Estimated Workload Justification

This document is an extension of the Estimation Plan excel spreadsheet, which justifies the amount of time needed to complete each task within the allocation list.

# Minimum Viable Features

The tutorial is one of the parts which may take a while, as it introduces all of the primary concepts of the game. Each group member will probably spend around 30 hours on just the tutorial alone. This will include designing sprites, associating sounds to particular actions, coding mechanics, troubleshooting errors and more. As this is a hefty task, sections of game design will be distributed and allocated appropriately.

The player character and how they move would probably not take too long. This is because this part involves the design of the player’s sprite, how they move, and implementing collisions with the player character and the game world.

The first level, which is technically the second level, would probably take a far bit of time to do. Enemies, items, traps and level design will be implemented here, meaning this process would probably be the most time-consuming. This would require level design, item effects, sprite design, sound design, and more. This level is essentially the culmination of the entire project’s work, and has much overlap with the other minimum viable features. For these reasons, this would be the most time-consuming part of the project, and may take around 20 hours. However, it might also be comparable to the tutorial

The health mechanic should not be a time-consuming process. This would above all else require coding, as this mechanic is concerned primarily with how threats in the environment can influence a character’s health. Some design would be required here, as the hearts themselves would need to be designed. Depending on the allocation of work, this could take anywhere from 2 to 15 hours per group member.

The traps might take a fair bit of time. The sprites for the enemies and obstacles would need to be designed. These are not restricted to the static sprites themselves, but also when they are attacking, moving, or in the case of the traps, whether they have been triggered. These also include coding trigger events, the general process of creating something through code, and how it interacts with other mechanics like the HP. The spider web slow effect trap would also take coding because it applies a slowing effect to the player character. Therefore, the enemies/obstacles design and coding might take around 5-20 hours based on work allocation.

The initial death handling refers to when the character dies or reaches 0 health. They are restarted and put towards the beginning of the level. This term also refers to the difficulty of the enemies and obstacles, as it is associated with how the player dies. This section itself would probably take around 2-5 hours per person. The primary parts that would take the most work are the coding and some general theory surrounding the difficulties of the traps and enemies, and seeing that implemented into practise.

# Extended Features

The second section of design involves the extension of new levels, challenges, obstacles and fleshing out the lore of the game. Although this also needs to be bug tested. For example, the player character should go through the “new” levels a couple of times and try to break the game in order to root out any glitches or bugs. The time spent on this section is entirely subject to change, because it depends how much time the group has left to work on it. However, considering the remarkable amount of progress the team is making, we could see at least 15-20 hours worked on this section per group member.

The checkpoint system refers to what happens when the player dies. Like the design of the health mechanic, this probably would not take too long. However, once again, this is contingent on whether the group has enough time to work on it. If they do, the following can be expected. With proper team collaboration and contributions, this should not take more than 5-15 hours per person. We would need to bug test whether the checkpoint system can glitch, like if an enemy get’s stuck in the respawn point and blocks it or unfairly hits the protagonist upon spawning. The necessary coding and the process that involves would also need to be underwent, which is where the bulk of the time could potentially come from.

As this is another extended feature, this again depends on whether the group has enough time to do this task. However, if it were to be done, it would probably involve the implementation of a gliding cape, a shield which gives one extra life to the player character, or magical boots that allow double jump. In order to bug test this, we would need people within or outside of the group to play the levels with the items over and over to see whether there’s an internal inconsistency in the game. This could take 5-15 hours per person depending on work allocation and optimisation.

The last extended feature is the checkpoints. To avoid repeating myself, refer to the potential time allocation on this above. This would require setting checkpoints in specific parts of a level, across the levels. For this to be bug tested, a player would need to see whether certain combinations of items or pathways cause a glitch in the checkpoint system and if it effects the reliability of loading checkpoint saves. This could take 2-15 hours per person as well.