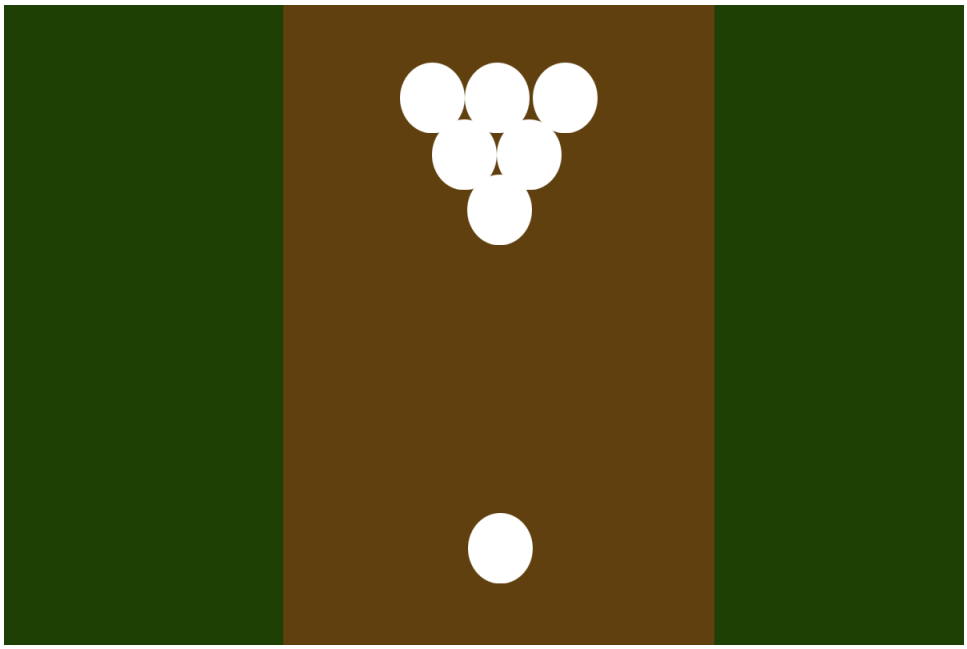


“Pool” Physics Demo GDD

Project collaborators: Ryan, Norm and Jay

Game Title:

So far, “Pool” seems to be a suitable name for our tech demo.

Target Demographic:

Anyone who wants to play pool!

Target Rating:

Unless we opt for R-rated add-on content, I’d say ‘E’ for ‘Everyone’ would be a suitable option.

Overview:

“Pool” is a tech demo for Norm, Ryan and Jay to flex their incredible math knowledge and game dev abilities. This demo will allow a player to shoot a ball, with realistic physics, and hopefully crash into the other balls on the table. We’ll talk about adding in pockets and scoring after we get the basics down.

Gameplay:

The player will use their mouse to line up a reticle somewhere on the surface of the cue ball. They can click their mouse to apply force to the ball within the area of the player’s reticle. That force sends the cue ball flying in a direction related to where the ball was hit, (and potentially how hard it was hit). The cue ball will (hopefully) hit the other balls on the table, separating and scattering them within a realistic scope based on real-world physics. Later on we can add in the usual six pockets found on a pool table, and throw in a scoring system, too!

Physics overview:

