Odd Semester (2020)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work****)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | |  | |  | |
| **Student Information**: **Surname** | | | | | **Given Names**  **Jason** | | **Student ID Number**  **2201796440** | |
| 1. | | **Sianandar** |  | |
|  |  |
| **Course Code** | **: COMP6056** |  |  | | **Course Name** | | **: Program Design Methods** | |
| **Class** | **: L1AC** |  |  | | **Name of Lecturer(s)** | | **:** 1. Minaldi Loeis | |
|  |  |  |  | |  | |  | |
| **Major** | **: CS** |  |  | |  | |  | |
| **Title of Assignment**  (if any) | : SPEEDING! | |  |  | |  | |  | |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** |  |  | |  | |  | |
| **Due Date** | **: 20 – 11 - 20178** |  |  | | **Submission Date** | | **: 20 -11 - 2018** | |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

Binus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to Bin us International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. Jason Sianandar

**FINAL PROJECT REPORT**

**Name**: Jason Sianandar

**Class**: L1AC

**Introduction**

In the first semester of university life, as a Computer Science student enrolled in Binus International University, we are obliged to take “Program Design Methods” and “Introduction to Programming” courses. In these courses, we would learn the programming language, “Python”.

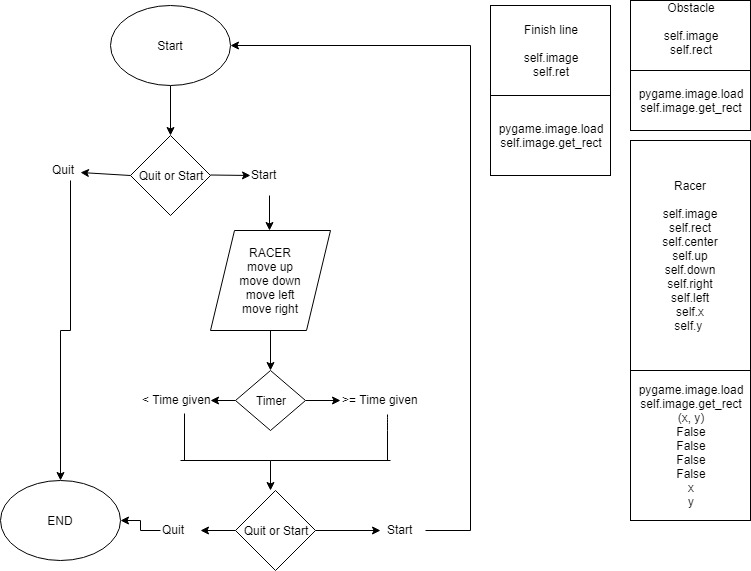
“Python is a high-level programming language designed to be easy to read and simple to implement. It is [open source](https://techterms.com/definition/opensource), which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to [Java](https://techterms.com/definition/java) and .NET virtual machines.”(techterms.com, 2018)

In this report, I will be using the Python programming language to create a pygame which is basically designing and developing a game based on the python language and engine. For this pygame, the game that I will be making is a motorcycle racing game in which you will race against the time or you can call it as time attack but there will be obstacles in the tracks which are trees and that will slow you down.

**Problem (Why this program?)**

First of all, video games has always intrigued me since I was a kid. I would always get crazy everyday and play video games in my PlayStation 2. As time went on, and I know that I can play video games on PC and laptop computer, I started buying video games and installed it on my laptop. I’ve installed one hundred games in my laptop throughout my life and I would often dream that I can design my own video game. The point is, I wanted to create a video game and Python has granted me this opportunity and I took it.

Secondly, I like to ride motorcycle. By motorcycle, I mean like a really big motorcycle (>250cc). I have a Honda Sport Naked motorcycle. I ride it almost everyday. I like to speed with the motorcycle and do cornering in tight junctions in the street. I also like to watch MotoGP racing competition and from these two reasons, I am inspired to create a motorcycle racing game but it doesn’t just stop there. When I first learned pygame, we were asked to make the alien invasion game and from there I get the inspiration that I create obstacles in the track so the game will be a time attack race while there will be obstacles.

