FOOTBALL-PENALTY-SHOOTER

CS352 Project-Computer Graphics and Visualization lab

PROJECT REPORT

COURSE INSTRUCTOR:
DR. SOMNATH DEY

SUBMITTED BY:
NISHCHAY SHROFF - 200001055
AMIT KUMAR MAKKAD-200001003
ATHARV BHAGYA-200001011

INTRODUCTION

FootBall Penalty Shooter, as the name suggests is a game similar to football penalties in real life.The game can be a 2-player game or an interactive game with the computer.The interaction can be in several ways.

1. Set the level of game to be played(easy , medium, hard, move goalpost)
2. Move around the camera with the mouse/touchpad to get a better field of view.
3. Zoom in/out using the keyboard, again to get a better field of view.
4. Set the angle of the shot, and thus point it in the desired direction.
5. Set the spin of the ball, that is the angle by which you want the ball to turn and its direction as well.
6. Set the power of the shot, that is, the speed of the ball.
7. View the goal statistics and switch from one Category to the other using the keyboard.

Note that the goal is defended by a moving and jumping Goalkeeper, which moves in the direction of the shot(in case of interactive game). He will deflect the ball on contact. Points are awarded whether the shot is a goal.

In 2-player Category , One player can be the attacker and can kick the ball in order to score a goal. The attacker has 5 chances, after which the game is over and the winner is declared.

SPECIFICATIONS

LIBRARIES

The main library used in our project is Open Graphics Library. Open Graphics Library (OpenGL) is a cross-language (language independent), cross-platform (platform-independent) API for rendering 2D and 3D Vector Graphics(use of polygons to represent image). OpenGL API is designed mostly in hardware.

To install OpenGl, type the command below
sudo apt-get install freeglut3-dev

RUN COMMANDS

To run the program, follow the give steps

- Open Terminal.

- Git clone https://github.com/amitmakkad/football_penalty_kick_opengl_cpp.git
- Move to the destination folder using the cd command i.e cd OpenGL-3D-Football-Penalty-Shooter
- Type the following command - g++ main.cpp functionalities.cpp shapes.cpp constants.h -lGL -lGLU -lglut -lm -o run && ./run
- The program will start running.

KEY CONTROLS

When the game has begun, an instruction page pops up on the screen.It has all the instructions which are described in the below paragraph.

Press **Esc** to move further into the game

- For Interactive Category- Type 2,3,4 to get into easy medium,hard level respectively
    1. Use the +/- keys for zooming in/out
    2. Press Enter key and the 4 arrow keys in order to set direction of ball velocity using the arrow.
    3. Use key ‘p’ to add spin in ball (Up on bar scale -> high spin and Low on bar scale-> low spin)
    4. Use the key spacebar to set the initial velocity of the ball.

- For 2-player Category- Type 1 to get into 2-player Category. The key controls for the attacker are the same as that in interactive Category.Now, 2nd player can control the goalkeeper. The key controls for goalkeeper are as given below
    1. Use ‘w’ to make the goalkeeper jump.
    2. Use ‘a’ to make the goalkeeper move left
    3. Use ‘d’ to make the goalkeeper move right

FUNCTIONALITIES

REFERENCES:-

GITHUB

THE PHYSICS

- **Ball physics**-This will consist of the physics and the trajectory tracer when the ball is shot. It applies gravity effects, frictional effects as well as controls the bouncing of the ball. It also looks after the collisions of the ball with the defender as well as the poles of the Goal Post, which uses oblique collisions.The calculations done for collision part are quite complex and are omitted for simplicity.Another interesting aspect of ball physics is Magnus Effect.
- **Magnus Effect-** We defined the velocity of the sphere/ball to change in the updatePos() method. We have used 2D physics of swinging ball and the equations of motion are:-

$$F_{netx} = -Cv_x\sqrt{v_x^2 + v_y^2} - S_0|\vec{\omega}|v_y \qquad (3)$$

$$F_{nety} = -mg - Cv_y\sqrt{v_x^2 + v_y^2} + S_0|\vec{\omega}|v_x \qquad (4)$$

One can use the left power bar for setting spin and right power bar to set initial velocity.

- **Goalkeeper physics**- The velocity,displacements and acceleration are calculated using the following formulae

v_x = constant (reversed in direction when pole is encountered)

$$a_x = \mathbf{0}$$

$$s_x = v_x t$$

$$v_z = u_y - gt$$

$$a_z = g$$

$$s_z = u_z t - \frac{1}{2} g t^2$$

Here uz=initial velocity in z-direction(constant), vz=current velocity in z-direction, az=acceleration in z-direction, sz=displacement in z-direction, vx=current velocity in x-direction, ax=acceleration in x-direction, sx=displacement in x-direction. In the MOVE_POST mode the goalkeeper moves with respect to the goal posts as well.

