Pokemon game

1. Project Overview

Provide a brief overview of your project, explaining the game's concept, mechanics, and functionality.

Players take on the role of a Pokémon trainer, venturing through diverse environments to discover, capture, and train Pokemon. As they progress, they will battle AI opponents, including Gym Leaders, to strengthen their Pokemon and unlock evolutionary forms. The ultimate goal is to defeat all Gym Leaders and become the Pokemon Champion.

2. Project Review

Study and then present a relevant existing project, outlining the specific improvements you plan to make. Highlight the key areas where your project enhances or builds upon the original.

https://youtu.be/8OMghdHP-zs?si=0ifqhYZAsc7DSjI8 In this clip it will be 5 games from pygame. I'm interested in Monster Battle, which I will make into a map where I can find Pokemon and gradually level up Pokemon evolution.

3. Programming Development

3.1 Game Concept

Describe the game in detail, including its mechanics and objectives. Highlight the key features or selling points of the game.

The game is a Pokémon-style RPG where players explore different environments, catch Pokémon, and engage in battles to train and evolve their Pokémon.

Objective:

The main goal of the game is to become the ultimate Pokémon Champion by capturing Pokémon, training them, and defeating all Gym *Leaders.*

Players must strategically build a strong team, evolve their Pokémon, and overcome increasingly difficult challenges.
Game Mechanics:

Exploration: Players control a character who walks through various maps (forests, caves, lakes) searching for wild Pokémon.

Catching Pokémon: When encountering wild Pokémon, players can use Pokéballs to catch them.May be 5 pokemon

Battles: Players fight Gym Leaders and AI-controlled opponents using turn-based combat mechanics.

Leveling & Evolution: Pokémon gain experience from battles, allowing them to level up and evolve into stronger forms.

Strategy & Team Building: Players must manage their Pokémon team, selecting the best moves and types for battles.

Game Completion: The final challenge is defeating the Champion in an ultimate showdown after all Gym Leaders have been defeated.

3.2 Object-Oriented Programming Implementation

List and describe at least five classes you will implement in your game. For each class, briefly explain its role, key attributes, and methods. You can also include a UML diagram for better understanding. (Later, this UML will be required as part of the full proposal.)
Might have to have

class Pokemon to create Pokemon
Attributes:
name (str): Name of the Pokémon
level (int): Level of the Pokémon
hp (int): Health points of the Pokémon
Methods:

attack(target): Attacks an opponent Pokémon
level_up(): Increases level and boosts stats

class GymLeaderRepresents .Gym Leaders in the game, each having their own Pokémon team and battle strategy.
Attributes:
name (str): Name of the Gym Leader
pokemon_team (list): List of Pokémon owned by the Gym Leader
Methods:
choose_pokemon(): Selects the first Pokémon for battle
battle(player_pokemon): Manages the battle logic against the player

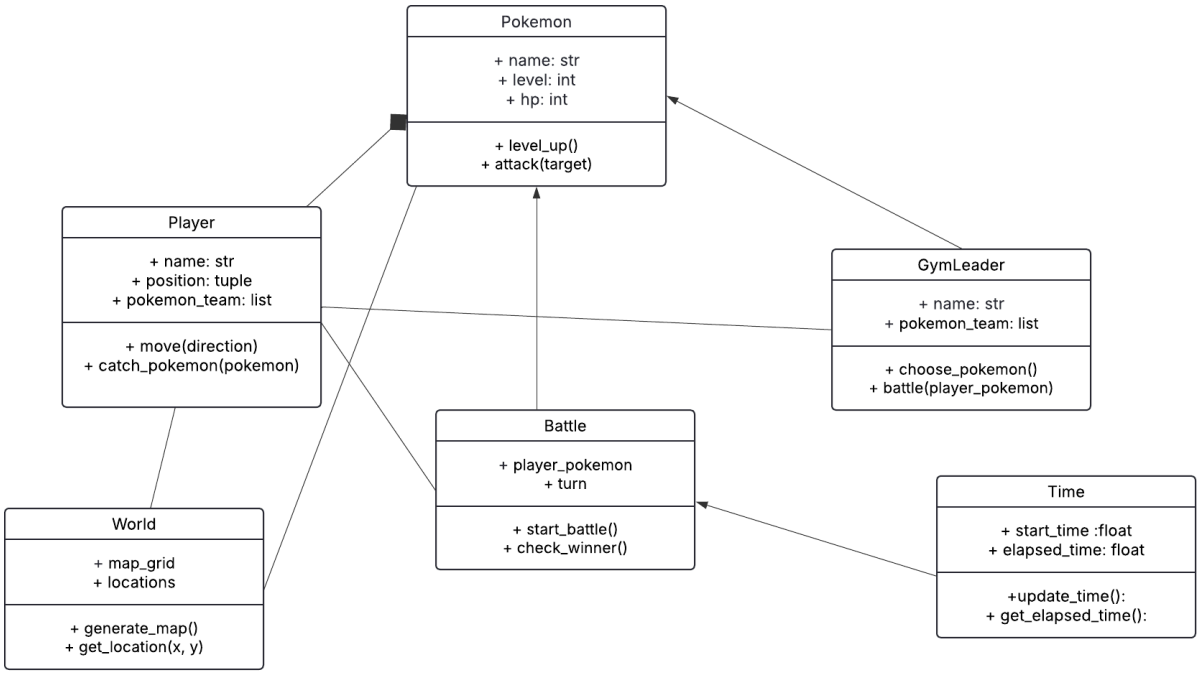
class Battle is for Pokémon that have to fight to reduce their health by using skills.
Attributes:
player_pokemon (Pokemon): Pokémon selected by the player
turn (int): Determines whose turn it is
Methods:
start_battle(): Initiates the battle
check_winner(): Determines the winner after the battle ends

