Programming Project #6
CpSc 8700: Data Driven OO Game Development
A Playable Game
Brian Malloy, PhD
November 16, 2016

In order to receive credit for this assignment, your solution must be submitted, using the `handin` command, by Noon EST, Friday, December 2^{nd}. You may receive 90% of the grade if you submit the project by 8 AM on Monday, December 5*th*.

The goal of this project is to produce a playable game that incorporates a coherent theme and reaches a **conclusion**. The conclusion might entail the player winning or losing, or reaching a desired goal. You should make the final project as robust as possible, which means that it doesn't crash and doesn't have memory leaks or errors reported by Valgrind. Of course your game should be data-driven, with the important constants and parameters read from an XML file.

The projects in the course have diverged and many of you have different goals and features. Nevertheless, your final game might include the following features or be replaced by comparable features, including:

1. Video (mp4 format, ≤ one minute). You can use:
   (avconv -r 60 -i <username>.%4d.bmp <username>.mp4)
   or you can use simplescreenrecorder.
   Try to show off the key features of your game and show the goal being achieved.
2. Goal: The game should have a **stated** goal (HUD, readme, . . .).
3. HUD shows how to play, stays at beginning for ≈ first 3 seconds
4. Sound and Music
5. Collision Detection
6. Explosions (chunks or frames)
7. Projectiles
8. AI
9. *god* mode
10. restart feature
11. A Demonstrated Object Pool. Provide a HUD option, using F2, that illustrates your Object Pool.

There are examples in the repo that will help with these requirements:

```
4160fallAssets-2016/examples/sdl/sound
4160fallAssets-2016/examples/sdl/collisions
4160fallAssets-2016/projects/5/explosions
```

Your assignment will be tested on a Linux Ubuntu platform using gcc or clang; however, you should test your project on several different platforms and it should be independent of platform and language implementation. Compress your project directory, including the code, Makefile, media, and README, using either tar or zip, and submit your project using the web handin command.