System design document for Group 16

Group 16

10th October 2022

1 Introduction

This is a design document explaining the inner workings of Feyrune, a Pokémon-style RPG, but with the ability to be easily extended with new functionality to differentiate itself.

1.1 Definitions, acronyms, and abbreviations

Feyrune:	The name of the application
DoD	Definition of Done

2 System architecture

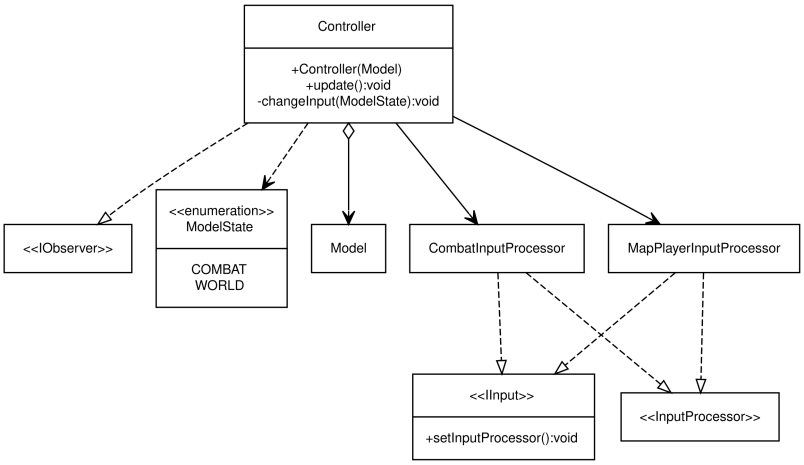
Feyrune was mainly created using libgdx , a java game engine, to render and run the game [1], and Tiled to create and design the maps [2].

3 System design

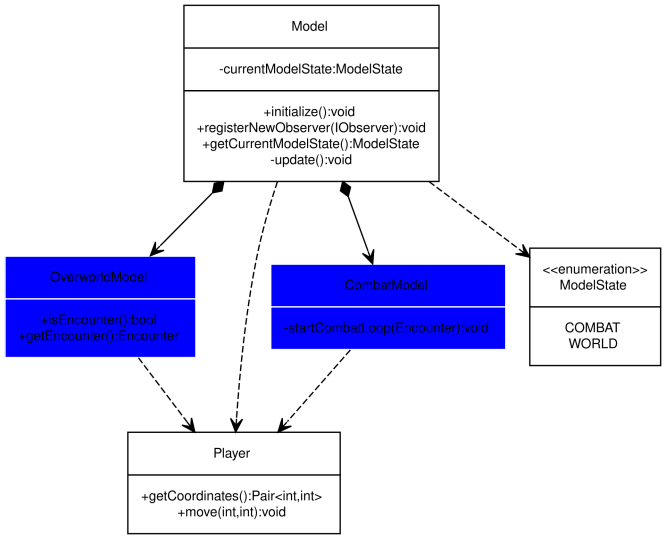
3.1 Packages

- 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
  - vew.player
  - view.scenes
  - view.utils

