CS3330 Final Project

Overview

This Java game was made as a final program for a Java Object Oriented Programming course I took, however, I also made this game for fun. I spent a ton of time on and I'm proud of what I have. Some of the code may seem strange due to the limitations of JavaFX and the FXMLLoader, but I did my best with what I knew. I've tested the game thoroughly and everything works as intended. In the future I may add the ability to swap the position of Java Monsters and add types to each monster. I was thinking of having four types, water, earth, fire, and air.

Purpose

The purpose of this game was to push my knowledge of Java and object-oriented principles to the limits. I learned a lot while creating this game and I'm glad that I created it. For those playing the game, the purpose is to beat the final trainer and maybe even attempt to beat my world record.

Required Information

This game is a heavily inspired from the Pokémon games. I used to play them for hours and hours when I was a kid, so they hold a special place in my heart. When you start, you will notice a path leading to the left, right, and upwards. To the left is the shop where you can buy fruit and JavaBalls. To the right of the is the Health Center where you can heal your Java Monsters. Heading upwards leads to the battles. You can encounter a wild Java Monster by walking through the grass or battle a trainer by following the trail upwards again.

How to Play

You move your character with w,a,s,d or arrow keys. You start with one Java Monster, Crookodile, and your primary objective is to beat the sixth and final trainer, Strong Man Sam. To beat Sam, you will have to capture Java Monsters and train them to a high level. The Trainers get progressively harder leading up to Sam so don't be afraid to battle them. The only way to obtain new Java Monsters is to buy JavaBalls from the Shop and use them on wild Java Monster. You cannot catch trainer's Java Monsters. You can have a maximum of six Java Monsters. To train you Java Monsters you must battle and defeat enemy monsters. When you battle enough enemy monsters your Java Monster will level up. Leveling up increases your monsters attack and defense. I've included save files in the project if you want to load those. One starts on the final battle with Sam.

How to Battle

The battles with wild monsters or with trainers have the same controls. You continue the battle by clicking in the TextArea. The battles are turn based and the starting side is random. When it's your turn, you can fight, heal, or catch. The fight button attacks the enemy dealing damage based on your Java Monster's level and the enemy's level. Pressing the "bag" button opens a menu to use fruit or to use a JavaBall. Fruit can only be used if your Java Monster's health isn't full and you have at least one JavaBall. JavaBalls attempt to catch the enemy monster, but once again, it cannot be a trainer's monster. The battle ends whenever the enemy monster is caught, or you win or lose the battle. That's all you need to know. Good luck!

Required Elements

Classes

1. Tmh9gvFinalProject
2. FightSuper
3. FirstPlayer (Enumeration)
4. JavaMonster
5. MainController (Singleton)
6. ShopController
7. FirstController
8. GrasslandController
9. HealController

Subclasses

1. BattleController extends FightSuper
2. EncounterController extends FightSuper
3. Model extends JSONModel

Abstract Class

1. JSONModel
2. FightSuper

Interface

1. CharacterMove

Collection Class

- ArrayList<String> in Model on lines 22, 36, 73-84

Exception Handling

- BattleController
  - There is a try catch() in for setting enemyTrainerImage (line 100-109).
  - There is a try catch() for creating the new Thread since it's essential to the program running correctly (line 121).
- EncounterController
  - There is a try catch() like one in BattleController that creates the thread (line104).
- FightSuper
  - There is a try catch() to pause the thread and wait for user input (line 48).
  - setEnemyMonsterImage() and setJavaMonsterImage() are contained in try catch() and used by BattleController and EncounterController to attempt to create and set an Image to an ImageView.
- JSONModel
  - There is a try catch() to parse a JSON String into a JSONObject (line 148).
  - There is multiple try catch() to create a new JavaMonster for each member of the party that is coming from the JSON String and add them to the party array (lines 236, 281, 325, 369, 413, and 457)
- MainController
  - There is a try catch() to include a Hyperlink to a video of my game (line 87).
  - There is another try catch() for a FileReader and BufferedReader in the handleOpen() method that was taken from lecture (line 155).

Model

- JSONModel
    - JSONModel contains all the data required to save and open JSON save files. This includes the next trainer to battle, the number of fruit and JavaBalls the player has, money, and the selected Java Monster. It also contains the methods to create the JSON String and to initialize the model from a JSON String based on the lecture.
- Model
    - Model is a subclass of JSONModel and contains all the other data including the enemy's party, an ArrayList of names of trainers, and a bunch of method to access the data from this class and its superclass's members. This includes handling the data of battles and buying items.

Multiple Scenes (in Main Controller)

- I have a bunch of scenes and every one of them is dynamic. Either the character moves dynamically, or the battle is dynamic.
- FirstScene (line 45)
    - This is the Scene you start on that is between the shop and the Health Center.
- ShopScene (line 46)
    - This is the Scene of the Shop
- HealScene (line 48)
    - This is the Scene of the Health Center
- GrasslandScene (line 55)
    - This is the scene that with the path to the trainer battles and the random wild encounters.
- EncounterScene (line 56)
    - This is the Scene with random encounter in the grass.
- BattleScene (line 47)
    - This is the Scene where you battle enemy trainers.

About Section

- The about section can be accessed from any Scene by using the menu at the top of the application. This section also pops up at the very beginning of the application to inform the player about the game and how to play (MainController line 75).

Save/Load Data

- Saving data is also accessed through the menu, however, the player cannot save or load while in battle because that will create unforeseen consequences and offers a way to cheat. The method to save (handleSave() MainController line 185) and load (handleOpen() MainController line 139). Both functions use the JSONModel functions (toJsonString() on line 32 and initFromJsonString() on line 140) to access and update the data.
- Saving and Loading will update the next trainer fight, number of fruit and JavaBalls, money, the selected Java Monster, and the player's party.