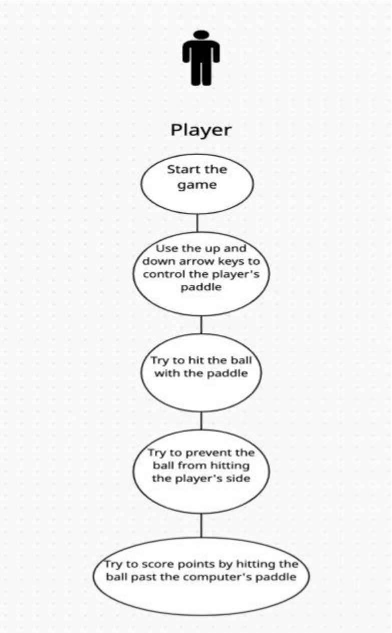
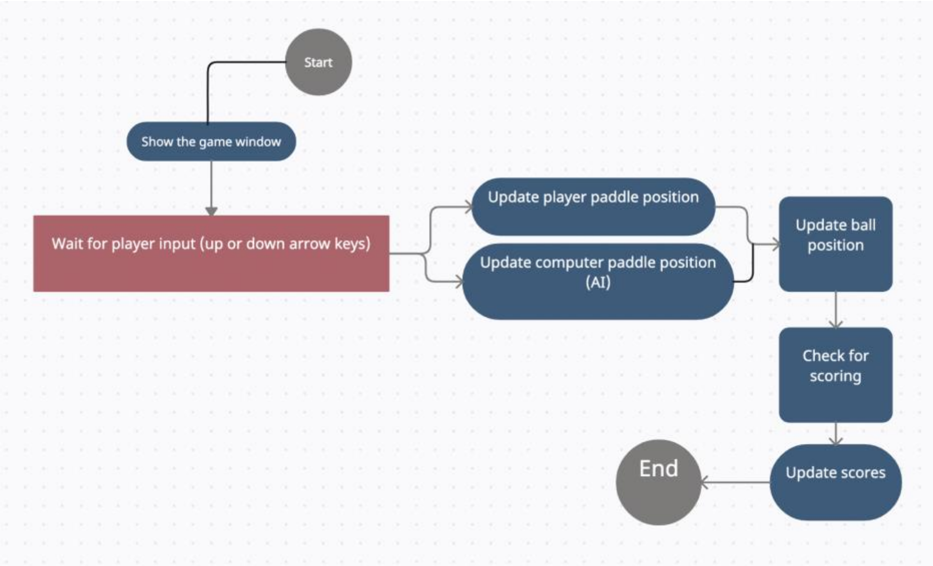
<u>Prigorian Pahsya L1CC (2602186011)</u>

1. This script is a basic implementation of the classic game Pong using the Pygame library. It sets up a game window, defines game variables such as scores, and creates classes for the player and computer paddles, as well as the ball. It includes methods for moving and drawing the elements on the screen, handling collision detection, and scoring. It allows user to control the paddle with the up and down arrow keys, while the computer's paddle is controlled by the computer. The game is won when one of the player or computer scores reaches to 11. The game also includes a basic AI that moves the computer's paddle to try and hit the ball. The script allows the player to play against the computer.

2.


Player
Start the game
Use the up and down arrow keys to control the player's paddle
Try to hit the ball with the paddle
Try to prevent the ball from hitting the player's side
Try to score points by hitting the ball past the computer's paddle

3.


Start
Show the game window
Wait for player input (up or down arrow keys)
Update player paddle position
Update computer paddle position (AI)
Update ball position
Check for scoring
Update scores
End

```
4. +----------------+   1  +-------------+
   |  paddle        |------>|  ball       |
+----------------+      +-------------+
| - x            |
| - y            |
| - rect         |
| - speed        |
| - ai_speed     |
| + move()       |
| + draw()       |
| + ai()         |
+----------------+


+----------------+
|  ball          |
+----------------+
| - x            |
| - y            |
| - rect         |
| - ball_rad     |
| - speed_x      |
| - speed_y      |
| - winner       |
| + move()       |
| + draw()       |
| + reset()      |
+----------------+

+----------------+
|  draw_board()  |
+----------------+

+----------------+
| draw_text()    |
+----------------+

+----------------+
| font           |
+----------------+

+----------------+
| fpsClock       |
+----------------+

+----------------+
| screen         |
+----------------+

+----------------+
| cpu_score      |
+----------------+

+----------------+
| player_score   |
+----------------+

+----------------+
| fps            |
+----------------+

+----------------+
| live_ball      |
+----------------+

+----------------+
| winner         |
+----------------+

+----------------+
| speed_increase |
+----------------+

+----------------+
| margin         |
+----------------+
+----------------+
| bg             |
+----------------+

+----------------+
| white          |
+----------------+
```

5.1) pygame: The main module for the Pygame library, which provides functionality for creating games and other multimedia applications.

