Assignment 1 report - INF-1400

Marius Isaksen

February 17 2022

1 Introduction

In this assignment, we make a breakout clone using OOP principles, with an emphasis on using classes and methods. The libraries that were used was Pygame and OS. The coding is based on rectangles and circles that together make a game called breakout, with the help of functions for a paddle, rectangular blocks and a circular ball.

2 Technical Background

The coding is based on rectangles and circles that together creates a game where you have a movable paddle that you can steer from left to right. You then have to hit a circular ball on to colorful rectangular blocks. The blocks then loses strength and disappear. If you hit every block you will win, but if you miss the ball with the paddle, the game will end, and you will lose.

The technical background behind this is made with the help of the Pygame library and OS library. The Pygame library is a set of python modules designed for writing video games. This adds functionality and the wonderful SDL library [1]. SDL is a library that brings the joy of implementing keyboard, audio, mouse and graphical hardware [2]. The OS library helps when you need to interact with the operating system. This was used when implementing music and sounds.

3 Design

The design is made so the coding is easy to read from top to bottom. As you can see in figure 1 the game elements are divided in different classes called paddle, circle, rectangle and the function called extra. These are mainly based on variables, if-else sentences and some loops. The rectangular blocks have different colors based on how many times you must hit them to disappear. This is showed on Figure 2.

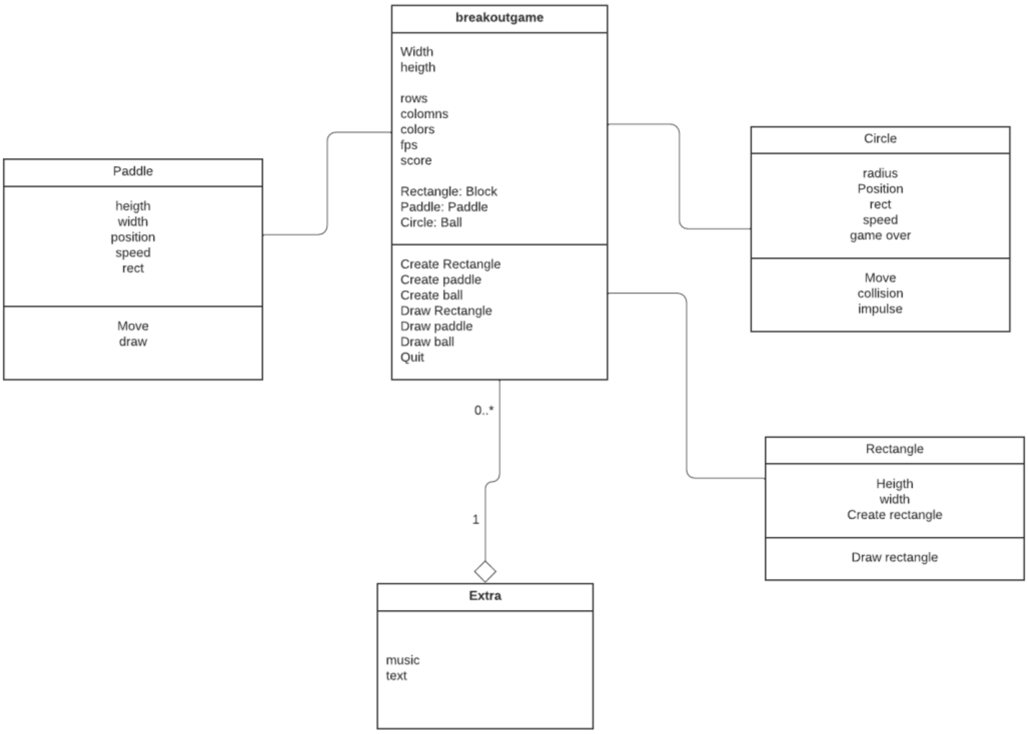

Figure 1: UML diagram of the coding

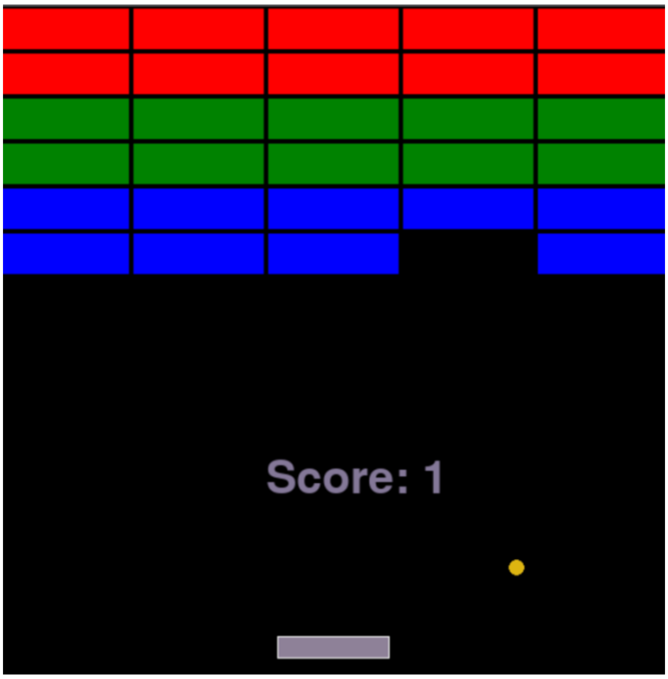

Figure 2: The design of the game

4 Implementation

For the implementation of the collision between the ball and the blocks, the if sentences were based on absolute values of the numbers. By representing the collision in this way, we erase all the negative numbers. This makes it so the collision doesn't run into any errors with negative values. [3]

5 Evaluation

In this solution, all requirements are fulfilled, but collision detection between the ball and paddle could be done better. This is because when the ball hits the paddle it will always go in the same pattern since it doesn't matter where on the paddle it hits unless it hits on the left or right side which is very unlikely. Therefore, it would be better if it mattered where on the paddle the ball hit.

6 Discussion

The implementation of vectors instead of tuples made the coding a little bit harder, but because the numbers were based on coordinates, I understood that it was easier to use vectors.

7 Conclusion

I have implemented a solution that fulfills the requirements, the implementation is based on OOP coding with the help of classes and methods. The game is made by rectangular and circular objects. The functions are made with the help of variables, if-else statements and some loops.

8 References

[1] Pygame library

URL https://www.pygame.org/wiki/about

[2] SDL library

URL https://www.libsdl.org/

[3] Python breakout help on creating the objects and collision

URL https://www.youtube.com/watch?v=NIfkaOF3Hjs&ab_channel=CodingWithRuss