*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 - Senior Capstone Project

Software Engineering Focus

Final Deliverable

Project Title: Building Brain V4

**Team Members:**

Justin Fletcher

Emmanuel Vinas

**Product Owner(s)**:

Leonardo Bobadilla

Ali Mostafavi

**Mentor(s)**:

Peeraya Inyim

Maria Presa

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Leonardo Bobadilla, Ali Mostafavi*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*This document is designed to provide an in-depth look at how Building Brain V.4 creates a simulated environment to test the effectiveness of the Building Brain Android application. You will be presented with an introduction, the set of user stories, the project plan, system design, system validation, glossary, and appendix.*

**Table of Contents**

**Introduction**

Current System

Purpose of New System

**User Stories**

Implemented User Stories

Pending User Stories

**Project Plan**

Hardware and Software Resources

Sprints Plan

*Sprint 1*

**System Design**

Architectural Patterns

System and Subsystem Decomposition

Deployment Diagram

Design Patterns

**System Validation**

**Glossary**

**Appendix**

Appendix A - UML Diagrams

*Static UML Diagrams*

*Dynamic UML Diagrams*

Appendix B - User Interface Design

Appendix C - Sprint Review Reports

Appendix D - Sprint Retrospective Reports

**References**

# **Introduction**

You will be provided with a run down of the current Building Brain Android App and how it influenced the features for the Building Brain Simulation, followed by a description of what the purpose or goal of the Simulation is and how it plans on testing and helping to improve the app’s features.

After this section we will cover the implemented User Stories, the User Stories that we were unable to implement, the Project Plan, the System Design, Validation, and finally the Glossary and Appendix.

## **Current System**

The current system, Building Brain V1 - V3, is an Android Application that is intended to decrease its users energy waste behaviors. It does this through various forms of gamification such as notifications and rewards.

On a country sized scale, 1% of energy waste translates into hundreds of millions of dollars. So to know how effective this application can help reduce these wasteful behaviors, it needs to be properly tested. This however is expensive and time consuming.

## **Purpose of New System**

The new version of Building Brain, which does not directly improve on the application itself, is a simulation, and more specifically, a fast and cheap test environment for the Android app’s features.

The simulation will provide the project owner with a flexible experiment that can simulate an entire day of energy usage behavior in under 5 minutes. The use of the Oculus Rift will provide the user/subject with a more immersive experience that not only feels more real, but blocks out any peripheral vision that could suspend disbelief.

# **User Stories**

The User Stories section will be split up into to two parts, the first being User Stories that were implemented and the second, User Stories that were not unable to be implemented. The User Stories provided references for the development of the project’s many features.

Our User Stories contained 3 different actors. Project Owner, Tester, and User, although Tester and User are exactly the same and the terms will be used interchangeably throughout the User Stories shown. The Project Owner User Stories are behind the scenes, development features, that make designing the simulation easier. The Tester / User User Stories are for anything that goes on while the simulation is actually running, such as moving objects, and turning on energy using objects.

## **Implemented User Stories**

**User Story #725 - Read Light Emission**

Description:

* As a tester I would like to be able to turn lights and items on/off in order to emit light readings during the simulation.

Acceptance Criteria:

1. Items in use emit power readings readable by the app
2. Lights in use emit power readings readable by the app
3. When the light/item is switched off it emits nothing and sends that info to the app as well.

**User Story #727 - Control Character**

Description:

* As a tester I would like to move around the simulated house in order to travel between the different rooms and be able to reach items I can interact with.

Acceptance Criteria:

1. Smooth life like movement
2. No Jumping
3. Needs to be able to turn and move forward.

**User Story #730 - Energy Usage**

Description:

* As a User, I would like to be able to have objects that use energy, so that I can record that energy usage in the future and when not plugged, be able to use battery power, if they have a battery.

Acceptance Criteria:

1. Be able to add Script to an empty game object and have it start consuming energy.
2. be able to display the usage from all energy using objects, based on it’s values.
3. Add a battery to an energy using object.
4. Charge and deplete the battery if the object is not plugged in.
5. Shut off if it is unplugged and/or battery power is all gone.

**User Story #732 - Notification System**

Description:

* As a tester, I would like to receive simulated Notifications in the simulation so that they can be tested before using a notification system directly to the Application on an Android device.

Acceptance Criteria:

1. Simulates the receipt of a Notification within the simulation.
2. Knows the time of the notification, to know how fast it was responded to.
3. Knows what actions were taken if any in response to the notification.
4. Should be able to display any message based on the context of the notification being sent. Such as “Turn off the light in room x”.

**User Story #733 - Plug Load HW Sim**

Description:

* As a tester, I would like to have a simulated hardware device that can be placed in any room in the simulation and provide energy, so that I can connect energy using devices into it to record their usage data.

**Acceptance Criteria:**

1. Can have any number of energy using devices connected.
2. Supplies energy to connected devices.
3. Can display total energy used, possibly with a breakdown of each device’s usage.
4. Can deny energy supply to any connected device.

**User Story #734 - UI For Energy Using Objects**

**Description:**

* As a tester, I would like to have an interface to press buttons linked to various objects in the simulation to turn them on/off so that these interactions are simple and don’t rely on the editor.

**Acceptance Criteria:**

1. For each energy using object in the simulation, have buttons linked to their controlling scripts that turn them on / off.
2. Should only require mouse clicks on a button.

**User Story #749 - HUD**

Description:

* As a user, I would like to have a HUD that displays the time of day, what room I’m in, possibly the weather, and other things related to the simulation, so that I can know things about the simulation in the same way I would know things about the world.

Acceptance Criteria:

1. Hud shouldn’t obstruct the view of the simulation.
2. Display the time.

**User Story #750 - Time Tracking**

Description:

* As a user or tester, I need to know how much time has passed in the simulation so that I can know how much a unit of energy has been used per unit of time.

Acceptance Criteria:

1. Able to return time in seconds, minutes, hours, and days.
2. Time should be returnable in a human readable format i.e. 0 days, 3 hours, 10 min, 27 sec.
3. Be able to set a start time (time of day) for the simulation i.e. 7:00 AM
4. Time should be returnable as time of day in the simulation i.e. 8:45 AM
5. Time scale should be equal to real time. (for now).

**User Story #751 - Time Scale**

Description:

* As a tester, I would like to be able to set the time scale (in relation to the real world) of simulation, in order to speed up or slow down tests in the simulation.

Acceptance Criteria:

1. Should be able to increase time scale.
2. Be able to decrease time scale.
3. Have all other scripts that work over time, adjust to this time scale when it’s changed. such as Energy Usage speeding up and slowing down.
4. Integrate increase/decrease into the HUD.

**User Story #754 - Cycle through Day and Night Time**

Description:

* As a tester, I would like to simulate a day and night transition so that we can adjust the amount of lighting used within the simulated house in order to achieve an optimal and realistic lighting environment during the simulation.

Acceptance Criteria:

1. The length of time for a full day to pass can be adjusted.
2. Based on percentage, the lighting makes a smooth transition between day and night with respective lighting. (moon and sun)
3. Uses Justin Time Trackers

**User Story #779 - UI For Plug Load**

Description:

* As a user I would like to be able to see plug load information when I select one, so that I can see how much energy the room is using, as well as any device in the room attached to the plug load.

Acceptance Criteria:

1. Of the objects attached, user should be able to unplug them, or deny power supply.
2. The user should be able to attach any object to the plug load if it isn’t already.
3. Display the total energy usage of all objects attached to the plug load.
4. Display the individual usage of all objects attached to the plug load.

**User Story #801 - Start Simulation With Main Menu**

Description:

* As a user, I would like to have a main menu before the simulation begins, so that I can adjust specific starting settings to initialize the simulation with.

Acceptance Criteria:

1. Simulation Loads into this scene.
2. Able to choose Simulation length (3 hours, 6 hours, 12 hours, 24 hours, etc.)
3. Able to choose starting time of day (hours, minutes, seconds, am/pm)
4. A start button that loads the simulation with the correct settings.

**User Story #803 - Have a Simulation Manager**

Description:

* As a developer, I would like to have a simulation manager class/object to keep track of any necessary data to be tracked during the time of the simulation, and to have all the data be in one location.

Acceptance Criteria:

1. Simulation Manager as singleton.
2. record total energy usage.
3. List of completed/failed tasks.
4. Start time of simulation.

**User Story #815 - Monitor Room**

Description:

* As a tester I would like to keep track of energy usage by room and objects So I can pinpoint the source of high energy consumption.

Acceptance Criteria:

1. Be able to track what room the user is in
2. Contains a list of all possible energy using equipment that is plugged in or Lights
3. Update Notification system with name of the room, and total energy consumption.

**User Story #826 - Create Tasks**

**Description:**

* As a project owner, I would like to be able to create tasks that are asked of the user to be performed during the simulation, so that I can create a simulation with a specific sequence of tasks that should be performed.

**Acceptance Criteria:**

1. Project owner can create a new task.
2. Can set task start time.
3. Can set a task time limit.
4. Can set task point value.
5. Can set task name and description.
6. Have task types. (turn on, turn off).

**User Story #827 - Begin And End Tasks**

Description:

* As a user I would like to have a task begin at the time it is supposed to start, and end a period of time after that based on its time limit, so that I may perform those tasks.

Acceptance Criteria:

1. Taks begin during their specified start time, their state becomes in progress.
2. Tasks end after their time limit has been reach, after they have started, their state becomes failed.
3. Tasks can also end if they’ve been completed before the time limit, their state becomes completed.

**User Story #832 - Control Character with Console Controller**

Description:

* As a user I would like to be able to control my player with a console controller and Menu used to change the values of my character as I am playing with the oculus in order to make handling the character much easier and allow me to move around the simulated house with full comfort.

Acceptance Criteria:

1. Must be compatible to an Xbox360 controller
2. Must be able to work with oculus as well
3. esc/pause toggles menu.

**User Story #846 - Interact with objects during tasks.**

Description:

* As a user, I would like to have objects to interact with to complete my tasks, so that tasks have substance and something more to do.

Acceptance Criteria:

1. Add object to a task.
2. Be able to interact with object.
3. turn on/off, move.

**User Story #847 - Complete Tasks**

Description:

* As a user, I would like to be able to complete tasks, because that is what is necessary for the simulation to have the correct data when it is complete.

Acceptance Criteria:

1. Task completes.
2. Triggered by moving object to specific location.
3. Have a drop location.
4. Triggered by turning on or off an object.

## **Pending User Stories**

**User Story #780 - Total Building Energy Usage Display**

Description:

* As a user, I would like to see in the HUD the total energy usage of the building, so that I can see how much energy is being used by all objects together.

Acceptance Criteria:

1. display total energy used by all object in one value, in the hud.

**User Story #802 - End Simulation Statistics Display**

Description:

* As a user I would like to see when the simulation ends, statistics on my performance over the simulation, so that I can compare my performance to past performances or other people’s statistics.

Acceptance Criteria:

1. Simulation ends when duration specified in main menu has been reached.
2. Statistics of total usage over the time of the simulation.
3. List of completed tasks.
4. Calculate total points earned.

**User Story #833 - Create Notifications Dynamically**

Description:

* As a tester I would like to have quests created when a rooms emission is too high so It can simulate the applications notification system and allow for objectives to be created based on the tester’s performance and habits in order to further study apps usefulness.

Acceptance Criteria:

1. Based on Emission of Room create a task using Justin’s Task script.
2. Add task to tool bar and Make sure player knows a task has been added to his tool bar

**User Story #728 - Application Integration**

Description:

* As a tester I would like to be able to send my usage of lights and electronic devices to the android application Building Brain, so I can track the usage with the Application and test the functionality of the app.

Acceptance Criteria:

1. Send proper data to the app
2. Make sure app is visible during the simulation

**User Story #729 - Measureable Unit For Unity**

Description:

* As a User, I would like to be able to view my power usage in some measurable amount to be able to gauge my energy waste habits.

Acceptance Criteria:

1. Needs to be able to convert/translate to Brain Building readings.
2. Unity’s Monobehaviours should be able to use it.

# 

# **Project Plan**

The project plan will describe many of the hardware and software resources used to bring the Building Brain v.4 Simulation to life. We used many productivity applications to plan and communicate, and a few main development applications to create the simulation.

Following the hardware and software, all the Use Cases for each sprint will be laid out from most to least important for each sprint respectively, whether they were implemented or not.

## **Hardware and Software Resources**

1. ***Unity3D -*** The Game Engine used to create the simulation.
2. ***Monodevelop -*** The IDE used to write all the source code. Integrates directly with Unity3D.
3. ***C# -*** Unity3D allows the choice of programming languages: Javascript, C#, or Booscript. We chose C#.
4. ***GitHub -*** Used to create file backups, and branches to avoid data loss.
5. ***Mingle -*** Used to plan sprints and tasks for finishing sprints.
6. ***Google Drive -*** Used to store all documents and other files that are not source code or direct assets for the project.

## **Sprints Plan**

### ***Sprint 1***

(01/16/2016 - 01/29/2016)

**User Story #730 - Energy Usage**

Description:

* As a User, I would like to be able to have objects that use energy, so that I can record that energy usage in the future and when not plugged, be able to use battery power, if they have a battery.

Acceptance Criteria:

1. Be able to add Script to an empty game object and have it start consuming energy.
2. be able to display the usage from all energy using objects, based on it’s values.
3. Add a battery to an energy using object.
4. Charge and deplete the battery if the object is not plugged in.
5. Shut off if it is unplugged and/or battery power is all gone.

**User Story #725 - Read Light Emission**

Description:

* As a tester I would like to be able to turn lights and items on/off in order to emit light readings during the simulation.

Acceptance Criteria:

1. Items in use emit power readings readable by the app
2. Lights in use emit power readings readable by the app

When the light/item is switched off it emits nothing and sends that info to the app as well.

**User Story #734 - UI For Energy Using Objects**

**Description:**

* + As a tester, I would like to have an interface to press buttons linked to various objects in the simulation to turn them on/off so that these interactions are simple and don’t rely on the editor.

**Acceptance Criteria:**

1. For each energy using object in the simulation, have buttons linked to their controlling scripts that turn them on / off.
2. Should only require mouse clicks on a button.

### ***Sprint 2***

(01/30/2016 - 02/12/2016)

**User Story #727 - Control Character**

Description:

* As a tester I would like to move around the simulated house in order to travel between the different rooms and be able to reach items I can interact with.

Acceptance Criteria:

1. Smooth life like movement
2. No Jumping
3. Needs to be able to turn and move forward.

**User Story #750 - Time Tracking**

Description:

* As a user or tester, I need to know how much time has passed in the simulation so that I can know how much a unit of energy has been used per unit of time.

Acceptance Criteria:

1. Able to return time in seconds, minutes, hours, and days.
2. Time should be returnable in a human readable format i.e. 0 days, 3 hours, 10 min, 27 sec.
3. Be able to set a start time (time of day) for the simulation i.e. 7:00 AM
4. Time should be returnable as time of day in the simulation i.e. 8:45 AM
5. Time scale should be equal to real time. (for now).

**User Story #751 - Time Scale**

Description:

* As a tester, I would like to be able to set the time scale (in relation to the real world) of simulation, in order to speed up or slow down tests in the simulation.

Acceptance Criteria:

1. Should be able to increase time scale.
2. Be able to decrease time scale.
3. Have all other scripts that work over time, adjust to this time scale when it’s changed. such as Energy Usage speeding up and slowing down.
4. Integrate increase/decrease into the HUD.

**User Story #754 - Cycle through Day and Night Time**

Description:

* As a tester, I would like to simulate a day and night transition so that we can adjust the amount of lighting used within the simulated house in order to achieve an optimal and realistic lighting environment during the simulation.

Acceptance Criteria:

1. The length of time for a full day to pass can be adjusted.
2. Based on percentage, the lighting makes a smooth transition between day and night with respective lighting. (moon and sun)
3. Uses Justin Time Trackers

**User Story #749 - HUD**

Description:

* As a user, I would like to have a HUD that displays the time of day, what room I’m in, possibly the weather, and other things related to the simulation, so that I can know things about the simulation in the same way I would know things about the world.

Acceptance Criteria:

1. Hud shouldn’t obstruct the view of the simulation.
2. Display the time.

### ***Sprint 3***

(02/13/2016 - 02/26/2016)

**User Story #733 - Plug Load HW Sim**

Description:

* As a tester, I would like to have a simulated hardware device that can be placed in any room in the simulation and provide energy, so that I can connect energy using devices into it to record their usage data.

**Acceptance Criteria:**

1. Can have any number of energy using devices connected.
2. Supplies energy to connected devices.
3. Can display total energy used, possibly with a breakdown of each device’s usage.
4. Can deny energy supply to any connected device.

**User Story #732 - Notification System**

Description:

* As a tester, I would like to receive simulated Notifications in the simulation so that they can be tested before using a notification system directly to the Application on an Android device.

Acceptance Criteria:

1. Simulates the receipt of a Notification within the simulation.
2. Knows the time of the notification, to know how fast it was responded to.
3. Knows what actions were taken if any in response to the notification.
4. Should be able to display any message based on the context of the notification being sent. Such as “Turn off the light in room x”.

**User Story #779 - UI For Plug Load**

Description:

* As a user I would like to be able to see plug load information when I select one, so that I can see how much energy the room is using, as well as any device in the room attached to the plug load.

Acceptance Criteria:

1. Of the objects attached, user should be able to unplug them, or deny power supply.
2. The user should be able to attach any object to the plug load if it isn’t already.
3. Display the total energy usage of all objects attached to the plug load.
4. Display the individual usage of all objects attached to the plug load.

### ***Sprint 4***

(02/27/2016 - 03/11/2016)

**User Story #803 - Have a Simulation Manager**

Description:

* As a developer, I would like to have a simulation manager class/object to keep track of any necessary data to be tracked during the time of the simulation, and to have all the data be in one location.

Acceptance Criteria:

1. Simulation Manager as singleton.
2. record total energy usage.
3. List of completed/failed tasks.
4. Start time of simulation.

**User Story #815 - Monitor Room**

Description:

* As a tester I would like to keep track of energy usage by room and objects So I can pinpoint the source of high energy consumption.

Acceptance Criteria:

1. Be able to track what room the user is in
2. Contains a list of all possible energy using equipment that is plugged in or Lights
3. Update Notification system with name of the room, and total energy consumption.

**User Story #801 - Start Simulation With Main Menu**

Description:

* As a user, I would like to have a main menu before the simulation begins, so that I can adjust specific starting settings to initialize the simulation with.

Acceptance Criteria:

1. Simulation Loads into this scene.
2. Able to choose Simulation length (3 hours, 6 hours, 12 hours, 24 hours, etc.)
3. Able to choose starting time of day (hours, minutes, seconds, am/pm)
4. A start button that loads the simulation with the correct settings.

**User Story #802 - End Simulation Statistics Display**

Description:

* As a user I would like to see when the simulation ends, statistics on my performance over the simulation, so that I can compare my performance to past performances or other people’s statistics.

Acceptance Criteria:

1. Simulation ends when duration specified in main menu has been reached.
2. Statistics of total usage over the time of the simulation.
3. List of completed tasks.
4. Calculate total points earned.

### ***Sprint 5***

(03/19/2016 - 04/01/2016)

**User Story #826 - Create Tasks**

**Description:**

* As a project owner, I would like to be able to create tasks that are asked of the user to be performed during the simulation, so that I can create a simulation with a specific sequence of tasks that should be performed.

**Acceptance Criteria:**

1. Project owner can create a new task.
2. Can set task start time.
3. Can set a task time limit.
4. Can set task point value.
5. Can set task name and description.
6. Have task types. (turn on, turn off).

**User Story #827 - Begin And End Tasks**

Description:

* As a user I would like to have a task begin at the time it is supposed to start, and end a period of time after that based on its time limit, so that I may perform those tasks.

Acceptance Criteria:

1. Taks begin during their specified start time, their state becomes in progress.
2. Tasks end after their time limit has been reach, after they have started, their state becomes failed.
3. Tasks can also end if they’ve been completed before the time limit, their state becomes completed.

**User Story #832 - Control Character with Console Controller**

Description:

* As a user I would like to be able to control my player with a console controller and Menu used to change the values of my character as I am playing with the oculus in order to make handling the character much easier and allow me to move around the simulated house with full comfort.

Acceptance Criteria:

1. Must be compatible to an Xbox360 controller
2. Must be able to work with oculus as well
3. esc/pause toggles menu.

**User Story #833 - Create Notifications Dynamically**

Description:

* As a tester I would like to have quests created when a rooms emission is too high so It can simulate the applications notification system and allow for objectives to be created based on the tester’s performance and habits in order to further study apps usefulness.

Acceptance Criteria:

1. Based on Emission of Room create a task using Justin’s Task script.
2. Add task to tool bar and Make sure player knows a task has been added to his tool bar

### ***Sprint 6***

(04/02/2016 - 04/15/2016)

**User Story #847 - Complete Tasks**

Description:

* As a user, I would like to be able to complete tasks, because that is what is necessary for the simulation to have the correct data when it is complete.