**Design**

**Discussion**

**Implementation**

The API that I used is only pygame, time, random, and sys:

* Pygame = I used pygame for the main framework. Without using pygame, I cannot draw images, game functions, game methods, and make the game in Python
* Time = I use time to configurate the frames per second (FPS) rate in the game and also for creating the timer in which this is the time attack so the player have to beat the time given to win the track.
* Random = I used random so I can spawn the obstacles which in this case, the trees randomly in the track and the player have to avoid the obstacles so it doesn’t slow his lap time.

I also used a lot of def functions and classes for this project.

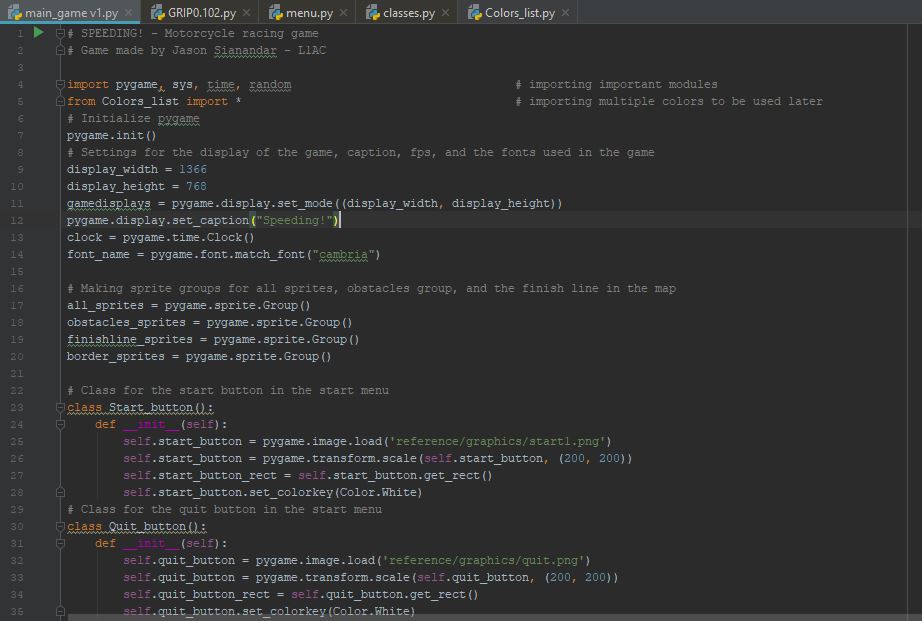
**Gameplay**

In the start menu, you will see a logo of start and quit. In the menu, there will be an instruction to press return key to start the game, or q to quit the game. If you press the return key, there will be the game manual shown for 5 seconds and the race will start. You use the arrow keys to move the racer in the track, you must avoid the obstacles to reach the finish line before the time ends to win the game or you will fail but you have the option to go back to the start menu or quit the game.

