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.

June 14th

Today I would like to....

~~- Set up some of the new art I have been provided with.~~

~~- Set up some of the new UI I have been provided with.~~

~~- Set up some of the fonts I have been provided with.~~

If time permits, I also want to get queuing at the triage started.

Couldn't find a proper receptionist image so I took one of the source files, played around with it and made one.

Officeobjects may need to have another child, or PatientObejcts may need to be replaced. Since the player no longer clicks on the patient objects, there is not much of a reason/need to have them split up as they are, but on the other hand, there are lots of objects that only the nurse/player interacts with. These are: the sink, reference computer, bloodwork machine(backlog), exam room computers, patient ID bands, and the stethoscope/ get vitals object. Each and every single one of these objects is used more by the nurse than the patient. And only 2 of these even need a reference to a patient. So i believe it may be best to scrap the PatientObject initially created and make a class called NurseObject.

At the moment, the different patient object child classes don't do anything to differentiate themselves from each other. This is because the game now allows the player to click on specific patients instead of clicking on specific areas/objects in the game. The patients simply reference the old patientObject. And it's not a reference to the PatientObject's child class, but the parentclass PatientObject.

NurseObjects will be a parent class that will have mousover effects and a reference to a specific mouse icon. Each child class will have it's own separate functions. For instance, the Sink will inform the player/nurse object that it's hands are clean. The Reference Desk/Computer will open up the Practice Tool and Reference information. Exam room computers will display information about the patient, and possibly cycle through multiple interfaces. The list goes on. But at the moment, the current goals goals/expectations of the game lead me to believe that PatientObjects can be removed, and NurseObjects can be created in their place.

I think I am going to hold off on the queue until I figure some of this other stuff out. Also, I want to ask Professor Warren and Cory a question. The way that we have the game set up now, with the long desk and what not, we are still sending a/the patient to the waiting chair or exam room. Should we be doing this? Does it make sense for a patient to stand around after talking to someone at the triage? Or would the patient go and sit down themselves. I know in reality I would sit down after talking to the receptionist. I always have. But this is a game, and we may need the extra step of the player choosing whether the patient sit's down, leaves, or goes to the exam room.

While placing the new UI I noticed that the mouseover effect created by the patient still occurs when the mouse is over the UI if a patient is behind it. I am not entirely sure of the reason why, but I am assuming it's due to the function I am using for mouse over detection. I think this may be a reason why, because I don't remember any past situations in which objects still interacted with the mouse even though Unity's UI was up and over the object. The past situation I am thinking of most though was using raycasts for detection instead of the built-in functions. I think I'm about done for the night. I'll do some more tomorrow during the day and night. I need to finish setting up the UI as well as determine what updates/changes I will be making during this upcoming week.

The things I want to finish by Tuesday the 23rd...

Patient Interaction UI

Build into the Gameplay UI

Reference Patient Information

Display Specific Patient Information

Button Choices/Options for each Interaction

Patients

Data/Information

Name

Diagnosis

Story

etc...

Stages

Triage (waiting)

Waiting Room (Waiting)

Exam Room

ID Check

Vital Check

History & Request Bloodwork

Diagnosis

Manager

Record Information

Spawn Patients Randomly

Implement a placeholder point system

Keep track of points / time

Display Satisfaction points/time

NurseObject Class - Objects that only the nurse interacts with.

Mouse Over Icon

Nurse Standing Location

Child Classes

Sink

Patient ID

Patient Vitals

Patient Computer

Reference Desk

Bloodwork Machine (backlog)

More??

June 16th

Goals for today -

Finish adding in new UI assets

~~Add queue to new patient spawn at triage~~

~~Patients will sit down in a waiting chair after triage~~

~~Patients will have a conversation with receptionist~~

~~Toggle the Collider/hotspot for patients depending on their current state~~

Begin work on new classes.

So now that we are skipping the portion with the triage where the nurse walks over and tells them to either go to an exam room or wait in the waiting room, that UI can be removed/swapped/adapted for the waiting room. So that should not be much of a problem. One thing I'm trying to figure out now though, is whether or not I should begin work on the new classes (NurseObject, etc.) or work on the other things first. Both will require the other at some point, and this means that the way they are created/implemented will effect the other. I guess the first thing I'll do is update/build a new triage class.