- **Goal Post Physics:-** The velocity has been given to goal posts and they move horizontally in the MOVE_POST mode. And, as they move we have to score the game. We have implemented a new function namely updateGoalPostPosition() in order to bring this functionality.
- **Rotating Message box:-** We updated the message box so that it only rotates after the 5 goal game is completed.
- **Detecting Goal Score:** When the goal posts are moving it is difficult to detect and store the goal scored, But we have implemented that as well.

ELEMENTS OF GAME

- **The Attacker**- This will consist of the shooting power and angle which the player can set. The angles can be adjusted to a range of -60° to 60° horizontally and from 0° to 50° vertically. Implemented using the Flatarrow Class, Axes Class
- **The Goalkeeper**-Similar to a real-life goalkeeper, it moves to defend the shot. The hands of the goalkeeper move continuously. The goalkeeper can automatically or can be made to move as per the category of the game. Implemented Using Defender Class.
- **Camera angles**- The player can change the camera angles for the penalty using the mouse so as to view the shot in different angles.Implemented using Camera Class.
- **The Ball-** The ball has its own set of rules defined for moving when kicked by the attacker and when colliding with a defender or the pole. Implemented Using sphere class
- **Texture and Font** -These contain the required functionalities for loading textures with transparency. Further we have created a custom font system, which uses transparent textures to use them as fonts, which can be used in 3D and can be resized, unlike the Raster Fonts, and also have a good width and thus can be easily seen, unlike the Stroke Fonts.

These classes used for their implementation are as given below(all classes are implemented in C++)

- **Axes** -A helper class for modularity of values of quantities along the 3D axes.
- **Camera** - A class that contains information about the camera: Horizontal and Vertical Angles and distance from the point to be seen.
- **Defender** -Contains the physics of the defender as well the drawing methods.

- **Sphere**-Contains the physics of the defender as well the drawing methods.
- **FlatArrow**- The arrow used for setting the shot angle – contains the angles and drawing methods.
- **Physical State**- The helper class that contains the physical state – velocity, acceleration and position – of any object, like the defender and the ball.

CATEGORIES OF GAME

As discussed in the introduction, this game has 2 categories. Interactive Category(Play with Computer) and 2-Player Category(Play with Human).

- **Interactive Category** further has 4 Levels-

  These Levels are

  1. Easy Level- Goalkeeper is the slowest in this level and hence scoring goals is very easy.
  2. Medium Level-This level is slightly more difficult as the goalkeeper is quicker in this level.
  3. Hard Level-This level is the most difficult as the goalkeeper is quickest in this level.
  4. MoveGoalPost- This level is a special level in which the GoalPost also displaces from its position(so does goalkeeper). This level makes the game even more interesting.

  In these levels, the goalkeeper is interactive i.e. it is not controlled by the user. It moves by analyzing the direction of the ball. If the ball is coming towards left, the goalkeeper will move towards the left.If the ball is towards right, the goalkeeper will move towards right.The Goalkeeper also jumps in vertical direction.

- **2-Player Category -** In this category, there is a competition between 2 players.One is attacker and the other is goalkeeper.The attacker has the same role and powers as in interactive category.

  The goalkeeper in the 2-player Category can move in horizontal direction, jump in vertical direction and can save the ball from getting into the goal post. The goalkeeper in this level too can deflect the ball on contact.

WORKING OF GAME

In this game, the player can get 5 chances after which the game ends and the result is printed. The goals and misses are shown on the top in the game screen. The game progresses as a series of transitions of user-defined modes.

The various modes used in this game are as below.

- **HELP MODE-** The first instruction page that pops up when you start the game is in this mode.
- **CHOOSE MODE-** This mode is used for implementation of the page which gives options of play with computer/ play with human in the game. It also gives the options of selecting the levels.
- **ADJUSTING MODE -** The initial screen after quitting the instructions is internally called adjusting mode and is used for just looking around in the game with the mouse
- **AIMING MODE-** The player can change the angle of the shot (same as the angle of the blue arrow) using the arrow keys on the keyboard
- **POWERING_ACC MODE**- In this Mode you have to set the spin by holding down the 'p' key
- **POWERING_IDLE-** This is a dummy mode in order to separate the spin and initial velocity.
- **POWERING MODE-** Herein, the player can press and hold the spacebar for the desired power which can be seen using the arrow on the power meter and then release it at the desired power level
- **SHOOTING MODE-** Internal to the game, the shooting mode is the actual mode where we can see the shot in action.

