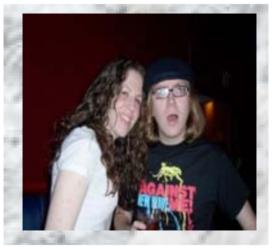
EleMental: The Recurrence

Andrew Hicks, Katelyn Doran, Graduate Advisor: Amanda Chaffin, Mentor: Dr. Tiffany Barnes
{agh001@marietta.edu, kedoran@uncc.edu, katla@wulfcub.com, tbarnes2@uncc.edu}



Katelyn Doran
UNC Charlotte

Andrew Hicks
Marietta College

Introduction

Educational games have gained popularity as tools to enhance the existing curriculum in the classroom. For the field of computer science, however, many educational games do not provide the hands-on experience that students need from their assignments. Because of this, educational games remain as forms of extra practice rather than alternative assignments.

Our hope with this project was to create a game which could be used as a basis for assignments. The game world would provide a context for the traditional programming assignment.

Background

The existing EleMental game engine was created in Spring 2007 as part of the Advanced Game Design course at UNC Charlotte. This game engine features an in-game compiler system which allows students to enter, compile, and run their own code. The opportunity for students to write and use their own code within a game setting goes beyond current games, which often feature drop down menus or drag and drop selection of code.

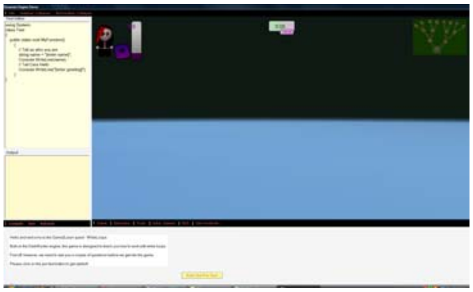
We feel that use of in-game compiling will enhance student understanding of course materials. All too often, students do not see the relevance of the assignments they are given for their computer science classes.

We hope to resolve this issue by providing real coding experience within the context of the game. Currently, our game, EleMental: The Recurrence is intended to teach recursion and depth-first search. However, there is potential to teach any coding concept.

Research

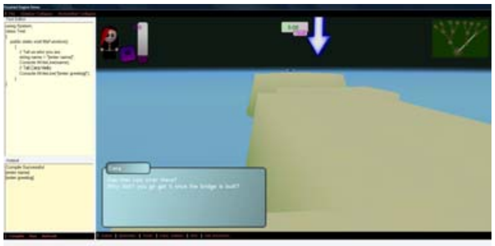
Our project began with us familiarizing with C# and the XNA framework. Once familiar with the syntax of the language, we moved on to learning the intricacies of the EleMental engine. As we grew more familiar with what the game engine was capable of, we began to form an idea of what kind of game we would like, and be able, to create.

The engine provided us with several unique tools which we wished to incorporate into our game. The first, of course, was the compiler. Naturally, the best use of this feature is to take in the student code. The second feature of the engine we wanted to utilize in the game was that of terrain modification. At first, we were not entirely sure of how to best incorporate the terrain tool into our game. We had already been asked by our mentor to focus our game on recursion, as it is one of the more difficult concepts in computer science.



EleMental: The Recursion view with compiler

In order to best incorporate the features we wanted to use, while still keeping our focus on recursion, we chose to base our game around depth-first search of a binary tree. This particular search uses a recursive algorithm. Also, a tree can be translated easily into a three-dimensional environment so students can actually be a part of the assignment. Finally, the pathways between tree nodes can be created using bridges built with the terrain modification tool.

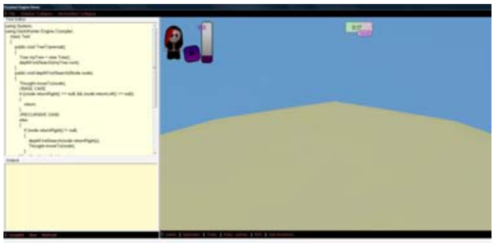


EleMental: The Recursion view with terrain modification

Impact

This game holds to potential to replace or augment traditional coding assignments. In-game compiling allows for a wide range of concepts to be taught effectively in the context of the game.

We have also used a system where the player becomes a part of the coding assignment. Student code directly impacts the game play. If a student uses proper syntax, but not the correct code, the code will be able to run, but the results will be unexpected. Being able to visualize failure as well as success helps to enforce the concepts we are attempting to teach.



EleMental: The Recursion view with second coding assignment

Conclusions

Microsoft's game development framework, XNA, is well suited to the development of an educational game. The EleMental engine has many unique features which allow for an immersive three-dimensional world in which students can interact with their code.

The use of an in-game compiler is tricky and needs a fair amount of fine tuning. However, we stand by our belief that in-game compiling is a better tool for games which strive to be assignments.

Creation of an events manager for the EleMental engine during the development of The Recurrence will allow the EleMental engine to be a platform for more varied games, including RPG's.

Future Work

Currently, EleMental: The Recurrence consists of two levels. We hope to incorporate additional levels to further enhance the instructional value of the game. Dialogue events throughout the game need to be improved in order to better guide the player through the world. We also require much clearer instructions, as this game possesses features which will be unfamiliar to most gamers.

A fair amount of fine tuning is needed with regards to game play. Features such as automatically refreshing the code when the level changes need to be implemented. Also, creation of more varied level maps and challenges needs to occur to avoid repetitive action.

Our technique for parsing through student code will need to be refined in order to allow students to express their creativity and personal coding style while still ensuring that they complete the assignment as directed.

Finally, a study needs to be conducted in order to determine if EleMental: The Recurrence could, as we hoped, serve as an alternative to the traditional coding assignments.