Requirements and Analysis Document

Group 16

13th October 2022

Contents

1 Introduction

1.1 The Background

Feyrune, as is the name of the application, was first conceptualized in the beginning as a Pokémon-style RPG, but with the ability to be easily extended with new functionality to differentiate itself.

1.2 Functionality

Functionality wise the game has two main areas, the overworld and combat. The overworld pertains to functionality such as moving around the map, and travelling to new areas. Combat is the other main area of functionality, and is where the player will be able to fight against enemies. The combat system is turn-based, and will be able to be extended with new functionality to make it more interesting. These two different areas of functionality is glued together by an encounter system, which determines when the player encounters an enemy and initiates combat with them.

1.3 Stakeholders

The stakeholders for the project is the general populace, but mostly the people who are interested in RPGs and games with tactical The game will be free to play, and will only be available on Windows, Linux and Mac OS X.turn-based combat. The game will be free to play and will be available on PC.

1.4 Definitions, acronyms, and abbreviations

Feyrune:	The name of the application
DoD:	Definition of Done
Player:	The user controlled element of the game used to interact with the rest of the world.
Tile:	A square of 16x16 pixels.
Tile set:	An image containing multiple different tiles.
Map:	A collection of tiles in a grid.
Libgdx:	A graphical library and game engine
Tiled:	A program for drawing maps from tile sets

2 Requirements

2.1 User Stories

User story:
Story Identifier: 1
Story Name: Move around the world
DONE
Description:
As an explorer i want to be able to to move around in the world so i can see new places
Functional:

+ Change location of player character.
+ Move player according to input.
+ Animation when walking.
+ Animation when standing still.

User story:
Story Identifier: 2
Story Name: See the enviorment
DONE
Description:
As a player i want to see my environment so I know where I am
Functional:

+ Parse tile sets.
+ Implement map renderer class

User story:
Story Identifier: 3
Story Name: Read inputs
DONE
Description:
As a user I want the game to understand my inputs so I can play the game
Functional:

+ Implement an input listener
+ Make input listener take input from keyboard
+ Connect Input to model

User story:
Story Identifier: 4
Story Name: Parse mapse
DONE
Description:
As a modder, I want to be able to parse and make my own Tiled maps
Functional:

+ Load tile sets from Tiled file
+ Load tile information from Tiled file

User story:
Story Identifier: 5
Story Name: Handle collitions
DONE
Description:
As a user, I want the world to limit my movements so I cant't walk through walls
Functional:

+ Implement collision logic
+ Parse collision from a map file

User story:
Story Identifier: 6
Story Name: Combat Monsters
DONE
Description:
As a user, I want to be able to fight enemys
Functional:

+ Design combat interface
+ Create a combat scene

User story:
Story Identifier: 7
Story Name: Encounter Monsters
DONE
Description:
As a user, I want to be able to fine a monster to attack
Functional:

+ Encounters to trigger a new combat
+ Change scene to view combat

User story:
Story Identifier: 8
Story Name: Attack Monsters
DONE
Description:
As a player i want my monsters to be able to attack, so I can win against other monsters.
Functional:

+ Create an attack handler witch handles attacks
+ Create a way for monsters to take damage
+ Design health interface
+ Implement health interface

User story:

Story Identifier: 9
Story Name: Transition between maps
DONE
Description:
As an explorer i want to be able to transition between different maps, so that i'm not stuck in the same area
Functional:

+ Update Map not to be static and use a map manager instead
+ Do the same thing as above for texture map
+ Create multiple maps
+ Update parser to use map transition data
+ Transition logically
+ Make TextureMapManager observe model MapManager for changing map

User story:

Story Identifier: 10
Story Name: Escape combat
DONE
Description:
As a pacifist I want to be able to run from combat so my monsters don't get hurt
Functional:

+ Run action in combat menu
+ Chance to end combat when running away

User story:
Story Identifier: 11
Story Name: Differnt attacks in combat
Description:
As a player, I want to be able to pick an attack so that I can strategically control my monster during combat
Functional:

+ Design a ”pick an attack”- interface
+ Implement visuals fo said interface
+ Create several (more then one) different attacks to choice from

User story:
Story Identifier: 12
Story Name: Collect multiple monsters
Description:
As a user, I want to keep multiple monster with me so that I can use more then one
Functional:

+ Implement somewhere to store multiple monsters
+ Design an interface for seeing your monsters
+ Pick one that is the current one

User story:
Story Identifier: 13
Story Name: Befriend monsters
Description:
As a user, I want to befriend my enemies so I can have more friends
Functional:

+ Design a befriend interface
+ Add befriending difficulty to monsters
+ Display befriending difficulty in combat
+ Add befriended monster to you monster party

User story:
Story Identifier: 14
Story Name: Run faster
Description:
As a user, I want to be able to run around in the world so I can move where i want faster.
Functional:

+ Optional faster movement
+ New input to trigger faster movement
+ New animation for faster movement

User story:
Story Identifier: 15
Story Name: Ambient sound
Description:
As a user, I want ambient sound so i can be immersed in the game.
Functional:

