Deggendorf Institute of Technology

Faculty European Campus

Rottal-Inn

Bachelor Industrial

Engineering

Informatics for Engineering II

Summer Semester 2023

Assignment (2) report: Kicking a ball into a net

Han Lin Aung

Matriculation Number – 22201576

Semester - 2

Designing and implementing a computer game using the "graphics" module

"Kicking a ball into a net"

Introduction:

In this python program, I aim to design and implement a small computer game in object-oriented style using the "graphics" module used in the lecture sessions. The Player aims to score as many goals as possible within a limited number of lives. The game features a game window, a ball and a goal post represented by a customized image. Once all lives are used, a "Game Over!" message appears, displaying the final goal count. The program demonstrates object-oriented game design principles and provides an engaging and interactive gaming experience.

Code Explanation:

1. Creating the game window

First, the "Game" class is defined using the 'class' keyword. It contains a function called **"__init__"** that takes two parameters, "width" and "height," representing the dimensions of the game window. The game window is created using the **'GraphWin'** function, given a title 'My Game' with dimensions previously represented and is assigned to the **'self.win'** attribute. Using the 'Image' class from the graphics module, the attribute **'self.field'** is created with x and y positions and downloaded and edited image file (png file). For the movement direction of the net, **'self.direction'** attribute with the value "1" is initialized which means the net will initially move towards the right side. The **"self.ball_move"** attribute is set to "0" to control the vertical movement of the ball. Additionally, the **"self.goal"** , **"self.life"** and **"self.highest_score"** attributes are also assigned. These attributes are used to keep track of the player's score and remaining lives.

2. Creating and displaying the score text and highest score text

Using the 'Text' class from the graphics module, first argument 'Point' is specified with the position **(self.width/2, self.height/2 +100)** and the second argument which is actual text content **" Your Score:" + str(self.goal)"** are set. Various formatting and styling attributes such as Text color, Size, Font and Style are also created. After that, the score text is drawn on the game window using the draw () method. Creating and displaying the highest score text will follow the same instructions as score text.

3. Creating life text and "Game Over!" and displaying them on game window

Like creating the score text, similar code lines can be developed for life text and 'Game Over!' text. However, the "Game Over!" text will only be displayed after all the limited number of lives have been used by the player.

4. Creating and displaying a circle (ball) and net

The ‘create_circle’ method is responsible for creating and displaying a circle (ball) in the game window. It takes two parameters, x1 and y1, which represent the initial x and y coordinates of the circle. Inside the method, the initial coordinates are stored in ‘self.original_x’ and ‘self.original_y’ attributes for later reference. Then, an Image object is created with the given coordinates and an image file representing the ball, which is 'football_1-removebg-preview.png' in this case. Finally, the ball object is drawn on the game window ‘self.win’ using the draw method. A net can also be created as an image, identifying the position as "self.width / 2, 60 ".

5. Moving the net back and forth

First, it checks if the right edge of the net (obtained by adding the x-coordinate of the anchor point with half of the net's width) is equal to the width of the game window. If it is, it means the net has reached the right boundary, and in that case, the direction attribute is set to -1. This change in direction will make the net move towards the left.
Next, it checks if the left edge of the net (obtained by subtracting half of the net's width from the x-coordinate of the anchor point) is equal to 0. If it is, it means the net has reached the left boundary, and in that case, the direction attribute is set to 1. This change in direction will make the net move towards the right.
Finally, the net is moved horizontally using the move method of the net object. The amount of movement of the net (speed) is determined by multiplying 0.1 (user can adjust) with the direction attribute, which adjusts the net's position either towards the left or the right. The net does not move vertically, so the second parameter of the move method is set to 0.

6. Moving a ball using “Up “key in keyboard

First, it calls the **“checkKey”** method of the “win” object to check for any key that has been pressed by the user. The returned value is stored in the temp variable. Next, it checks if the value of temp is equal to "Up", which indicates that the "Up" key has been pressed by the user. If the condition is true, it means the user wants the ball to move in the upward direction. In that case, the **“ball_move”** attribute is set to -0.2, which means the ball will move upward by a certain amount on the y-axis. It is important to note that in most computer graphics systems, including the Python graphics module, the y-coordinate is measured from top of the window to downwards which is different from Cartesian coordinate system used in mathematics. That is why, I set the ball’s direction as negative sign to move upward along the y-coordinate.

7. Checking if the ball enters the net

To check whether the ball enters the net or not, two conditions need to be checked. First, y-coordinate of the ball and net is checked. If the y-coordinate of the ball's anchor point is equal to the y-coordinate of the net's anchor point, this ensures that the ball is aligned vertically with the net. Next, x- coordinate of the both objects should be checked. If the x-coordinate of the ball's anchor point falls within the x-range of the net's corners (the ball is between the two poles of the net), this ensures that the ball is within the horizontal boundaries of the net. If both conditions are satisfied, it means that the ball has entered the net. In this case, the goal count is increased by 1, the ball's movement is reset to 0, and the score text is updated. Furthermore, the ball is undrawn from the window and its original position is recreated using the **“create_circle”** method, resetting its position for the next shot.

8. Calculating the highest score and explaining how this works

In addition to goal score, I have added a condition to check if the current goal count **“self.goal”** is greater than or equal to the **“highest_score”.** If it is, the highest score will be updated with the current goal count and update the “highest_score_text” to display the new highest score.This allows us to keep track of the highest score achieved during the game session and display it to the player.

Therefore, the highest score in the game represents the maximum number of goals a player can achieve within their five lives. When a player scores goals during their attempts, the game keeps track of the highest score. For instance, if a player scores 5 goals in his first life and then fails to score in subsequent attempts 2,3,4 or 5, the highest score remains at **5**. However, if the player exceeds this score in any of their remaining lives, for example, scoring **7** goals in 5th round, the highest score will be updated to “7” which reflects the new achievement. This ensures that the highest score represents the player's best performance across all five lives. So, each life matters for a player and even though he could not score many goals in his first life, he can have a chance to score more goals in his subsequent lives.

9. Checking the life count and Game Over!

If the ball's y-coordinate is less than or equal to 0, it has reached or passed the top of the window, which means that the ball has not entered the net. If this happens, first the life count will be decreased by 1, the vertical movement of the ball will be stopped, the life text on the window will be updated, the ball from the window will be removed and then the ball's position will be reset to the original coordinates. If the life count is equal to 0, **“Game Over!”** text will be appeared on the game window.

10. Closing the game window

If the key "c" is indeed pressed, it proceeds to close the game window by calling the **“close ()”** method on the **“self.win”** object. This allows the user to manually close the game window by pressing the specified key.

References

I downloaded the 3 photos (png files) and removed the background and reedited them for this Python programm. All files are also attached together with a.py file when submitting to ilearn.