Automata-Project

- By **Muhammad Umar Anzar**
- Seat No. **B19102104**
- University Of Karachi
- UBIT department of computer science
- Semester 5
- Subject Computer Theory (Automata)

Finite ball

A basic two-player 2D soccer game created in Python with the Pygame framework.

Purpose

The goal was to include a finite state machine in a project.

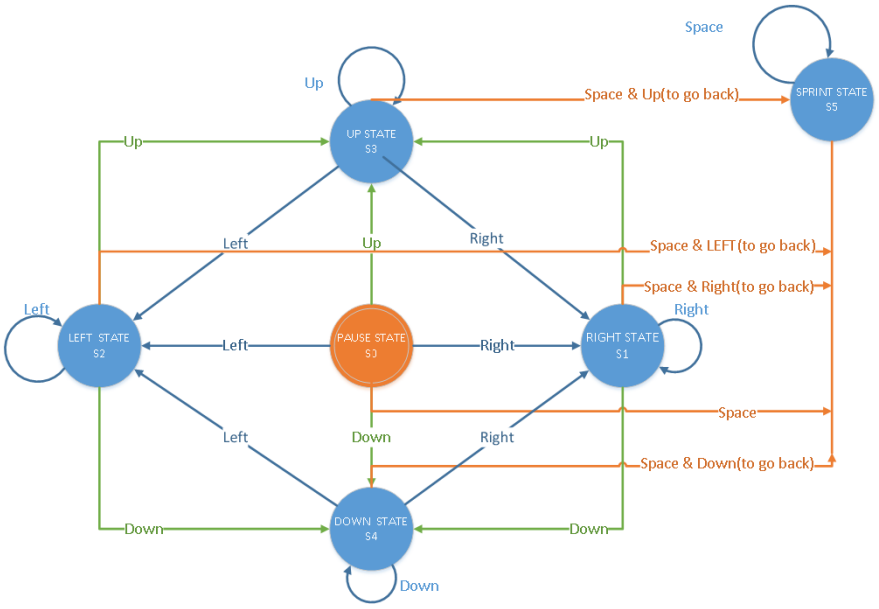
Overall Description

- Language: python3
- Library: pygame
- requirements to play: Two users and only one keyboard

Features, Functions

1. Python `pygame` is used to make the entire GUI game, and `auto-py-to-exe` is used to make the executable file.

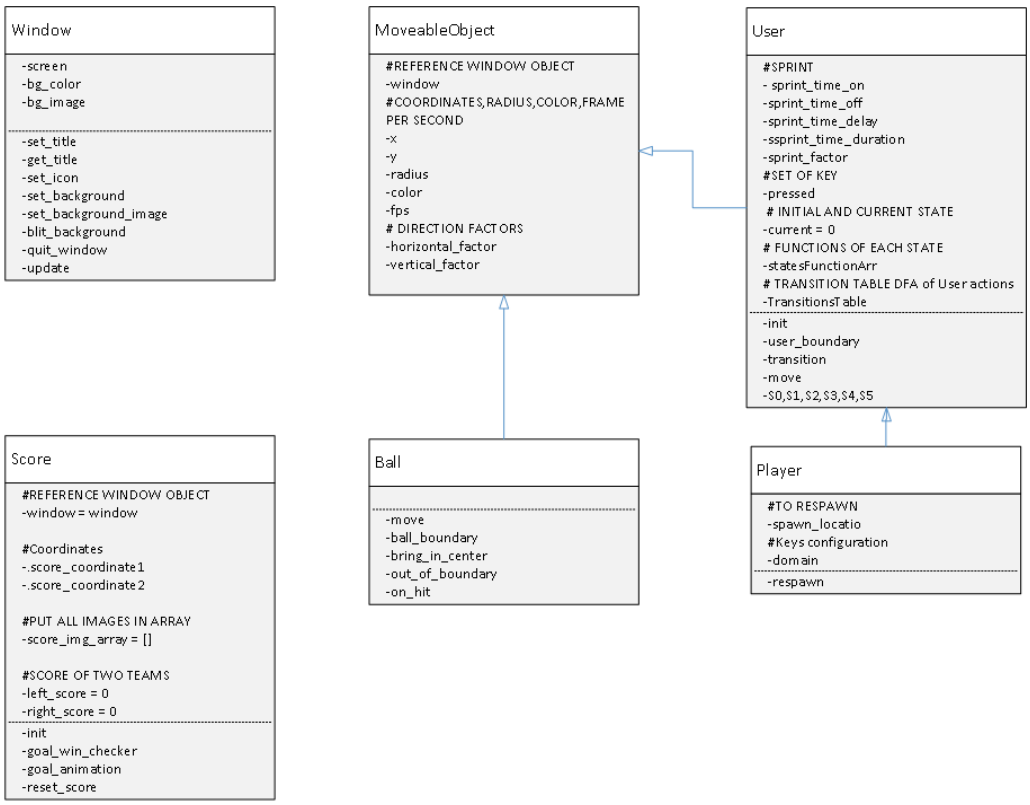
2. The player's entire movement is controlled by a finite state machine (Deterministic Finite Automata).



To handle the movement `key release` and `key pressed` events, a new condition and additional `set array` are introduced. When a user presses numerous keys and only one or two of them are released, the unique condition causes the transition to return to the initial state and utilizing the set array, the keys that haven't been released yet cause the transition to new states following the keys that are already pressed.

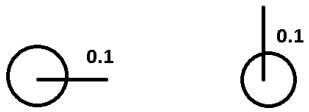
```
self.TransitionsTable = [
      [1,2,3,4,5],    #initial    0
      [1,1,3,4,5],    #right      1
      [2,2,3,4,5],    #left       2
      [1,2,3,3,5],    #up         3
      [1,2,4,4,5],    #down       4
      [1,2,3,4,5]     #sprint     5
  ]
```

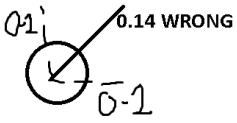
3. The concept of Inheritance is used a little.

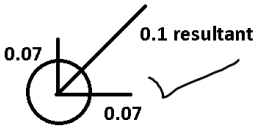


4. Controlled the increase in diagonal speed as a result of the horizontal and vertical motions.

x = 0.1
y = 0.1
h_speed = sqrt(x^2+y^2) => 0.14
But i wanted to go at same speed diagonally so,
so i need to decrease x and y speed in order to go at 0.1 diagonally
0.1^2 = x^2 + y^2 where x=y=k
0.1^2 = 2(k^2)
k = sqrt((0.1)^2/2))

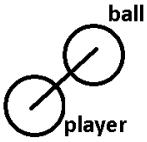






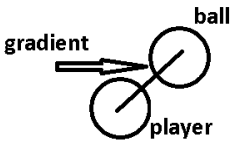
5. In-Elastic Collisions physics is applied with true angles between the user and the ball, as well as between the ball and the window boundaries.

BALL HIT FUNCTION



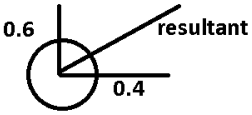
if player is not moving then simple refelection both horizontal and veritcal component *-1

if player hits the ball



deg =tan_inverse(absolute(gradient)) : in degrees
deg = deg/90 normalize between 1 and 0

60/90 = 0.6
vertical_increase 0.6
horizondal_increase = 0.4



vertical_factor = deg
horizontal_factor = 1-deg

if player axis is from right hand side means greater than ball axis then movment should be opposite so
vertical_factor *= -1
horizontal_factor = *-1