

Game Document

Introduction:

- The name of the game is called FoosPong
- Game Screenshot



- Instructions
 - Version
 - OS Version : macOS Mojave 10.14.6
 - Python Version : Python 3.9.12
 - Pygame Version : pygame 2.1.2
- Controls
 - Q and A control the white goalie and defenders on the left side
 - W and S control the white midfielders and attacker in the middle
 - Up Arrow and Down Arrow control the black goalie and black defenders on the right side
 - Period and Semicolon control the black midfielders and attacker in the middle
 - Space will be used after a player scores. It helps to unstick the game if the ball is stuck in the goal

Game Design:

- Mechanics

- The gameplay loop is the continuous button pressing that the user will do to move their characters around the field to try and score against the other team.
- The core mechanics of the game are that of pong. They are very simple where the player will move up and down to hit the ball the opposite direction that it came.
- The gimmick that was discussed in the game idea submission was the idea of changing the size of the ball after a certain amount of time or making the players bigger. I have decided to try and make the ball smaller. It seems to not work as intended as this is a known error. I have been trying to fix it but to no avail. It will make it smaller but eventually just disappear after 30+ seconds.
- I have looked all over the internet and I have not seen a pong/foosball game like this. I think this game might be the only one out there that offers this sort of nostalgic feel for the good old days of playing foosball with your friends.
- It utilizes the strength of pygame but utilizing some sprites and rectangles. Pygame loves sprites and their collisions and I utilized those for my game to be successful.
- Story
 - There is no real story behind the game. It is foosball. The only story is the story of remembering how they played the game when they were younger as a way of reminiscing. There is no protagonist, it is a two player game so the only antagonist is the other player you are playing.
- Player Experience
 - I want the player to experience the game like they are playing foosball but on a computer. As someone who enjoys foosball and other games like that, I wanted to create that kind of experience for the player.
 - One challenge is that the player needs to be cognizant of the fact that their players can move off screen. They need to make sure to keep the players on the screen for the best experience in the game.
 - Scoring a goal is the best reward they will get in the game.
 - The score will be updated when they score a goal as a way of providing feedback for them.
 - The visual element that will enhance the game is the score updating but also the ball being a soccer ball so they remember the game they're playing.

Game Design Changes:

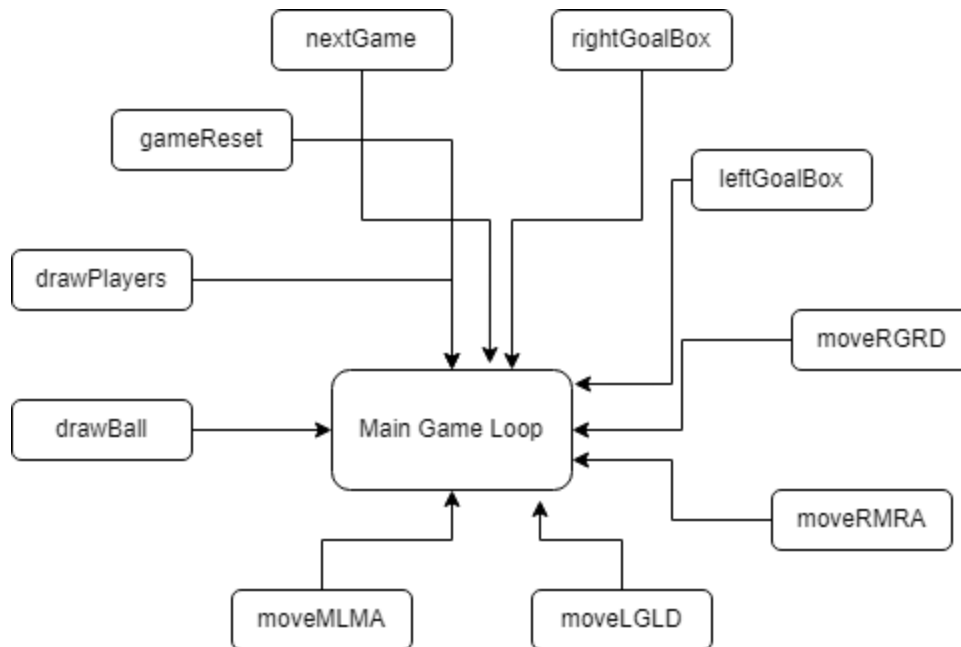
My original 2D game is a 5v5 soccer game. There will be one attacker, two midfielders, and two defenders, and a goalie. The movement of the footballers will be similar to that of foosball where the defenders and goalie will move in parallel and then the midfielders and attackers will move in parallel together. The ball will fall to the closest footballer. Once you have the ball, the player cannot move any of their footballers. The player will then aim with the mouse and shoot in a direction. Initially there is no thought of using any power ups but two power up

