

Requirements

Functional Requirements

Must-haves

- Scene
 - The game must have a board & all 15 balls present on the board at the beginning of the game.
 - The entire board must be visible to the player from an orthographic top down view.
 - The cue ball, 8-ball, solid and striped balls must all have different textures in order to be distinguished from one another
 - The game must start from the break shot: the cue ball must be positioned on one side of the board, while the remaining 15 balls are positioned on the opposite side of the board and formed into a pyramid shape. The cue ball must also be centered with height of the board.
- Collisions & Ball Movement
 - The table must have a border to bound the board so that the balls could not go out of bounds. These borders must be at the edges of the board.
 - The balls and the board must be able to collide. When a ball hits the board, it must stop moving. (**Note:** This is extended in a should have by adding realistic collisions)
 - The balls must be able to collide with each other. When one ball (ball 1) hits another ball (ball 2), the other ball (ball 2) must be moved by the ball that it was hit by (ball 1).
 - The balls must be movable when a force is applied to them externally (either by the cue or by other balls)
 - When a ball collides with the predefined pocket region, it must be potted.
- Interaction
 - The player must be able to hit the cue ball with a cue during their turn by clicking and releasing the mouse button.
 - The hit of the cue must be based on the current position of the mouse in order to be able to hit the ball in different directions.
- Database & Authentication
 - The user must be able to authenticate via username and password & must do so before being able to play a game.
 - The credentials for each user must be stored in a database. This must include the username and password.
 - Upon entering a username that does not exist in the database, the game must register the new user.
- Game rules: The game must implement the proper 8-ball pool rules and follow the 8-ball pool game flow

- The game must have all 4 types of balls present and the types must be clearly distinguished by having a class per ball
 - Cue ball
 - Black ball
 - Solid balls
 - Striped balls
- The game must have local multiplayer for 2 players. The game must have a turn-based system where the next Player is able to hit the cue ball after the previous Player has made their move which did not lead to them gaining another turn.
- The game must start from the break shot, where one of the players is randomly selected to take their first turn.
- The players must have their ball type assigned after potting the first ball (**Note:** this is extended in a should have by extending the ball type assignment logic)
- Potting the black ball before potting every other ball of the Player's assigned ball type (or before it is even assigned) must result in a loss.
- Potting the black ball after potting every other ball of the Player's type must result in a win (**Note:** this is extended in a could have by letting the Player pick a pocket to pot the 8-ball into)
- After potting the cue ball, the ball must be placed at the middle of the board for the next player.

Should-haves

- Sound: The game should have sound effects that get triggered upon collision.
 - Ball vs ball collision
 - Board vs ball collision
 - Cue vs ball collision
 - Board vs pocket collision
- Ball Movement & Collisions
 - The balls should be rolling when force is applied to them, and the roll should be based on the strength of the force applied to the ball.
 - The balls should have rolling friction: the balls slow down overtime after force has been applied to them. This should happen every update (iteration) of the game loop.
 - The balls and the board should be able to collide realistically.
 - Ball vs. board: Balls must bounce off of board cushions upon collision
 - Ball vs ball: Balls should bounce off each other relative to the speed of both of the balls.
- Cue
 - The cue should be visible to the Player when they are performing the cue shot.
 - The force of the cue should be controllable by the player when hitting the cue ball within a power range of 0 to 100%. The force should be controlled by dragging the mouse.
 - When the cue force reaches 0%, no cue hit should be performed, and the shot should be cancelled.
 - The applied force to the cue should be visible to the player as a power meter that accurately shows the percentage of force applied between 0% and 100%. It should update as the Player drags the mouse while holding the left mouse button.
 - The player should be able to rotate the cue around the ball by swiping the mouse left/right. The cue should be centered around the mouse cursor's current position.
- Game rules: The game should implement additional 8-ball pool rules
 - If the first ball that the Player touches is the ball of their own type (or the 8-ball if the Player has potted all of their own pool balls), and if the Player pots the ball of their assigned type, and does not pot the cue ball, the next turn should be granted to them. Otherwise, the turn should be granted to the other Player.
 - The Player should be able to place their cue ball during the break shot in the predefined break shot region of the board, which should cover roughly 10% of the board starting from the left end of the board. The region should also cover the board completely in the vertical axis.
 - The ball assignment of the game should happen only after the break shot. Upon potting a ball during the break shot, the ball type should not be assigned to the Player immediately.

