**CLASSES**

**NOTE: All of these classes are in the project folders please look through them all to have a full understanding of the classes. Children are not mentioned here because they are talked about in the parent class.**

**Application:**

The Application class is responsible for putting everything together. It is the only entry point for the engine. It inutilities everything and shuts it down later as well as runs the engine main loop. Initialization will create each of the engine class variables to get the engine going. It makes sure all the parts of the engine initialize correctly before proceeding with the game loop. Also checks computer specs and makes sure that everything can be run without issue. After all checks come back perfect it starts the main loop and loads game world actors and actor components. Main loop includes creating delta time each frame and running update for our scene (which is all our objects) also includes rendering for our scene (from graphics class). The application class also checks to make sure the window has not gotten the close event from the window class if it does proceeds with shutdown. Shutdown will run shutdown for all parts of engine. It will ensure they all close properly if not, give diagnostics and close anyways. Finally this class removes its process form windows.

**Window:**

Initialization will bring window up and make it ready for drawing, producing sounds and sending raw input to input class. Shutdown will remove it for drawing, sound and will stop sending input. Also, this class is the first thing to initialize and last thing to be destroyed. Also pools raw windows key events to the input class.

**Input:**

Input class contains enums for keyboard keys and general input. These values are used in actors and components that require player input. Initialization will check to ensure it is receiving input from window class. It will also test inputs to make sure they work and aren’t receiving inputs that lead nowhere. Shutdown will stop receiving inputs and make itself unavailable to components and actors.

**Save:**

The save class will have functions meant for saving game state. It will have a few children all used for saving game state via different methods. Children include cloud save for saving files to the cloud, text save for saving games in a blank text file and XML save for saving game state in XML format.

**Graphics:**

Graphics will do the actual drawing to the window class. It will have children that use different technologies such as DirectX, Open gl and Vulcan. Initialization will load the base graphics portion that will allow DirectX, Open gl and Vulcan to function in the application. Shutdown will stop drawing all items and shutdown the base graphics portions. This class will just implement base graphics class functions that draw to the active window.

**Sound:**

The sound class will play sounds from the game. Actor components will be able to call functions of this class which then play sounds. This class plays sounds via the window so it needs that to work. Initialization will make sure the window is ready to play sounds as well as initialize all its members. Shutdown will remove its members and stop all sounds from playing as well as return an error if sounds try to play from here on out.