



Danemon



Daniel Jun
CS 5004 Final Project



Goals

- Create Pokemon-like battling game
- Have an interactive Java Swing GUI to display Pokemon and narration for progressing the battle
- Use MVC design architecture
 - Model has all of the game logic
 - View creates the Java Swing display based on inputs
 - Controller listens to actions from the JavaSwing GUI, implements changes to the Model, and updates the View

The Game

- The player will face against a computer player
- Both players have 4 randomly selected Danemon
 - Each Danemon has a unique set of stats (health, attack, defense, speed) and attack skills.
 - Attack skills can do damage, buff / debuff stats of the attacking or attacked Danemon
- Attacks are performed once per “turn”. Both players’ Danemon will attack, unless it faints before it can attack.
 - Attack order is dependent on the Speed of the battling Danemon.
- Text display will narrate what occurred in the Danemon battle
- Once one Player has caused all 4 of the other player’s Danemons faint, they win.



DEMO

Tools + Techniques

- Inheritance
- Final data types
- Error checking
- Dynamic Dispatch
- Enums
- Equality
- Higher Order Functions
- Model - View - Controller Design
- Java Swing
 - JFrame, JPanels, JButtons, ActionListeners
 - Found that JFrame `repainting()` will occur at some later, optimized point in the thread. It may also combine repainting events such that it effectively occurs less often than otherwise planned.

Learning

- With MVC, it is important that action Listeners are added after the View object is created
- JFrame has a repaint() method, but it is optimized to run at the end of the operating thread. To run it multiple times with delay between calls requires running multiple threads
- There may be a need to “lock” the GUI when processing a request, else further button presses will trigger action listeners to run process again
- Introducing “randomness” in a model can make testing tricky. Using a seed to define the randomness can help with testing

Future

- Expand allowable range of Pokemon and Skills
 - More Pokemon and Skills
 - Add abilities to Pokemon
 - Skills can perform more types of effects: healing, status ailments, recoil
- User / Enemy player Danemon swapping
- Item usage
- “Smart” enemy that performs decisions based on max damage ordered decision tree - different difficulty scales have
- Add Danemon battle animations
- Refactoring Controller to allow multiple operating threads