Billiards

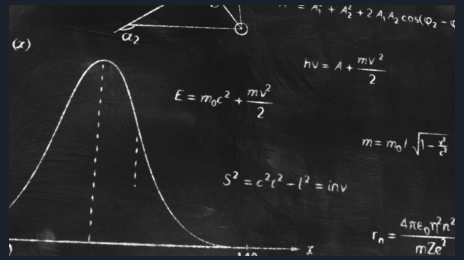
Karim Botros
Dylan Bobb, Vincent Bruzzese, Daniel Cicciarelli

Table of Contents

Project Purpose

- Physics simulation of a game of billiards

- Demonstration of physics-related formulas



Scope

- Classic game of Billiards
  - Pool Rules
  - Scratches
  - 2 Players
  - Winning and Losing
- Sandbox Mode
  - Modifiers
  - Graphs
  - No Rules

Work Breakdown

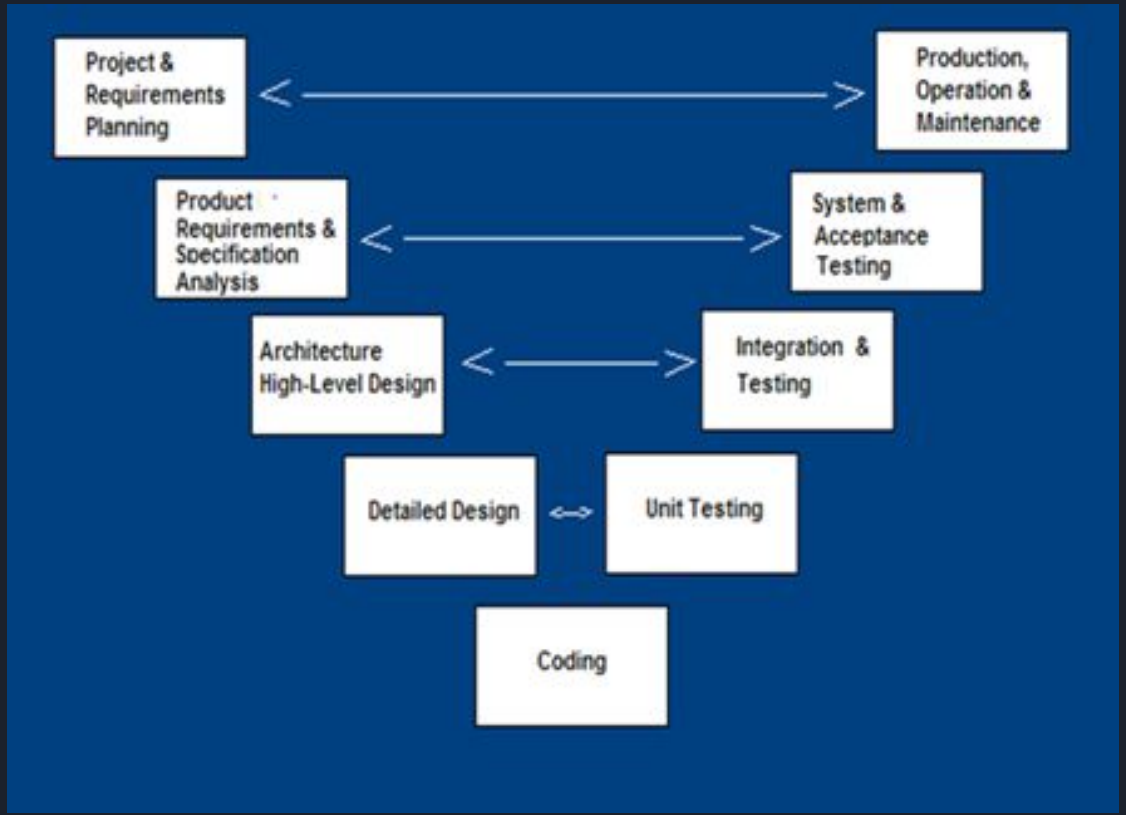
- GUI & Assets : Vincent Bruzzese 
- Physics : Karim Botros & Dylan Bobb
- Game Logic : Daniel Cicciarelli

Schedule

Elements of the project done in this order:

