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 into this journal. 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.

June 1st

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.

June 2nd

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.

June 3rd

Today I would like to...

Have 3 hotspots working - Reception, Waiting Room, Patient Room

This would include

- Location for nurse and patient

- Send both the nurse and patient the correct location data

- Become Idle/Active Depending on status

- Know if a patient is using this hotspot, so if a patient is currently in this waiting chair, and return the patient

- Have an active function such as clicking on an active waiting room chair, will make the chair inactive, and the patient should move to an open room if one is available

Manager

- Keep track of the number of open rooms

- Keep track of the number of open waiting chairs

- Get/Send the above information if needed. So being able to verify the number of open rooms

Nurse

- Move to correct location for each hotspot

Patient

- 3-4 different states that have their own specific timer values

- Move to correct location for each hotspot

- Tick/Countdown the patience timer

Creating a person class to be the parent of both Nurse and Patient. The main reason is for movement. If movement is handled through the person class, then I won't have to differentiate between calls later on. I feel like there should be more that the person class can do, but I can get back to that later once I have movement working.

I would like a way to differentiate between nurses and Patients, and I think tags may be the best way. This will allow the Person class to differentiate what it does. So for instance, when a location/hotspot has been reached, the Person class will inform the object to update it's status. A patient that was moving to the reception desk would therefore update it's status to match that. And a nurse on the other hand would open up the dialogue box for the first available patient.

Patient's should not continue to tick down their own clocks if they are being interacted with. So I need to create a check somewhere that will prevent this from happening. It can be in the patient class itself. The problem is, do I have a function called inside the patient that flips the switch or should it be done from outside? For instance, if the player clicks on this patient while their in a waiting room chair. The patients UI will pop up. At this point, the patient needs to stop ticking down its timer. And when this UI is closed, either by performing an action or inaction, the timer needs to either continue clicking, or remain stopped.

I feel like outside would be better. For instance the UI will know what action was made by the player, and can inform the patient of this action. So in this instance. UI Opens. UI informs patient to halt timers. If the player clicks "Treat", the UI closes, the patient's timer remains halted, and the patient moves. If the Player clicks "A room will be ready soon", the UI should close, the patient should have the pacification delay added to it's current timer, and the current timer should become un-frozen/halted. In both of these situations, I also feel that the UI should inform the current hotspot of what happened. So if the patient is moving elsewhere, the hotspot should free itself up, and change it's status.

At the same time though, a Patient may also need to be the one informing it's current holder/hotspot that it's leaving. So it may be better if the UI doesn't handle this, and only the patient does. This is simply because the UI would have to have more added onto/into it to handle that, while the patient already has to inform the hotspot due to the fact it can leave/storm out.

Should the ever present patient patience timer be a total amount of time or segments of time? And if it's segmented, should leftover time be added back to make a new total or should it simply be gone? The former would mean that the player has at least 50 seconds to get the patient in and out. The latter would mean that the player has only 15 seconds to sit the patient down in a room or in the waiting room. If this time elapses, the patient leaves. And the addition to segmented would mean if the player sat the patient down with 5 seconds left, do those 5 seconds simply get lost? Or do they get added onto the next timer. Also, if the patient is briefly pacified (more time added), does this extra time get added as well or no?

Only Exam Rooms, Waiting Room/Chair, and Triage actually use a/the patients. So should patient interactions be done individually or within interactableObject or should another parent/child relationship be created? I think I'll make another parent/child relationship. This will trim down what's in the top parent, and make it so that other objects, such as reference desk, sink, and more, have much less involved with them.

Those 3 will require 2 states, idle/active and Hovered. They do not need a patient variable since they will not interact with them. They will also not need a vector2 for the patient's location. They will each perform their own functions. I think I may re-write the Interactable Object Class, and the other few really quick because they are currently very small, and I would feel better renaming and repurposing them now, instead of possibly later. New Names will be:

OfficeObject - Objects around the office. This will encapsulate everything.

- Access to the manager

- Location for the nurse to stand

- Access the object's animator

- State Controller/manager for animations and actions, Individual classes should be able to set their state. So the object's script should determine if the object is idle or ready.

\*Going to stop using Active, and start saying Ready. Also, I started saying Exam Room instead of Patient Room.

- OnMouseOver / Exit, since each of the objects will have these, it would be better if handled from inside.

