**Assignment 4 – Game Design & Writing Reflection Combined | Ty Keith | 3270281:**

*This document includes both documents requested in the assignment to be handed in that includes:*

* *A document in which you discuss the research you conducted to create the game, which has the heading of “****Initial Research & Design Considerations”***
* *A design document in which you fully describe your game’s design elements, concept, plan, play and goals, which has the heading of “****Design Elements”***

**Game Project**

**Start Date: May 16, 2021**

**Initial Research & Design Considerations:**

**Sources:**

* <https://www.youtube.com/watch?v=pFSYsJE0vRs&list=PLZHx5heVfgEvT5BD8TgLmGrr-V64pX7MD&index=3>
* <https://www.youtube.com/watch?v=GB3pgc9PCmc>
* <http://programarcadegames.com/index.php?chapter=lab_adventure>
* <https://landing.athabascau.ca/discussion/view/8377027/assignment-4-question-regarding-text-files>

**Colossal Cave Adventure Wiki Page:**

* Developed between 75’ and 77’
* Gameplay is all text based with humorous and witty interactions with the user

**Elements Needed in the game:**

* Player (3 lives) with a goal to complete the game
* Interconnected rooms (minimum of 5 with an entrance and exit and 1 room with 3 exits)
* Items (min. 12 items)
* Actions to get to a final room
* Must have min. 5 characters from the book
* Actions words class
* Direction objects

**Alice in Wonderland story:**

Although I read this book in high school and watched the Johnny Depp movie adaptations I haven’t interacted with the storyline in many years. With that being said I will be refereeing to SparkNotes for key information and characters but will be making up my own storyline for my game. I decided that I would use ideas and characters from the book like request but that I would have my own scenario that Alice has to maneuver through. I’m sure the AE’s are tired of similar storylines etc… so this give it a different flavour to play around with but still adheres to the main theme of the project.

I decided to choose 9 rooms/scenes for Alice to work through with some being interconnected and some being unavailable after a certain choice is made. Attached with this document is a workflow diagram that I sed to visualize what’s happening where and when in the story line so that I could clearly know what I’m doing as I program the classes for the game.

I wanted to make a game that had consequences as alice progresses through each choice she makes effects how hard it will be to battle the bosses at the end of the game, this makes the user have to decided what risks to take and what risks to not take. I also introduced a mushroom item that has a somewhat random outcome which means the user takes a risk when they plan to choose that option.

* <https://www.sparknotes.com/lit/alice/>
* <https://heritagecalling.com/2015/07/03/6-places-that-inspired-alice-in-wonderland/>
* <https://en.wikipedia.org/wiki/Alice%27s_Adventures_in_Wonderland>
* <http://www.bookrags.com/notes/aiw/obj.html#gsc.tab=0>
* <https://landing.athabascau.ca/pages/view/8966214/assignment-4>
* <https://landing.athabascau.ca/discussion/view/7975091/assignment-4-control-class>
* <https://landing.athabascau.ca/discussion/view/7487753/assignment-4-external-resources>
* <http://www.javacoffeebreak.com/text-adventure/index.html>

**Testing and Issues while Programming:**

**May 22:**

I created my player class which when an instance is created sets up our player as alice with 3 lives.

**May 29:**

I started by coding one class at a time and test that classes before moving on. My first class that I worked on was the item class. I’m starting small and working big. This class was easy to code but I had a bug that took me quite a while to figure out. It was a scanner exception error. I googled and googled and it came down to a <scannername>.close() that I was using in my setter methods. Once I removed that from my 3 setter methods it worked. I’m not 100% sure why this is the case but it is and now that class is working as it should.

Next I created my action class so the user can choose their actions. I had no troubles when creating this class and it tested just fine.

