Graphics Programming

SNOOKER DIMENSIONS AND RULES

🎱 Snooker App – High-Level Project Plan

Phase 1: Understand & Prepare

- [x] ~~**Read Requirements Thoroughly**~~
  - [x] ~~Re-read the assignment PDF (done!)~~
  - [x] ~~Skim the Snooker Wikipedia page for table layout, ball colors, positioning.~~
- [x] ~~**Set Up Development Environment**~~
  - [x] ~~Install/verify **p5.js** and **matter.js** libraries.~~
  - [x] ~~Set up a clean project folder with an `index.html` and `sketch.js`.~~

Phase 2: Build Core Structure

1. **Define Variables & Data Structures**
   - Variables for table dimensions, pockets, lines.
   - Arrays for red balls & colored balls.
   - Object for the cue.
   - Physics world setup with matter.js.
2. **Draw Static Table**
   - Draw the table rectangle with the 2:1 ratio.
   - Add pockets with correct size (1.5× ball diameter).
   - Draw lines: D zone, baulk line, spots for colored balls.

✅ Phase 3: Implement Ball Mechanics

1. **Implement 3 Ball Modes (key '1', '2', '3')**
   - Mode 1: Proper starting position (manual placement).
   - Mode 2: Random reds + colored balls.
   - Mode 3: Random reds only, colored balls on correct spots.
2. **Add Physics Properties**
   - Apply restitution and friction to balls.
   - Add physics for cushions with correct restitution.
3. **Draw Balls & Update Positions**
   - Use matter.js bodies for each ball.
   - Draw them each frame based on physics engine.

✅ Phase 4: Cue Implementation

1. **Cue Control & Physics**
   - Draw cue stick.
   - Use **mouse + key** interaction for aiming and striking.
   - Apply force on the cue ball within speed limits.
   - Insert cue ball in D zone interactively (not pre-placed).
2. **Collision Handling**
   - Detect collisions: cue ball hitting red, colored ball, or cushion.
   - Show prompts in console/log or overlay.

✅ Phase 5: Gameplay Logic

1. **Pocketing & Rules**

- Remove red ball from array/world when potted.
- Re-spot colored balls.

- Return cue ball to D if potted.
- Warn player if two colored balls pocketed consecutively.

✅ Phase 6: Testing & Debugging

1. **Test All Modes & Edge Cases**

- Test 3 modes.
- Test cue ball placement.
- Test collisions and prompts.
- Verify correct ball removal and re-spotting.

✅ Phase 7: Extension & Innovation

1. **Design & Implement Extension Feature**

- Brainstorm unique ideas (e.g., slow-motion replay, different camera views, AI opponent cue shots).
- Implement and test thoroughly.

✅ Phase 8: Finalize Deliverables

1. **Polish Code**

- Clean up: comments, indentation, remove debug logs.
- Bundle JS files using the module's bundler tool.

1. **Write Commentary (≤500 words)**

- Explain cue design, mode logic, extension.
- Place in the main `.js` file.

1. **Record Demo Video**

- Use OBS Studio.
- Walk through:
  - 3 modes.

    - Cue ball pocketing & return.
    - Red ball removal.
    - Colored ball re-spotting.
    - Error prompts.
    - Cue ball collision detection.
    - Discuss extension.
- Keep console visible.
- Keep it under 5 mins.

1. **Submit**

- Zip all code.
- Submit video as `.mp4` and via YouTube link.
- Submit merged `.js` file (excluding libraries).

✅ Key Tools

- **Libraries:** `p5.js`, `matter.js`
- **Bundler:** Module-provided JavaScript Bundler
- **Recorder:** OBS Studio (no phone recordings!)