Dungeon Diver

<https://github.com/comp195/spring-2021-final-project-dungeon-diver>

Ethan Lo: [e\_lo2@u.pacific.edu](mailto:e_lo2@u.pacific.edu)

Last Revision: 5/1/2021

**Evaluation Procedure**

In order to make sure my game properly functions, it is important to regularly test and check every new change and addition to the project. Unity, the game engine I am using, allows you to easily test your code.

For my type of project, it’s as easy as pressing a button, and watching as Unity tries to compile your game and execute it. If the game does NOT compile, Unity will let you know and even point out the areas of error. For example, it will print out a message letting you know about a compile error in your C# code, or in your Unity developer settings.

Due to this ease-of-access mechanic letting me test and compile my game, I was able to make sure that my game would successfully compile and work. Every time I added a new feature or new snippet of code, I would constantly compile and save my work.

Of course, if one sees my game, they will clearly see that a lot of things are very “rough” around the edges. However, I believe that I was successful in developing my game to the very least where the core gameplay loop is functional. Each time you start the game, it randomly generates rooms. It randomly places the boss at the last room generated, and it also successfully spawns the rest of the enemies. The animations are also a little rough, but they still function and work as intended.

Players can kill enemies, and they in turn can kill you. The Main menu works, although the volume option does not. A pause menu is also functional in game by pressing ESC.