Mini Project Report

on

Flappy Bird Game Using Python

Submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

Session 2020-21
in

Information Technology

By
Harsh Sharma
1900320130069

Ms. Nandita Goyal
Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
ABES ENGINEERING COLLEGE, GHAZIABAD

ABES Estd. 2000		NBA NATIONAL BOARD OF ACCREDITATION

AFFILIATED TO
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW
(Formerly UPTU)

Student’s Declaration

I hereby declare that the work being presented in this report entitled **“Flappy Bird Game”** is an authentic record of my / our own work carried out under the supervision of Ms.**Nandita Goyal ,Assistant Professor, Information Technology.**

The matter embodied in this report has not been submitted by me / us for the award of any other degree.

Date:December 5,2020

Signature of student
(Name: Harsh SHarma)
(Roll No. 1900320130069)
Department: Information Technology

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of HOD	**Signature of Coordinator**
(Name: Mr. Amit Sinha)	**(Name:Ms. Nandita Goyal)**
(Information Technology)	**(Assistant Professor)**
	(Information Technology)

Date:December 5, 2020

Acknowledgement

The satisfaction that accompanies the successful completion of this project would be in complete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. I consider myself privileged to express gratitude and respect towards all those who guided us through the completion of this project. I convey thanks to my project guide Ms.Nandita Goyal of Information Technology Department for providing encouragement, constant support and guidance which was of a great help to complete this project successfully. Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

Signature of student
(Name: Harsh Sharma)
(Roll No. 1900320130069)

Table of Contents

INTRODUCTION

My project name is “ **Flappy Bird Using Python**” In this project, we design and implement a Flappy Bird like video game. Flappy Bird is a very popular mobile game on Android platform,driving a lot of people crazy. In this game, the player can control the vertical movement of bird (everypressing on the keyboard makes the bird leap upward for a little bit,and the bird will fall freely without control).As soon as the game begins,tube will keep appearing from the right side of the screen and start moving leftwards.The goal of this game is to control the bird, dodging and passing the incoming tubes, as many as possible. The game is endless until the bird eventually hit one of the tubes.

1.1 Purpose

Flappy Bird is an arcade-style game in which the player controls the bird Faby, which moves persistently to the right. The player is tasked with navigating Faby through pairs of pipes that have equally sized gaps placed at random heights.

1.2 System Overview

In my project i have used **Pygame.**

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

Pycharm

It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions.

2.Problem Statement

In our day to day life we need something to relax and have something which is simple and keep our mood relaxed, so this project is there to serve this purpose one can relax and play the game and forget our tension for some time.The game also increases the concentration of the user.

3.objective

It's flappy bird with more randomness. the goal is to get past the obstacles without getting detracted by everything always changing. If you touch the ground or an obstacle you lose. In this game, the objective is to pass through all the obstacles and not hit the ground or the obstacles.

4.Project Meathodology

The basic methodology behind my project is :

