languages.
- Instant reinforcement and debugging: Can test program and instantly see results of each step.
- Cost: The software is free.

Cons
- High threshold for game production: Students with limited programming experience may need to spend 10 hours doing instructional "challenges" before they can make their own game.
- Instructional materials: Tutorials are useful but do not teach about interactivity. There are no curricular materials available for Storytelling Alice; Generic Alice textbooks are not appropriate for this age-group.
- Limited creativity: Gallery of characters is limited and creating new ones is not easy for children.
- Frustration: Working in 3-D can be confusing and makes it easy to lose objects or perspectives.
- Availability: Not available for Mac. (Generic Alice is available for both PC and Mac platforms).
- Playability by others: People must download the software in order to play the games.

Stagecast Creator

How We Taught it:
Length and frequency of instruction: 16 hours (2 hours/week for 8 weeks)
Total number of students: 150
Age range of students: 10-14
Gender of students: 100% female
Types of games created: Adventure, Action, Maze, Trivia

Pros
- Very low threshold for entry: Students with limited computer experience can make a game in a few hours after completing the tutorials.
- Variety of games: Students can choose among a wide range of game types.
- Frustration is reduced by prevention of syntax errors: Students learn via "programming by example" to create "visual roles" which control program characters' behavior.
- Instant reinforcement and debugging: Can test program and instantly see results.
- Students can learn key programming concepts: Rules, if-then-else.
- Allows for creativity: Students can draw 2-D characters and import background images from the internet resulting in full color images.
- Pre-existing images allow students to start programming quickly: A library of characters, backgrounds, and simulations with pre-existing rules are available.
- Playable by others: The games can be played without downloading software.

Cons
- Appearance: The 2-D interface looks more simplistic than current digital games.
- Instructional materials: Tutorials provide a good introduction and are necessary, but additional guides are needed to help students make their own games.
- Does not allow students to explicitly learn certain programming concepts which would be transferable to other programming languages, such as iteration.
- Does not promote clarity of thinking: Students can quickly create many "visual roles" and are not required to clean them up for the program to work. The large amount of unused code makes it difficult for anyone (including the student) to understand.
- Cost: Must purchase Stagecast Creator software ($775 for 25 user lab package).