Deliverable #3
SE 3A04: Software Design II – Large System Design

Group #2 T02

1 Introduction

1.1 Purpose

It is the intention of this document to serve as an exhaustive overview of Climatar's architectural design, and thus its implementation of the PAC architecture pattern. This serves to say that all classes, including their respective states, interfaces, and behaviour adjacent to other classes of the system should be well described. Such completeness aims to achieve that an initial implementation of the system comes with as little overhead as possible.

As such, at its foremost, this document is intended to be used as reference by any developer implementing, extending, or maintaining the Climatar codebase. As this document describes Climatars architecture via UML (Unified Modelling Language) diagrams, it should also serve as use to any technically inclined stakeholders involved.

1.2 System Description

Climatar is a world simulation software system illustrating both the long and short term effects that actions have on climate change, greenhouse gas levels, economic stability, and social relations with the governing bodies in a model world. The model world derives most of its themes from the Avatar: The Last Airbender universe. For those unfamiliar with the series, it suffices to know that there are in essence four nations: Fire, Earth, Water, and Air. These nations are not dissimilar to countries on Earth.

The game can either be played in survival or overlord mode. In survival mode, users are given governance of one of the four nations. Overlord mode sees the user take control of all nations. Based on the state of the world and the regions that compose it, news events will be posed to the user that require action. Decisions made by the player will oftentimes result in a direct impact on the world, which will in turn morph around the change with its consequences propagating throughout the simulation. All decisions have consequences associated with them, and the user can only react to events pertaining to their region(s) that are non passive. Passive events are those that occur and consequences are applied without any actions being made by the player. The aspects of events occurring that the user cannot act on remove the omnipotent aspect of control the game would otherwise give, with the player needing to react to a dynamically changing world.

1.3 Overview

This document contains a variety of UML (Unified Modelling Language) diagrams serving to organize and illustrate Climatar's architectural design. By section, these will include state charts for controller classes, sequence diagrams, and finally a detailed class diagram. Details on the intentions of each diagram are widely available but for completeness they will be summarized as follows.

The state charts describe how the state specific to a controller class changes and evolves over time in response to actions or events.

The sequence diagram relates Climatar's use cases to a series of sequential (and sometimes parallelized) steps undergone by the system to respond to said use case.

The detailed class diagram gives light to both the public and private interface of each class in the system. It should describe all methods and variables for each said class.

2 State Charts for Controller Classes

This section should provide a state chart for each controller class for your application.

3 Sequence Diagrams

Each of Climatar's use cases is joined by an accompanying sequence diagram. This section includes said diagrams.

