IERG3080 Sofwware Enginnering Practice

(PROJECT PART II)

1155063984 Chan Chi Hang
Partner: 1155065267 Lui Cheuk Yan

Our project is consutructed by four classes: Pokegym, Pokemon, Poketrainer and Pokeworld. Each of them handle and manage their related properties.

At the beginning when the game was launched, a screen (MainWindow) will shop up and ask for a click. After a click was received, the mainview window will shows up as the MainWindow will hide.

About the world, user can go left or right by holding their left/right arrow key. They can click on objects on the map to initiate an interaction. (The image of the map was downloaded from Internet, background house are not objects in our program!)

The element (pokemon) will start spawning at this moment, we try to made the program as generic as possible and the spawn logic is capable of handling other objects as well.

we control the pokemon population based on the following three rules,

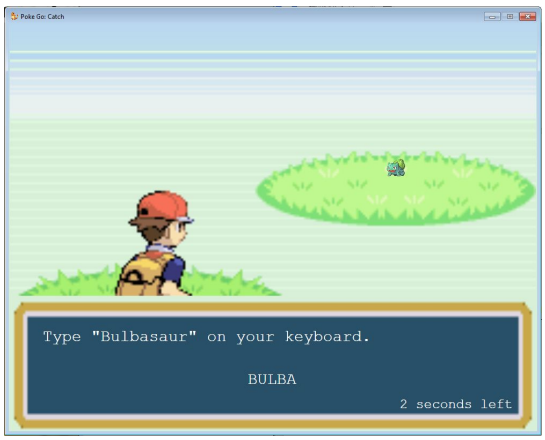
1. every certain 4 seconds, there are some 50% chance that appears a new random pokemon, and they will be within player's sight.

2. pokemon that out of player's sight will be disappeared every 4 seconds.

3. there are at most 5 pokemon in the whole map.

Of course those variables are adjustable according to user needs. Basically in our setting, you will almost always have pokemon in your sight. We believed that this is a good approach as it removes any redundant elements on map as soon as possible while keeping pokemon appears on player's sight.

After user click on any pokemon on the map, they will enter the catchview, which they can choose to catch the pokemon or run away. If they run away however, the pokemon will be disappeared on the map.



When user choose to catch the pokemon, they will have to enter the pokemon's name within five second in order to successfully capture it. The pokemon will be added to user's inventory if the capture was successful or otherwise he will receive nothing and the pokemon will run away.

When user click on its own image, it will enter the inventory view. In inventory view, user's information including their numbers of candy, numbers of pokemon and numbers of badges are shown, user can also manage their captured pokemon like "powerup", "heal" or "sell them" for stronger pokemon or more candy.

User may come across with Pokegym, they appear like the figure above, a orange little house. If they click on it, they will enter the battle view, which they can choose to have a battle with a random generated pokemon. User can choose which pokemon to be place on the battle and if they win the battle, they will receive 10 candys and 1 bandage as a reward. However, player's pokemon may die in a fight when opponent is having a larger cp/ap/hp.

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