ALGORITHM AND PROGRAMMING
FINAL PROJECT REPORT

“HARDPOOL”

Lecturer: JUDE JOSEPH LAMUG MARTINEZ, MCS

MADE BY:

JEFF MATTHEW HADISAPUTRO - 2602118906

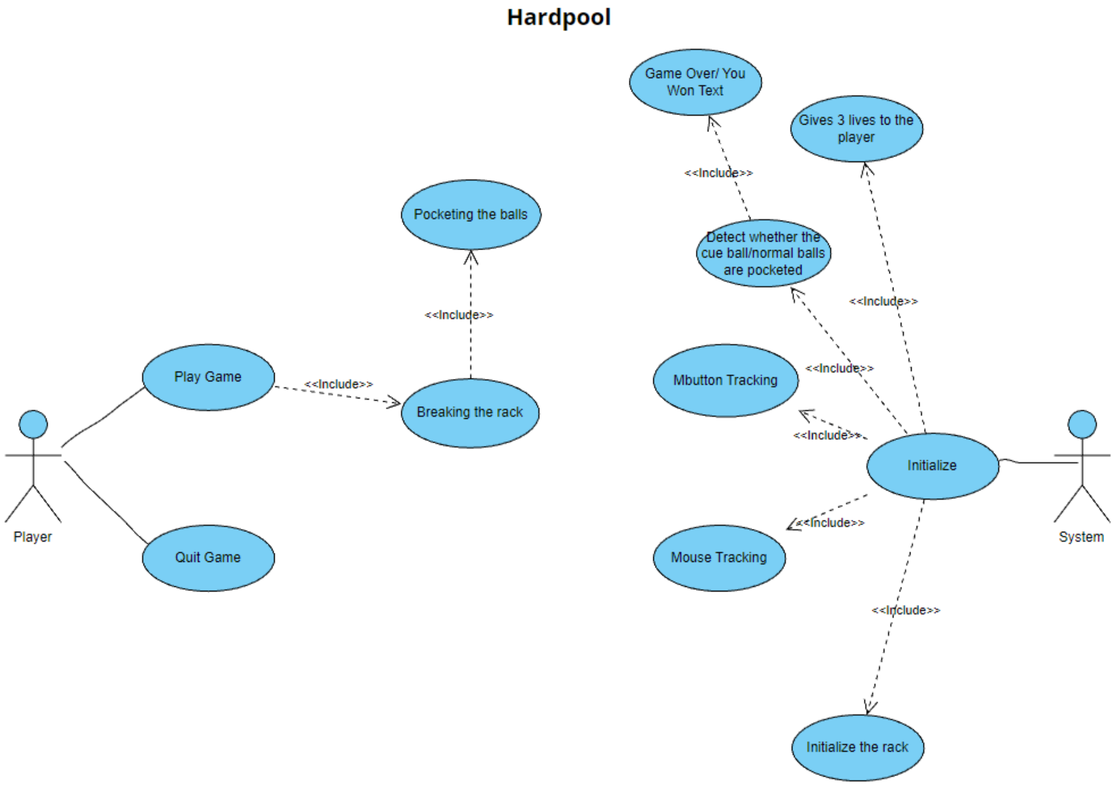
BINUS UNIVERSITY INTERNATIONAL
2023

1. Brief Description

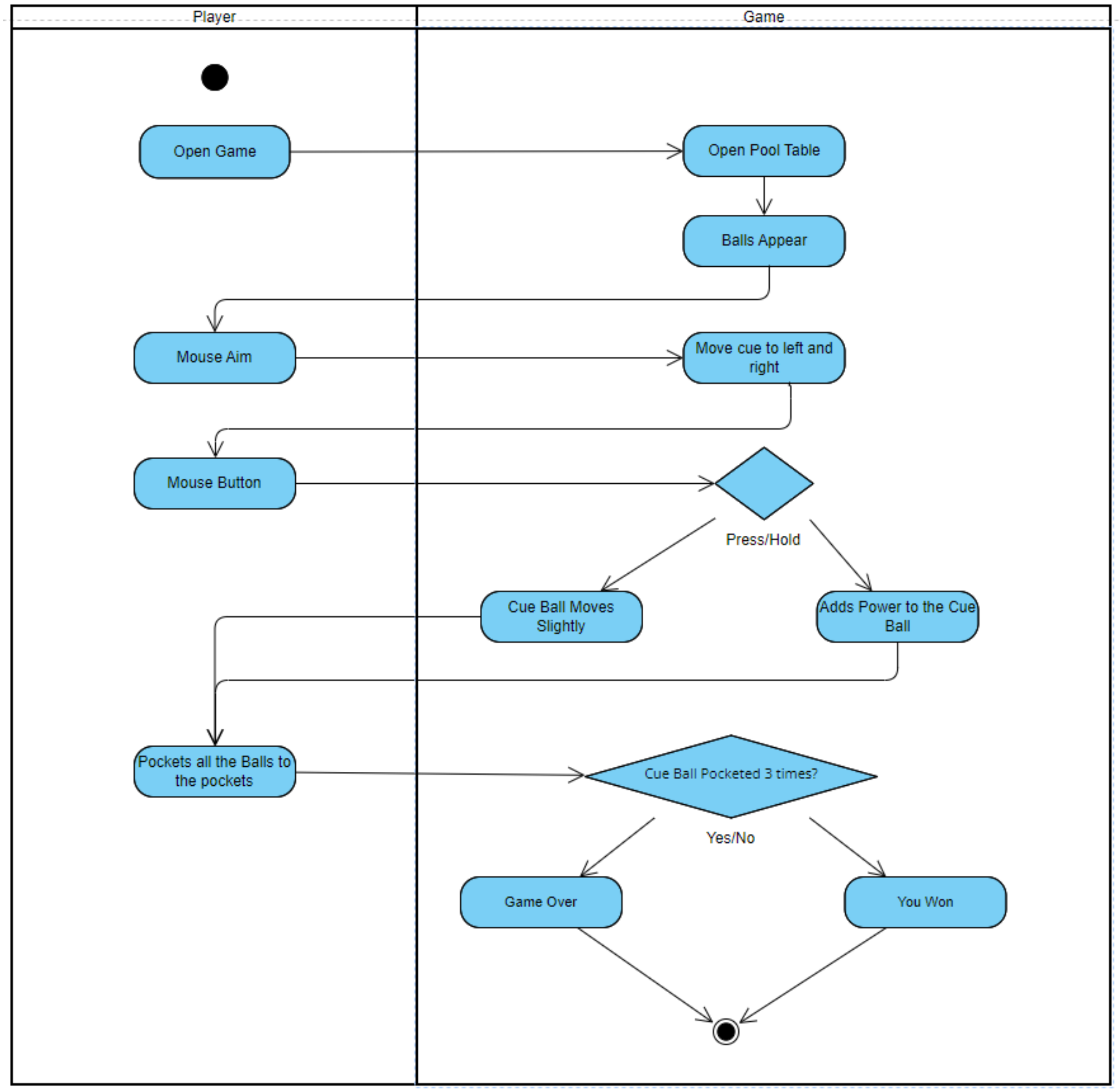
For my Algorithm & Programming final project, I made a working 2-player game of pool called “Hardpool” which in some ways are similar to normal pool like 9-ball, 8-ball and 7-ball. What makes my game different than those other pool games are the rules and the simplicity of the game itself. The rules are somewhat similar to pool such as not allowing the cue ball to be pocketed, breaking the rack at the start of the game, and hitting the balls with strictly the cue ball only. The difference is on how the players can win the game. In normal pool, the players can win the game by pocketing the number 9 ball last on a 9-ball game, pocketing the number 8 ball last on a 8-ball game and so on. On my game, all the balls need to be pocketed to win and the first person to pocket all balls is the winner.

This game is made by using the pygame and pymunk library and it CANNOT be run without those 2 libraries. Pygame is a cross-platform toolkit which is available for free and is used to create multimedia applications like video games such as “Hardpool. For "Hardpool," Pymunk is the ideal physics library to use whenever 2d stiff body physics from Python is required.

2. Use-Case Diagram



3. Activity Diagram



4. Class Diagram

CUE
+ original.image(self) : image + angle : int + image : image

+ update(self,angle)
+ draw(self,serface)

5. Libraries

- *Pygame* : The open-source pygame package can be used to create games and other multimedia applications using the Python programming language.
- *Pymunk* : An open-source module for the Python programming language. When the game requires 2d rigid body physics from Python, Pymunk is a simple-to-use package that can be employed. Ideal for use in games, demos, and simulations that require 2D physics! has the express purpose of assisting in creating games and other multimedia applications.

