

TYE HADFIELD

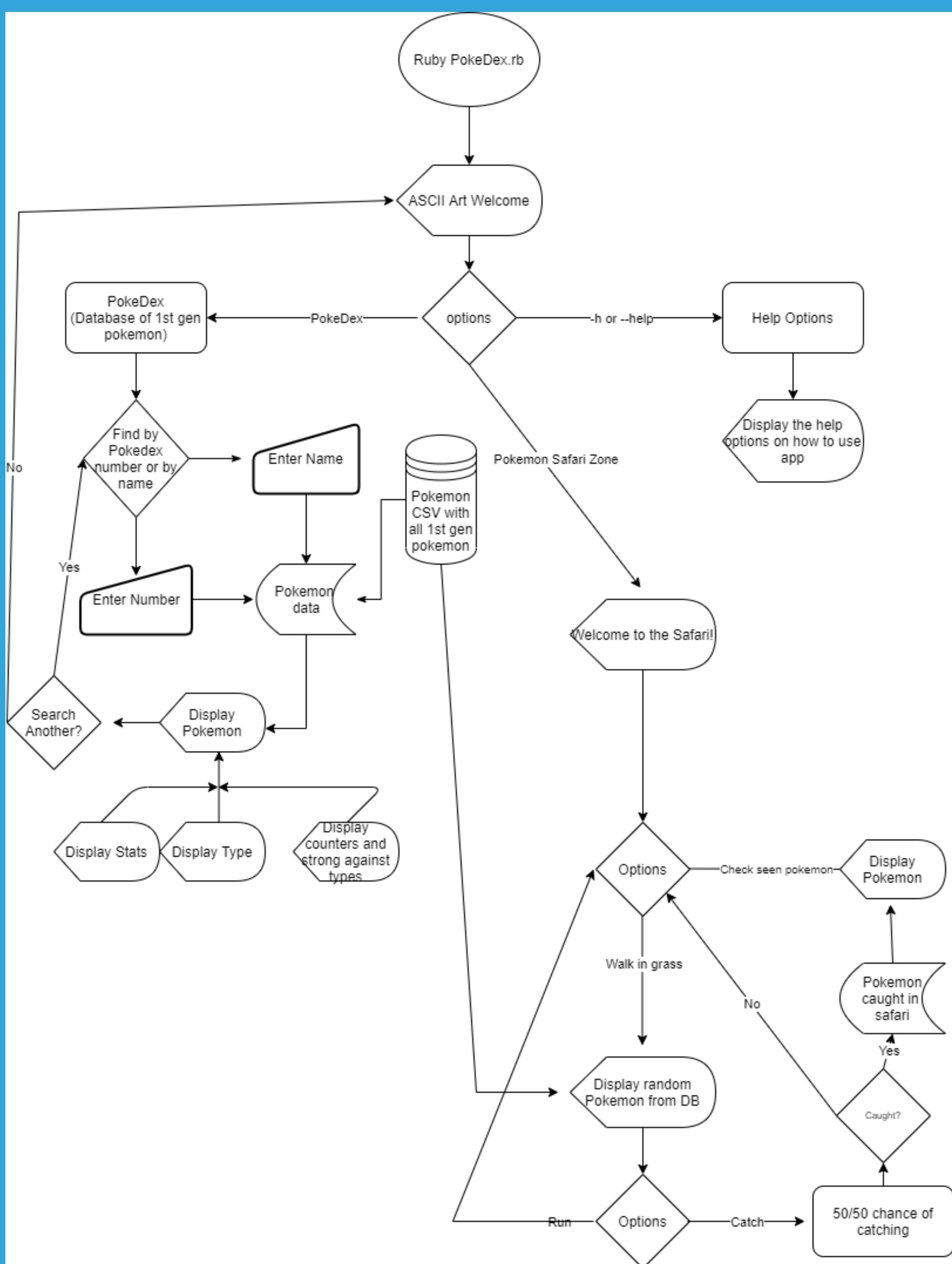
POKEDEX & SAFARI

**PROBLEM - NOT BEING ABLE TO REMEMBER
ALL THE POKEMON'S TYPES AND WHAT THEY
ARE EFFECTIVE AGAINST AND STRONG AGAINST**

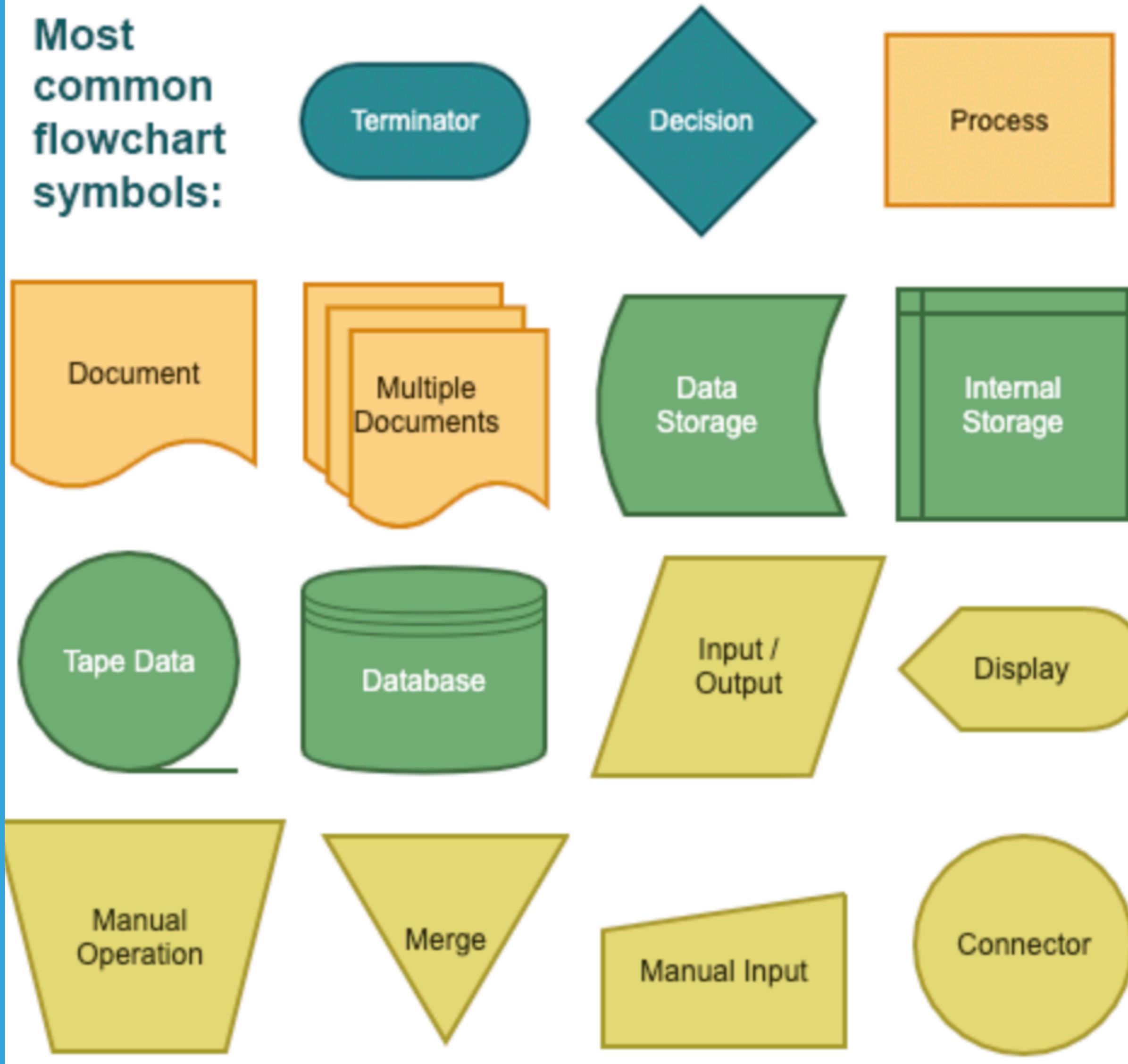
**SOLUTION - HAVING A QUICK TERMINAL APP
THAT CAN PROVIDE THE USER A QUICK DISPLAY
OF THE POKEMON'S TYPES AND WHAT THEY ARE
WEAK/ STRONG AGAINST, WITH THEIR BASIC
LEVEL 1 STATS.**



Modern problems require modern solutions



Most common flowchart symbols:



