Beautiful Corridor: Battle System

By Brian Baron and Gabe Pereyra

The Battle System

The Battle System is the container class that will be used for the main game logic. This includes allowing for turns, as well as checking if the end condition has been met. The end condition is satisfied when either of the Pokemon reach 0 health, making the other the winner.

Pokemon Base Class

Within the class there are 2 Pokemon pointers for the Pokemon active in the battle. The Pokemon class is a base class that contains functions and values essential to all Pokemon to be implemented: Health, Attack, Defense, Special Attack, Special Defense, and Speed. There is a member that tracks the current health, which will trigger the end state upon reaching 0. Each Pokemon has a name, a list of 4 Moves available to them, and are assigned 2 Types.

During damage calculation, if the attacking Pokemon's Type matches the move, it gains a Same Type Attack Bonus (STAB), which multiplies the regular output by 1.5. The rest of the calculation is determined by the attacking Pokemon's offensive stats, the power of the Move, the defending Pokemon's defensive stats, and the Type weakness/resistance coefficient (which is calculated by a table containing the Type weaknesses and resistances). The output is also multiplied by a random value from .85 to 1, to allow for varying outputs within a rather controlled average.

Types

There are 16 possible Types implemented, and an individual Pokemon can have 1 or 2 types (known as MultiTyping). Each type corresponds with a table of weaknesses and resistances, stored in a 2D array. This table is used to get the coefficient for weaknesses used in the damage calculation, by checking the indexes [attacking move type][defending pokemon type].

Move Class

Each move contains a Type, Power/strength, Accuracy, PP (Power Points/usage amount), and a name to correspond with it. Any move with a negative value is a healing move, which heals the Pokemon using the move for ½ of the max health.

Translating to Unreal

Pokemon Base Class

The Pokemon base class will be merged with Actors, and be used for loading models. In future implementations of the final project, we plan to have the ability to switch between other Pokemon.

Move Class

The Move and Type classes will be merged with visual feedback, allowing for representation of attacks and damage inflicted.

Types

Types will keep the same functionality as in a plain C++ project, because they are used solely in data, and not in the visual representation of the game.