Patient Objects

- Child of OfficeObject

- Location for patient to stand/sit

- Return location for patient to stand/sit

- Patient variable for current patient.

- Return the current patient

- Add/Remove the current patient

- Set the status of Idle or Ready based on current patient (OfficeObject handles hover)

WaitingChair

- Child of PatientObject

- Patient sit animation

- UI Information/Data

Triage

- Child of PatientObject

- Queue up patients

- Patient Story UI

ExamRoom

- Child of PatientObject

- Patient sit animation

- UI Information

- Update Patient State (multiple Times)

Sink

- Child of OfficeObject

- Set the Nurse's clean bool to true

Reference Desk

- Child of OfficeObject

- Turn on the Computer / Show Help UI

- Pause Game?

After writing these up, the differences between them seem very negligible, but I feel that I should separate the classes anyways since more may be added on down the line.

Still need to queue up the patients that appear at the triage desk. Other than that, The patients currently go to an open slot/position,and will wait there until there time runs out. Then they will leave. Of course, it doesnt look pretty at the moment due to a lack of assets, and I don't feel like making some animations right now. But they will look better later. I need to add some kind of animation or a small sprite change to the hotspots themselves so I can determine what's going on during testing. The triage only appears to work if I click it on the left side, which is strange, but since I don't have a visual effect when I mouse over the hotspot, it will be difficult to know exactly where it is. Which is why I need to add one!!

I would say the majority of the goals I wanted to get accomplished for tonight have been reached, but I am currently too tired to verify each and every single one of them. I will be meeting with Professor Warren tomorrow so that will be a good time to bounce some ideas off of him and see what he thinks so far. I plan on having the majority of this movement shenanigans finished by the end of this weekend. This is definitely taking a bit longer than I thought it would, but I guess that's because one does not simply program movement. In order for movement to work, states had to be implemented, states required hotspots, and hotspots required different classes containing all sorts of information. I honestly feel like The majority of the most trivial portion of the coding will be done once movement is. And thats simply because movement required so many of the other aspects of the game to be at least started.

June 4th

Meeting with Warren, Cory Boyd, and Colin (skype)

Vary the spawn time from initially every 20 seconds to randomly maybe 2 at a time or 3 over the course of 10 seconds

Allow the player to send the patient to either the waiting room or the exam room if both options are available.

Random/ not needed patients are on backlog now. "You're in the wrong department"

Nurse should stand next to patient when they are at the triage. This is because the receptionist will be behind the desk.

Repeat visitors are greeted by the receptionist - Backlog now

Computers on wall next to each examination room to bring up more information about the patient.

Reception will get information about patient. Story, Name, and DOB. Player should try and ID/Compare the information the player receives from the computer to the information on the ID band that reception gives to the player.

Possibly find a way to hotspot the patient's arm, or something that symbolizes the patient's ID band.

June 7th

Today I want to implement very basic/small UI that appears after the nurse has reached each location. This would be UI for the triage that says Leave, Wait, or Exam Room. UI for the waiting chair that says (Exam Room, Pacify) and UI for the Exam Room that says Goodbye. This UI will be changed later, and this exam room UI will need to cycle eventually. I also want to make sure that the nurse walks to specific locations for each object. I also want to make sure that each and every single object has a proper hotspot position.

I noticed a small problem occurring within the waiting chair animation controller and fixed it.

In order to make it so that the colliders/triggers created and being used by the polynav agents didn’t interrupt any actions the player made while clicking, I turned the \*Raycasts hit triggers\* option off in project settings.

Going to create multiple Classes for UI specifically for interactions between patient/nurse. The parent will be PatientUI. The purpose of this class is to give all child classes access to their own patient. The purpose is also to allow different sources, such as a nurse, OfficeObject/PatientObject, or manager to access and update the data in the UI itself. And child classes will be able to pull exactly what they need to populate their own specific UI. So the triage UI can pull the data it requires such as the story, dob, and name. At the moment, the waiting chair may display the same information, but will have different options within the UI. And although they display the same information, I feel that creating a separate class for each will be beneficial due to the change/difference in the UI itself, as well as the fact that changes may be made in the future, that would further differentiate the two. And it would be more difficult/time consuming to either add onto them or split them later on.

After a bit of testing and making sure all interactions were happening properly, I have gotten more of movement finished, and one more day should see all of it (what is currently in the game) complete. At the moment, clicking the triage area will move the nurse there, and then open information about a patient. Then the player can choose to send a patient to an exam room, waiting room, and simply kick them out. When I work on this next, I need to add some UI for the exam room, and make sure the buttons work for both the triage and waiting chairs.