At the moment, the triage has a total of 7 locations that the patients will stand in before being seen by the receptionist. I feel like 7 is a bit too much, but at the same time it's better to have them and never use them, than to not have them and need them at some point. Still need to make the triage go through the motions of speaking to each patient and then sending them to the waiting room.

I have decided to make the speech bubble object on both the receptionist and patients accessible by an animation. The reason is because the receptionist and patients will have several animations in the long run, so it wouldn't make sense to access it by calling and referencing the specific game object each time. And I can make these animations easily accessed by placing creating a function within the patient that accepts a string and turns an animation on or off. So I should be able to make calls like Patient.Patient\_Animation("Talking",true), and that will turn the talking animation to true.

Ran into a small problem with the triage and it's patient list. Not entirely sure why, but the problem is strange. It keeps saying that the list I initialize early on is null/doesnt exist.

Figured out that the problem was being created due to there being two objects in the scene with the tag, Triage. Although I removed the script from the older one, I never changed the tag. I can only imagine how much more difficult this bug would have been able to solve if I had left the script enabled as well. Because then the list being referenced would have actually been there.

I'm going to change up how some calls and functions are made so that they are more self inclusive. So for instance, instead of making the triage add the patient to the waiting chair, and then the triage telling the patient to move to the waiting chair, the waiting chair can perform that as well as some other things the triage is currently doing. Of course, this is a change in the PatientObject class and not the waitingchair child class.

Just took a look at the trello and noticed that I recieved two conflicting responses from a question I posed so I will need to get some more information during the next meeting. The two responses also kind of tangle up what I'm currently working on with the patient queue. So do I put a pause on this and continue with something else? Or do I finish it and simply come back and make changes based on the decision later?

I can work on setting up some UI and make a decision about the queue later on. Ive made some notes that detail what I will need to do the next time I work on it including changes/updates that may need to be made to other classes.

So I have the queue working now, but after doing some testing, I have noticed that as expected, the order in layer is not effecting where/how the colliders for each object are being drawn/created. I'm going to disable the colliders for the old hotspots and see if that makes any difference/fixes the problem. Woohoo! that has solved the problem.

Patients can now only be clicked on at the proper time. The mouse change indication no longer happens unless their is an action that the player can make.

I added a fourth chair. Simply because there is room for one now. It can always be removed later if need be. I need to figure out why the conversation occuring at the triage is happening immediately for a queued patient. It's supposed to wait 1-2 seconds before starting. Turns out a line was simply missing. I dont know if I forgot it before, but all I had to add was timer = value. So that's fixed, and now the player has another small buffer created by the receptionist.

I'm not sure if I want to try and add in the Sink/ID Badge/Exam Room Computer tonight or if I want to add in some more UI functionality. Both shouldn't take very long, but I am out of time for tonight. Also, after taking another look, I may not need to create another class NurseObjects and children of it. My OfficeObject appears to have the majority of functionality that the nurseobject would have. I'm not sure how I forgot that or overlooked it. So it looks like I can make each of the objects, Sink/ID Badge/ExamRoom Computer a child of the OfficeObject.

I think I'll take this time to work on a small amount of UI and possibly work in some of the Sink functionality. This would mean, the sink is clickable, and when clicked on, the nurse moves over to it.

Can't seem to figure out why I need a second click when I am at the waiting chair after telling a patient I will be with them soon. I am currently not seeing anything within the code that may cause it, but it's possible I have been looking at the code for too long today. I'll come back to this tomorrow.

I still need

Images for... ID, Vitals, Computers getting bloodwork.

Different mouse icons for different actions including handwashing, id badge, vitals, possibly bloodwork, waving hello or greet.

UI textures for Practice/Reference tool

Character Images and possible animation sprite sheets.

This includes the Nurse, Receptionist, and the different patients.