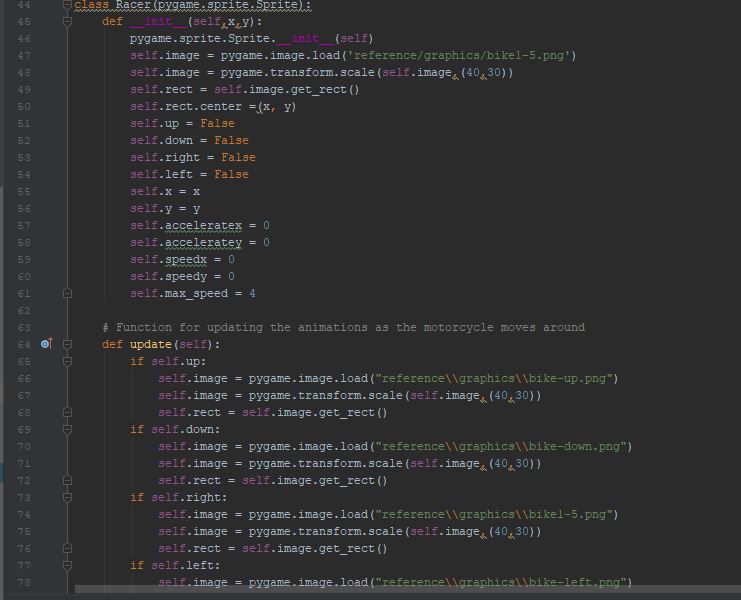
**How it works**

How the game work is simple. First, I make the start menu using 2 separate functions. One function to blit the entire start menu, the other function is so I can press some keys to either play the game or quit the game. In the start menu you have the options to start the game or quit the game. If you press start which in the program you press the return key, the program will blit a 5 second game manual screen that shows you the controls to play the game and the race will start in seconds. During this manual screen, I use pygame.time.wait to make the screen pause 5 seconds before going into the game. In the game, you use the arrows in your keyboard to move the racer throughout the track. In this track, you are racing against the time. You must complete the track before the time limit or exactly at the time limit but even though the track is easy but there will be trees that will be blocking your way and that will be the difficult part. I use classes and multiple functions for this and also use the pygame.event so I can use to the arrow keys to move the racer. I blit the trees randomly using random on the track. After you finish the track whether before the time runs out or after the time runs out, the game will blit the game over screen in which you have the option to go back to the main menu or quit the game. In this game, there will be sound effects when you move the racer and hit the obstaces. This is all made within a while loop with multiple variables and many functions and many loops inside the main while loop.

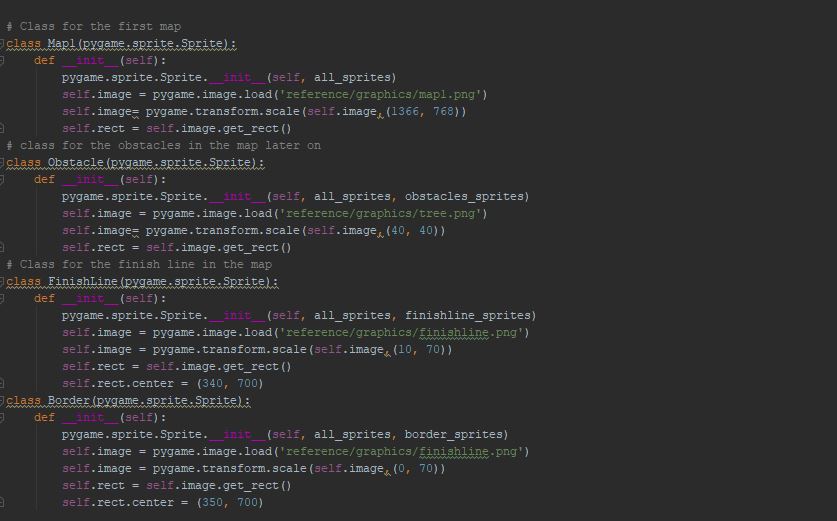
**Code Explanation:**

****

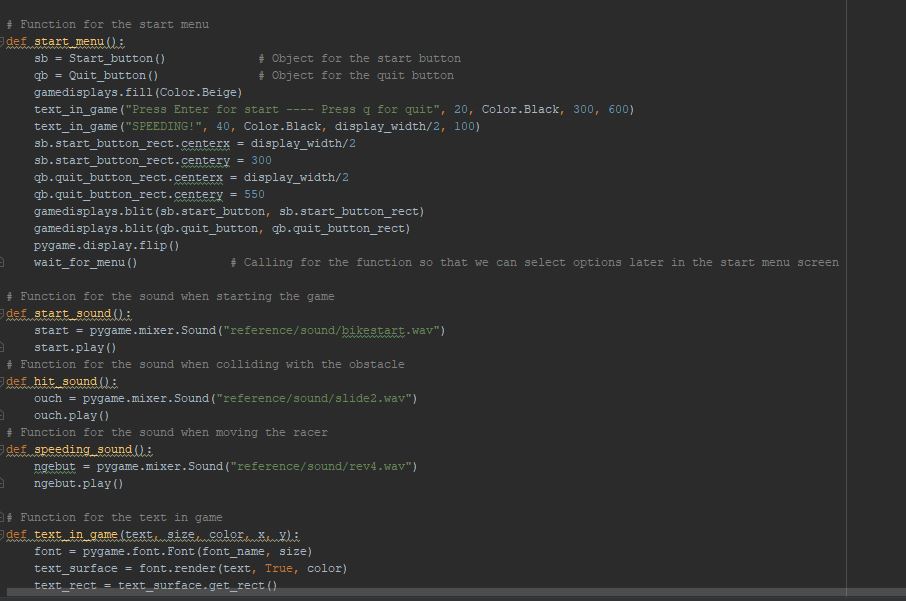
In this image, I import the pygame API, sys, time, and random and I also imported a python file which consists of multiple classes of colors that allows me to use a color anytime I want. The rest of the code here is the configuration for the resolution of the game, grouping all the sprites so I don’t have to blit them one by one. Also show the classes for the start and quit button which will be used in the start menu.