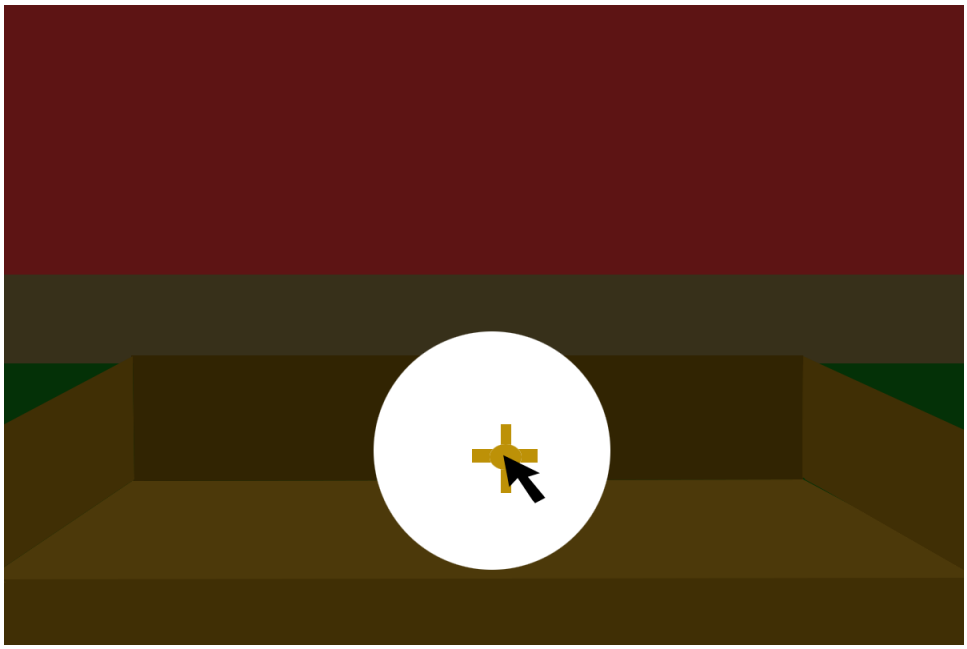
The physics in this game will be coded by hand by our lovely dev’s. Together we will research and implement true-to-life physics such as the law of motion, while utilizing algebraic properties within our calculations.

Ball physics:

Let’s break this one down piece-by-piece.

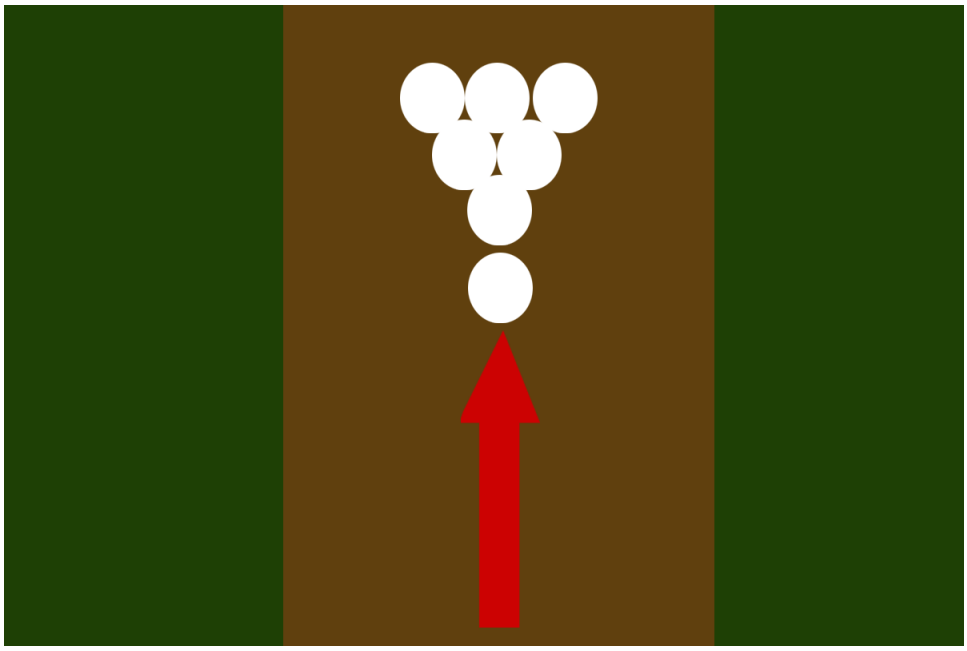
Gravity is exerting a negative value onto the y-axis of everything, while the table underneath it must exert an equal force to counteract gravity. This keeps the ball from moving anywhere but its standing position.

Player sets up reticle on ball and clicks to apply impulse to the cue ball.