So I have started the Sink and PatientComputer classes. The sink should be pretty simply since it will just toggle a value within the nurse. The patient computer on the other hand will be more complicated since it will be going back and forth between different Interfaces, and displaying different information for each at specific times. This object will also have to have it's own patient information, which will be recieved from the exam chair. Also, I need to connect the exam room and the patient\_computer, as well as the soon to be vital sign object. I still need to add in the reference desk, but that should be simple because 90% of that was already completed during the semester. All it needs is a bit of a reskin, possibly a change in 1-2 features, and then just add the reference material/data.

June 19th

My deadline of Tuesday is coming up so I have decided to spend an hour or two determining what I need to accomplish and How I plan on doing that. I’ve also decided to try and organize and determine what I may need, either art assets, data for patients, and simple direction on how different aspects of te game should play out.

Just noticed that with the change to the interactions during the waiting room to the exam room may have caused a bit of a disconnect with the flow of the game. Initially, the player would simply tell the patient to go to an empty exam room and that was it. The way we spoke about and I plan on implementing will move the player along with the patient.

While this does allow us to display how the nurse (player) gets the information about the patient on the computer, the nurse (player) is now situated next to the patient. And since the hands of the patient need to be clean, the nurse (player) must now move to the sink to wash their hands, and then return to the patient to get the rest of the information (vitals/Bloodwork). I'll have to ask Warren and Cory what they think about this and whether or not I should continue with the implementation.

After taking a look in the drive folder, it looks like not all of the art assets are ready. So I took the time to inform Colin of what I would need. After taking a look through the project folder, I noticed how cluttered the art/sprite and texture folder is due to all the separate pngs. The same happened to be true for the drive folder with the images. So I mentioned that sprite sheets may be the way to go in the future. It allows the creator to make sure each asset is a uniform size, and the folders with the art will be less cluttered overall. And since Unity accepts sprite sheets with very little work, it won't be a hassle for me to use them.

I was hoping to get some Idea of what the UI for the Patient/Exam Room computer was supposed to look like from warren, but since I don't have that yet, I'll create a default one. Of course the functionality will be there, but since I am unsure of how the information should be displayed, it definitely won't be final. I just hope that I will be able to get a more final version or visual for the following week.

So professor warren got back to me and explained that the hand washing situation is fine. And it won't be that much of a disconnect. There's also the fact that they should wash their hands anyways since they just touched a dirty keyboard. Ha. Okay, so it looks like I will be moving forward with that.

I think for Tuesday, I may also want to try and get the/a Diagnosis Tool/Reference Computer in the game as well. I don't have all the textures for any of it at the moment though, so that may not look final on Tuesday. Then again, Tuesday is a date set by me, and has no real meaning to any of the members on my team. So while it's okay if I don't get all of these things done, I would like to so that there is less to do the following week, and the team has more of an idea of what will be the next playable version of ABG Rush.

June 22nd

First thing I want to do is create the UI or atleast a placeholder version of it for all the interactions. I also want each part of the UI to be accessible from the gameplayUI script/object. And after taking a look at some of the things going on, I plan on splitting up 1-2 scripts. For instance, the ABGtool manager currently has a lot of functionality that is not necessary for it, and definitely not needed within a monobehaviour object. I can and plan on making a static class that can be accessed by both the Toolmanager if needed, but also the regularmanager/Patients.

I also plan on allowing the manager to choose how the patient is generated and all of the proceeding information instead of letting each new patient generate their own information randomly. This will reduce the number of calls to different classes and the number of references needed to them as well. So instead of having 10 patients with a reference to the manager, abgtool manager, and then calling each of them separately after being created, the manager will make a single call to a new class, and give the patient all of the required information.

I also can't forget to link up the exam rooms to their computers so that the computer cannot be accessed unless there is a patient, and that both share the same information/references.

Just checked trello and saw Colin's update. Looks like I have some sprite sheets to work with and I can also work on re-skinning the ABG tool.

I put together a placeholder patient computer, and the data that it will display. After thinking of how I should program everything and the layout, I realized that there are several ways I could go about displaying the information so I'll probably have to speak with the team at some point. To give an example, the data could be static, and always display Symptoms, Conditions and Medication. It could be dynamic, and only display medications, if medication is being used. I could display images as well as words to better get the point across. I could also group the information into columns. And there are many more possibilities and I won't know how I should go about programming different aspects of it and whatnot until we have talked it over. One thing that is constant however, is that I would like the data/information to b functional and look correct in different languages.

