Intelligent Search and Games Assignment

Giorgos Vainterlis
i6367338

What is Godot?

- Third most popular game engine after Unity and Unreal

- Open Source, Cross Platform, 2D and 3D support

- Uses a Scene System for organization with script assignment.

- Supported Programming Languages
  - **GDScript**: Python-like scripting language designed explicitly for the game engine
 - **C#**: Used for functionalities that require more efficiency and speed

Program Logic
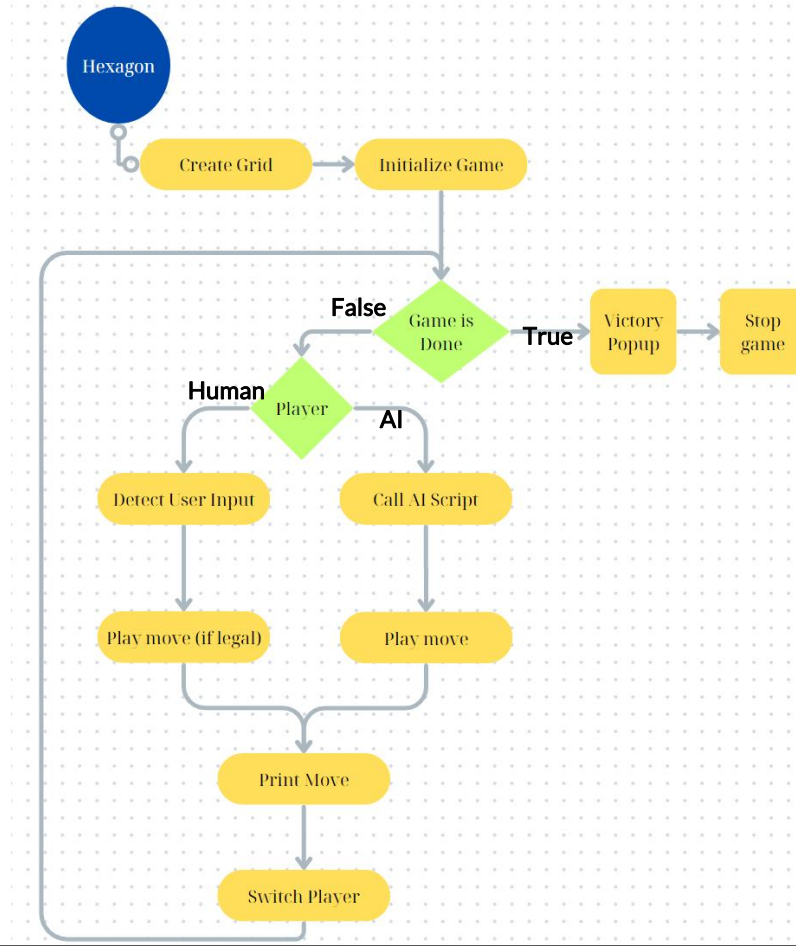
Scenes : Hexagon

Responsible for:

- Detecting User Input and notifying its parent (Grid)
- Displaying the Hex Coordinates
- Changing the hexagon mesh depending on its state(= Assigned Matrix Cell value)
- States:
  - v = 0 : Empty Hexagon
  - v = 1 : White piece on Hexagon
  - v = 2 : Black piece on Hexagon
  - v = 3 : Out of bounds or blocked Hexagon (after a capture)
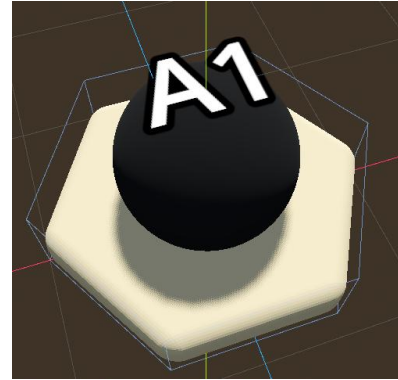
Scripts : Grid

Responsible for:

- Initializing the grid board
- Creating the 10x10 matrix
- Transmitting the signal that a hex has been clicked to main
- Playing the ADD and REMOVE components of a move made by a player
- Updating the board-matrix
- Function for checking if a player has won
- Function for checking if you can capture after a move

# Representation of the game board with a 10x10 matrix

Scenes : Main

- Connects every script and scene together
- On program initialization, assign pieces to AI or Human player
- If AI is playing call C# AI_script and update board with the move returned
- If human is playing have grid script do the move and update the board
- When a move is made sent signal to Move_log script to print its details
- Has a Reset game button
- If a player wins, a popup appears saying who won

Functionalities: MoveLog, Reset button, Starting Menu

- **MoveLog**
  - Handles the printing of the moves on the side
  - Separate Scene
  - Called whenever a move is made
  - Minor bug: Called even if the mode made doesnt do anything
    - e.g. Player clicks on a tile that already has a ball
- **Reset button**
  - Reset the board, current player's turn, move log and other visual indicators.
- **Starting Menu**
  - Gives the player three options:
    - Play as white vs AI
    - Play as black vs AI
    - Play vs another player

Played F6 by Player 1.
Played E5 by Player 2.
Played F6 by Player 1.
Played F5 by Player 1.
Played E6 by Player 2.
Played G7 by Player 1.
Played H8 by Player 2.
Removed F6 by Player 2.
Played F7 by Player 1.
Played E7 by Player 2.
Played E8 by Player 1.
Played E4 by Player 2.
Played E3 by Player 1.
Played H7 by Player 2.
Removed F7 by Player 2.

Reset Game

Player vs AI(You play 1st)

Start
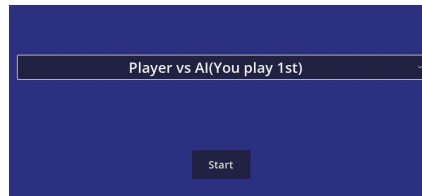
Class AI Script (C#)

Functions

1. **Play(board, player, captured piece):** Connects GDScript to C#, calls Alphabeta Minimax and returns a play position.
2. **IsTerminalNode(board):** Checks if WIN/LOSE/DRAW.
3. **InMap(position):** Checks if position exists in the board.
4. **CheckForCapture(board, position, player):** Checks if by playing move on position they player has made a capture. Returns List of potential candidates that can be captured next turn or an empty list otherwise.
5. **DoMove(board, move, player, previous captured piece):** Does a "ghost" move that is used for Minimax.
6. **FindSuccessors(board, player):** Finds and returns all the children of a node.
7. **EvaluateBoard(board):** Evaluates the board(Distance from center, 3s and 4s in a row and pieces count for each player).
8. **MiniMax(board, depth, alpha, beta, player, captured piece):** ab pruning minimax algorithm modified to also return move and to handle captures for previous depth.