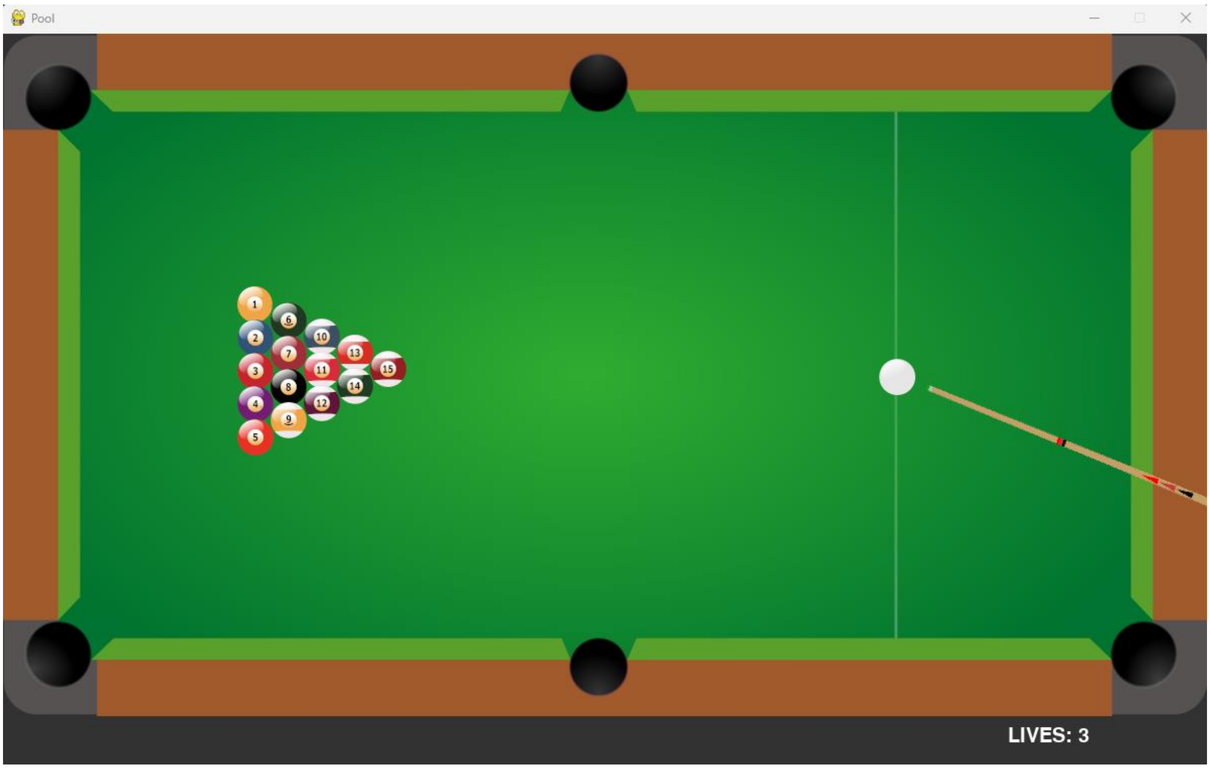
6. Modules

- *Pymunk.pygame_util* : This module includes assistance functions to facilitate rapid prototyping with Pygame and Pymunk. Intended more for prototyping and debugging than for use in a finished application. This module's methods are not optimized in any manner and have strong opinions about your coordinate system.
- *Math* : Access to the mathematical functions specified by the C standard is provided by this module. If support is needed for complex numbers, we can use the functions with the same name from the cmath package rather than these ones.