Legend - Flow Chart

FEATURE LIST

1 - Functional
pokemon searching
via name or pokedex
number

2 - Type database
that will run against a
chosen Pokemon's
type(s)

3 - A mini style game
of Pokemon Safari
that allows users to
see and catch a full
range of 1st gen
Pokem

ATLASSIAN SUITE

TRELLO BOARD

Poke Dex - Terminal App | Personal | Private | Invite | Trello | + | i | Bell | TH

Ruby Gems | ...
Colorize
ruby rspec
smarter_csv
+ Add another card

backlog | ...
Welcome Screen - ASCII Art
1
+ Add another card

Features | ...
ASCII Art used with colorized gem
Oct 2
+ Add another card

Need to do | ...
help menu - items
Oct 1 1
Create Error Exceptions
Oct 2 0/3
+ Add another card

Working on | ...
Display the pokemon's weak against and strong against
Oct 1 6 2/3
+ Add another card

Bugs | ...
Need to have a way to ask user if they want to see types or not
Able to use shortcuts for input (Y,N, name,) rather than full words (Yes,No,Pokemon Name)
1
When exiting safari does not loop back to main menu
+ Add another card

Finished | ...
Remove Mega's from DB
Sep 26 1
Set up file ready for code - Gems and folders initialized
Sep 18 1 5/5
1st Gen Pokemon data base (CSV File)
Sep 19 1 4/4
Search via Pokemon name or dex number function
Sep 20 1 4/4
Display Pokemon Name, Type 1/2, HP, Attack, Defence, Sp.Atk, Sp.def, Speed, total stats
Sep 18 1
Walk through tall grass "Display random Pokemon"
Sep 30 3 4/4
Pokemon Safari - feature
Sep 30 1 3/3
Add a 50/50 capture feature for pokemon safari
Sep 30 2 4/4
Check Pokemon caught in the Pokemon safari. (Display caught pokemon)
+ Add another card

Activity Show Details

TH Write a comment...

TH Tye Hadfield 5 hours ago
changing to later in the week as I need more time on the stuff I may require in class help on.

SMILEY - Edit - Delete

TH Tye Hadfield yesterday at 3:52 PM
changing due date for tomorrow, Glen threw spanner in the works and made me redo the class

SMILEY - Edit - Delete

TH Tye Hadfield yesterday at 11:10 AM Glen things
finished the class and now ready to be utilised in main code.

SMILEY - Edit - Delete

TH Tye Hadfield yesterday at 10:50 AM
Finished types_chart class now just need to do the outputs

SMILEY - Edit - Delete

TH Tye Hadfield yesterday at 9:52 AM
added to class and working on creating method for each to be able to be called in the program

SMILEY - Edit - Delete

TH Tye Hadfield Sep 25 at 7:56 PM
This was bigger than I thought, need to give each type a hash to be called upon later

SMILEY - Edit - Delete

COMMENTING ON THE IMPORTANT MATTERS

JIRA (BUG TRACKING)

The screenshot shows the Jira Software interface for bug tracking. The top navigation bar includes links for Jira Software, Your work, Projects, Filters, Dashboards, People, Apps, Create, and a search bar. The main area is titled "Bugs" and shows a list of issues under the "Poke Dex - Terminal..." project. A sidebar on the left contains filters for "Search issues" and "Starred" items, with "Bugs" currently selected. The main content area displays a list of three bugs, ordered by creation date:

- PDTA-11: When exiting safari does not loop back to m... (selected)
- PDTA-8: Able to use shortcuts for input (Y,N, name,) r...
- PDTA-5: Need to have a way to ask user if they want t...

The detailed view for PDTA-11 shows the following information:

Bug Details:

