Coding VR Games

J. Chen¹, M. R. Zargham¹, M. Rajendran¹, and J. Cheng²
¹Department of Computer Science, University of Dayton, Dayton, Ohio, United States
²Department of Computer and Information Technology, Miami University, Ohio, United States

Abstract - This paper represents the progress of our ongoing effort on a new way of teaching an introductory programming course while incorporating a game strategy. Students use a simple environment, called Innovative Coding (IC), to write a code for designing a game or a virtual world. As they code, they see the development of their world in a 3D environment step by step.

This paper describes the addition of a new module that allows students to design a game in a breathtaking landscape. Before the game starts, the user (the student who will set the game) will write codes for a set of robots that prevent the player (the individual who will be playing the game) from reaching to the destination. The game starts when the player found himself/herself trapped in an island and needs to find a set of clues to reach the final destination. The robots will attack the player on the way to the destination. Throughout the game, the user will see the effectiveness of his/her code by the degree of the player’s success.

Keywords: VR Headset; Coding; Game; Virtual Reality; Education; Programming.

1 Introduction

One of the most effective uses of VR is in the education field. Within the last three years, we have developed a software package, called Innovative coding (IC) that allows students to learn coding in an engaging environment. IC is intended to encourage more female students to pursue computer science as their major in college [1, 2]. IC allows students to write simple codes consisting of several statements to create a world of their choice in a 3D and Virtual Reality environment. VR goes beyond a simple visual stimulus, and allows students to become directly involved, experiencing the code they write in a very tangible, interactive, and expeditious scheme (strategy). This allows them to experience both the successes and failures of writing (creating, developing) correct code versus incorrect code. Students can appreciate both the scale and the scope of the results of their code immediately, which prompts them to become invested in the quality of their coding skills and strive to continually improve their logic, programming, and code writing abilities.

Through this Immersive and Responsive Visual Stimulus Learning, students become more engaged in coding. In general, the idea is to teach the introductory programming course in such a way that follows the project-driven learning process and encourages students to develop problem-solving and teamwork skills while fostering creativity and logic.

This year, we were able to extend the functionality of IC by adding a game development environment. Our goal of designing the game is to help 9- to 13-year-old students to learn coding and problem-solving in an engaging environment. They can see the progress and result of their code in the 3-D environment. Different types of interaction and excitement are possible through different code complexity of the main objects in the game. By following step-by-step instructions and observing the game outcome, students can learn how to develop, test, run and debug a computer program. They learn a disciplined and structured approach to their program development. They will also learn main control structures of procedural programming languages, becoming familiar with the concepts of loops, assignment statements, and condition statements while others are playing their engaging and interactive game.

Besides 9 to 13 years old, students in K-12, computer science, electrical/computer engineering, and business, students from other disciplines such as mathematics, physics, chemistry, biology, communication, and arts are also encouraged to utilize IC for learning how to code.

2 Game Environment

There are four islands with different climates: Japanese Garden, Volcano, Athens ruins, and Ice islands. These four islands shown in Figure 1 are connected with bridges allowing the player to travel from one island to the other.
Figures 2 to 4 represents the Japanese Garden which is calming and growing. This island is built around an ancient Japanese tradition with Bonsai trees, water, stone, the lantern, and bridges.

Volcano Island, shown in Figures 5 to 6, is located on the north side of the Japanese Garden. This island has a sharp contrast with the Japanese Garden. It is an active volcano forming a series of boiling lava flows. The eruption is throwing lava into the sky from the top of the mountain, and molten rock pouring from the vent. The entire island surrounding with gas, heat and is covered by ashes.

After the volcano island, players will arrive at the Athens Ruins, see Figure 5. The Athens ruins demonstrate a vivid picture of how impressive the ancient Athens once was and now is covered by a tropical forest. Small bricks are the foundation of an ancient temple; stand-alone columns are all the remnants of a building built over 2,000 years ago.
3 Game Strategy

Before the game starts, the user can place robots, called enemy robots, anywhere in the landscape and give them life through action commands within the command window. Also, the robots possess intelligence based on the user code that allows them to recognize the player within a specific range and destroy her/him through their personalized (program) attacks. During the game, the player starts from a point in the Japanese Garden and explore the island to find her/his first hint. The hint will be marked with a balloon that floats up and down in the air shown in Figure 9.

