

System design document for Group 16

Group 16

October 7, 2022

1 Introduction

1.1 Definitions, acronyms, and abbreviations

| | |
|----------|-----------------------------|
| Feyrune: | The name of the application |
| DoD | Definition of Done |

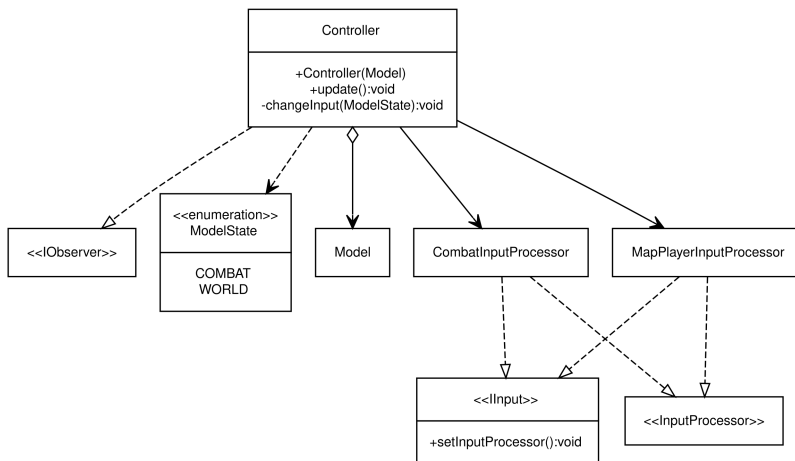
2 System architecture

3 System design

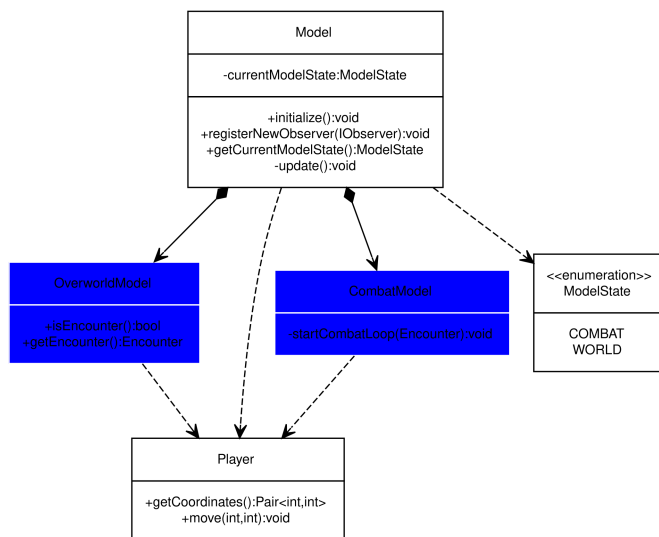
3.1 Packages

- controller
 - controller.combat
 - controller.combat.ui
 - controller.enums
- interfaces
- model
 - model.combat
 - model.combat.actions
 - model.combat.actions.abilities
 - model.combat.creatures
 - model.creature
 - model.overworld
 - model.overworld.encounter
 - model.overworld.map
 - model.player
- Util
- view
 - view.combat
 - view.components
 - view.overworld
 - view.overworld.textureMap
 - view.player
 - view.scenes
 - view.utils