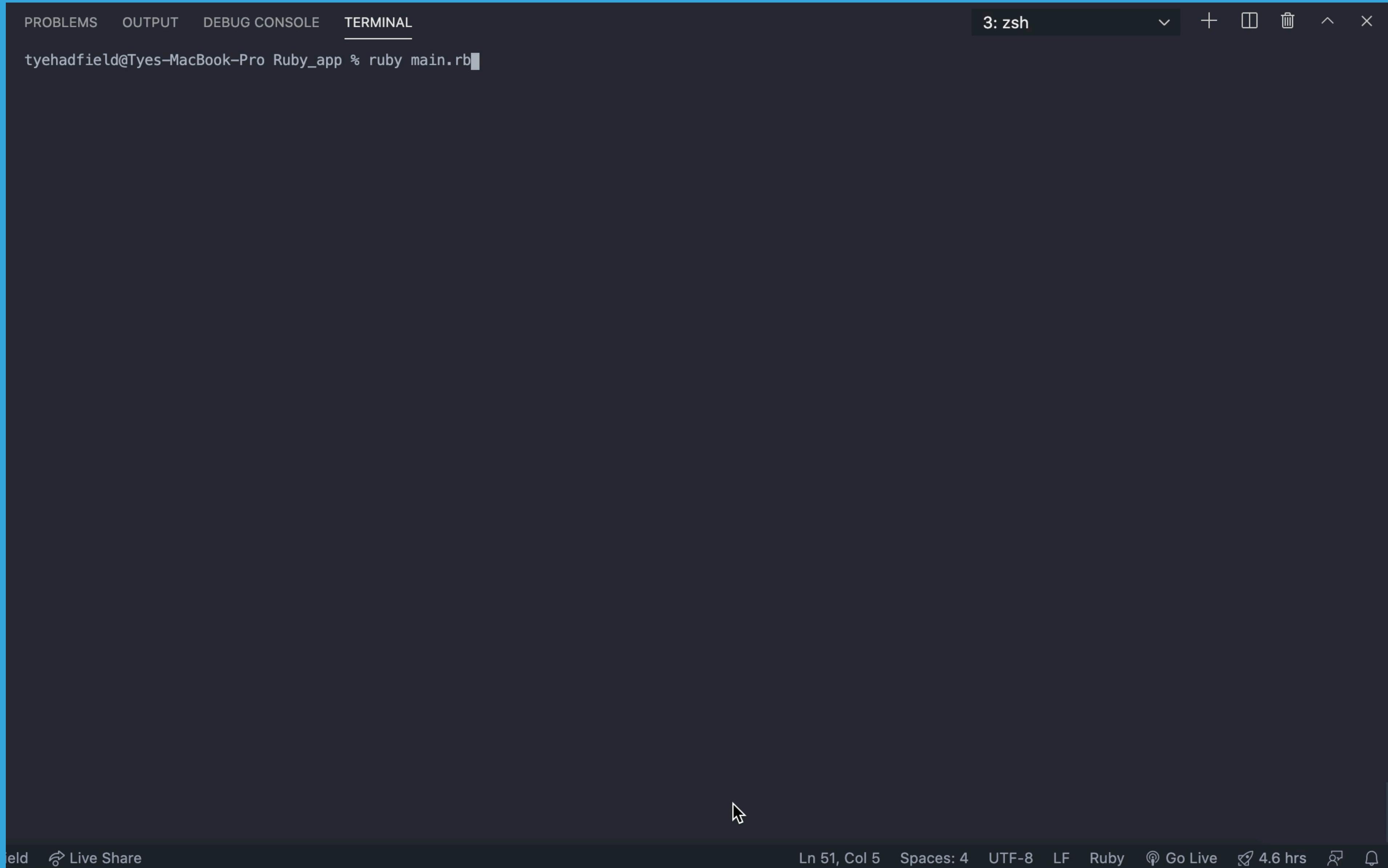
- Project:** Poke Dex - Terminal...
- Type:** Bugs
- Assignee:** Tye Hadfield
- Labels:** None
- Reporter:** Tye Hadfield
- Created:** 20 minutes ago
- Updated:** 4 minutes ago

Description: Add a description...

Activity: Show: Comments (selected) History

Add a comment... (with a keyboard tip: press M to comment)

APPLICATION SHOW CASE



A screenshot of a terminal window within the Visual Studio Code interface. The window title is "3: zsh". The tab bar at the top shows "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL", with "TERMINAL" underlined. The terminal content shows a user's command line: "tyehadfield@Tyes-MacBook-Pro Ruby_app % ruby main.rb". The status bar at the bottom displays "Ln 51, Col 5" and "Spaces: 4" along with icons for "Live Share", "Go Live", and "4.6 hrs". A cursor arrow is visible at the bottom center of the terminal window.

