University of Dhaka

Department of Computer Science and Engineering
CSE - 2112
Project Report: JAVABALL

Submitted To:

Mr. Md. Shiplu Hawlader
Lecturer, Dept. of CSE, University of Dhaka
Mrs. Asma Enayet
Lecturer, Dept. of CSE, University of Dhaka
Mrs. Iffat Anjum
Lecturer, Dept. of CSE, University of Dhaka

Submitted By:

MD. Tanvir Alam
Roll: 061

2nd Year 1st Semester
Session: 2015-16
Date-of-Submission: 2nd May, 2016

Table of contents:

Serial	Topic	Page number
01	Introduction	3
02	Project introduction	3
03	User end outline	4
04	Game outline	6
05	Special features of the game	7
06	Custom controller	8
07	Details of search with automatic ranking	9
08	Server and outline	9
09	Helpline chatbot with automatic reply	10
10	Basic concepts of OOP	11
11	Area of further improvement	11
12	CPU Performance	12
13	Discussion	12
14	Overview & Conclusion	12
15	Special thanks to	13
16	Software & Used References	13

Introduction:

The project is intended to make the students familiar with application of object oriented programming]. Implementation of the students' theoretical knowledge of object oriented programming in some real-life scenario was the objective of this project.

Project introduction:

The project is a mainly an implementation of 3D football game, in which the players have to control their avatar to score goals. The game is named “JAVABALL”, which is obviously because of its implementation using java language. We also made our own controller to play the game which is made to be interact with JAVA. Beside the game, we also implemented Artificial Intelligence, related graphical user interface and JAVA based networking. Lastly, there is also an implementation of data searching where the search ranking updates automatically based on user activity. This search is used to provide store finding option.

