Alec Kretch

Morgan Heck

Jamie Hosmer

Hari Mainali

Design Doc

Model View Controller – architecture of game

* We used MVC as the architectural design pattern of our game.
* The Model holds all the data for the pokemon, the player, and the battle
* The view represents the data in the model, and also contains the tile based map
* The controller launches the application

Template Method – tiles

* We used the template method for the tiles used in the tile-map.
* The Tile class is abstract, the subclasses redefine the Tile based on the filename of the image passed into the constructor

Factory – pokemon in model

* We used the Factory design pattern for the pokemon in the model
* The Creature class was abstract, and all subclasses extending it were instantiated based on the parameters passed into the constructor

Singleton – player

* The player class in the model uses the Singleton design pattern.
* The Player class is one instance and accessed globally in the application

Composite Pattern – FrogRunner

* The FrogRunner is where the panel with the animation of the pokemon character is painted over the tile map, using JLayeredPanes
* The FrogRunner is a JFrame that constructs the panel with the animated sprite, and the Map is a JPanel that is declared in the class

Builder Pattern (Discussed)

* We discussed using the builder pattern to create the Map piece by piece upon the user’s different selections (player, map, and desired win condition). We ended up deciding, however, it would be more efficient to use different constructors for each of these different views.