June 8th

I only have about an hour to work on this at the moment, but I want to continue working on movement. And possibly get it to a point where I can post a version that others can try.

The Triage is working well, and so is the waiting room/chairs now. I now plan on creating some placeholder Ui for the exam room. I'll just give it an exit button for now. This will clearly be changed later because the exam area and the other objects near it will require so much. But at the moment, this should allow me to cycle through patients, and test movement of the most important things. After this is complete, I'll need to add in the sink, reference desk, bloodwork machine/tube, exam room computers, and id checking hotspots. These should work largely the same. And because of the way I created my classes, the sink and ref desk will be normal office objects. The exam room computers, and id checking will most likely be patient objects, and the bloodwork machine could be either or, but most likely remain as an officeobject.

Noticed that several of the UI classes have the same or similar methods, so I simplified them and placed them into the parent class UI\_Patient. Now each specific class will have access to Send\_Away, Send\_ExamRoom, Send\_WaitingRoom, and Pacify. I also Noticed that the current 3 PatientObjects , Waiting Chair, Triage, and Exam room have the same exact method for onmouse over/click. I may need to simplify that and place it in the PatientObject class if I see no reason to differentiate them.

I updated the main menu a small amount and disabled buttons that currently have no function, as well as fixing the other buttons so that they redirect to the proper scenes. I removed unused scenes from the build settings, so builds should be smaller than they previously were. I then created a build, and placed it on mywebspace:

<http://mywebspace.quinnipiac.edu/rmburgess/ABG/ABG%20Rush%206-8/ABG%20Rush%206-8.html>

June 9th

Meeting with Warren

Click on patient instead of clicking on the objects the patient is currently using. I will need to come up with a collider that does a good job of always fitting the patient sprite. Most likely going to be a square/rectangle, but should it ever change size depending on what the patient is doing? Also, make it so the patient cannot be clicked on while they are moving from point a to point b.

Make sure to add something for a bracelet/ID interface.

Look into changing cursor when action is possible instead of animations. Would need a different mouse cursor for each state that the patient is in. This would mean one for checking symptoms, one for diagnosing, one for taking blood, etc. Might also need cursors for different objects in the game. So hands or soap for the sink, and maybe an id Badge/Card for the triage.

Clicking on patient computer offers no help sliders/scales. Clicking on the reference desk will allow the player to use the practice tool with random values.

Patient computer has 2 tabs, and there will be a button if the patient has been misidentified.

- One Tab shows patient information/history

- One Tab shows the diagnosis.

- This tab does not have values or allow the player to give an answer until bloodwork has come back.

You're a world class nurse!!!

June 10th - Day

I need to Queue Patients.

I'll look over what Ryan did previously to see if there may be something that I can take and use from there.

I have come up with my own option though that I believe could work quite well.

Due to the way the triage will work, there will essentially be two different queues. Both queues will have to have a max size, and if both queues are full, no more patients will be able to come. Or better yet, patients will arrive, mention something about it being too busy, and then leaving.

1. This is the initial Queue, at this point, patients are not able to be interacted with by the player. The patients will wait in line FIFO to speak to the receptionist. After the receptionist has seen them, and done the animation/alerted the player, the patient will then move over to another area.

2. This is the secondary Queue. Patients become active at this point and are waiting for the nurse to speak to them. The queue and resulting positions will most likely need to be verticle as opposed to horizontal because of how the sprites are drawn. The Nurse would have to stand on the right side of the patient. This way, although all of our sprites's faces are always facing the screen, nothing will appear to be amiss, at least with the interactions between the people in the game.

I also need to make it so that patients can be clicked on instead of hotspot objects.

I believe that the way that Patients are currently scripted will allow this quite easily. Hopefully it shouldnt be more than a few lines in a few different scripts. The idea is simply, when a patient's collider is clicked on, it will notify it's hotspot. This means that all the other interactions already coded will remain virtually the same and should not have to be changed.

An example would be:

Currently the game and hotspots do this:

If (Patient is on the Hotspot **AND** The Hotspot has been clicked) Then Hotspot Do (The Thing)

But this new change will do this:

If (Patient has been clicked **AND** Patient has a reference to the hotspot) Then (Patient tells the Hotspot Do (The Thing))

So this is essentially the same amount of steps, and just references the hotspot.