+ Implement sound player
+ Change sound depending on environment
+ Sound toggle/ volume control

User story:
Story Identifier: 16
Story Name: Combat sound effects
Description:
As a player, I want sound feedback from engaging monsters in combat
Functional:

+ Play sound according to action taken
+ Play combat music to set the mood
+ Play sound depending on hit or miss of attack

User story:
Story Identifier: 17
Story Name: Populate the world
Description:
As a user, I want NPC:s that i can talk to so that i feel less alone
Functional:

+ Non playable characters (NPC) rendered on screen
+ interaction input
+ Text for NPC to say
+ Text shows up on screen
+ The player can continue the text (so more is shown or it disappears)

User story:
Story Identifier: 18
Story Name: Items
Description:
As a user, I want consumables to help me fight monsters so that i can prepare more for combat
Functional:

+ Design consumable interface
+ Create regenerative consumables
+ Create damaging consumables.

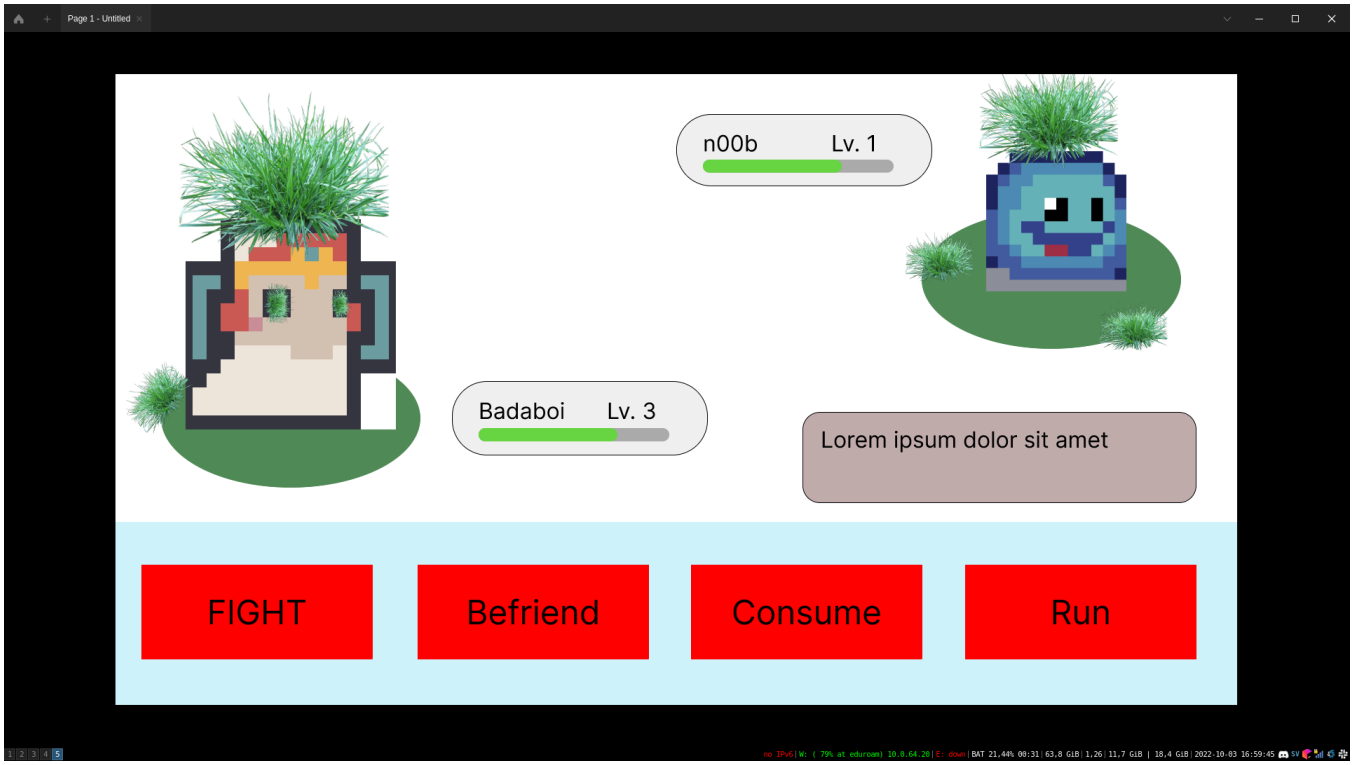
2.2 Definition of Done

The project code was structured in such a way that when a person felt that they were done with a user story, they had to get it reviewed and be questioned about it by at least one other contributor. The purpose of this practice was not only to make sure that the user story was implemented fully, that can be done with tests, but to iron out code smells and to keep the code somewhat coherent and similar-looking throughout the project.

2.3 User interface

Feyrune's UI is split into two major parts, the combat view and the overworld view, where the combat view is used to represent and control encounters between the player and all of Feyrune's different creatures, and the overworld view manages how the player moves on the game map and the exploration aspect of the game.