- If the player does not hit their type of ball or pots the cue ball, the other player should be able to place the cue ball anywhere within the boundaries of the board (apart from the pockets)
 - Potting the black ball together with the cue ball should always result in a loss.
- Scores
 - At the end of each Pool game, the winning & losing Player's names should be recorded via a prompt. Then, their win & loss stats should be updated in the database accordingly.
 - At the end of each game, the scores of the top 5 players should be shown on the screen in ascending order (top 1 player first).
 - The stats for each Player should be recorded in a database. The stats include win and loss count.
- Interaction
 - The player should be able to restart the game immediately after they finish the current game

Could-haves

- Menus
 - The game could start from the main menu in which they can start a new game, access the stats menu or quit the game.
 - Players could be able to forfeit the game at any moment during a match by clicking a button on the keyboard or on the UI. After doing so, the stats of both participating players could be updated, so that the forfeit is counted as a loss.
 - The game could have a help menu which shows the rules of 8-ball pool to the player. This could be accessible in the main menu or while playing the game via the click of a keyboard button.
 - The game could have a stats menu, which contains the leaderboard and user stats. This could be accessible in the main menu of the game.
- Online
 - The game could have server connection functionality to allow extending the game to support online multiplayer. This would involve the Player sending requests upon making a move, and a server sending responses to both Players to sync state.
 - The game could have an online multiplayer system which matches players on a first-come first-serve basis. The implemented server connection functionality is then utilized to sync the game across different machines so that the players could be able to play online.
 - The game could have an online tournament function which matches 4 to 16 players on a first-come first-serve basis. The players then play against each other and can advance to the next levels of the tournament, where they play against a randomly selected winner. This repeats until the final match commences, and the winner is determined.
 - Additional statistics could be stored for the Players to keep track of how many tournaments they have won.
- Personalization
 - The game could have a function for players to create a profile, which allows them to add (or choose) a profile picture, add a description about themselves and allow them to change their username.
 - The game could allow customization by allowing the Player to pick different ball, cue and board skins in their profile
- Settings
 - The game could have a settings function for controlling the rate of friction (ball slow down)
 - Players could be able to control the volume of the game (including mute) via clicking a button on the keyboard and opening a menu with a volume bar.
 - Player could be able to change their password in the main menu, which then also gets updated in the database.
- Game extensions
 - The game could have a single player practice mode where a single player can simply keep potting the balls until none remain on the board, in which case the game ends.

- The game could have a timer for the games which allows the user to input a timer value from 5 to 120 seconds, or to disable it completely.
- The game could have a rock-paper-scissors determination of who breaks first
- The game could have an 8-ball pocket selection once the player has potted all their balls. Upon potting the black ball to the wrong pocket, the player should lose, and win otherwise.
- Sound
 - The game could have music that plays in the background during the game and in all other menus.
- Visual display
 - Player could be able to visually see which balls are potted in a corner of the screen (left, right, top or bottom), and which ones are yet to be potted. This information could be visible for both players at a single time.
 - The ball trajectory could be visible to the player up to a predefined threshold with regards to unit distance relative to the game's board size (e.g. centimeters) that can be easily changed in the code
 - The game could have anti-aliasing of a fixed amount (e.g. 4 samples) to provide a nice visual display with smooth edges.
- Cue
 - The cue could be visually dragged back as the power of the cue increases, and dragged forwards as the power decreases.