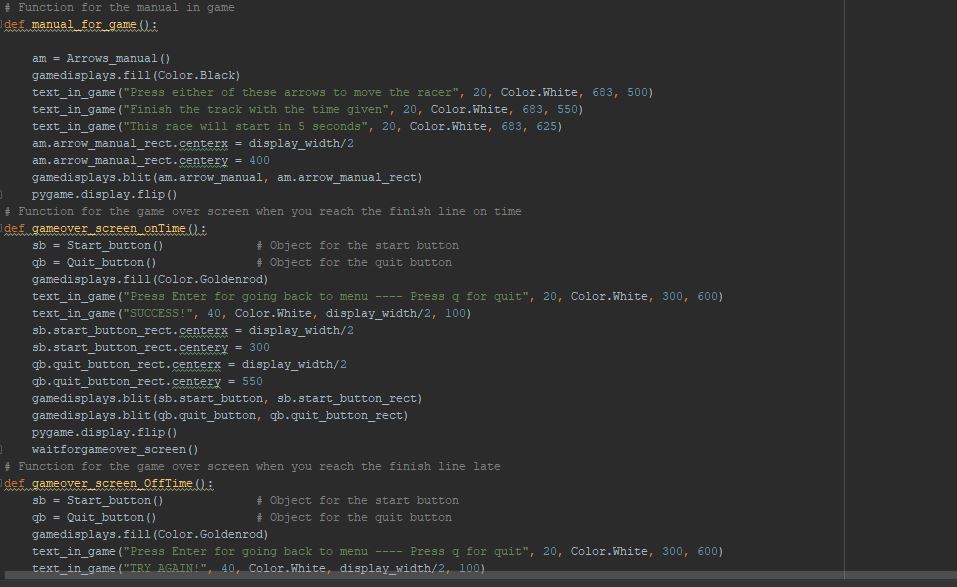
This is the class for the racer. Inside there is multiple attributes like the image itself, self.rect., and also the function to update the images of the racer as its going up, down, right, and left.



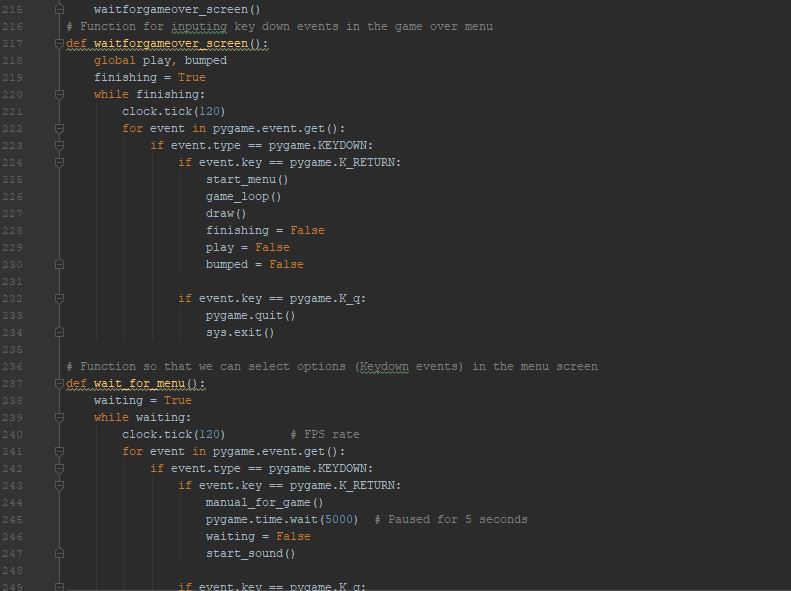
This is the classes for the Map, Obstacle, Finish line, and the border (not used in the final product). Inside the class of Map consisting the attribute of the image and rect. It is the same for the 3 other classes in the code above.