June 24th

After taking some time to think about it, I have decided to create a new class specifically for diagnosis. This will allow patients to simply provide a reference to their diagnosis and only contain specific information like name, and DOB instead of all the other information that a diagnosis requires.

The diagnosis class will have:

-String Short Story

-String Long Story

-String RespMet

-String AcidAlk

-String Comp

-String Symptom\_1, Sympton\_2, Symptom\_3

-String Condition\_1, Condition\_2, Condition\_3

-String Medication\_1, Medication\_2, Medication\_3

-float valPH

-float valCO2

-float valHCO3

Several functions including

- Answer\_RM

- Answer\_AA

- Answer\_Comp

- Conditions

- Symptoms

- Medications

Although the creation of this class will require some changes to the current functionality of things, it will be better overall from the previous method especially since one goal of the functionality is to be editable without going into Unity. The Manager or some other class can generate (read in) all of the different diagnosis information and create an array of them at startup. Then the manager would simply assign a random Diagnosis to each patient after they are created.

My goals for today are...

~~Creation of Diagnosis class and any others that may be involved. This includes the ABG class.~~

~~Speaking of, the ABG class can be the manager of all the different diagnosis.~~

Linking Exam Rooms to computers.

Nurse Walking to Patient computer after waiting room interaction.

Nurse Chatting with patient at exam room.

Exam Room interface 1 & 2 popup integrated.

Possibly

I'm not sure if I plan on adding any sprites or animation type things today. While I do have access to some, not all of them are readily available.

While working on ABG and Diagnosis class, realized that the ability to read in values for resp/met, acid/alk etc may mean that the current way of generating the diagnosis may not be needed in its current form. The reason is because the current version accounts for both random, and specific diagnosis to be created. And it alters variables inside of returning variables. After taking another look and thinking about it, I plan on making different versions of it. 1 will generate the string values/answers and return those. The second will accept strings and return float values.

This way, if in practice mode, I can use the initial function to string answers. And then use those string answers to get the values.

Nevermind, it looks like there may be a way to solve this. I can return a tuple of string[] and float[] and then access that. And nevermind that. It looks like Unity does not currently allow for C# to use Tuples. But I have come up with another solution.

In the case of the diagnosis patient's have, they already have a reference to a diagnosis class. However, when random diagnoses are created for practice mode, only the string answers and float values are needed. So if I create a function that fills in the blanks, everything should work fine, and not much editing/updating will be required. The diagnosis class will need to be altered so that the values can be changed after creation though.

Ideally... A Diagnosis with or without the string answers will be passed into the function. If it has string answers, it will move to the next section of getting float values assigned. If there are no string answers, then the function will generate them, as well as values to go with them. This diagnosis will then be returned. This should allow diagnoses made for patients and diagnoses made randomly for practice to use the same function.

So after doing quite a bit of coding, I think that my ABG class is just about done, and my diagnosis class may be complete. I am a bit afraid to try them out right now though because I don't have much time left tonight. This is especially true because implementing this would require alterations to other classes that already need to be updated. And I think I would rather save the headache for tomorrow or friday if something went wrong. And anything could have gone wrong. Most likely something small like a single value could completely throw everything off.

Went back and added some public functions to return a string of conditions, symptoms or medications to the diagnosis class. These may be changed later depending on how we choose to display this information if we choose to display it at all. Creating these classes has taken more time than I initially thought it would, and I don't think I'll have time to work on my other goals for the day.

June 30th

Nurse

- ~~Walk over to the Exam Room computer after sending a patient to that exam room.~~

- ~~The nurse must then have a brief conversation with the patient. During this time, the player should not be able to click on anything.~~

- ~~Have a boolean for clean/dirty hands that can be toggled after interactions are made~~

~~Working Sink~~

~~- wash/clean hands~~

Dirty Hands

- Hands become dirty after interacting with a patient or touching a computer.