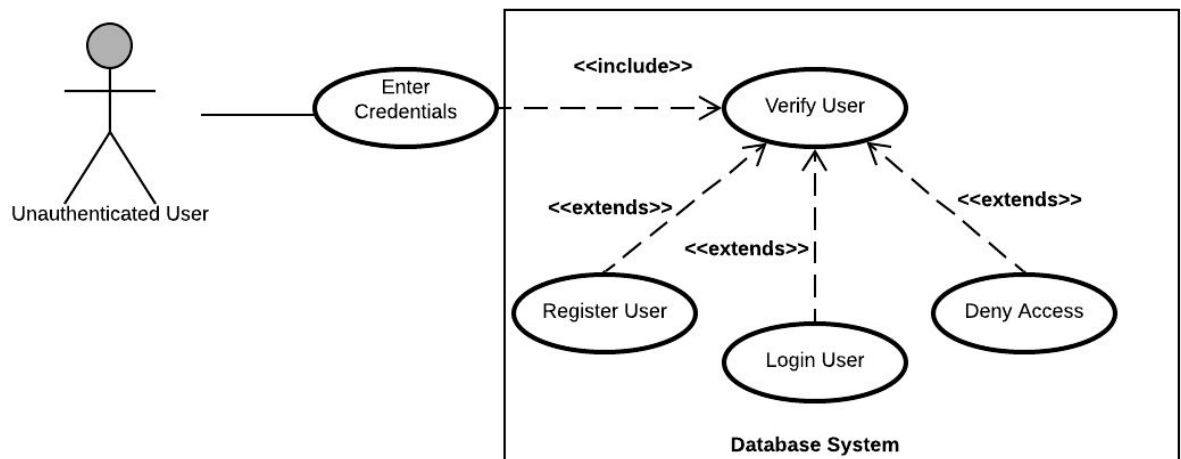
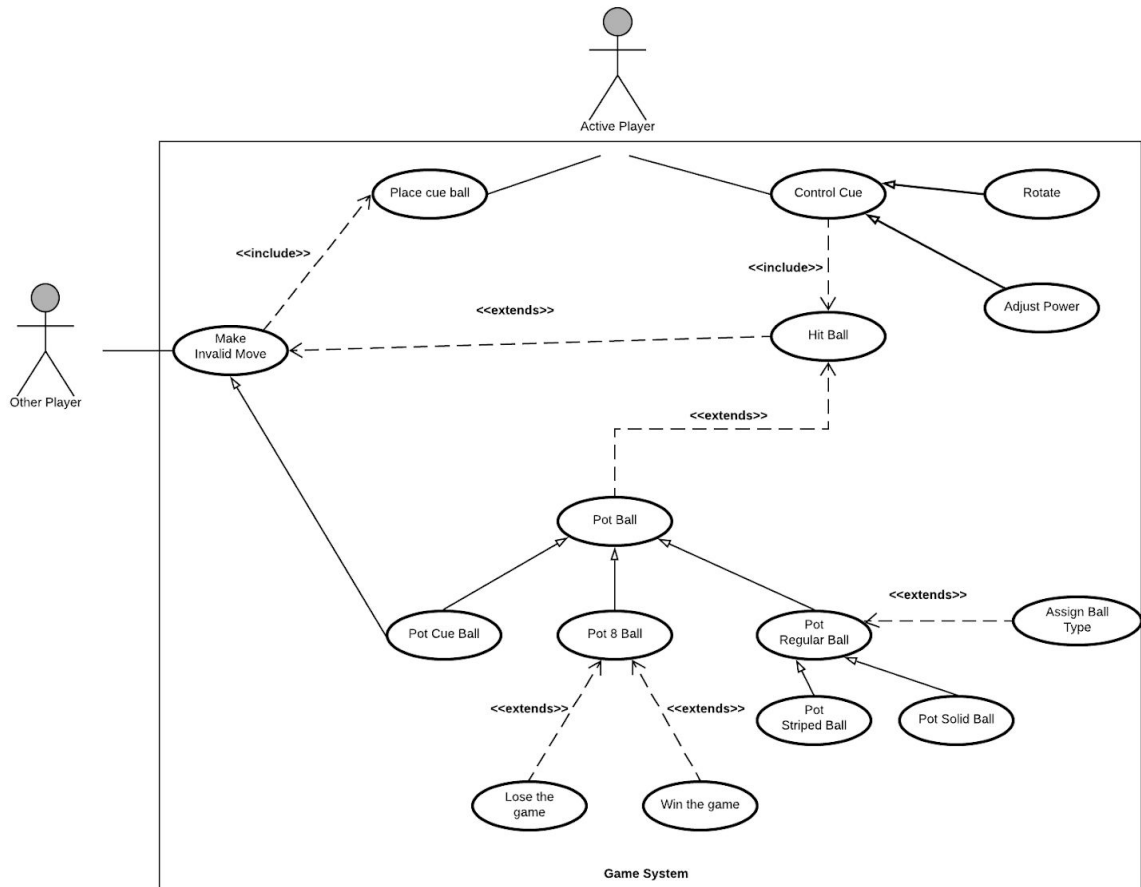
Won't-haves

- The game won't be in full 3D. The camera will not follow the movements of the balls and will not dynamically adapt to the current position.
- The game won't have an AI computer to play against, and the only possibilities will be to play against a local player or play online.
- The online multiplayer game's animations won't be fully synced, and will instead be played when receiving an HTTP request.

