*Impossible3*

<u>Software Design Description Document</u>

<u>7.0    Detailed Design Specification</u>

<u>7.1    Introduction</u>

*Impossible3* is a roguelike dungeon crawler that contains multiple elements and designs. The following document will contain design descriptions for the CSCs and CSUs.

<u>7.2    Detailed Design Description</u>

These next sections revolve around the descriptions of *Impossible3*'s CSCI (the game itself) and the descriptions of the details of the design. There are two CSCs, the game in play and the menu.

<u>7.2.1   CSC 1 Descriptions (Menu)</u>
   ○   User can pause the game and enter the Menu any time. This is also the landing page for our application.

       <u>7.2.1.1  New Game:</u>
           ○   Start a new game.
       <u>7.2.1.2  Continue:</u>
           ○   Pick up where you left off. Due to perma-death, this option disappears upon losing one's team of three.

<u>7.2.2   CSC 2 Descriptions (The Game)</u>
   ○   The game itself! Like real-time chess.

       <u>7.2.2.1  Cooldown System</u>
           ○   A player cannot always be a attacking, only when the timer is "full" can a player attack.
           ○   A player cannot always be moving, only when the timer is "full" can a player move.
           ○   These restrictions also apply to AI units. In addition, these restrictions apply singly to individual units and not just the player or the AI itself.
       <u>7.2.2.2  Board/Movement System</u>
           ○   Board is highlighted to show valid spots for movement.
           ○   Board is highlighted to show valid spots for attack.
           ○   A unit can obstruct another unit's attack, and a unit can obstruct another unit's range of motion.

### 7.2.2.3 Enemy AI
- *Impossible3* contains different AI's attached to varying enemy types. Each enemy type should act in a different manner.
- A ranged AI would stay stationary while lobbing arrows at the player's team.
- A melee AI would move towards the player's team and attempt to the three characters down.
- AI would generally not attempt to be stealthy or flee from the player's characters.

### 7.2.2.4 Combat (Health Deduction)
- Removes units from the board upon death.
- Ensures that armor and other mitigating factors are taken into account when dealing damage to units.

## 7.4 Detailed Interface Descriptions
- The entirety of the game is written in the Unity game engine. Most of the game is made with C# scripting, so all communication is handled by the engine.