Project Ngeen

Analysis and Design Document

Student: Daian Dragos

**Group: 30434**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 3/20/2016 | 1.0 | Initial Commit | Daian Dragos |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

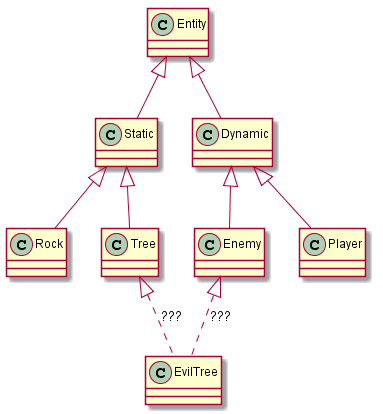
1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

The project is to be used in game development in order to simplify some parts of the code and develop on more platforms. The project uses Java and entity component system for design.



The image above describes the need for such an organized pattern, in order to not have as many different classes, we have components that build up entities.

# Elaboration – Iteration 1.1

# Domain Model

# 

Components Model

# 

UI\* specifies all the UI components.

Some components change their behavior according to others. For example, every component changes it’s position according to the point component, if the object has it.

Ngeen holds all the references to objects. Systems process entities. Entities contain components which hold data or logic. More on this will be explained on the next chapter.

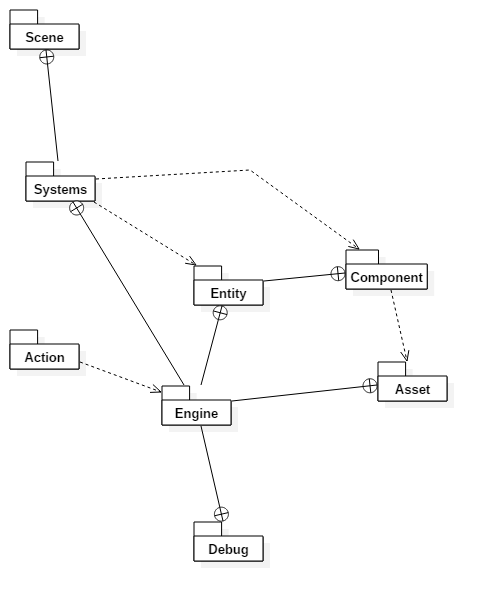
# Architectural Design

## Conceptual Architecture

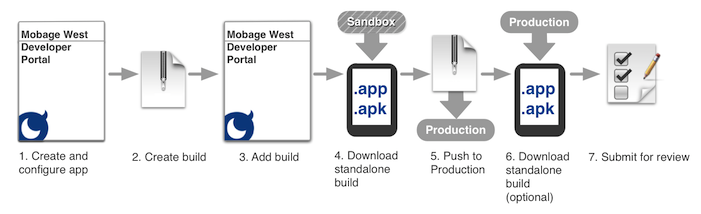
## 

Entities are held in a factory, such are components and systems. Components are observers and observable, that is, they receive and process their own events. Such are the systems. Also, user commands are saved in a CommandFactory to be later processed if needed(do-undo).

## Package Design



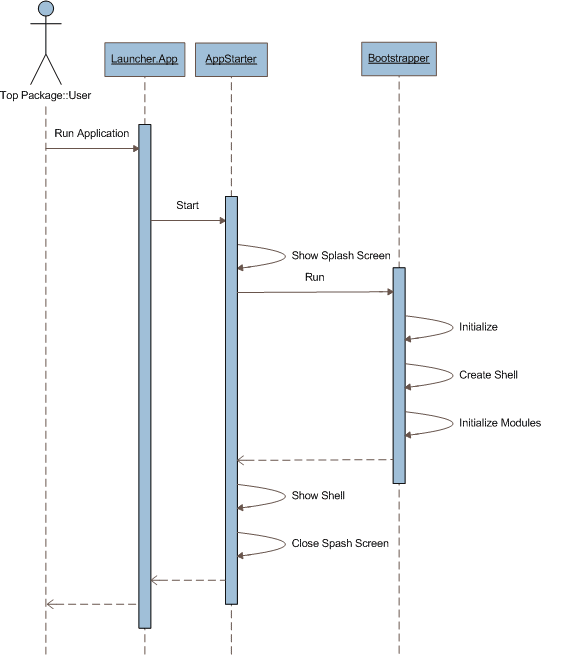
## Component and Deployment Diagrams



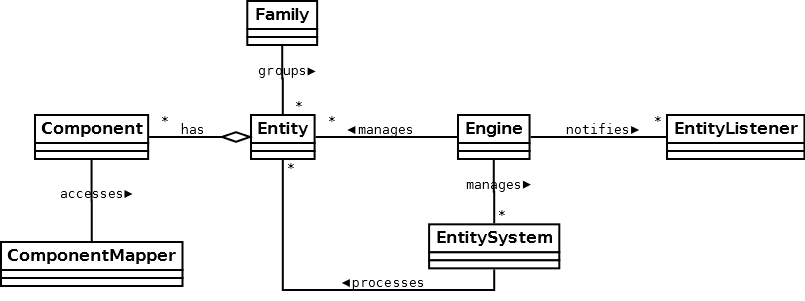
# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