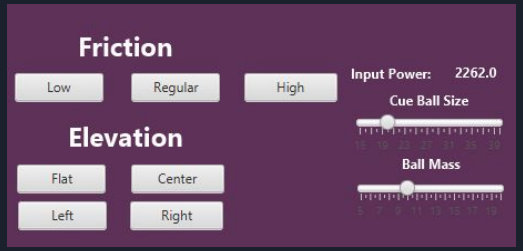
1. Table and Ball assets (GUI)
2. Game Objects and Movement Formulas (Physics)
3. Phases and Turns (Game Logic)
4. Menu (GUI)
5. Velocity Modifiers (Physics)
6. Scratches (Game Logic)
7. Sandbox Settings Buttons and Sliders (GUI)

Software Development Life Cycle

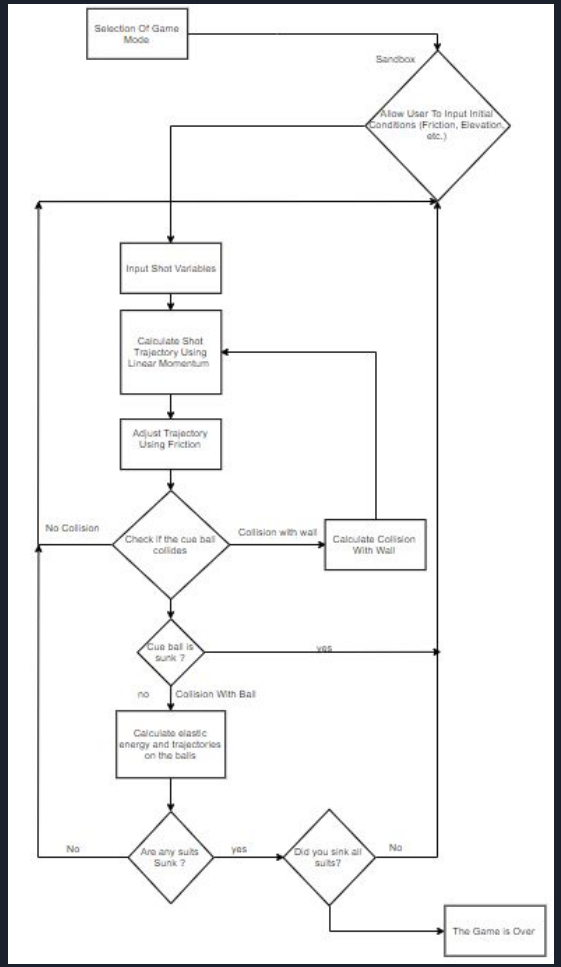


Sandbox

- No game rules being applied
  - No turns
  - No scratches
  - Cannot pot the cue ball
- Testing Grounds for all of the physics formulas
  - Low, Regular, High Friction Tables
  - Table Elevations:
    - Flat
    - Left Raised
    - Right Raised
    - Pyramidical
  - Slider for Ball Size
  - Slider for Ball Mass
  - Different combinations of all of the above
- Output shown on the table and the graphs contain:
  - Position and Velocity in the X or Y axis of:
    - Striped balls
    - Solid Balls
    - Cue Ball



