

## **Description of Plan**

We are designing a game using Java and therefore will be utilizing lots of objects, which means we must have good class diagrams and leverage Java's OOP. In addition we expect our game will undergo a lot of evolution as we proceed so our group intends to adopt an agile process model. We chose this model because we do not know if our requirements will change suddenly and with agile you can adapt easily. We intend on utilizing scheduled scrums after class and hope to be able to assign work evenly, utilizing code sprints in order to hit deadlines. All members will review and assist each others work via Gitlab. Our main mode of communication will be through our group chat on Facebook. After the planning phase we hope to get out a prototype, which will adjust and evolve as we see fit into a full release in April. We divided the bulk of the work from phase 1 as follows:

1. Kevin – Description of plan and Game
2. Jasim – Use case diagrams
3. Vera – Mock up of UI
4. Brian – UML diagrams

## **Description of the game:**

The user must guide the hero of the game, Prometheus, through a dark labyrinth and gather all the fire as they make their escape back to the heavens. The player must avoid the moving enemy, the Minotaur, whose contact means certain death. This moving enemy will always take the shortest path to the player but will move slower. Since both player and enemy are moving at the same time we will use multi-threading. Neither player nor enemy can pass through walls in the maze.

The player must also survive entering rooms which may have painful traps that will lower their score and if this score goes to 0 the player loses. Moreover the player may decide to gamble on opening treasure chests which will either give bonus points or turn out to be a trap. One idea we have is to also add the idea of keys and doors which would allow the player to move through what would normally be a door/wall. The Minotaur would not be able to follow the player through these doors and so this may allow for some interesting escapes.

The player begins the game at the entrance of the labyrinth, an  $n$ -by- $n$  board that has one exit. The size of the board may change as levels advance or stay same, this remains to be determined, what is certain is that the perimeter is surrounded by impassable walls except for the entrance and exit. Upon entering the player has a complete birds eye view of the 2 dimensional map and we might have is so that the view of the player is determined by the score. As time passes this fog of war would come closer to the player, when the player moves to a spot with a reward the score increases and so does the radius of sight, a loss in score would have the opposite effect. This would likely be the most difficult thing to implement and would be implemented last as it is just something we thought may be fun and thematic.

We hope to be able to generate random 2D grids so that the user has a unique experience every time however the first prototype will likely start with smaller test grids and build up from there. We hope to have 2D sprites for all objects which will be invoked by the draw method. We do not know what will make a fair game at the moment so ticks are not well defined. At the moment we are thinking every millisecond is a tick and player can move 1 room per millisecond and Minotaur would be a fraction of the speed of main character to make things fair.