When creating my locations class I used a series of if statements to test the user inputs but I let my python knowledge (to cheat that is cause you should use an elif in python but you can get away with not using it) which I use at work get in the way by stacking if statements which was letting my class to work but not properly based off my defined logic. Once I remember to add an else if

**May 30:**

Today I worked on completing the Characters class to which I had very little problem creating. I now need to focus on how I want to start implementing things in the actual game. I believe my control class will create each instance needed for each room/stage of my game and then my main will call those instances into play. I’m really confused on how I will implement this and have found no super helpful answer on the internet thus far. I am just am going to have to wing it and see how it goes/troubleshoot as I go.

I also created a robust inventory class that can ask the user to replace an item when their inventory is full. I made sure it can catch input errors and rerun its user input statement. I’m pretty happy with this class as it stands.

**June 6:**

Today I coded the entire control class. I made it so that the control class is the foundational class that drives the main class. It took about 4 hours code and test to make sure that it works properly. I also coded 99.9% of my main program but had some difficulties with an infinite loop that I need to sort out. I thought I had it but then I made it worse and after a full 10-hour day I can’t even think anymore so I need to teak a break and come back. I also added a get attackdefensescore method into my inventory so that I can use that and probabilities to determine my % of success chances when fighting the final boss. This is the areas that when I changed this an infinite loop opened up and I’m thinking I need to segment a piece that I had in a nested for loop so next time I’m working on this that is what I will try and then I’ll see if that works for me or not.

**June 8:**

Today I was able to fix my issues from the other day but it presented me with kind of what I believe is my last ‘main final one’. I can’t get my do/while loop to exit immediately when a condition becomes untrue. I decided to need to refactor my code an bit and introduce a switch statements that will break my code down into a more modular way. That way I can easily identify each area and step. I will work on that later but like the idea I have I think that will add more clarity and readability to my code while also making hopefully helping it work properly.. time will tell I guess.

**June 9:**

I wasted 2 hours today trying to refactor into a switch statement only to have very similar errors as yesterday. What I ended up doing was checking to make sure that every piece of my code was properly wrapped in if statements to check that the player still has enough lives to continue. That way the game is forced into the play again else statement when player1 dies. Once I got the game actually working, I noticed that when playing multiple games in a row the inventory would reset so I added in a new method to the player inventory class so that when a new game was started the inventory would be cleared and player1’s lives would be reset to 3 (the starting amount). Tomorrow I will work on formatting and start on the documentation part of the assignment.

**June 10:**

Today I worked on competing my txt dialogue documents so the story made sense as the player progressed through. I also went back through my code and started to comment where I felt I missed commenting and I also started to work on the very long documentation process. I decided not to go over every potential encounter in the documentation since they are linear and repeatable based off what scene is activated and an instance created. That will keep the documentation to a reasonable limit.

**June 13:**

Today I finished up the documentation for all classes and worked on my last little bit of debugging before submitting the assignment. I realized that the very last portion of my game needed an if statement added to it to check if the player is still alive otherwise even when the player died it would move them into the ‘game won’ sequence even when they died. I almost missed this and that could have been a big boo boo!!

**Design Elements:**

**Game Flow:**

I have created a workflow diagram that shows Alice’s options throughout her adventure to escape Wonderland. Below I will list each Character, Item, Scene(room) and action that Alice must use in order to traverse her escape from wonderland. The workflow diagram is only a reference and information in the finished product of my game won’t always exactly match what is in the diagram. The diagram is for purely visual purposed to aid in completing this assignment.

**Locations – Total of 9 (separate of locations class which will initialize locations as they are called into play:**

**Start Location:**

**Dungeon (Starting Point)** – This is where he game beings. Alice is trapped in the dungeons awaiting her trial. The Cheshire Cat appears and warns here that the King and Queen of Hearts are planning to execute her before the trial is even held. The cat offers alice the ‘Golden Key’ to escape the dungeon. If Alice declines the key she will be executed by the King and Queen of hearts and will lose 1 of her 3 lives. If she accepts the key her adventure will begin. Once her adventure begins she will have to choose a direction to go. She is unaware of which direction leads to where but still make a choice.

