Cloaked: Design Document

General Information:

Cloaked is an overhead view dungeon based adventure game, where you have a main character who needs to fight his way further through the linear rooms of the dungeon. Each room holds new challenges. The original inspiration for the game was the theme of claustrophobia. The wall bound system implemented instead of a static constant for wall positions was originally implemented with the goal of scaling level difficulty with maneuverable space (walls closing in as you progress through the game). Due to time constraints, this idea was never implemented.

The game story chronicles the story of a cloaked figure who is stuck in a dungeon and has lost his memories. This cloaked figure does not know anything about himself, his name, or how he even looks beyond the cowl of his cloak. His only goal is to persevere, as the dungeon he finds himself trapped in is formed in such a way that forward is the only option, and the player himself knows nothing about the setting or whether or not progressing forward is even getting him closer to an exit.

The game was originally going to be narrative driven, clarity being provided through the spoken thoughts of the characters analysing his current situation.

Controls:

- Move Up – W
- Move Down – S
- Move Left – A
- Move Right – D

- Slash (attack) – SPACEBAR

Pieces:

- *Player:* Your controllable piece. Can attack other pieces and is controlled using the controls listed prior in the document
- *Rats & Bats:* Very similar, but rats are represented as small grey blocks and bats as small black blocks. Bats can fly over traps, whereas rats cannot. Bats are also a little faster.
- *Knockbax:* These enemies knock the player back on contact. This can put the player in dangerous situations, such as getting knocked into a pit or into other traps.
- *Wizards:* These wizards are generally out of the players reach. They will teleport to random accessible positions around the map and shoot projectiles the player will have to dodge.
- *Spectres:* These enemies are impervious to traps, can fly through walls, and charge the player at high speeds. They disappear on hit, and reappear elsewhere.
- *Boss Wizard:* The final boss of the game. It is a large wizard with a lot of health. Its projectiles spawn new enemies periodically which adds to the challenge of defeating it. When it is defeated, all enemies it has spawned are also defeated.

- *Pits:* Large holes in the ground. If a non-levitating enemy or the player steps on it, their health drops to 0.
- *Spikes:* Traps on the ground that toggle on and off. When the spikes are retracted they are safe to walk over, but walking over them when they are out will deal damage.
- *Flamespitters:* A large flamethrower block that shoots out fire in a certain direction on certain intervals. The fire deals damage over time.

- *Walls:* Blacks of developer specified length that serve as unsurpassable terrain for most movable pieces.

Game Logic, Interface and AI:

The game is essentially formed of various states, there's the main gameloop state "PLAY" which calls upon the gameloop and then various other states that displays events such as the start screen, game over screen, credits, etc. The underlaying area (a room of the dungeon) remains regardless of game state, as even on the game over screens or title screens the player is still in the dungeon room. This is a deliberate choice added in for effect.

The game loop essentially does nothing but loop through the lists of each individual types of pieces in the game and call their appropriate update methods. As such, when a level is complete the "runLevel" code is run one time, which is code that points to the requested levels "setup" function. These are functions (found in the level tab of the project) devoted entirely to level design. To add pieces to a level, all that is required is that this piece be added to the appropriate list (which is reset every time a level is completed or the player restarts). When a piece is added, this means adding to the list a "new" piece created with it's appropriate constructor. The constructor usually outlines the details such as it's starting position, for traps the state and timers relative to the trap, etc. The object instances will then take care of updating themselves appropriate through the calls made in the game loop. For this reason, adding to the game is relatively simple.

Inheritance is used to simplify the wizard code as there are a few different types of wizard, as well as to create projectiles.

The general state of enemies is an offensive one in which they chase the player. This is achieved through trigonometry, essentially using the vector of velocity and modifying their x and y components so that they will be moving towards the player at a constant speed of "maxSpeed". The main update loop for the enemies is the movement loop, and changes of movement state will affect how they move and thus how they react to certain things as well.

The latest addition to the game is the level events loop, which like the runLevel code, is a grouping of level specific codes. This said, this particular one runs every update in the gameloop, allowing for events such as checks to see if a particular event unique to that level has been triggered.

Design Elements:

While currently there is minimal replayability in the game, the player will initially find fun discovering the different interactions of pieces in the game.

Variety of encounter was added through a variety of different enemies, revealed gradually as the game progresses. The player gets to discover the various properties of these enemies upon encounter, and receives awards akin to illumination when he figures out how to deal with them.

The fun is primarily agonistic, especially if you ignore any seems of mimesis that could be drawn from the story. The player's ultimate goal will be to beat the game, playing through the various levels until the end. The levels grow quite challenging towards the end of the game (in this case the last two) but have been balanced in such a way that they are very possible to complete. This will award the player through the emotion of Fierro when the final boss is finally taken down.

