Candidate No: 22113381

**Game Overview**

The primary objective of the game is for the player to start from the initial position (of a turtle) and progress upward trying to avoid cars on his path at a specified speed(8unit). There are various cars of different colors moving from the right side of the game page to left side. However, these cars are the obstacles that the player must avoid in order to exit the field successfully and proceed to the next level. This section elaborates on the description of the game play and player controls.

**Game Description**

In the game, the player is played as a turtle that has to find its exit through a field of moving cars. The field is designed as an empty space for cars to move. To escape from the field, the player must avoid contacts with cars of any color by carefully progressing towards the upper part of the field. Throughout the field, the player can find cars of different colors.

The speed of the game increases once the player successfully exit the field and progress to next level.

The player(character)

A turtle object generated from turtle module serves player`s image.

**User Manual**

The player controls the game play using just the upper key on the keyboard. The back key on the keyboard is not enabled so the player cannot move back to correct his movement mistakes. This is done to make the game a bit challenging and make player more careful when playing. Direction Keys:

 Key ‘^’: press the upper arrow key on your keyboard to move turtle from its position.

**Game Design**

In creating the game, many different programming techniques are used for drawing and viewing objects and adding special effects. This section describes the methods of how the game is implemented.

3.1 3-D Viewing and Objects

The visualization effect (GUI) of the game is done with a module called turtle. This basically used for the graphical interface aspect of the game, the on screen method showed us the display and other turtle styling came in handy.

**Self-Designed modules**

Five different modules were designed to carry out different functions with the use of object oriented programming (OOP). This is done to make the code neat and easy to read. The modules are as follow

* Lane.gif
* Car\_manager.py
* Player.py
* Scoreboard.py
* **Main**.py

Lane.gif: This is just a file under the “Image” folder, it is a picture used as the game field.

**Car\_manager:** This python file contains a neatly packaged functions that generates cars of different colors and generate movement speed and direction

def create\_car(self):  
 '''  
   
 Create cars that are 20px high by 40px wide that are randomly generated along the Y-axis and move to the left edge of the screen.   
 This method is going to create cars somewhere along the Y axis with a given dimension.  
 '''  
 random\_chance = random.randint(1, 6)  
 if random\_chance == 1:  
 new\_car = Turtle("square")   
 new\_car.shapesize(stretch\_wid=1, stretch\_len=2)   
 new\_car.penup()  
 new\_car.color(random.choice(COLORS))  
 random\_y = random.randint(-250, 250).   
 new\_car.goto(300, random\_y).  
 self.all\_cars.append(new\_car)

The method above positions (right side of the field) again and move towards the left side of the field generates a square image of cars and randomly choose different colors between the defined colors. The cars re generate from the starting point(right side of the field) to the left side(finishing point).

**Player.py**: This module is designed to generate a player object(turtle) as assign some attributes to it. Some of the attributes include shape(turtle), movement position, starting and finishing point of the player(turtle), the movement distance

def is\_at\_finish\_line(self):  
 '''  
 This method is going to return True if the turtle is at the finish line and False while it is not  
 '''  
 if self.ycor() > FINISH\_LINE\_Y:  
 return True  
  
 else:  
 return False

The method above is inside the player.py module and its designed to validate the completion of each game level. If player makes it to the end of the field, this function is activated and player automatically be returned to the starting line for the next level.

**Scoreboard.py**: There are four different methods in this python file. The methods are designed to store players level and increase it. These methods show when player loses and write Game over. It also let the player know the stage the turtle is when he loses.

def increase\_level(self):  
 '''  
 This method increase the level of the game on the scoreboard  
 '''  
 self.level += 1  
 self.update\_scoreboard()

The method above is set up to increase the level of the game. The player can continue playing as long as he can keep up with the increasing speed of the game.

Main.py: This python file is the main file where all these modules are invited to perform the functions we designed them for. We combine all these modules together here and when we hit the run key we are able to visualize everything we have designed and the game can finally be played. The game will ask for the player name before it can start working and print a welcome message.

**Conclusion**

In the game The CROSSING GAME, we have implemented a game environment that includes viewing and objects within a designed field space , a player in turtle image, To enhance the game environment we used turtle. This is our graphical user interface. The key feature of our randomly generated car colors makes our game original and appealing to viewing. When player collide with this car obstacle, game is over. This shows the purpose of the game and make it tedious.