Acceptance Criteria:

1. Task completes.
2. Triggered by moving object to specific location.
3. Have a drop location.
4. Triggered by turning on or off an object.

**User Story #846 - Interact with objects during tasks.**

Description:

* As a user, I would like to have objects to interact with to complete my tasks, so that tasks have substance and something more to do.

Acceptance Criteria:

1. Add object to a task.
2. Be able to interact with object.
3. turn on/off, move.

**User Story #848 - Build House Model**

Description:

* As a developer, I need to change the 3d model of the house to better refelct the needs of the simulation.

Acceptance Criteria:

1. 3 rooms
2. Kitchen
3. living room
4. bedroom
5. Props
6. Various interactable objects.

### ***Sprint 7***

(04/16/2016 - 04/29/2016)

Sprint 7 was dedicated to finish any unfinished user stories from Sprint 6 and finalizing any documentation from previous sprints, such as diagrams, tests, and use case descriptions, as well as preparing for the final week.

Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in these sprints.

# **System Design**

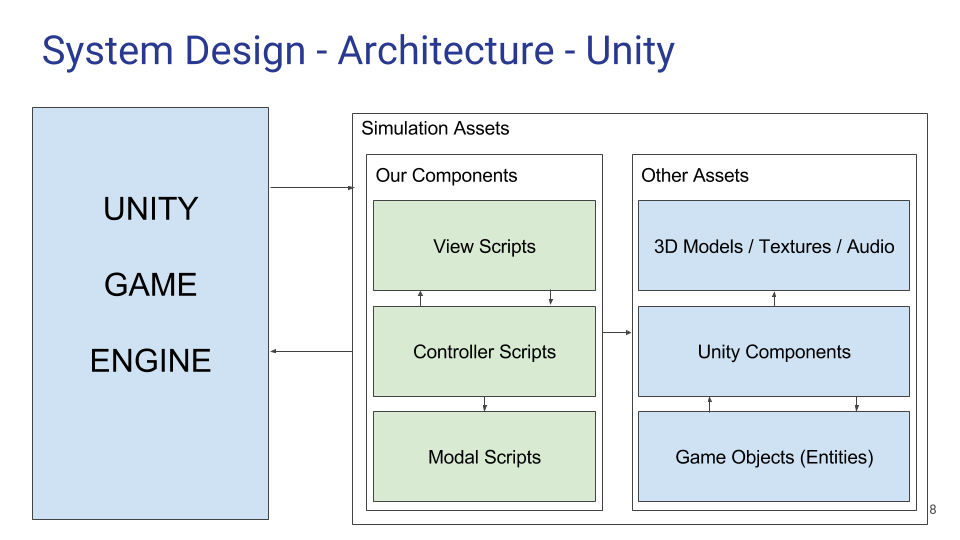
The System Design for Building Brain v.4 consists of Architectural patterns, the System and Subsystem Decomposition, where you’ll see how what we created takes advantage of and builds on top of the Unity Game Engine, and finally the Deployment Diagram which shows how the simulation was planned to be set up on a local machine for users.

## **Architectural Patterns**

## **System and Subsystem Decomposition**

The main subsystems of the simulation are broken up into 2 parts. The first is the underlying Unity Game Engine architecture. The second is a mix of components that we create, that share dependencies with higher level components provided by unity. Our components are further broken up into Model, View, and Controller components.

### Figure 1 - System Decomposition And Architecture



## **Deployment Diagram**

### Figure 1 - Deployment Diagram



## **Design Patterns**

The primary design pattern used in this project is the Object - Component pattern. It is used due to the fact that it is the most efficient way to assign various behaviours to gameobjects in Unity, and Unity is already designed to have objects built this way.

Another big design pattern was MVC, and while it’s not used everywhere, is it used whenever possible for very large scale objects such as the HUD.

There is one instance of the Singleton Pattern, which can be seen in User Story #803, for the SimulationManager class. It was debated whether to use the pattern for other things, but as it wasn’t absolutely necessary the choice was made not to use it in those instances.

# **System Validation**

Provide a one- or two-paragraph overview of this section. You can think of this introductory paragraph as a kind of abstract specific to this section.

### User Story #725 - Read Light Emission

## **Unit Test**

|  |  |
| --- | --- |
| TestID | LightEmission\_001 |
| Objective | While game runs a bogus value should update on the screen every second. |
| Steps | 1. Make sure script is attached to gameobject 2. Create Text Gameobject and attach to light emission script 3. Press Play |
| Expected Results | Every second the text gameobject will update and become larger. |

## **Integration Test**

|  |  |
| --- | --- |
| TestID | ILightEmission\_001 |
| Objective | Set a button and lights that the player can interact with and can turn on/off the lights of a room. |
| Expected Results | When the player walks up to the light switch and puts the input in which will toggle the lights. |
| Results | Success; No Errors or Warnings |

## 

### User Story #727 - Control Character

## **Unit Test**

|  |  |
| --- | --- |
| TestID | CharacterControl\_001 |
| Objective | When input is pushed using keyboard the player does its respective action |
| Steps | 1. Set Speeds 2. Press W/S 3. Press A/D 4. Press Q/E |
| Expected Results | At step 2 the Player should move forward and back respectively. At step 3 the player Rotate Left and right respectively. At step 4 the player should Strife left and right respectively. |

## **Integration Test**

|  |  |
| --- | --- |
| TestID | ICharacterControl\_001 |
| Objective | The character should be able to move around the main project environment with ease while also sending appropriate signals to features |
| Expected Results | The Character Controller should be able to move around freely and smoothly. Should also be able to rotate the camera and move sideways. |
| Results | Success; No Errors or Warnings |

## 

### User Story #730 - Energy Usage

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 730-UT01 |
| **Objective** | Make sure that energy is supplied in some way to be able to add energy usage to the object. |
| **Steps** | 1. Check if the object has energy supplied from being plugged into the plug load. 2. If energy is supplied, add the energy usage. |
| **Expected Results** | Energy usage is added only when the object is plugged into the plug load. |

## 

|  |  |
| --- | --- |
| **TestID** | 730-UT02 |
| **Objective** | Don’t deplete battery charge if battery level is 0. |
| **Steps** | 1. Check the object’s batteries. 2. Loop through all batteries. 3. Once one is found that has charge, deplete energy from that battery and break the loop. |
| **Expected Results** | If no battery is found to have charge, no energy is depleted. |

## 

|  |  |
| --- | --- |
| **TestID** | 730-UT03 |
| **Objective** | Don’t charge battery if all batteries are full. |
| **Steps** | 1. Check the object’s batteries. 2. Loop through all batteries. 3. Once one is found that has less than its total charge, charge that battery and break the loop. |
| **Expected Results** | If no battery is found with less than full charge, do nothing. |

## 

### User Story #732 - Notification System

## **Unit Test**

|  |  |
| --- | --- |
| TestID | Notification\_001 |
| Objective | Notification Pop up is enabled when the model emission has passed a certain limit. It will give detail on the room name, and the amount of emission it is giving off. The space bar will close the popup and add it to the quest panel that can be viewed when pressing ‘i’. |
| Steps | 1. Set Limit, input room names in the model script. 2. Add a random number b/w 1 - 10 to emission every frame. 3. If popup, press Spacebar to close. 4. If disabled press ‘i’ to open quest panel |
| Expected Results | The popup should have faded in with the inputted room names, and their respective emission levels in the info section. SpaceBar should have slowly faded out the panel ‘i’ should have shown a panel at the bottom with the quest info buttons. |

## **Integration Test**

|  |  |
| --- | --- |
| TestID | INotification\_001 |
| Objective | Integrate Feature to main project with all components working at the same time. |
| Expected Results | The Notification System should work with light emission in the room, and not conflict with any other inputs of the game. |
| Results | Success; No Errors or Warnings |

### User Story #733 - Plug Load HW Sim

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 733- UT01 |
| **Objective** | Insert EnergyUsingObject into a list of similar objects, to test the bounds of the list. |
| **Steps** | 1. Have the EnergyUsingObject instance. 2. Create an index variable starting at -1. 3. Call the setEUO() method in the PlugLoadController class, passing the object and the index variable above. 4. Repeat the above steps, each time increasing the index. 5. If successful 10 times, increase index by a count equal to the size of the array and try again. |
| **Expected Results** | The first iteration of Step 3 should catch an error, since inserting at index -1 is illegal. Step 5 should do the same, since there should not be more than (list.Count) elements. |

|  |  |
| --- | --- |
| **TestID** | 733-UT02 |
| **Objective** | Add an EnergyUsingObject to a list of similar objects, and make sure it does not currently exist in the list. |
| **Steps** | 1. Clear the list of EnergyUsingObjects in the PlugLoadController. 2. Have the EnergyUsingObject instance. 3. Choose One of the following:    1. Call the addEuo() method of the PlugLoadControllerClass, passing the object from step 1.    2. Call setEuo() of PlugLoadController passing the object from step 1 and random index from 0 to the size of the list. 4. Repeat step 2 a random number of times (at least once). |
| **Expected Results** | Any repeat of step 2 should catch an error saying that the object already exists in the list, not allowing it to be added. |

## **Integration Test**

|  |  |
| --- | --- |
| **TestID** | 733-IT01 |
| **Objective** | An energy using object can not select a plug load to plug into if it is already plugged into one. |
| **Steps** | 1. Make sure a plug load is in the scene. 2. Select an unplugged energy using object in the scene hierarchy. 3. Press the Connect To Plug Load button in the inspector. 4. Select a plug load from the picker. 5. Press the Connect To Plug Load button. |
| **Expected Results** | Step 5 should be impossible because the Connect To Plug Load Button should not be available if the object is plugged into a plug load already. |

### User Story #749 - HUD

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 749 - UT01 |
| **Objective** | Make sure the selected time format int value corresponds to the TimeFormat enum after a time format is selected from the drop down. |
| **Steps** | 1. The setTimeFormat method is called with a value of 0 - N-1, N being the number of time formats in the enum. 2. The int value is cast as a time format. 3. This is then set as the time format in the HUD modal. |
| **Expected Results** | Since the int values are retrieved from casting the enum to ints, casting the ints back to enums retrieves the correct value. |

## **Integration Test**

|  |  |
| --- | --- |
| **TestID** | 749 - IT01 |
| **Objective** | When no object is selected, the hud does not show any information of any object. |
| **Steps** | 1. Run the simulation. 2. Do not select an object, or select an object, then deselect it. 3. The HUD’s selectedObject is set to NULL by default, or upon deselection the EnergyUsingObject calls the setSelectedObject method of the HUD, passing NULL as the value. 4. In the HUD’s update method, it checks that the selected object is null. 5. If it’s null, set all values to blank strings or 0, and disable any buttons. |
| **Expected Results** | Since the values are set to blank strings and/or 0, no information about an object is show. |

### User Story #750 - Time Tracking

## **Unit Test**

|  |  |
| --- | --- |
| TestID | 750 - UT01 |
| Objective | SetSimulationStartTime interprets the string parameter correctly into an int which should be the number of seconds in the specified time, given the strings format being “00:00:00 AM” |
| Steps | 1. Call the method providing a random time as the string such as “10:06:07 AM” 2. Print the current time to prove that the current time of day in the simulation is the provided start time + the time that the simulation has been running. 3. The repeat steps 1 and 2 providing PM as the last two characters in the string, then stop. |
| Expected Results | The method should correctly interpret the string, converting hours to seconds, then minutes to seconds, then adding all seconds together. It should correctly differentiate AM from PM, adding 12 hours worth of seconds if PM is there instead of AM. |

## 

### User Story #751 - Time Scale

## **Unit Test**

|  |  |
| --- | --- |
| TestID | 751-UT01 |
| Objective | Check if Trying to set a scale larger than the max scale, or smaller than 0 (out of range), is possible. (it should not be). |
| Steps | 1. Print the current time scale. 2. Call the method to set the scale to a random float, greater than the max. 3. Print time scale again, to see it’s value. 4. The repeat steps 1 - 3 but use a random float less than 0, then quit. |
| Expected Results | If a value being set is greater than the max or less than 0, they should set themselves to max and 0 respectively. |

### User Story #754 - Cycle Through Day and Night Time

## **Unit Test**

|  |  |
| --- | --- |
| TestID | DnNCycle\_001 |
| Objective | As player plays the game, the universal lighting should rotate from day to night based on the amount of time the game is played for. |
| Steps | 1. Set speed for day and night using Vector3s 2. Make sure values and gradients are in the desired values. 3. Make sure Stars are on 4. Press play and watch the lighting change |
| Expected Results | The directional light should rotate based on the amount of seconds you stand there. As well as stars rotate with the sky |

## **Integration Test**

|  |  |
| --- | --- |
| TestID | IDaynNight\_001 |
| Objective | The Lighting of the project must be accurate according to the Time Tracker. |
| Expected Results | Based on the simulated time of day, the lighting should represent it accurately. |
| Results | Success; No Errors or Warnings |

### User Story #801- Start Simulation With Main Menu

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 801-UT01 |
| **Objective** | Type a number outside of the range of hours, minutes, or seconds, into the respective text field, and upon pressing start simulation, output a start time that does not exist on a traditional 12 hour digital clock. |
| **Steps** | 1. Type into the hour text field a number less than 1, or greater than 12. 2. Type into the minute text field a number less than 0 or greater than 59. 3. Type into the second text field a number less than 0 or greater than 59. 4. Press the Begin Simulation Button. |
| **Expected Results** | 1. If a number is less than the bottom range in any of the text fields, it is set to the bottom range value. 2. If a number is greater than the top range in any of the text fields, a modulus operation is performed against the range size to create a value within the range. |

## 

|  |  |
| --- | --- |
| **TestID** | 801-UT02 |
| **Objective** | The number provided using the start time text fields and the am/pm select box does not produce the correct number of seconds in a day that corresponds to that time. |
| **Steps** | 1. Type numbers into the hour, minutes, and seconds text fields. 2. Select am or pm from the drop-down. 3. Press begin simulation button. |
| **Expected Results** | 1. Modulus operations are used for the number of seconds remaining in the hours and minutes. 2. Those seconds are added to the seconds entered in the text field. 3. If pm is selected, 12 hours worth of seconds are added. 4. This should produce the proper number of seconds in the time selected. |

## 

|  |  |
| --- | --- |
| **TestID** | 801-UT03 |
| **Objective** | Selecting a Simulation RunTime from the runtime dropdown list will not be converted into the correct number of seconds. |
| **Steps** | 1. Select a time from the runtime dropdown list. 2. Each selection is in exact hours so the number selected need only be multiplied by the number of seconds in the selected number of hours. |
| **Expected Results** | 1. The number of seconds is correct for the chosen number of hours for the simulation runtime. |

## **Integration Test**

|  |  |
| --- | --- |
| **TestID** | 801-IT01 |
| **Objective** | To make sure the simulation start with the correct start time when the simulation loads, after pressing the Begin Simulation button. |
| **Steps** | 1. Add the TimeTracker to the main simulation scene. 2. Run the program. 3. Press the begin simulation button. 4. Wait for the simulation to load. |
| **Expected Results** | 1. When the TimeTracker Awakes it will assign itself to the simulation manager, whereupon the simulation manager will assign back the new starting value. 2. The above step will happen in the Awake method of the TimeTracker so the simulation will definitely not have run for any amount of time. |

### User Story #803 - Have a Simulation Manager

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 803-UT01 |
| **Objective** | Try to get the total simulation run time, when a Time Tracker object is not present. |
| **Steps** | 1. Get the singleton instance of SimulationManager. 2. Get its instance of the time tracker. If it is not null, destroy its gameobject. 3. call the getSimulationRunTime method and print out the value returned. |
| **Expected Results** | The value returned will be 0 if.   1. There was never a time tracker present in the simulation. 2. The time tracker was present but the getSimulationRunTime method was not previously called.   The value returned will be non-zero and incorrect if.   1. The getSimulationRunTime method had been previously called with a time tracker present. |

## 

|  |  |
| --- | --- |
| **TestID** | 803-UT02 |
| **Objective** | Try to get the total energy usage without any plug loads in the scene. |
| **Steps** | 1. Get the list of available plug loads int the scene. 2. Loop through the list and destroy each associated game object. 3. call the getSimulationTotalEnergyUsage method. |
| **Expected Results** | The value returned will be 0 if.   1. There were never any plug loads in the scene to begin with. 2. The getSimulationTotalEnergyUsage was never previously called.   The value returned will be non-zero and incorrect if.   1. The getSimulationTotalEnergyUsage method had been previously called with at least 1 plug load present. |

## 

## **Integration Test**

|  |  |
| --- | --- |
| **TestID** | 803-IT01 |
| **Objective** | Try to create multiple instances of the SimulationManager from outside of itself. Effectively making its role void, since it is trying to hold data in ONE place. |
| **Steps** | 1. Inside of simulation manager create a static int variable called counter. 2. In the constructor of SimulationManager, incremenet the counter by 1. 3. Go to any other class in the project, and inside of a method create a loop that will run 10,000 times. 4. During each iteration of the loop, retrieve the public instance of SimulationManager, since it’s getter method will instantiate it. 5. Also during each iteration of the loop, print the counter from SimulationManager. |
| **Expected Results** | For each of the 10,000 iterations, the counter should always be what it was first initialized to, +1, for the first time the constructor of SimulationManager was called. |

### User Story #815 - Monitor Room

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | Room\_Manager\_1 |
| **Objective** | Test to see if the display is updated and the emission works properly |
| **Steps** | 1. As the player walk into the room which is set up with a box collider trigger 2. Make sure the scripts are populated with correct emission items and the string rmName is not “” |
| **Expected Results** | The room name is displayed in the middle of the screen and |

## **Integration Test**

|  |  |
| --- | --- |
| TestID | IRoomManager\_001 |
| Objective | The character enters the room and shows the correct corresponding name |
| Expected Results | When a player enters the room then the name of the correct room is displayed on the screen. |
| Results | Success; No Errors or Warnings |

## 

### User Story #826 - Create Tasks

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 826 - UT01 |
| **Objective** | Make sure a task is not created outside the timeline. |
| **Steps** | 1. Press the create simulation button. 2. Click outside the timeline, attempting to create a task. 3. The x and y position of the mouse will be checked against the boundaries of the timeline. |
| **Expected Results** | Since the x and y positions of the mouse are outside the boundaries of the timeline, no task will be created. |

|  |  |
| --- | --- |
| **TestID** | 826 - UT02 |
| **Objective** |  |
| **Steps** | 1. Press the create simulation button. 2. Click outside the timeline, attempting to create a task. |
| **Expected Results** | Nothing should happen because the x and y pos of the mouse are checked in the onMouseUp event. |

### User Story #827 - Begin and End Tasks

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 826 - UT01 |
| **Objective** | Make sure a task is not created outside the timeline. |
| **Steps** | 1. Press the create simulation button. 2. Click outside the timeline, attempting to create a task. |
| **Expected Results** | Nothing should happen because the x and y pos of the mouse are checked in the onMouseUp event. |

### User Story #832 - Control Character With Console Controller

## **Unit Test**

|  |  |
| --- | --- |
| TestID | ControllerTest\_001 |
| Objective | Test the Controller inputs for the left analog stick |
| Steps | 1. Plug in remote 2. Set the values > 0 3. Play the simulation 4. Use the Left analog stick alone in every direction. |
| Expected Results | The player controller should have moved left, right, forward and back only. No rotating. |

|  |  |
| --- | --- |
| TestID | ControllerTest\_002 |
| Objective | Test the controller inputs for the right analog stick |
| Steps | 1. Plug in remote 2. Set the values > 0 3. Play the simulation 4. Use the Right analog stick alone in every direction. |
| Expected Results | The player should have rotated to the left and to the right. |

|  |  |
| --- | --- |
| TestID | ControllerTest\_003 |
| Objective | Test the controller and keyboard inputs to display the menu on screen. |
| Steps | 1. Plug in remote 2. Play the simulation 3. Press Start 4. Press Esc on Keyboard. |
| Expected Results | The menu should have appeared when start was pressed and then disappeared when the Esc key was pressed. |

## **Integration Test**

|  |  |
| --- | --- |
| TestID | IControllerTest\_001 |
| Objective | Test Controller with main project environment, make sure it still moves, rotates and does not go through walls. Check for bugs |
| Expected Results | The First Person Controller should be responsive to the console control, should still do the normal actions,such as move and rotate, and should not clip through the wall only bump into them. |
| Results | Success; No Errors, but Camera can see through walls if too close. |

