Hyunsirk Choi

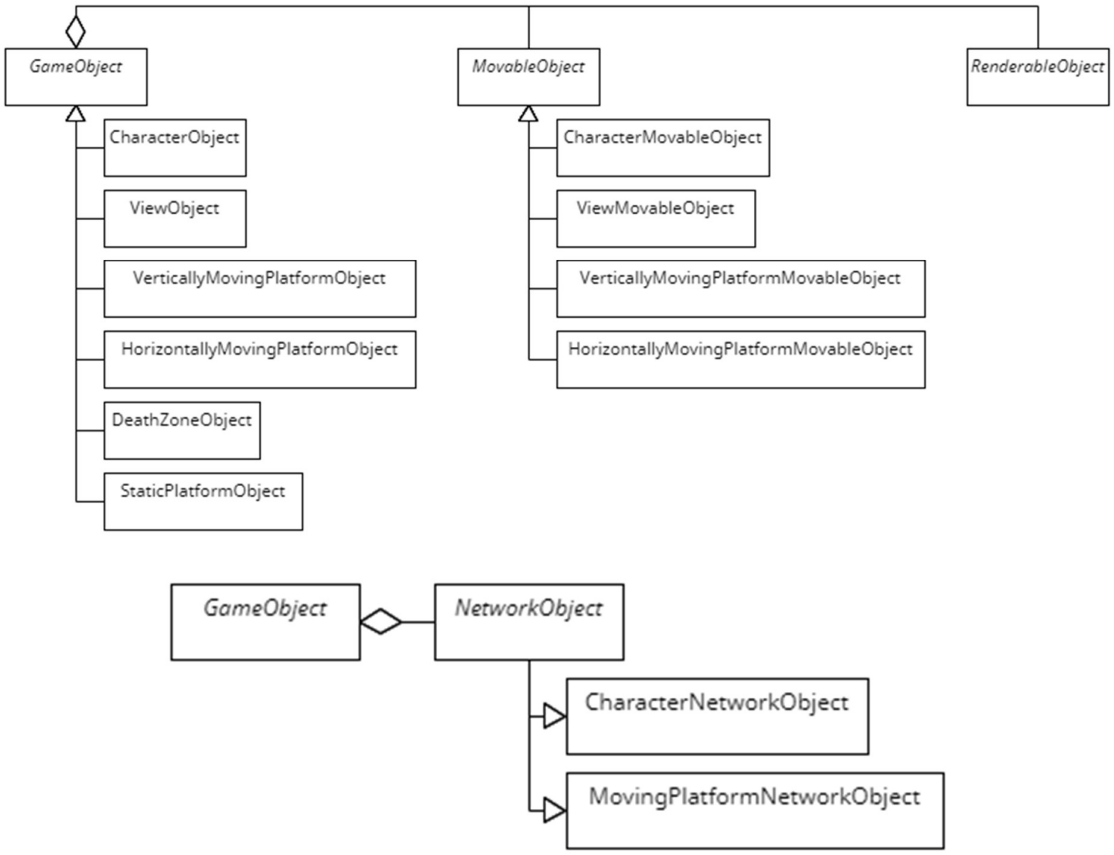
CSC 481

Dr. Robert

06 December 2022

Homework 5 Reflection

I have chosen to use the Rigid Component Model for my game object model. The simple relationships between classes are shown below:



As there are different variants of `MovableObject`, each child of `GameObject` will have an appropriate `MovableObject` or not at all. The `MovableObject` has a pure virtual function named `run()` that the inheriting variant class will define its behavior depending on the type of the `GameObject`. For example, the `CharacterObject` will have `CharacterMovableObject` as one of its fields and `CharacterMovableObject::run()` is defined to update the character position based on the keyboard inputs. Similarly, `VerticallyMovingPlatform` and `VerticallyMovingPlatform` objects will have their own `MovableObject` that has their own defined `run()` function for its intended behavior. While there are several variants of `MovableObject`, `RenderableObject` is a stand-alone class that only has a

`render(sf::RenderWindow *window)` function that behaves exactly the same amongst all `GameObject`s; however, while `ViewObject` and `DeathZoneObject` also have defined `RenderableObject` field, both objects can set the `RenderableObject` as `NULL` since both objects are never rendered. I have defined the `RenderableObject` field just for visual representation reasons for placements.

By extending each `GameObject`, `MovableObject`, `RenderableObject`, and `NetworkObject`, it is highly reusable with high flexibility. Also, as behaviors are represented in classes like `MovableObject`, `RenderableObject`, and `NetworkObject`, it is easily expandable for adding new behaviors to the game engine. However, the current game engine system does not handle the integration of the event management system and the V8 scripting manager system which will need more improvement in the future. While the game engine design itself is flexible enough, the `NetworkObject` may require an update to support flexible network communication. Currently, the `NetworkObject` is focused on supporting server-centric architecture which can be incremented to support other network communication designs as well.

As a project in a whole, it was a nice project that slowly improves throughout each homework assignment with its new features; however, the environmental setup for WSL2 with SFML and Xming has caused some stress as these problems are not directly related to the project but critical. It would also be a good idea to have a basic lecture on how to start a C++ project as I was confused about the `Makefile` and source directory setup. Currently, I have a directory with over 50 files which is the source and the header files of the entire project.