

Object Oriented Analysis and Design Hw2

-Report-

İlkan Mert Okul, 1801042649

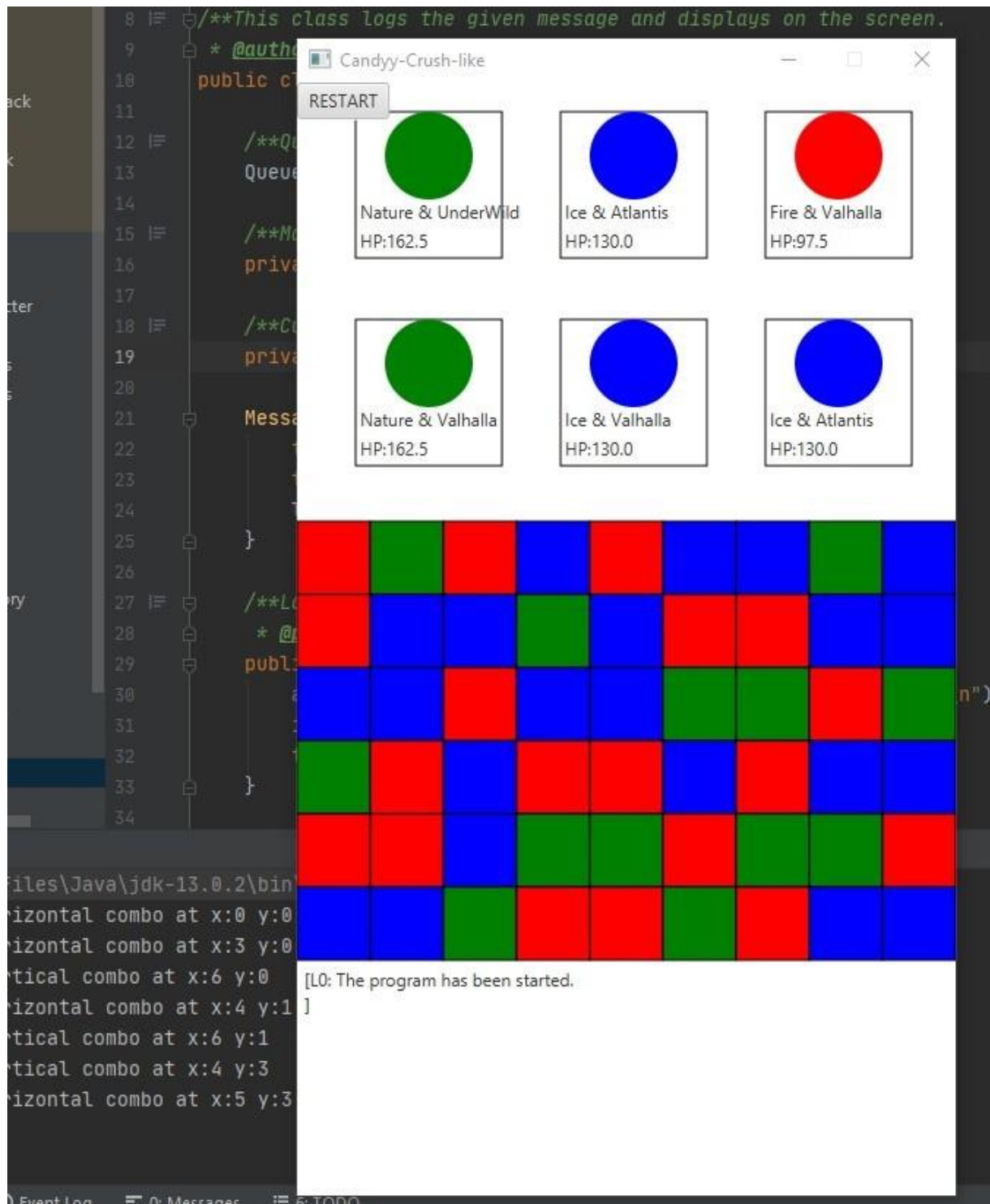
The code itself and their functionality and the whole plan of the program is heavily described in uml diagrams, javadoc, and the code itself.

The Gameplay:

The gameplay is quite simple, you click two block that are **neighbour**, and they swap. If you have any occurrence of a combo of three same colors, they disappear and it damages the corresponding enemy. Then, an alive enemy attacks a character of yours.

There are no animations involved, so when blocks disappear, new color of blocks appear right in there instead of other blocks coming from beneath. **This is the only missing part of the homework.**

When player kills all enemies, their healths replenish and they face a new set of enemies. If all friendly characters die, the game is over, and it cannot be played until user clicks on the restart button.



**Gameplay screenshot. Top three players are friendly,
Bottom three are enemies that attacks.**

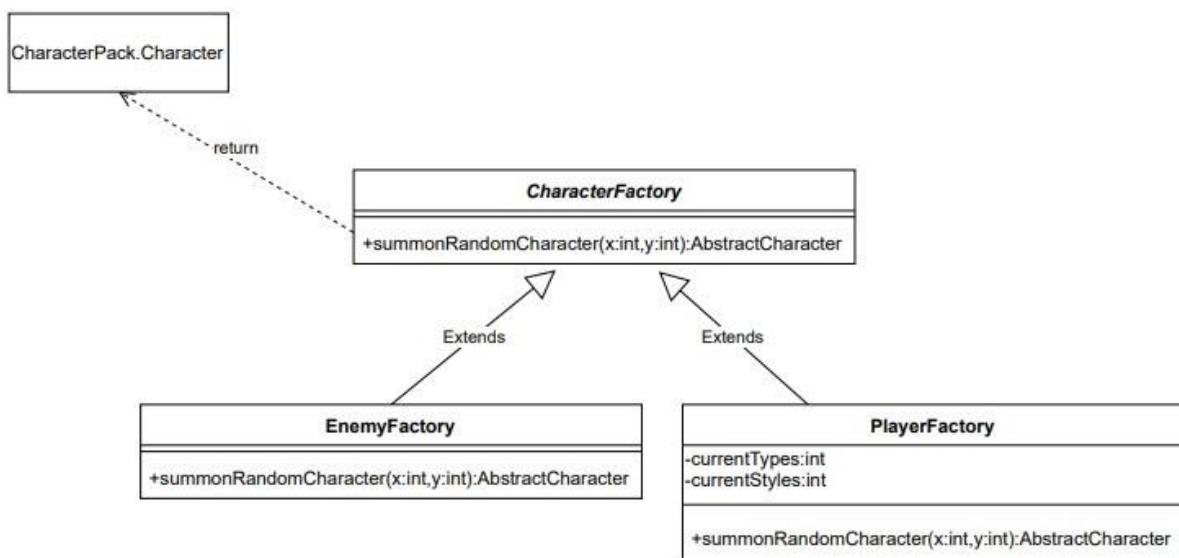
The Abstract Factory

This pattern used in character creation: for enemies and friendly characters.

The gameplay needs to have two different behaviors to create characters. Enemies should be truly randomized among their styles and types. So their factory creates truly randomized characters. And this factory tagges created characters as “Enemy”.

But friendly characters are unique. We have 3 characters and their style and type is unique. They are randomized inside to have different variatins and different locations.

So when GameManager wants to create a character, it gets the corresponding Factory class, and uses it.



DamageRule

This class helps the program to keep damage rules like :

GREEN attacks BLUE as 2X.

RED attacks BLUE as 0.5X.

Plus, if game expands in content, this helps to keep code updated easily.

Thanks for reading, İlkan Mert Okul 1801042649