class World is for creating maps that you can play with.
class sound for adding various sound effects
Attributes:
map_grid (list): A 2D array representing the game world
locations (dict): Key locations such as Pokémon Centers, Gyms, and PokéMarts
Methods:
generate_map(): Generates a new world map
get_location(x, y): Retrieves information about a specific area

class Player is used to create a person to walk around looking for various Pokémon.
Attributes:
name (str): The player's name

position (tuple): Player's current coordinates on the map (x, y)
pokemon_team (list): List of Pokémon the player has caught
Methods:
move(direction): Moves the player in a given direction (up, down, left, right)
catch_pokemon(pokemon): Adds a Pokémon to the player's team

class Time is used to measure time
Attributes:
start_time (float): The time when the game starts
elapsed_time (float): The total playtime
Methods:
update_time(): Updates the elapsed time based on real-time progress
get_elapsed_time(): Returns the total time the player has played



3.3 Algorithms Involved

Mention the algorithms used in the game. If your game incorporates techniques such as pathfinding, sorting, AI behavior, rule-based logic, or event-driven mechanics, describe them here.

Use sorted() to sort the names of the Pokemon you've captured in A-Z so that when calling them you can view them by name.

4. Statistical Data (Prop Stats)

4.1 Data Features

Describe at least five features you will track in the game, which may include player behavior or game-related metrics (e.g., character movement, keystrokes, pixels moved, player score, time played, enemies defeated, accuracy, and completion rate). Each feature should have at least 50 rows of data. (For example, in football, a player's performance can be analyzed by collecting and analyzing statistical data.)

1.Player Movement - Distance traveled, Number of steps taken, Which areas are explored the most?
2.Pokemon Caught - Name of the Pokemon caught, Location where the Pokemon was found
3.Battle Performance - Number of battles fought, Most used attack moves
4.Pokemon Evolution - Pokemon that evolved, Which Pokemon evolve the most frequently?
5.Playtime Statistics - Time spent per session, Number of times the player quits the game

4.2 Data Recording Method

Explain how the game will store statistical data. Will it be saved in a database, CSV file, or another format?

Json - Do like Professor Piya used to teach about uploading data to Json.

4.3 Data Analysis Report

Outline how you will analyze the recorded data. What statistical measures will you use? How will the analysis be presented (e.g., graphs, tables, charts)?

The recorded JSON data will be analyzed using statistical measures such as

1.Averages - *To determine the average distance walked, playtime.*
*2.*Frequency Counts - To see the most commonly caught Pokemon, most used moves

Use - Tables for numerical summaries such as playing time Number of Pokemon available
Use - Graphs & Charts for trends & comparisons such as bar Graph for Pokemon caught per location, Line Graph for player progress over time

