Doriqoute

1. Project Overview

Turn Based RPG game inspired by pokemon and RPG-maker Games. With a maximum of 4 characters on the player side to fight with other people.

2. Project Review

RPG-like game. Not that original. The player can walk around to find another npc to fight off in a glorious 4v3 pokemon-like battle. In combat, players can choose between attack and defense. Also each pokemon have ability to choose.

3. Programming Development

3.1 Game Concept

Controllable Player to Explore the game with turn based fighting mechanics.
With interactable NPCs but no story for this version. The dialog will be recycled over and over.

Objective is to have fun fighting and defeat every fightable npcs.

3.2 Object-Oriented Programming Implementation

Game Class:
For the game init, store and load game assets.

Entity Class:
For Player and NPCs which separate them into Player class and Characters Class by inheritance. This class controls all things related to these characters such as movement, animation and dialog.

Sprites Class:
For tile rendering. Using pytmx module to store tiles information and map decorations such as trees and rocks.
Some have specific subclasses for its purpose.

Dialog Class: use to show the dialog and we split for each sentence.

Battle Class: This part is for combat systems that also help in collecting data and calculations.

DataCollector: This class solely used to save csv files.

3.3 Algorithms Involved

For now I don't have big algorithm related code(except sorted lambda for layer rendering). But I plan to make one for the enemy attack pattern.

For attack patterns mostly random due to the fact that enemies also have ability too.

But the data collector and calculation also done by algorithm.

4. Statistical Data (Prop Stats)

4.1 Data Features

In Sprite Class having a hitbox that stores the x and y coordinates for collision.
Entity classes have functions that get the facing directions for tracking animations.
And the Player(Entity) class keeps track of player keyboard input and player object collisions.

3.2 Data Recording Method

I'm using a .tmx and .tms file (Tiled file format) for display maps,
It can store layers of items.

And using pygame.sprite.Group() to store everything and using it to render the game.
Having load_assets to import all needed resources once for game usage.

3.3 Data Analysis Report

I plan to display combatant stats in number.
And save the progression data for later if I can make it in time.

	Usage	**50 Values**	**Used Classes**	**Display**
import_assets	Loading png, tmx, tms, font for game usage.	Import alot of images.	All of Them.	Via rects.
DataCollector	**Collect any value needed**	**Current_health, Current_MP, MP_Usage**	**DataTkin**	**Tkinter, matplotlib, seaborn**

4. Project Timeline

Week	Task
1 (10 March)	Proposal submission / Project initiation
2 (17 March)	Full proposal submission
3 (24 March)	Finalized map mechanics.
4 (31 March)	Finalized Combat system.
5 (7 April)	Visual Improvements.
6 (14 April)	Submission week (Draft)

5. Document version

Version: *1.4*
Date: *31 March 2025*

Date	Name	Description of Revision, Feedback, Comments
27/3	Rattapoom	● You may want to research other turn-based RPG games to complete the Project Review section. ● What exactly are you going to analyze by “importing assets”? ● I suggest something like “Damage taken when fighting enemy x, where x is a fairly hard-to-defeat enemy that spawns quite frequently.”
30/3	Parima	- Project Overview and Game Concept need more detailed explanation.

		- Please edit 3.2 Programming Implementation according to your diagram and complete the Statistical section.

UML Diagram Draft