

IERG3080 Software Engineering Practice

(PROJECT PART II)

1155065267 Lui Cheuk Yan

Partner: 1155063984 Chan Chi Hang

Another part of this project is to do a text ver of this game.

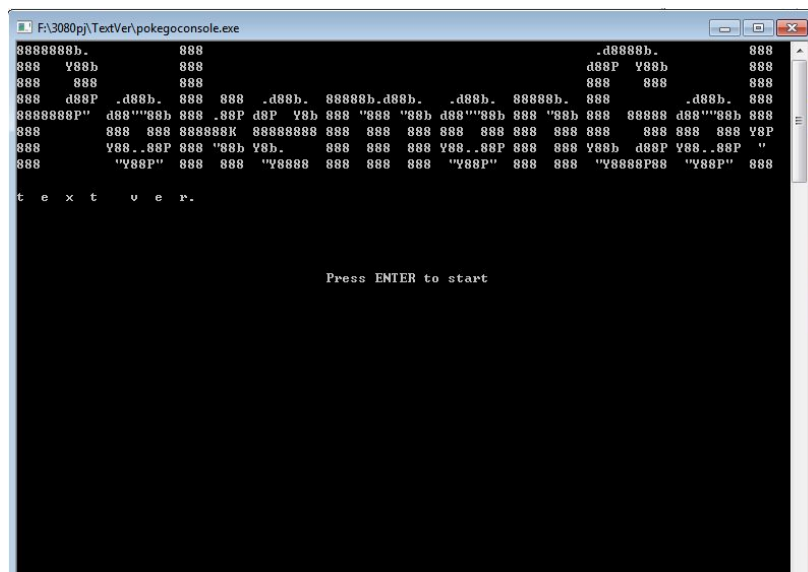
In the View part, they have completely different look.

It is Controlled by input text in a Console window.

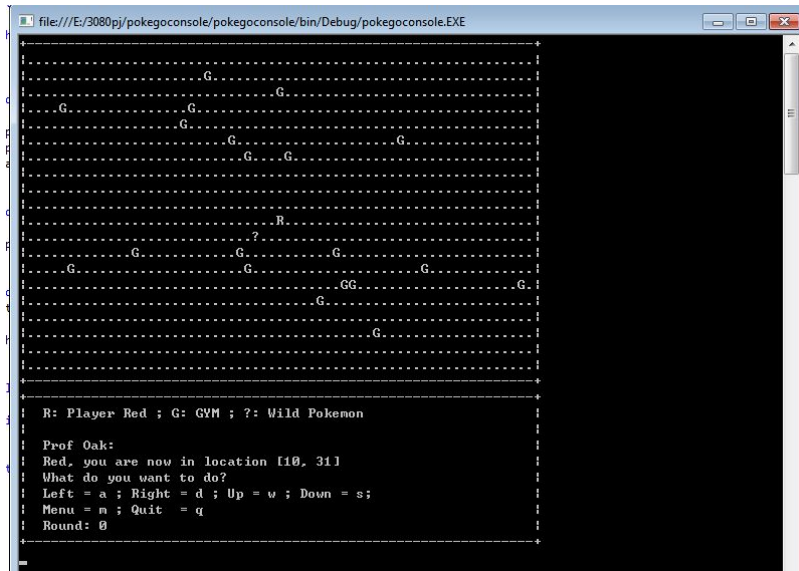
In the Logic part, they share most of the classes, instances, field and methods.

(Text ver. Class: ConsoleInterface)

(Shared Class: Pokemon, PokeTrainer and PokeGYM)



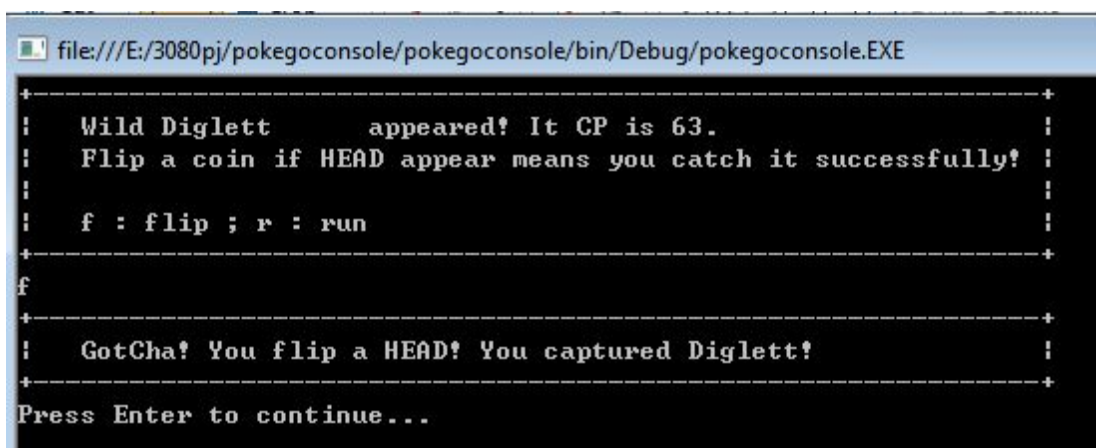
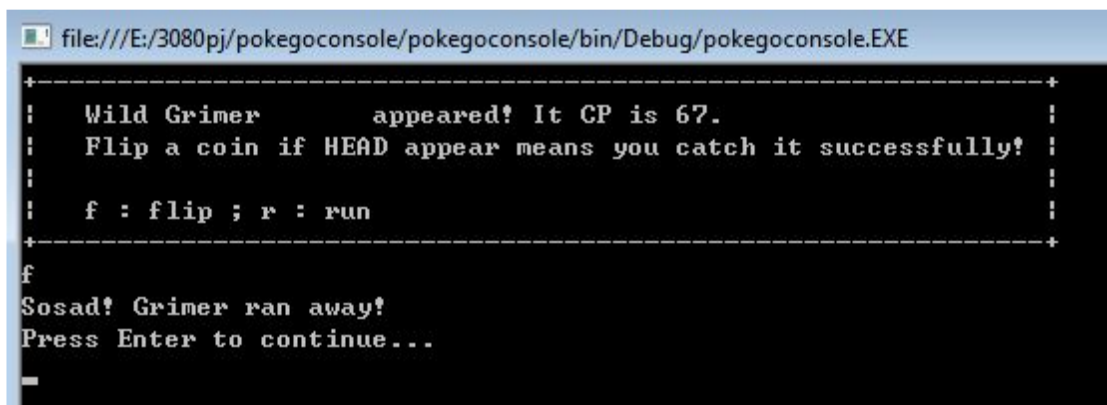
Starting screen