APPLICATION SHOW CASE

The screenshot shows a terminal window with the following interface elements:

- Top bar: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined).
- Title bar: 3: ruby.
- Close button: X.
- Minimize button: ^.
- Maximize button: □.
- Icon: A small icon representing the application.
- Content area:
 - User input: tyehadfield@Tyes-MacBook-Pro Ruby_app % ruby main.rb
 - Application output:

```
*****  
Welcome to the Pokedex and Pokemon Safari  
*****  
Please choose an option to begin!  
PokeDex | Safari | Help  
*****  
Pokedex  
*****  
How would you like to search the Pokedex, Name or Pokedex number?  
|
```
- Bottom status bar: Ln 51, Col 5 Spaces: 4 UTF-8 LF Ruby ⚙ Go Live ⏱ 4.6 hrs 🔍 🔔

APPLICATION SHOW CASE

A screenshot of a dark-themed code editor, likely Visual Studio Code, demonstrating a Ruby application. The top tab bar shows four files: `main.rb`, `caught_pokemon.csv`, `Types_chart.rb`, and `safari_class.rb`. The `caught_pokemon.csv` file is open, displaying the first two rows of data:

Name
1
2

The bottom navigation bar includes links for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, showing the command `tyehadfield@Tyes-MacBook-Pro Ruby_app % ruby main.rb`. The status bar at the bottom provides information about the current file: `dfield`, `Live Share`, `Ln 2, Col 1`, `Spaces: 4`, `UTF-8 LF`, `Plain Text`, `Go Live`, `3 min`, and a bell icon.

APPLICATION SHOW CASE

The screenshot shows a dark-themed code editor interface, likely from Visual Studio Code, displaying a Ruby application. The main window contains two tabs: "main.rb" and "Types_chart.rb". The "Types_chart.rb" tab is active, showing code that defines various Pokemon types and their properties. The code includes:

```
Classes > Types_chart.rb
73 types.push PokemonTypes.new("Ground", "FIRE ELECTRIC POISON ROCK STEEL", "GRASS BUG", "POISON ROCK", "WATER GRASS ICE")
74 types.push PokemonTypes.new("Bug", "GRASS PSYCHIC DARK", "FIRE FIGHTING POISON FLYING GHOST STEEL FAIRY", "GRASS FIGHTING GROUND", "FIRE PSYCHIC FAIRY")
75 types.push PokemonTypes.new("Water", "FIRE GROUND ROCK", "WATER GRASS DRAGON", "FIRE WATER ICE STEEL", "ELECTRIC GRASS")
76 types.push PokemonTypes.new("Flying", "GRASS FIGHTING BUG", "ELECTRIC ROCK STEEL", "GRASS FIGHTING BUG", "ELECTRIC ICE ROCK", false, "GRASS FIGHTING FAIRY")
77 types.push PokemonTypes.new("Fire", "GRASS ICE BUG STEEL", "FIRE WATER ROCK DRAGON", "FIRE GRASS ICE BUG STEEL FAIRY", "WATER GROUND ROCK")
78 types.push PokemonTypes.new("Psychic", "FIRE ELECTRIC POISON ROCK STEEL", "GRASS BUG", "POISON ROCK", "WATER GRASS ICE")
79 types.push PokemonTypes.new("Electric", "FIRE GROUND ROCK", "WATER GRASS DRAGON", "FIRE WATER ICE STEEL", "ELECTRIC GRASS")
80
81 types[5].output_all
```

Below the code editor is a terminal window titled "3: zsh" showing the command "tyehadfield@Tyes-MacBook-Pro Ruby_app % ruby Classes/Types_chart.rb". The status bar at the bottom of the screen displays "Ln 81, Col 20" and other system information.