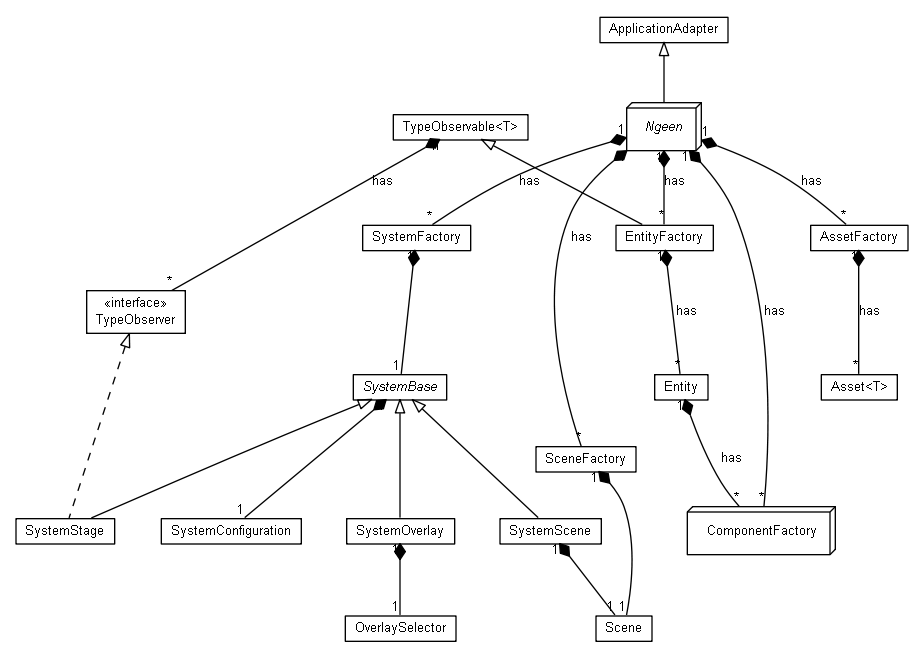


*Normal game sequence diagram.*



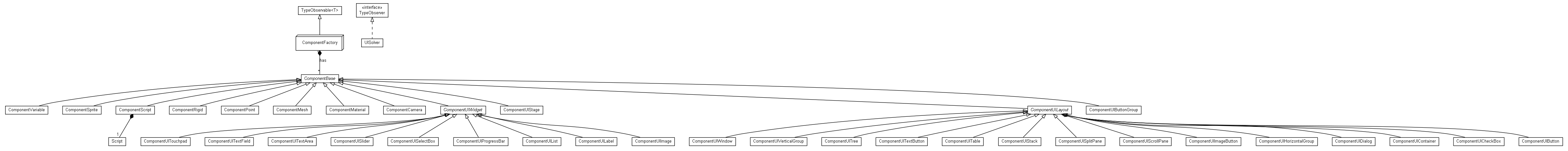
*Comunication diagram, above.*

## Class Design



# Data Model

*Same as the one before, plus addition of some UI elements. All the intended components are below:*



# Unit Testing

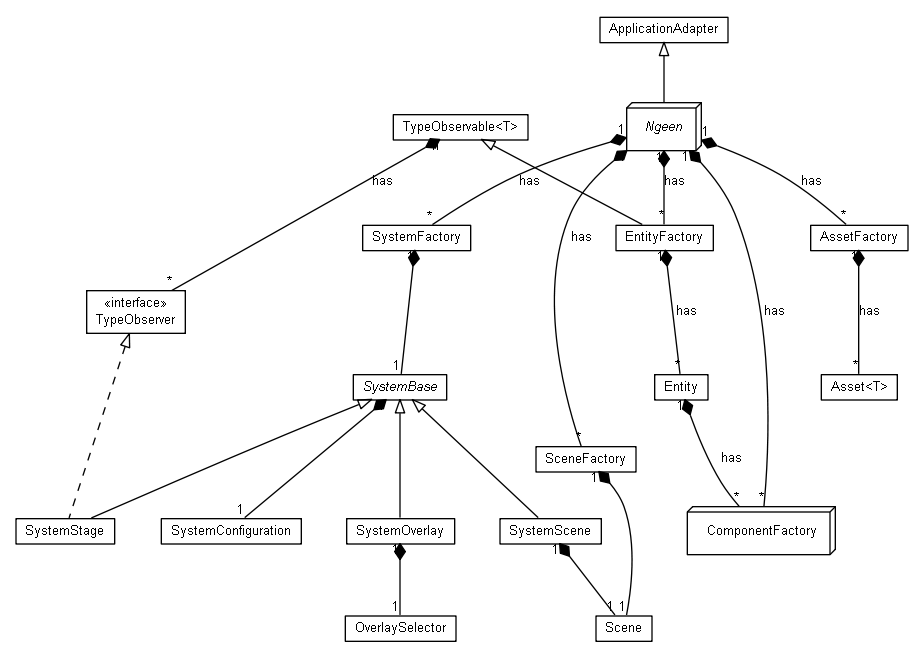
*Mainly stress test for different scenarios, such as create or remove components/entities.*

# Elaboration – Iteration 2

# Architectural Design Refinement

# Design Model Refinement

*With this design, factories hold everything, and it has a nice layered like touch to it. Also, it has the component entity system design pattern and flyweight.*



# Construction and Transition

# System Testing

*Scenes were used as testers. Entities, components and assets were made/updated/removed for stress test.*

# Future improvements

*Physics components, more UI support and further optimizations can be made.*

# Bibliography

<https://en.wikipedia.org/wiki/Entity_component_system>

<https://libgdx.badlogicgames.com/>

https://developer.mobage.com/ - diagram