Multiplayer Mobile Strategy Game Technical Explanation

Design Patterns used in the project are Singleton, State. I could've used dependency injection but for this project's sake I just decided to go with the singleton pattern instead. State machine helped the lumberjack's states to determine if they were idle waiting, moving to a target or chopping a tree.

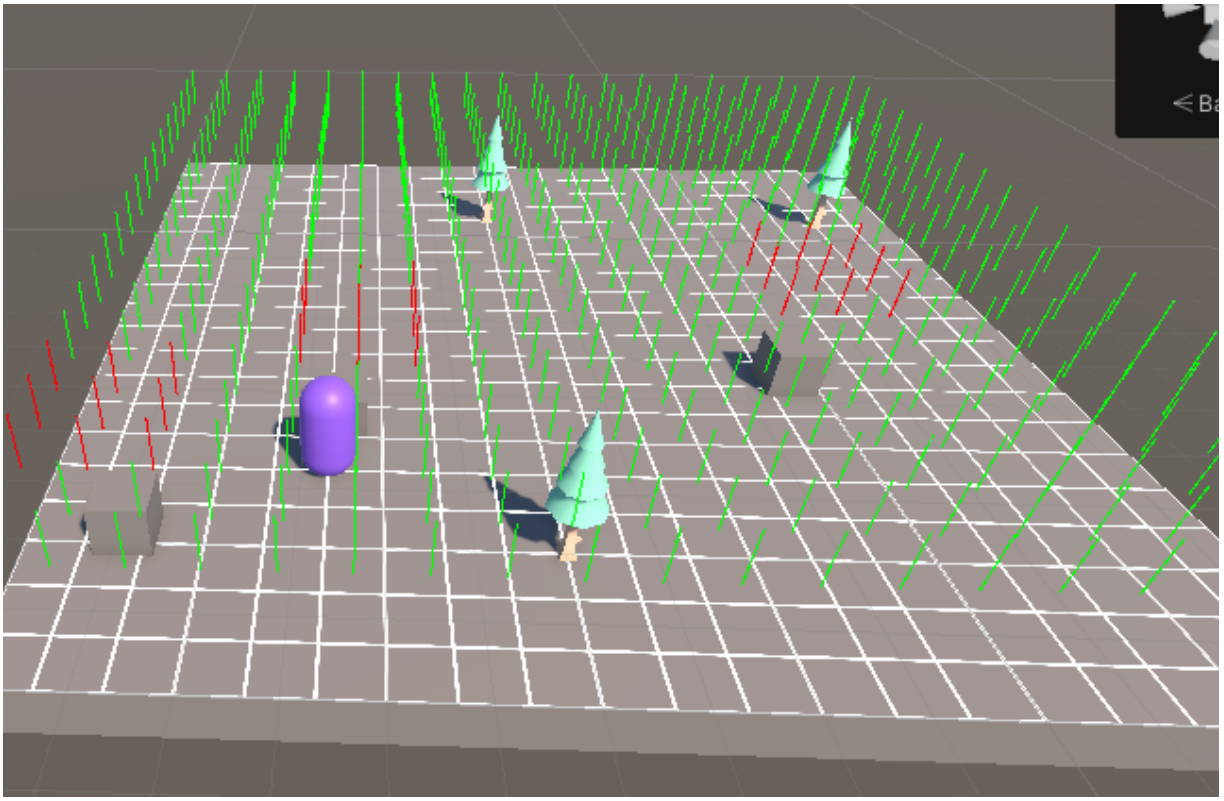
Networking with PUN - Main Menu - Lobby

I’ve used the **Photon** Unity Networking package for multiplayer system implementation as stated in the case documentation. It automatically connects to the master server and joins the lobby. The player can set their **nickname**, **create** a room or **find** and join by selecting a room. After starting the game, PlayerManager is being instantiated (each player given a random **color** and **synced**) for each of the players in the room. PlayerManager then instantiates the **lumberjacks**(TimberMan).

Mechanic

Player can click on his own lumberjacks to **select** and **deselect** action. If the player clicks on the **ground**, all of the selected lumberjacks try to get a path to the destination from **Pathfinding** Manager which uses **A star** pathfinding algorithm. If the player clicks on a **tree**, all lumberjacks first try to get close to the tree and when they get close enough, start to chop the tree with small animation.

Obstacle Avoidance



After constructing the pathfinding class, the pathfinding manager fires boxcasts for each of the grids (from top of the grid to the grid). If boxcasting returns true with “unwalkable” layer parameter, it sets the grid to unwalkable (This feature is not designed to be changed on runtime, it only checks on the start of the game). The screen shot above shows which grids are unwalkable (red line) and which are walkable (green line).

Additional Information

No external asset used in the project (other than the tree model which I downloaded from kenney.nl with free use license).

I’m used to using the observer pattern in my projects a lot but in this project's scope, observations are being made by the photon package itself so I didn’t need any.

Thought about the selection system for the lumberjacks, since this will be a mobile game and not an actual product, I decided to go with single click select-unselect action.

Every lumberjack’s rigidbody is assigned with random mass between 1f, 3f. This is to avoid them getting stuck while pushing each other to get to a position. This way, they can get to the position by pushing each other out of the way naturally.

Also I wanted to say, It's really thoughtful of you to give 1 week for this case. Although I think this is a 2 day case, since I’m looking for a great team to work with I’m not rushing anything and working freelance to earn my living. Because of that, 1 week was really a relief for me to gather myself. Even if you guys decide to proceed with another developer,I wish you all the best :).