Muigai Unaka 04.19.2016
Jake Rudin GAME 4355

Written Brief – Aversion_WrittenSubmittal.pdf

- **Game Description ("back cover" 150 to 250 words)**
  - Many a man has been subject to the confines of Apartheid Era South Africa's most storied military hospital, Ward 22, yet none to this day have left as they came in. With boastful claims of curing all who enter his facilities, Canadian-born Dr. Levi N. Aubrey has made a name for himself amongst the people of the Rainbow Nation. Dr. Aubrey would go to extreme measures in order to help his patients get the help they so truly need, no matter what it takes. You are a retired journalist and Dr. Aubrey's newest patient admitted into Ward 22 to help spell some of your deepest demons including the gambling addiction that cost you nearly everything. Shortly into your stay, you find out you're in for more than you bargained for. Soon you realize you are not a patient but merely a test subject in a sick, twisted medical experiment. Escape in one piece and leave as you came, if you can.
- **Game Instructions ("how to", controls and major goals)**
  - Press 'N' key to turn on/off the Nightmode Camera
  - Press 'O' key to zoom in
  - Press 'P' key to zoom out
  - Press 'B' to use a battery
  - Use arrow keys to move around
  - Left click on the mouse to open doors, pick up items and push buttons
  - Goals: document the facilities to incriminate Dr. Aubrey
- **Technical Desc. (Desc. Of how the script are layed out, main scripts and variables)**
  - ElevatorScript
    - Variables
      - Have a Boolean flag for if the door is open or not,
    - Scripts
      - elevatorDoorOpen() function – the Boolean flag is inversed (if was true is now false when function is called, if was false is now true when function is called). This method is then called

by my the main character (the user) when the main character mouse left clicks on an elevator button (which I added box colliders to along with a game object tag of “elevatorDoor” to keep in line with all of my door interactions being within one function.

- InteractScript
    - Variables
        - This contains an item inventory as a generic list of game objects, in the future I will reference the items in this list and populate a UI window with game objects that share the same tag as those game objects.
        - distanceToItem – the distance one can be to pick up an item
        - distanceToDoor – the distance one can be to interact with a door
        - three int variables to store information on quantity of batteries, ~~tapes~~ (now documents) and keys
        - displayItemText – A Text to display for when users pick up items, have an objective to do, etc.
    - Scripts
        - Collect() – Picks up an item and erases the visual of the item from the world, while also adding the item to the inventory and setting the Text to be displayed. Collect only happens if the mouse if over the item and the item is within the distanceToItem amount
        - useBattery() – at the moment it decreases the amount of batteries you have by 1, in the future it will affect the UI of the battery icon on the camera
        - DoorInteraction() – Uses the same Raycast physics from the Collect method to see if the mouse’s position is within the distanceToDoor in relation to the collider on a tagged door (door gameobject tags are door, metaldoor, elevatordoor); Each

instance of collision between mouse and door calls a method from a script placed on that object

- hideItemText() – sets the text of the displayItemText UI to be an empty string
- displayItemtype(string item) – it takes in a string and returns a string that reads: “You found a <string item>”

- OpenDoorScript
  - Variables
    - Angles to rotate the door at
    - Audio for the door opening/closing
    - Boolean flag for is a door is open
  - Scripts
    - ChangeDoorState() just inverses the Boolean flag and is called by the FPS main character
    - Update() Plays the opening/closing sound and rotates the door open based on the length of the audio clip
- metalDoorScript
  - same as OpenDoorScript just had to have two separate because not all doors that I have are the same (axis wise)
  - Variables
  - Scripts
- NightModeCameraScript
  - Variables
    - Use – Boolean flag for is the night mode camera is in use
    - Two cameras, the main camera that users see the game world in and the camera with nightmode on
    - Light for the nightmode camera’s green color
    - camScreen - Canvas – UI for the camera screen
    - recSecs – the amount of seconds that the recording has been going for

        - recordingTime - Text to display the time that the camera has been recording
        - camBeep – an audio source that contains the audio clip for when the camera is turned on
    - Scripts
        - ZoomOut() – adjusts the field of view for the camera, increments it until it reaches 60
        - ZoomIn() – adjusts the field of view for the camera, decrements it until it reaches 10
        - formatRecordingTime(float time) – prints out a format string of the recording time
        - Update() – pressing the N key inverses the use Boolean; if use is true the camera sound plays, the recording time increments based on time spent open and allows users to zoom in and out if they wish. If false, the regular camera returns, working on finding a proper sound for when the camera turns off also would like to get a camera object model to show being opened
  - AIEnemyScript
    - Variables
    - Scripts
    - Could not get this to work as we did in class, had to scrap everything after multiple different attempts as well as following unity tutorials. Will revisit in the future though since AI is not that much of a biggie for me

- **Process Brief**

All and all this was a wonderful first foray into game design and game scripting for me. I learned how to use many of Unity 3d’s tool to build environments and make them interact-able through scripts. Everything from the coding language basics, to UI, Light, Asset usage and more were newly learned since I have never done game design before. I struggled a lot earlier on with using Unity and trying to manipulate objects to my desired outcome. I also struggled with trying to get an AI enemy working within my own script along with animations for that AI (so that they animate in a custom way, not in the way that they came with). The future of this game is just more fleshing out of the plot, adding more scary elements, I would like to get the elevator and AI animations working so that I can have players move around the environment. Also, following inspiration from the guest lecturer, I would like to introduce a shader for another type of view that is sort of foggy as if looking through the eyes of someone who has just been drugged (think Scarecrow from the batman type drugging). This shader would also play a role in my new plot feature that I am still thinking out but hope to figure out a way to have this weird, drugged images to be hallucinations and then at the end of the game, everything that the player recorded and thought they saw with enemies and such would be abandoned building recordings of mannequins. Sort of a hallucinations versus what the camera actually captures dynamic going on. I want to work on scripting in EventListeners so that certain actions or certain events in a location trigger location or instance specific events to help maybe create opportunities for jump scares and suspenseful moments.