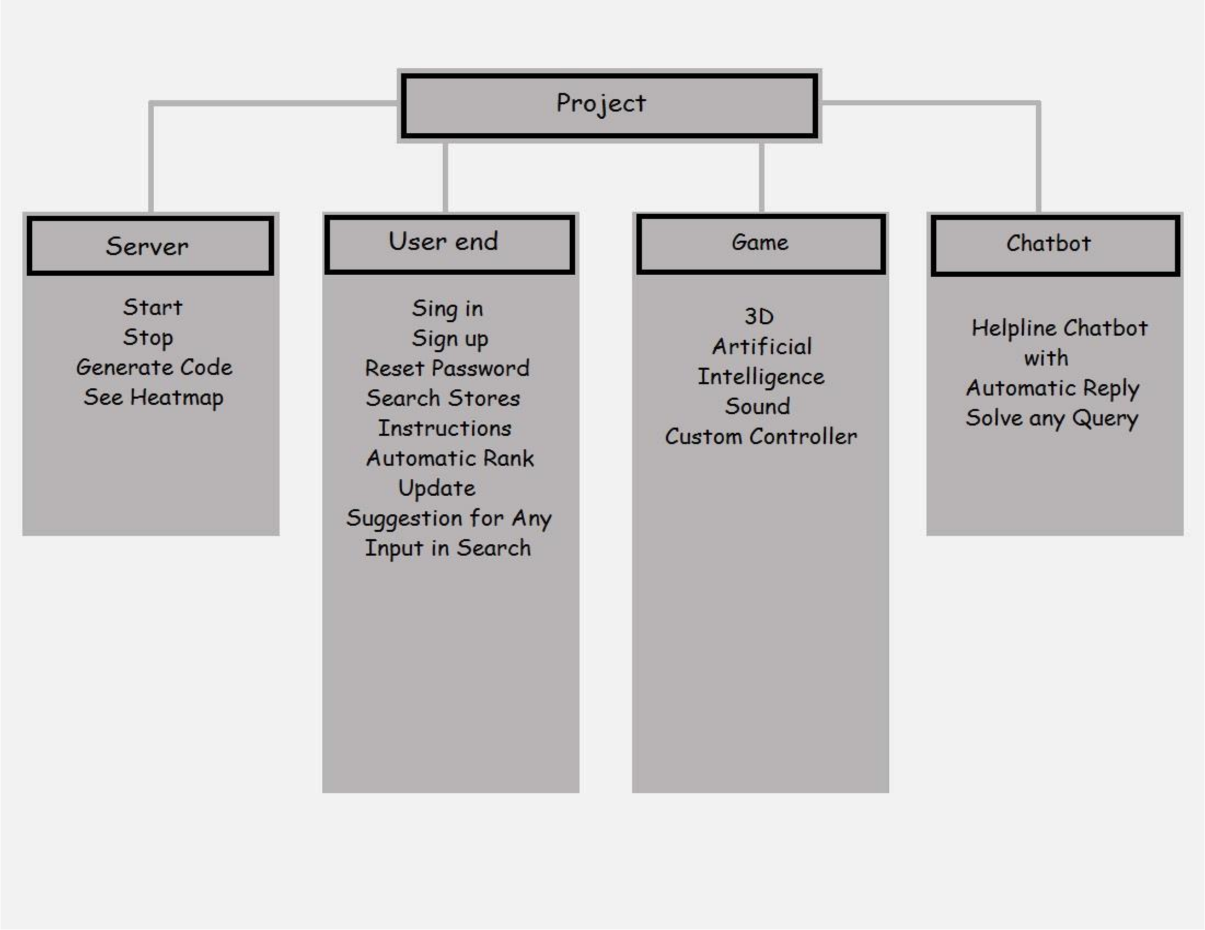


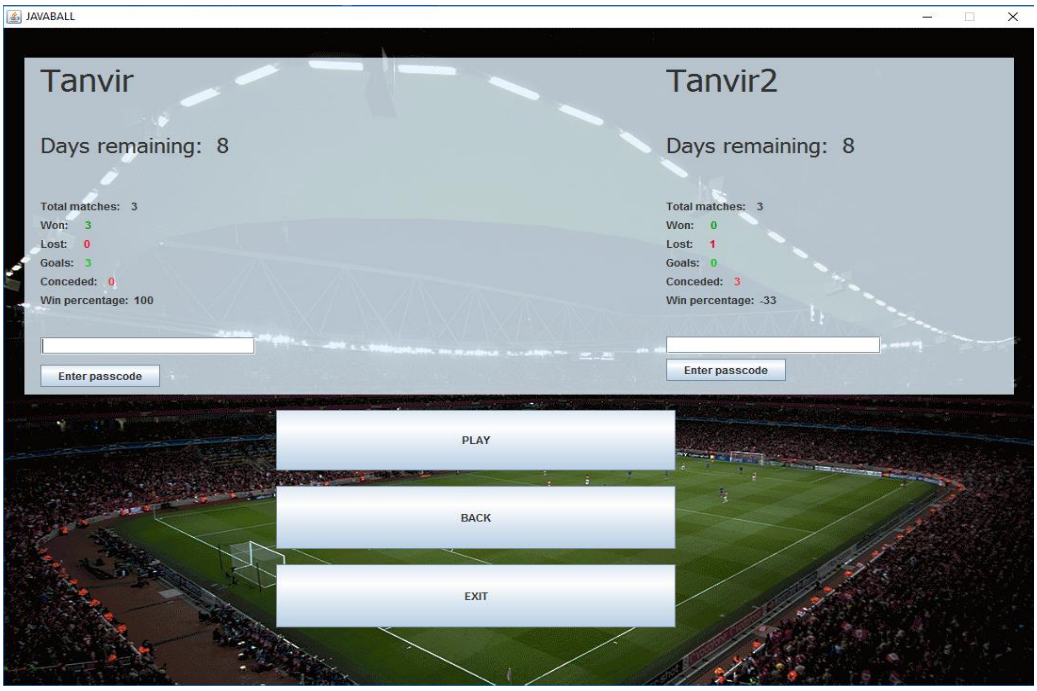
Figure 1: Project Structure

User end outline:

Firstly, the game is implemented as a premium one. User can download or get the game for free but they have to pay to play. Here we used some codes to give access to play for certain days. User have to buy access codes from stores.

To control the access of the game there is a graphical user interface to sign up, sign in, play game etc. To sign up user will initially need an access code.

The simple version of the game is played by two users. Both users have to sign in with their account. After signing in users will be able to watch their game statistics.