* If she chooses left she will go to the Croquet Grounds
* If she chooses right she will go to the road to the hatter
* If she chooses straight she will go into an underground tunnel

**Part II:**

**Croquet Grounds** – once Alice makes it to the croquet grounds she encounters the Knave of Hearts who is try to secretly eat a tart. Alice startles the Knave of Heart’s and the knave of hearts reacts in anger and attacks her. Alice can choose to escape but if she does she will not gain an advantages for later on in the game. If she chooses to fight she will have a 25% chance of losing one of her lives and having to attack the knave of hearts again or flee. If Alice beats the Knave of hearts she will be rewarded with a bonus item that will help her later on. She has a 25% chance of receiving the Vorpel Sword (+2 attack) or a 75% chance of receiving a croquet mallet. Once past the croquet ground alice will move onto the Wonderland forest.

**Road to the Hatter** – once Alice makes it to the Road to the Hatter she encounters the Mad Hatter who is caught by surprise and attacks out of fear without even realizing its Alice that stumbled upon him. Alice can choose to fight or to flee. Again, if she flees she moves on but has no chance to gain an advantage for later on. If she fights the Mad Hatter she has a 25% chance of dying and having to repeat the encounter with the Mad Hatter. By beating the mad hatter Alice will collect an advantage item. She will have a 25% chance of receiving the mad hatters hat (+2 defense) and 75% chance of receiving the white rabbits fan (+1) defense. Once completed the road to the hatter alice will move onto the sea of tears.

**Underground Tunnel** – once alice makes it to the underground tunnel she will encounter no resistance but will happen upon a strange mushroom. She can decide to eat this strange mushroom but will have no idea the consequences of doing so. If she eats this strange mushroom she has a 25% chance of dying immediately, and if she doesn’t die immediately she has a 50/50 chance of gaining a life or it having no effect on her. Once she continues through the tunnel she will happen upon a loot room on her way to the White Rabbits house

**Part III:**

**Wonderland Forest** – When Alice reaches the wonderland forest she stumbles upon The Caterpillar. He busy smoking his hookah and doesn’t even notice alice. She has the choice to sneak past or to sneak attack the caterpillar. If she chooses the sneak past, she will move on without acquiring any benefits of advantages. If she sneaks attacks the caterpillar she will have a 10% chance of dying immediately to the caterpillar but if she wins she will receive a power up defense item. That item has a 25% chance of being the mad hatters watch (+2 defense) and a 75% chance of being a shield made of sticks (+1 defense). Alice will then move onto the The White Rabit’s Hut.

**Sea of Tears** – When Alice reaches the see of tears she will stumble upon Tweedledee and Tweedledum who are playing at the edge of the beach. They do not realize she is there with all the fun they are having. Alice has the option to sneak past or to sneak attack the Tweedles. If she sneaks past she will move on without gaining any advantages but if she sneak attacks the Tweedles she will have a 15% chance of dying immediately but if she defeats the tweedles she will gain an attack item. She has a 25% chance of receiving a deck of throwing cards (+2 attack) or a 75% chance of receiving a pointy coral reef stick (+1 attack). After the Sea of Tears Alice will move onto the White Rabbits Hut.

**Loot Room** – When Alice makes it to the Loot Room she will notice a few items in the room that she but she only has time to collect one of the items. She will quickly scan the room and choose either a Little Cake (+1 Attack), Little Bottle(+1 Defense), or a strange mushroom (same properties as the mushroom in the underground tunnel) like what she encountered before. Once she chooses, she must now choose to move right or left out of the room. If she chooses left she will enter into the Wonderland Forest and if she chooses right she will enter into the Sea of Tears.