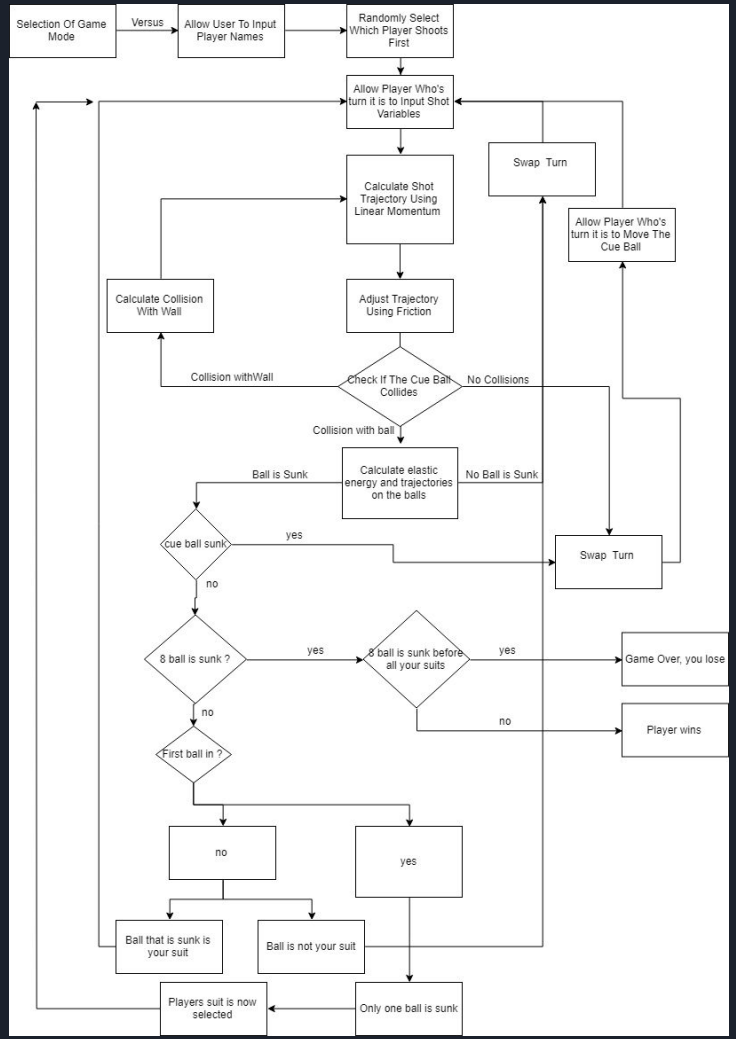
Sandbox Algorithm



8-Ball Game

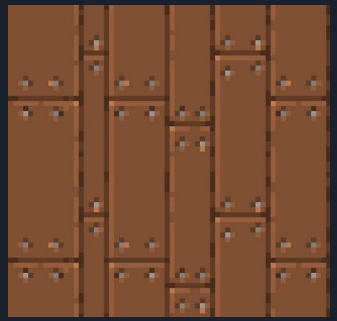
- Two-player game
- Sandbox settings are not available
- Pool game rules apply:
  - Scratching
  - Turns
  - Pocketing
  - Winning
  - Losing
- Formulas that apply:
  - Friction
  - Cushion
  - Conservation of Momentum

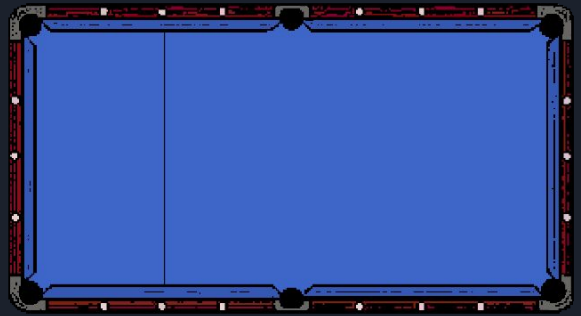
8-Ball Algorithm



GUI

- Designed and implement by Vincent
- 56 hand-designed images
- Retro - themed





Game Logic

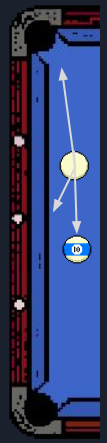
- Game Logic implemented by Daniel
- 4 Phases
  - Cue Phase
  - Action Phase
  - Check Phase
  - Place Phase
- Scratches
  - Opponent’s ball hit first
  - Potting the Cue ball
  - etc.
- Winning and Losing
  - Potting all of the balls that match your suit (Win)
  - Potting the 8 ball when not allowed to (Loss)
- Turns
  - Keeping track of balls sunk
  - Recording collisions between balls

Physics

- The first half of the physics portion was implemented by Dylan:
  - Collisions between balls
  - Collisions between balls and walls
  - Collisions between balls and pockets
  - Conservation of Linear Momentum
  - Low-velocity jittering fix
- The second half of the physics portion was implemented by Karim:
  - Friction Modifier
  - Cushioning
  - Table Elevation
  - Stopping Effect
  - Shot Input
  - High-velocity missed clipping fix

Collisions

- Collisions between Balls:
  - Center + Radius & Center + Radius

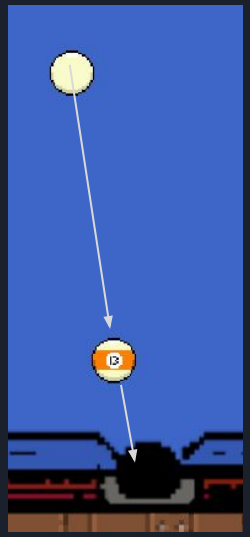


- Collisions between Balls and Walls:
  - Formula requires less computing power
  - Center + Radius & Wall Position

