PI: Devorah Kletenik, Brooklyn College, City University of New York  
Co-PI: Deborah Sturm, College of Staten Island, City University of New York

**Summary:** We developed a 3D game to teach and reinforce fundamental programming concepts. An innovative feature allows instructors to create customized programming challenges that students solve in the context of the game. The game was designed to engage and motivate all students, with a special focus on women and racial minorities. A preliminary evaluation of the game in Brooklyn College, College of Staten Island, and Kingsborough Community College indicates promise in the use of the game as a teaching tool.

1 Background

Majoring in computer science fields offers a number of benefits to students, including a growing and lucrative job market [30]. However, many students seem to be unable to progress past the first introductory programming course. Introductory programming courses frequently suffer from high drop-out and failure rates of over 30% – 50% [5, 28, 16]. The attrition rate is often significantly higher for female students.

It is commonly observed that learning to program is difficult and that difficulty may account for the poor retention rates in the introductory class [12, 24, 17]. Our experience with our computing students is that learning to program demands significant practice time. Students learn how to program by programming, and the more students program, the easier it is for them to master the material and become successful in the course. However, many introductory courses suffer from a lack of sufficient programming practice [4].

The literature on serious games discusses strategies to motivate students to practice programming. Researchers report that the long-standing formal lecture teaching mode is the least popular with students [27] and less effective than more interactive modes of learning [18]. Serious games describes those games that accomplish a goal in addition to entertainment. Research on serious games for education shows that, compared to traditional methods of instructor, game-based learning is superior at teaching subject matter [26, 15, 23, 31, 29] and increases both long-term retention and student motivation [26, 23, 13, 29]. Students who learn through games have increased feeling of alertness, activity, and involvement in contrast to standard lectures [11].

We therefore developed a serious educational game to help engage, motivate and improve the learning experience for students taking the introductory programming courses at our institutions and elsewhere. Our innovative platform, called Code Control, is a game that helps reinforce programming concepts where instructors can create customized programming challenges that students solve in the context of the game. An automated compiler and testing environment gives students instant feedback, and completing successful programming challenges advances students in the gameplay.


2 Code Control

Code Control is a digital game developed using the Unity 3D platform, that teaches and assesses programming concepts. It allows instructors to create customized programming challenges for their students as well as provide correct solutions. The programming challenges are then posed to the players in the context of the game. We use the Judge0 API\(^1\), which is a free open-source API for code compilation and execution.

Code windows give players code snippets with missing sections to complete; once the players fill in the missing code, the code is compiled. The students’ solutions are then checked for correctness by comparing them to the instructor’s solution to see if the output matches. Compiler errors, execution errors, or positive feedback are provided. Customized scripts provide syntax highlighting. Judge0 supports 42 programming languages; so that any introductory language will most likely have support in our game. Furthermore, with Judge0 support for multi-file programs, Code Control can be used in coding courses at all levels.

While a number of serious games for computer science have been created, including some to teach introductory programming (e.g., [1, 2, 3, 8, 10, 9, 22, 19, 20, 21]), all of these games involve fixed programming challenges. In contrast, our approach allows instructors to specify their own challenges that best reinforce what is being taught in class. Additionally, many of these games (e.g. Gidget [19]) use a simplified programming language that was created for the game, and are designed to teach abstract programming concepts. In contrast, our game uses standard programming languages and is designed to augment existing programming courses. It is targeted to students who are learning the basics and want practice to solidify their skills.

The game’s storyline involves a woman looking for endangered animals who are missing from an animal rescue. The digital name tags contain code that was broken by enemies. The player needs to find the lost roaming pets by solving the code to fix their name tags, to save the pets. The short programming challenges act as mini quizzes based on research of the “testing effect” that indicates that tests actually improve learning [6, 7, 14, 25].

In designing our game, we consulted with students, both gamers and non-gamers, men and women, and varied races to appeal to a broad audience. It has been particularly designed for a female audience, featuring a female lead character, avoiding violence and focusing on social goals. The avatar’s customizable skin/hair colors avoids specifying a specific racial group and gives players co-ownership.

A database connection allows us to collect detailed analytics of all users’ gameplay (e.g. time spent per challenge, successes and mistakes made, levels completed, score, etc.) This provides a wealth of information to instructors as to how this game is being played and what students are struggling with and succeeding at. Instructors who adopt this game as a course material will be able to use the game as an informal evaluation mechanism to see which concepts their students are successful with and with which they need additional practice. Code Control is deployed as a WebGL and can be played in a browser without requiring installation.

Pictures of the game are given in Figures 1 and 2. Figure 1 depicts the student views of the gameplay. Figure 2 shows instructor and student views of a problem set.

3 Evaluation

We conducted a feasibility study of a preliminary version of the game at the end of the Fall 2018 semester. Students were recruited from introductory programming courses at Brooklyn College.

\(^1\)https://api.judge0.com/
College of Staten Island, and from Kingsborough Community College. These three colleges are part of City University of New York (CUNY), the nation’s leading urban public university and the most racially and ethnically diverse university system in the country. The three CUNY colleges were chosen because together they represent a variety of student experiences within CUNY (senior, comprehensive, and community).

Participants in the study were asked to take a short programming quiz, play the game, and fill out a survey (including another short programming quiz). In total, 75 students joined the study. We present the institution and genders of the players below:

<table>
<thead>
<tr>
<th>institution</th>
<th>total</th>
<th>male</th>
<th>female</th>
<th>preferred not to say</th>
<th>% female</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brooklyn College</td>
<td>52</td>
<td>36</td>
<td>15</td>
<td>1</td>
<td>40.5%</td>
</tr>
<tr>
<td>CSI</td>
<td>20</td>
<td>13</td>
<td>7</td>
<td>0</td>
<td>53.8%</td>
</tr>
<tr>
<td>Kingsborough</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

We note that female students were disproportionately represented among the participants, compared to their representation in the programming courses at Brooklyn College and CSI (approximately 31% and 18% respectively). Moreover, they seemed enthusiastic about the game, even though it was still in “beta” mode and under construction, with survey responses such as “It seems like a fun and engaging way to teach programming.”, “Fun and helpful!”, and “Brought a twist to coding that made programming a lot funner than the cut and dry homework assignments.”

In our pilot, a number of students could not complete the entire game due to browser incompatibilities. Although our focus was on measuring student interest, we also noted that the game had a measurable effect on student performance: the students who played the entire game had statistically significantly higher scores on post-tests than on the pre-tests (by an average of about 6%); the students who did not complete the game showed no such learning gains. We are confident
that when the game is further developed, student interest and performance will correspondingly increase.

(As a result of the feedback on the game, we conducted a dramatic overhaul of the structure, design and story line to make it more flexible, appealing and fun. As a result, the game was not yet ready for evaluation by the end of the Spring semester. We hope to conduct another study during the Summer semester.)

4 Funding

The SIGCSE Special Projects funding of $5,000 was used to support the work of an undergraduate student who developed the game. We aim to receive future funding from other sources to continue our work with Code Control, including creating a dedicated instructors’ portal to see student progress; to make the game more flexible in allowing instructors to pose non-programming challenges (e.g. tracing, Parson’s puzzles); and to conduct a more comprehensive evaluation of its effectiveness, particularly for female and other underrepresented students.

5 Conclusion

We’d like to thank the SIGCSE Board for supporting our work. We are hopeful that it will have an impact on students taking the introductory programming course at CUNY as well as at other institutions.

References


4


