**Project Checkoff List**
**6.111 Final Project: “La PC-na”**
**Matt Basile & Zareen Choudhury**

## The Commitment: Bear Necessities

- **Camera Processor:** The camera processor should track multiple IR LED “zones” on the cue stick and returns return a vector representation of cue’s position (cue_x, cue_y, cue_xhat, cue_yhat) with respect to the board’s coordinates. The position should be accurate within a fairly lenient margin of error.
- **IMU Processor:** The IMU processor should read accelerometer data from the MPU and return the magnitude of the cue’s speed. It should start listening when a player presses a button on the cue.
- **Cue Collision Checker:** The cue collision checker should take in a vector of the cue’s current position and its speed, and output a signal when a hit has occurred as well as the updated vector for the ball it collided with. The response time of this module does not have to be immediate and can have a few clock cycles of lag.
- **Virtual Collision Checker:** This module should handle 2-3 round balls colliding off one another and walls. Collisions should be elastic between each ball responding to the collision. Final directions should appear realistic.

- **Ball Position Handler:** Ball position updates should occur semi-smoothly, with some degree of skipping or lagging. Ball's should slow down at a consistent rate every time step in response to friction.
- **Game FSM**: Basic form of billiards. Players can strike any ball of their style (stripes or solids - set to player 1 or 2) in an attempt to sink it. After each strike, regardless of a player making a ball in a pocket, move to the next player's turn. A player wins after they sink all the balls of their color. The 8 ball is irrelevant and counts for neither striped or solid.

## The Goal: Average Joe (Steinmeyer)

- **Camera Processor:** The camera processor should track 3 IR LED "zones" on cue stick and return vector representation of cue's position (cue_x, cue_y, cue_xhat, cue_yhat). Cue position should be fairly accurate.
- **IMU Processor:** The IMU processor should read accelerometer data from MPU and return magnitude of cue's speed. It should start listening when a player presses a button on the cue.
- **Cue Collision Checker:** The cue collision checker should take in a vector of cue's current position and its speed, and output a signal when a hit has occurred as well as the updated vector of the ball collided with. This module should respond close to real-time.
- **Virtual Collision Checker:** There should be 5-6 round balls all colliding off one another and the walls. Collisions should appear realistic and occur as perfectly elastic collisions.
- **Ball Position Handler:** The ball positions should update smoothly and realistically, and each ball should slow down consistently with an actual table's friction. Ball movement should be reasonably realistic.
- **Pocket Checker**: A ball should disappear after rolling sufficiently close to a pocket (center of the ball is within the pocket's radius). The ball no longer appears on the table, and no balls interact with it in the future.
- **Game FSM:** The game should progress through a basic game of pool. Specific stages include:
    - Start screen
    - Alternate between each player's turn
        - Each player is assigned a ball type at the beginning of the game
        - Each player gets one cue hit per turn.
        - First player to sink all the balls of their type wins
    - Victory/Defeat Screen
        - Reset option to start screen

## The Stretch: Yoga Matts

- **IMU Processor:** The IMU processor auto-detects when cue movement, as opposed to being triggered by a button press

- **Video:** Balls cycle through a series of different sprites to create the illusion of rotation while moving.
- **Sound:** Output recorded sound of cue-ball collision to external speaker when a cue collides with a virtual ball, and output ball collision sounds when two balls collide. This would be a new module.
- **Virtual Collision Checker**: Handle a full pool game with 12-15 balls
- **Game FSM**: Additional pool rules
    - Getting a ball of the correct color in a pocket extend a player's turn
    - Hitting in the 8 ball at the correct time wins the game
    - Hitting in the 8 ball at an incorrect time loses the game
    - Getting the cue ball in resets the cue ball to a random position and moves to the next player's turn
    - Otherwise, go to the next player's turn
    - Picking pocket to sink 8-ball in
    - Reset cue ball in position of the other player's choice after a scratch