**CSc 335 Spring 2018 Final Project: Game Logic and Design Overview**

Team Members: Alan Deitz, Paria Khamsehzadeh, Samantha Felzien, Rajeev Ram

Game Overview:

Our final project is a reconstruction of one of the internal, non-main quests in Nintendo’s Pokémon: the Safari Zone. The game includes a Pokémon town with multiple views and in-game interactions. From this town, the player can decide to initiate the Safari Zone quest, which also has multiple views and in-game interactions. The bulk of the design, including win-loss conditions, is detailed below.

Game Design:

1. **Starting Game**

The game is booted up by executing the Controller.java class. When this JavaFX application is run, a start-up menu appears. The user has the option between choosing a Girl Trainer or a Boy Trainer to represent them during the game. They also have the opportunity to enter a custom name. Should they decide not to, the default names are Lyra and Ethan. The Controller then places the user in the building from which user gains keyboard control of his or her character.

1. **The Views**

All of the views described below inherit from an abstract class called GeneralView.java.

Building View – This is the view from which the user starts his or her adventure. The building contains some NPCs and decorations. The user can walk out of the building into the town, and back from the town into the building.

Town View – This one of the main views of the game; it contains NPCs, a Trainer to battle, and other hidden features that the user can learn about in-game. From this view, the Trainer can talk to the Safari Zone agent to deploy the Safari Zone quest.

Safari View – This is one of the main views in the game; the Safari Zone quest takes place entirely within this view. Within the Safari Zone quest there are various win loss conditions that are detailed below.

Cave View – This is one of the main views in the game; the is a second portion within the Safari View.

1. **Trainer Battle**

When the player is in the town, they can initiate a trainer battle with the NPC in the southwest corner of the map once per day. If the player wins the battle, he or she receives $1000. If the player loses the battle, nothing happens. If the user cannot pay out, he or she blacks out and returns to the previous save state in the game.

1. **Starting Quest**

Starting Quest – At any point from the town, the player can talk to the Safari Agent NPC in the north. If the player decides to start the quest, he or she must pay $500. Subsequently, the Safari Agent will explain the rules of Safari Quest and provide the user with Safari Balls, rocks, and bait.

Win Conditions – The player catches all ten types of Pokémon in the Safari Zone. The player catches two each of five types of Pokémon. The player catches five each of two types of Pokémon. Upon winning, the player receives $3000 and gets to keep one of the Pokémon he or she caught during the quest.

Lose Conditions – The player does not meet one of the win conditions before one of the following happens: using all 30 safari balls, taking 500 steps, or spending 45 minutes in the Safari Zone. Upon losing, the player exits the Safari Zone and is rewarded nothing.

Ending Quest – The player can play the quest up to three times per day. While in the quest, the player can exit the Safari Zone at any time from the menu, which triggers a loss condition. Both a win condition and loss condition increment the number of daily quests performed by one. The player is returned to the town.

1. **Safari Battle**

The only way to catch a Pokémon in the Safari Zone is by encountering a wild Pokémon, which triggers a battle.

Ground Battle – These are initiated at random when the player walks in the tall grass. The Pokémon that can be encountered on the ground are (from least to most likely): Jigglypuff, Pidgeotto, Gloom, Pikachu.

Ocean Battle – These are initiated when the trainer uses a fishing rod in the water. The Pokémon that can be encountered in the water are (each with equal probability): Tentacool, Seaking, Staryu.

Cave Battle – These are initiated at random when the player walks around in the cave. The Pokémon that can be encountered on the ground are (from least to most likely): Zubat, Golbat, Growlithe.

Battle Mechanics – The player can throw a rock to increase a Pokémon’s catch probability and run probability. Rocks reduce the hit points of the Pokémon by a set amount. The player can throw bait to decrease the Pokémon’s catch probability and run probability. The player can throw a ball to try to catch the Pokémon. The Pokémon always runs away after 10 turns.

Ending Battle – The battle ends when one of the following happens: the Pokémon is caught, the Pokémon runs out of hit points, the Pokémon runs away, the Trainer runs away. Any of these can happen on any turn given the proper pre-conditions.

1. **The Menus**

There is always a main menu on the left side of the screen which contains the following options: Inventory, Pokémon, Save, Quit, and Close. The Inventory submenu allows a player to scroll through his or her collected items and use them as appropriate. The Pokémon submenu allows the player to scroll through his or her caught Pokémon and use them as appropriate.

The other buttons are rather self-explanatory.

1. **Other Features**

Will complete soon. There is a Merchant whole sells Potions, X-Attack, and X-Defense.

Game Logic: Our project work is split into six main categories: Controller, View, Model, Tests, Documentation, and Media.

1. **Model**

This package contains three enumerated classes: Direction, Location, and Terrain. Direction is used to identify in what direction the Trainer is facing on the map for still posing and animation purposes. Location is used to identify in which view of the map the trainer is, to keep track of spawning and exiting positions. Terrain is used to keep track of which spaces a Trainer can access and what images appear as tiles on those spaces.

Pokémon – This subpackage contains an abstract class and ten children to represent the Pokémon in the game.

Items – This subpackage contains an abstract class and six children to represent the items in the game.

Trainer – This subpackage contains an abstract class, the Boy and Girl, and an enemy Trainer that inherits from the NPC interface

NPC – This contains the main (abstract) interface and ten children of varying complexity

Observer – This contains two observable lists classes that are instantiated to represent inventory in the game.

1. **View**

General – This subpackage contains the introduction screen(s), and all the non-battle screens throughout the game. The parent view is a border pane that is subclassed as general view for gameplay.

Battle – This subpackage contains an abstract class and five children to support the different battle styles, backgrounds, and animations

1. **Controller**

Controller – This mediates between the model and the views. It is almost entirely static and supports persistence, i.e., saving the game

MoveHandler – This is event handler whose functionality is adapted out to a different class; mediates communication via static reference to the Controller

SoundPlayer – This handles the playing of background music, and sounds triggers by in-game events; it utilizes a separate thread to prevent interference.

1. **Media**

All sprites and sprite sheets used in the game are located in the images folder. All sounds and noises are located in the sounds folder.

1. **Tests**

Contains a comprehensive testing suite for all classes in the model. A JavaFXThreadingRule is used to mimic a JavaFX environment for view-based model tests.

1. **Documentation**

Contains this document as well as other reference documents pertaining to the project.