### User Story #846 - Interact with Objects During Tasks

## **Unit Test**

|  |  |
| --- | --- |
| TestID | Interaction\_001 |
| Objective | Player Walks up to Light Switch and toggles the lights on or off. |
| Steps | 1. Set the light objects into the Light Emission Object 2. Approach Gameobject 3. Press the Input button |
| Expected Results | The Lights in the room should turn off with the press of the inputed keystroke. |

|  |  |
| --- | --- |
| TestID | Interaction\_002 |
| Objective | Player Walks up to Cube and Picks it up |
| Steps | 1. Set Distance on Interaction Controller. 2. Fill in needed fields in inspector. 3. Walk up to an object 4. Press the PickUp Key |
| Expected Results | The item you approach and look at will be picked up and will be infront of the players view. |

|  |  |
| --- | --- |
| TestID | Interaction\_003 |
| Objective | Drop object player is carrying |
| Steps | 1. Follow Steps in Interaction\_002 2. Press the PickUp Key |
| Expected Results | The Item being held will be dropped in the place you are standing it. |

## **Integration Test**

|  |  |
| --- | --- |
| TestID | IInteraction\_001 |
| Objective | Attach feature to main character, and add objects to be moved in the main project. |
| Expected Results | The player should be able to walk up to the object and be able to interact with it appropriately. |
| Results | Success; No Errors or Warnings |

### User Story #847 - Complete Tasks

## **Unit Test**

|  |  |
| --- | --- |
| **TestID** | 847-UT01 |
| **Objective** | Make sure when completing a task that the object interacted with is part of a task. |
| **Steps** | 1. Interact with any object. 2. The object calls the checkTaskCompletion method. 3. A for loop runs through all the tasks that have their state set as inProgress. 4. If any task has its task object set to the current object, continue, otherwise quit. |
| **Expected Results** | In step 4, if the object does not find itself set as the task object for any task, nothing should happen. |

## **Integration Test**

|  |  |
| --- | --- |
| **TestID** | 847-IT01 |
| **Objective** | Make sure that the SimObject script is attached to any interactable object by applying the RequireComponent directive. |
| **Steps** | 1. Create a reference to SimObject in InteractionController. 2. When the object is interacted with in the desired way, call the checkTaskCompletion method. |
| **Expected Results** | There is no need to test for the SimObject reference to be null due to RequireComponent. |

# **Glossary**

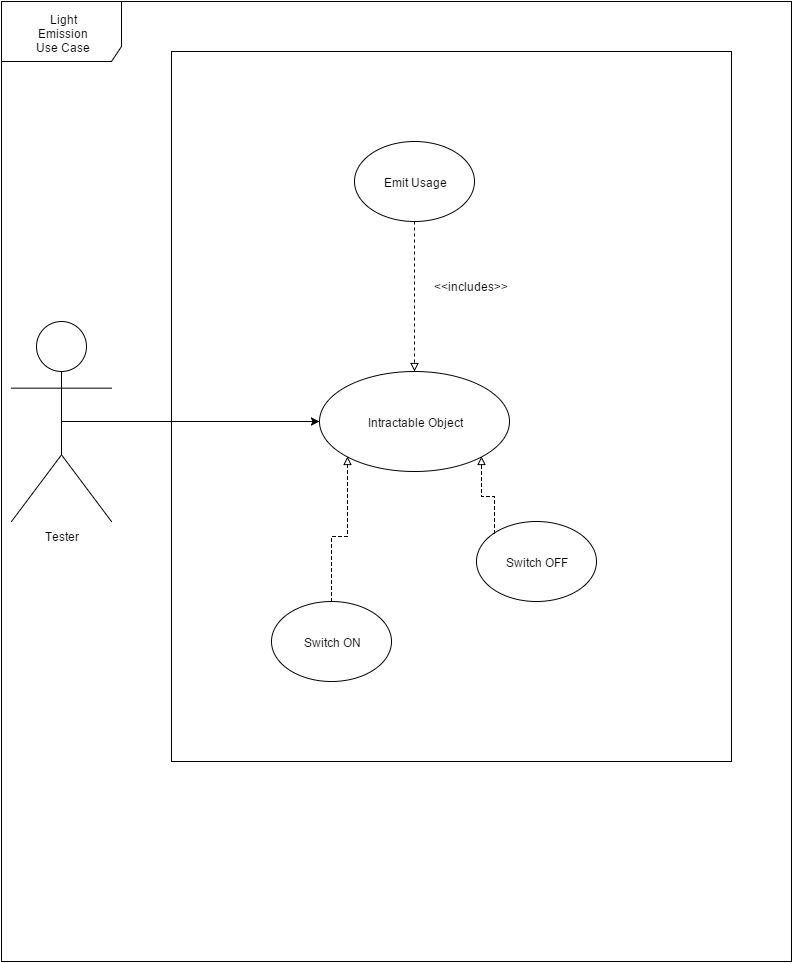
1. **Plug Load** - The device that measures the energy usage of every energy using object plugged into it.
2. **Energy Waste** - An instance of energy usage that is unnecessary for whatever reason.
3. **Unity3D** - The game engine and development environment used for the simulation.
4. **Inspector** - The window within Unity where you can view all the exposed properties of your game objects.
5. **Component** - Any script that extends Monobehaviour and is attached to a game object. These can be added and removed from the inspector.
6. **Script** - A source code file in Unity.
7. **Game Object** - Any object in a scene that has at least one component.
8. **Scene** - A workspace in Unity that contains various game objects. Scenes have the ability to transition between one another.
9. **Primitive Shape** - Any basic shape provided by Unity that can be added to a scene, such as a Cube or a Sphere.
10. **Model** - A 3d model that can be added to the scene to create a game object that resembles the model.
11. **Texture** - Artistic detail file that can be added to a 3d model.
12. **Android** - The operating system of many non-Apple mobile devices.
13. **App** - An application on an Android device.
14. **Monodevelop** - The IDE used to write scripts for Unity3D.
15. **Oculus Rift** - Hardware worn over the User’s head to allow them to see the 3D environment as if they were standing in it.
16. **Game Controller** - A control pad inspired by or directly from a video game console.
17. **HUD** - A display that overlays a simulated environment. Usually to show on screen statistics and information that are not directly part of the world.
18. **Character Controller** - What moves the main User around the environment.
19. **Player** - The main User of the simulation.
20. **User** - An actor on the system whose roles take place only when the simulation is running.
21. **Tester** - Used Interchangeably with the term ‘User’.
22. **Project Owner** - An actor on the system whose roles are primarily behind the scenes in such areas as design and development.

