CSE 1321L: Programming and Problem Solving I Lab

Assignment 4 – 100 points

PyGame

What students will learn:

1) Incorporating prior programming knowledge to game development

2) Developing a larger scale, graphical program

3) Problem solving

Overview: We will be making a basic video game using PyGame. While the result of this project should be fun, the real goal is to give you the opportunity to apply everything you've learned about programming up to this point. There are many fields of software development, but the fundamentals you've learned so far can be applied to all of them.

Assignment 4A: Fruit Catcher Game

Create a simple "catch the falling objects" game using Pygame where the player controls a basket to catch falling fruits. The game should allow the player to score points by successfully catching the falling objects.

Requirements:

1. The game window must be 800 pixels wide and 600 pixels tall.
2. Player Control:
    - The player will control a rectangular basket (white) at the bottom of the screen using the left and right arrow keys.
    - The basket should have a width of 100 pixels and a height of 20 pixels.
3. Falling Object:
    - A square fruit (red) will fall from the top of the screen.
    - The fruit should be a 20x20 pixels square.
    - The fruit will spawn at a random horizontal position at the top of the screen and will fall at a constant speed of 3 pixels per frame.

4. Collision Detection:
    - If the player's basket collides with the fruit (using the colliderect() method), the following should occur:
        - The player's score should increase by 1 point.
        - The fruit should reset to a new random horizontal position at the top of the screen.
        - The updated score should be printed to the console.
5. Game Continuation:
    - The game should continue running indefinitely, allowing the player to catch as many fruits as possible.
    - The score will only be displayed in the console (not on the game window).
6. The game should exit gracefully when the user closes the game window.

Hints:

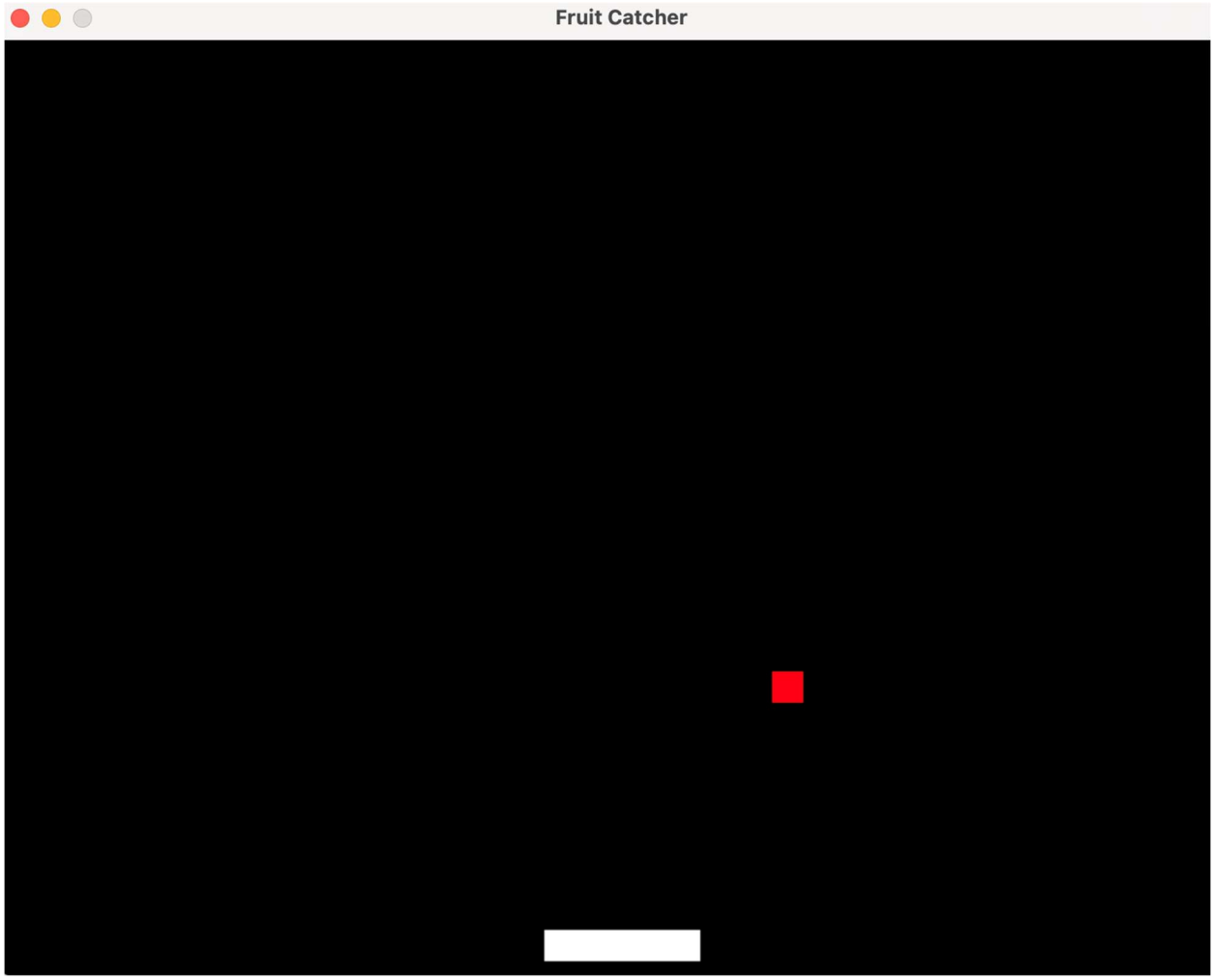
1. Drawing and Movement: Use pygame.draw.rect() to draw the player and the fruit on the screen, and update the player's position using the arrow keys (K_LEFT and K_RIGHT).
2. Handling Randomness: The fruit should spawn in a random horizontal position using the random.randint()function. For example:

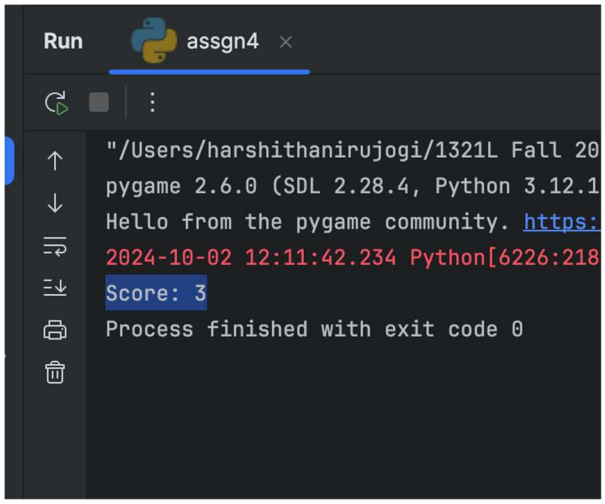
    ```
    import random

    fruit.x = random.randint(0, WIDTH - fruit_size)
    ```

    This will select a random x coordinate for the fruit, ensuring that it fits within the screen width.
3. Make sure to handle the game loop properly to update the display and manage events. Use the pygame.time.Clock() object to control the game speed.

SAMPLE OUTPUT:





Submission:

1. You will submit all required files to run your game.

2. File names must be correct.

3. Upload all files (simultaneously) to the assignment submission folder in Gradescope.