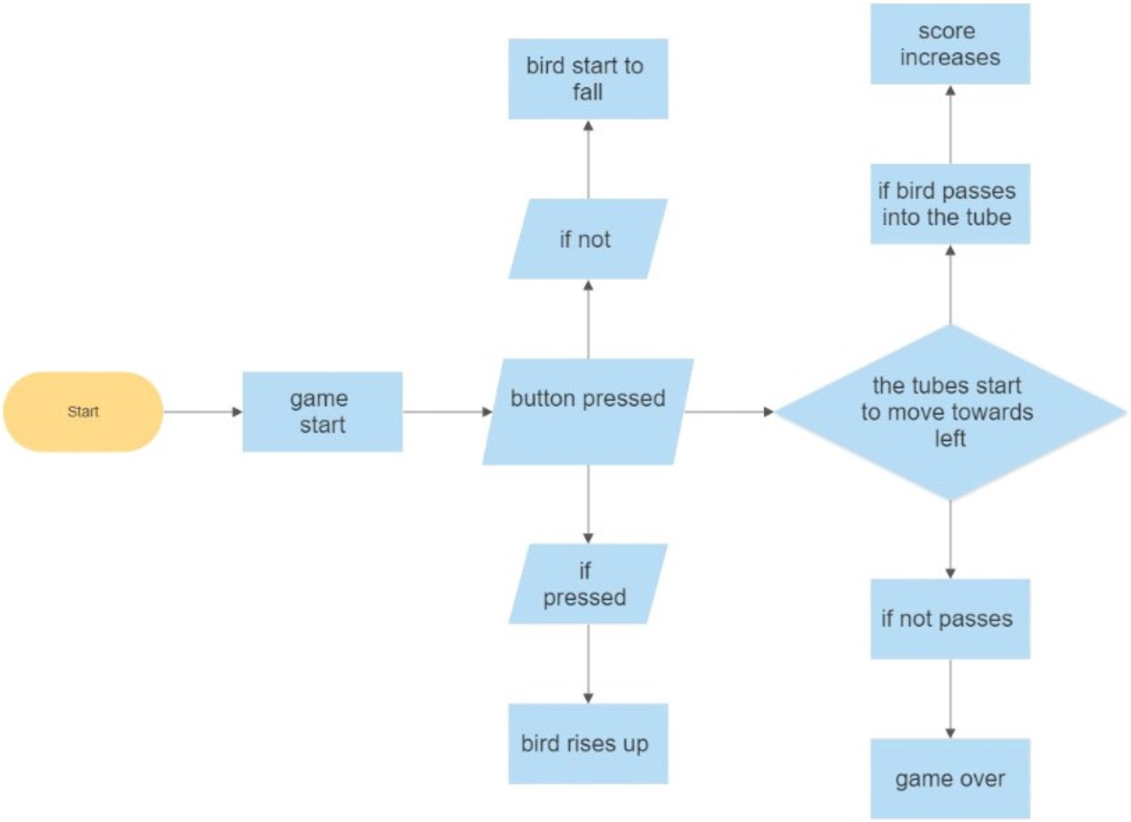


FIG1 : flow chart of working of project

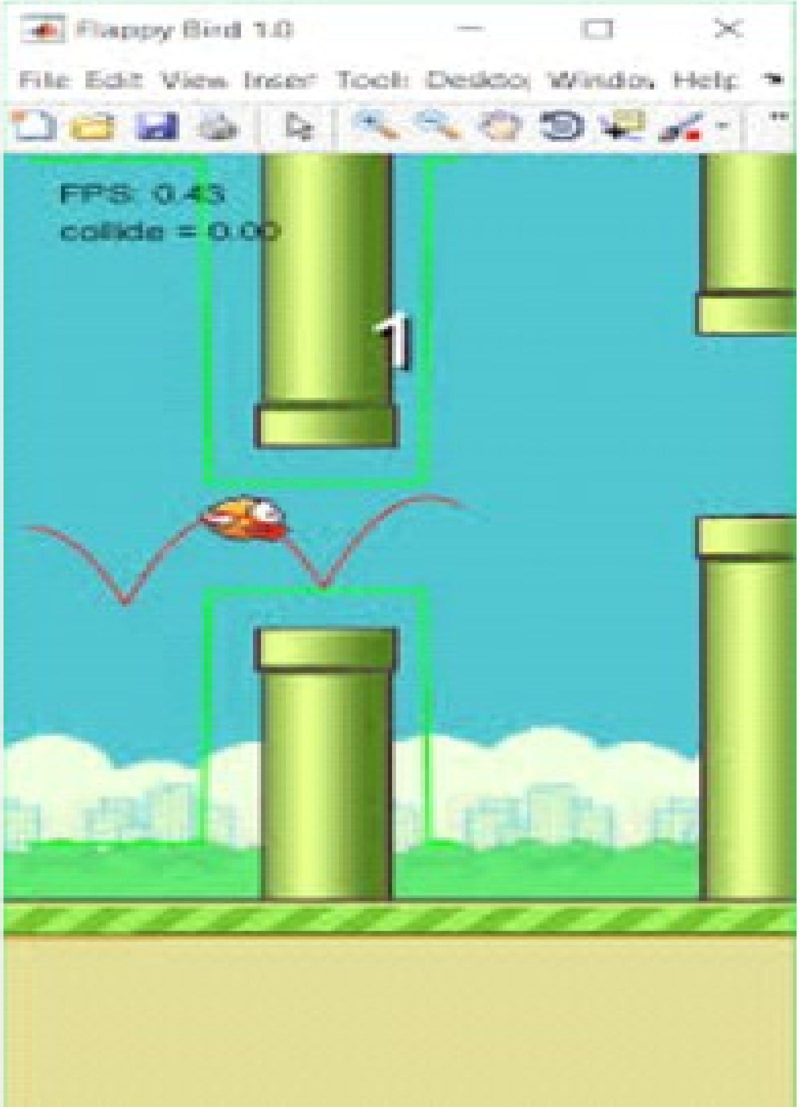


FIG 2: bird passing through obstacle

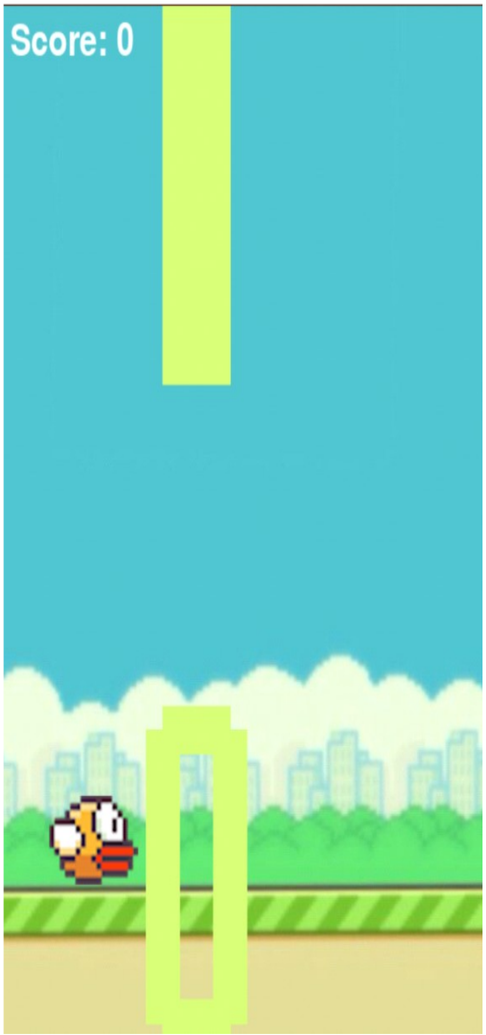
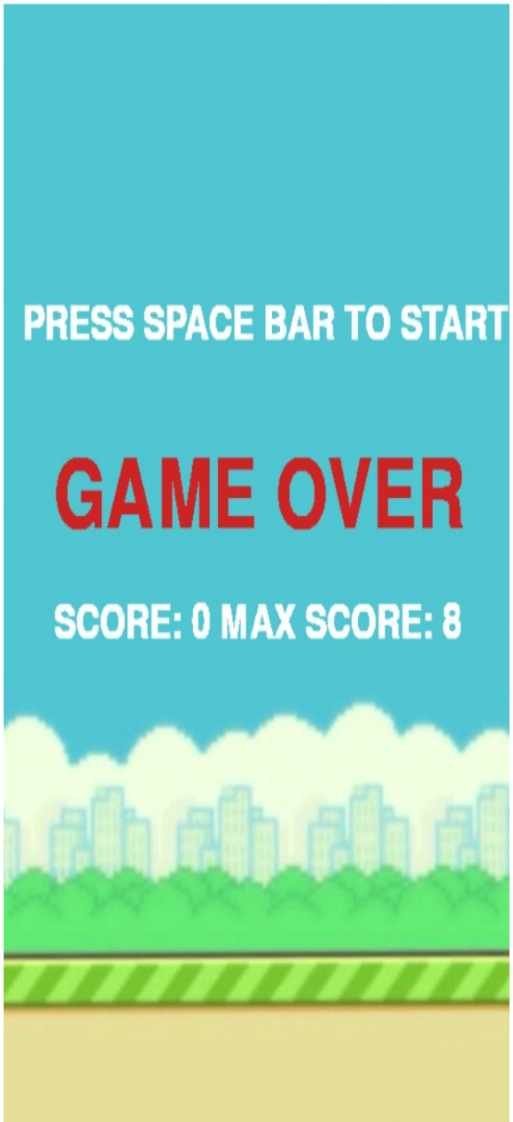