- ~~Setting Up Exam Room Computer~~

-

Working Exam Room Computers

- Display information about current patient, based on state of patient in room.

- ~~Only Clickable if/when has a patient.~~

Patients

- Make Sure all states work.

- ~~Triage,~~ ~~Waiting,~~ ~~Exam Room,~~ Vitals, Bloodwork, Diagnosis

Fix that strange double click

- If the Nurse/Player closes an interaction, such as by saying I'll be with you shortly, the player must click the patient twice in order to get the UI to popup again.

Verify that only available actions are clickable

- ~~Patient~~

- ~~Sink~~

- ~~Patients~~

-~~While Moving~~

-~~While Sitting~~

-~~While Waiting~~

Reskin

- ABG Tool

- ExamRoom Computer Interface

- Reference Desk Computer Interface

Rework ABG Tool

- Work with New Class: Diagnosis

- Work with Class: Patient

Manager

- Spawn Patients over time

- Initialize Diagnoses

- Spawn Patients with Names, Dates of birth, and diagnoses.

Animations & Sprites

-Patient

-Sitting Waiting Chair

-Normal

-Upset

-Angry

-Sitting Exam Room

-Normal

-Upset

-Angry

-Triage

-Back Turned

-Nurse

-Normal

-Back Turned

Going to give the nurse a busy flag that gets set if an interface is open or if moving and whatnot. All mouse over interactions will require this busy flag to be false before they go ahead with any other actions. I find it funny that out of all the scripts in the game, the one for the nurse is the smallest. It currently only has 16 lines total.

Thinking about possibly giving OfficeObjects a perform action method that is overridden. That way, when movement is done, the OfficeObject will be called back and have perform action called. This action will do whatever is required/necessary for that specific object.

The nurse's busy flag is set and works fine. When an interface is open, or if the nurse is moving, nothing else can be clicked. The nurse currently follows the patient to the exam room. At the moment, the nurse goes straight to the computer, but I could change it so that the nurse goes to the patient, and then the computer. The next thing to do with this sequence is to incorporate the dialog between patient and nurse.

While the status of patients has been movement based up until now, it will soon be based on other factors as well, and be set from different places. Due to this, the names of some methods have changed to not be movement related. A Status string has been added to the Patient class as well as new timers for vitals, bloodwork, and diagnosis. The Person Class is also seeing some changes since some of the interactions within the game have changed.

So at the moment, the nurse walks to the corresponding computer after sending a patient to the waiting room. I found a problem with this though. Due to a change I made about 1-2 weeks ago, the corresponding ExamRoom does not have a reference to its patient until it arrives. The way it worked before is that it gained that reference immediately after being assigned. This is a problem because if the nurse ever arrives at the computer before the patient arrives at the exam room, then the nurse won't setup the exam room and its corresponding information.

One way to solve this is to change the speed of the patients to be faster than the nurse, but this can still lead to some problems later on if the way movement works ever changes or if someone else take's the reigns and don't learn about that fix. I would rather create some kind of delay or failsafe in the code that waits until the patient has arrived, and then proceed. That way, there are no changes for this hiccup/error/glitch to occur. I could simply make the function break/stop if it looks for a reference and doesn’t find one. It would then try again every .25 seconds. Another option is to turn this function into a coroutine that simply waits .5 - 1 second before taking action. However, this could lead to the same problem as the changing speed option. Another option would lead to changes outside of the script, but I'd prefer not to do that since it would/may require multiple changes in multiple scripts.

Hmmm... currently, this check only needs to occur after sending a patient from the waiting room to the exam room. So I could essentially, create a very small method/function that is delayed by a second or so. While this is a good solution, and would not be any trouble to create, I still feel that I should probably go with an asynchronous method that waits for both the nurse to arrive, and the patient to arrive, and then acts. But the problem with this is that it would require quite a bit more work to get working. Well, not quite a bit, but it's an extension off of the delayed version, and this one could/will use more cpu since it is constantly checking to verify that both references are available.