Tanvir

Days remaining: 8

Total matches: 3
Won: 3
Lost: 0
Goals: 3
Conceded: 0
Win percentage: 100

Tanvir2

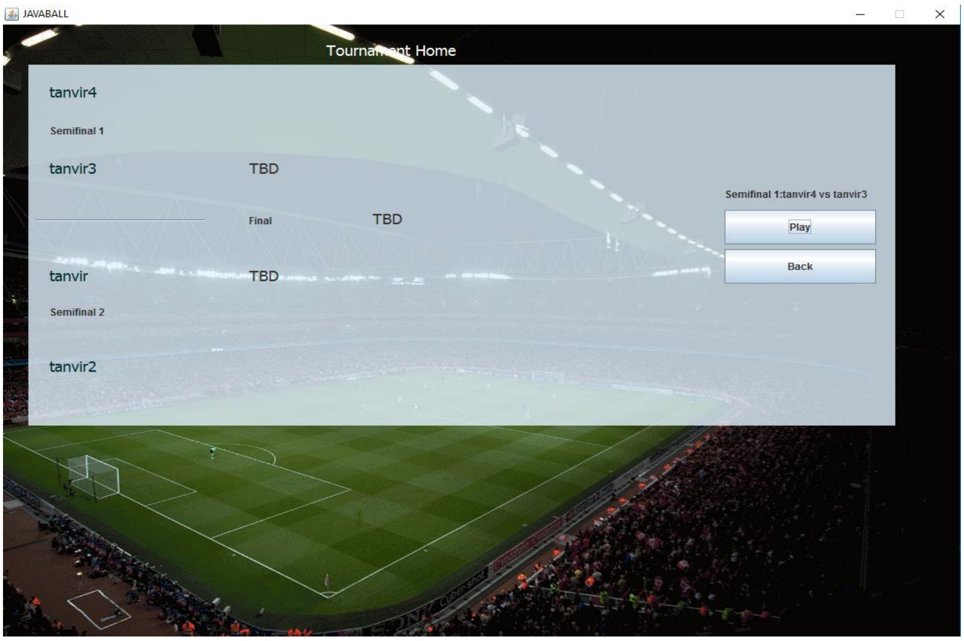
Days remaining: 8

Total matches: 3
Won: 0
Lost: 1
Goals: 0
Conceded: 3
Win percentage: -33

Figure 2: User Statistics

There user can also see how many days he has remaining. User will also have the option to enter new access code or passcode. If both user have enough access days remaining they can play a game. The game is described later.

There is also another version of the game. Here four users can play a tournament. There will be matches between users in the format shown below.



tanvir4
Semifinal 1
tanvir3
TBD
Final
TBD
tanvir
TBD
Semifinal 2
tanvir2

Semifinal 1:tanvir4 vs tanvir3

Figure 3: Four Player Tournament Fixture

We have added sufficient instructions for users. There is also an option to reset the password. The store searching system is also described in the following past as the game description.

Game Outline:

As before we said the game is a football game. Precisely it is an implementation of "Sunday league football" which is actually football for amateur with no goalkeeper. The game will continue for two minutes per match. There will be eight players in each team. User will automatically get control of the player who is nearest to the ball.

The objective is to scores goals by kicking the ball into the post. Whoever scores most goals in two minutes will be declared as winner.

Player will have the option to pass or kick the ball when he possesses the ball. When a player lost the possession, he will try to recover by running towards the ball. If the opposition possess the ball then he will have to tackle to get it back.

In a game, one user will play with the keyboard and another user will use the controller made by us.