While there is a relatively high degree of performative choice, especially towards the end, the design of the levels lends deliberative choice to the player as well. As an example, the player must generally think quick on his feet and determine the "safest" path to get to the end of any given level. There are often a few ways to choose from, and the way the AI in the game functions each individual decision will create a unique gamestate to evaluate.

In cases like the later levels, deliberative choice is also achieved found through the need for strategy. The difficulty in these later levels raises not simply because more performative action must be taken, but is in a large part influenced by the sudden need for critical thinking. One cannot beat the double spectre level without giving some though to their strategy, and the randomly generated level terrain will add some variety to this strategy, making it so strategies have to be slightly modified for every play through of the level. Defeating the boss level also requires some form of strategy. The choice is up to the player, a player with more performative prowess can opt to risk head on assaults, while other choices might include focusing on dodging and waiting for the boss to come to you. When non-wizard enemies begin to spawn, one can consider it a better idea to stop and defeat them before focusing on the wizard, to try and ignore them by staying on the move, or simply to split attention between them and the wizard when appropriate. All of which could be good choices depending on the players level of skill and general playing style.

Some resources have to be managed in this game, such as health and arguably more importantly the slashes. The slashes are on a scaling cooldown timer system, meaning the average slashes the player will have at his disposal will likely be lower than the maximum of 3. This makes every slash count, particularly in the later levels where there are more enemies to deal with.

The game really shines in the more difficult levels, and the player is most likely to find fun in these such levels, for the fact that managing resources starts to play a big role in the game at this point as using the slashes at your disposal in the right times becomes crucial to survival and success. The player will feel a lot more accomplished upon succeeding these such levels comparatively to previous levels that may seem quite the bit easier.

Balancing:

Various changes were ultimately made to help with the balancing of the game. Level design was a focus to the game, and often the AI was adapted to try and fit the individual level design ideas a little better (such as level 4 where a wandering state was added to make the knockbacks more threatening).

Controls were also changed up. The reason for this being that original placeholder controls such as click to slash were causing issues in balancing between players. Players with better mouses that responded quicker were given an advantage. It was also not a very good control choice where processing is involved, as the processing platform has always appear particularly sluggish where the mouseClicked() function has been involved.

Another important form of balancing was the slash balancing. Originally the slashes had too long a duration which made many parts of the game unnecessarily easy and took away from the need to keep track of this resources, as a result making the game easy and making the very idea of enemies a pointless thing due to the ease at which they could be defeated. The solution was to find a god compromise where the slashes were long enough to fight off enemies properly while beings short enough that they had to be used sparingly.

Finally, boss balancing and balancing of the spectre level was a huge focus. The spectre level is quite challenging but is very possible to beat with proper management of one’s slashes. This required balancing as originally the spectre’s speeds were either too high to make to allow for proper flow, or one single spectre was too easy to overcome. The boss also needed a lot of balancing. Originally there was no cap on wizards and the spectres had a chance of spawning much sooner. This was incredibly hard to manage, eventually there were too many bullets to properly dodge. Another compromise made after adding a cap to wizards spawned and changes spectre spawns later on to bat spawns was that of lowering its starting health to a more acceptably range. While arguably possible at higher health too, fatigue set in pretty quick when after spending a lot of willpower to dodge the other wizards you noticed that the health bar was going down really slow. For the sake of player flow, this had to be changed, and the end result is a cap of about 10 extra wizards spawned, a health of 120 which, and adding a chance of spawning bats after 50 health has dropped and a chance of spawning a spectre when there's about 1 hit remaining on its health.

Level by level:

The following is a level by level design journal that gives some insight into why the levels were created:

Level 1: This level was designed to allow the player to explore the games mechanics. To a returning player, this level would be incredibly simple, but it provides a good battlefield against the two different basic enemies of the game in which the player can learn to use his offensive techniques. As the first level it is also where the player learns he must go to the top of the screen to complete the level.

Walkthrough: To beat this level, simply walk through it. You can opt to just rush the top of the screen or fight off the weaker enemies quickly and then finish the level.

Level 2: This level was designed to introduce the player to the knockbax, and also show different environmental interactions possible between various enemies and their surroundings. As the first introduction of pits, the player discovers via the choice of placement for the initial positions of the enemies that if an enemy such as a rat falls into the pit, it dies. The player also realises bats can fly over such pits, and that he/she cannot fly over the pits.

Walkthrough: Head up towards the middle and push back or dodge around the knockbax. Proceed to try to knock the knockbax into the pit. After knocking the knockbax into the pit, the level's end will unlock. Proceed to complete the level.

Level 3: This level introduces a variety of traps and wizards. It is again, more informative in nature than anything, the chances of failing to progress through this level are pretty slim.

Walkthrough: Simply walk up the middle alley, the damage over time is much less dangerous than the spikes. Keep moving and you should dodge the wizards with relative ease.

