Timothy Hoang

Professor Rivas

5/8/17

Project 2 Writeup

"Rage Circle"

Anstract

Over the last couple months, I have been taking Software Development I at Marist College. As a final project, I chose to develop my skills with graphics in Java Programming. Specifically, I created a game called "Rage Circle". The rest of this paper shows my process in the development of my program from inspiration, to learning new techniques, to the coding of my project.

Introduction

The Java programming language that is general-purpose, concurrent, class-based, and object-oriented. Using knowledge that I have gathered and refined from Software Development I at Marist and Youtube videos, I have used the Java programming language to create a fun game. I chose to make a videogame because I wanted to learn how to make a graphics-based program that the user can interact with in real-time. Creating a videogame seems to have been the best choice for learning that path of Java programming since videogames are not only extremely reliant on visuals (aside from text-based games), but also require constant interaction from the user to run properly. Although I have not learned how to create visual graphics in my Software Development I class, I have learned to organize and classes and allow them to interact with each other in a meaningful and efficient manner. From Youtube, I have learned to take advantage of

Processing core software to create graphics in a window that are drawn and altered by my Java program. The coding used contained processing.core in its heart to create basic graphics to go along and make the game visually exciting. The reason I chose this is because I researched Java graphics and discovered that processing is simple to use and easy to learn. Due to this factor, and its popularity, there were many Youtube tutorials on the internet available for my use to learn how to create shapes and movement with Java. However, a problem that occurred from using the processing library is that it was only compatible with Eclipse, making my program only runnable in Eclipse itself instead of console. The math used in my program was constantly tested and edited and it still is not perfect. The system takes advantage of the classes I composed, breaking down each aspect of the game into several, manageable chunks. I also created a simple game which allows almost anyone to run it and play with little to no instructions on how to play the game. Overall, the game was created with skills that I learned in class and outside of class in order to hone my skills as a programmer. I chose an accomplishable but not impractical goal by making this game and now I am also more confident with my skills and am ready to tackle on bigger and better challenges.

Detailed System Description

The system consists of several classes to maintain an image of several rotating gates around layers of circles surrounding an end goal in the center of the game screen. The user avatar, a pixel is able to move in all directions from some sort of key input as well as having the game end if you touch a circle or when you reach the end goal. This was accomplished by first having default values for the speed of the rotating gates, character avatar, and degrees of rotation. Then, create a movement class for the avatar so that it can move in four directions; up, down, left, and right as well as any combination of those directions. Two more classes were created in conjunction with the movement class for the avatar one class for the key being pressed to start the movement in the designated direction and another class to stop the movement when each respective key is released by the user. A class was also made for tracking whether your avatar reaches any point of interest and depending where it was, you win, lose, or continue the game. Another class was created to draw each shape and allow some of the shapes passive movement, meaning that it does not require user input to move. This was used for the rotating white rectangles on the circle that move around each circle creating a “gate” illusion. This was also used for the actual points that the gate used although those points are invisible unless you chose to un-comment the test code lines which revealed the points in real-time. Lastly, a class was made to track and move the gates around the circles, allowing the avatar to pass through the circles when between two points.

Draw
Ellipse; rectangle; rectangle
Draw circle; draw goal; draw avatar

Move Avatar
Move (up, down, left, right) (Boolean)
+x, -x, +y, -y

Key Pressed
Pressed (w, a, s, d) (Boolean)
Up, down, left, right

Key Released
Released (w, a, s, d) (Boolean)
Up, down, left, right

Gates
Position (float); rotate
Posx; posy; speed

The math used in my program is still not perfect as the gates are truly not aligned with their visual counterpart.

Requirements

The program requires Eclipse to run rather than out of the console solely because Processing only supports the Eclipse and its' own integrated development environment. Other than that, most modern computers will be able to run the program as it not labor-intensive and vigorous.

Literature Survey

Current systems that exist which are similar to "Rage Circle" are other hand-eye coordinated games that exist on website gaming platforms. However, I was unable to learn from their code since it is inaccessible to the public. I do not know of any other games that are exactly the same as what I imagine, but I received inspiration from similar games that had mazes and simple character design.

User Manual

I made the w, a, s, and d keys for movement of your avatar according to where they are placed relative to each other. Also, any combination of up, down and left, right will allow you to move diagonal for more dynamic movement. When changing the speed in the code, if you make the default speed for the avatar too fast, you are able to make it to the end goal by zooming straight towards it from a standstill position since the avatar will move too fast for the program to catch it going through the circles regardless of gate position.

Conclusion

Overall, I hope to take advantage of this learning opportunity in order to expand my understanding of Java. By combining interesting aspects of computers such as visual graphics and real-time user interaction, I expect to expand my capabilities in creativity and design in Java programming for my future in software development and computer science in general. In the end, this challenging project will evolve my interest in programming as well as allow me to push myself to develop new skills and become better in my field. Overall, I am now more knowledgeable in the field of programming and am also more confident with my skills. I feel prepared and ready to tackle on bigger and better challenges.