2) pygame.locals: A submodule of Pygame that provides constants such as key codes and mouse button values.

3) pygame.init(): Initializes all the required pygame modules.

4) pygame.time: A module that provides timing functions for games and other multimedia applications.

5) pygame.display: A module that provides functions for creating and managing windows and other display elements.

6) pygame.font: A module that provides functions for creating and manipulating text in games and other multimedia applications.

7) pygame.draw: A module that provides functions for drawing graphics on the screen.

8) pygame.key: A module that provides functions for detecting and responding to keyboard input.

9) pygame.event: A module that provides functions for handling events such as key presses and mouse clicks.

10) pygame.Rect: A module that provides a class representing rectangles and also used as a base class for many other pygame classes.

11) pygame.K_UP: A constant that represents the up arrow key on the keyboard.

12) pygame.K_DOWN: A constant that represents the down arrow key on the keyboard.

6. Ball Movement Algorithm: The script uses the move() method of the ball class to update the position of the ball on the screen based on its current speed and direction. This algorithm takes into account the collision with the top and bottom margins of the screen and with the paddles.

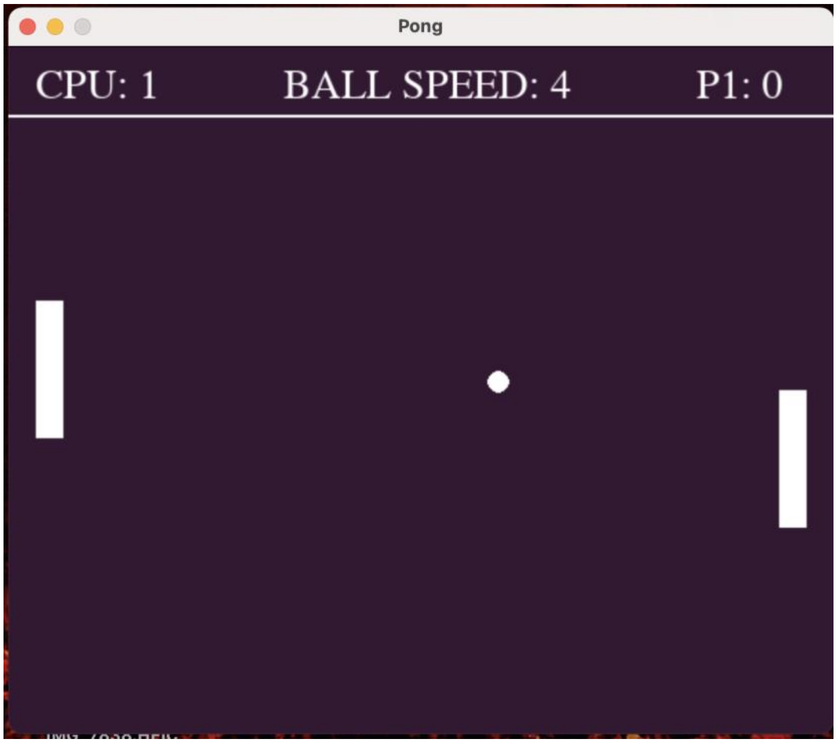
Paddle Movement Algorithm: The script uses the move() method of the paddle class to update the position of the paddle based on user input. The algorithm checks the state of the up and down arrow keys and updates the position of the paddle accordingly.

Paddle AI Algorithm: The script uses the ai() method of the paddle class to move the CPU paddle automatically. The algorithm checks the position of the ball and updates the position of the CPU paddle to track the ball.

Collision Detection Algorithm: The script uses the colliderect() method of the rect object to detect collisions between the ball and the paddles. This algorithm is used in the move() method of the ball class to determine if the ball has hit a paddle and should change direction.

Score update Algorithm: The script uses the winner variable to update the score of the CPU or the player when the ball goes out of bounds. The algorithm checks the winner variable and increases the score of the corresponding player.

7.



8. I’ve learned how actually hard it is to create a game even a simple and “easy” one. I have gained knowledge by creating this game and I still need to work on my creativity and idea because that is the thing that I lack.