The goal of this document is to encapsulate my thought process while working on ABG RUSH. Of course, not everything will be here, but I will be copying and pasting anything that I write down on sticky notes on here. That way I have an idea of what was going through my mind while I was working on a specific aspect of the game. This will also allow me to write down ideas I am either not able to implement, due to not being in the scope of the project, or because I am simply not able to focus on that part of the project at the time.

6/1

Each game object could set its own z value. This would require another script on each and every object in the scene. This script would simply set the z value = y value. This way the objects that are higher(y value) in the scene will appear to be behind objects that are lower in the scene (y value).

Since each object would be doing this themselves, this could cause some trouble with processing, but it shouldn’t be much, even if it is being done each frame.

Another option would be to have the games manager handle this as well. Each time a new object is created (sprite wise) it informs the manager, the manager adds it to a list, and each update, or specific frame (1/30) the manager will set the z values of each object to its own y value. This would still require each and every object to inform the manager of its existence, but since each object will most likely be communicating with the manager, no extra step is required on startup. Objects that don't move should probably still inform the manager and be updated each frame, or at least on startup. This will prevent changes in the editor, such as moving objects around from ruining anything.

6/2

Each object that the player can interact with will have a location for patients, and a location for players. So if the player interacts with a chair, the player will stand in front of it. If a patient on the other hand interacts with a chair, the patient should stand on top of it, and change it's animation.

All objects that the player can interact with should have 3 states. Idle, Active, and Hovered (which is a pseudostate).

Idle will have a very simple/normal looking sprite and no animation.

Active will have a brighter looking/more eccentric sprite, and possibly an animation. Either that, or the patient within the object shall have some kind of animation/change.

Hovered will possibly change the cursor icon, and change the objects sprite or animation.

Hovered can only be accessed if the object is both active, and the mouse is over it.

Removing the mouse or clicking on the object will turn it back to it's active state.

For the z-axis, only objects that move (nurse and Patients) need to actively update their y/z position. All others only need to set it once.

All interactable objects will be based off of a single class InteractableObject

Keep track of state : idle, active, hovered

Nurse Position Location

Patient position Location

So for instance, if the player clicks on the reception desk:

Player.setposition.receptiondesk.NLocation and the nurse will move to the reception desks n location.

Since I plan on making the waiting room chairs and testing those first, I have made animations for them. These animations are: Idle, Active, and hover. They are for the states that the chair can be in. So when the chair has nothing going on, it will be in its idle state. When it has a patient, it will be active and it will slightly change color. When the chair is hovered over, the chair will slightly rotate.

Before making these animations and any others that will be in the game, I have decided that objects that will be interacted with will have their specific sprite as a child component of themselves. So for instance, I made the chairs first, set up all the properties and components. I made sure that the collider fit the chair, and then I removed the sprite. I then created a child gameobject of that object and placed the sprite there. Due to this, animations will only incorporate the gameobject’s child sprite, and should never effect any part of the actual object. This will allow me to create animations without fear of anything else possibly being messed up. So for instance, If I made the hover sprite interpolate between sizes of normal and large, and rotate, the actual object itself, and it's colliders would not be changed or affected by this. So the hitbox will remain exactly the same, and not cause any problems for the player.

Appear to have run into a small unforseen problem with layering. Although I am setting the z position each update, it's staying/resetting to 0 during movement, and remains that way until the nav agent has stopped moving. This leads me to believe that I may have to change something within the navagent scripts to convert them from vector2 to vector3. This would allow there to be a constant z value instead of it being reset to 0 each frame when the vector3 position I create is beind downcast to a vector2, which is what appears to be happening.

After taking a brief look through the polynav code... it looks like more time may need to be devoted to this in order to make objects be drawn on top of each other in the proper order. I will probably ask warren to give me some feedback or ideas on what to do. i think I am done for the night. I'm going to copy and paste this to the journal, upload the repository, and upload the current version of the game to google drive.