FIG 3: Collision made



PRESS SPACE BAR TO START

GAME OVER

SCORE: 0 MAX SCORE: 8

FIG 4 : Game over screen

5.Details of Project Work

We choose game for our first software project. Actually game is entertaining for anybody and in leisure time we can spend our time nicely by playing game. The flappy bird game implemented for only desktop.

5.1 Requirements

A requirement is a singular documented physical or functional need that a particular design, product or process aims to safety. It can be divided into functional requirements and non- functional requirements.

Functional – 2D animation, objectives selection, moving wall, collision detection, moving background etc.

Non-functional – We can keep the bird playing by pressing mouse and move it in the space of pipes.

2D Animation

Animation is a complex subject in game programming. Animation is rapid display of sequence of images which creates an illusion of movement. Java games are expected to run on multiple operating systems with different hardware specifications. Threads give the most accurate timing solutions.

Objective Selection

We create a bird object which is flying until any collision occurred and the bird is flying in the wall objectives which are begin from top and bottom of the screen.

Moving Wall

The wall moving on and it will come randomly in size and distances. The bird is flying in the middle of the wall.

Collision Detection

When the bird touch the anywhere of a wall it cause a collision. Collision detection is one of important task of the game. If the bird touch any wall (pipes) the game will end.

Moving Background

The picture used as background image is moving on analogously. We used two same image which are coming one after another regularly.

Score Counting

Score counting is the interesting for user. By the score the player knows his/her performance. If the bird cross a pipe without collision or not fall in ground his/her score increment one.

5.2 Working of project

The program run using a python compiler,then start screen appears which asks for pressing space bar then blitting process starts in which the background and bird are placed then the bird starts to fall and when space is pressed the bird verticle position changes a little upwards .
The pipe start to enter the screen and shift towards right with some gfap between them through which the bird passes.
if the bird passes through the pipe the game continues with the score increasing with every passageon the top corner.
if the bird fails to pass through the pipe then the game ends with the score on display.

6. Results and Discussion

The program runs successfully without any type of trace back, and I got the output which I expected.

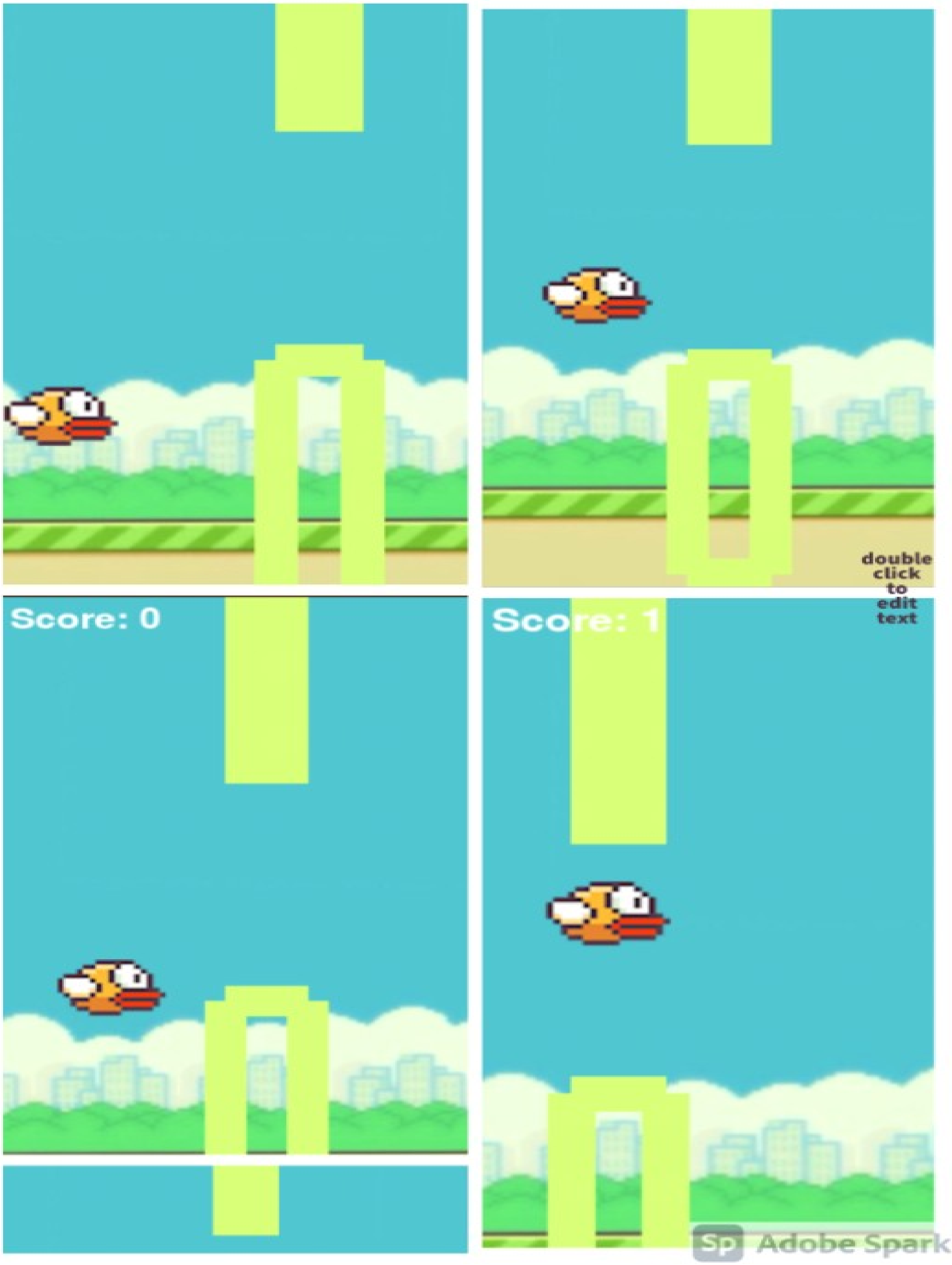
To avoid any type problem, please follow the rules