I think it will be best to come back to this later after I have a better handle on how I have everything working. There are too many unknowns at the moment. So at the moment, I'll simply just increase the speed of the patient and reduce the speed of the nurse. This is an external change that requires no coding, and it should work every time, at least in theory. After doing some testing with the speed, it's definitely a fix, but I don't like it. And the movement seems a bit wrong/badly paced now. I'm also going to increase the conversation length of the patient and nurse when setting up the exam room computer, and doing the vital signs.

Created a script for the ExamRoom Computers. I'm not sure if I want them to be called from the manager however. I'll figure that out tomorrow. Looks like I got quite a few things done today, and I plan on getting the rest done tomorrow.

It seems like it's slow going, but I think that’s because I'm currently waiting on assets that may change how some things work. So each time I work on something, for instance: today I worked on the different states of the patients and going through the cycle. Although I am close to finishing, I stopped to work on other things because I know I won't be able to finish it and test it thoroughly. I'm still waiting on Different Interfaces, and animations. The animations are on the simple side because I just need to add in a few lines of code to fill in gaps I already made for them. Interfaces on the other hand are more difficult because graphics are difficult to change, and the interface itself requires code specifically for it. And until I know exactly what the interface will have, I can only build a skeleton in anticipation.

I guess the best way to explain what I mean is... Having a car, all the gas you need, and knowing the destination, but not having the map. You could head to the destination without the map, but there’s a chance you won't get to the destination, and all the gas would be wasted. You would be stuck wherever you end up. You could head there with 80% of the map, and guess the other 20% but unless you're really good at guessing, you're liable to end up stuck once again. A third option is to follow the map to the extent you can, Fill in more of the map, and then continue moving. While this way may be slower, you’re far less likely to run out of gas, and much more likely to reach the destination. The only problem is: What do you do when you're waiting for more of the map?

Luckily for me, this is only a metaphorical example, and while I wait for one map to fill in, I have plenty more maps to reach other destinations while I wait.

July 1st

As for the problem I came across with the patient/nurse arriving at different times, I have decided to create a third, very small method. This method will not do anything unless the script has a reference to a patient, and the nurse. When it has both of these, it will perform whatever needs to be done, and then reset the references. This will go in the nurse class. And... it works.

I'm going to start linking the Manager, GameGUI, and other required interfaces now. The ExamRoom UI is now linked. Now I'm going to update it since I have the assets for it now. The layout won't be final, but this will allow me to get a bunch more testing done.

Created some of the interface for the ExamRoom COmputer. Did some animations for the buttons as well, that may have been a waste of time depending on how things go later on. I need to add in the data fields next. Then I should be able to test the states further. Oh, and at the moment, The nurse brings the patient to the exam room, they talk briefly, and then the nurse opens up the computer.

While testing and debugging the ExamRoom computer, I noticed a problem with the ExamRoom Chat that occurs when a patient has just been moved from the exam room. Although I fixed it earlier, I happened to come across a few occurrences where nothing would happen. After spending some time debugging that, I realized that it's because the patient may be in 1 of two possible states. This is because of the movement speed of the objects as well as a few other factors at the moment. But the Patient could have their status as either WaitingChair (since it has not left yet) or ExamRoom (if it got there first).

So at the moment, I have implemented an if statement that catches and handles both the same way. Once I have adjusted the movement back to a same/similar amount for both, it shouldn't be a problem anymore. But I'll keep it in there anyway.

Speaking about the ExamRoom Computer interface, I think it looks okay for now. It's got text fields for name, dob, bloodwork values, history, symptoms, conditions, and medications. All of them work at the moment and populate accordingly (or as much as I have tested). There are two tab buttons for Patient History and Diagnosis. Those are currently being implemented. There’s also a close button and a bloodwork button. At the moment, the Idea is that the bloodwork button will appear 80-90% of the time. The initial 75%, it's not interactable since the player is not ready to request bloodwork. When the player is ready however, the button becomes interactable. Clicking this button will then start the bloodwork timer, and the button will become disabled again. The next time that the player looks at the interface, if the bloodwork is finished, the ABG values of PH, CO2, and HCO3 will have taken the spot of the button.