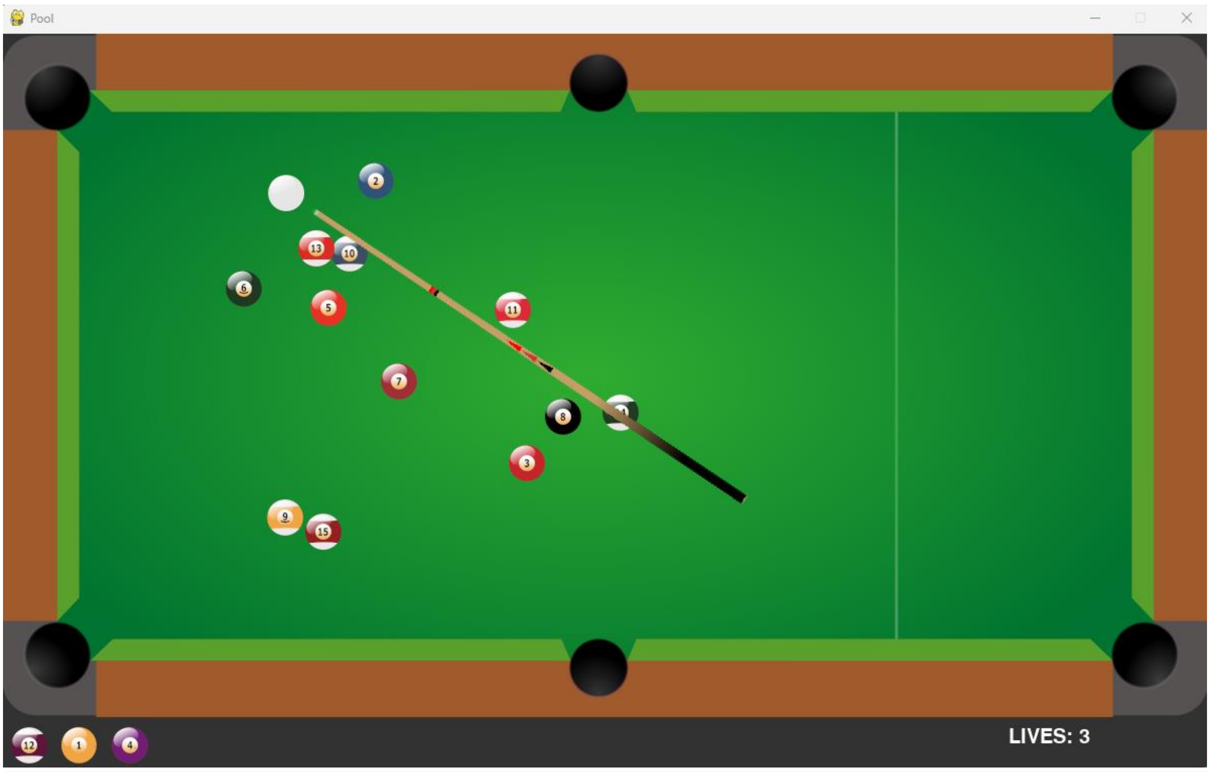
7. Game Variables

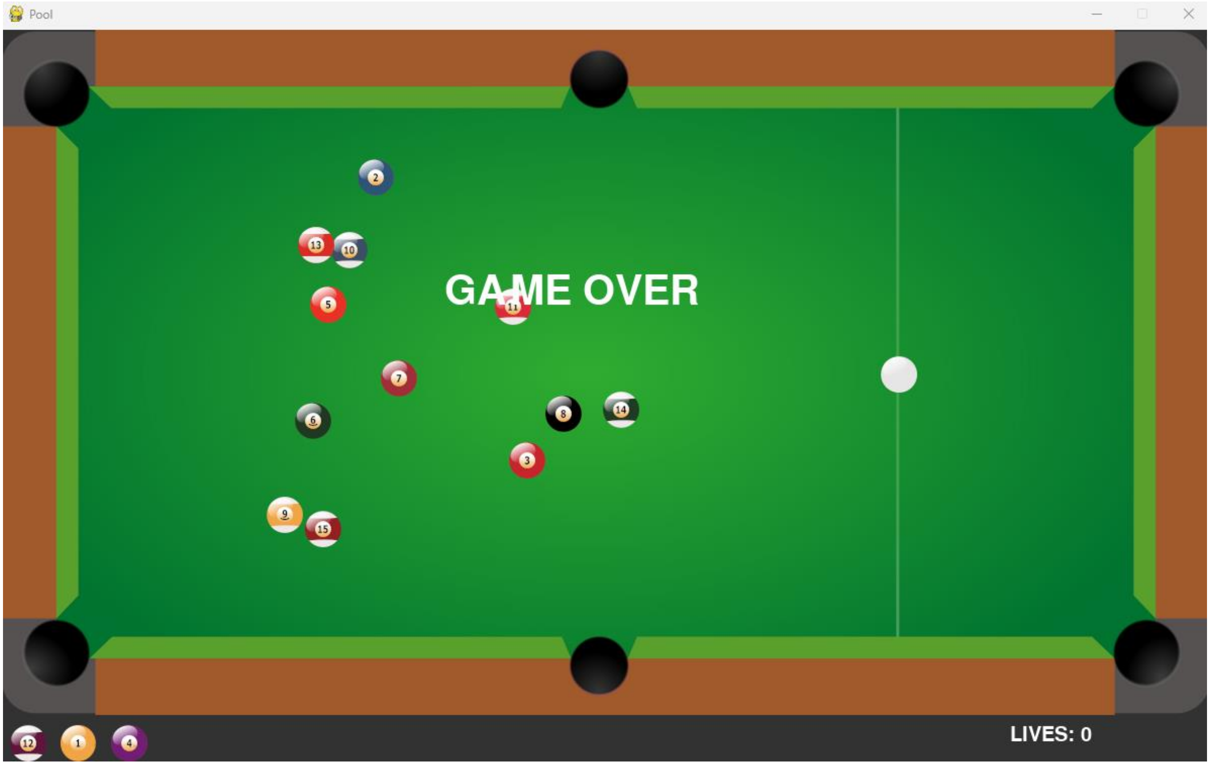
- SCREEN_WIDTH : To set the width of the entire game
- SCREEN_HEIGHT : To set the height of the entire game
- lives : to set the lives of the player
- diameter : to set the ball diameter value
- pocket_diameter : to set the pocket diameter to a certain value
- force and max_force : to set the amount of force the cue can transfer to the balls.
- cue_image, table_image, ball_image : to load the images of the game
- cue_ball_potted, taking_shot, powering_up, potted_balls : to make sure the game runs according to the rules of the game.
- pivot : to add friction to the table
- pos : Ball position
- pockets : pockets of the table
- cushions : for the ball to bounce off of
- power_bar : to add the amount of power for the cue

