Title: Virtual Pool

Topic: simulating a game of pool

Genre: Sports

Team member: Jinwoo Yom

Description:

Virtual Pool is a two-player game using realistic ball collision physics to model a widely known cue sports called pool. In terms of the rules, it will follow all of the official rules of actual real-life game. Players will alternate back and forth attempting to hit all of their assigned type balls into the six pockets on the table. First player to hit all of their assigned balls and an 8 ball in the pocket wins. However, if one of the player accidently knocks the 8 ball in one of the pockets before hitting in all of his assigned balls, the other player wins. There are few other ways for a player to win or lose the game, which are all part of the official pool rules.

Topics learned in class that I will use:

- Using radians
- Custom characters
- Animations
- Collision Detection (Bounding Circle)
- Translate()
- Rotate()
- pushMatrix() & popMatrix()
- PVectors (add(), mult(), set(), sub(), heading(), mag(), normalize(), angleBetween())
- FSM
- Momentum
- Drag Force
- MouseClicked(), MouseMoved(), MouseDragged(), mouseX, mouseY
- transparency
- Particle effects

Milestones:

- Mouse control
    - o Create placeholder arts that I can use while working on the project.
    - o Create ball and stick objects
    - o Establish boundaries
    - o Using the built-in mouse control functions, I will display the pool stick on the cue ball, and apply force to it accordingly.
    - o Create six pockets
- Ball physics
    - o Use bounding circle collusion detectors.
    - o Utilize physics collusion formula into the game.

- o Use drag forces to slow the ball down
  - o Upgrade the ball objects so that it can detect accurate collusion within themselves
- Core game loop
  - o Have instruction screen
  - o Make it a two player game
  - o Use FSM to divide the game into different states
  - o Use flags
  - o Apply all of the game rules
- Art works, particle effects, display info (directions, scores, etc) and polishing codes
  - o Create better & cleaner art work
  - o Add animations
  - o Write info and add art work on the instruction screen
  - o Display scores
  - o Add in particle effects to the game
  - o Fix bugs and improve game quality
  - o Clean up codes and make them neater