Graphics Programming

Report

This snooker game was developed using p5.js for rendering and Matter.js for 2D physics simulation. I decided to implement mouse input for aiming and pressing space for shooting, and shooting is only possible after all the balls have stopped moving.

The aiming system begins when the user clicks and holds the mouse. A line appears, representing the cue's direction and power. The line grows as the player drags the mouse away from the ball. This gives a visual cue showing where the shot will go. The direction is determined by the vector from the ball to the drag position, and power is based on drag distance. When the mouse is released, the aiming phase ends, and the player presses space to shoot.

Event handlers are used to detect collisions, and OOP practices are used. There are separate classes for Ball, Cue, and Table, with their own members and methods, where their behavior and appearance is defined. Each class is in its own .js file. Aiming and shooting is handled in the Cue class, table design, cushions, and pockets are defined and drawn in the Table class, and ball properties defined in the Ball class.

In the sketch.js file, matter.js aliases are defined, and all the global variables are declared. In the setup functions the canvas is created, dimension related variables are calculated, and all the setup functions are called. The event handling functions are called as well, and the rewind button for the extension is created. The draw function calls all the other draw methods, including some helpers that display messages and instruction. Input related functions are also defined in sketch.js, as well as functions that set up the balls in the three different ways. The positions for the random balls are decided using the p5.js random function, using the table dimensions and offsetting them by the ball diameter to ensure they are placed within the table bounds. Sketch.js also includes all the helper functions for collision detection, velocity control, state saving, and checking if balls are stopped, or in a pocket.

Two game extensions were added to expand gameplay and provide unique features:

1. Rewind Button with Cooldown

A clickable button allows the player to rewind to the previous ball positions. This is useful if a foul occurs or a shot goes wrong. It gives players a second chance without restarting the game. To keep it balanced, the button has a visible 10-second cooldown, preventing constant use. This feature is a fun way for the user to correct mistakes, and it's an interesting addition to the game, since it's something that can only work in a video game and not real life.

2. Lucky Pockets extension

In the beginning of the game and after a ball is pocketed, two pockets are randomly assigned to be lucky pockets. These are not known to the player, and when a ball falls in the pocket, the player gets x5 points. Confetti appears over the pockets after scoring. This is a fun feature because the user can unexpectedly earn a bonus, but does not know how, so there is no strategy involved, just luck. Together, these extensions add some fun features that enhance the gameplay, without modifying the core mechanics. The user can decide whether they want to use the rewind button or not, and the lucky pockets don't change the flow of the game, they just add an extra layer of surprise and luck. Overall, the app is a simple implementation of snooker in p5.js, with accurate physics, implementing collision detection and handling, scoring, and interesting visuals with some 3D-like elements, that can be replayed indefinitely.