MANIPAL UNIVERSITY JAIPUR

PROJECT REPORT: 'CORONA RAIDERS'

Department of Computer Science and Engineering 2017-2021

School of Computing and Information Technology

Manipal University Jaipur

Jaipur, Rajasthan

SUBMITTED BY:

Rishabh Saxena (179301163)

Poojan Soni (179301143)

SUBMITTED TO:

Mr. Nitesh Pradhan

INTRODUCTION:

Computer graphics are visual representations of data displayed on a monitor made on a computer. Computer graphics is made up of number of pixels**.** Computer-generated imagery is used for movie making, video game and computer program development, scientific modeling, and design for catalogs and other commercial art.
Computer Graphics can be used by any language which:

- Runs on your target platform and
- Has OpenGL/any other rendering framework library bindings

Languages like Python, C, C++, Java, C#, etc., have Libraries or Modules which implement and help the developers to create and use system graphics.

OBJECTIVE:

In this project, we tried to simulate the current COVID-19 pandemic in a fun and useful way visualizing :

- If we stay Home, we remain safe
- If we go outside, we can stay safe using precautions such as mask, sanitizers
- If we go out and remain inactive to protect ourselves, we could get infected

Motto is to spread awareness using game play, you score if you fight corona, you loose (game ends) if you get infected.

LANGUAGE AND LIBRARIES:

PYTHON3

Libraries used:

- **Turtle**: “Turtle” is a Python feature like a drawing board, which lets us command a turtle to draw all over it. We can use functions to draw around.
- **OS:** used to run some specific commands from the program directly by the operating system.
- **Math:** used for distance calculations for checking collision among the turtle objects used (like player-corona, corona-sanitizer)
- **Platform:** used to detect type of OS
- **Random**
- **Time**

FUNCTIONS:

1. **Creating the screen**:
   ⇨ screen = turtle.Screen()
   turtle returns graphics windows as a playground for the drawing turtles.

2. **Creating Main player and weapon**:
   Using Turtle we create a main player and the sanitizer bullet which will have the image we register.
   ⇨ player = turtle.Turtle()
   ⇨ player.shape("classic")
   ⇨ bullet = turtle.Turtle()
   ⇨ bullet.shape("triangle")

3. **Keys used:**
   a => move left
   d => move right
   space => fireBullet
   enter=> exit game in the end

4. **moveLeft():**
   - ⇨ moves the player to the left
   - ⇨ checks the boundary
   - ⇨ updates the string like (stay safe in red/green)

5. **moveRight():**
   - ⇨ moves the player to the right
   - ⇨ checks the boundary
   - ⇨ updates the string like (stay safe in red/green)

6. **fireBullet():**
   This function controls the unity of the sanitizer drop (as bullet) and motivates to remove the corona using audio clips included.

7. **playSound(soundFile):**
   plays audio clips used in different cases of the game interval with checking the operating system in which it is used.

8. **isCollided(turtle1, turtle2):**
   This function checks if the two turtle objects have collided using the basic distance formula for the x and y coordinated.
   It return True if collides, False if not.

9. **Creating ENEMY (corona):**
   For generating multiple enemies to strike the user, we created a class called enemy which takes xspeed and yspeed as arguments and has random attributes x and y coordinates to start.

➔ Move():

This function inside Enemy class is called to move or update the position of corona turtle. It also checks the boundaries of the game as well as the home, if it collides it bounces back (changing the direction of the speed)

10. Score:

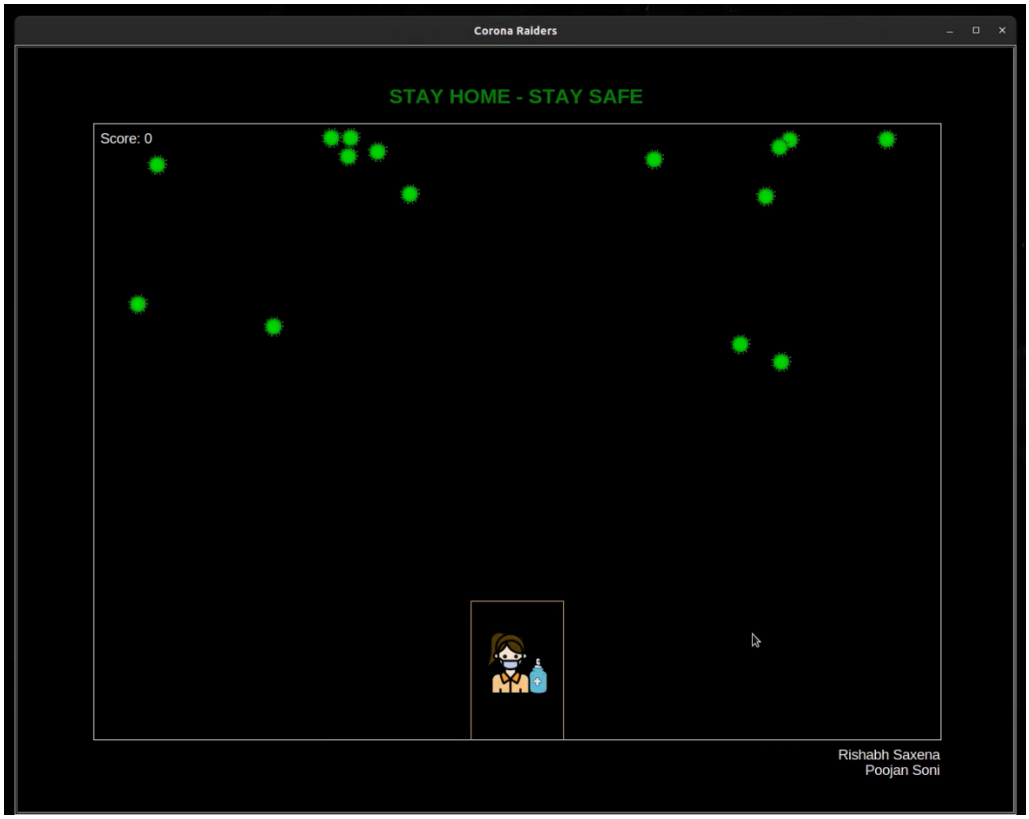
User gets score if his/her bullet successfully kills corona. For each kill the score is 10

11. Player infected:

If there is collision between the player and corona then the game automatically stops displaying the end screen and final score with image of infected player.

OUTPUT SCREENS:

i> Initial screen with player inside house



ii> **In the middle of the game using sanitizer drop as bullet to kill/fight corona**

iii> **Ending screen / when player gets infected:**

COMPLEXITY:

Although this game is fun to play but the graphics are used multiple times dynamically which can slow or can produce lag in the game.

Complexity of the main code running is $O(n^2)$.

FUTURE WORK AND CONCLUSION:

For future use and implementations, we need to improve the performance. Also, we need to add:

- More animations
- Graphic improvements

To conclude, using technology in more expressive way to influence good to the society is the best way for an Engineer to give back to the society.