CODE SHOW CASE

The screenshot shows a dark-themed code editor interface with the following details:

- EXPLORER** sidebar on the left:

 - OPEN EDITORS**: main.rb, safari_class.rb (selected), Types_chart.rb
 - RUBY_APP**: Classes (safari_class.rb, search_class.rb, Types_chart.rb), spec, caught_pokemon.csv, Gemfile, Gemfile.lock, main.rb, Poke_safari.rb, pokemon.csv

safari_class.rb content:

```
1 require 'csv'
2 require 'smarter_csv'
3
4
5 class Safari
6
7   def initialize
8
9   end
10
11
12 def safari_Catch
13
14   data_safari = SmarterCSV.process('pokemon.csv', {header_transformations:[ :none ]})
15
16   return_to_lodge = false
17
18
19   puts "*****"
20   puts "Welcome to the Pokemon Safari lodge!! Type catch to begin catching pokemon!!"
21   safari_choice = gets.chomp.to_s.capitalize
22
23   if safari_choice == "Catch"
24
25     until return_to_lodge == true
26
27
28     random_pokemon = data_safari.sample
29
30     puts "*****"
31
32     puts "You were walking through the grass and a wild #{random_pokemon[:name]} appeared!!"
33
34     puts "*****"
35
36     puts "Would you like to try and catch #{random_pokemon[:name]}?"
37
38     puts "*****"
```

Bottom status bar:

- master 0 0 5 0 0 Tye-Hadfield Live Share
- Ln 19, Col 1 Spaces: 4 UTF-8 LF Ruby Go Live 4.8 hrs

CODE SHOW CASE

The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, displaying a Ruby application for searching Pokédex data. The application uses CSV files for input and output.

File Explorer: Shows the project structure under "RUBY_APP". The "Classes" folder contains three files: `safari_class.rb`, `search_class.rb`, and `Types_chart.rb`. Other files in the root include `caught_pokemon.csv`, `Gemfile`, `Gemfile.lock`, `main.rb`, `Poke_safari.rb`, and `pokemon.csv`.

Search Bar: Shows the current file is `search_class.rb`.

Code Editor: The code in `search_class.rb` is as follows:

```
1 require 'csv'
2 require 'smarter_csv'
3
4 #####Pokedex searching by Pokedex number function#####
5
6 class PokemonSearch
7
8   def initialize
9
10  end
11
12
13
14  def pokemonTestPokeDexNumer
15
16    data = SmarterCSV.process('pokemon.csv',{header_transformations:[:none]})
17
18    user_answer = true
19
20    while user_answer == true
21
22      puts "Choose number between 1-151"
23
24      user_choice = gets.chomp.to_i
25
26      select_pokemon = data.select { |hash| hash[:pokedexnumber] == user_choice }[0]
27
28      puts ****
29      puts "Pokedex Number - #{select_pokemon[:pokedexnumber]}"
30      puts "Name - #{select_pokemon[:name]}"
31      puts "type(s) - #{select_pokemon[:type_1]} #{select_pokemon[:type_2]}"
32      puts "HP stats - #{select_pokemon[:hp]}"
33      puts "Attack stats - #{select_pokemon[:attack]}"
34      puts "Defense stats - #{select_pokemon[:defense]}"
35      puts "SP_Atk stats - #{select_pokemon[:sp_atk]}"
36      puts "SP_Def stats - #{select_pokemon[:sp_def]}"
37      puts "Speed stats - #{select_pokemon[:speed]}"
38      puts "Generation - #{select_pokemon[:generation]}"
39
```

Bottom Status Bar: Shows the file is "master*", line 3, column 1, with 4 spaces, encoding is UTF-8, line separator is LF, the language is Ruby, and the file was last modified 4.9 hrs ago.

CODE SHOW CASE

The screenshot shows a code editor interface with a dark theme. The left sidebar contains icons for file operations like Open Editors, Search, and Git. The Explorer panel lists files in a tree structure under 'RUBY_APP'. The main editor area displays a Ruby script named 'Types_chart.rb'. The script defines a class 'PokemonTypes' with methods for outputting attack and defense effectiveness against various types. A preview pane on the right shows the output of the script.

```
#This is the types and what their Strength and weakness are (Move to class when got time)

class PokemonTypes

  def initialize(name,attack_super_effective,attack_not_super_effective,defence_not_very_effective,defence_super_very_effective)
    @name = name
    @attack_super_effective = attack_super_effective
    @attack_not_super_effective = attack_not_super_effective
    @defence_not_very_effective = defence_not_very_effective
    @defence_super_very_effective = defence_super_very_effective
    @attack_no_effect = attack_no_effect
    @defense_no_effect = defense_no_effect
  end

  def output_effective_against
    puts "#{@name} moves are super-effective against: #{@attack_super_effective}"
  end

  def output_not_effective_against
    puts "#{@name} moves are not very effective against: #{@attack_not_super_effective}"
  end

  def output_defence_not_very_effective
    puts "These types are not very effective against #{@name} Pokémon: #{@defence_not_very_effective}"
  end

  def output_defence_super_very_effective
    puts "These types are super-effective against #{@name} Pokémon: #{@defence_super_very_effective}"
  end

  def output_attack_no_effect
    if @attack_no_effect == false
      return
    else
      puts "#{@name} moves have no effect on: #{@attack_no_effect}"
    end
  end
end
```