# **Appendix**

## **Appendix A - UML Diagrams**

### ***Static UML Diagrams***

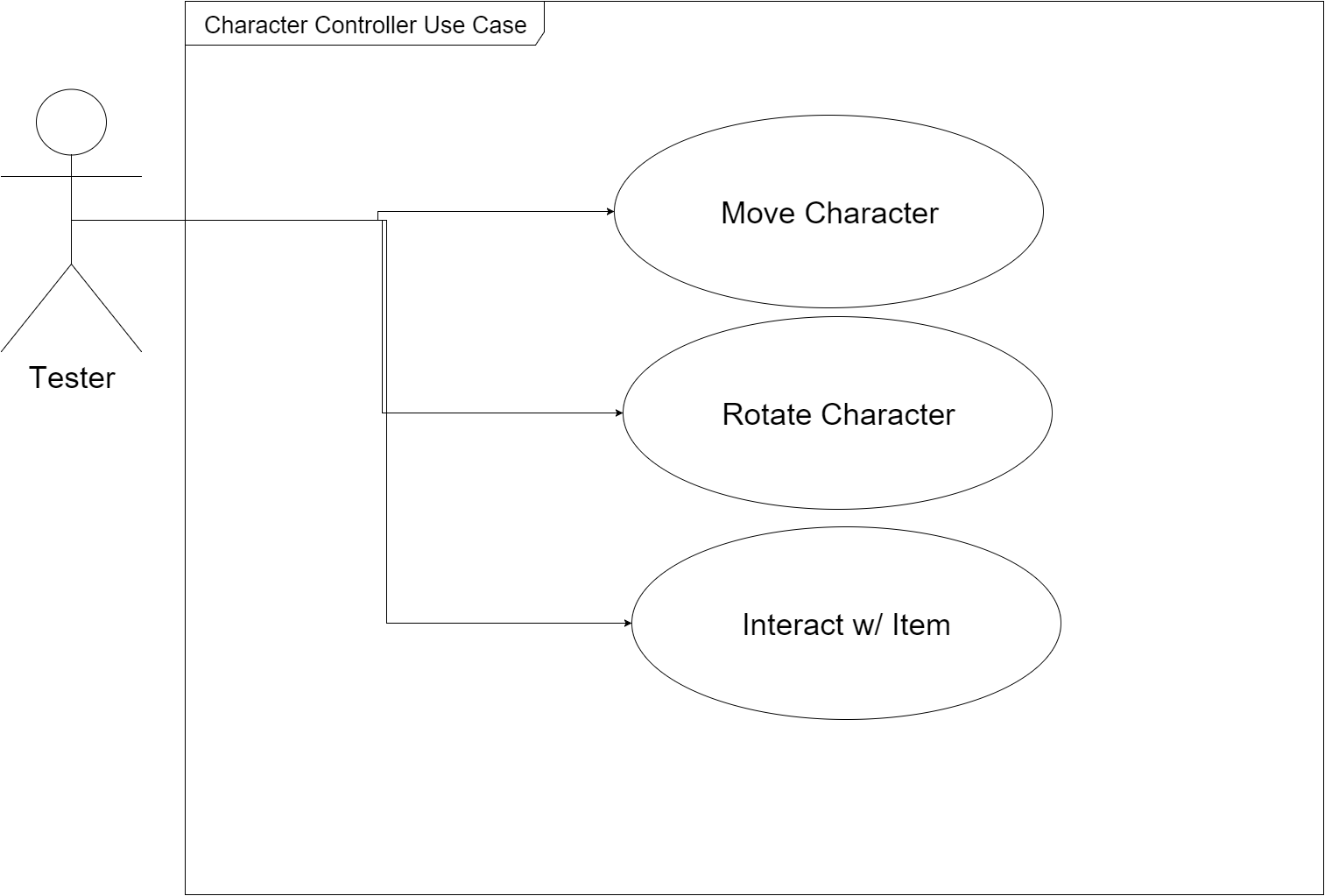
### Figure 1 - 725 Use Case Diagram



### Figure 2 - 725 Class Diagram



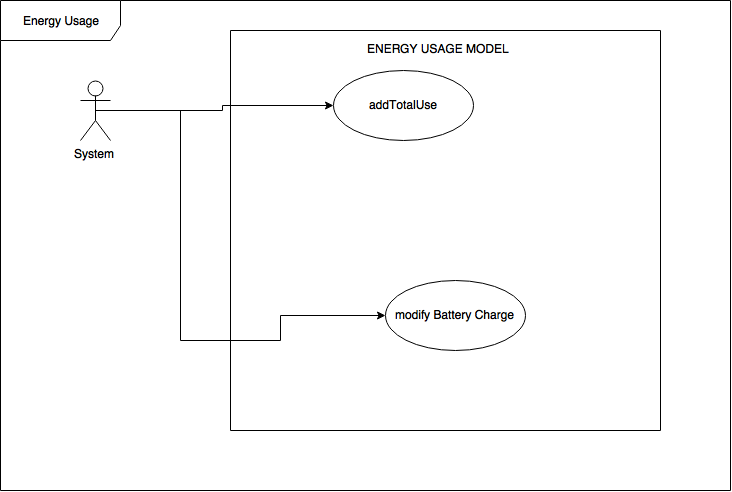
### Figure 3 - 727 Use Case Diagram



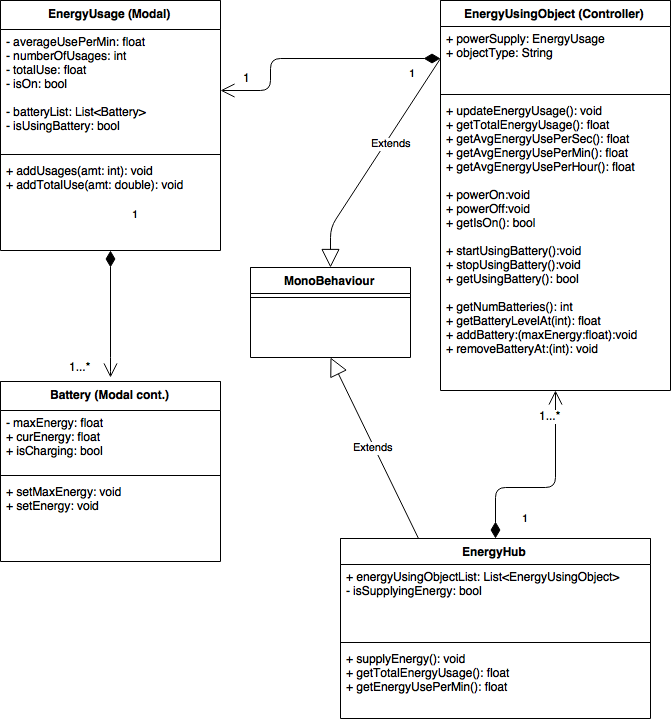
### Figure 4 - 727 - Class Diagram



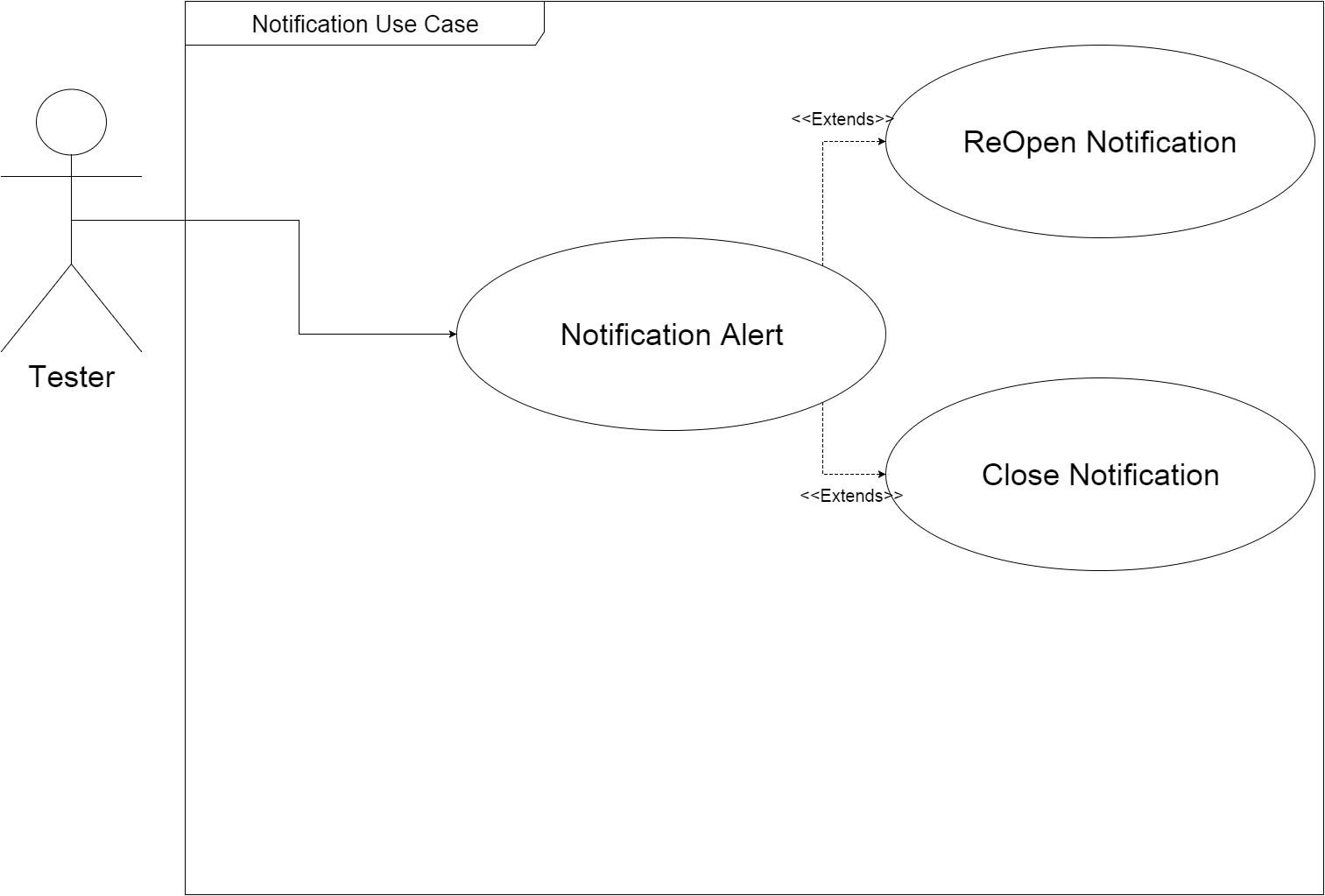
### Figure 5 - 730 Use Case Diagram



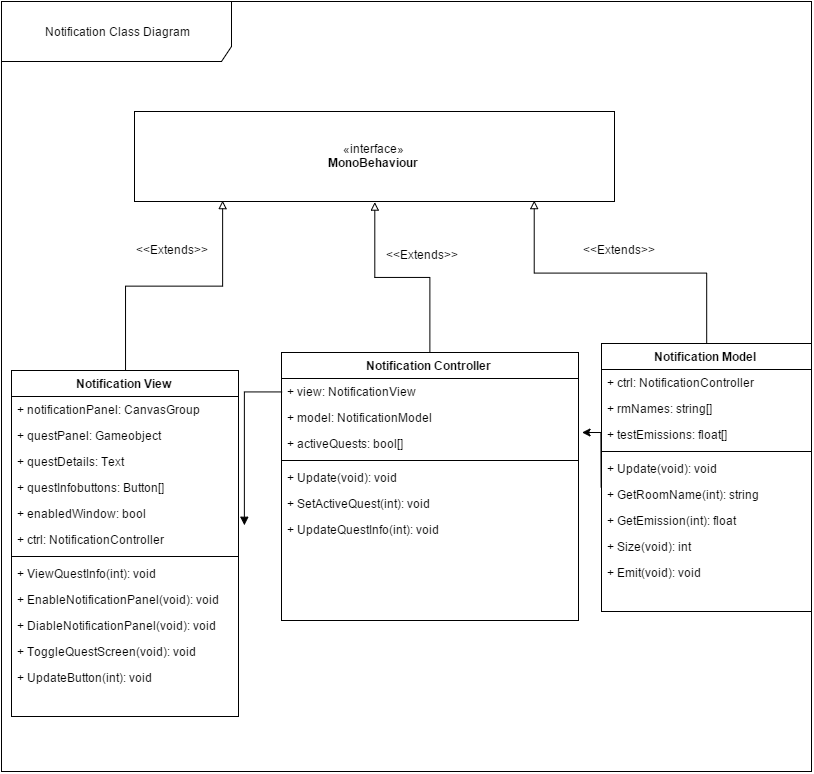
### Figure 6 - 730 Class Diagram



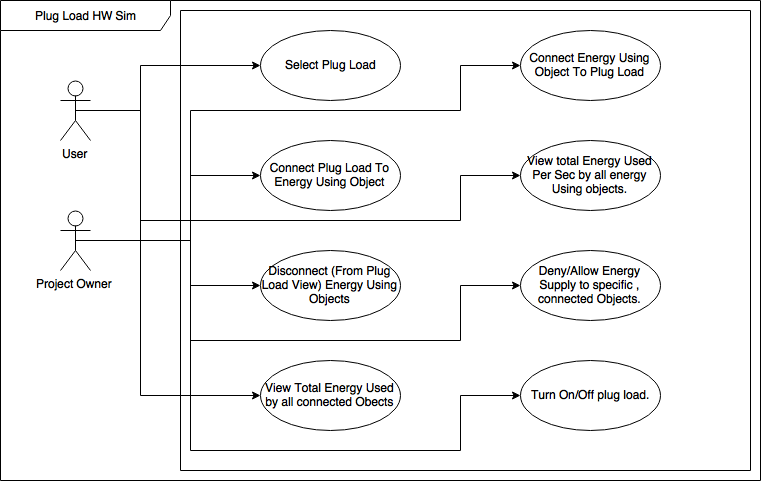
### Figure 7 - 732 Use Case Diagram



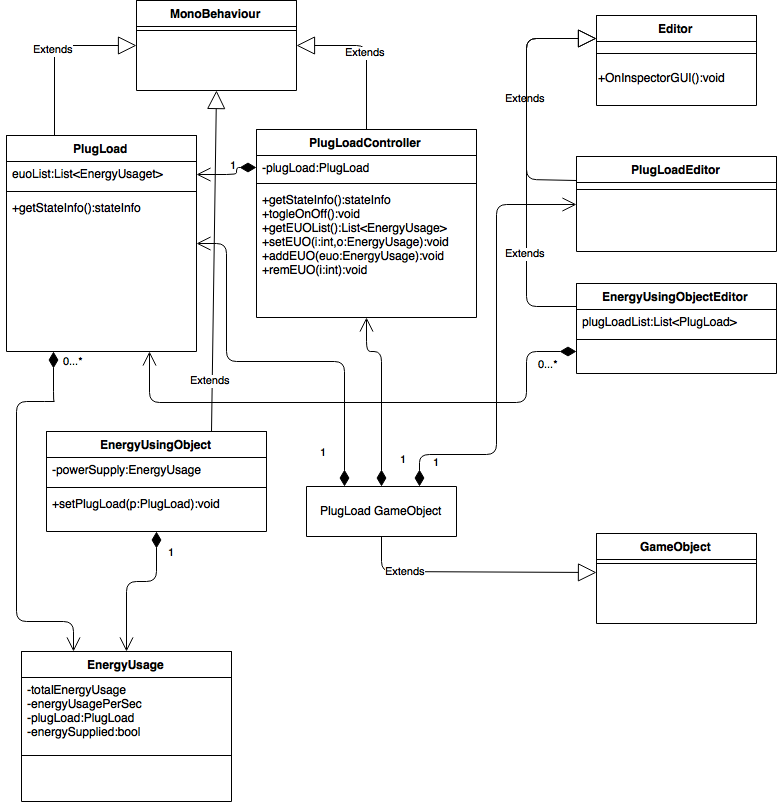
### Figure 8 - 732 Class Diagram



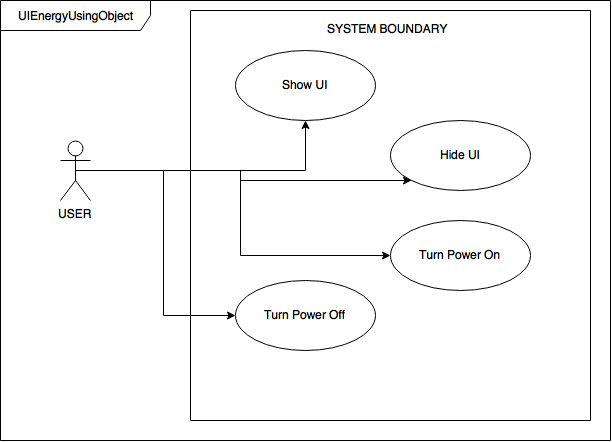
### Figure 9 - 733 Use Case Diagram



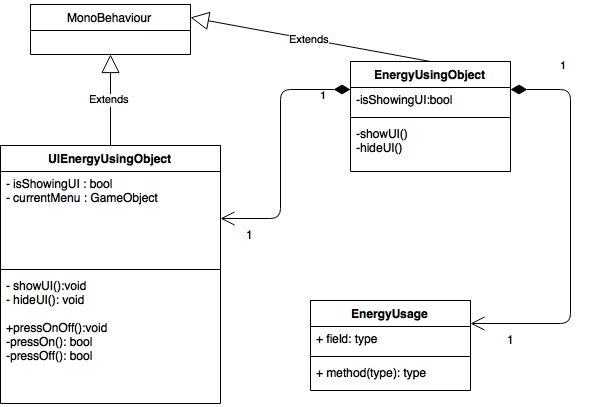
### Figure 10 - 733 Class Diagram



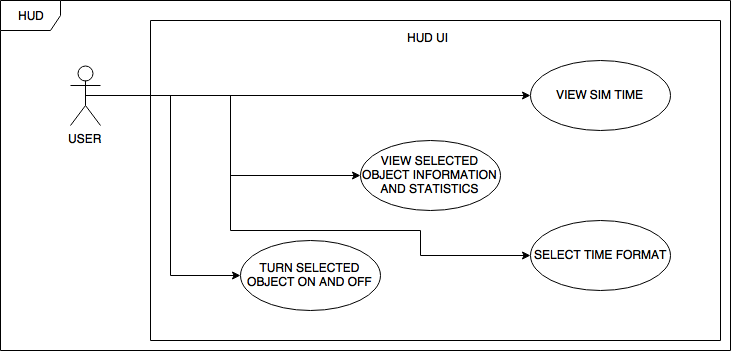
### Figure 11 - 734 Use Case Diagram



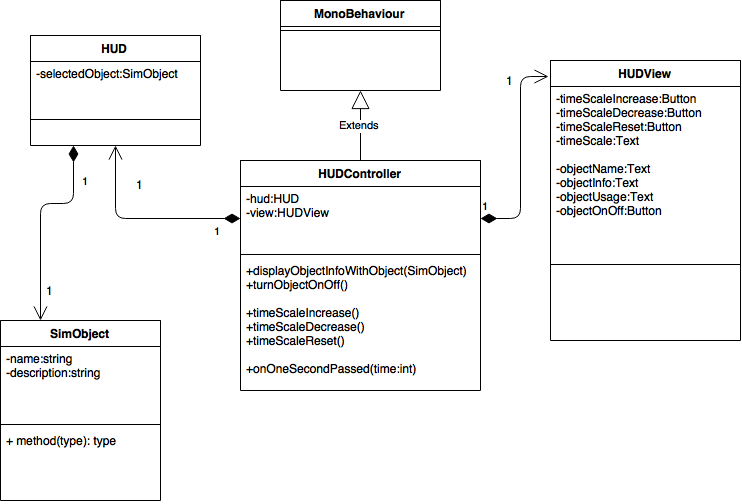
### Figure 12 - 734 Class Diagram



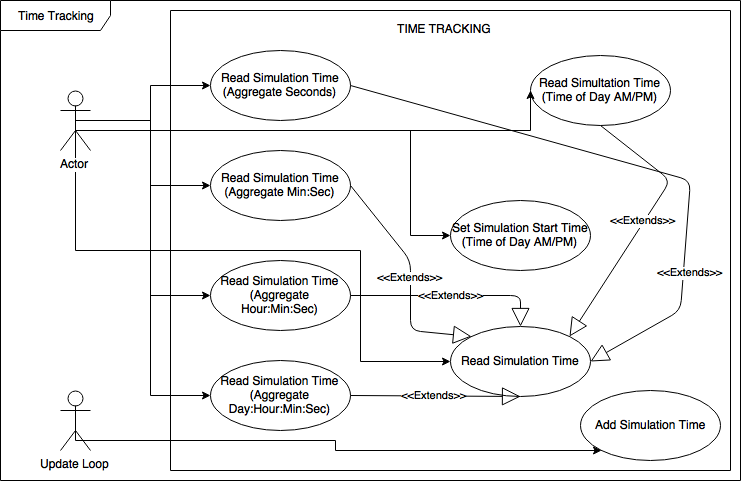
### Figure 13 - 749 Use Case Diagram



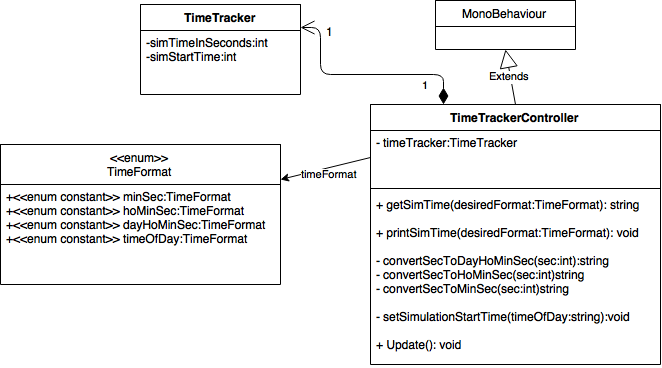
### Figure 14 - 749 Class Diagram



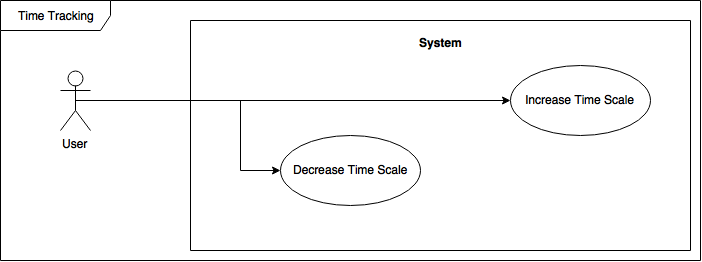
### Figure 15 - 750 Use Case Diagram



### Figure 16 - 750 Class Diagram



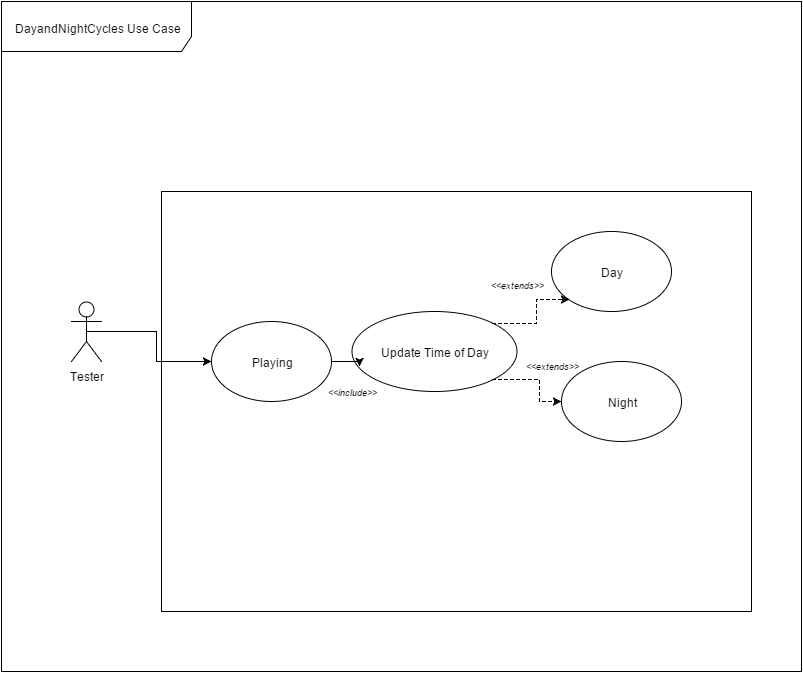
### Figure 17 - 751 Use Case Diagram



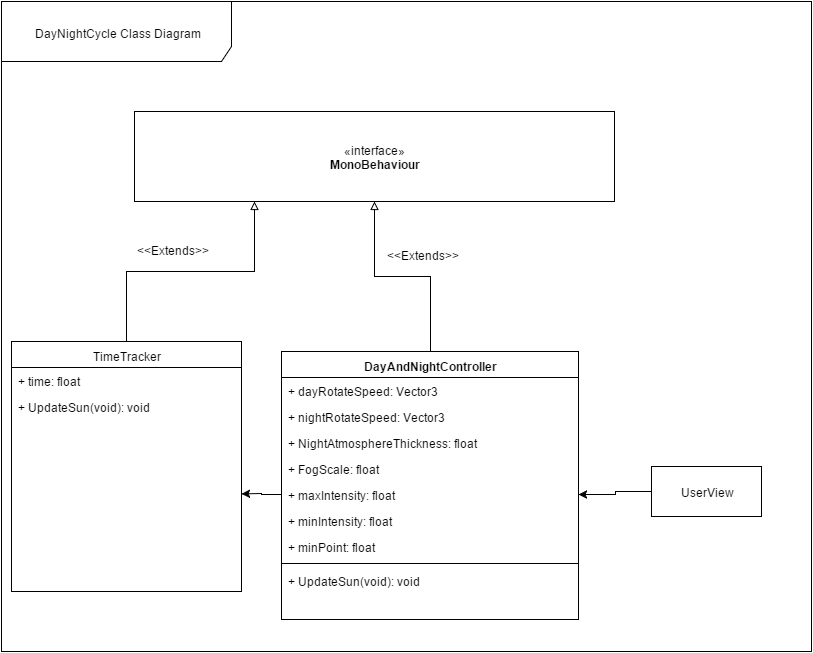
### Figure 18 - 751 Class Diagram

## 

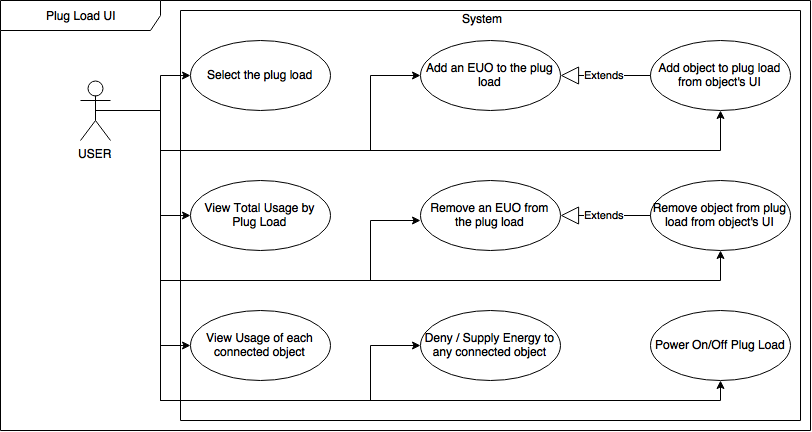
### Figure 19 - 754 Use Case Diagram



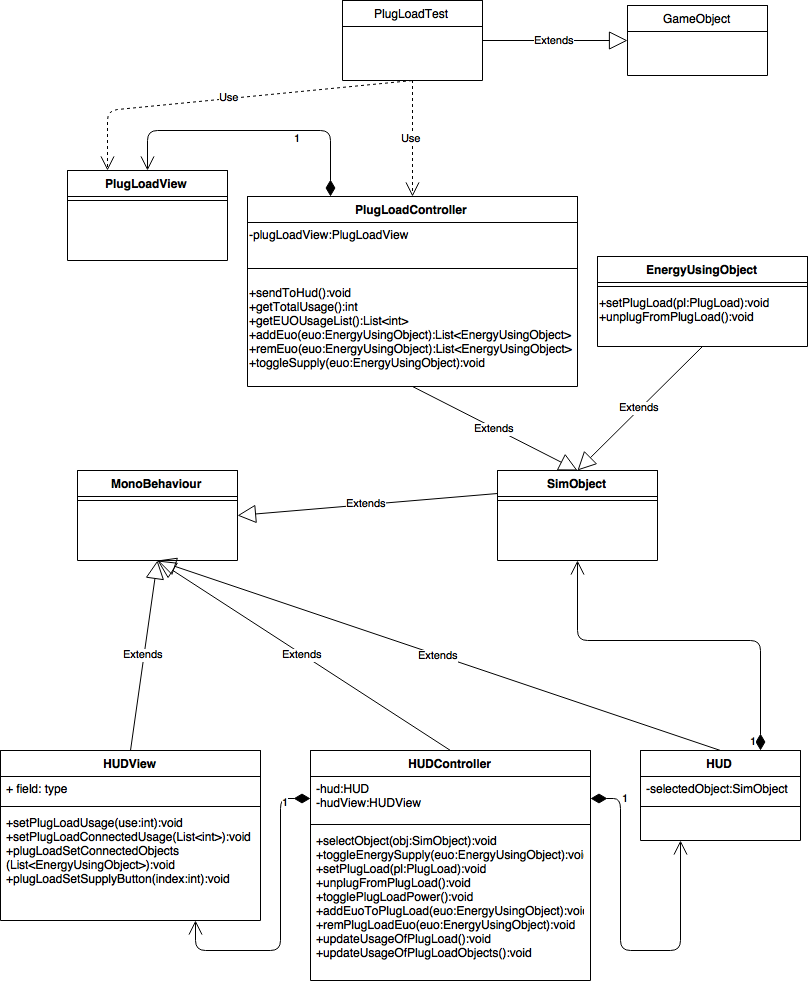
### Figure 20 - 754 Sequence Diagram



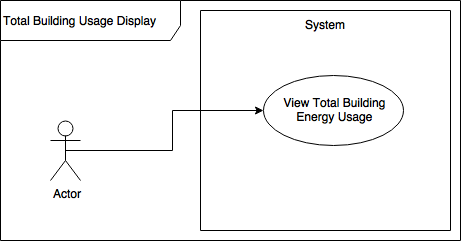
### Figure 21 - 779 Use Case Diagram



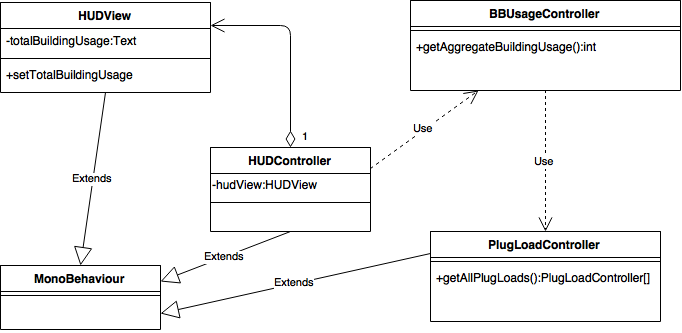
### Figure 22 - 779 Class Diagram



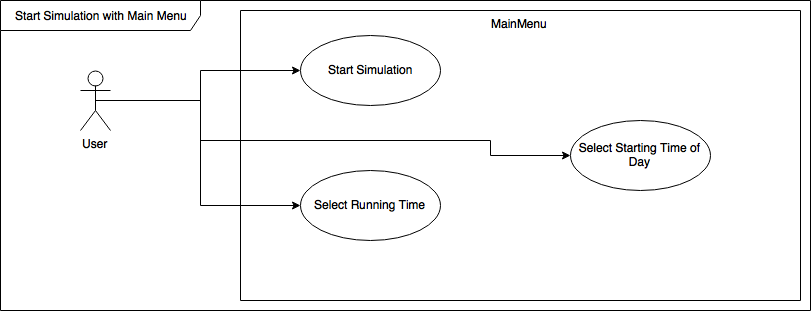
### Figure 23 - 780 Use Case Diagram



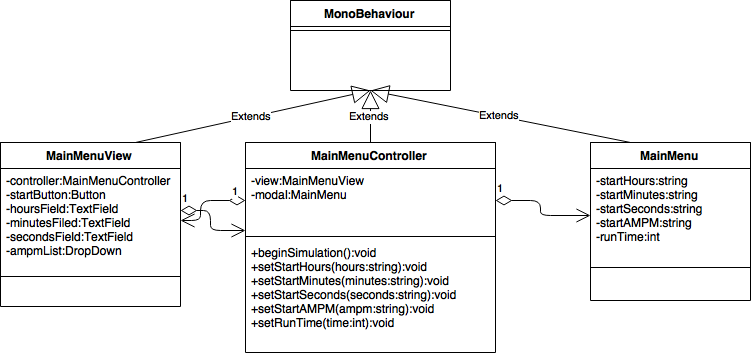
### Figure 24 - 780 Class Diagram



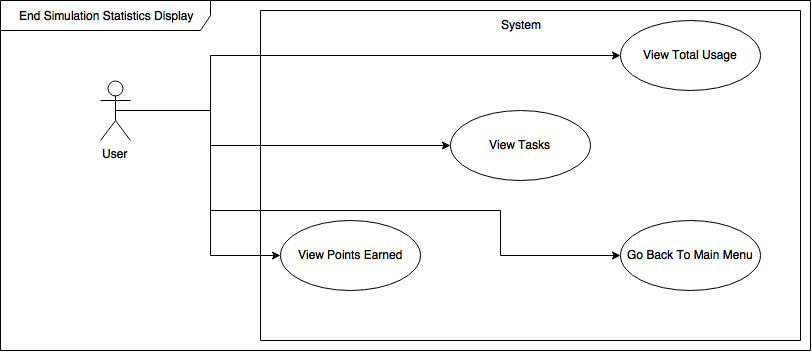
### Figure 25 - 801 Use Case Diagram



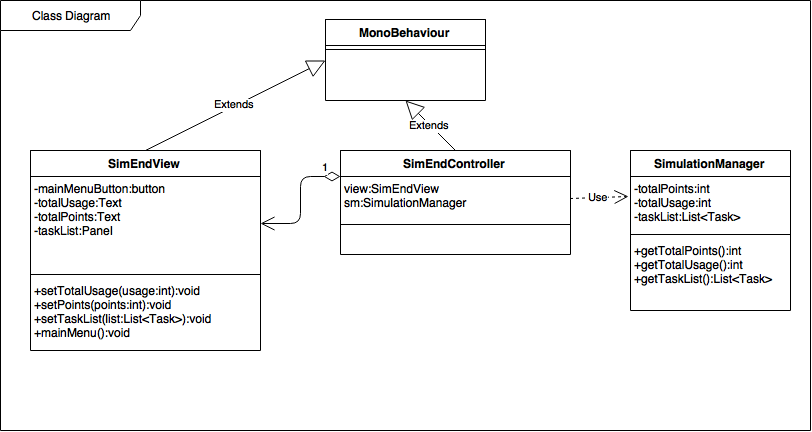
### Figure 26 - 801 Class Diagram



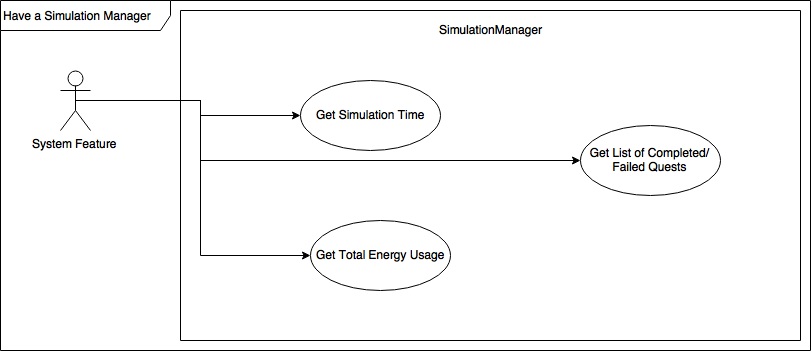
### Figure 27 - 802 Use Case Diagram



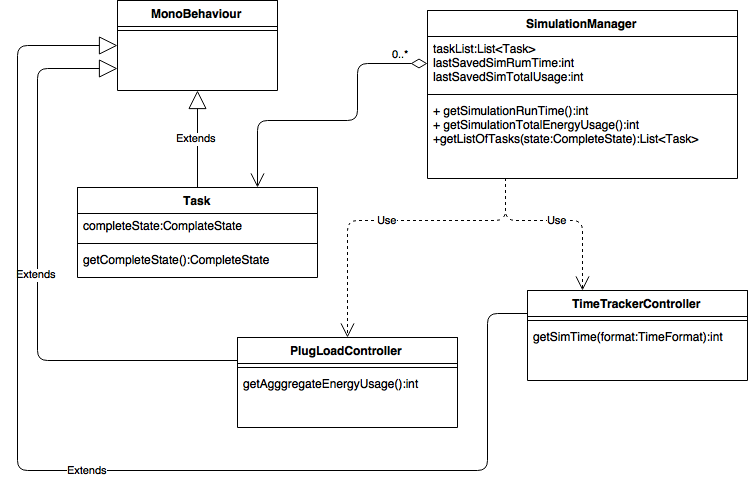
### Figure 28 - 802 Class Diagram



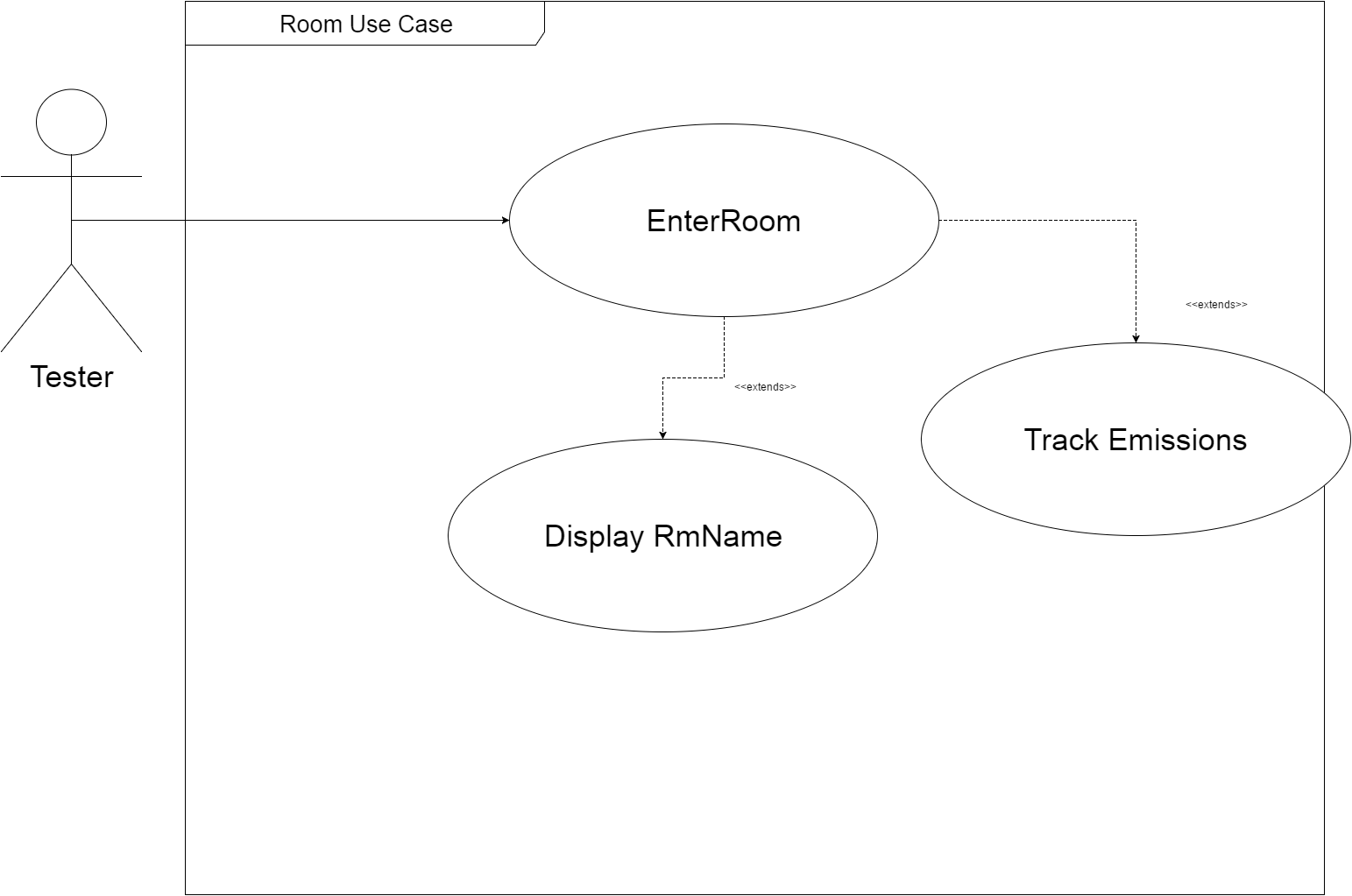
### Figure 29 - 803 Use Case Diagram



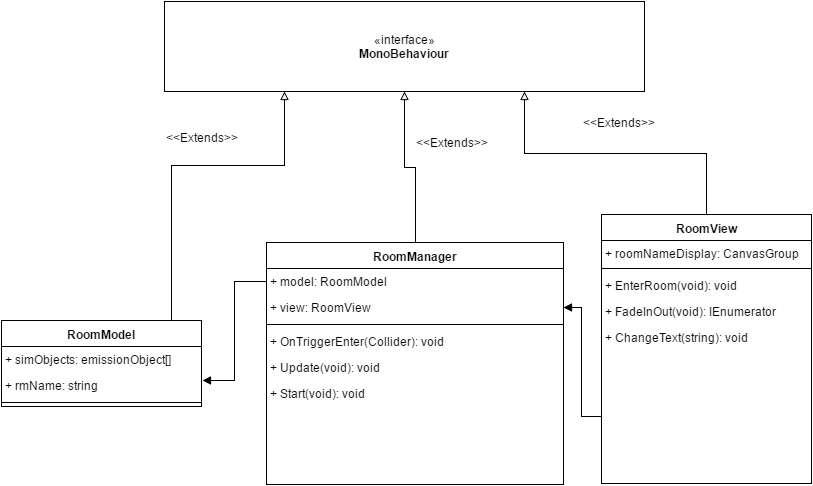
### Figure 30 - 803 Class Diagram



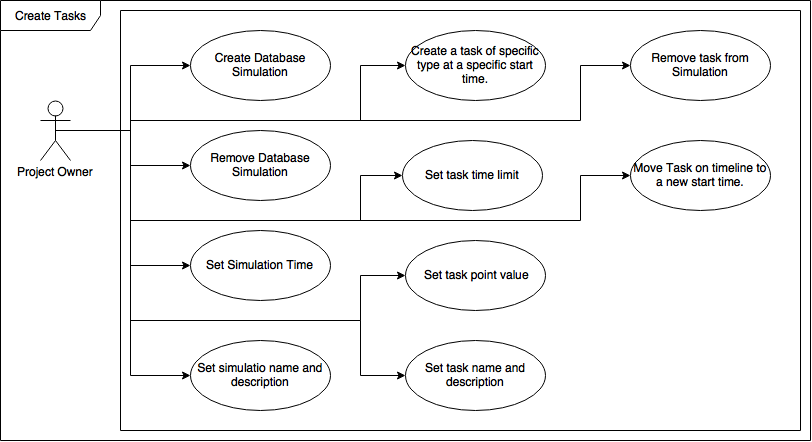
### Figure 31 - 815 Use Case Diagram



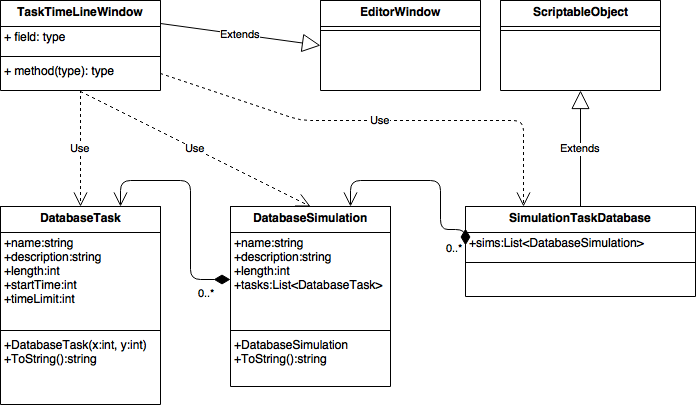
### Figure 32 - 815 Class Diagram



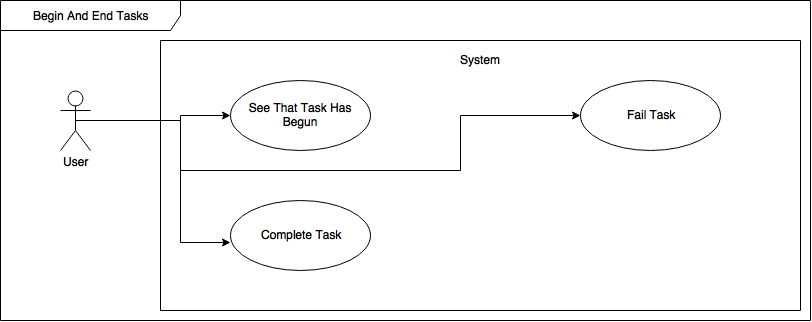
### Figure 33 - 826 Use Case Diagram



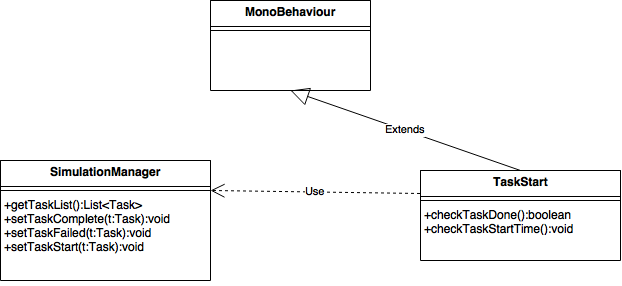
### Figure 34 - 826 Class Diagram



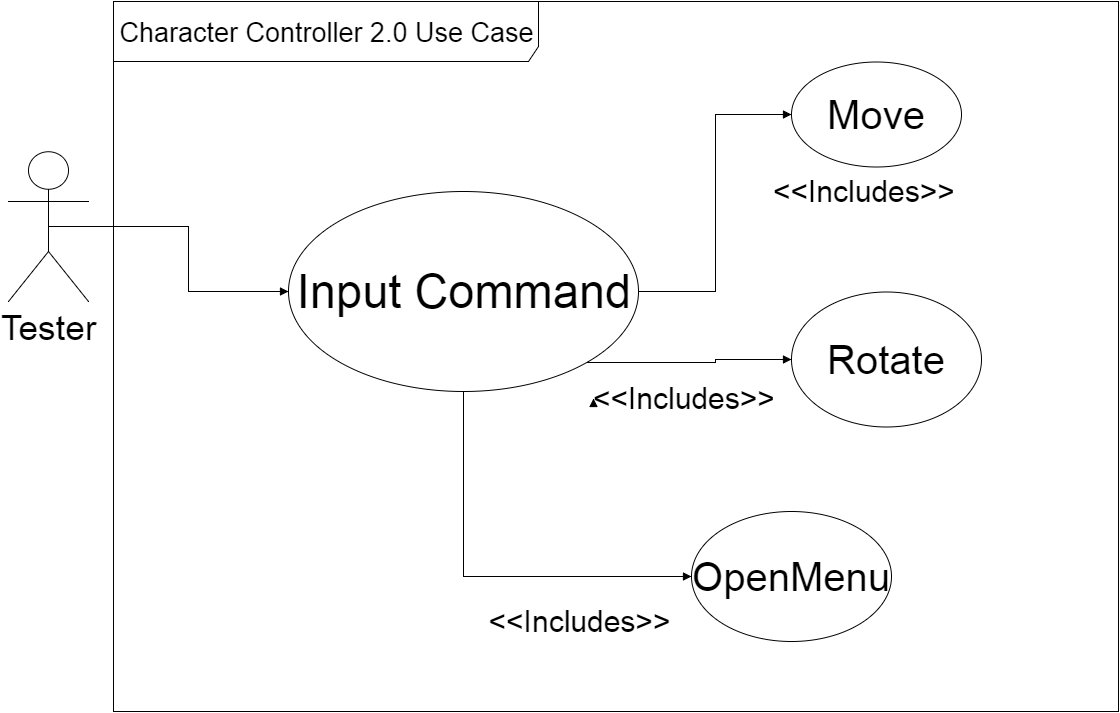