Non-Functional Requirements

- The game must be programmed in Java
- The LibGDX library should be used for the game
- The game must be compatible (playable) with the Windows operating system.
- The game should use a SQL database & JDBC to store user credentials and user scores
- Every (public) method should have a clear & descriptive JavaDoc that documents all the parameters, the return type (if applicable) and a short description of what the method does.
- Prepared statements must be used for SQL queries to avoid code-injection vulnerabilities
- The game's interface must be intuitive and easy to use. A new player should have navigated from the main menu to the game and started their first move in roughly 20 seconds. The player should be able to perform their first cue shot within around 30 seconds.
- The game must be framerate independent in the sense that every update of the game (movement, animation) does not get affected by the current framerate.
- The response time when making a move in local games must be around 5ms. The response time when making a move in online games (if the extension is made) must be at most around 50ms when testing the responsiveness on connections within localhost.
- The project must be developed in a Test Driven Development fashion and should have 80% meaningful branch coverage (meaningful - methods properly tested and not just executed)
- The classes in the codebase should not have more than 30 methods. Each class should be at most 200 lines long.
- The maximum cyclomatic complexity of a method in the codebase should not exceed a complexity of 10.
- Scrum methodology must be applied while developing the project. The sprints should be 2 weeks long, and a retrospective must be written at the end of each Sprint.
- User data storage should not include sensitive information. It should not include any address details (except country and city, possibly), payment details, first and last names (only username), date of birth or profile links.

Use Cases



Use Case: Placing the cue ball (after invalid move by other Player)

Purpose

Define the coordinates of the cue ball after an invalid move by the other Player.

Overview

After an invalid move was made by the other Player, the current Player will have the ability to place the cue ball (within bounds) at the position of their cursor. If the player tries to place the ball out of bounds the ball will not be placed and the Player will be prompted to place the ball within bounds.

Actors

Primary actor - Other Player

Primary actor - Active Player

Pre Conditions

The other Player must have made an invalid move

Post Conditions

The cue ball must be within bounds

Use Case: Controlling Cue

Purpose

Control the direction and power of the cue shot

Overview

The Player initially starts from their own turn. From that point, they are able to control the cue by both rotating it and adjusting the power of the shot.

Actors

Primary actor - Active Player

Pre Conditions

It must be the Player's turn

Post Conditions

- * The cue must be rotated based on the Player's rotation amount
- * The cue's power must be adjusted based on the Player's setting

Use Case: Hitting cue ball

Purpose

Allows the Player to be able to make a cue shot

Overview

The Player initially starts by having adjusted the power and rotation of the cue. Then, the Player can hit the cue ball with the cue. If the cue power is not 0%, the cue ball should be hit by the cue. Otherwise, no cue shot should be performed.

Actors

Primary actor - Active Player

Post Conditions

The cue ball's force & speed must be set accordingly to the cue hit performed (if performed)

Use Case: Potting pool balls

Purpose

Allows the Player to be able to pot pool balls after making a cue shot

Overview

After the Player makes the cue shot, the cue ball might touch other pool balls which results in them being moved. This might lead to the touched pool balls being potted by falling into one of the 6 pockets of the pool table. The player might pot either a striped, solid or 8 ball.

Actors

Primary actor - Active Player

Post Conditions

When a ball is potted, it must disappear from the board.

Use Case Making an invalid move

Purpose

The player makes a move that is invalid.

Overview

The Player performs a move that is considered invalid to trigger the cue ball placement use case for the other Player. Invalid moves include

- Potting the cue ball
- Touching 8-ball first with the cue ball (provided the Player has not yet potted all of their assigned balls)
- Touching pool ball of a type that is not the Player's assigned type first

Actors

The player

Use Case: Potting the 8 ball

Purpose

Allows the Player to be able to win or lose the game once the 8 ball gets potted

Overview

After the Player makes the cue shot, the cue ball might touch other pool balls which results in them being moved. This might lead to 8 ball being potted by falling into one of the 6 pockets of the pool table. If the player who potted the 8 ball already potted all balls of its assigned suit, then the player wins the game. Otherwise, the player automatically loses the game.

Actors

Primary actor - Active Player

Pre Conditions

The player has potted all balls of its assigned suit.

Post Conditions

When the 8 ball is potted, the player wins the game.

-- OTHERWISE --

When the 8 ball is potted, the player loses the game.

Use Case: Ball type assignment

Purpose

Allows the Player to be able to have a target ball type

Overview

The first Player that legally pots a ball after the break shot is assigned the ball type of the potted ball. If the Player pots both types of balls, they are allowed to choose the preferred ball type. The goal for the Player is to pot all this type of balls and after that the 8 ball.

Actors

Primary actor - Active Player

Pre Conditions

The shot was legal and not the break shot.

No Player is assigned a ball type.

Post Conditions

Both Players have a ball type assigned.

Use Case: User Login

Purpose

Allows an unauthenticated user to login to the system.

Overview

The use case begins from the user being unauthenticated. Then, the user enters their credentials. If the user existed in the database, they are logged in if the password is correct. Provided that the password is incorrect, the user is denied access.

Actors

Primary actor - Unauthenticated user

Pre Conditions

User is unauthenticated

User exists in the database

Post Conditions

User should be authenticated if they entered the correct credentials.

If the entered credentials were incorrect, the user should be denied access.