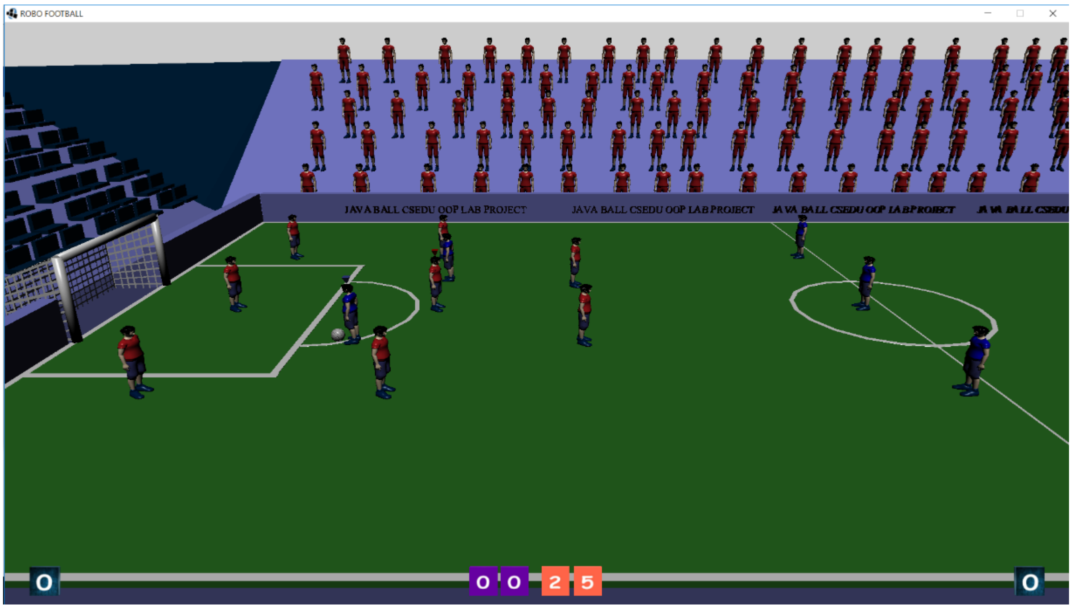

ROBO FOOTBALL
JAVA BALL CSEDU OOP LAB PROJECT
JAVA BALL CSEDU OOP LAB PROJECT
JAVA BALL CSEDU OOP LAB PROJECT
0
00 25
0

Figure 4: Game Interface

Special features of game:

1. Artificial Intelligence: In this game, the users only control the player who is nearest to the ball. But the other seven players need to be in correct position to get a pass or defend. To define the position of these non-active players with the positon of the ball we implemented Artificial Intelligence.
2. Interactive 3d Audio: While playing the game, sound of crowd will be played. There are many interactive sounds as when any team scores we can hear the sound of the commentator screaming “goal”.
3. 3d game: The whole game is displayed using 3d graphics. The stadium, player, ball everything is modelled in 3d by us. Autodesk maya and Blender is used to model the 3d objects.
4. Player animations: All the player’s animation is implemented by coding. We did not use any software generated animations. All the walk-cycles and kick animations are implemented by swapping 3d models frame by frame.
5. Reduce CPU usage: We used various approaches to reduce the CPU and graphics usage. For example, we limited the quality of rendering the 3D objects with distance. Here we do not need to render the spectators in the gallery in full resolution. So, we cut the number of polygons to save memory.
6. Collider: We did not use any game engine or library to make the collider. We implemented this thing successfully. It will prevent all the glitches. For example, in many 3D games it is seen to characters running through other objects as shown below. But in our game, it will never happen.

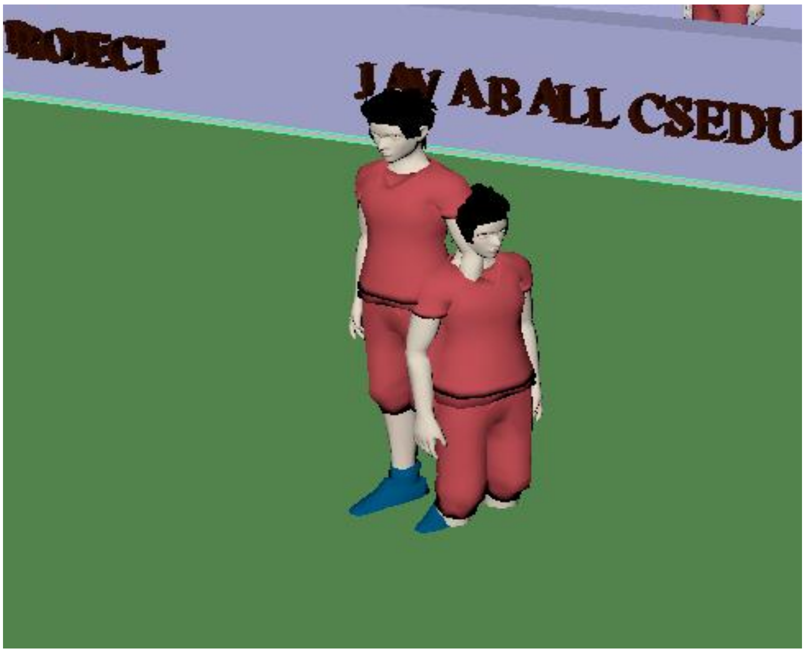


Figure 5: Glitch of 3d game. (There will be no glitch in our game)

Custom controller:

We made our own controller using Arduino. We used serial ports to send data. A thread in the game continuously read the data and performs action according to the data. We used six pulse switches to build this controller. It can be used in any computer through USB port.