options I can look into are a “Boost” and a “Giant” powerup. “Boost” will make the ball go faster towards the goal, almost guaranteeing a goal. The “Giant” powerup will make your players bigger making it better to defend but harder to score as your players will block the ball. There will be a scoreboard and the first to 10 wins the game. After completion, it will ask if you want to play again or quit and then the program will act accordingly. At the beginning of every game, the player can choose between 8 different flags to represent their footballers and then the second user will choose between the other 7 flags. (USA, Argentina, Brazil, France, England, Netherlands, Belgium, Italy). It will be a 2 player game.

The change that is consistent from the first milestone is still the same. This was to not implement an AI but rather to focus on making the game two player first. A two player game is much easier and makes more sense to complete rather than try to implement an AI that makes its move sensibly. The architecture of the project will remain the same. I am attempting to implement a MVC and so far this is going well so I am pleased with that. The user interface has not changed at all. I still plan on using the user controls similar to that of foosball and that is the plan going forward. The other change I want to implement is to make the game closer to foosball. The amount of difficulties with just putting the sprites on the screen and the files interconnected has really delayed my milestones and I think foosball would still be a moderately difficult game to produce and still be fun to develop. With a week left until the deadline, I think this is reasonable to pursue. Along with the change of the overall objective of the game, I want to implement a random number generator at the beginning that will ultimately decide which player gets the ball when the game starts. Should really allow the game to have a bit of random fairness to it.

The final game design was to make a foosball and pong combination game. These changes were made because of difficulties encountered with the files being linked. There were a lot of setbacks in that department which made for these necessary changes. They have changed as I have not learned too much more useful aspects of pygame and I had to stick to basic ideas and what I knew to actually complete a game.

Game Development and Documentation:



This is how the different functions and classes work together. The player input is received in all of the functions with the prefix move. The state of the game is stored in both gameReset and nextGame as these help move the game along. The main loop also contributes a little to the overall function and state of the game. The view is updated in rightGoalBox, leftGoalBox, and the two draw functions. These help to draw the different things on the screen to make the game look good and so the player can actually play.

A major bug is that the gimmick does not work as intended. The ball is supposed to get smaller but eventually if no one scores, the ball will just disappear. This is a bit of a problem but forces players to either score or concede before you lose all of your progress. Another bug is that sometimes the goals are a little off because of how the rectangle is sized so the ball may look out of the goal but it counts as a goal.

Group Member Roles:

I did all of the work of the group since I was in a solo group.

Milestone One (3/30)

- Field, and Ball Created
- Initial Sprites uploaded

Milestone Two (4/15)

- Players created with appropriate sprites
- Movement of the players

Final Game Submission + Exam Presentation (5/3)

- Completed game with completed Game Documentation
- Combination of players moving, adjusted ball physics, and footballer sprites
- Scoreboard updating every time there is a goal

GitHub Link:

<https://github.com/djhusto/4160FinalGameProject>

It is run by downloading the file and typing “python3 allinone.py”

Video Link:

It is uploaded onto github. Could not find a good way to link it so you can see it in the github if you want to watch a video of the gameplay.