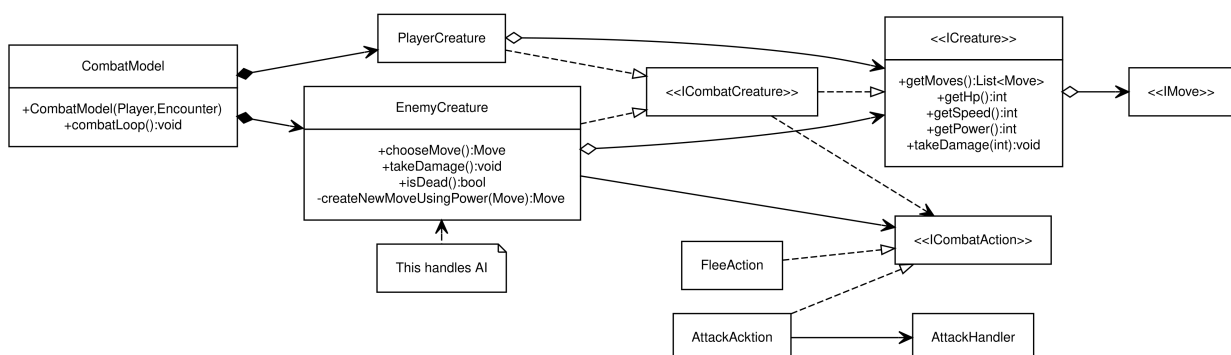
3.2 Controller



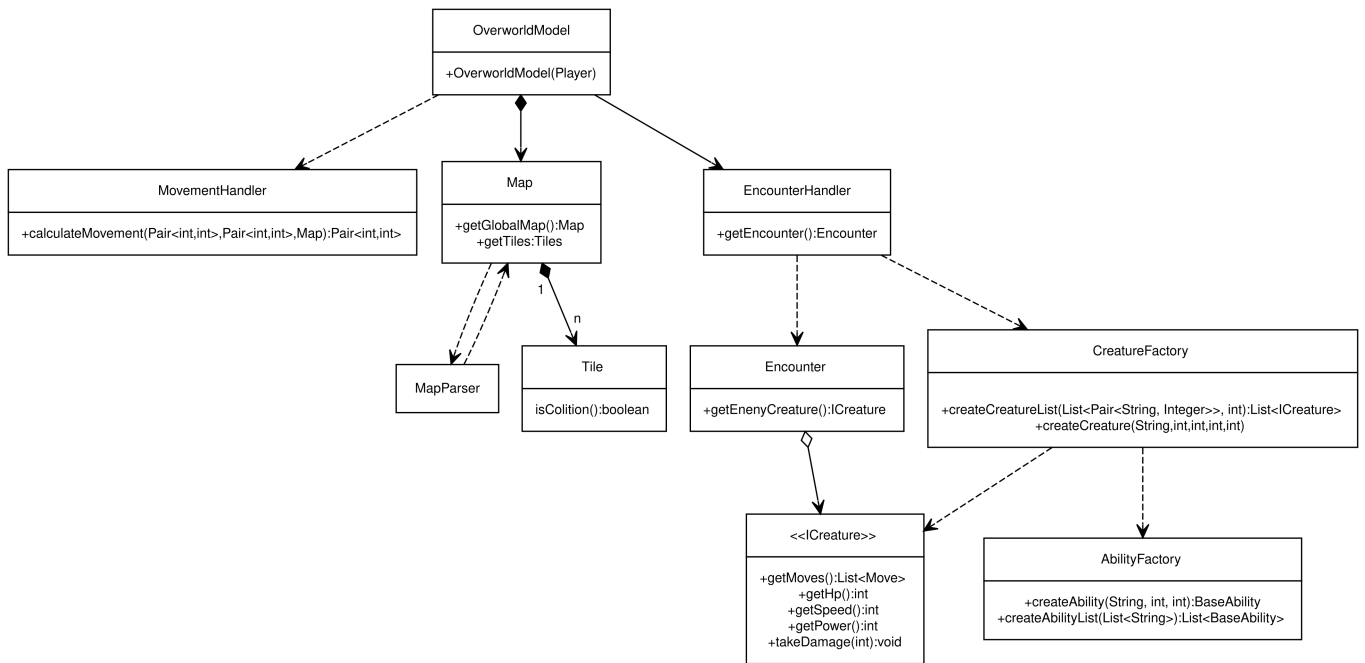
3.3 Model



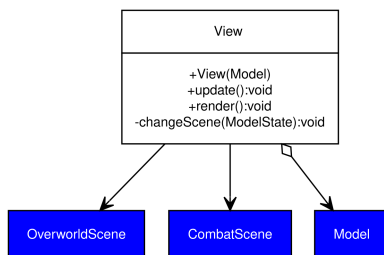
3.4 CombatModel



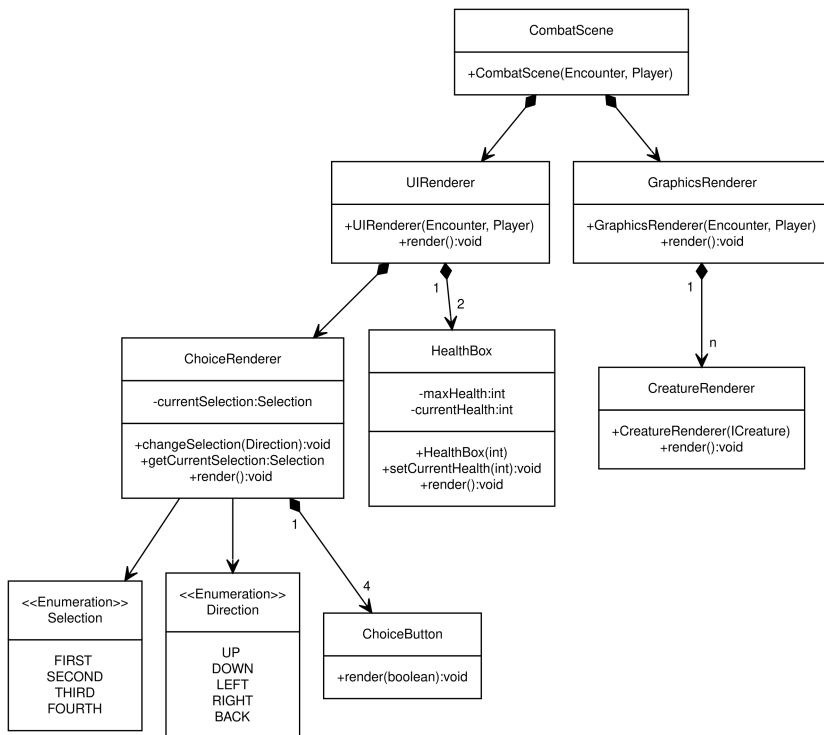
3.5 OverworldModel



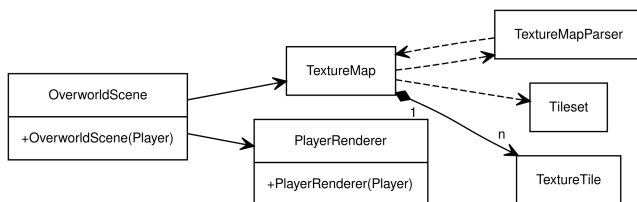
3.6 View



3.7 CombatScene



3.8 OverworldScene



3.9 Design Patterns

Throughout the code, Group 16 has tried to implement design patterns wherever feasible. Examples of design patterns used are:

Model-View-Controller

Factory pattern

State pattern (hopefully)

Observer pattern

Singleton

4 Persistent data management

No data is stored and all images are stored using tileset .png files in the assets folder. The maps are stored in a similar vein by using .tmx files that are also stored in the assets folder. The .tmx files contain *all* relevant data for the given map.

5 Quality

5.1 Access control and security

6 References