Level 4: This is a level based off the threat of knockbax. There are many instances of flamethrowers, and if the player gets knocked around too much by the knockbax they could end taking a lot of damage from flamethrowers. The challenge of the game starts to grow at this point and the player discovers a need for planning a safe path through.

Walkthrough: When the knockbax are in a wandering state, they will only snap out of it if you get relatively close, so try to stay far from them and dodge the fire bursts to make it safely through the level.

Level 5: Level 5 is a level featuring just about everything the game has to offer in one place, also presenting a good challenge. The player has a few different paths he can take, facing the spikes may seem more tedious than going through the hoard of enemies head on, but the spikes can potentially deal more damage than the fire can if handle improperly and some of the enemies such as the spectre will pose a threat either way. This level is the introduction of the fast moving spectre as well.

Walkthrough: Sneak away from the main pool of enemies off to the right. Wait for the bottom-left most spike to be deactivate and walk through that opening to get passed the traps portion. At this point the main threat will be the spectre so fight it off whenever it is near.

Level 6: This level displays a level of alleatory appeal when it comes to variety in the game. This level's walls randomly generate, and it also serves to show an interesting property of the spectres that may or may not have been prior noticed by the player, that they can go through walls. This level requires a fair amount of deliberative thought when it comes to planning the route to take and deciding on how to go about the threat of two spectres at once. The general strategy is to maximize the effectiveness of the slashes and try to space out the attacks so that the cooldown always has time to reset before the spectres reach you. One of the harder levels of the game, but has been well balanced and beaten numerous times in testing.

Walkthrough: Try to get yourself into a position where with one swing you can switch directions mid slash to hit both spectres at a time, maximizing the return on each slash. This may require you to move back or away from the goal in an attempt to buy time and make sure they draw closer together to pull off such a hit. Make your way gradually towards the goal keeping this strategy in mind and you will likely find yourself defeating the spectres before the end of the level.

Boss Stage: The final stage of the game: in this stage the player must defeat the boss so that he can unlock the exit of the stage. The boss spawns wizards in the trajectory of its own shots, and is essentially a larger wizard that does not take into account the players position (which is important so that it has a chance of spawning close enough for the player to attack it). The boss uses random number generators to determine the frequency at which it spawns new enemies around it. As such, the boss fight gets progressively harder as enemies are spawned in, and a greater variety and number of enemies can be spawned as it’s health depletes. This truly is the most agonistically based level in that the player’s goal is to defeat a powerful opposing force. When the boss is defeated the way opens and the level clears of all enemies and shots it may have spawned so far.

Walkthrough: In the early stages, when there are no more than 4 wizards, focus on getting in close and dealing damage to the boss whenever possible. When the number of wizards onscreen begins to grow, switch to a defensive strategy and simply dodge incoming projectiles, waiting for the boss to teleport closeby to deal more damage. When the boss appears one hit away, use the health you’ve saved by playing defensively to go in for the final blow, as you do not want to risk leaving him alive long enough to spawn a spectre as this will greatly decrease chance of survival at this stage. As an extra hint, when he appears close to you, be aware he will soon after fire a projectile your way so try to dodge around this as soon as you see it coming as it will fire in a linear path based on where you were when he fired it.

Known bugs:

There are a few known bugs worthy of mention. The first, the “enraged spectre” glitch. This one is usually not a problem as it takes very specific conditions for this to come into play. It seems to only happen if somehow the spectre damages the player at the same time as the player damages it, causing some type of state mismatch that leads the speaker to turn red permanently and become invulnerable. This is most common in level 5, as it generally only happens in the very specific situation where the player is knocked back into the spectre after having hit the spectre with a slash. Of the few state mismatch type glitches that occurred in the game, this one is relatively rare and harmless so has not yet been fixed as other bugs were a priority.

There is another bug relating to the randomly generated terrain in level 6. For whatever reason, while terrain is not supposed to spawn on top of the character due to conditions set, it occasionally does anyways, locking the character in place. The player can simply wait for a game over from the Spectres who can go through these walls, but this helplessness is of course unwelcome. It is also possible, though unlikely, that 3 separate walls box the player in. This is extremely unlikely so it has not been addressed, though it could be relatively quickly by putting a safe path where walls can never spawn. In practice, the likelihood of getting boxed in does not warrant the drop in visual appeal of the player seeing a clear path of the same shape appear every time the level loads, defeating the purpose of the randomization.

In level 4, some of the bounds on the knockbax are glitching. It is difficult to discern why this is, most likely pertaining to some reaction between the auto-displacement of their position exceeding the boundaries and the fact that they are switching to the wandering movement state instantaneous on spawn because of this. It is very difficult to discern what the problem is for sure, let alone fix it with limited time.