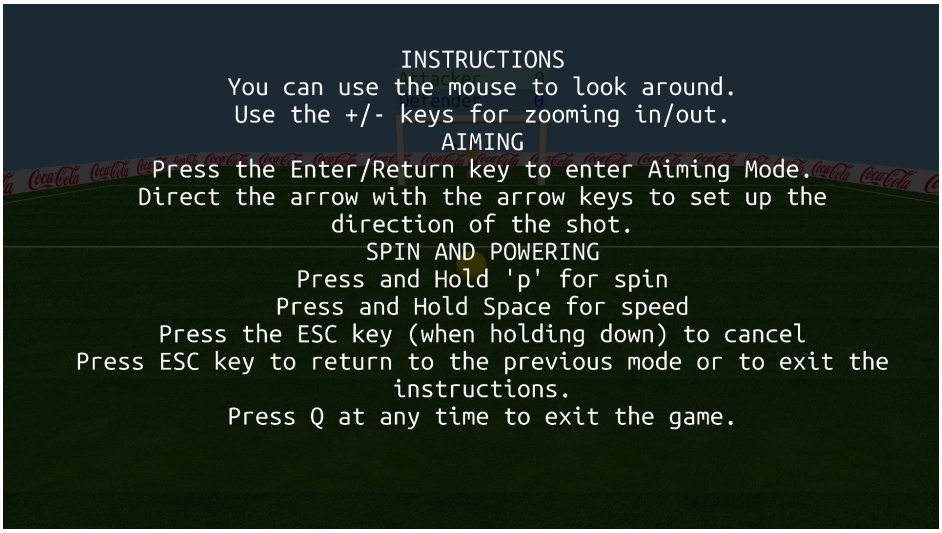
The Sequence of transitions of mode as we progress in the game are

HELP->CHOOSE->ADJUSTING->AIMING->POWERING->SHOOTING->ADJUSTING/CHOOSE

Note that the last mode can be ADJUSTING/CHOOSE. The last mode is ADJUSTING if 5 chances are not completed else the last mode is CHOOSE and the game is over.