### Figure 35 - 827 Use Case Diagram



### Figure 36 - 827 Class Diagram



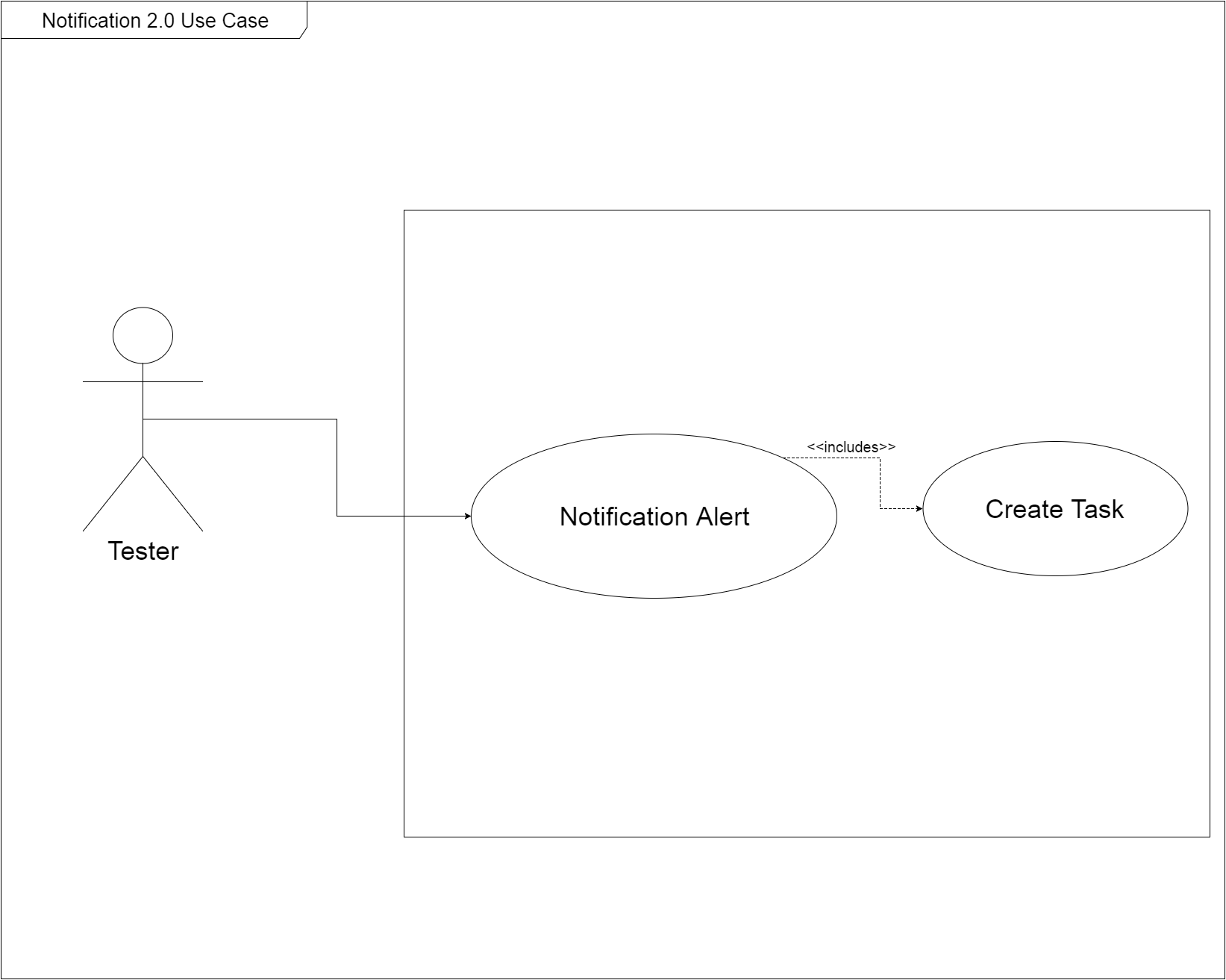
### Figure 37 - 832 Use Case Diagram



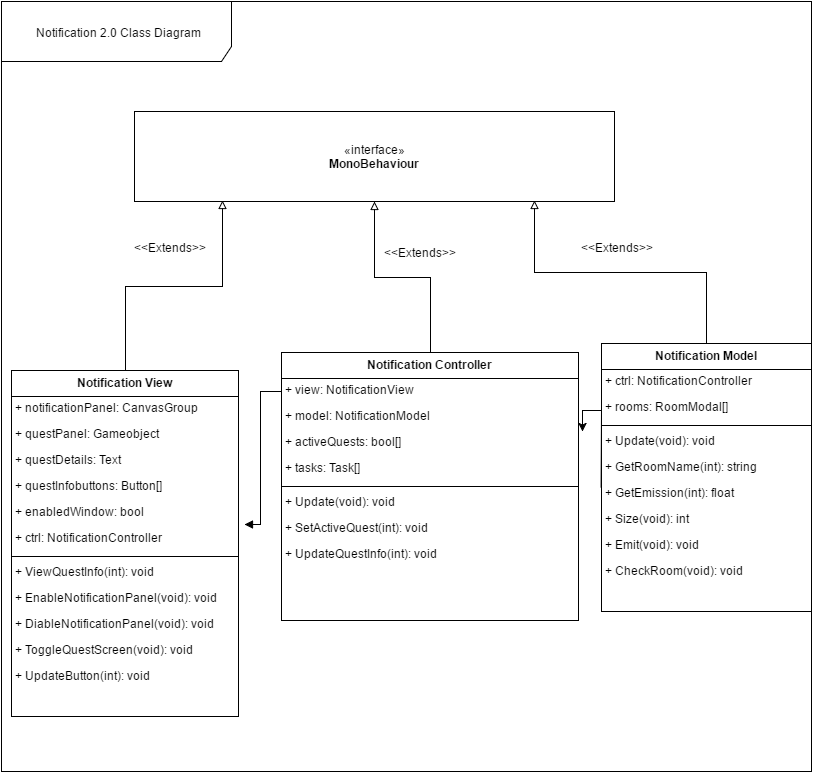
### Figure 38 - 832 Class Diagram



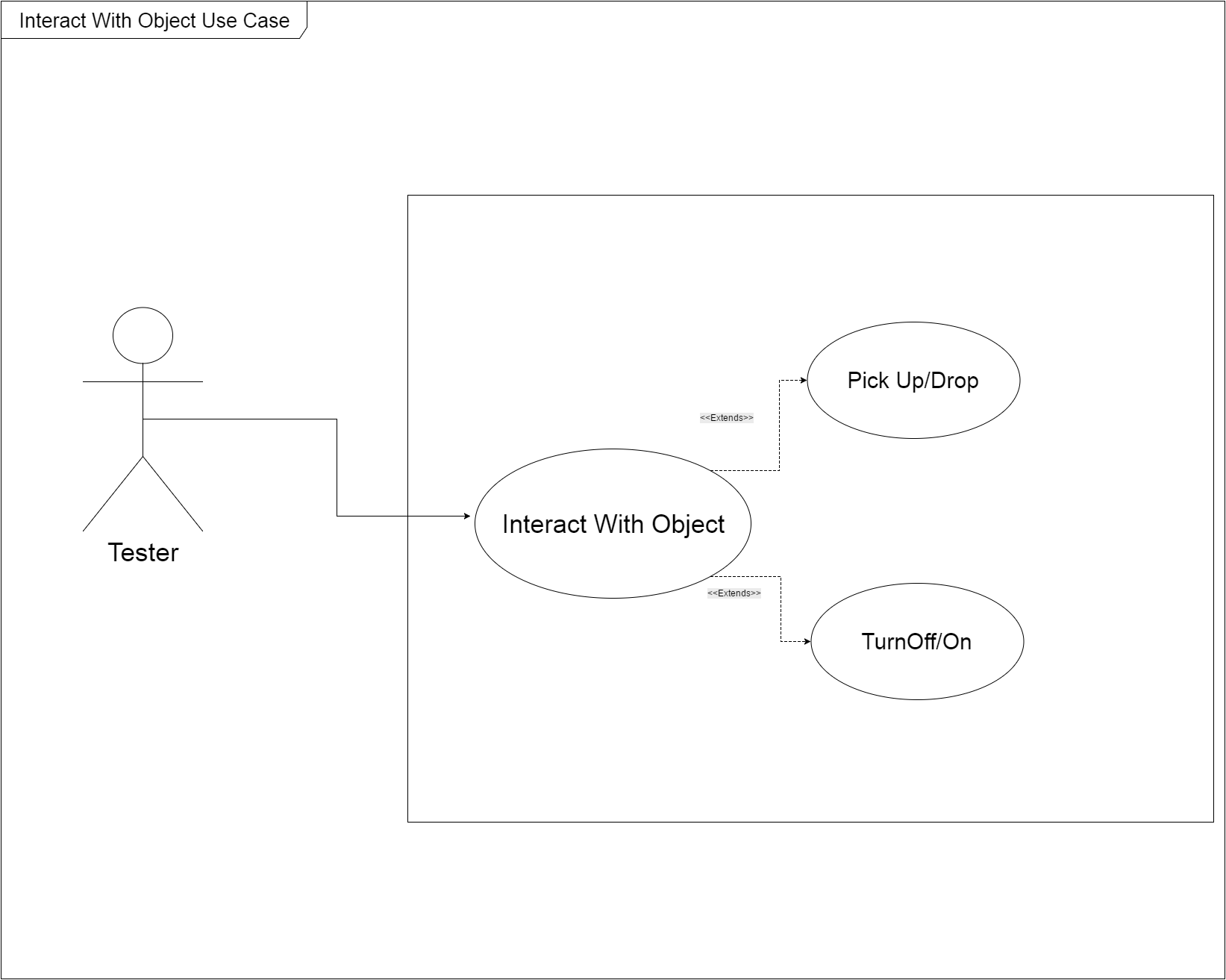
### Figure 39 - 833 Use Case Diagram



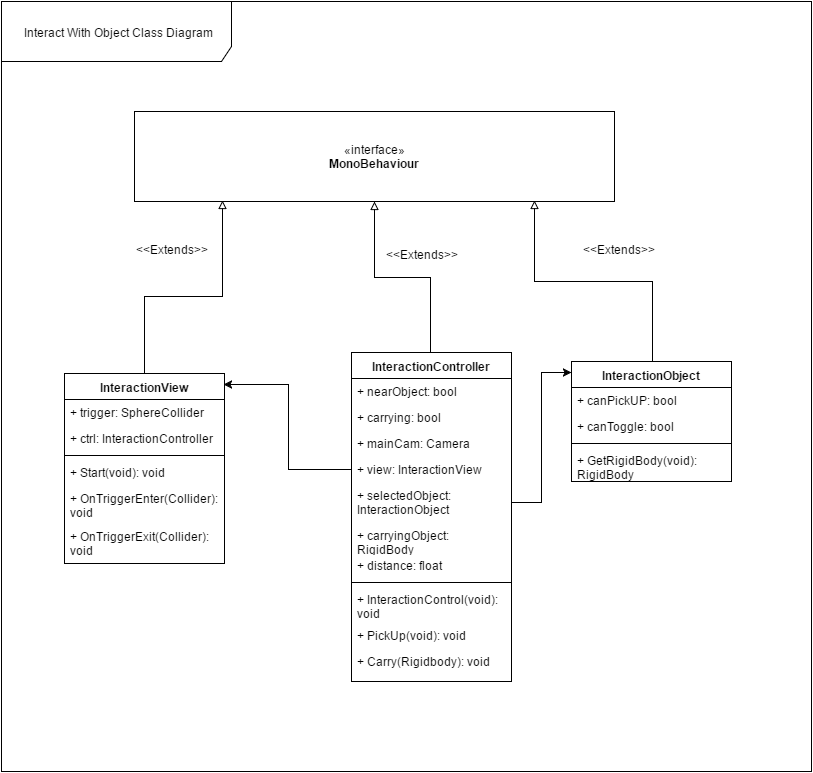
### Figure 40 - 833 Class Diagram



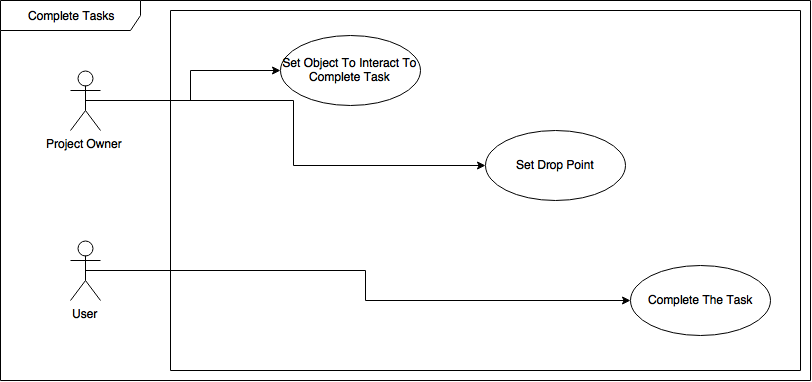
### Figure 41 - 846 Use Case Diagram



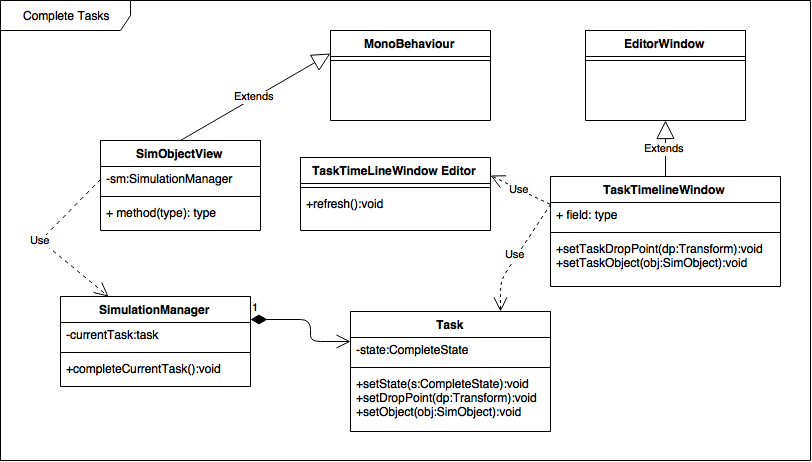
### Figure 42 - 846 Class Diagram



### Figure 43 - 847 Use Case Diagram

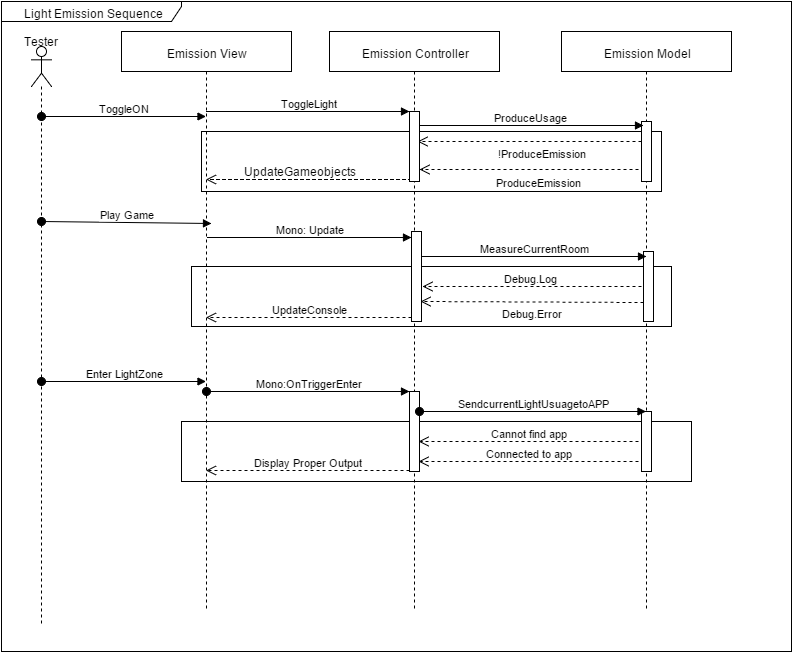


### Figure 44 - 847 Class Diagram

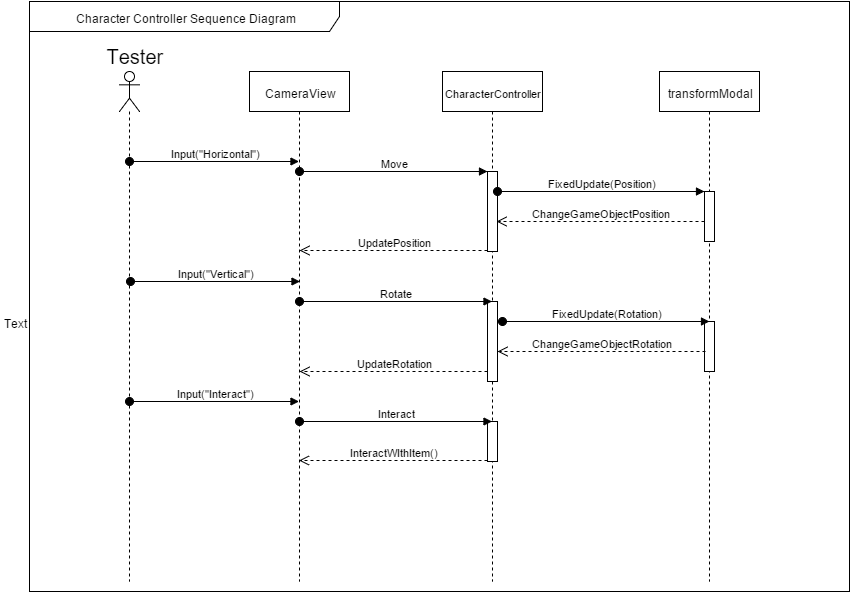


### ***Dynamic UML Diagrams***

### Figure 1 - 725 Sequence Diagram



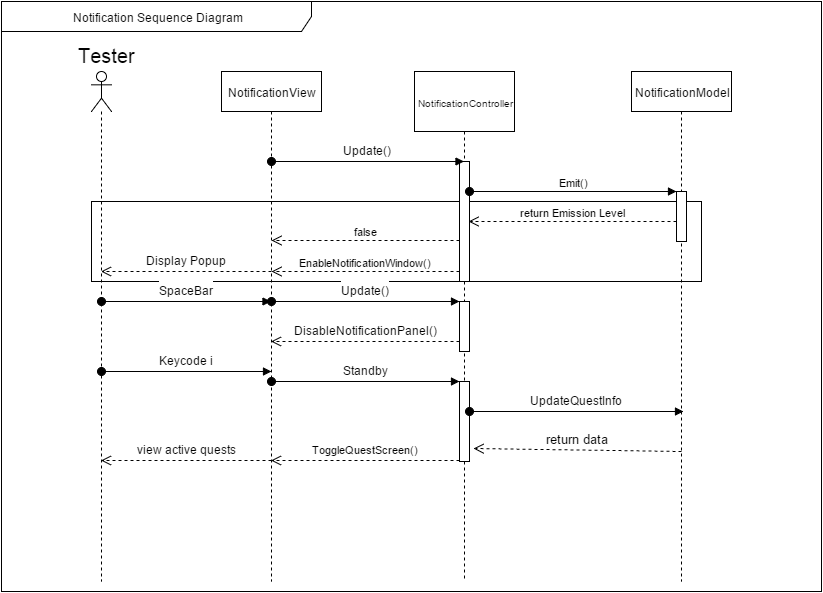
### Figure 2 -727 Sequence Diagram



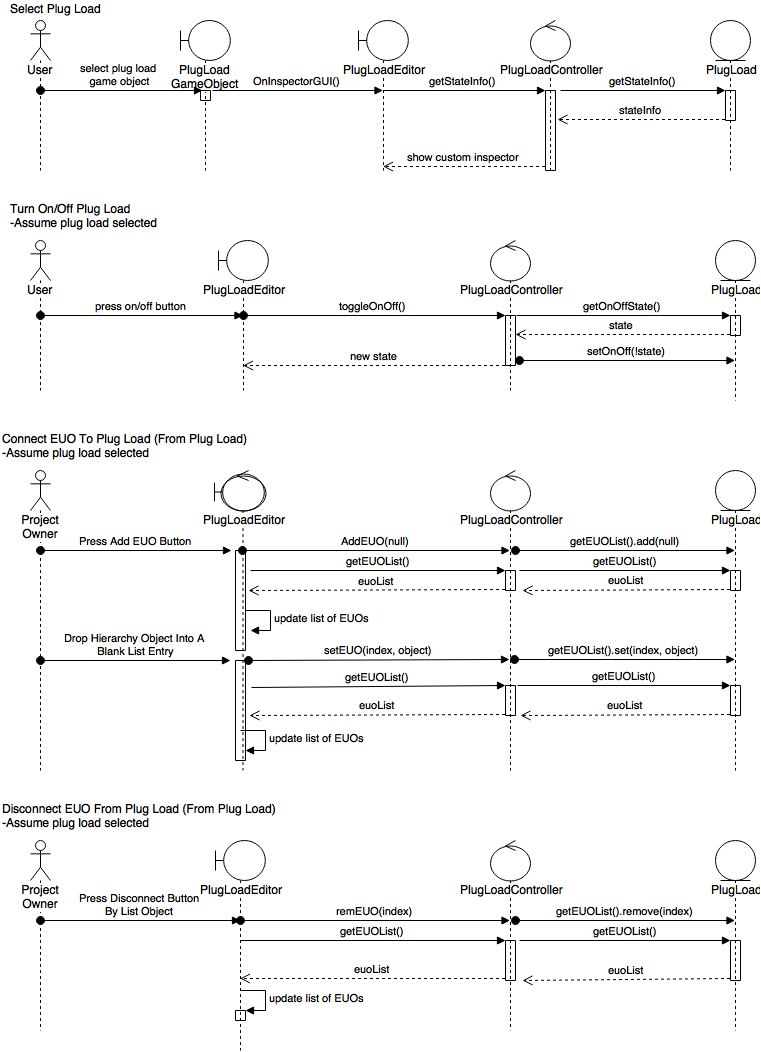
### Figure 3 - 730 Sequence Diagram

## 

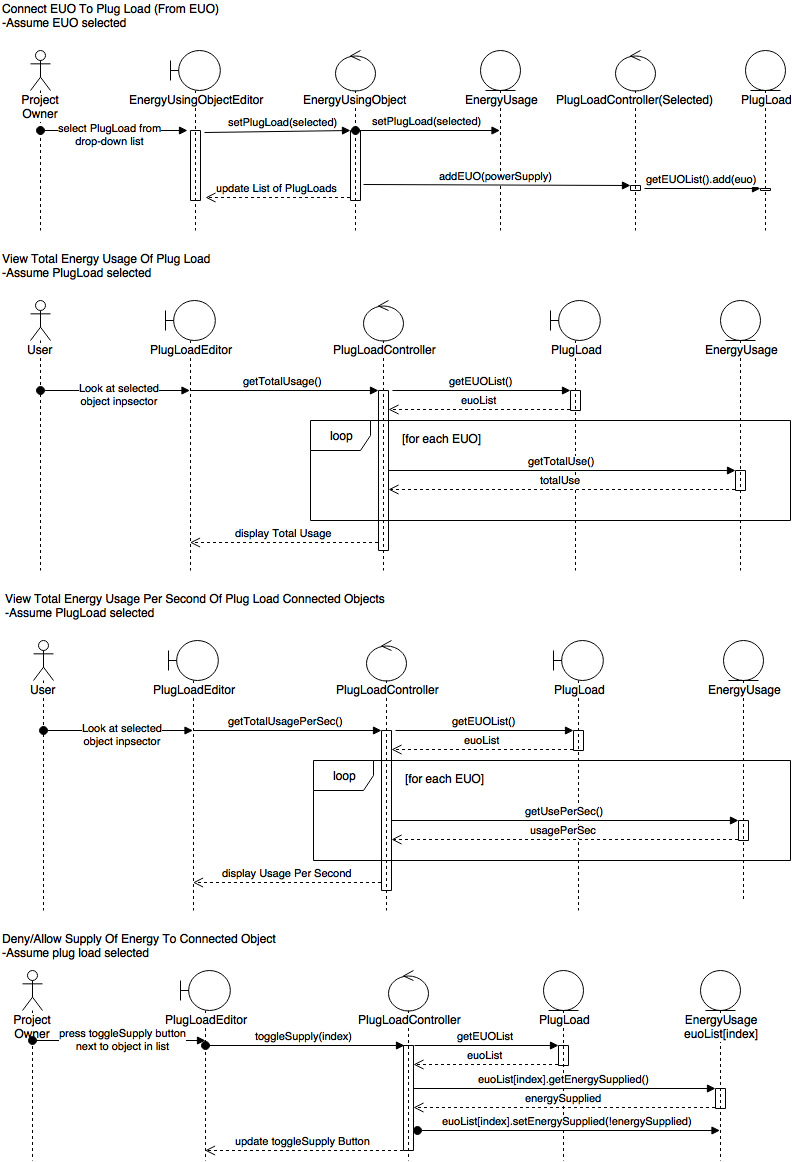
### Figure 4 - 732 Sequence Diagram



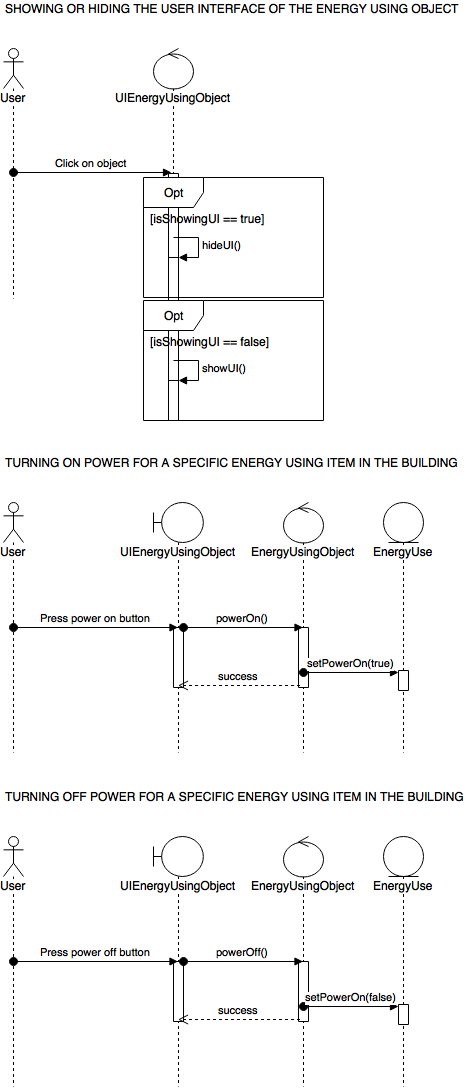
### Figure 5 -733 Sequence Diagram 1



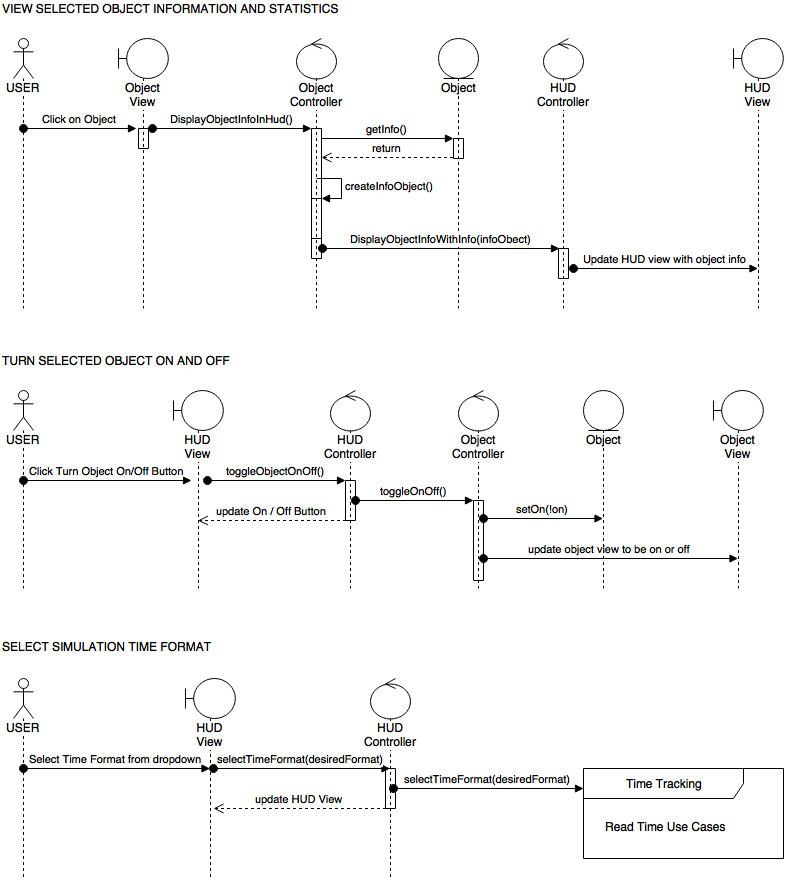
### Figure 6 - 733 Sequence Diagram 2



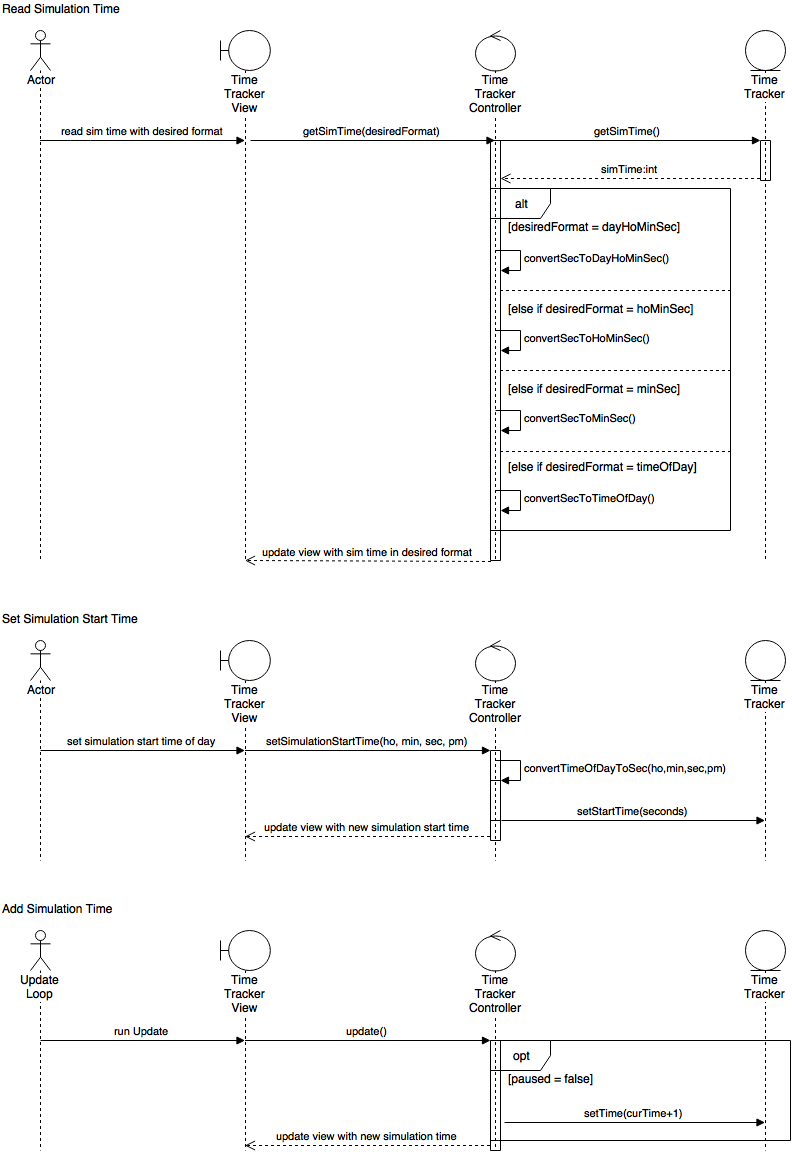
### Figure 7 - 734 Sequence Diagram



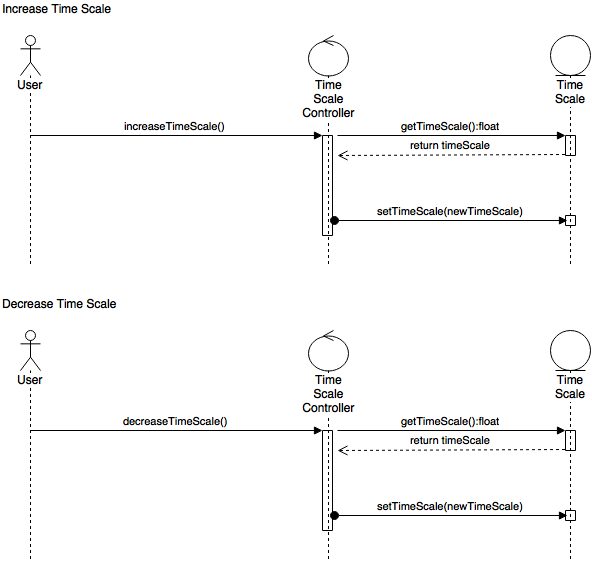
### Figure 8 - 749 - Sequence Diagram



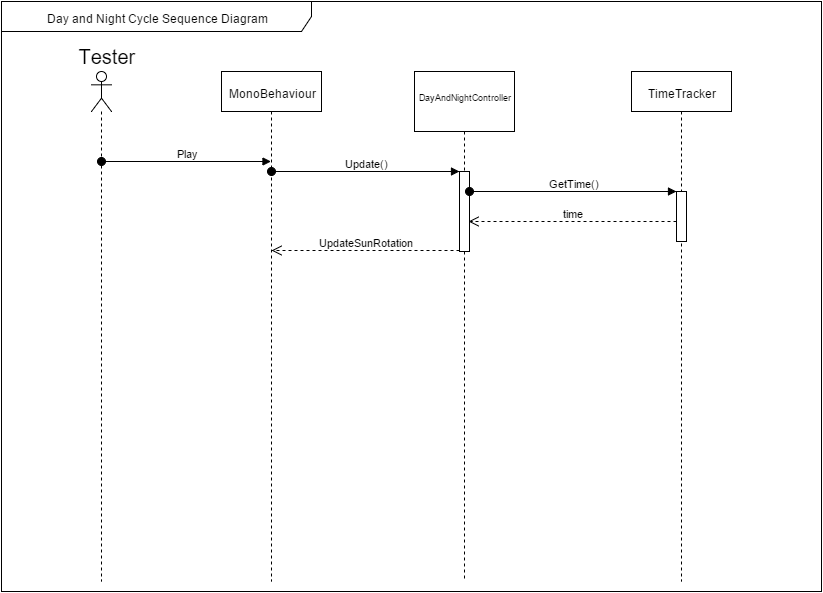
### Figure 9 - 750 Sequence Diagram



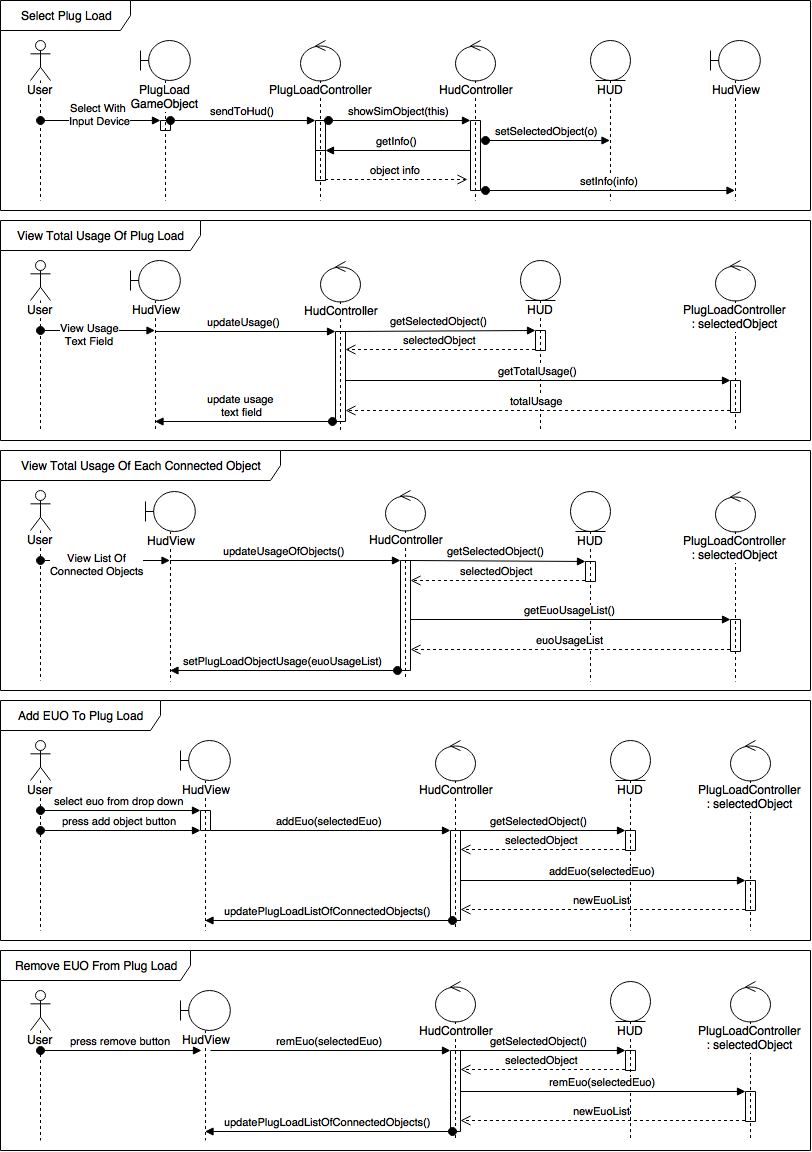
### Figure 10 - 751 Sequence Diagram



### Figure 11 - 754 Sequence Diagram



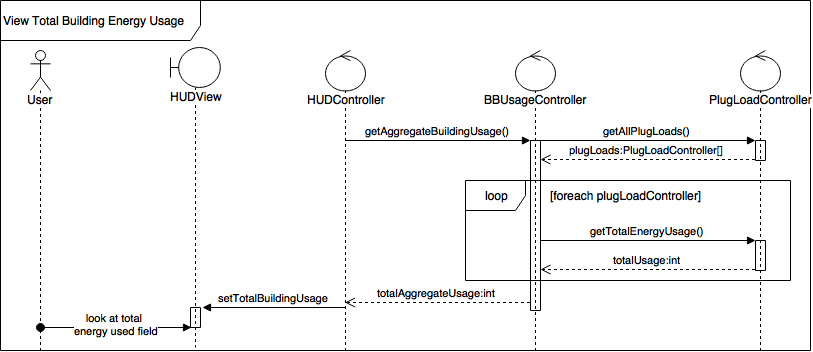
### Figure 12 - 779 Sequence Diagram 1



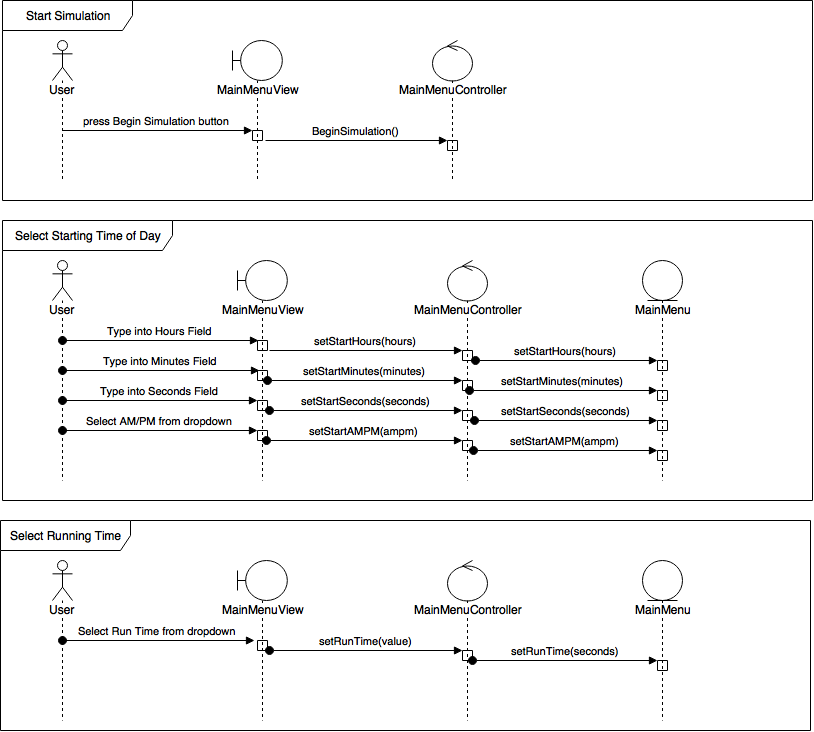
### Figure 13 - 779 Sequence Diagram 2



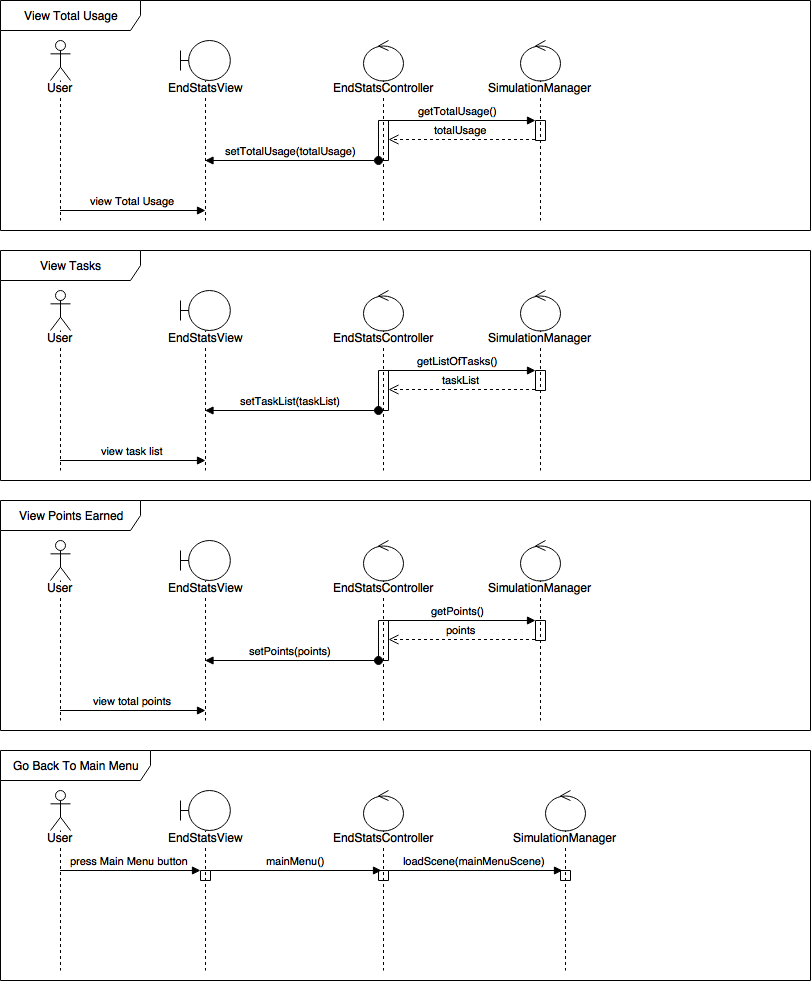
### Figure 14 - 780 Sequence Diagram



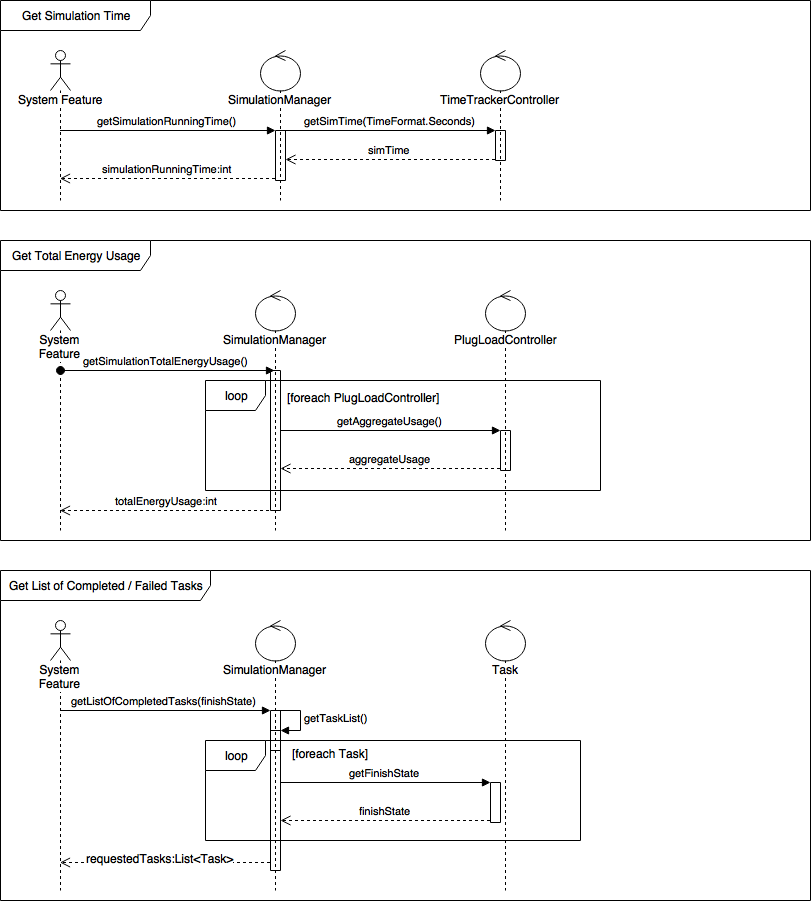
### Figure 15 - 801 Sequence Diagram



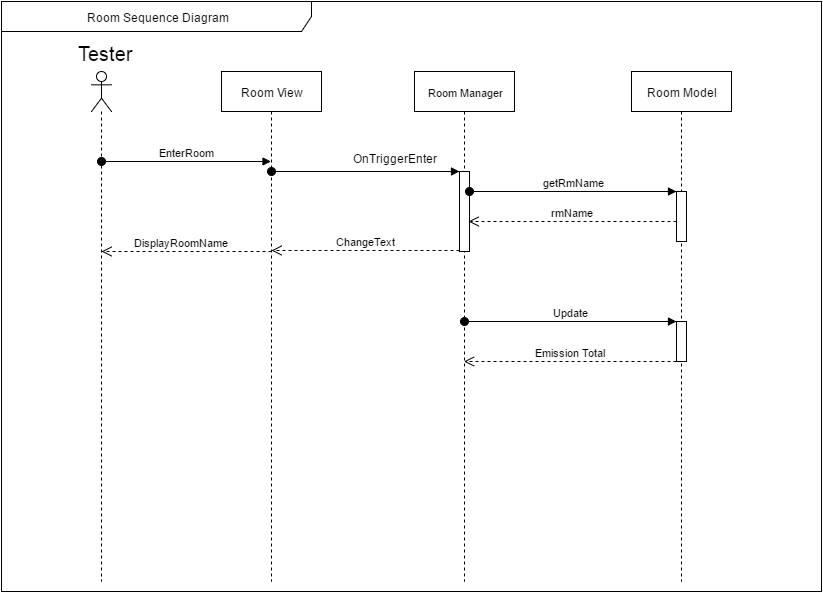
### Figure 16 - 802 Sequence Diagram



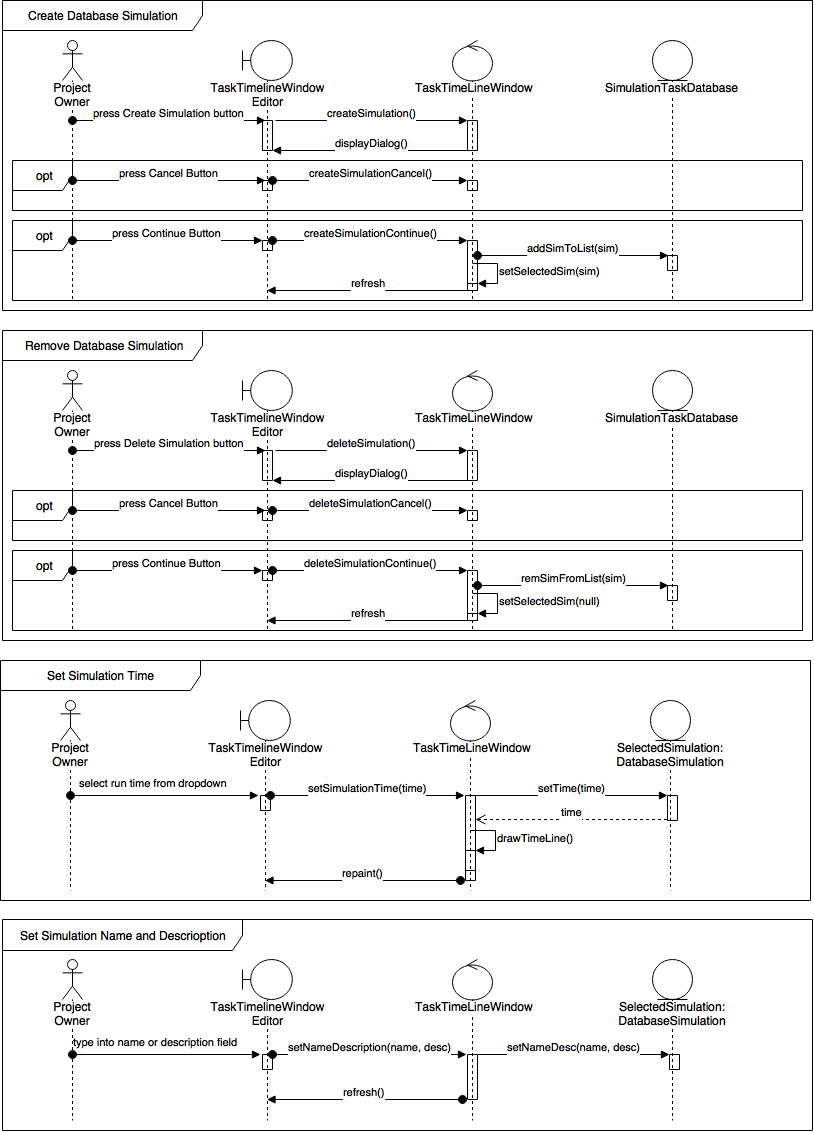
### Figure 17 - 803 Sequence Diagram



### Figure 18 - 815 Sequence Diagram



### Figure 19 - 826 Sequence Diagram 1



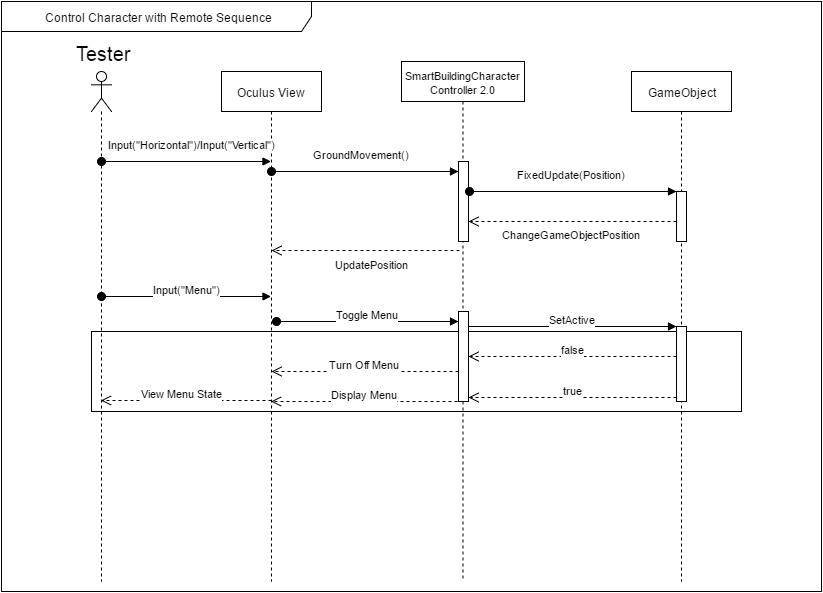
### Figure 20 - 826 Sequence Diagram 2

## 

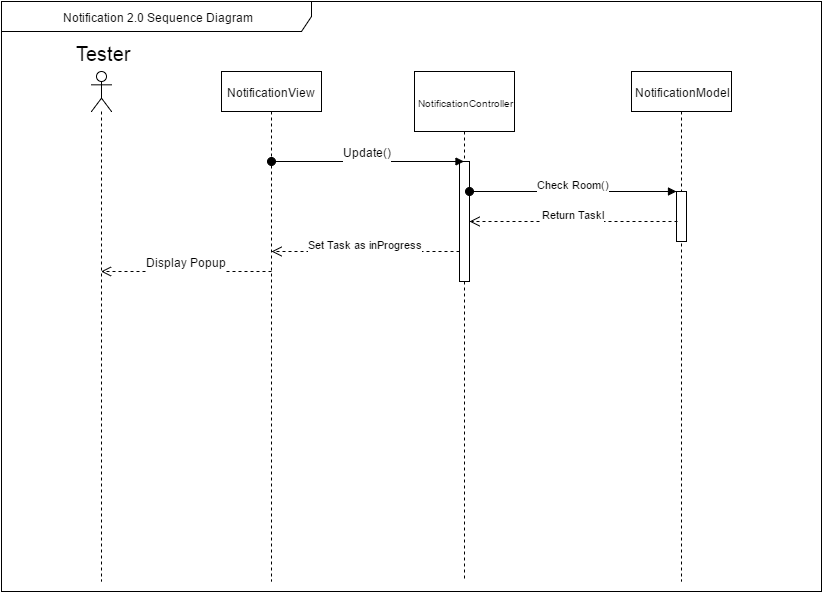
### Figure 21 - 827 Sequence Diagram



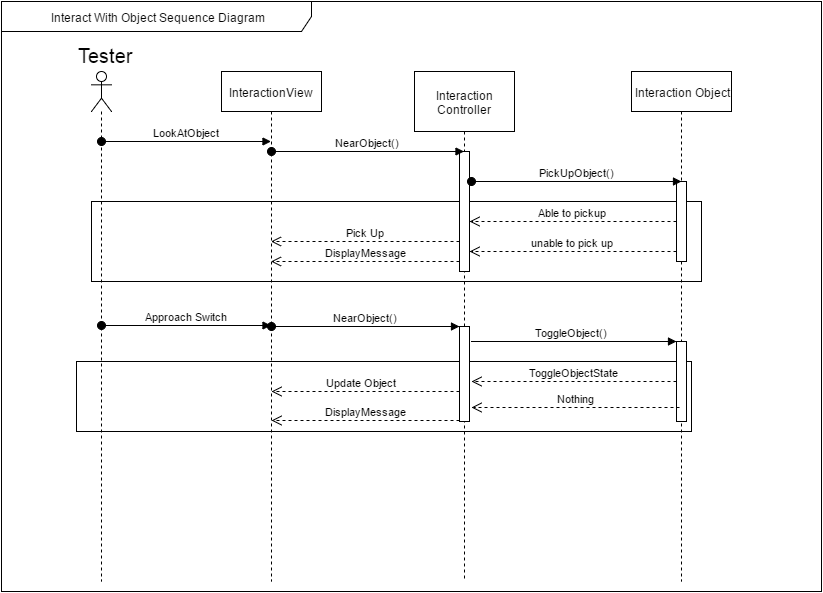
### Figure 22 - 832 Sequence Diagram



### Figure 23 - 833 Sequence Diagram



### Figure 24 - 846 Sequence Diagram



### Figure 25 - 847 Sequence Diagram



## 

## 

## **Appendix B - User Interface Design**

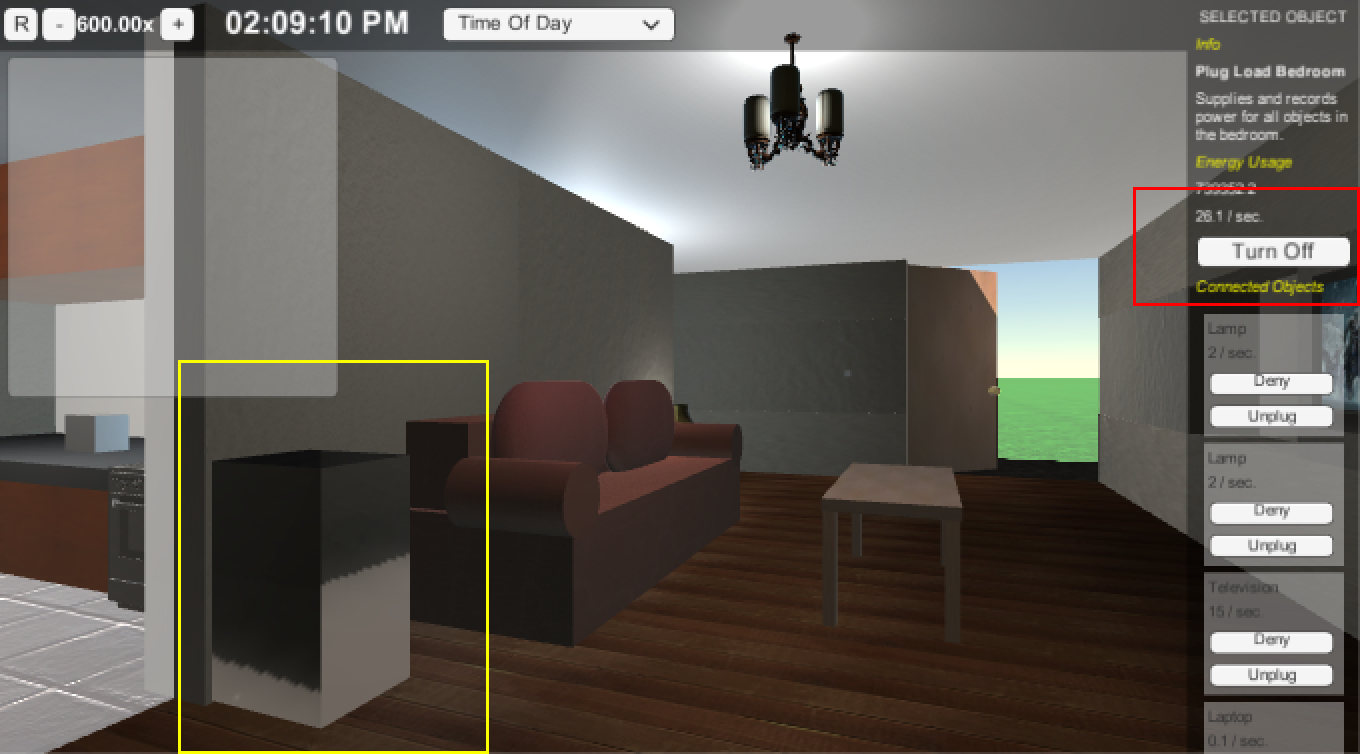
Include screenshots of the user interface of your system. For new versions of existing sytems, include only screenshots of the new or modified aspects of the user interface.

