Title: Simple Soccer Game

Description:

This program implements a basic soccer game using the Pygame library. It displays a soccer field background with a soccer ball moving horizontally across the field.

Import pygame

Dependencies:

- Pygame library

Entities:

- SoccerField background: An image representing the soccer field background.

- SoccerBall: An image representing the soccer ball.

- Variables:

- SoccerBall\_x: The x-coordinate of the soccer ball's position.

- SoccerBall\_y: The y-coordinate of the soccer ball's position.

- clock: Pygame Clock object to control the frame rate.

- keepGoing: Boolean variable to control the main loop.

Functions:

1. main():

Description:

- Initializes Pygame.

- Sets up the display window with a specified size and caption.

- Loads and scales the soccer field background image.

- Loads and scales the soccer ball image.

- Initializes variables for the soccer ball's position.

- Enters the main game loop:

- Controls the frame rate using the clock object.

- Handles events such as quitting the game.

- Updates the position of the soccer ball.

- Checks boundaries to ensure the soccer ball stays within the screen.

- Refreshes the screen by blitting the background and soccer ball images.

- Quits Pygame when the main loop exits.

Usage:

- Run the program to start the soccer game.

- The soccer ball moves horizontally across the field and wraps around the screen when it reaches the edge.

Initialization:

Pygame is initialized using pygame.init().

Display Setup:

A Pygame window is created using pygame.display.set\_mode((640, 480)), with a size of 640x480 pixels.

The window caption is set to "Soccer!" using pygame.display.set\_caption().

Entities Setup:

The background image of the soccer field is loaded from the file "SoccerField.jpeg" using pygame.image.load().

The background image is converted to alpha format using convert\_alpha() to enable transparency.

The background image is scaled to fit the size of the screen using pygame.transform.scale().

Soccer Ball Setup:

The image of the soccer ball is loaded from the file "SoccerBall.png".

The soccer ball image is converted to alpha format using convert\_alpha().

The soccer ball image is scaled to a size of 100x100 pixels using pygame.transform.scale().

Variables Setup:

SoccerBall\_x is initialized to 0, representing the initial x-coordinate of the soccer ball's position.

SoccerBall\_y is initialized to 200, representing the initial y-coordinate of the soccer ball's position.

Main Loop:

The main game loop begins with a while loop that continues while keepGoing is True.

Inside the loop:

The frame rate is controlled to 30 frames per second using clock.tick(30).

Events such as quitting the game (closing the window) are handled using a for loop iterating over pygame.event.get().

The x-coordinate of the soccer ball (SoccerBall\_x) is incremented by 5 units, causing the soccer ball to move horizontally across the screen.

A boundary check is performed to ensure that the soccer ball stays within the screen width. If it exceeds the screen width, its x-coordinate is reset to 0.

The screen is refreshed by blitting the background image and the soccer ball image onto the screen using screen.blit().

Changes are updated on the display using pygame.display.flip().

Termination:

Pygame is quit using pygame.quit() when the main loop exits.

This step-by-step breakdown illustrates how each part of your code contributes to creating a simple soccer game with a moving soccer ball.