This is how the game looks like. The size of the map is adjustable. (by changing the value of the constructor)

Player follow the instruction on screen the play.

The Pokemon will randomly generated around the player. (default: 8*8 around player)



This is what happen if player meet a pokemon.(success or failure)

A coin flip mini game will determine (50%) where player can or cannot catch the pokemon.

```
file:///E:/3080pj/pokegoconsole/pokegoconsole/bin/Debug/pokegoconsole.EXE
+-----+
|:      *: Pokemon Candy : 6 | GYM badges : 0 *:      |:
|:-----+
|: No.      | Name      | CP      | Size    | Hp / Max HP |:
|: 1        | Diglett   | 95      | x1      | 61 / 61     |:
|: 2        | Tauros    | 68      | s       | 44 / 44     |:
|:-----+
|: p : power-up pokemon ; s : sell pokemon ; h : heal pokemon |:
|: m : return                                     |:
|:-----+
s
Sell one pokemon can get 3 Pokemon Candy.
Which pokemon do you want to sell?
2
Are you sure you want to sell Tauros? [y/n]
y
You get 3 Pokemon Candy from Prof Oak.
```

This is the inventory view. User can power-up, sell and heal pokemon and return to the main game. Power-Up a pokemon need 1 pokemon candy. Sell a pokemon will receive 3 pokemon candy. Heal pokemon can recover a pokemon HP to Max Hp.(After it get damage from GYM battle)

```
file:///E:/3080pj/pokegoconsole/pokegoconsole/bin/Debug/pokegoconsole.EXE
+-----+
|: This is a Gym.                                     |:
|:-----+
|: The latest Champion is Pidgey.   It's CP value is 339. |:
|: Do you want to fight it? [y/n]   |:
|:-----+
y_
```

This is the GYM view.
It shows the CP value of the GYM Champion pokemon and name.
Player can choose where they want to fight or not.
Player also can choose which pokemon to send out to fight.

```
file:///E:/3080pj/pokegoconsole/pokegoconsole/bin/Debug/pokegoconsole.EXE
Champion Pidgey CP:339 vs Player Diglett CP:378
*****
Champion Pidgey Attack!
Your Diglett HP: 250 / 304 !
*****

*****
Your Diglett Attack!
Champion Pidgey HP: 213 / 289 !
*****

*****
Champion Pidgey Attack!
Your Diglett HP: 196 / 304 !
*****

*****
Your Diglett Attack!
Champion Pidgey HP: 136 / 289 !
*****

*****
Champion Pidgey Attack!
Your Diglett HP: 142 / 304 !
*****

*****
Your Diglett Attack!
Champion Pidgey HP: 56 / 289 !
*****

*****
Champion Pidgey Attack!
Your Diglett HP: 88 / 304 !
*****

*****
Your Diglett Attack!
Champion Pidgey HP: 0 / 289 !
*****

Congratulation! You have defeated the GYM!
You earn 1 GYM badge and 10 Pokemon Candy
```

This is the GYM battle View.

Each pokemon attack each turn.

Damage per round is base on the base statistic(few randomness invote)

It stops will any one of pokemons is dead (HP<0).

Player will win 1 GYM badges and 10 pokemon candy.

Further explained,

class *Pokemon*, mainly handling spawning new pokemon, powerup and healing, Pokemon related calculation like their attack power, isDead method are included.

[reuse]Whole class is reuse because the logic is of Pokemon DATA is unchange in different view.

class *PokeGym*, handling gym logics, including generate random pokemon for battling with users.

[reuse]Whole class because the GYM logic is the same.

class *PokeTrainer*, handling trainers related logics, stored data of owned pokemon list, numbers of candys, pokemon and win badages.

[reuse]Most of the Method is reuse

[reuse]private List<Pokemon> POwnpokemon and public List<Pokemon>

[reuse]public method powerUpPokemon(), sellPokemon(), winGYM()

Other non-reuse classes, instances, field and methods are mostly related to in-game control/ in-game view logic, so we cannot reuse them.

Division of labour

Interface Design	Lui Cheuk Yan & Chan Chi Hang
Shared Class Design	Lui Cheuk Yan
GUI Ver Design and logic	Chan Chi Hang
Text Ver Design and logic	Lui Cheuk Yan