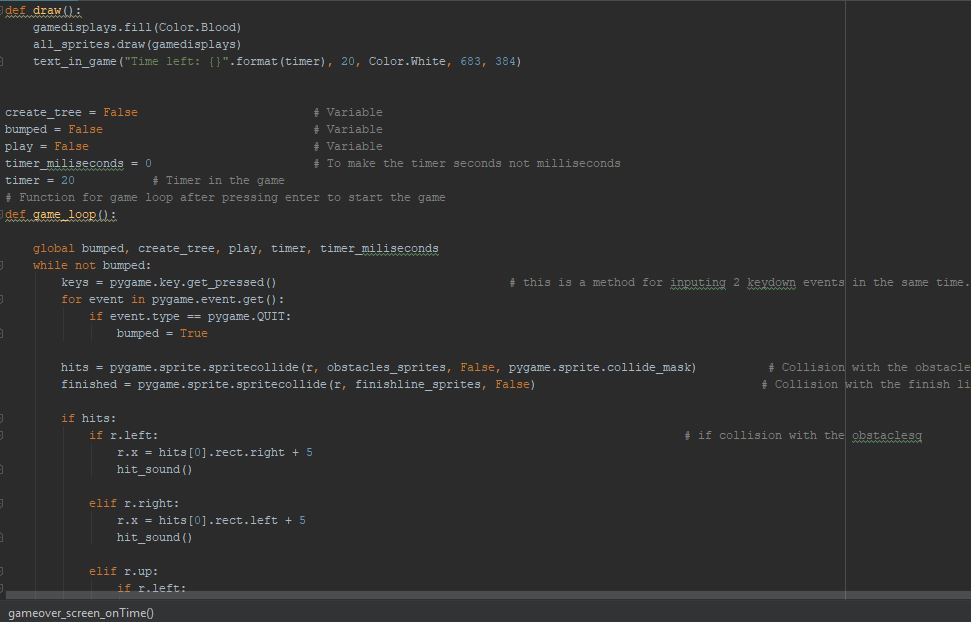
This is the function for the start menu in which I call two objects from their respective classes which are the start buttons and the quit buttons. I also made functions to be called later in the program for the sound effects. I also made the function for text in game so I can call it anytime I want to blit a text in the program.



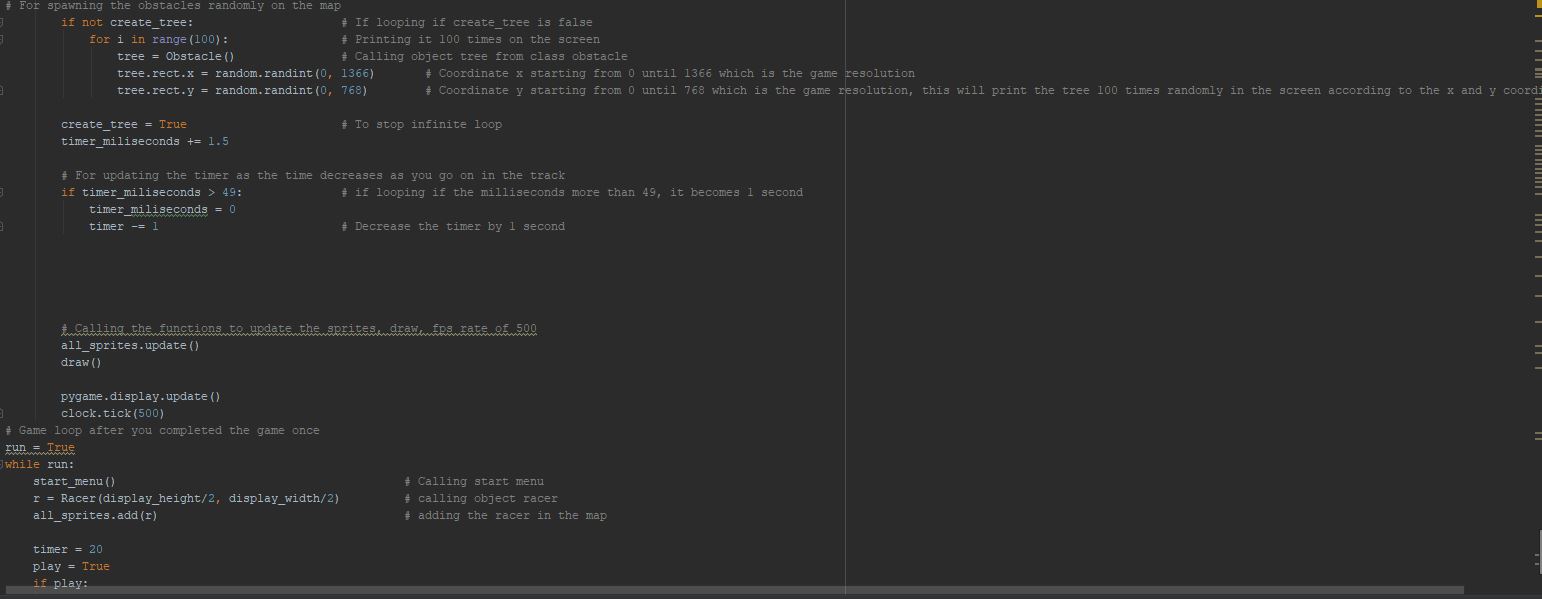
In the code above, I made the function for drawing the game manual screen after the main menu. Below is the functions for the game over screen when you finish the track on time and late.



