BOTS: GUI Programming for Beginners

Veronica Catete, Joshua Situka, Drew Hicks, Dr. Tiffany Barnes
{vcatete, jsituka, ahicks37, Tiffany.Barnes}@mail.uncc.edu

Veronica, *NC State Univ.,* Joshua, *Univ. Huston,* Drew, *UNCC,* Dr. Barnes, *UNCC*

Introduction

Introductory programming classes contain minimal need for or allowance of collaboration. This may discourage socially oriented students from continuing to pursue their degrees in computing.

The goal of our research is to investigate the effectiveness of community-driven, game-based educational tools using a program called Bots in retaining at risk students.

Background

In using video games to teach education we know from previous work that:

- Systems can make programming more approachable for novices by eliminating syntax errors (Maloney et al.)
- Providing immediate feedback about actions the player makes helps the player learn better (Maloney et al.)
- Visual feedback provided by a graphical interface has positive effects on the attitude and performance of the user. (Barnes et al.)
- Students are more likely to have meaningful interactions in online communities if doing so by game mechanics. (Powell et al.)

Research

This summer, our team started the development of BOTS, a game-based environment in which programming challenges can be:

- Easily deployed
- Presented
- Evaluated

While maintaining student engagement with the material through:

- Dynamic Interesting game play
- Omnipresent social interaction

Our first assignment was to become familiar with the Unity Game Engine using online tutorials. Once familiar with the engine, we moved on to learning Unity's special version of JavaScript. As we grew more familiar with Unity, we started to form an idea of how we would like our game to look.
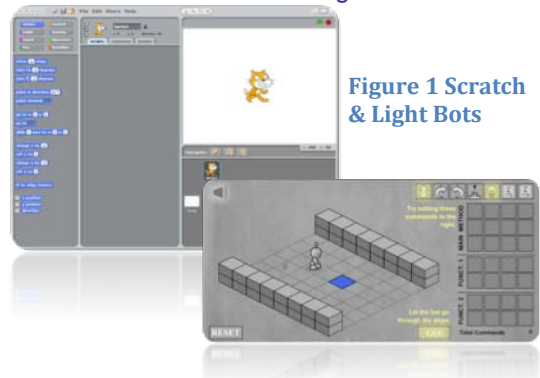

Figure 1 Scratch & Light Bots

We used other games like Scratch & Light Bots for interface design ideas. Combining the simplicity of Light Bots with the functionality of Scratch we began to code our game. Split into two main sections, the programming GUI and the 3D levels, we developed an interface that the user can program in, that will send results to the 3D puzzle world where the robot will perform its actions

Figure 2 The BOTS programming GUI

Impact

Our hypothesis is that students will learn more effectively when using an environment like the one proposed in BOTS because of both the inherent qualities of games as motivators, and the additional motivation provided by the social aspects of the environment.

Additionally, because of the wider demographic of social games, particularly when compared with "typical" games, we hypothesize that the game will help foster interest in computer science, as well as encourage retention in more individuals (particularly more women) than traditional instruction.

Through BOTS, students will:
- Play levels which teach basic computer science concepts like looping, conditional statements, and functions
- Interact with peers by sharing their solutions to problems and commenting on the solutions of their friends
- Be more inclined to participate in CS courses in the future and consider CS as a major, or stay in the major
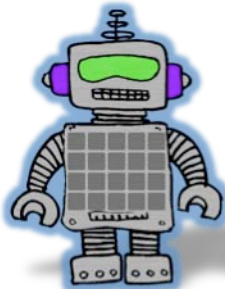
Conclusions

The Unity Game Engine can be used to develop games that incorporate programming concepts. The GUI portion of our game simulates an IDE. This provides many similar features such as modifying and debugging your current program. The lack of strict language syntax is important because it allows the game to teach fundamental programming concepts regardless of language dependencies.

Increasing difficulty in level design has the potential to simulate programming problems that force users to incorporate higher level programming techniques in their game. We can use the Unity Game Engine to create an in game level editor. The player can use the level editor to potentially customize and build their own levels; providing the cornerstone for creative interactions in an online community.
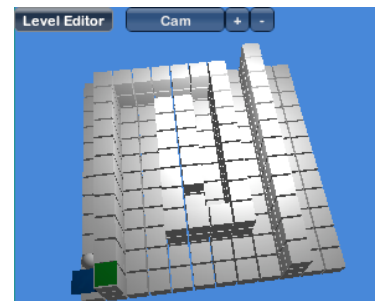

Figure 3 Level 8 of Bots, which requires functions & looping to finish

Future Work

We created a rapid prototype of the BOTS programming GUI. Our next steps include:
- Build the social community counterpart
- Create more levels and programming puzzles
- Allow users to save & share puzzle solutions
- Allow users to create their own puzzles
- Enable hints & high scores