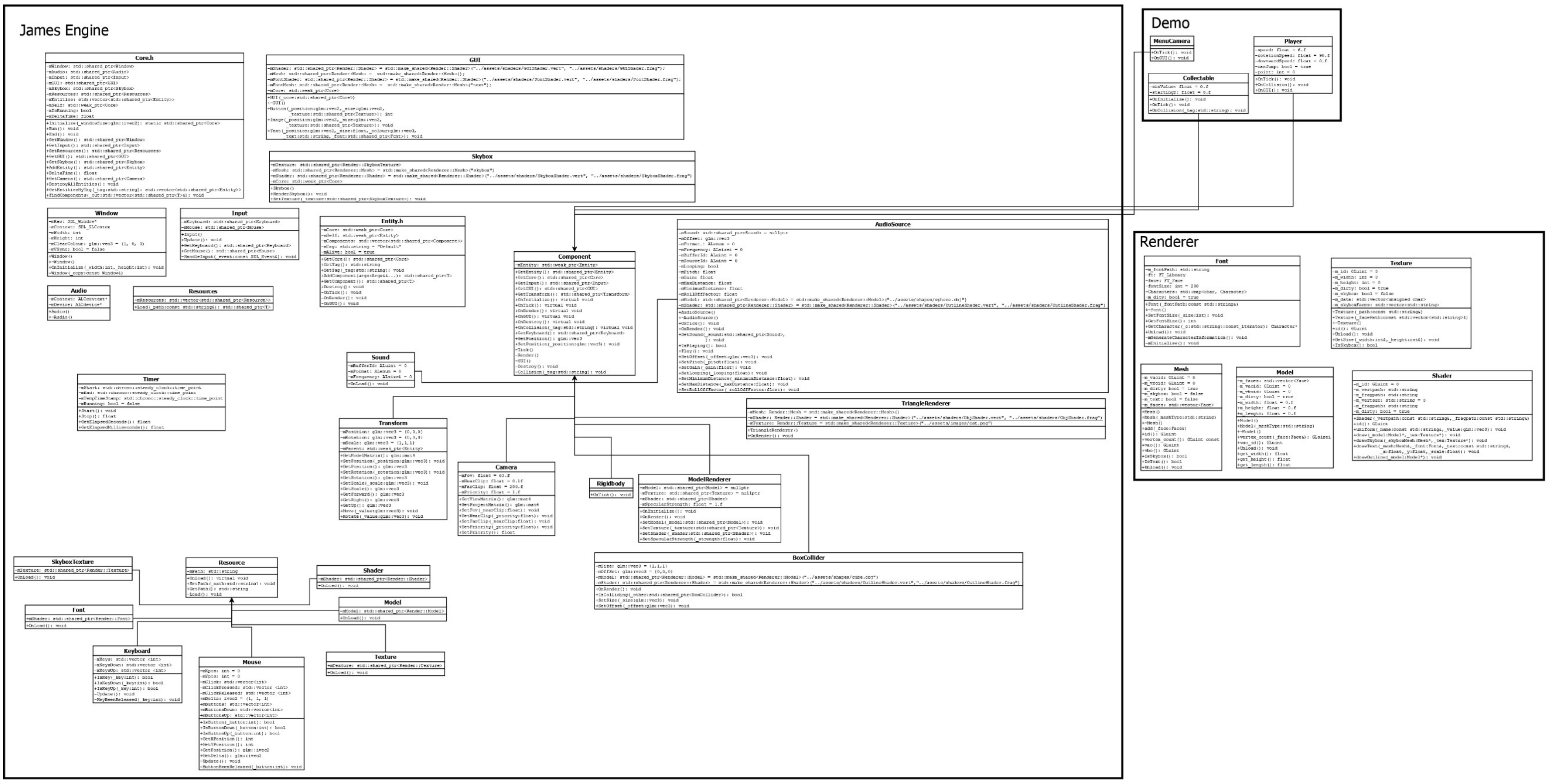
Game Engine Programming

Overview

I have created a light weight, easy to use, 3D engine which allows the user to add entities and then add existing components to those entities, as well as creating their own components. The goal is to make something modular enough that it can have more features added to it in the future. My components include a transform system with parent transform capabilities, a model renderer, an axis aligned box collider that calls OnCollision() in the entities components when it has collided, a rigidbody which provides a response to a collision by moving the entity accordingly, an input system that handles mouse and keyboard input, a resource system that loads resources once and then reuses the previously loaded resource, a skybox that can be set by the user, ability to change the lighting in the scene (limited to one light), a sound system that handles directional sound as well as sound fall-off, and a gui system which has text integrated. I have also include debug information if compiled in debug mode in the form of displaying the box colliders and the minimum and maximum distance for a sound source.

Design



My engine mirrors Unity in terms of the entity component system and the pre-existing functions that will be called at a specified time in the scripts (Unity's MonoBehaviour). The main game loop runs inside of the core class, calls the correct component functions on all of the components on all of the entities at the correct time. Inheritance has been utilised in Resources and Component to allow for easy expansion in the future with different kinds of resources and components, as well as allowing the user to create their own components. Smart pointer have been used wherever possible ensuring memory safety. In terms of the programs architecture, core is the main class where all the entities are stores, as well as all of the subsystems. The entity class has a pointer to core, and the component class has a pointer to entity, meaning that you can access core from any component.

Conclusion

My engine does what I set out to do with no obvious faults. I am pleased with the extra debug information and the text implementation in the gui system. I feel as I have also taken steps to make the experience easier for the user, things like gui elements being centred at their position allowing for easy stacking (such as text on top of a button) and customisability to do with the lighting in the scene. A feature that I am missing in my engine is shadows. Without them everything looks fairly unrealistic, so this would be a very good feature to add. In future I would like to add different collision shapes, and possibly mesh collision, which would benefit from a base class for all of those collision shapes to inherit from. I also personally struggled for a while to find models online that were consisting of only 1 model file and 1 texture file, so the ability to be able to have multiple model and texture files condensed into one model in the engine would be very useful.