The code above shows two functions in which I can input key down events which are the return key and the q key for the start menu screen and the game over screen. To do this I need to create a variable and call it as True. After that I use a while loop based on the variable and inside the loop, I can input the key down events.



This code above shows the function of draw in which I can draw everything on the screen and also shows multiple variables that I made global so it can be used in any while loop. I also created the variable timer so the game have a timer that the player must race against. Finally is the game loop. I used while loop based on bumped variable. Inside the loop is the controls that the player can input to move the racer inside the game and also the speed of the movement and the direction based on the key down input by the player. Inside the loop also consist of the variable hits and finished. Hits is the variable in which the racer collide with the tree from any direction will cause the racer to stop moving and the player must move in the other direction. Finished is the variable in which the racer collide with the finish line.



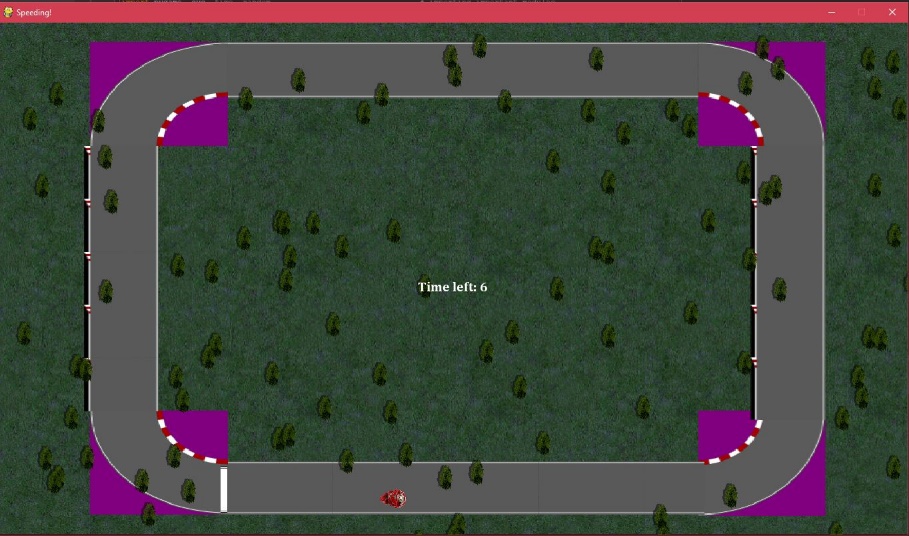
The code above shows an if conditional based on not create\_tree. In this loop, it shows that I call object tree from the Obstacle class to print it 100 times randomly based from the coordinate x, 0 until 1366 and coordinate y, 0 until 768. I also created the variable timer\_milliseconds. I create another if conditional in which if the timer\_milliseconds is greater than 49 then the timer inside the game will be decreased by 1 second. Because if I don’t use this, the timer will count itself in milliseconds instead of seconds.

