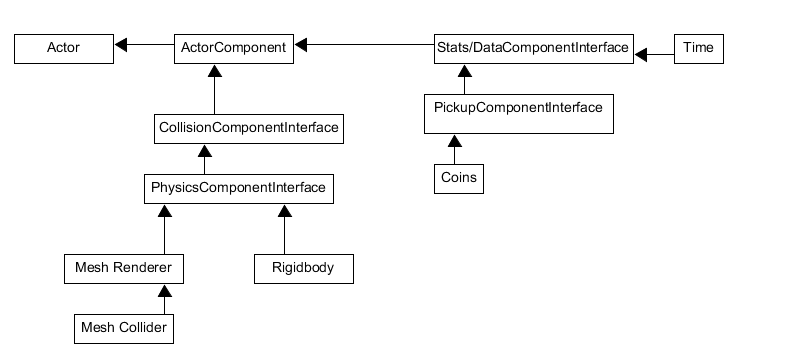
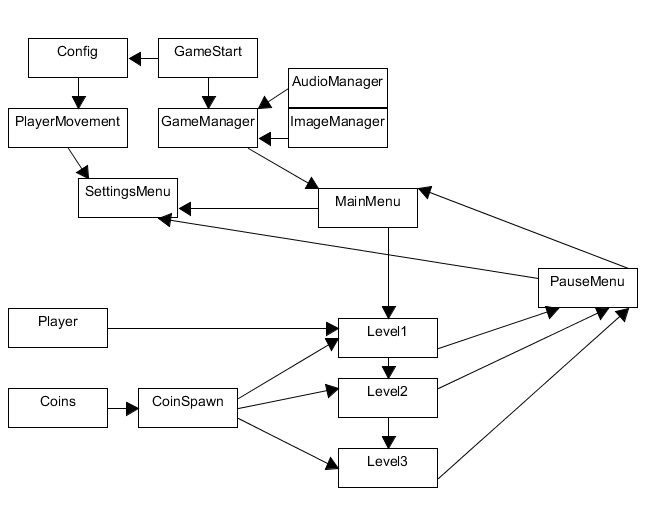
We have agreed to use standard C++ coding conventions in order to maintain a uniform code style and look. This will prevent parts of the code standing out from one another and will maintain a more cohesive look as a whole. Our project structure will be as follows :

The actors folder will contain all of the code relating to the actors in the scene. Assets that we need to use will be placed in the assets folder. Documents relating to the project such as the GDD will be in the documents folder. The Game folder will be where the exe of our game is placed. Source is where all the main code of the program is. Temp is for the temporary files created during the coding process and the Test folder is for tester code if needed such as extended time for the levels.

The requirements for this game will be extremely low as it is just a simple 2D platforming game.

The following is the component architecture for the game:

UML Engine Class Structure:



Main Loop:

The game loop will first call the game manager to check if the player has been spawned into a level or not. After the check complete the loop will load the time remaining counter, coins remaining, and the player into level 1. When all of the coins are collected and the time remaining is >0 then the player will be allowed to move into the next level. The cycle repeats until the player has beaten all the levels in the game. After the levels are completed it will send you to the main menu where the player has the option of repeating the levels again.