There’s no need for introducing this section.

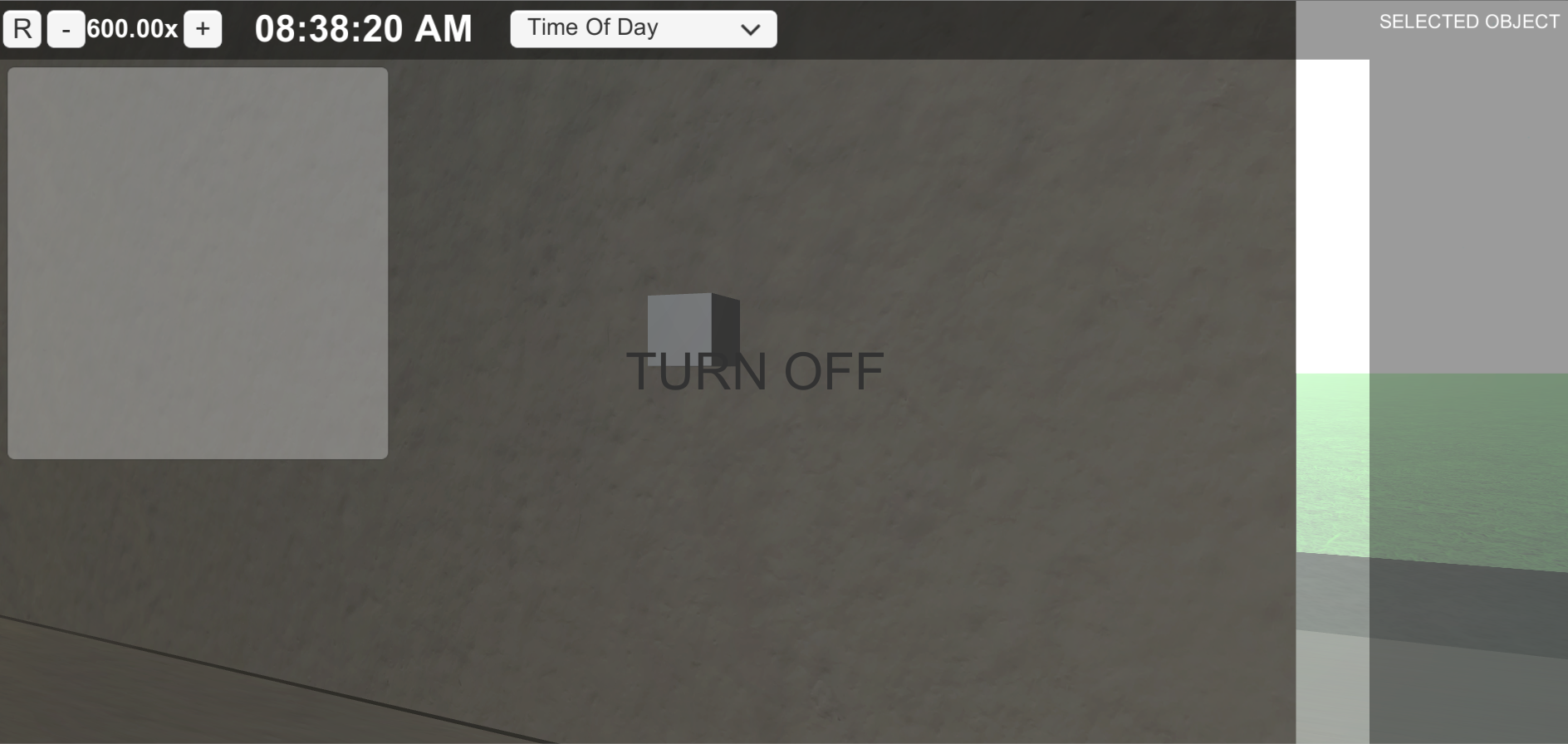
### Figure 1 - HUD



### Figure 2 - Plug Load in HUD



### Figure 3 - Light Switch UI



## **Appendix C - Sprint Review Reports**

**Sprint 1 Report**

Date: 1/29/16

Attendees:Justin Fletcher, Emmanuel Vinas, Ali Mostafavi, Leonardo Bobadilla, Peeraya Inyim

Start time:6:13

End time:6:30

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* User Story #725
* #730
* #731
* #734

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint 2 Report**

Date: 2/12/16

Attendees: Justin Fletcher, Emmanuel Vinas, Peeraya Inyim

Start time: 6:00pm

End time: 6:45pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* User Story #749 : As a user, I would like to have a HUD that displays the time of day, what room I’m in, possibly the weather, and other things related to the simulation, so that I can know things about the simulation in the same way I would know things about the world.
* #750 : As a user or tester, I need to know how much time has passed in the simulation so that I can know how much a unit of energy has been used per unit of time.
* #751: As a tester, I would like to be able to set the time scale (in relation to the real world) of simulation, in order to speed up or slow down tests in the simulation.
* #727: As a tester I would like to move around the simulated house in order to travel between the different rooms and be able to reach items I can interact with
* #747: As a User, I want to be able to see the house I’m in, to better immerse myself in the simulation.
* #754: As a tester, I would like to simulate a day and night transition so that we can adjust the use of lighting within the simulated house.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* N/A

**Sprint 3 Report**

Date: 2/28/16

Attendees: Justin Fletcher, Emmanuel Vinas, Peeraya Inyim, Ali Mostafavi

Start time: 6:00pm

End time: 6:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* User Story #732 : As a tester, I would like to receive simulated Notifications in the simulation so that they can be tested before using a notification system directly to the Application on an Android device.
* User Story #733 : As a tester, I would like to have a simulated hardware device that can be placed in any room in the simulation and provide energy, so that I can connect energy using devices into it to record their usage data.
* User Story #780: As a user, I would like to see in the HUD the total energy usage of the building, so that I can see how much energy is being used by all objects together.
* User Story #779: As a user I would like to be able to see plug load information when I select one, so that I can see how much energy the room is using, as well as any device in the room attached to the plug load.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* N/A