1. Install pygame on your system.

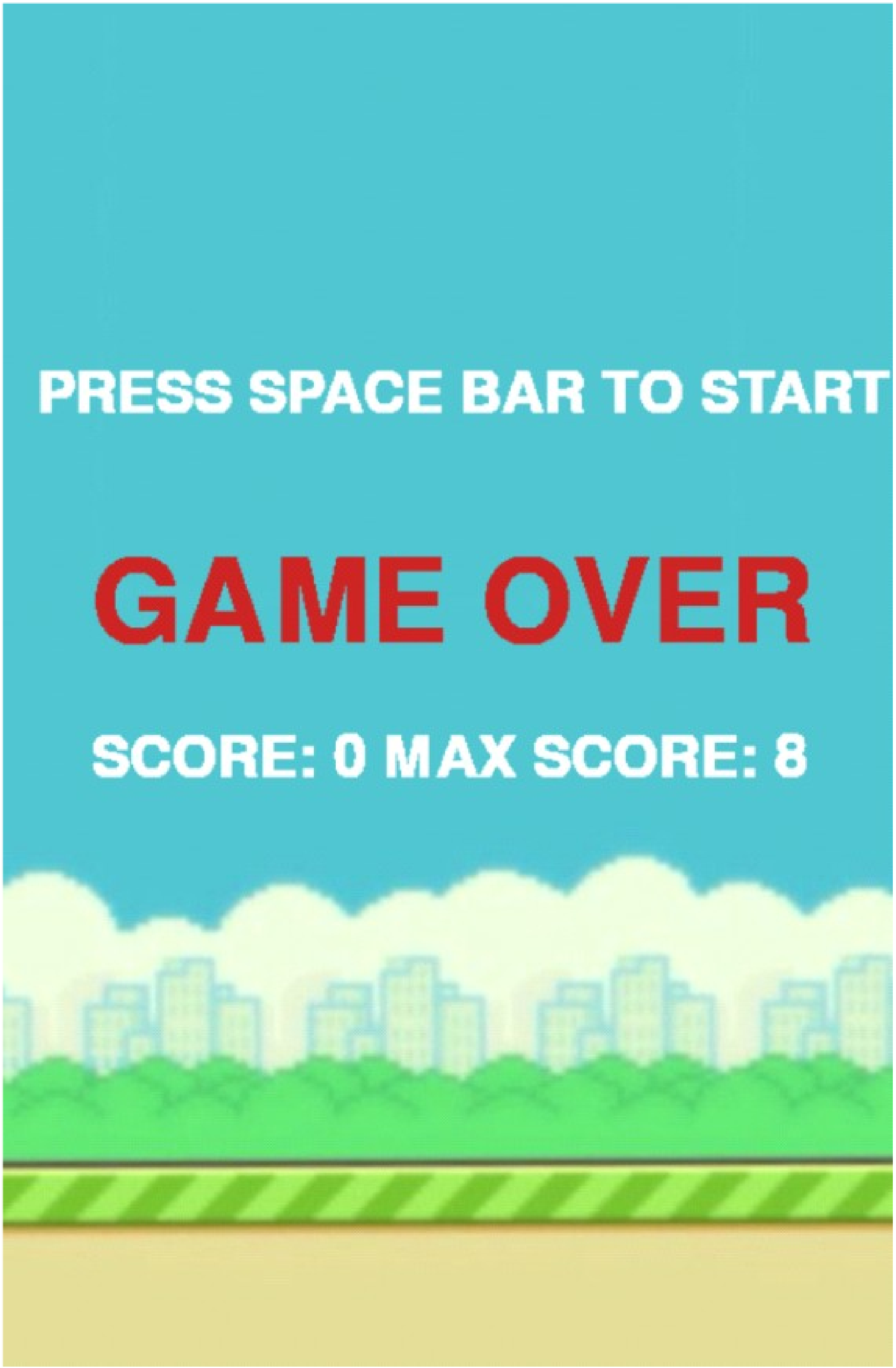
2. all the file including bird and background shoud be present in the system.

3. The python compiler should not be to old.

Here it is the resulted first output screen :-



GAME OVER SCREEN



PRESS SPACE BAR TO START

GAME OVER

SCORE: 0 MAX SCORE: 8

7. Conclusion and Future Scope

Future scope :-

This game can be made more interactable and interesting by adding sound track and time to time changing backgorund.

There can be a score board which can give a list of sevral people playing the game.

Conclusion:-

Overall, the performance of the game was met to standards. The game included all of the features that wished to be included: an introduction screen, a main menu, replay ability.

The design that included pixels and screen colors came out to work out the way as according to plan, as well as being able to introduce a “gravity” feature which allowed the bird to move up from the stylus command or fall down due to gravity. The game gave a good difficulty and stimulated what the original game was like.