When the player locates the first hint, he/she will shoot the balloon to activate a hint window. The window will represent a programming question to the player. The player needs to answer the question. If her/his answer is correct, the next hint is activated and the player directed toward a shortcut. Otherwise, the player is led toward a complicated path for reaching the next hint.

The hints are located in sequential order, one on each island. Therefore, the player will go through all four islands in order while enjoying different perspectives of different climates. On her/his journey to find the hints, the player might come across enemy robots that have been set up and programmed with the user before the game starts. If the player gets into the robots premise, the robot will follow her/him and starts attacking according to the command. At this time, players can either run away to keep far from danger or shoot the enemy robot with a laser gun and earn bonus points.

The end point is located somewhere on the ice island. When the player finds the last clue, he/she celebrates his/her victory by saving the scores. Among different players, the winner will be the one with the highest points in the rank. Points are calculated based on the time consuming to finish
the game, enemy killing level, and question-answer sessions provided by the hint window.

4 Programming

The syntax of our programming language is similar to Java. The language supports statements such as variable declaration and initialization, assignments, loops, condition, arrays, and functions.

There are two types of loops, “for” and “while” statements defined as below:

```java
for (initiation, condition, increment/decrement)
{
    Statement1;
    Statement2;
}
```

```java
while(condition)
{
    Statements;
}
```

The syntax of “if-else” statement defined as below:

```java
if(condition)
{
    Statements;
}
else
{
    Statements;
}
```

User can declare and initialize one or two-dimensional arrays. Below is the syntax of declaring an integer array:

```java
int array_Name(size of the array);
```

Moreover, several function statements can be called to perform specific tasks. The syntax of some of these functions are shown below:

```java
//Display values to the console;
display variable_name;
```

```java
//Display string values to the console;
print string;
```

//control movement of objects
walk float direction float;

//Detect enemy in a certain region
findEnemy(distance);

//attack number of seconds
attack int;

The user can utilize the above statements to write a controller for each robot. For example, the following code represents a simple controller for an enemy robot.

```java
while(1<2)
{
    if(findEnemy(5))
    {
        attack 10;
        print “hello”;  }
}
```

5 Coding Camp

The Department held a coding camp for the students in fifth to eighth grades in summer 2018 [3]. This workshop was a project driven learning platform that helped students to learn programming through developing their own virtual world/ game with their creative and logical/critical thinking skills. Throughout the coding session shown in Figure 10 and Figure 11, students were inspired and learned a well-structured approach to developing computer programming. Students designed and wrote simple codes for their worlds games with the learned concepts of variables, assignment statements, conditional statements, and principles of data storage.

Students were very keen to learn programming by applying their learned concepts practically. The camp showed that VR significantly enhanced student learning ability. This is because students were able to see the effect of their codes and engagement with their design.

Unlike writing complex code, students applied simple assignments, control statements to define the length, direction, and angle of each section of a roller coaster track. Students could see the progress and result of the track being
created with each new statement. Below is an example of how users write code for designing a roller coaster.

During the camp week, students went home every day with excitement and were excited to get back the next day! At home, they were thinking of ways that they can improve their programs.

According to the post-camp surveys, 52 out of 53 students listed programming a roller coaster and riding it in VR as the most attractive and memorable aspects of the experience. Almost all students indicated that the camp was fun, exciting and educational. Many also expressed that they desired to come to the camp next year and maybe even pursue computer science as a career in the future.

6 Conclusions

In this paper, we have described a 3D environment, IC, that can enhance learning how to code for 9 to 13 years old students. Every student that has tried our system has indicated that his/her experience has been productive and fun. With the new addition to the IC, students are able to design a game in a breathtaking landscape. The game requires programming of the enemy robots by the students. This is where students can challenge themselves by writing a simple controller for the robot or more advanced code that incorporate learning by experience. In near future, we are hoping to have another summer camp to evaluate the effectiveness of such game.

7 References