3.1 Use Case A

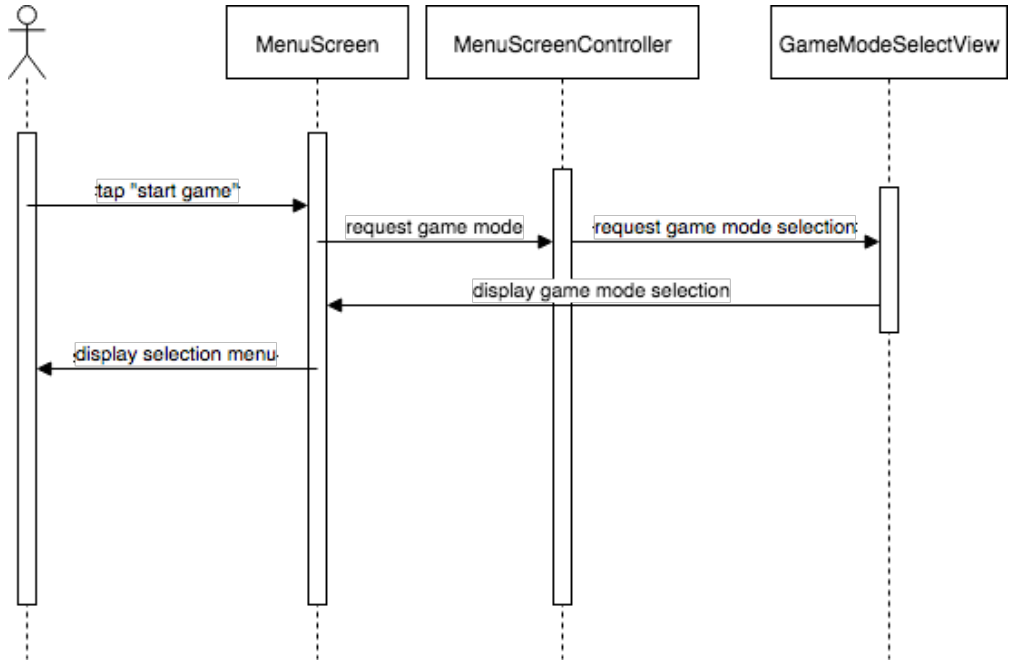

Figure 1: Use Case A

3.2 Use Case B

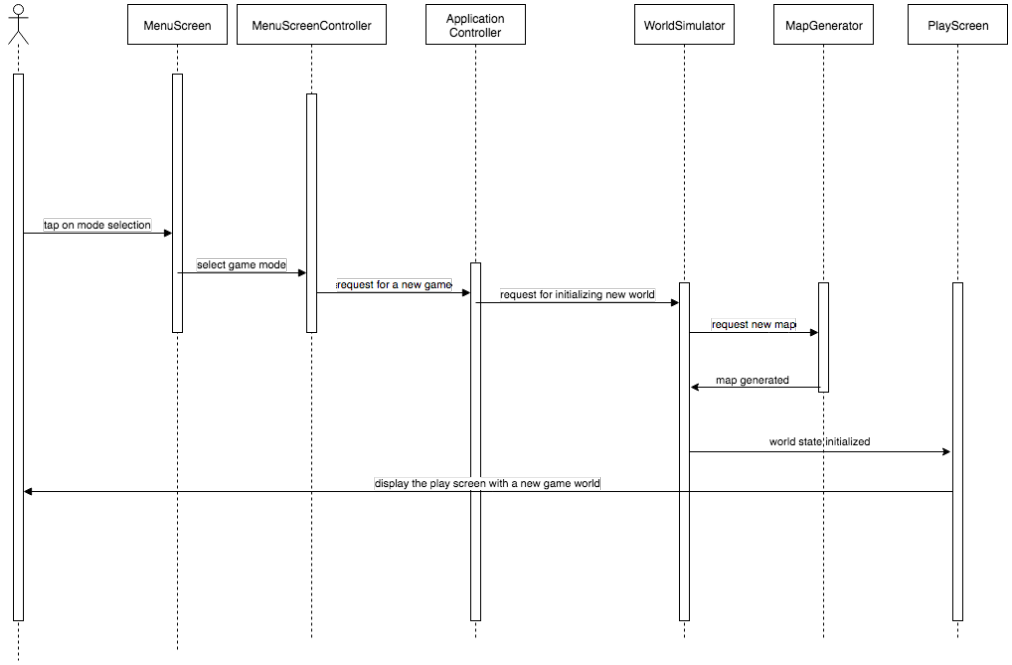

Figure 2: Use Case B

3.3 Use Case C

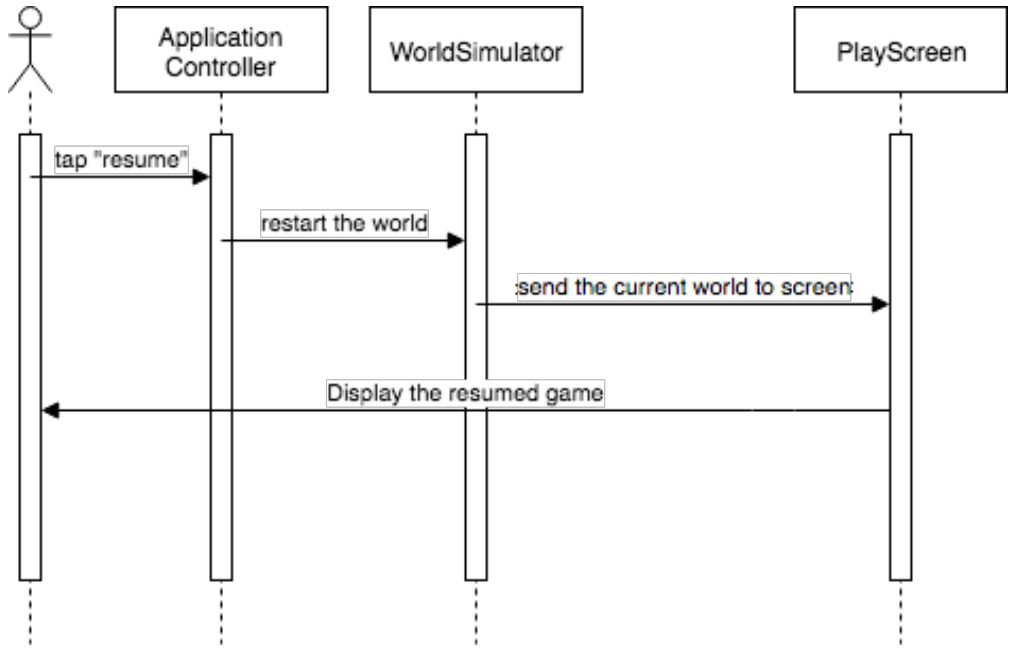

Figure 3: Use Case C