References

1. https://www.youtube.com/watch?v=itB6VsP5UnA

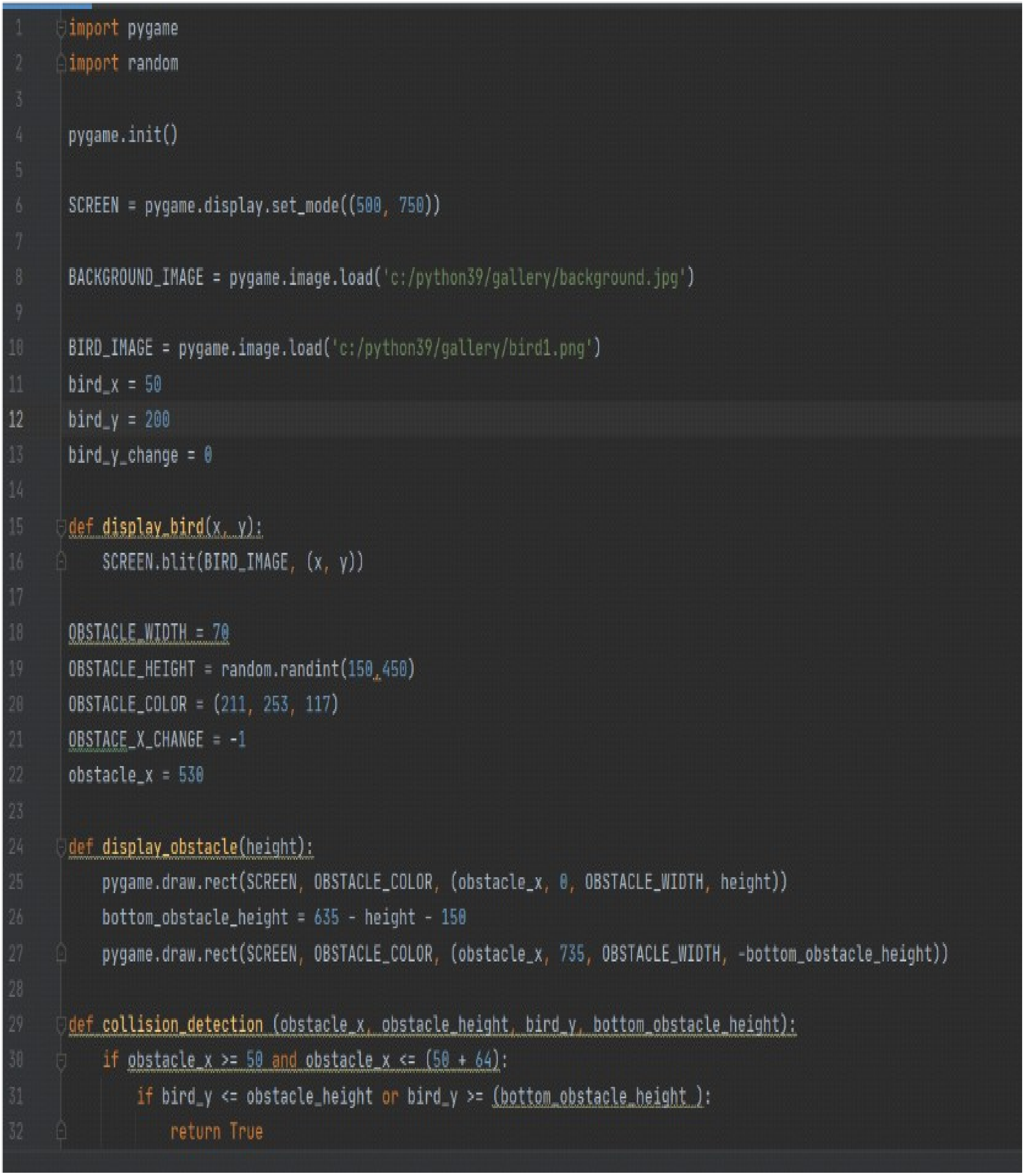
2. https://stackoverflow.com/

3. Lectures of the course (Getting