The only thing that might be a problem is the colliders of different objects. How can I make it so that the colliders fit the patient will enough for all the different sprites, and overlap correctly with different objects within the environment? And since I will be using order- in layer instead of z ordering, does that also make anything attached to that specific sprites game object appear first as well or no? That's something I'll have to look into as well. Who knows, it may not even be a problem but if it is, It would be better to look into it and figure out some kind of a solution now. If it's really bad, I could simply just space out different objects more, or I could turn specific colliders on/off based on different circumstances.

As for setting/changing the cursor, it should not be as bad as I was initially informed. Mostly because of the new function available. <http://docs.unity3d.com/ScriptReference/Cursor.SetCursor.html>

I would just need to ask for some cursors to be made for the game. These may include a pair of hands, a stethoscope, a needle, ID badge/bracelet etc. Each of these would be able to portray what the player will be doing next, and would mean that the game does not need as many sprites or animations for the different objects we have within the game.

June 10th - Night

I don't really have any goals for tonight since I am a bit busy. I guess I would like to try and get some of the things I began planning out earlier today put into motion though. If I can get any of these done, or partially done, that would be a help because I will not have a lot of time to work on this tomorrow either. At the moment, it looks like I will have to do most of my work this weekend (fri-sun).

So I spent a few minutes finding a cursor to test with online. Somehow it took 10 minutes to find a transparent texture of a cursor that wasn't the plain white arrow. Anyways, I took another few minutes and placed it in the game. The functions being used right now will be easily transitioned into a final version or more sophisticated version as well, so that’s good. And since it looks like we will be going with cursor transitions instead of sprite animations, I think that using a dictionary for sprites may work out quite well. This would allow me to simply populate the dictionary each time the game is started, each occurrence where a cursor is needed could have its own unique cursor, and I'm not sure about this last part, but it may be possible to change the cursors without someone to go within Unity and make the changes. I am assuming that a person will be able to simply change the images/textures as easily as they can change the xml files.

When it comes to the collider for patients and the changing sprites, it may not be as much of a problem as I was initially thinking. I won't know until I have all the different sprites of course, but if I use a box collider, I can simply set all of the different sprites to have different pivot points that work with the box collider. This would mean all of the settings are set beforehand and not during runtime which would require a lot of thinking and more work on my end.

Another thing that I somewhat planned earlier today was the idea of clicking on patients instead of on the hotspots they were near. So I currently made it so that you can click on patients now instead of the hotspots. As I expected, it only required a few lines to be changed within a few classes. And the classes of Triage, WaitingChair, and ExamRoom actually got a simplification that made their parent classes PatientObject and OfficeObject do all the work now. But now that the patients have to be clicked, The order of the sprites has become more of a problem, and so I will try and find a solution to that with the time that I have left. The problem is, I don't know if the order in layer will also stop the colliders from overlapping improperly. If it doesnt another solution will need to be found, or we could try and make a change to the polynav script that will allow it to use vector3 instead of vector2.

The lower that an object is in the order, the sooner they are drawn. So an object with order 7 is drawn before an object with order 22. So the lower object (7) would appear to be underneath object (22). In our game, objects that are lower on the screen need to be on top of objects that are higher on the screen. So 22 would need to be behind/underneath 7. A simple inverse or negation (\* -1) needs to be done. So then... 7 would become -7 and 22 would become -22. And so, -22 would be drawn first, and -7 would be drawn second. Therefore placing the object at order -7 on top of/in front of the object in order -22. On top of this, we will be working with floats, and the unity sprite component's order in layer does not support floats. So each of these numbers we use (the y value of the object) will need to be multiplied be 100 to make sure it's a whole number no matter what, and then rounded up/down, and then converted into an integer.

Ordering in the layer is now partially complete (all objects in the scene are now ordered, and moving objects (patients & nurse) will order themselves dynamically. At the moment, I have not run into any trouble with colliders and clicking the wrong thing at the wrong time. Also, I saw that before I made the change, due to the cursor change, I could see that the patient was only clickable on their feet or head. This was because the triage desk was in front of them (order wise). After I made the change and had everything drawn correctly though, the patient's entire body was clickable without any hitches caused by the triage desk. So it's possible that the draw order in layer affects the order of colliders. I'll know more after further testing is done.