Group Member: Anthony Chen
Group Name: achen fan club
Class: AP Computer Science, period 2

8-Ball

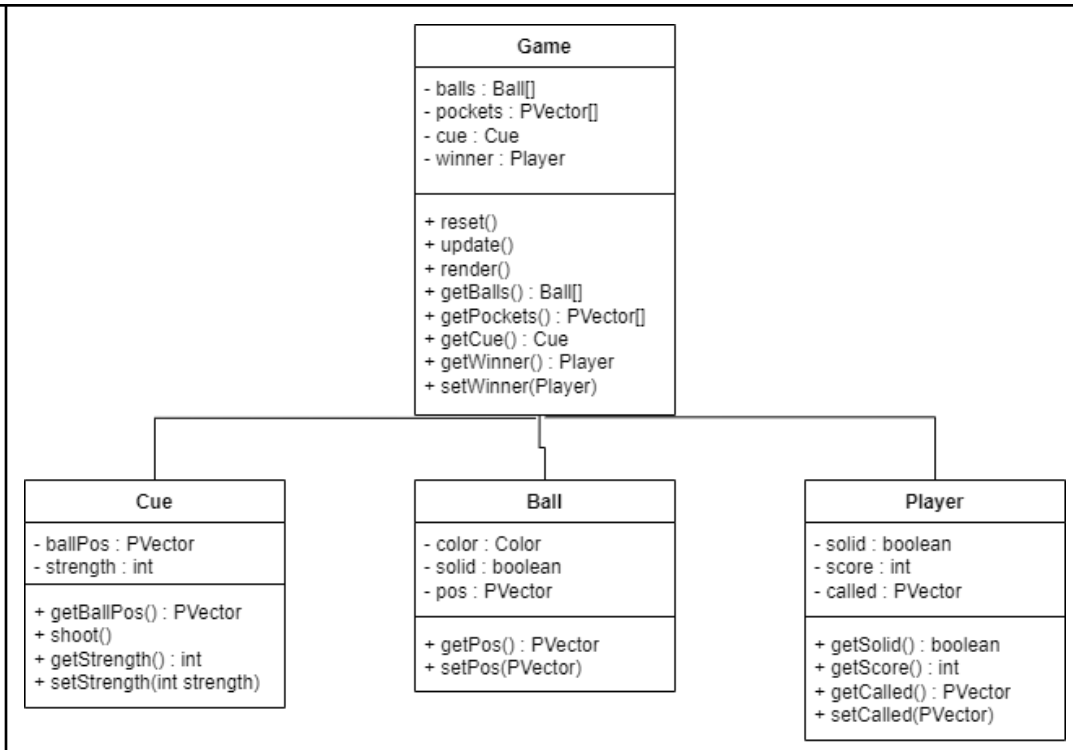
The goal of 8-Ball is to first pocket all of the assigned balls, then pocket the 8-ball in a called pocket. There are 7 balls for each player, an 8-ball, and a cue ball. The balls are placed in a triangular formation, while the cue ball is placed on the other end of the board.

During the game, players can only hit the cue ball. The cue ball can collide with other balls, causing them to move as well. The player gets another turn if they pocket one of their assigned balls first. When a player has no more assigned balls, they can choose a pocket to pocket the 8-ball in. If they succeed, they win the game.

I was inspired by the GamePigeon variant, which is the only experience in pool/8-ball that I have. I also learned about the physics of billiard balls last semester, regarding the conservation of momentum and collision. I would like to apply my knowledge in this project.

Current	● Draw the game board ○ Dark green table, with 6 pockets ● Initialize the arrays of balls and pockets ○ The balls are just in a straight line, so I could see if they render correctly. ○ The pockets are roughly in their right positions. They might be a few pixels off, but that's a small issue I might fix later.
Future	● Setup game such that balls are in their triangular starting positions ○ Calculate positions, initialize PVectors to them ● Implement a cue stick to strike the cue ball ○ Draw the stick ○ Allow the stick to rotate around the cue ball ○ Adjust strength and shoot ● Ball movements ○ Implement the physics behind it (momentum/collision, reflection, etc.)

UML


Game
- balls : Ball[]
- pockets : PVector[]
- cue : Cue
- winner : Player
+ reset()
+ update()
+ render()
+ getBalls() : Ball[]
+ getPockets() : PVector[]
+ getCue() : Cue
+ getWinner() : Player
+ setWinner(Player)
Cue
- ballPos : PVector
- strength : int
+ getBallPos() : PVector
+ shoot()
+ getStrength() : int
+ setStrength(int strength)
Ball
- color : Color
- solid : boolean
- pos : PVector
+ getPos() : PVector
+ setPos(PVector)
Player
- solid : boolean
- score : int
- called : PVector
+ getSolid() : boolean
+ getScore() : int
+ getCalled() : PVector
+ setCalled(PVector)