started with python) from Coursera.

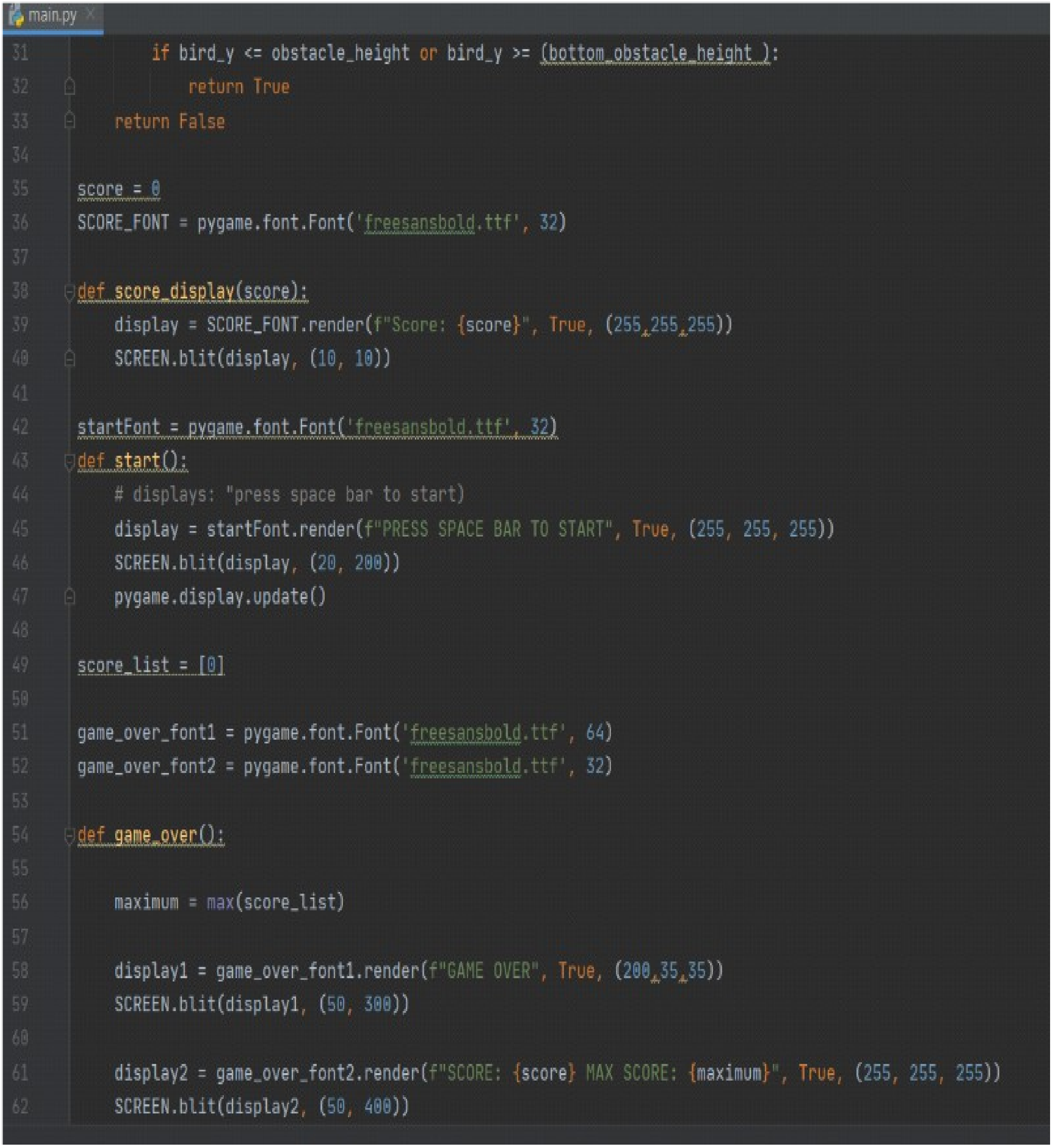
4. Adobe Photoshop

5. google images

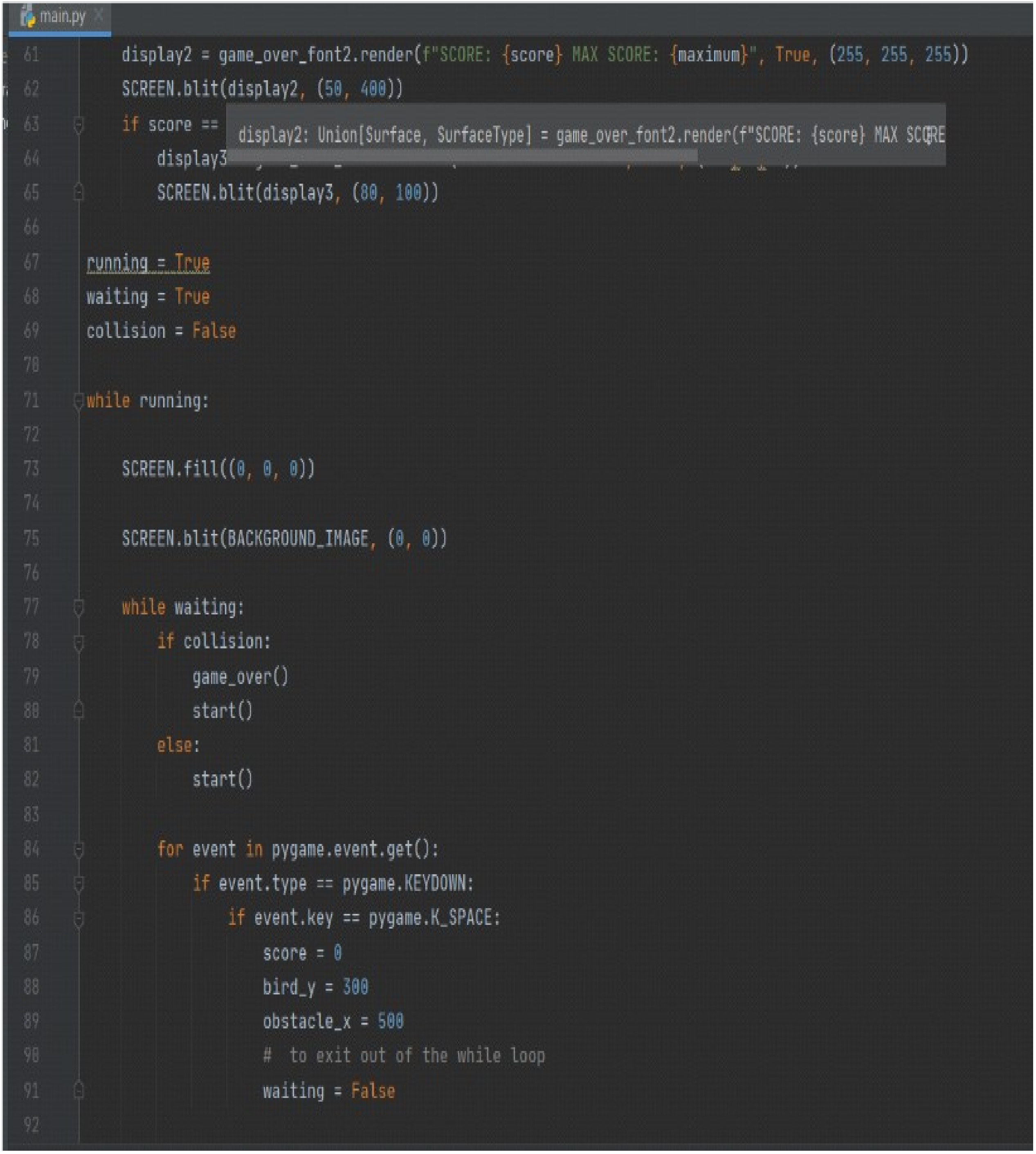
APPENDIX 1:



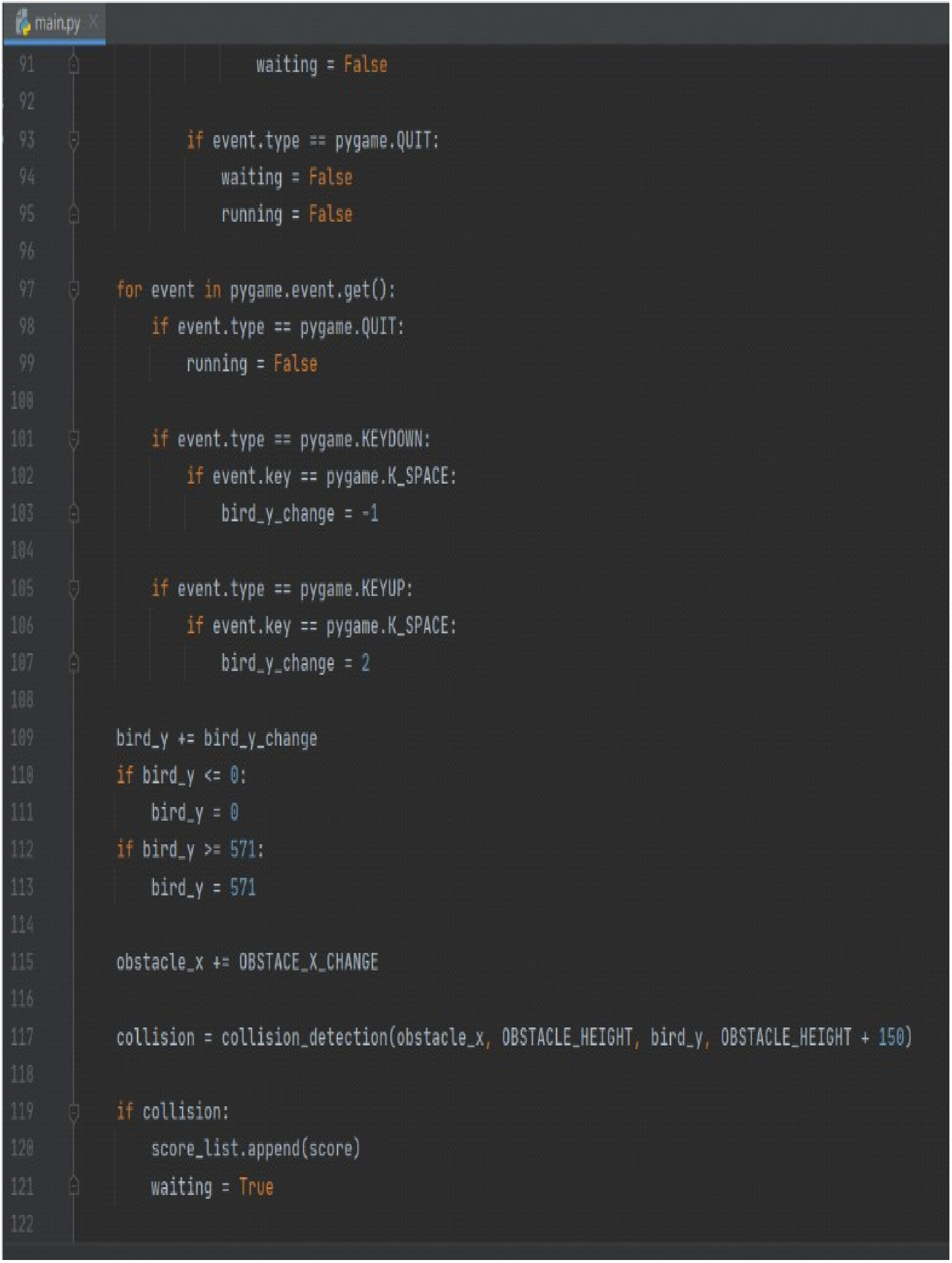
```python
import pygame
import random

pygame.init()

SCREEN = pygame.display.set_mode((500, 750))

BACKGROUND_IMAGE = pygame.image.load('c:/python39/gallery/background.jpg')

BIRD_IMAGE = pygame.image.load('c:/python39/gallery/bird1.png')
bird_x = 50
bird_y = 200
bird_y_change = 0

def display_bird(x, y):
    SCREEN.blit(BIRD_IMAGE, (x, y))

OBSTACLE_WIDTH = 70
OBSTACLE_HEIGHT = random.randint(150,450)
OBSTACLE_COLOR = (211, 253, 117)
OBSTACE_X_CHANGE = -1
obstacle_x = 530

def display_obstacle(height):
    pygame.draw.rect(SCREEN, OBSTACLE_COLOR, (obstacle_x, 0, OBSTACLE_WIDTH, height))
    bottom_obstacle_height = 635 - height - 150
    pygame.draw.rect(SCREEN, OBSTACLE_COLOR, (obstacle_x, 735, OBSTACLE_WIDTH, -bottom_obstacle_height))

def collision_detection (obstacle_x, obstacle_height, bird_y, bottom_obstacle_height):
    if obstacle_x >= 50 and obstacle_x <= (50 + 64):
        if bird_y <= obstacle_height or bird_y >= (bottom_obstacle_height ):
            return True
```