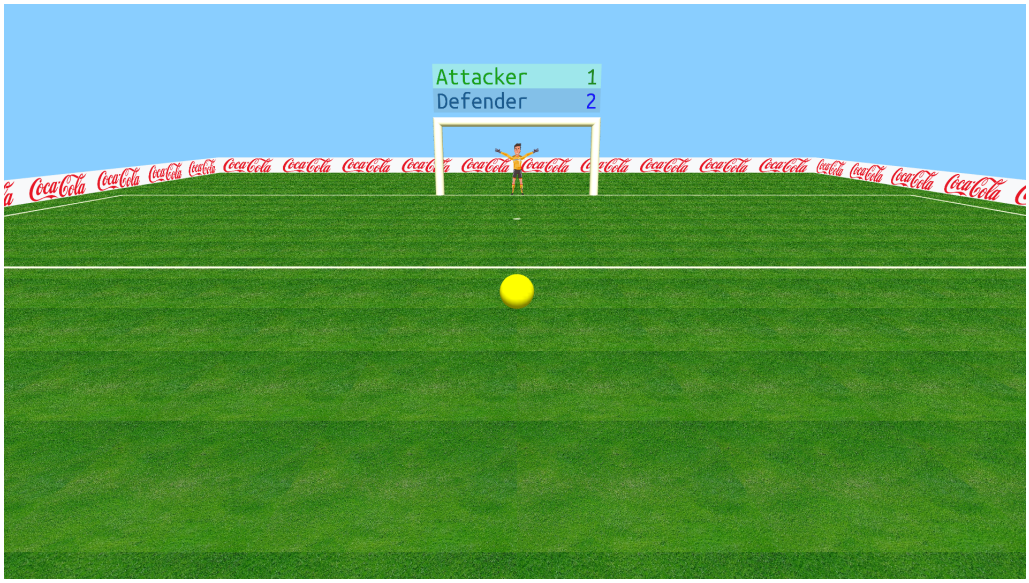
OUTPUTS

INSTRUCTION PAGE

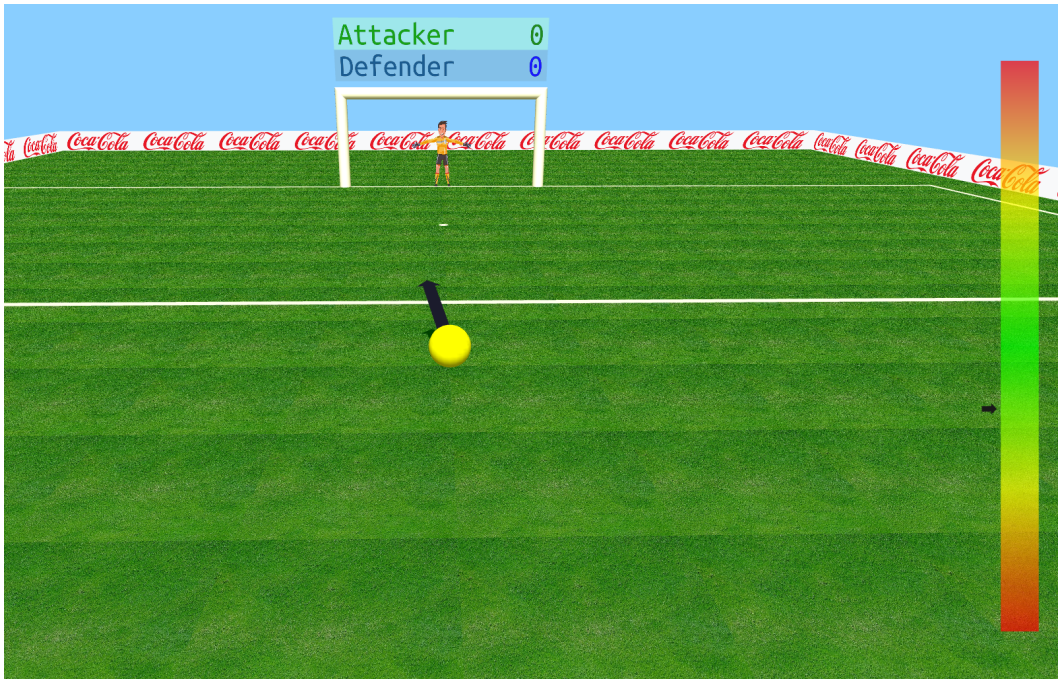


LEVEL CHOOSE PAGE

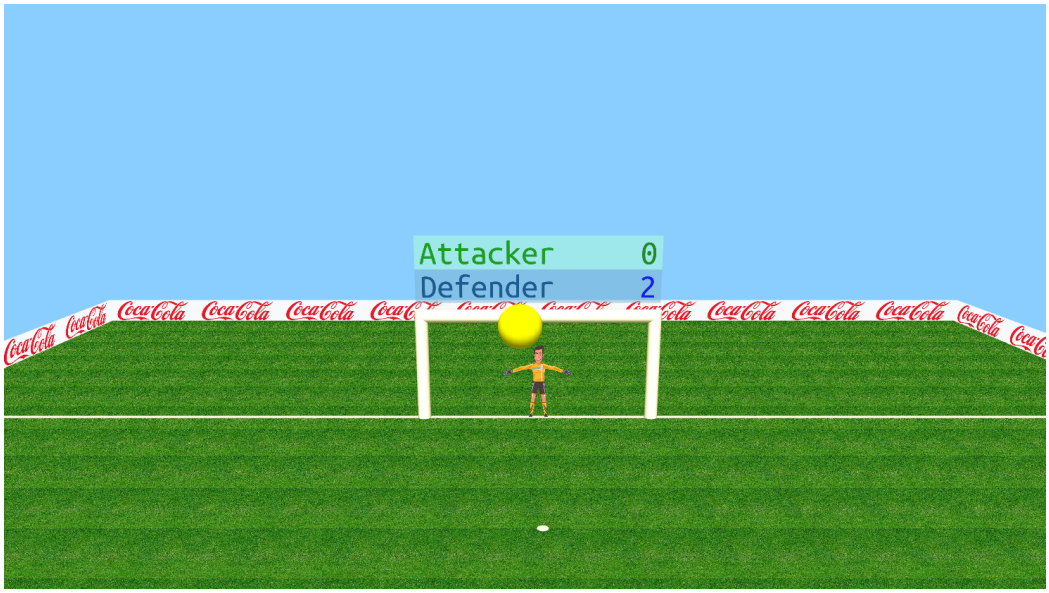
INTERFACE



BALL ARROWS AND POWER METER



BALL MOTION(including collision with goal post)



SCORE BOARD

GOAL OR MISS MESSAGE

FINAL RESULTS WITH ROTATING MESSAGE(after 5 Tries)