COM S 327, Spring 2022
Programming Project 1.08
Pokémon Battles

The most important part of this update is to add the Pokémon battle minigame.

If you are not familiar with Pokémon battles (or if you are), here's a YouTube video of an example battle: https://www.youtube.com/watch?v=Jpdy9pZyGH4. Please don't ask me why this video has over 2.1 million views. I don't know. Perhaps there are a lot of professors out there having their students write Pokémon-inspired games?

In order to battle, a trainer must have at least one Pokémon. The PC will be given the choice of three randomly generated level 1 Pokémon when the game starts. All generated trainers will be given at least one random Pokémon. Assuming the trainer has n Pokémon, there is a 60% probability that the trainer will get an (n+1)th Pokémon, up to a maximum of 6 Pokémon. The generated trainers' Pokémon will be generated with levels and movesets as per the Pokémon generation rules for their map.

Pokémon battles are characterized by a turn-based interaction between two trainers or between a trainer (the PC) and a wild Pokémon (the ones which appear in the tall grass). Battles with a wild Pokémon end with the capture of the wild Pokémon, with either the wild Pokémon or the trainer fleeing the battle, or with either the wild Pokémon or all of the trainer's Pokémon getting knocked out. Trainer battles end when all of one trainer's Pokémon are knocked out. Pokémon are not available for capture in trainer battles.

A Pokémon is knocked out when it's hitpoints fall to zero. A knocked out Pokémon is unavailable for battle until it is revived.

As with other areas of our game, we'll be simplifying Pokémon game mechanics in implementing battling. The most important simplifications we'll make are that we won't be implementing status effects (there are simply too many of them to make it a reasonable assignment; of course, you're welcome to do it if you want to) and we won't implement Pokémon abilities.

Each party takes turns in a Pokémon battle, with each taking an action each turn. If a trainer's chosen action is anything other than a Pokémon move, this receives maximum priority. If both moves in a turn are Pokémon moves, then the Pokémon whose move has the higher priority (moves.priority) goes first; if both moves have the same priority, then the Pokémon with the highest speed (level adjusted) goes first, and if that is also equal, then choose one at random.. If a Pokémon is knocked out or captured before using its move, its turn is forfeit.

Pokémon moves can miss. The chance of a Pokémon evading a move is modified by status effects, which we will ignore. This simplifies accuracy to be a simple percentage given by moves.accuracy. If rand() mod 100 $<$ moves.accuracy, the move hits; otherwise it misses.

Once a move hits, damage is given by the following formula:

$$\left(\frac{\left(\frac{2\times Level}{5}+2\right)\times Power\times\frac{Attack}{Defense}}{50}+2\right)\times Critical\times random\times STAB\times Type$$

where $Level$ is the level of the attacking Pokémon, $Power$ is the power of the move (moves.power); $Attack$ is the attacking Pokémon's level-adjusted attack stat; $Defense$ is the defending Pokémon's level-adjusted defense stat; $Critical$ is 1.5 if the attack is a critical hit, otherwise 1; $random$ is a uniformly distributed random number in $[85, 100]$; $STAB$ is the same type attack bonus of 1.5 if the move's type matches the attacking Pokémon's type, else 1; and $Type$ is the type effectiveness modifier of 0, .25, .5, 1, 2, or 4, depending on both the move type and the target Pokémon's type (implementation saved for 1.09; for now, you may use 1). The move's damage is decremented from the defending Pokémon's HP, and the defender is knocked out if its HP reaches zero.

In order to calculate STAB, you'll need the attacking Pokémon's type. Parse in `pokemon_types.csv` for pokemon type information. You'll find that many pokemon have two types (are dual typed). A move gets STAB if any of a Pokémon's types matches the move's type.

An attack earns a critical hit if it hits and a random number in the range $[0, 255]$ is less than a threshold value given by $\lfloor \frac{BaseSpeed}{2} \rfloor$ (floor is just integer division!) where $BaseSpeed$ is the base value (not level-adjusted) of the attacking Pokémon's speed stat.

During battle, the PC has the following options, which should all be implemented in the battle placeholder(s) that were added the previous two weeks:

1. Fight - The active Pokémon uses one of its moves; player chooses the move if PC, otherwise the AI chooses (see below).

2. Bag - Use an item from the trainer's inventory. We will implement revives, potions, and Pokéballs (more below)

3. Run - Attempt to flee the battle

4. Pokémon - Switch to a different active Pokémon. A trainer may have up to 6 active Pokémon.

The fight mechanic is described above.

The bag mechanic consumes an item from the trainer's inventory. The trainer must be carrying the item in order to use it. A revive will revive a knocked-out Pokémon and restore it to half its max HP. A potion will restore up to 20 HP, never taking the Pokémon above its max HP value. Pokéballs may only be used in battles with wild Pokémon and will attempt to capture them.

The PC may only run from wild Pokémon. Battles with trainers must go until all of one trainer's Pokémon are knocked out. Attempts to run may fail. Escape is calculated by

$$Odds_{Escape} = \left\lfloor \frac{Speed_{Trainer} \times 32}{\lfloor \frac{Speed_{Wild}}{4} \rfloor mod 256} \right\rfloor + 30 \times Attempts$$

where $Speed_{Trainer}$ is the level-adjusted speed to the trainer's active Pokémon; $Speed_{Wild}$ is the level-adjusted speed to the wild Pokémon; and $Attempts$ is the number of times the trainer has attempted to escape this battle (including the current attempt). $Odds_{Escape}$ is out of 256, thus if rand() mod 256 $<$ $Odds_{Escape}$, the trainer escapes.

Only one Pokémon may be active in a battle at any time. Switching Pokémon will change the active Pokémon. A Pokémon may only be switched in if it is not knocked out. A trainer may carry up to 6 Pokémon at a time. The first Pokémon (e.g., in the array) always starts the Pokémon battle, and switching Pokémon uses the turn.

We'll implement Pokémarts and Pokémon centers in 1.09, so for now, simply start the PC with a small number of potions, revives, and Pokéballs. These items are usable in battle (Pokéballs are only available in battles with wild Pokémon). Details of the catch mechanics will also be in 1.09. For now, if the PC has fewer than 6 Pokémon, every attempt to catch succeeds, but if the PC has 6 Pokémon, every attempt to catch will cause the wild Pokémon to escape from the Pokémon and flee.

The NPC side of Pokémon battles needs to be "AI" driven. I put this in quotes because you may make your AI as simple or as complicated as you like. It would be sufficient to choose a random move from the first Pokémon, switching Pokémon only when the current Pokémon is knocked out.

Pokémon are not restored to full health automatically after battle. In 1.09, they can be restored at Pokémon centers for free, but for now the only want to restore them is to use revives and potions. Add the 'b' command to allow bag access outside of battle so that your trainer can heal Pokémon at any time.