- Collisions between Balls and Pockets:
  - Pocket is treated as a ball

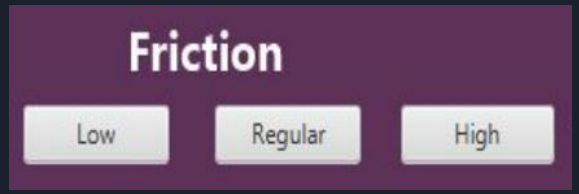
Conservation of Linear Momentum

- Formula: mA V1A = mA V2A + mB V2B ---> V1A = V2A + V2B

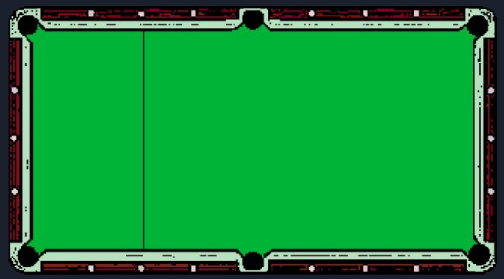


Friction

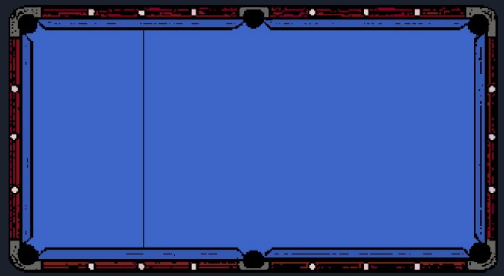
- Formula: $f = \mu\eta$
  - μ = friction coefficient
  - η = gravitational force
- Sandbox Table has 3 friction options:
  - Low
  - Regular
  - High



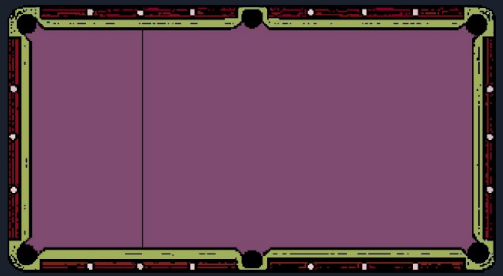
Low



Regular



High



Cushioning

- Formula: Mv' = Mv-p
  - M = Mass
  - v = Velocity
  - p = Cushioning factor

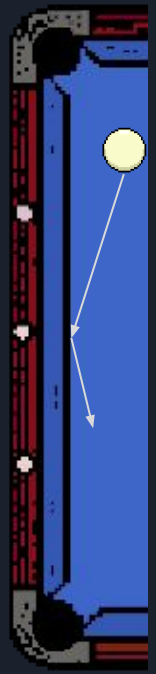
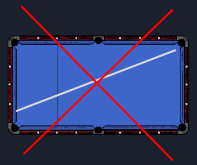
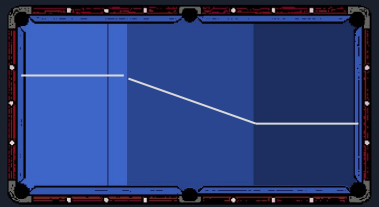


Table Elevation

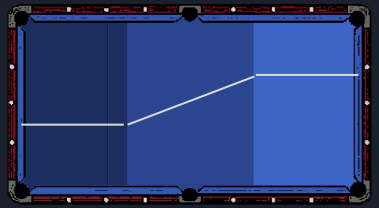
- Formula: M*G*sin(Θ)
  - M = Mass of the ball
  - G = Gravitational Force (9.8)
  - Θ = Angle of table elevation (3 Degrees)
- Issue with fully sloped tables
  - Never-ending loop of ball bouncing off of the wall



