

Terminal App (A1-1)

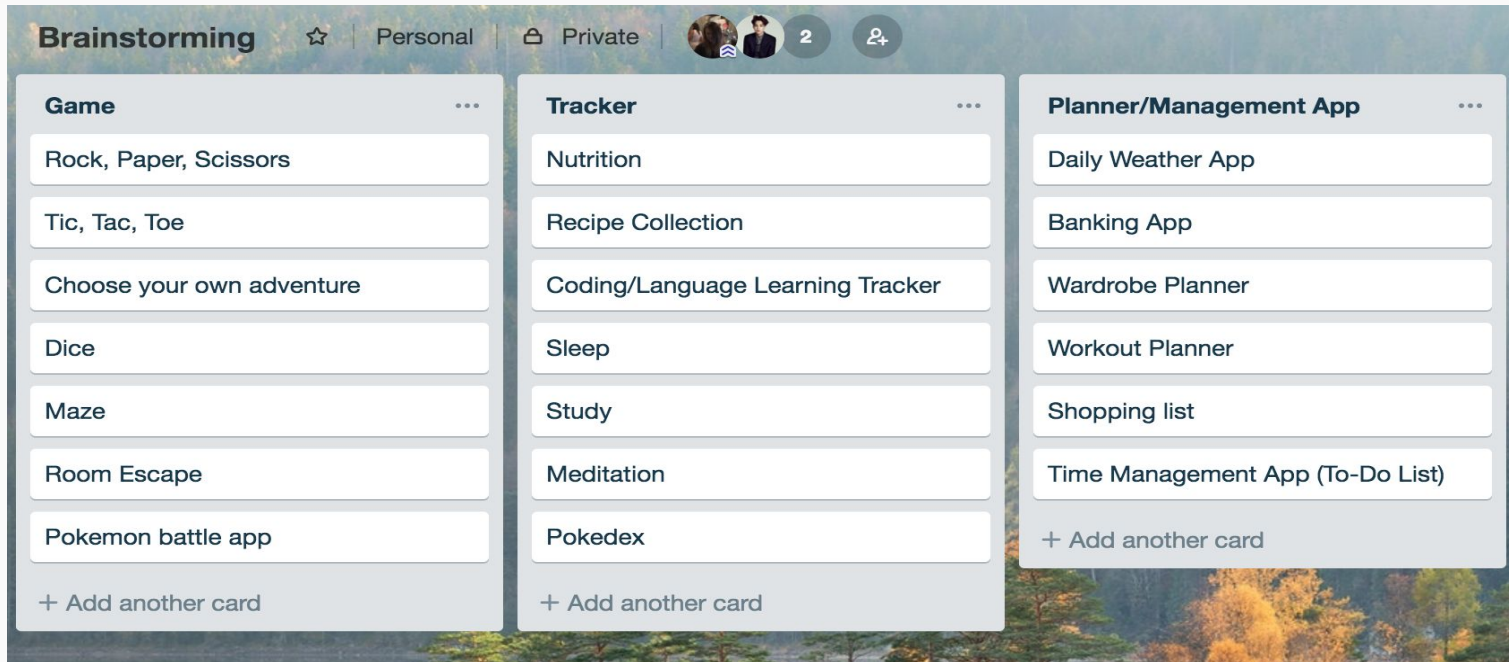
Shaahil Chand & Clementine Oldfield

Design Brief

“In groups of 2 create a Ruby terminal application (app).

“It can be something **to entertain**, to surprise, to solve a problem, to inform, to compute, etc.”

Initial Brainstorming



SQUIRTLE♂ Lv18

PAR



PIKACHU♂ Lv18

HP



43/ 43

EXP



Foe SQUIRTLE is paralyzed!
It may be unable to move!

Planning & Pseudocode

Pokemon Class

Game Loop

 Battle Loop

 end

end

Win / Loss Conditional

Planned Features

- Utilizing Pokemon types
- Leveling up
- Saving progress
- Critical Hits
- Moves that damage stats
- Poison/Paralysis
- Unique moves for each pokemon
-
- ASCII Animations
- Music
- Selection
- Critical Hits
- Moves that damage stats
- Unique moves for each pokemon
- Multiplayer modes

Core Features

- Option to choose a Pokemon from a list
- Option to choose a move from a unique list for each Pokemon
- HP stats for the user and the opponent that decrease with each attack
- Option for the user to play again after the battle has finished
- For the opponent to attack with a random move from its list of moves each round
- To have a user friendly graphical display

Feature ID	Test Case	Test Data	Expected Result	Actual Result	Status	Comments
1	Test fight between Pikachu (user) and Bulbasaur (opponent)	Bulbasaur dropped to -4 HP when Pikachu attacked	- "Bulbasaur fainted" - "You Win"	Bulbasaur still attacks	critical	Need to break out of loop if fighting attacks
1	Test fight between Magikarp (user) and Pikachu (opponent)	Magikarp's HP dropped to zero when pikachu attacked	- "Magikarp fainted" - "Game Over."	The loop begins again, prompting the user to attack.	critical	Need to refactor conditionals to check the correct time
1	Test fight between Bulbasaur (user) and Magikarp (opponent)		One attack for user, one attack for opponent	Opponent attacks twice	minor	
1	After choosing pokemon: Pikachu and move1: Tackle	Typed the string "tackle"	Pikachu used Tackle	As expected	passed	
2	After choosing pokemon: Pikachu and move2: Electric Shock	Typed the string "electric shock"	Pikachu used Electric Schock	"That is not a valid input"	minor	Need to capitalise every word with .
3	After choosing pokemon: Pikachu and move3: Growl	Typed the string "growl"	Pikachu used Growl	As expected	passed	
4	After choosing pokemon: Pikachu and move4: Body Slam	Typed the string "body slam"	Pikachu used Body Slam	"That is not a valid input"	minor	Need to capitalise every word with .
1	Chooses Pikachu all lowercase	types the string "pikachu"	Pikachu, I choose you!	As expected	passed	
2	Chooses Pikachu with a capital letter	types the string "Pikachu"	Pikachu, I choose you!	As expected	passed	
3	Chooses Pikachu spelt wrong	types the string "pikechu"	That is not a valid option	As expected	passed	
1	Battling for a second time in the same session		Battle starts again, as the first time, with a new Pokemon selection.	Pokemon HP stats have not been reset	critical	Create a class instance method that resets the pokemon before every battle
1	Typing yes after choosing a pokemon	typed the string "y"	Input is accepted and Moves displayed	As expected	passed	
2	Typing yes after choosing a pokemon	typed the string "Y"	Input is accepted and Moves displayed	As expected	passed	
3	Typing yes after choosing a pokemon	typed the string "yes"	Input is accepted and Moves displayed	That is not a valid option	minor	Add option to check for the word "y" "y/Y"
4	Typing yes after choosing a pokemon	typed the string "YES"	Input is accepted and Moves displayed	That is not a valid option	minor	Add option to check for the word "y" "y/Y"
5	Typing yes after choosing a pokemon	typed the string "yES"	Input is accepted and Moves displayed	That is not a valid option	minor	Add option to check for the word "y" "y/Y"
6	Typing a wrong input after choosing a pokemon	typed the string "a"	That is not a valid option	As expected	passed	

Code Demonstration