**Sprint 4 Report**

Date: 3/12/16

Attendees: Justin Fletcher, Emmanuel Vinas, Peeraya Inyim

Start time: 6:00pm

End time: 6:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* #800 - Universal Real Time Lighting
* #801 - Start the simulation with a main menu
* #802 - End the simulation with statistics screen
* #803 - Have a simulation manager
* #815 - Monitor room emission

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* N/A

**Sprint 5 Report**

Date: 4/5/16

Attendees: Justin Fletcher, Ali Mostafavi

Start time: 5:30pm

End time: 6:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* #826 - Create Tasks
* #827 - Begin and End Tasks
* #832 - Control Character with Console Controller
* #833 - Create Notifications Dynamically

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* N/A

**Sprint 6 Report**

Date: 4/25/16

Attendees: Justin Fletcher, Ali Mostafavi

Start time: 4:30pm

End time: 5:30pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* #846 - interact with objects during tasks.
* #847 - Complete Tasks
* #848 - Build House

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* #780 - Display total Energy Usage in HUD

## **Appendix D - Sprint Retrospective Reports**

**Sprint 1 Retrospective**

Date: 1/29/16

Attendees: Justin Fletcher, Emmanuel Vinas, Ali Mostafavi, Leonardo Bobadilla, Peeraya Inyim

Start time: 6:02

End time: 6:13

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Mostly. There was a bit of catch up at the end.
* Did we do a good job estimating the points (time required) for each user story?
  + Probably not. The amount of work dedicated to documentation and getting used to Mingle was not properly accounted for, but overall, good.
* Did each team member work as scheduled?
  + Yes
* No Unit or Integration testing was done.

What went right?

* The development itself.
* Both team members have experience in Unity.

How to address the issues in the next sprint?

* How to improve the process?
  + Plan for time required for documentation.
* How to improve the product?
  + Continue working on it. Need to do some sprint planning to see what’s most important and make sure it gets done.

**Sprint 2 Retrospective**

Date: 2/12/16

Attendees: Justin Fletcher, Emmanuel Vinas, Peeraya Inyim

Start time: 6:00pm

End time: 6:45pm

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + We’re getting better at it as the project goes on.
* Did each team member work as scheduled?
  + Yes

What went right?

* We were able to implement every user story we set out to in the sprint planning.

How to address the issues in the next sprint?

* How to improve the process?