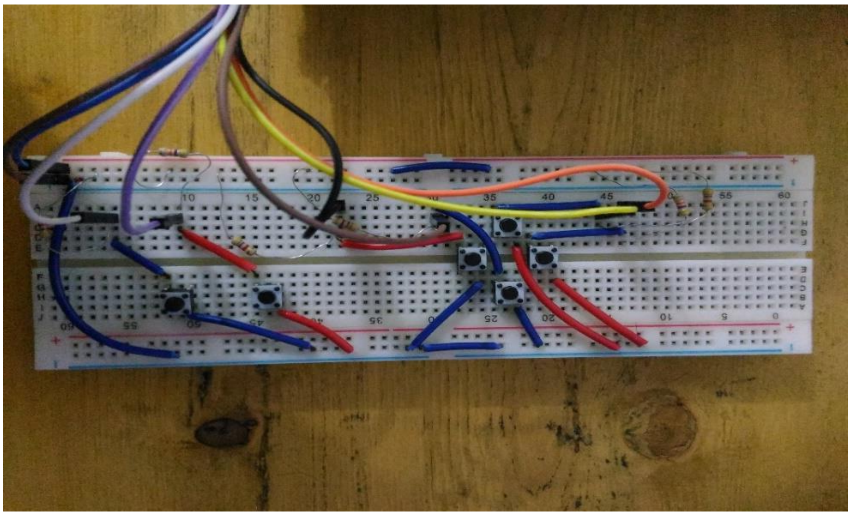
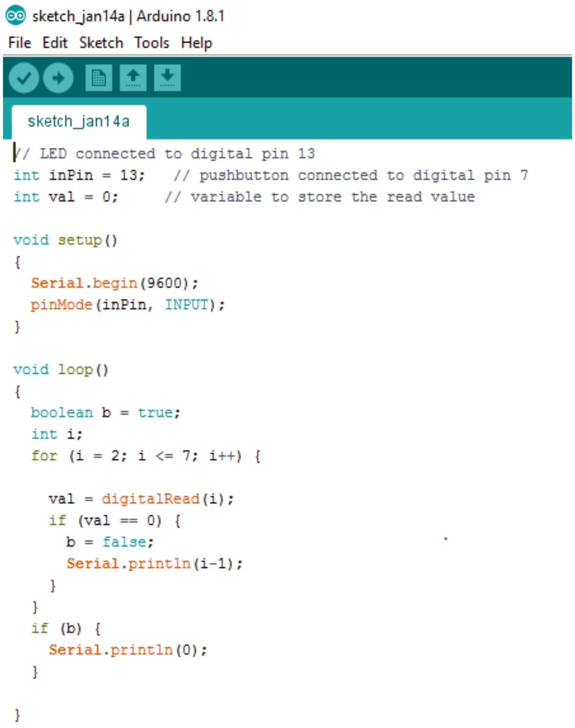


Figure 6: Custom Controller



```
// LED connected to digital pin 13
int inPin = 13;   // pushbutton connected to digital pin 7
int val = 0;     // variable to store the read value

void setup()
{
  Serial.begin(9600);
  pinMode(inPin, INPUT);
}

void loop()
{
  boolean b = true;
  int i;
  for (i = 2; i <= 7; i++) {

    val = digitalRead(i);
    if (val == 0) {
      b = false;
      Serial.println(i-1);
    }
  }
  if (b) {
    Serial.println(0);
  }

}
```

Figure 7: Arduino Controller code

Details of search with automatic ranking

In the user end we have an option to search for stores. Here user will type their area and we get suggestions of stores. When they click to see the detailed address of stores the key user searched for and the store id will be noted down automatically.

Initially, suggestions will be given based on the Euclidian distance of the character frequency vectors of the key user searched for and the results stored. This gives fairly relevant results. But it is obviously not the result. So, we put the noted key pairs described before in our data sets. Now as our data set get bigger the user will get more relevant result and the ranking will be updated automatically.

For example, let us assume when I search for “b” initially I will get the first two results appears as “Barisal” and “Bangkok”. Now if I have more users in Bangkok, I need to put “Bangkok” before “Barisal” in the search result.