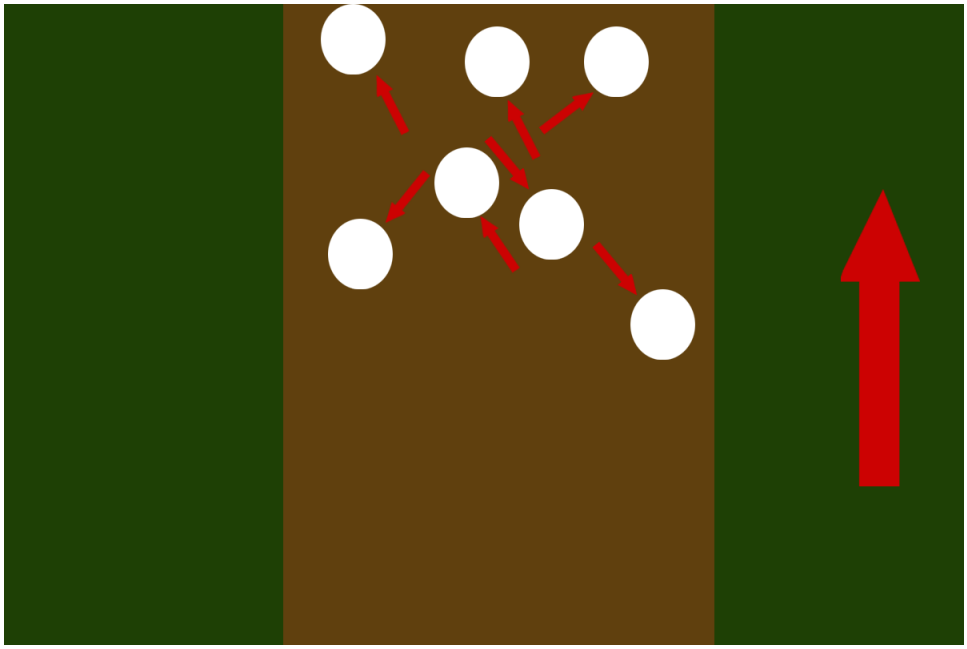
The diagram above shows a concept of the aiming screen

On-click, a raycast will get a vector co-ordinate position on the ball to apply the impulse to. The impulse applied will be a set velocity multiplied by set time. That impulse sends the cue ball down the table on a reflected line from the impulse point. The ball will be observing the conservation of momentum so that when it collides with another object, the system will know how much energy to transfer.



The diagram above shows a force sending the ball straight down the table

On collision between the any two balls, the momentum of object A will be transferred to object B and vice-versa. This means that when ball A hits ball B, ball A goes an equal and opposite distance dictated by conservation of momentum and Newton's Third Law.



This diagram shows the direction the ball was hit, and smaller arrows showing you where the balls are trying to hide. I'm kidding, those smaller arrows are showing the balls being scattered in different directions.

(Jay, don't be such a smart-ass)

If a ball it's the side rails of the pool table, considering the ball is technically hitting the mass of the table/earth and everything on it combined, the ball's energy is simply reflected on a trajectory with near identical speeds it met the collision at. (Not even kidding, after researching this specific property, I learned that the elastic properties of a pool table are meant to near perfectly reflect that momentum and force shoved into it from a pool ball)

Systems

There should be two major systems at play here, one of the balls and their respective properties, and one for everything else. Because most of the physics in a game of Pool come from the ball, (I.e where it gets hit, how hard, angles, rotational forces, etc) I want a manager to specifically oversee the retention and application of any physics regarding the ball. The second manager will overview things like gravity, the table and the initial impulse.

The Ball Manager should have methods that calculate momentum, speed, velocity, direction, and it should overlay an invisible sphere on every ball that allows us to get co-ordinate values for any ball. Those will aid the manager in dictating tangent lines for post collision effects. For the most part, the Ball Manager is just a glorified calculator that houses all the physics calculations for any pool ball on the

table, and it will update values accordingly. The Ball manager should be able to speak with the Animation Method every frame.

The second main system is the Other Manager. The Other Manager takes care of the physics calculations of everything else. Everything else being the impulse applied to the cue ball via the player, the overall gravity in the game, the friction coefficient of the table, and a method to determine how much force to apply to a ball colliding with the table's rails.

There should also be an overall Game Manager which updates animations and resets the table. It should also determine when a player has "won", when they just want to reset the table or when they want to quit.

Lastly, the Player entity should have the ability to affect the cue ball, and I would like to see the method for applying impulse to the cueball in the Player entity.

FORMULAS

Here are some of the formulas we may end up needing. Please feel free to add more as we go along

formula for gravity - (F = GM m/r squared) where G is the universal gravitational constant, M is one object, m is a second object and r is the distance between them

formula for friction - (f = uN) where u is the coefficient of friction and N is the normal force

formula for kinetic friction - (F(sub)f = u(sub)kN) where u(sub)k is the coefficient of kinetic friction and N is the normal force

formula for impulse - (F = ft) where f is force and t is time

Formula for linear momentum - (p = mv) where m is mass and v is velocity

Formula for velocity – V = distance over time