**Part IV:**

**White Rabbits Hut –** When Alice gets to the white Rabbits hut, she slowly makes her way up to it when she is ambushed by the King and Queen of Hearts. They realized that she had escaped and rushed to execute her. Alice has to fight for her life and has no options to run away. She will now have to use all the items she did or did not collect during her escape. The items will tally up her Attack and Defense (A/D) to better help her odds of defeating the King and Queen of Hearts. She will be an A/D score out of 4:

* 4/4 gives alice a 97% chance of defeating the King & Queen
* ¾ give alice a 75% chance of defeating the King and Queen
* 2/4 give alice a 50% chance of defeating the King and Queen
* ¼ will give alice a 15% chance of defeating the king and queen.
* At any given time during this fight alice has a 1% chance of realizing that she is having a dream and waking up by the river with her sister

Once Alice defeats the King & Queen the White Rabbit will stop hiding in his house and come to help Alice finish her escape. Once she escapes, she will wake up on her sister lap at the river and realize that it was just a dream all along.

**Part V:**

**River Bank** – Alice wakes up on the river bank after her scary and crazy dream and begins to tell her sister everything.

**Characters – Total of 7 (separate of Characters class which will initialize characters as they are called into play)**:

* **Alice** – main protagonist to be directed through the game by Alice
* **King and Queen of Hearts** – Antagonist that will be fighting to execute Alice for a mistaken theft
* **Knave of Hearts** – chance encounter that will attempt to kill alice for catch them eating the stolen tart
* **The Mad Hatter** – chance encounter that will become startled while looking for their watch and attack alice out of fear
* **The Caterpiller** – chance encounter that wont notice alice but will defend themselves if alice decides to sneak attack them
* **Tweedledee & Tweedledumb** – chance encounter that wont notice alice but will defend themselves if alice decides to sneak attack them
* **The White Rabbit** – None-interactive character that is Alice’s goal to get to in order to save her life

**Items list – Total of 12 (separate of items class which will initialize items as they are called into play)**:

* **Golden Key** – first item received to help escape the castle dungeon has a 0 attack & defense score
* **Vorpel Sword** – chance item the will give a +2 Attack Score
* **Croquet Mallet** - chance item the will give a +1 Attack Score
* **Mad Hatters Hat** - chance item the will give a +2 Defense Score
* **White Rabbits Fan** - chance item the will give a +1 Defense Score
* **Strange Mushroom** - chance item the will give benefit (+1 Life) or hurt Alice (-1 life) or do nothing
* **Little Cake** - chance item the will give a +1 Attack Score
* **Little Bottle** - chance item the will give a +1 Defense Score
* **Desk of Throwing Cards** - chance item the will give a +2 Attack Score
* **Pointy Coral Reef Stick** - chance item the will give a +1 Attack Score
* **Mad Hatters Watch** - chance item the will give a +2 Defense Score
* **Shield Made of Sticks** - chance item the will give a +1 Defense Score

**Inventory:**

* Single class that will be able to hold up to 3 items (the golden key from the Cheshire Cat and an attack and defense item.)

**Directions:**

* The player will be to move in 3 directions, with not all directions being available at one time.
  + Left
  + Right
  + Straight

**Game (main()):**

* This will be the main program that initiates instances of the game and works with all the other classes to help the user play a complete game where needed and also take the user input.

**Actions:** this class tracks the validity of the actions the players chooses and returns errors if they are entered**.**

* Fight – the user can use this to attack a chance character
* Escape - the user can use this to escape a chance character
* Sneak - the user can use this to sneak away from a chance character
* Sneak Attack - the user can use this to sneak attack a chance character
* Pickup – the user will use this to pick-up items as they progress through the game
* Drop – the user will use this to drop items if they choose during the game

**Control**:

* Creates instances of each room, items and reads in texts files for the character information. This class help to direct the game in a direction it should be going hence the name ‘control class’

**Player**:

* Class that initializes and tracks the players name and amount of lives. The name right now is statically set as Alice but in the future would not be hard to make it Dynamic and up to the player to set themselves

**Items**:

* Class that allows the initialization of items in the game and will be used by the inventory class and in the main class if needed.