

Project Report

Lab Batch 37

CS101 Course Project:

FOOSBALL

Introduction:

Foosball is basically table football. There's a Foosball table with two sets of players controlled by rods. There is a goal at each end and a place to serve on each side in the middle. Each player/team uses their handles to move their men to strike the ball into the opposing teams goal.

In our project , we captured the same essence of the game . Here we have two teams , Team Blue and Team Red, with 7 players each.

- 3 in attack.
- 3 in Mid-Field.
- 1 Goal-keeper.

Team Blue is A.I . Team Red is User Controlled. You can control the movement of all the players by clicking on left and right Bitmap images. There's a stop button to stop the movement. Its a 5 Goals Match.

The game is fast and funny and everyone cheers and laughs. Don't be surprised if you get so excited you sweat a little bit from the excitement!!!

Enjoy!

User Interface:

The interface mainly comprises of 4 windows namely :-

1. Foosball(W0): First window displayed when we start the game. Clicking anywhere leads to the main menu(W1).
 2. Main menu(W1): It includes button Play (leads to foosballWindow), Instruction (leads to W4), Exit (closes W1).
 3. Instruction and Credits(W4): It includes instructions to play and score, and Credits
 4. Game Window(foosballWindow): Game is played in this window. It contains the play field , one box to display the score and another that displays the time elapsed in seconds.
- Game ends when any team has scored two goals. The final score is displayed at the center of the window.

Running the Game:

1. Copy the file EzWindows.tgz untar the tgz file in your home directory
 - `tar -zxvf EzWindows.tgz`
2. To run programs, go to the Code_other_files directory of the untarred folder and click on the executable file "foosball2"
3. Click anywhere on the opening window to go to the Main Menu

To Compile:

1. Copy the file EzWindows.tgz untar the tgz file in your home directory
 - `tar -zxvf EzWindows.tgz`
2. Type the following commands in the terminal
 - `cd EzWindows`
 - `cd EzWindows`
 - `cd lib`

- make
- It will finish with errors. Make sure that these lines appear just about 7-8 lines above the last output.

```
ar: creating libezwin.a
ranlib libezwin.a
```

- If this does not appear then most likely your computer is missing the libx11-dev library file. Report this to the sysad and get it installed.

- Now to run the file "foosball2.cpp" run this

```
g++ -o foosball foosball.cpp -I ../EzWindows/include -L/usr/X11R6/lib -lX11 -L../EzWindows/lib -lezwin
/usr/lib/libXpm.so.4
```

- To run the program use

```
./foosball2
```

Method of playing

The user plays this game using mouse. Below the timer there are three rectangles provided.

1. The left-most rectangle which has a 'left arrow' sign when clicked, moves the user players leftwards.
2. The rightmost rectangle which has a 'right arrow' sign gives the user players a speed towards right when clicked
3. When the user clicks on the central rectangle (one with "STOP" written on it) his players stop.

Thus with these three mouse-driven operations, the user can conveniently control his players and position them properly to make glorious saves, retain possession of ball in the mid-field or hit superb winners.

Further details

The ball shoots off in three different directions (left, almost straight and right) depending upon which of the three parts of the player (left, middle and right) it collides with. The new direction attained by the ball is independent of the previous direction of the ball. It only depends on the part of the player it collides with and nothing else.

In a particular row any position can be attained only by one player. Thus the user must carefully make a good guess as to which is the player who will be able to reach out to the ball.

Artificial Intelligence of the Game:

Each team consists of 7 players-3 attackers, 3 midfielders and a goalkeeper.

The theme of A.I. is that the computer players are moved in such a way they reach such a place where the ball is expected to come.

The field is divided into 7 regions corresponding to each player. First the system checks the position of the ball (region). For this, the coordinates of the ball are received using

- a) GetXDistance
- b) GetYDistance

Now, the system predicts the x-coordinates of the ball when the y-coordinate of the ball is equal to the y-coordinate of the player of that region. In calculating the predicted coordinate, we used the initial coordinates of the ball and the direction of its velocity (i.e. $m = \text{increment in } y / \text{increment in } x$)

The ball can also collide with the wall, but since the angles are constrained to 45 degrees, therefore simple laws of reflection have been used in predicting the future x-coordinate of the ball. Now the players are moved by 0.1 cm in such a way that it comes closer to the predicted co-ordinate. This process is repeated till the player reaches the ball by including the entire code in the main Loop. When the player reaches the ball, it is commanded to remain stationary so that they are available to hit the ball. But this does not mean that they can always hit the ball because they might not always be able to reach the desired co-ordinate in time. The code also ensures that left players hit the ball with their right portion and right players hit the ball with their left part so that

ball is hit towards opponent's(user) goal.when the ball moves in the opposite direction i.e. towards the user's goal,the computer players are held stationary.

COMPLETION STATUS:

We divided our batch into three teams to do the following work-

1. Graphical user interface
2. Collision
3. A.I

In GUI, the initial main menu displays “play” , “instructions” and “exit” .When we click on play, main field appears showing the players and the ball moving and colliding perfectly. The screen also displays time and score. We have kept two mouse clicks to get user input through which the user can move his players and there is a stop button through which user can stop his players whenever he wants to.

In collision part, ball rebounds perfectly after colliding with the walls, following laws of reflection. Whenever ball enters the goal region, no collision takes place, instead the text “goal” appears at the centre of the screen. Ball collision with the players is designed in such a way that it gets deflected on different paths with different speeds according to the part of the player with which it collides.

The basic theme of the AI team is to predict the position of the ball and then move the AI players accordingly. Each player is restricted to move in particular region. The field is divided into seven such regions. The slope of the trajectory of the ball is determined by posy/posx (posy and posx are the increments in y and x coordinates respectively). The region where the ball is predicted to reach is calculated and the corresponding player is moved.

FURTHER IMPROVEMENTS:

1. Different difficulty levels could be introduced where the AI players move at faster speed and their chance of hitting the ball increases.
2. Better graphics using a more potent software.

Team Members:

AI Team:

Abhishek khanna*	10D070045 (Team Leader)
Hardik Agarwal	10D070042
Rohan pillai	10D070053
Karthik sharma	10D070014

Collisions Team:

Pavan jatale	10D070004
Prakhar Khandelwal*	10D070040 (Team Leader)
Shanay shah	10D070007
Virat shejwalkar	10D070021

Graphics and UI Team:

Ashwin Kachhara**	10D070048 (Team Leader and Batch Coordinator)
Himanshu soni	10D070050
Mrinal Joshi	10D070028
Prashant Borkar	10D070025
Mayank Maloo	10D070063

Total Lines of Code Written =1150

Total Hours Spent on Project =426