# CS 152 PROJECT REPORT

## Team:


1. **Mayank Meghwanshi (110050012)**
2. **Shubham Mehta (110050013)**
3. **Vivek Atulkar (110050039)**


## Problem description

1. We have made single-player (A.I.) and multiplayer
   9 ball pool. Player has to pot object balls according to their numbers using cue stick and cue ball.
2. The power with which cue stick hits the cue ball is decided by the distance by which mouse pointer is dragged.
3. After starting of the motion of cue ball as it comes in contact with other balls 2-D elastic collision takes place and as the balls reaches any of the 6 holes present on table it gets potted and moves to the top column made to hold potted balls.
4. If white ball is potted then opponent can place it anywhere on the line. User can also use the white ball present at the bottom of the user interface to hit cue ball at different position with cue stick.
5. CPU plays 4 types of shots on its turn and it decides the shot to be played by it using one level position play.
6. We have also made a side window which displays current scores of two players and also the current player.

## Program Design:

We have made following files:

1. **Ball-class**
   Each ball is represented as the object of this class. It includes all the properties of ball like position, velocity, radius, image, position and velocity update functions with time. Conversion of Cartesian coordinates to polar coordinates.

2. **Collision**
   It includes collision detection function and function which sets the velocity and position of the ball. In this class we have used various functions of mathematics vectors and collision concepts of physics.
3. **Table collision**
   It checks whether the ball is going towards the table wall and detects if ball is about to hit the wall and sets the after collision velocity. It also includes the modified function of deviated collision when rotating white ball hits the table wall.
4. **Potted**
   It checks whether the ball is in the area given for the hole and pots it if it's there. After getting potted the ball is transferred to the top column.
5. **Stick class**
   It include various characteristics of stick class like image, its position and function which controls its rotation with movement of mouse pointer
   It also includes the function which sets velocity of cue ball in the direction of the stick according to the distance by which stick has been dragged.
6. **Play-game**
   It includes player class of which both the players are object having scores, names and a Boolean value as their characteristic. If user is the player then Boolean value is true and all the mouse functions are active
   While if CPU is player then a function is called this generates a shot and accordingly gives direction and magnitude of velocity to the cue ball.
   It also includes game class which takes two players, continuously updates the score of both the players, functions which declares winning and losing of player. It also sends current score to function show-scores in graphics3.ss file and displays it on the score window.
7. **Direct-shot**
   It includes functions for cpu to play direct shot to lowest numbered ball for this it first checks which of the holes are possible, checks whether path to the ball to hall and ball to cue ball is clear, determines where cue ball should hit the object ball so that it goes directly in hole.

8. **Rebound-cue-ball-shot**

   It includes functions for shot in which cue ball 1st hits the wall and then hits the lowest no. object ball for it checks whether path is clear and then gives corresponding list of shot (velocity of cue ball) which are possible.

9. **Rebound-lowest-ball-shot**

   It includes function for shot in which cue 1st hits the ball and then ball hits the wall before getting potted. It also checks for obstacle in path and gives corresponding list of possible shots.

10. **Indirect shot**

    It includes function for shot in which cue 1st hits lowest no. ball which then hits some other ball to pot it. In it preference is given to pot 9th no. ball. It also checks for obstacle in path and gives corresponding list of possible shots.

11. **Pseudo shot**

    It includes function for position play for which we play various possible pseudo shot then grade them according to number direct shot generated if they are actually played. The shot which generates maximum no. of direct shot after it has been played is finally chosen and accordingly cue balls velocity is decided.

12. **No-shot possible case**

    If it happens then as we impart maximum velocity to the cue ball in direction of lowest no. ball's centre.

13. **Graphics**

    It includes two files, one for opening starting window at the start and other one for score window which opens when gameplay starts. It provides GUI at starting window. It includes functions which displays names and updated scores of players on score window.

14. **Game**

    It's the main file of the game. It includes all the libraries and files in the game. It has some list of positions for holes, walls etc. It has table images and ball vector. All global Boolean values are initialised. There are functions for single player and double player game initialisation. There is use of big-bang which launches game calls all functions of on-tick, on-redraw, mouse and keyboard functions and end-world conditions.

# Modules Used

1. For user interface we used graphics/graphics library.
2. For Game playing we used 2htdp libraries which have features for world and images.

# Limitations and bugs

1. The main problem occurs when collision happens between two ticks, although we have tried to lessen this error but bit of it still persists. Because of this only sometimes image of ball gets placed over restricted area of table and our A.I. isn't perfect.
2. We have done position play for just one level; it could be done for more levels.
3. CPU shots aren't perfect; sometimes it plays weird shots instead of direct shot possible.
4. We haven't made A.I. for placing white ball after it has been potted by the user.
5. For position play it checks only for possible no. of direct shots after a shot has been played.
6. Sometimes no shot is possible then a random shot is played in which cue ball goes in direction of centre of lowest no. ball with full velocity.

# Points of Interest:

1. Collision detection
2. Elastic collision between balls and ball and wall
3. Rotation feature in cue ball.
4. Artificial Intelligence – CPU plays 4 type of shots as described above
5. Position play  up to one level
6. User friendly user interface.

# How to Run:

Open game.rkt in drracket and press ctrl + r or click on Run button. This will open the GUI