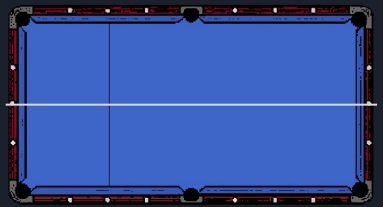
Sloped



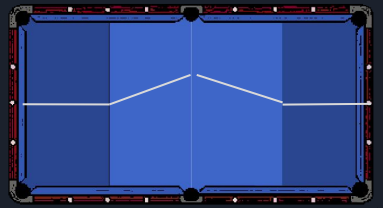
Left Raised



Right Raised



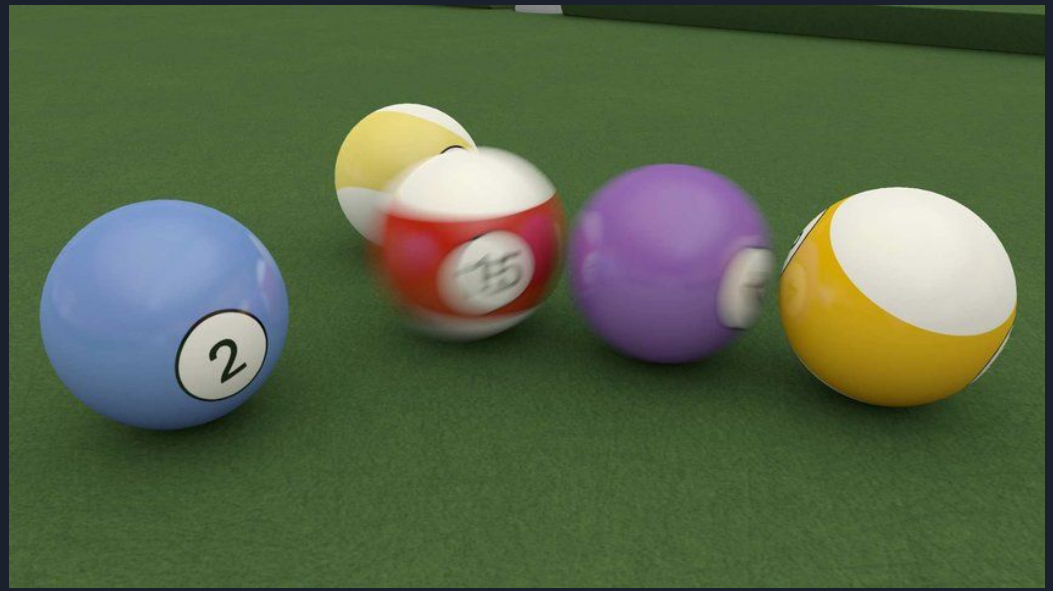
Flat



Pyramidal

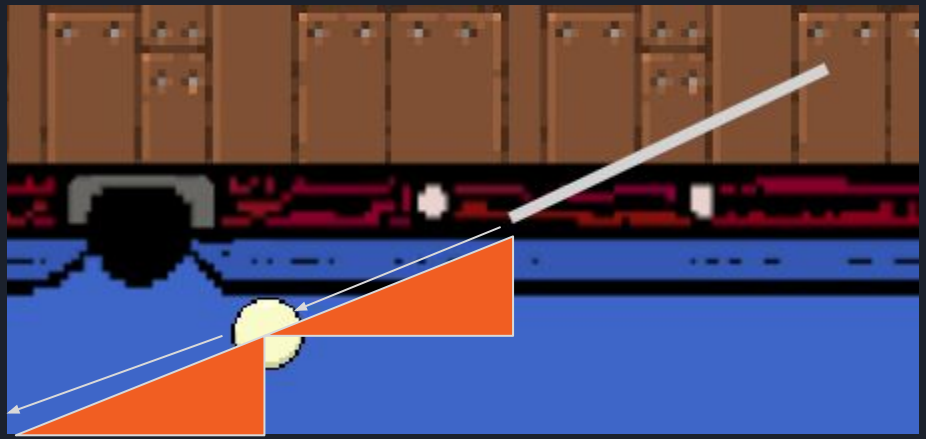
Stopping Effect

- Setting velocity to zero when velocity is very small (negligible) to avoid jittering of the balls.



Shot Input

- Trigonometry: shot vector is created along an imaginary line going from the cursor position to the cue ball to predetermine the cue ball's trajectory after a shot from the cue stick



Shot Input Limiting Bound

- Limiting the Maximum Shot input
- Too high velocities would cause clipping issues
- Square bound around the Cue Ball where the cue stick is limited within