3.4 Use Case D

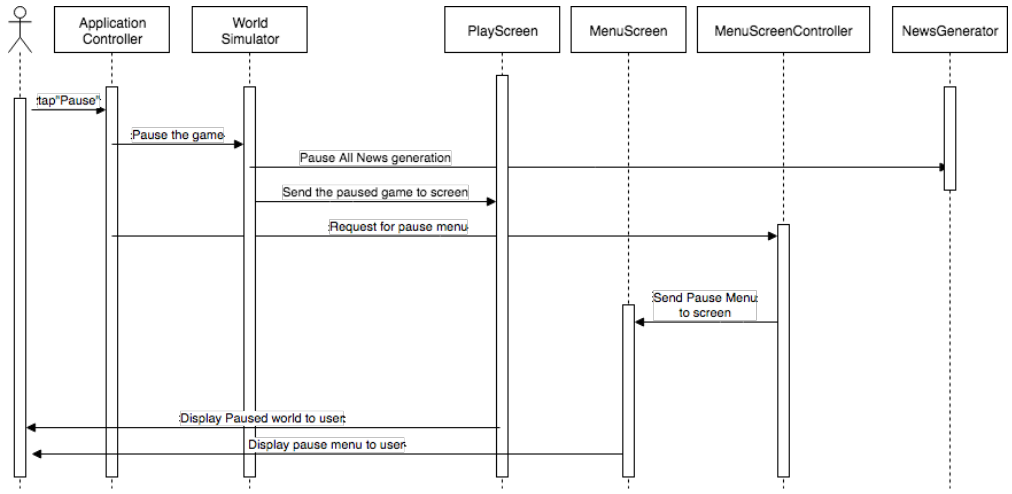

Figure 4: Use Case D

3.5 Use Case E

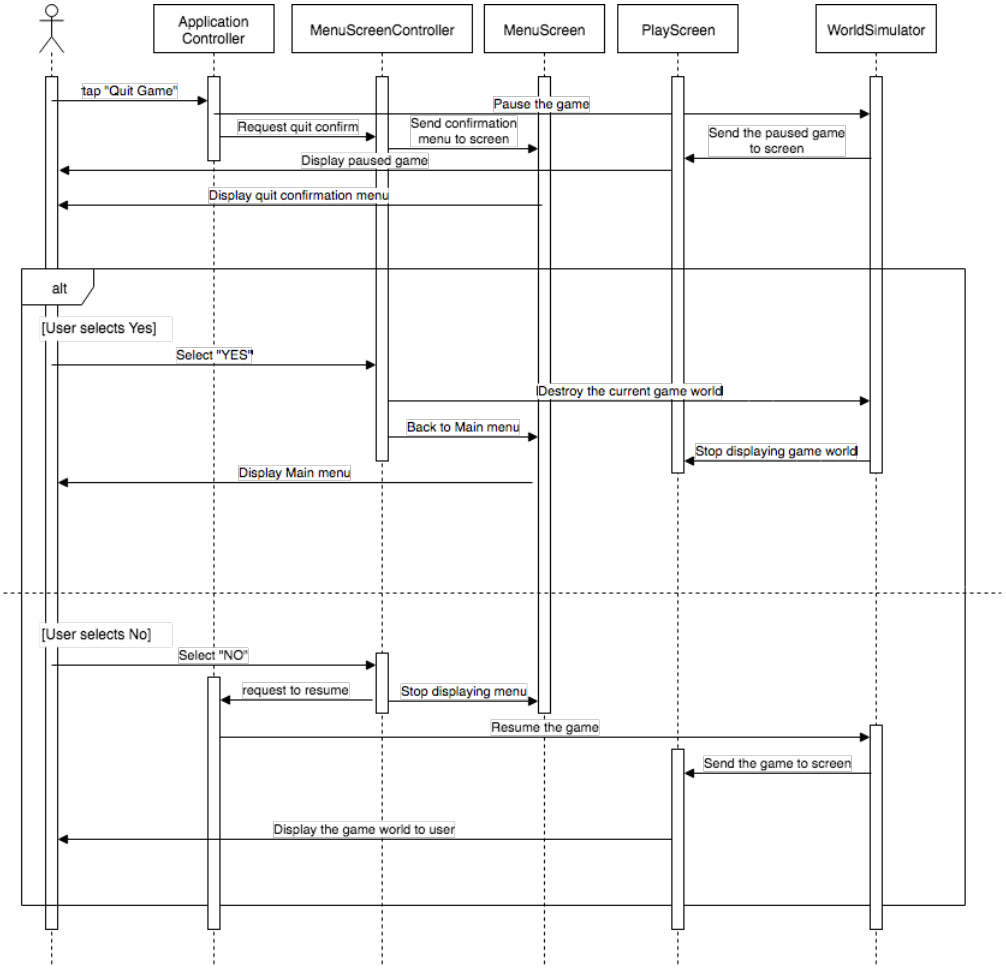

Figure 5: Use Case E

3.6 Use Case F

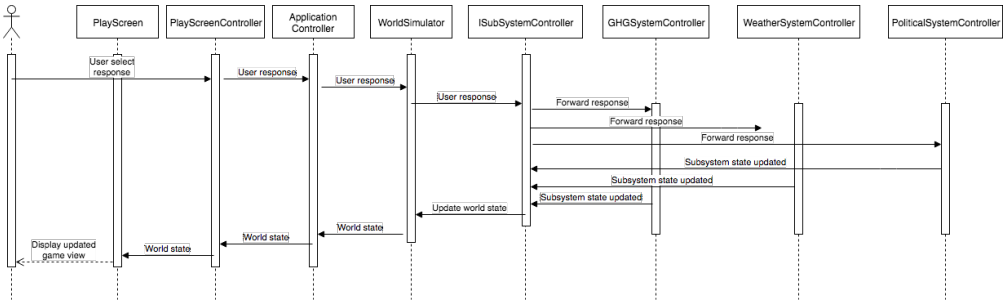

Figure 6: Use Case F

4 Detailed Class Diagram

The following is a representation of the connections of classes which will will compose Climatar.
Note: All edges which have no labels on their endpoints represent 1 to 1 relationships.