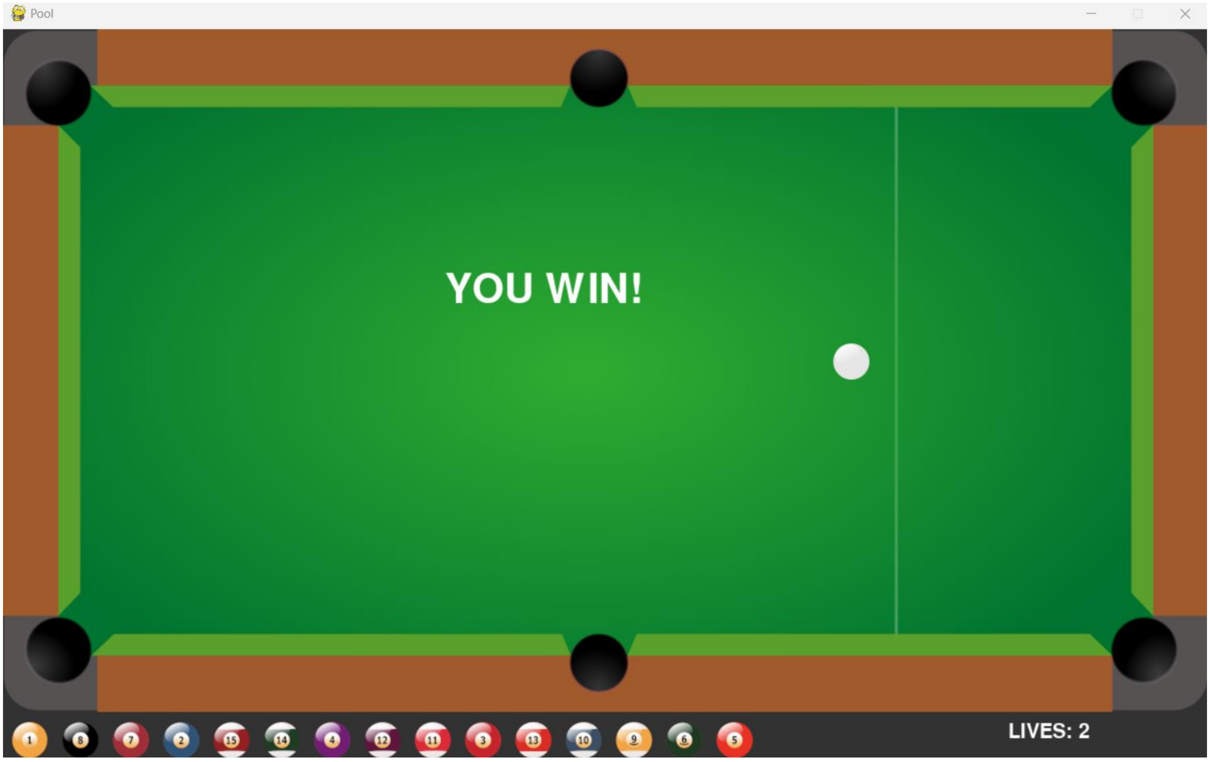
8. Screenshots



LIVES: 3


Pool
LIVES: 3


Pool
GAME OVER
LIVES: 0



9. Reflection

From this project, I have learned that making a game requires dedication, hard-work, time-management and also patience. Making a game is actually really hard especially working on the physics of the game and also the clipping of the game. I also have learned to not do things close to the due date because that can cause extensive stress and worry on the mind.