master* 0↓ 5↑ 0 △ 0 Tye-Hadfield Live Share Ln 81, Col 20 Spaces: 4 UTF-8 LF Ruby Go Live 4.9 hrs

CODE SHOW CASE

The screenshot shows a code editor interface with a dark theme. The left sidebar contains icons for file operations, search, and navigation, along with sections for 'OPEN EDITORS' and 'RUBY_APP'. Under 'RUBY_APP', there are sub-folders 'Classes' containing 'safari_class.rb', 'search_class.rb', and 'Types_chart.rb', and a 'spec' folder. The main workspace displays the content of 'main.rb'. The code is a script for a Pokedex and Pokemon Safari application. It starts by requiring relative files for classes and then defines a main function. Inside the main function, it prints welcome messages and a menu choice prompt. It then handles two main choices: 'Safari' and 'Pokedex'. For 'Safari', it creates a new instance of the 'Safari' class and calls its 'safari_Catch' method. For 'Pokedex', it asks the user how they want to search (Name or Pokedex number) and initializes a 'PokemonSearch' object if the answer is either.

```
1 require_relative 'Classes/Types_chart.rb'
2 require_relative 'Classes/search_class.rb'
3 require_relative 'Classes/safari_class.rb'
4
5 #####Main#####
6
7 def main
8
9   puts "*****"
10  puts "Welcome to the Pokedex and Pokemon Safari"
11
12  puts "*****"
13  puts "Please choose an option to begin!"
14  puts "PokeDex | Safari | Help"
15  puts "*****"
16  menu_choice = gets.chomp.to_s.capitalize
17
18  if menu_choice == "Safari"
19
20    poke_safari = Safari.new
21    poke_safari.safari_Catch
22  end
23
24  if menu_choice == "Pokedex"
25
26    puts "*****"
27
28    puts "How would you like to search the Pokedex, Name or Pokedex number?"
29
30    how_to_search = gets.chomp.to_s.capitalize
31
32  end
33
34  if how_to_search == "Pokedex number" || how_to_search == "Number"
35
36    puts "*****"
37    how_to_search = PokemonSearch.new
38
39  end
```

master* 0↓ 5↑ 0 △ 0 Tye-Hadfield Live Share Ln 51, Col 5 Spaces: 4 UTF-8 LF Ruby Go Live 4.9 hrs

IMPORTANT PIECES OF CODE

```
data = SmarterCSV.process('pokemon.csv',{header_transformations:[none]})
```

```
while user_answer == true
```

```
until return_to_lodge == true
```

```
select_pokemon = data.select{|hash| hash[:name] == user_choice}[0]
```

IMPORTANT PIECES OF CODE

```
if safari_pokemon_choice == "Catch"

    puts "*****"
    puts "You lock eyes with #{random_pokemon[:name]} and throw your Pokeball"

    number = rand(1..10)

    if number <= 5
        puts "*****"
        puts "Oh no! You missed and #{random_pokemon[:name]} got away"

    elsif number >= 6
        puts "*****"
        puts "All right! You caught #{random_pokemon[:name]}"
        CSV.open("caught_pokemon.csv", "ab") do |csv|
            csv << ["#{random_pokemon[:name]}"]
    end
```

UP'S AND DOWN'S

Up's

1 - Enjoyed prepping with Trello

2 - working with a data set I know and enjoy (Pokemon Gen 1)

3 - The little moments of holy crap it worked

Down's

1 - Not being sure if I am doing enough or pushing myself

2 - Realising I rely on VS code to do my commits and not doing the commands

3 - Ethically I am not sure if copy rights is an issue cause pokemon?

HOW I FEEL EVERYDAY

when you write 10
lines of code without
searching on Google



THANKS FOR LISTENING