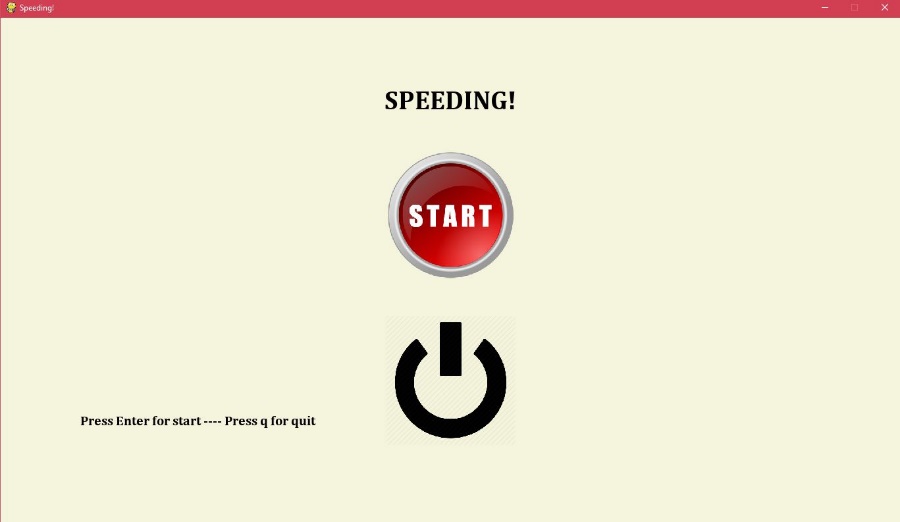
**Class Explanation:**

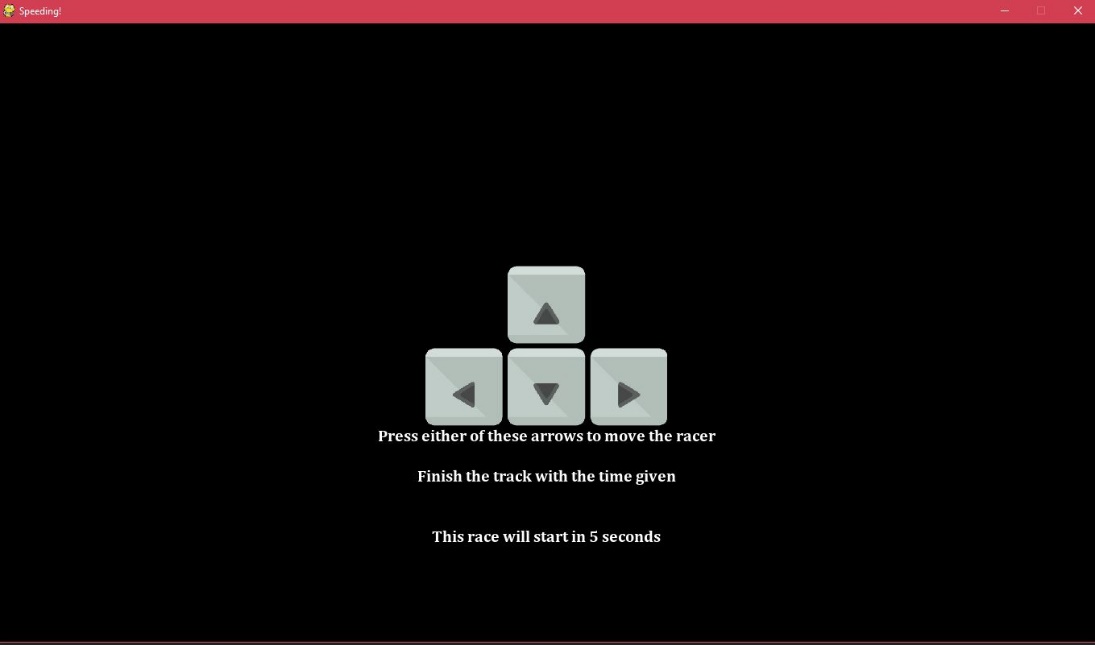
I created 4 classes for the main game which are Racer, Map1, Obstacles, Finishline. The function of the racer class within the program is so the program can blit the racer image and also transform the scale of the image and also enables the user to move the racer in the game. The function of the Map1 class within the program is so the program can blit the track and transform the scale of the image. The function of Obstacles class is to blit the image of the trees in the game, transform the scale of the image, and further spawned randomly using random and can be used to collide with the racer. The function of the Finishline class is to blit the finish line image in the game, transform the scale of the image and also use it to collide with the racer as the player finishes the track.

**Result**

Screenshots

****

****

****

**APA BIBLIOGRAPHY**

* Python. (n.d.). Retrieved November 19, 2018, from <https://techterms.com/definition/python>
* Images = Stuart Laxton (GRIP), 2018
* Debugging = Muhammad Erizky Suryaputra, Fernanda Dzaky
* Animation: <https://www.youtube.com/channel/UCej-wawhhPdjVKihCRk2Ang>, 2018