2.3.1 Combat view



The combat view in Feyrune is inspired by the Pokémon video games, sharing many familiar characteristics. It has four buttons at the bottom of the screen, one for changing the currently used player monster, one for attacking, another for fleeing and a last one for deciding to use an item.

When clicking any of these buttons the menu will change to show the options available for that action. The player monster button will show the player's monsters, and the player can choose which one to use. The attack button will show the attacks available for the currently selected player monster. The item button will show the items available for use. The flee button however, immediately executes the action to flee, as it cannot be done in multiple different ways. At which point the player would be sent back to the overworld.

The combat view shows the player and enemy creatures in the middle of the screen using sprites, and their health using green bars at the top of the screen to help the player see the current status of the battle. If the player creature "dies" during combat, the player gets sent back to the bed in the beginning of the bed.

2.3.2 Overworld view

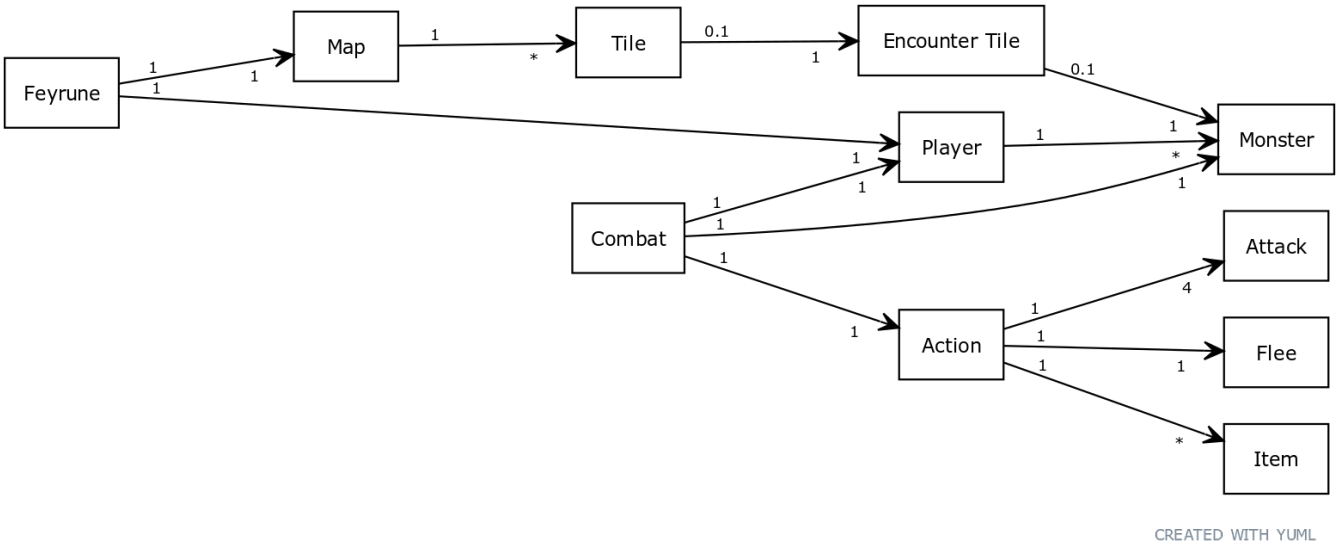
The overworld view in Feyrune is highly inspired from Nintendo games such as "The Legend of Zelda" and "Pokémon Pearl". The view consists of two different layers; a background layer displaying the ground of the map, whether it's made of grass or stone or something completely different, and a foreground layer displaying the player character and any other items visible, such as trees or benches.

The player character is represented by a sprite, and the player can move the character around the map by

using the WASD control scheme. Feyrune includes no form of tooltips or other hints to which buttons to press to do what, as the game is meant to be played on a keyboard, and most player should be familiar with the WASD control scheme.

When a player in the overworld enter a tile containing an encounter, something without a graphical representation as to create a feeling of randomness and suspense, the player will be moved to the combat view, where they can fight the creature they have just encountered.

3 Domain model



3.1 Class responsibilities

- Feyrune

  Feyrune is the name of our game/ application and is the main program.

- Map

  Map is a tiled map that has information about all that the can be seen and explored in the world

- Tile

  A tile is a small part of the map that has information about what that specific tile, ex graphics, collision and encounter

- Encounter Tile

  The Encounter Tile can with chance start a combat with generates a monster and takes information about the players owned monsters

- Player

  The Player has a collection of monsters, some items and a position on the Map. The player is also what you as a user can control.

- Combat

  The combat has one enemy and a players monster that fight until death or someone flees. There are a number of actions that both the enemy monster and the players monster can do

- Action

  An action is what you or the enemy can do in a combat

- Monster

  A monster is a creature with stats and is the cornerstone of this application

- Attack

  All monsters can have one to four different attacks that have different effects on the combat or simply do different amounts of damage

- Flee

  The flee action lets you have a chance to leave the combat without having to fight until death

- Item

  An item is a separate action from an attack that can have different effect on the combat

4 References