Arena: An Game Toolkit for the Capella's Song Project

Abigail Corfman, Tiffany Barnes
{abigail.corfman@gmail.com, tbarnes2@uncc.edu}

Abigail Corfman
Oberlin College

Introduction

The Game2Learn Lab at UNC Charlotte is developing a video game called Cappella's Song for teaching introductory computer science classes. The game features an in-game compiler so that students can write real code in the game.

Capella's Song is a role-playing game where students can select quests to learn computer science skills. My primary responsibility was to build a toolkit called Arena to rapidly develop new learning quests. Arena currently includes a simple MapMaker and a game called Blackwater Mines, which serves as a blueprint game that can be reused to build new Capella Song games that incorporate instant, visual feedback, an interactive tutorial, and a realistic programming environment.

Background

Arena was inspired by several existing games used for teaching introductory programming. Games are primarily used to motivate learning, and also to provide new, interactive metaphors for abstract concepts. Game2Learn has learned several lessons that I apply in Arena, including: 1) in-game feedback should be instant and related to learning goals, 2) instructions must be part of the game, and 3) interactive visualization of code is important for learning. RoboWars is a game for teaching Java, where the players program cars to fight each other. It exemplifies how a game can help players visualize what code is doing.

Moose Crossing is an online, text based game developed at Georgia Tech, where children create and program objects that can interact in the game.

Research

- Weeks 1-4

I learned C# and the Game Engine XNA. I designed and implemented a Map Maker program for quickly producing tile based maps for Arena games. I modeled it after RPG Maker.

- Weeks 5

I designed *Blackwater Mines,* a sample game to test the Arena framework.

- Weeks 6-8

I implemented game play *Blackwater Mines*. Implementing animation was particularly challenging.

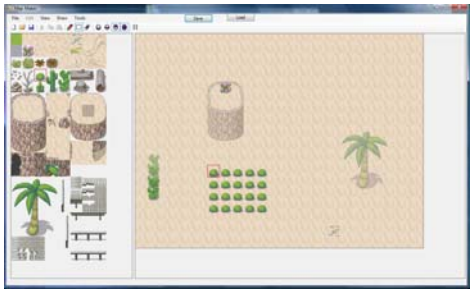
- Weeks 9-10

I implemented the GUI framework and the tutorial for *Blackwater Mines*.

Results

The Map Maker Program

- A universally compatible and easy to use utility for making game maps.
- The first step to making a generalized Game Maker that can handle all aspects of game play.



Map Maker screenshot showing placement of tiles in the map on the right.

Results Cont.

The Blackwater Mines

Blackwater Mines is a sample mission used for teaching the basics of methods. The premise is that the Blackwater mines have been flooded with toxic gas and the human miners can't work anymore. The player is hired to program a robot, called a Machina in the game, to go down into the mines and collect gemstones.

Blackwater Mines presents coding as a puzzle. It teaches the player a variety of skills and in later levels, the player figures out how to accomplish objectives by applying and combining things she has learned.

Blackwater Mines screenshot, showing the mine window on the left, and the in-game Mentor Alastor in the black areas on the right.

Impact

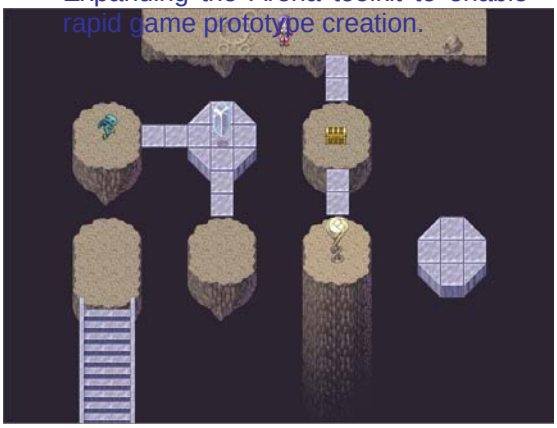
- Arena provides a framework with a comprehensive GUI and an in-game compiler, that allows quick creation of new tile-based maps and new games.
- *Blackwater Mines* is a working educational game for teaching methods, a difficult concept in CS1.
- *Blackwater Mines* also provides a game blueprint that facilitates building new games with the same structure, using a mentored gameplay format.

Conclusions

- *Blackwater Mines* and other Arena games bridge the gap between elementary or modular games and a full IDE, making it easy for a player to apply what he learned.
- Arena can be used to create minigames that could replace or supplement lab work.
- The games can be used independently to teach particular concepts.
- The games can be played outside of a classroom and facilitate independent learning.
- The modular design of the game and eventual toolset will let anyone build more levels and expand the game play.

Future Work

- Testing the game on students and measuring how much it truly facilitates learning, making sure it works as indented and addressing things they have trouble with.
- Using the information from testing to build new games that teach different aspects of computer science.
- Expanding the Arena toolkit to enable rapid game prototype creation.



Possible Future Level that requires students to program a robot to navigate a complex environment.