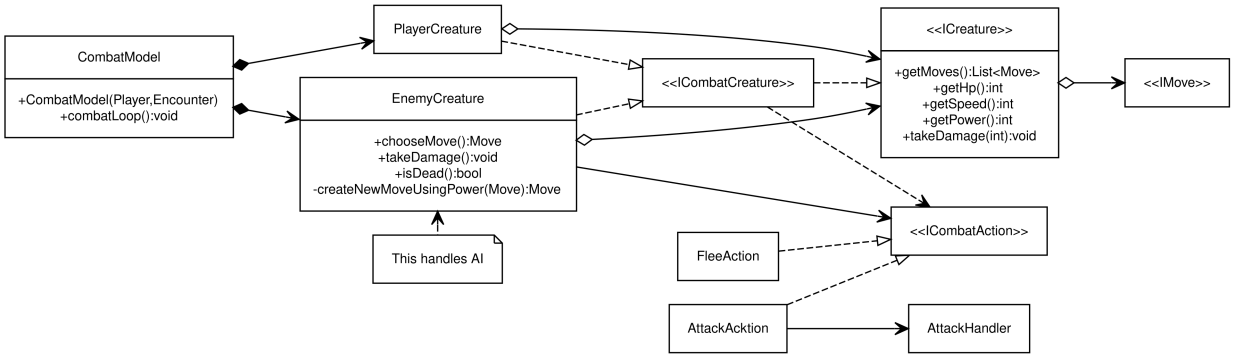
3.2 Controller



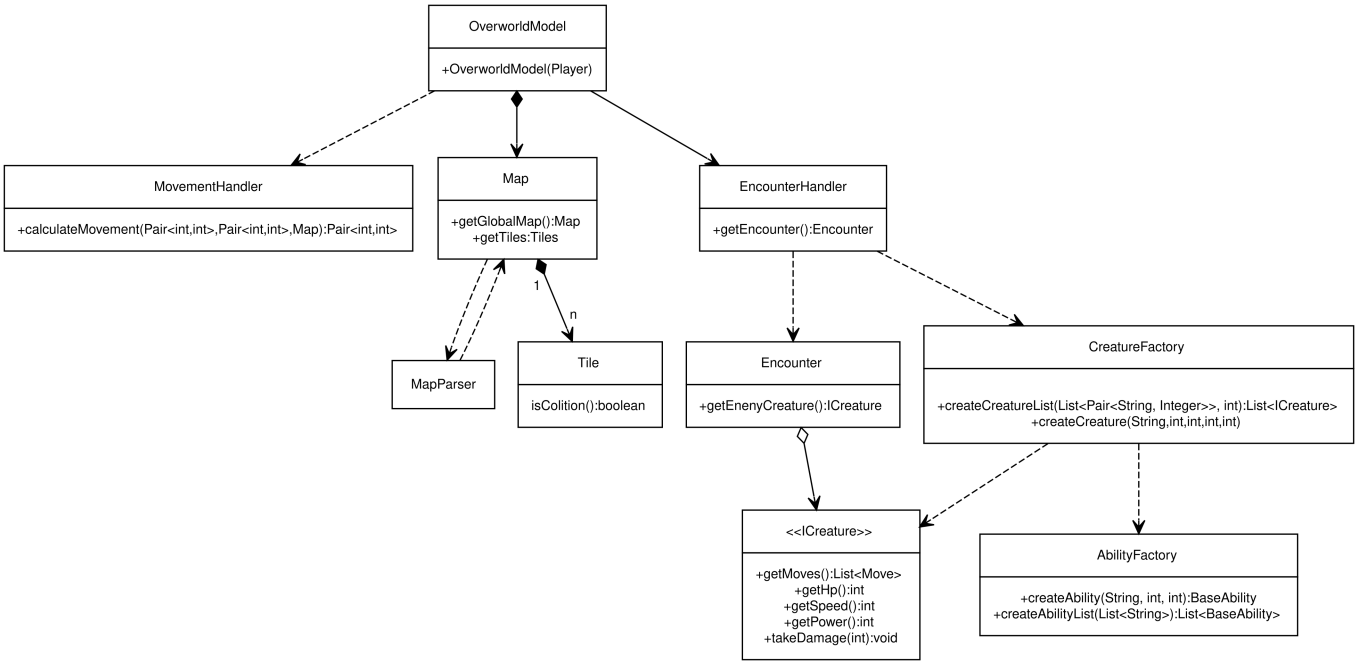
3.3 Model



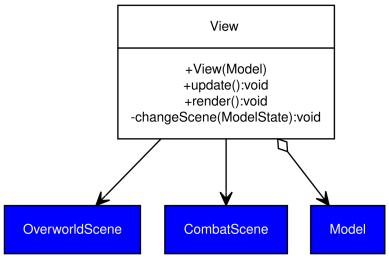
3.4 CombatModel



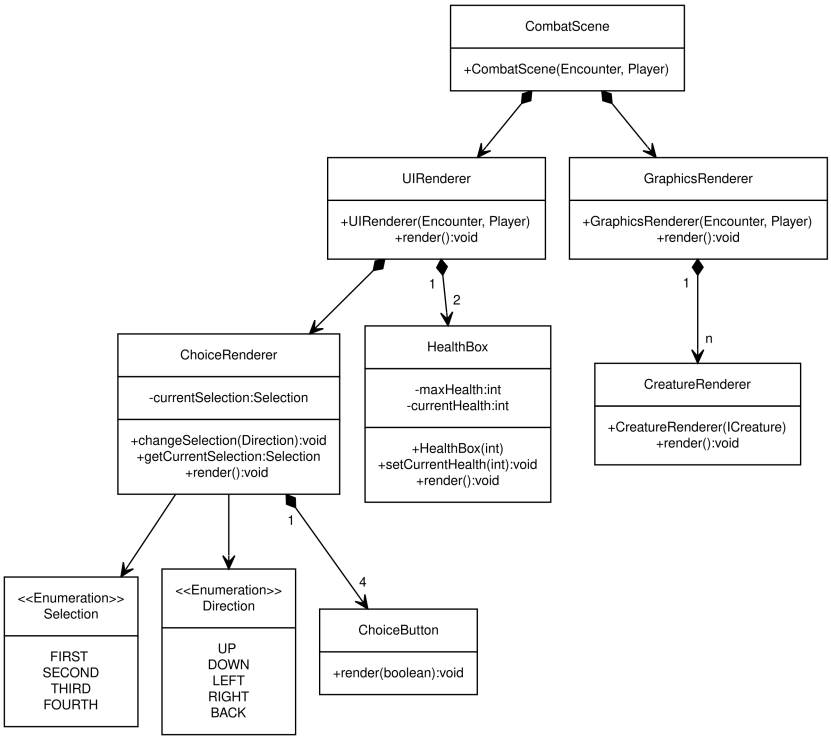
3.5 OverworldModel



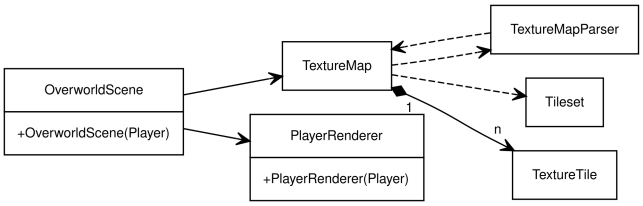
3.6 View



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

The entire project is structured as a classic MVC pattern. The model exists in its own vacuum without any references to ether the view or the controller. The view has a reference to the model and so does the controller. The controller also has a referance to to the view, this is to implement the sprite Batch thet is used to draw thing on the screen. This is so the controller can render its own buttons.

Factory pattern

We use a factory to create monsters...

State pattern (hopefully)

Observer pattern

Even if the code is running continuously and is polling, an observer is implemented in most places where we dont expect a change every frame.

Singleton

Façade pattern

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

References

[1] libGDX, *Libgdx*, 10th Oct. 2022. [Online]. Available: `https://libgdx.com/`.

[2] T. Lindeijer. "Tiled." (10th Oct. 2022), [Online]. Available: `https://www.mapeditor.org/`.