

Programming Project #4
Adding a Player Object
CpSc 4160/6160: Data Driven 2D Game Development
Computer Science Division, Clemson University
Brian Malloy, PhD
October 18, 2018

To receive credit for this project, your solution folder must be compressed and submitted to the web handin bucket by 8 AM, Friday, November 9th, 2018. If you cannot meet this deadline, you can receive 90% of the grade by submitting your project within three days of the due date. You will now begin to build a playable game with: a player object, HUD, AI, and collision detection. Your tasks include:

1. Incorporate a **player** object into your animation (encapsulated); use **asdw** to control the player. I will provide 3 different code samples illustrating possible design choices; however, you must choose the approach that you prefer and tailor the sample code to suit your game plan. The three choices are: (1) Inherit the player from **Drawable**, (2) Inherit the player from **MultiSprite**, (3) Use *composition*.
2. Collision detection and AI. Some of your sprites should be smart and react to the player when the player gets close. You must use the observer pattern to enable the player to notify NPCs of its position; use **notify**, **attach**, and **detach**.
3. At this point you should be able to design and build your own classes. Design and build a class that implements a heads up display (HUD). One approach is to use the SDL draw facility to make a HUD that is reconfigurable through modifications to your XML file. The player should be able to toggle the HUD with F1 and the HUD should display (1) the fps, and (2) info about how to move your player object so that we can test your game.
4. Your name printed clearly (font color/size) in lower left screen.
5. **video**: submit a short video, 30 seconds or less, demonstrating your games features. You can either make the video with a capture program such as *simplescreenrecorder* or use F4.
6. In your README for this project, include a paragraph that describes the game level that you will build for your final project. Provide some details about actions in the game, your sprite source, how you will keep score, and how the game will conclude.

The course repository contains code samples to help you get started in implementing a player and collision detection. You will need to design and implement a class for the HUD and figure out how to use the *observer* pattern to implement AI. As you build your solution for this project, strive for proper C++, and good object oriented principles. Your goal should be to write classes that “take care of themselves.”

If you would like to work synergistically with a partner, this project can initiate your collaboration.

The Light at the End: Project #5 will entail incorporating projectiles and shooting, explosions, object pooling, sound, and music. The final project, Project #6, will require that your game reach a conclusion, and you incorporate more pizzazz. You could incorporate a menu, a health meter that appears or disappears at strategic times in the game, *Painter's Algorithm*, etc.