Not sure how much I would like to finish for today. I still have a bit of linking/referencing to do, but those should not be difficult. I think that reskinning different areas of the game may be more time consuming because they also require new/updated implementation for everything. For instance, the new textures I got today took about an hour and a half to setup. And by setup I mean, breaking down the old/placeholder version, and rebuilding it with new assets. Then making sure that everything looks good at different sizes and whatnot.

Well, in my last bit of time working on this today, I happened to run into a problem that was causing me quite a bit of trouble. References were not being used for some strange reason. And checking in the editor with debug mode showed that they were indeed there. After debugging for nearly 30 minutes and double checking the spelling of everything, I found where the error was occurring.

The polynav's movement method allows for functions to be called back when they are completed. So when an object moves from a to b, the method specified will be called back. The polynav also handles everything separately, meaning that the instant I tell it what to do, it starts. And when it reaches its destination, it stops and calls my specified function.

I had been calling this method in between my function, and not at the end, meaning that several variables and values were set after the polynav would begin moving. And if/when the polynav was already at the location it needed to be, it would call my specified function immediately. So while I cannot be 100% sure, it appears as though the polynav (which didn't have to move) called my specified method immediately after it began since the destination had already been reached. Therefore, the variables and values I was setting after this call to the polynav may not (and were not set) by the time my specific method needed them.

When my method was called, it was unable to find references to the values/variables it required. This resulted in a null reference. I fixed the problem by setting all variables/data before I made the call to the polynav. Now I am having no further problems, and on top of that, I believe that the pesky double click problem has been solved as well.

Okay Yup!! That double clicking is fixed :D

I changed the locations of the nurse and patients for the exam room chairs. This isn't final, but it looks a bit better for testing. I also need to remember to shrink the sizes of the Receptionist and Patient Speech bubbles to be the same size as the nurses.

July 4th

Since I will be rebuilding the ABG tool I figured that I may as well revamp the code and it's functionality as well. I'm going to look at a tutorial for drag and drop functionality. It's a shame I didn't find this before when i was working on this. <https://www.youtube.com/watch?v=c47QYgsJrWc>

Ran into a small problem while adding the drag and drop system. I had disabled the image component attached to each slot. This however created a problem because the ray casts required for the event system were using the images. Once I added them back, everything worked fine. Now I need to make sure that whatever object is currently being dragged is always drawn on top. And that only took 2 minutes. Now I need to finish building the ABG tool/Diagnosis Tool. And then I need to go into the code and make adjustments for the updates.

Hmmm...

I want to conserve space on the screen, and the best way to do that would be to alternate the values within each dropdown box based on what answer is currently in the answer box. The problem is that this may take a bit of time to accomplish. This is especially true since I am still working on building the new tool itself. While the drag and drop portion of it is working fine at the moment, the dropdown answers don't have much space to work with. There are multiple ways I could approach a solution to this but I feel the aforementioned may be best. Another alternate would be to allow Unity to dynamically space the interface, but I feel this is unreliable, and could result with very inconsistent visuals.

I'm going to take a break for a bit, but hopefully when I come back, I'll have an answer of what I'd like to do.

It's more important for the ABG tool to be functional than it is for it to be pretty for now, so I'll just make sure it works and worry about that later. Speaking of looking pretty, I need to ask everyone else what I should do about the inconsistency. We currently have 2 new versions for the ABG tool. One is for game play, the other is for practice, but I think that the way it looks should be consistent. But both versions look drastically different both color scheme and objects/shapes used.

Since the Original ABG Tool was made quite some time ago and the game has changed a lot since then, I plan on making a new script for the tic-tac-toe tool. All it has to do is: Reset Pieces on the board, Accept an answer, Display correct or incorrect, and reference the ExamRoomComputer UI that it's attached to.

I think that if the answer is wrong, the submit button should become locked/disabled for a second or so. This would force the player to take a look at what they selected, as well as being a slight penalty due to the amount of time they are losing.

So the ABG tool works now, in gameplay mode at least. I only have to add 2-3 lines to finish functionality for practice mode. The if statements are there, but thats it. The game is playable and patients can go from start to finish and get a diagnosis now. I think the tool came out quite well, even though it doesnt look at great as it could.

But now that that's out of the way, the lesser/smaller things still need to be done.