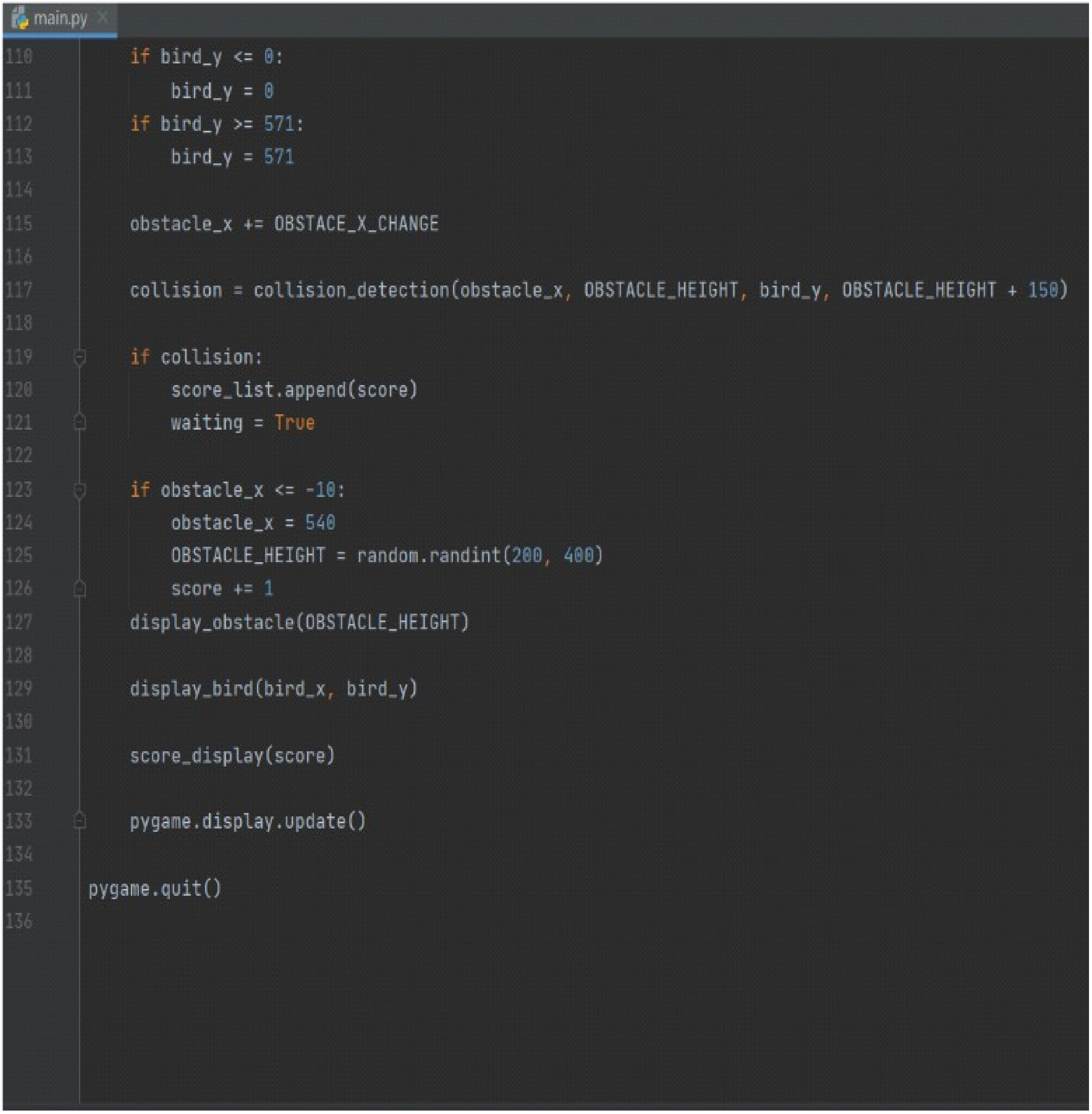
```python
        if bird_y <= obstacle_height or bird_y >= (bottom_obstacle_height ):
            return True
    return False

score = 0
SCORE_FONT = pygame.font.Font('freesansbold.ttf', 32)

def score_display(score):
    display = SCORE_FONT.render(f"Score: {score}", True, (255,255,255))
    SCREEN.blit(display, (10, 10))

startFont = pygame.font.Font('freesansbold.ttf', 32)
def start():
    # displays: "press space bar to start)
    display = startFont.render(f"PRESS SPACE BAR TO START", True, (255, 255, 255))
    SCREEN.blit(display, (20, 200))
    pygame.display.update()

score_list = [0]

game_over_font1 = pygame.font.Font('freesansbold.ttf', 64)
game_over_font2 = pygame.font.Font('freesansbold.ttf', 32)

def game_over():

    maximum = max(score_list)

    display1 = game_over_font1.render(f"GAME OVER", True, (200,35,35))
    SCREEN.blit(display1, (50, 300))

    display2 = game_over_font2.render(f"SCORE: {score} MAX SCORE: {maximum}", True, (255, 255, 255))
    SCREEN.blit(display2, (50, 400))
```


main.py
display2: Union[Surface, SurfaceType] = game_over_font2.render(f"SCORE: {score} MAX SCORE

```
    display2 = game_over_font2.render(f"SCORE: {score} MAX SCORE: {maximum}", True, (255, 255, 255))
    SCREEN.blit(display2, (50, 400))
    if score ==
        display3
        SCREEN.blit(display3, (80, 100))

running = True
waiting = True
collision = False

while running:

    SCREEN.fill((0, 0, 0))

    SCREEN.blit(BACKGROUND_IMAGE, (0, 0))

    while waiting:
        if collision:
            game_over()
            start()
        else:
            start()

        for event in pygame.event.get():
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_SPACE:
                    score = 0
                    bird_y = 300
                    obstacle_x = 500
                    #  to exit out of the while loop
                    waiting = False

```



```
                    waiting = False

            if event.type == pygame.QUIT:
                waiting = False
                running = False

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                bird_y_change = -1

        if event.type == pygame.KEYUP:
            if event.key == pygame.K_SPACE:
                bird_y_change = 2

    bird_y += bird_y_change
    if bird_y <= 0:
        bird_y = 0
    if bird_y >= 571:
        bird_y = 571

    obstacle_x += OBSTACE_X_CHANGE

    collision = collision_detection(obstacle_x, OBSTACLE_HEIGHT, bird_y, OBSTACLE_HEIGHT + 150)

    if collision:
        score_list.append(score)
        waiting = True

```



```python
    if bird_y <= 0:
        bird_y = 0
    if bird_y >= 571:
        bird_y = 571

    obstacle_x += OBSTACE_X_CHANGE

    collision = collision_detection(obstacle_x, OBSTACLE_HEIGHT, bird_y, OBSTACLE_HEIGHT + 150)

    if collision:
        score_list.append(score)
        waiting = True

    if obstacle_x <= -10:
        obstacle_x = 540
        OBSTACLE_HEIGHT = random.randint(200, 400)
        score += 1
    display_obstacle(OBSTACLE_HEIGHT)

    display_bird(bird_x, bird_y)

    score_display(score)

    pygame.display.update()

pygame.quit()

```