4.4 specify more about data collected

Feature	Why it is good to have this data? What can it be used for?	How will you obtain 50 values of this feature data?	Which variable (and which class will you collect this from?)	How will you display this feature data (via summarization statistics or via graph)?
Player Movement	Helps track exploration behavior useful to analyze which areas are most visited and to improve map design	Record player's coordinates every few steps during gameplay	position from the Player class	Line graph or heatmap to show movement patterns
Pokemon Caught	Useful to see which Pokémon are most frequently caught and where; helps	Log each time a Pokémon is successfully caught	Inside catch_pokemon(pokemon) method of Player class	Bar chart showing count per Pokémon or per location

	with game balancing			
Battle Performance	Shows player's engagement and skill useful to evaluate battle mechanics and move usage	Log data at the start and end of each battle	start_battle() and check_winner() in Battle class	Summary statistics: win/loss ratio, most used moves
Pokémon Evolution Playtime Statistics	Tracks progress of the player's team can be used to balance evolution difficulty	Record each time a Pokémon levels up and evolves	level and level_up() method in Pokemon class	Pie chart showing number of evolutions by type
Playtime Statistics	Measures player retention and game session duration helps evaluate game	Record session start and end times for every playthrough	start_time, elapsed_time from Time class	Average playtime shown as summary statistics; bar graph for total hours played

	engagement			
Area Visited	Helps identify which map locations are most/least popular improves world design	Record area name each time player enters a new zone	get_location(x, y) from World class	Bar chart for most visited locations

4.5 Displaying Data via Tkinter

The statistical data stored in JSON will be loaded and displayed using Tkinter GUI. The interface will include summary statistics (e.g., average playtime, number of battles) using Labels and Tables (Treeview), and visualized graphs (e.g., bar chart for Pokémon caught per location, line graph for player progress) via Matplotlib embedded in Tkinter using FigureCanvasTkAgg. This allows users to easily analyze game data in real-time.

4.6 Clarify and give more information about the earlier question: How will you display each feature (via summarization statistics or via graph)? In this proposal #4

1.1 (A) Feature to Present in Table:

-Feature: Playtime Statistics

(B) Statistical Values to Present:

-Average playtime per session

-Maximum and minimum playtime

-Standard deviation of playtime

1.2 Graph Plan Table:

Feature Name	**Graph Objective**	**Graph Type**	**X-axis**	**Y-axis**
1. Player Movement	Show how far players explore in the game	Histogram (Distribution)	Distance walked (steps/pixels)	Frequency of players
2. Pokémon Caught	Compare where most Pokémon were caught	Pie Chart (Proportion)	Location	% of Pokémon caught
3. Pokémon Evolution	Track evolution progress over time	Line Graph (Time-series)	Game time (minutes/hours)	Number of Pokémon evolved
4. Battle Performance	Show which moves are used most frequently in battle	Bar Graph (Relation)	Move name	Frequency used

5. Game Sessions	See number of sessions per day	Stacked Bar Graph (Time)	Date	Number of sessions per day

4.7 Submit your planning how you will approach and complete your project.

-26 March-2 April

• Finalize Proposal #4 (features, stats, graphs)

• Draft layout for graphs and tables

• Begin game dev: movement, map exploration, Pokémon catching

-3 April-9 April

• Implement JSON logging for player movement and playtime

• Add evolution system and basic time tracking

• First round of gameplay testing for data collection

-10 April-16 April

• Analyze first round of collected data

• Begin graph/table visual implementation using mock data

50% Milestone:

• Core game mechanics functional

• Data logging started

• Drafted 3 graph visuals + 1 table

-17 April-23 April

• Develop battle system and attack tracking

• Continue gameplay testing with real data

75% Milestone:

• Real data used in graphs

• Graph implementation complete

• Evolution & battle logs working

-24 April-11 May (Note that 28 April-9 May are final exam weeks)

• Polish graphs/tables based on full game data

• Debug and balance gameplay

• Prepare final report, presentation, and gameplay showcase

100% Milestone:

• All graphs and tables finalized

• Game fully working with data

• Ready for submission

4. Project Timeline

Week	Task
1 (10 March)	Proposal submission / Project initiation
2 (17 March)	Full proposal submission
3 (24 March)	Game system development (Basic mechanics, map

	exploration, Pokemon capture)
4 (31 March)	Battle system (Turn-based battle, Gym Leader, Pokemon evolution logic)
5 (7 April)	Game testing & debugging (Fixing mechanics, balancing Pokemon stats, optimizing performance)
6 (14 April)	Game testing & debugging (Fixing mechanics, balancing Pokemon stats, optimizing performance) Submission week (Draft)

5. Document version

Version: *1.0*
Date: *4 March 2025*

Date	Name	Description of Revision, Feedback, Comments
13/3	Rattapoom	• Evolution class may be unnecessary. • Some comments on formatting.
16/3	Parima	Game concept needs more details.
30/3	Parima	UML diagram needs some revision.