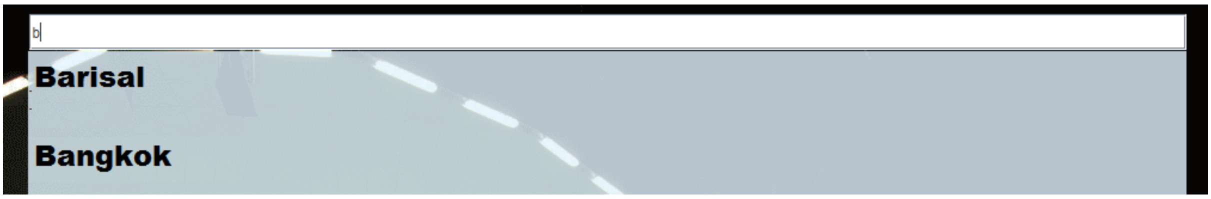


Figure 8: Search Stores

Now as most of the user will click “Bangkok” our data set will have more pairs of (“b”, “Bangkok”) than (“b”,” Barisal”). So, Bangkok will come first in results.

This machine learning approach is also helpful to recover the cases where Euclidian distance of the character frequency vectors doesn’t give the correct suggestions.

Server end outline:

Beside the start server and stop server, we have option to generate codes and update the codes in the text area in right side. Here code means “Access code”.

The most interesting feature in this end is that we can see the average position of the players for each user throughout the games. There is an option to directly get the heatmap of the best player directly. Here best player is determined by the user with most winning matches.

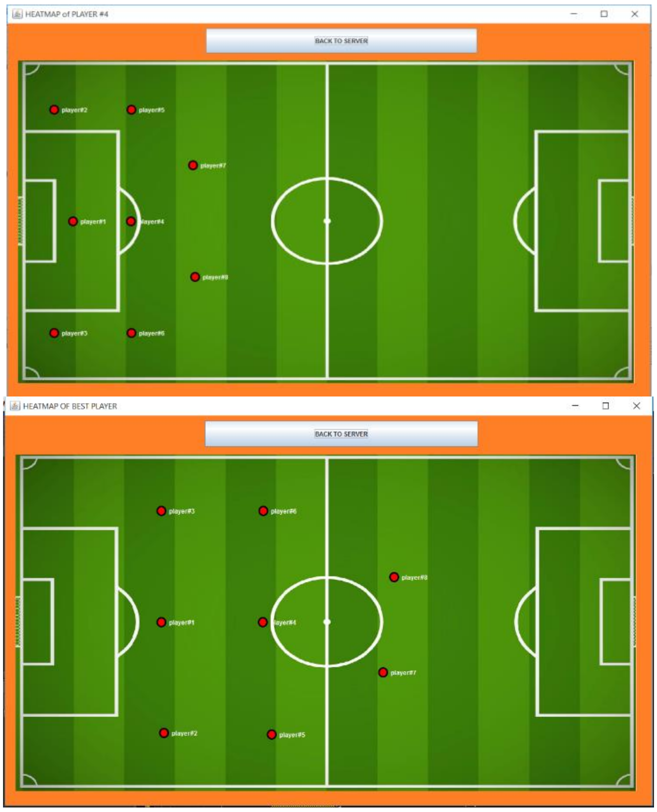
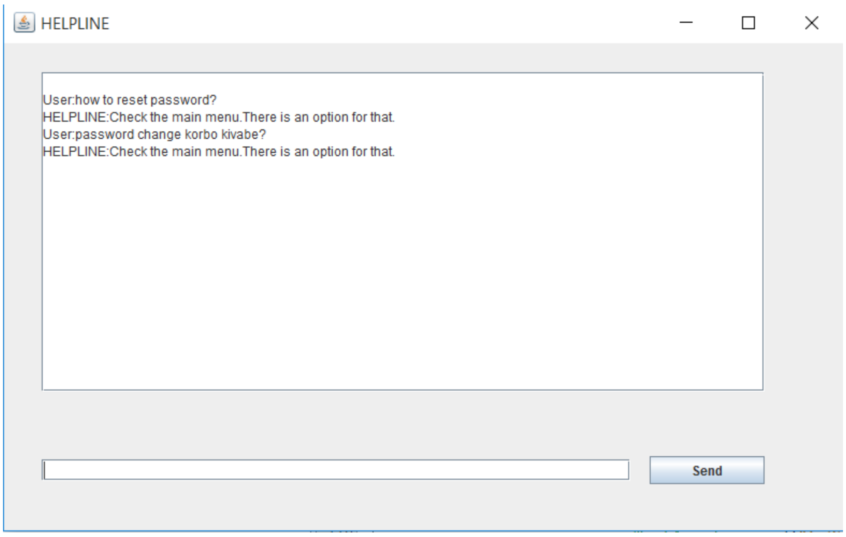


Figure 9: Heatmap

Here we can see first player played more offensive than next one. These positions are updated in server in each frame of the game. These data will also give us the hint about the best approach of playing the game.