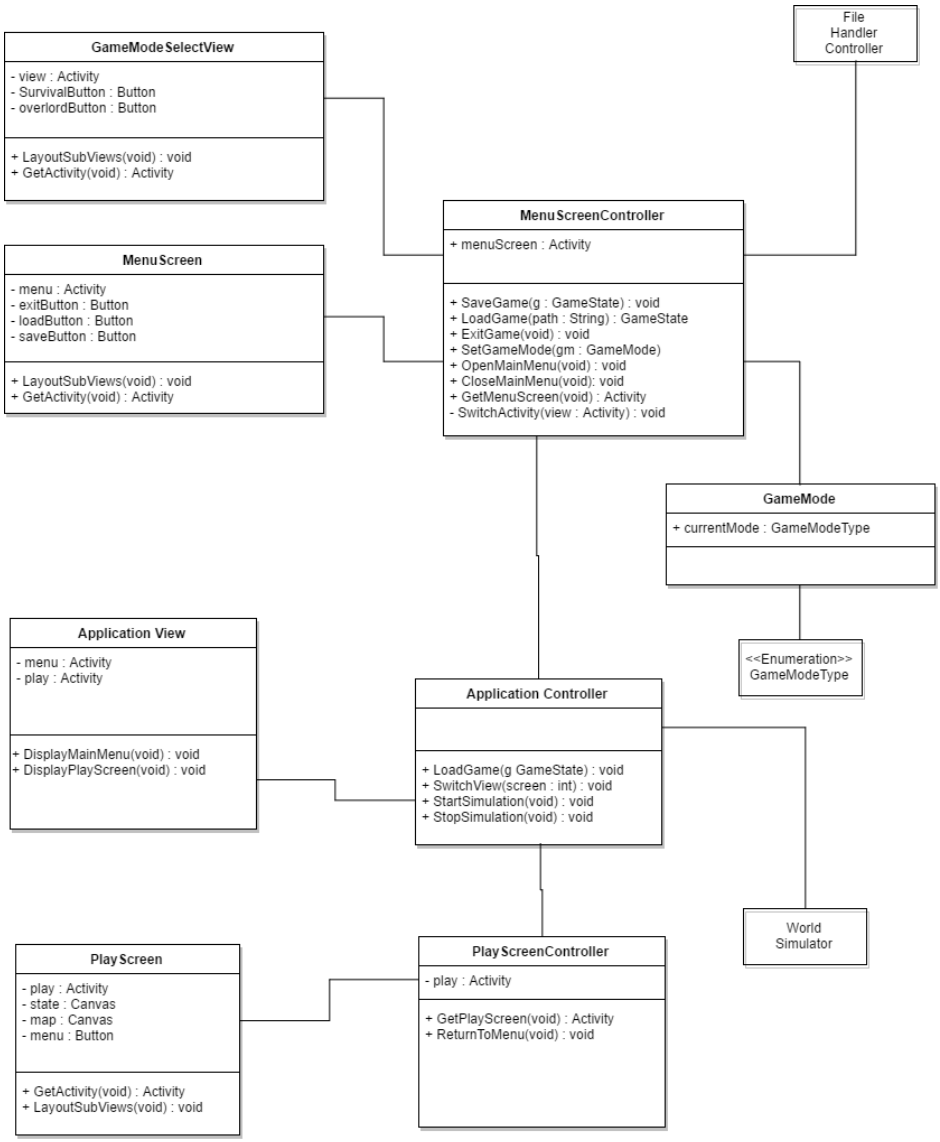

Figure 7: Class Diagram 1/5

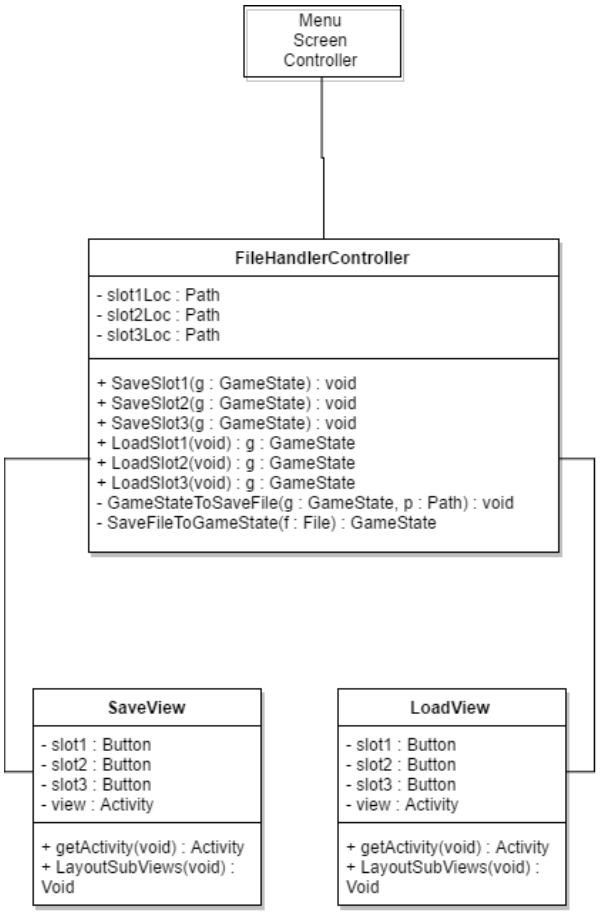

Figure 8: Class Diagram 2/5

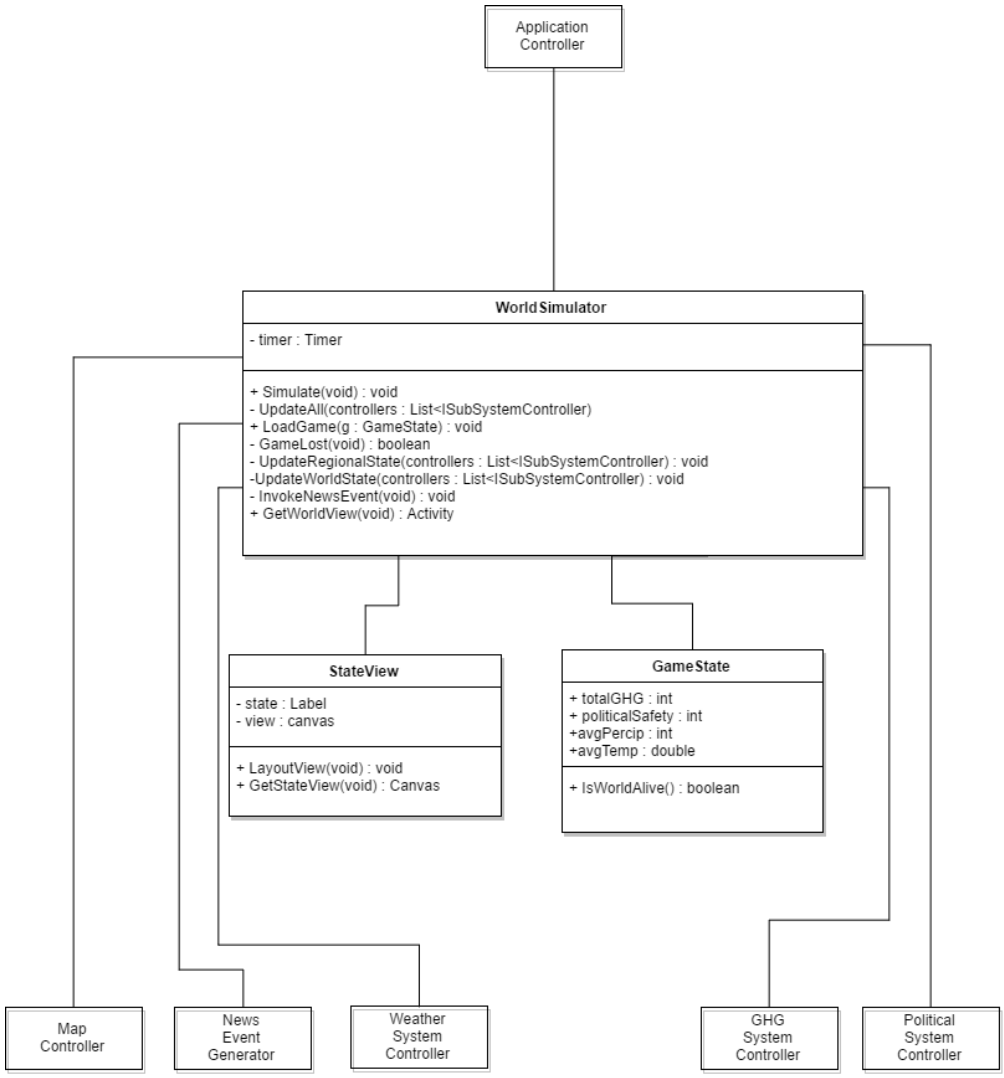

Figure 9: Class Diagram 3/5

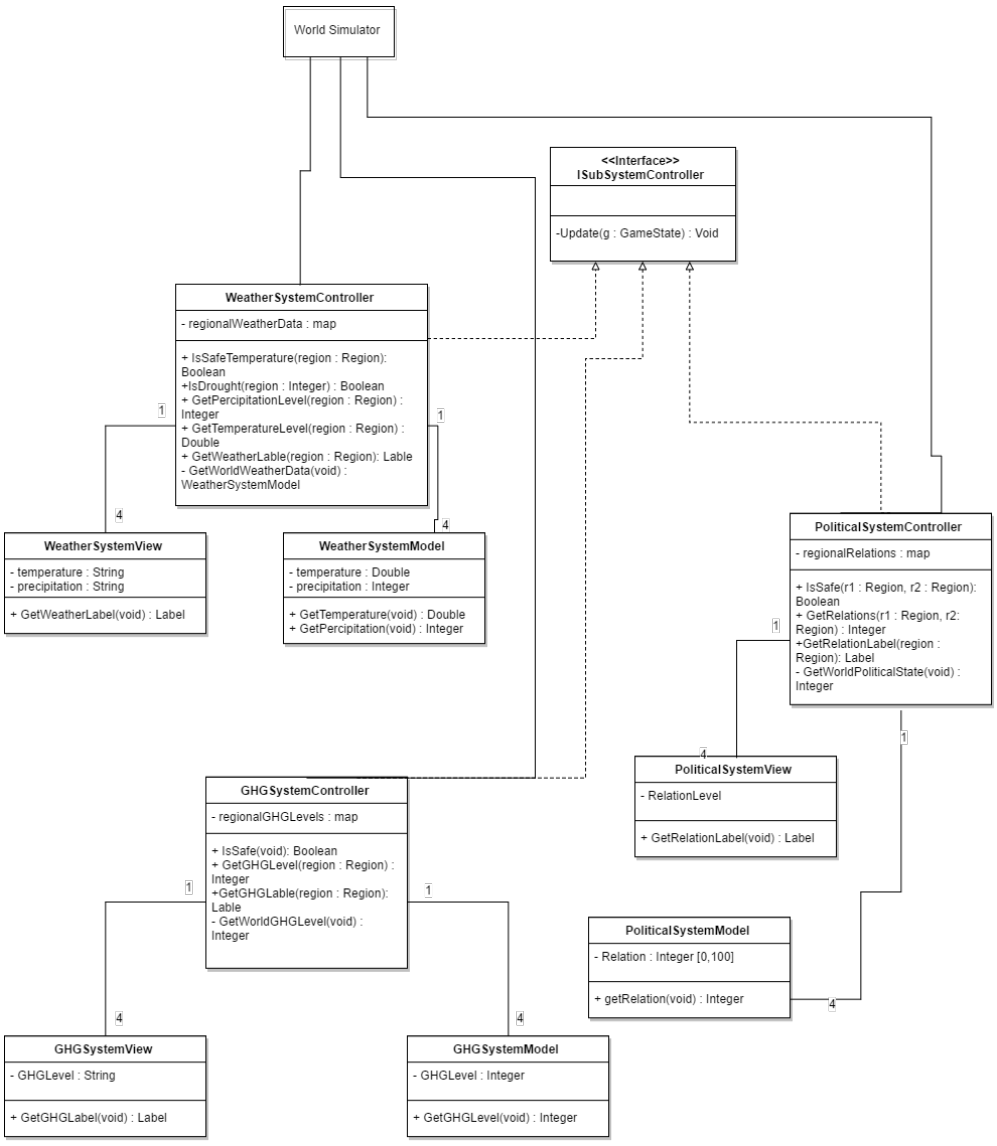

Figure 10: Class Diagram 4/5

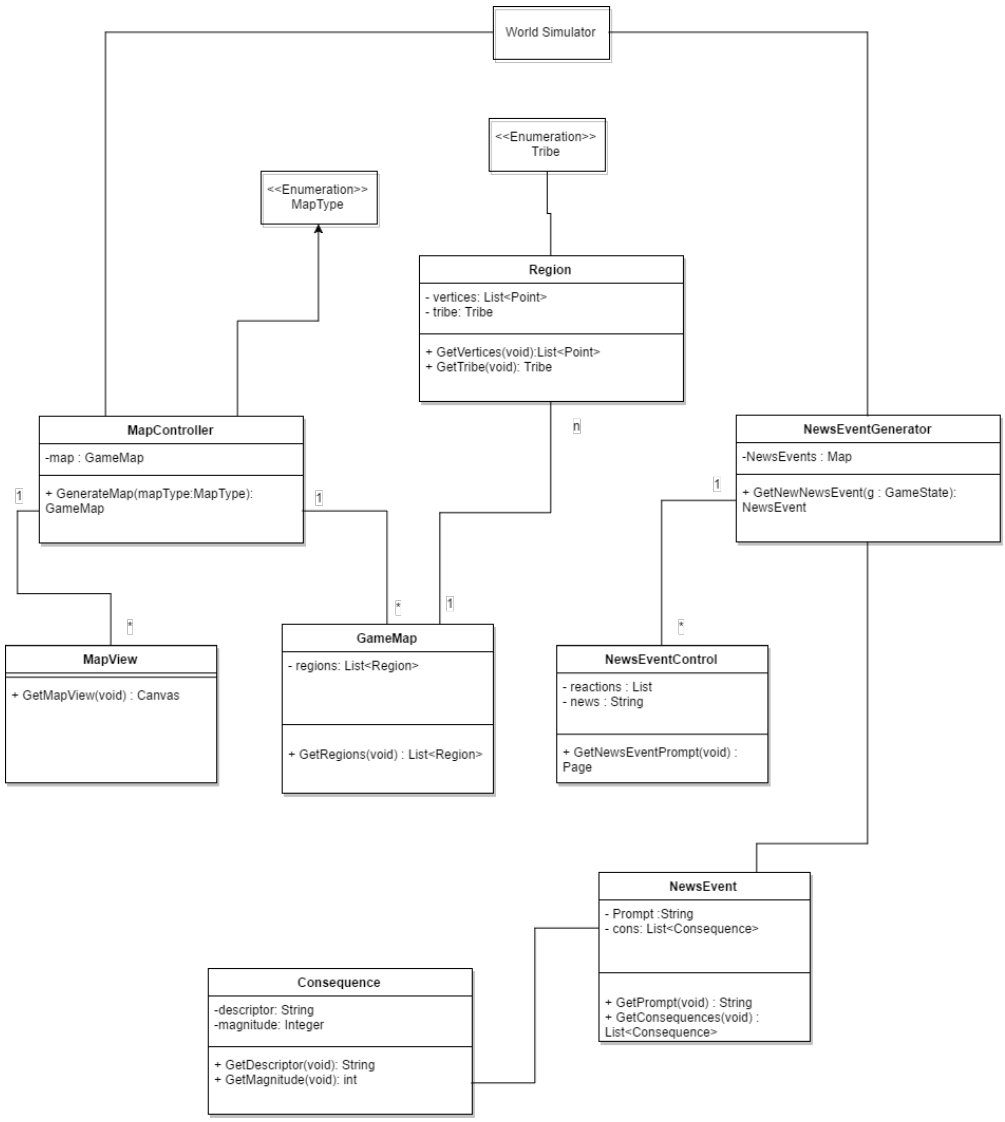

Figure 11: Class Diagram 5/5

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

Contributor Name	**Contributions**
Wenbin Yuan	Sequence Diagrams, 3.
Haris Khan	Review of Section 1.
Riley McGee	Draft of Class Diagram, 4.*
Vishesh Gulatee	State Chart Diagrams, 2.
James Taylor	Section 1. Reviews of Sections 2 and 3.

By signing below you agree to the work divisions stated above correctly representing all contributions made:

IMPORTANT NOTES

- You do NOT need to provide a text explanation of each diagram; the diagram should speak for itself
- Please document any non-standard notations that you may have used
  - *Rule of Thumb*: if you feel there is any doubt surrounding the meaning of your notations, document them
- Some diagrams may be difficult to fit into one page
  - It is OK if the text is small but please ensure that it is readable when printed
  - If you need to break a diagram onto multiple pages, please adopt a system of doing so and throughly explain how it can be reconnected from one page to the next; if you are unsure about this, please ask me
- Please submit the latest version of Deliverable 1 and Deliverable 2 with Deliverable 3
  - They do not have to be a freshly printed versions; the latest marked versions are OK
- If you do NOT have a Division of Labour sheet, your deliverable will NOT be marked