  + Emmanuel and Justin haven’t put their user stories together in one project yet, to see if everything works together as expected. So that is one thing we should do for the next sprint.
* How to improve the product?
  + Make the user experience much easier with simpler controls, and we want to go to the hardware lab to see how the Oculus interfaces with the the simulation.

**Sprint 3 Retrospective**

Date: 2/28/16

Attendees: Justin Fletcher, Emmanuel Vinas, Peeraya Inyim, Ali Mostafavi

Start time: 6:00PM

End time: 6:30PM

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes
* Did each team member work as scheduled?
  + Yes

What went right?

* Able to complete all user stories

How to address the issues in the next sprint?

* How to improve the process?
  + The UI of the simulation should look a little more like the UI from release 3 of smart building. Also the lighting of the inside of the house is not very good so these are things to work on next sprint.
* How to improve the product?
  + Fix the user controls and lighting of the scenes.
  + The UI design to look more like the application’s UI.
  + Add a no App mode for comparison research.

**Sprint 4 Retrospective**

Date: 3/12/16

Attendees: Justin Fletcher, Emmanuel Vinas, Peeraya Inyim,

Start time: 6:00PM

End time: 6:30PM

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes
* Did each team member work as scheduled?
  + Yes

What went right?

* Able to complete all user stories

How to address the issues in the next sprint?

* How to improve the process?
  + Weekly meetings
* How to improve the product?
  + Make controls to work with oculus.
  + Try to break dependency of drop downs.

**Sprint 5 Retrospective**

Date: 4/5/16

Attendees: Justin Fletcher, Ali Mostafavi

Start time: 5:30PM

End time: 6:30PM

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes
* Did each team member work as scheduled?
  + Yes

What went right?

* Able to complete all user stories

How to address the issues in the next sprint?

* How to improve the process?
  + More meetings with Ali to ensure the simulation flow matches his vision.;
* How to improve the product?
  + Completion of tasks.
  + Change the house.
  + 5 Min simulation.
  + Able to start and stop simulation with or without app.
  + Make stats look like app.

**Sprint 6 Retrospective**

Date: 4/25/16

Attendees: Justin Fletcher, Ali Mostafavi, Emmanuel Vinas

Start time: 4:30PM

End time: 5:30PM

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes
* Did each team member work as scheduled?
  + Yes

What went right?

* Finished the most important user stories.

How to address the issues in the next sprint?

* This was the last sprint with new User Stories and tasks.

# **References**