Helpline Chatbot with Automatic Reply:

We made a helpline where users can contact with server admin to solve their problems. In most cases, there solution will be automatically generated. But if our system recognize that problem cannot be solved by itself, it will notify the admin to answer the question. When the admin answers a question about a new topic, it will automatically learn to answer about the topic for next time.



User:how to reset password?
HELPLINE:Check the main menu.There is an option for that.
User:password change korbo kivabe?
HELPLINE:Check the main menu.There is an option for that.

Figure 10: Some automatic relpies from helpline.

Some Examples of Basic Concepts of Object Oriented Programming:

Inheritance: We defined players of two teams as “player” and “enemy” class. We just inherited the properties of player class to create enemy class and did some changes.

Polymorphism: There are some changes in methods in player and enemy class. So, when we extended the player class, we just applied overridden methods where changes were needed. Besides, there were two types of animations – single frame and multiple frames. We implied method overloading in constructor while creating the animations as the animation class is same.

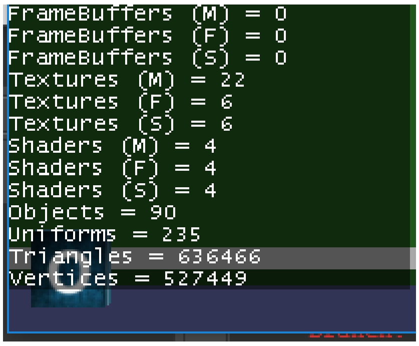
Area of further improvement:

Improved Artificial Intelligence: The Artificial intelligence can be improved by using machine learning approach. We thought of an approach by randomize the players position at some extent. In this version, the players’ position will deviate from current position at a small extent. Now as server is always getting the player and ball position, we can determine the better position according to game situation. As our data will get richer, the assumption will be more correct. Now if we change the position of non-active players according to our assumption from the data, their positioning will be much better.

Network Based Multiplayer: We tried to modify the same game to be played from multiple devices. We needed to transfer 28 floats per frame. At 30 fps, that will be 840 floats per second. But we faced some delay in the data transfer which caused synchronization problem in the gameplay.

CPU performance:

As the game is built using 3D graphics, it uses a lot of computing power. In the image below we have shown an arbitrary frame's display statistics. However, we can say that any device with at least Intel Core i3-3225 @ 3.30GHz, NVIDIA GeForce GTX 660 2GB / AMD Radeon HD 7850 2GB, RAM 8GB, OS Windows 7 64bit, DX 11 should work fine to run the game.



```
FrameBuffers (M) = 0
FrameBuffers (F) = 0
FrameBuffers (S) = 0
Textures (M) = 22
Textures (F) = 6
Textures (S) = 6
Shaders (M) = 4
Shaders (F) = 4
Shaders (S) = 4
Objects = 90
Uniforms = 235
Triangles = 636466
Vertices = 527449
```

Figure 11: 3D display statistics

Discussion:

In this project, our main aim was to apply object oriented programming for real life application. We tried to combine many others things also to make the project successful. We focused both on the outlook of the project and algorithms. The search, automatic helpline, and artificial intelligence needed to imply various algorithms.

Overview and conclusion:

However, we can say that the project was helpful for us to completely understand the usefulness of object oriented programming. Finally, our observation is object oriented programming represent real world better than structured programming.

Special thanks to:

Md. Shiplu Howladar, Lecturer
Dept. of CSE, University Of Dhaka
Mrs. Asma Enayet, Lecturer
Dept. of CSE, University Of Dhaka
Mrs. Iffat Anjum, Lecturer
Dept. of CSE, University Of Dhaka

Softwares Used:

- Autodesk Maya
- Blender
- Photoshop
- Netbeans
- jMonkeyEngine

References:

- **Java: The Complete Reference, Tenth Edition: Herbert Schildt**
- https://www.javatpoint.com/
- https://jmonkeyengine.github.io/wiki/jme3.html
- https://www.youtube.com/user/ThinMatrix
- https://www.youtube.com/watch?v=FHNWlECCN3U&list=PLYffAx5Cja9hoRVSXilO_vI16Z_ZEUnFF

The End