Foosball Game

Software Design Document

Clinton Oppong

Date: 2 October 2019

CS 225, Fall 2019

Embry-Riddle Aeronautical University

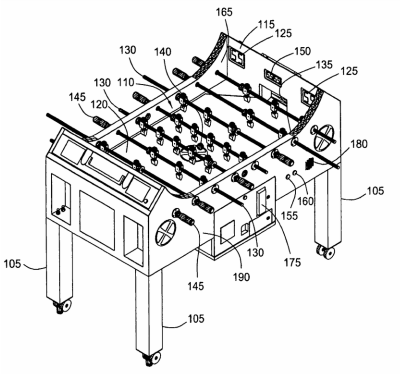
Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

**Introduction:**

This project is a simulation based on a table-top soccer game called Foosball. The aim of this game is to use the controls placed at the sides of the table using wooden characters to move the ball into the opponent’s goal. This game was invented in the 1921 due the rise and fame of soccer in Europe. So far, there have dozens of these games out there. Furthermore, majority of these game are direct descendants of PONG. However, what this project shall achieve is to fundamentally change the way people play the game by giving players different choices they couldn’t have achieved in the preceding versions.



Czech Republic Foosball (n.d)

**Problem Description**

This program shall simulate the base form of table-top soccer game with the implementation of an artificial intelligence (A.I). This game is a two-player game where one opponent faces against the other player. However, the core gameplay aspect will be based on player verses A.I, whereby, the A.I will pick a strategy in accordance with what formation the player picked in other to the player. This program shall have multiple input files which will be an array list of different formations. Also, Keyboard inputs which will act as the controls for the player and output them on GUI (graphical user interface). This program will use the error out to rule out incorrect inputs from the player. For example, wrong control inputs using alphabet instead of using the arrow keys or inputting the wrong time and difficulty level. The program will have a multi-level inheritance, in which the child class will identify different attributes of the game like the speed of the ball, the reaction speed of both the player, the A.I, time, and score limit. The subclasses are going to act as the parent of the child class by modifying the different attribute based on the users input. However, other child class will handle the A.I strategies and help it pit it against the formation the player picked. An example being if the people 3 – 4 - 2 – 3 formation the A.I will counter or “return” the formation by picking a 5 – 4 – 3 formation.

Tactlessly, this does not provide the best experience for the player because it could easily be re-countered, but unless the A.I can read predictable behaviors from the player and adjusting its behavior.

Although, using the first round to record the players behavior and using that to counter the opponent can provide a real-world aspect to the game.

**After the Code**

Interestingly, after I finished coding the foosball game, I encountered I few issues which were:

* Collision Detection

The ball was supposed defect as soon as it came in contact with the players.

* Path Transitions

The A.I were supposed to following a specific path while in transition however, the game would not start because both frames were different.

* Time Countdown

Whenever the timer starts it gets out sync and ends ups pausing the game.

Although, majority of the works without the game crashing the are certain codes that can’t function like they are supposed to like the players and AI scores can’t work because they can interact with one another. Due to overambition certain user stories didn’t make it.

**User Stories:**

|  |  |
| --- | --- |
| **User Stories** | **Notion of done** |
| 1. As a player I want to move my characters. | By running a for loop with (x, y) coordinates and have it run every time the user inputs the correct key. |
| 1. As a player I want to know if I scored. | By running a method with an if statement that will return a print of if the player scored. |
| 1. As a player I want choose between two player or player vs A.I. | By have a method with an if statement will return the right method that contains the A.I or Player based on the users input. |
| 1. As a player I want to choose what difficult I want. | By have a method with an if statement will return the right method that contains the right difficult based on the users input. |
| 1. As a player I want to pick how many rounds I want play for. | By having the user input the amount of round they want and have a for loop the will run that many runs, and plus have it displayed on the GUI. |
| 1. As a player I want to pick the time for each round. | By having the user input the amount of time they want and have a for loop take care of the rest, and plus have it displayed on the GUI. |
| 1. As player I want to pick the formation my players will take. | By giving the user a list of the available formation and having them inputting a value to pick which formation they want. Also, extracting the formation from an excel with the proper coordinates. |
| 1. As a game designer I want to prevent player form inputting the wrong inputs. | By running a while loop which will limit them form picking the wrong value also having it printing an error which will print as a result. |
| 1. As a game designer I want the A.I to counter player formation | By making a method with an if statement that will return the counter formation in accordance with the players input. |
| 1. As a game designer I want to the A.I to counter predictable behavior. | * Don’t know how to do this - |
| 1. As a game designer I want to have scores to be displayed | By having a GUI that shall display scores on the opposite ends of the interface. |
| 1. As a game designer I want to have winner declared after each game. | At the ed of each game I shall have an if statement will print out a statement that will declare the winner by comparing the biggest score. |

**Requirements**

1. Ball Speed: This affects the speed of the ball specifically which all the play to change its speed.
2. Player Movement: The player can move its characters with keyboard to help position the players, but they only move on the y axis.

Keys to control the players are Ups {QWERI} and Downs {ASDFK}.

1. Win or lose: After the timer runs out, they print out a statement to whoever has the most scores.
2. Game Timer: The player can pick the amount of time they want the game to run for.
3. Difficulty Pick: The player can pick how intensity of the game.

**Test Cases:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req’t**  **ID** | | **Test**  **Case**  **ID** | **Initial**  **Conditions**  **And Input** | **Expected Behavior**  **Or Output** | **Actual**  **Behavior**  **Or Output** | | **Pass**  **Fail** | |
| 1 | Ball Speed = sqrt (x2 + y2) | |  |  | | Seen On GUI | | p | |
|  | | A: Test Case 1 | X = 3; Y = 4 | 5 |  | |  | |
|  | | B Test Case 2 | X =15; Y= 20 | 25 |  | |  | |
|  | | C: Test Case 3 | X = 6; Y= 13 | 14.31 |  | |  | |
| 2 | Player Movement | |  |  | Seen On GUI | | p | |
|  | | A: Test Case 1 | left Arrow Key | Character moves left |  | |  | |
|  | | B Test Case 2 | Right Arrow Key | Character moves Right |  | |  | |
|  | | C Test Case 3 | Enter Key | “Nothing” |  | |  | |
| 3 | Win or lose | |  |  |  | | P | |
|  | | A: Test Case 1 | Player win  Print “player win” | Print “player win” |  | |  | |
|  | | B Test Case 2 | Player Lose  Print “player lost” | Print “player lost” |  | |  | |
|  | | C Test Case 3 | Player Draw  Print “player Draw” | Print “player Draw” |  | |  | |
| 4 | Game Timer | |  |  | Key Frames Slows down (stop) | | F | |
|  | | A: Test Case 1 | 30 seconds | 4:59 sec |  | |  | |
|  | | B: Test Case 2 | 60 seconds | Nothing |  | |  | |
|  | | C: Test Case 3 | 300 seconds | 29:59 sec |  | |  | |
| 5 | Difficulty Pick | |  |  | Seen On GUI | | p | |
|  | | A: Test Case 1 | Easy | Ball Speed = sqrt (x2 + y2) |  | |  | |
|  | | B Test Case 2 | Normal | Ball Speed = sqrt (x2 + y2) \* 2 |  | |  | |
|  | | C Test Case 3 | Hard | Ball Speed = sqrt (x2 + y2) \* 4 |  | |  | |

**References:**

Czech Republic Foosball Playing Style and Table Construction. (n.d.). Retrieved from https://www.foosballsoccer.com/czech-style-foosball.html.

**Appendices:**

This is optional, but may include external sources, source code, or other related material.