

Gebze Technical University
Computer Engineering Department
CSE 443 – Object Oriented Analysis
and Design
Fall 2021 – 2022
Homework 2

HOMework REPORT

Türker Tercan

171044032

turker.tercan2017@gtu.edu.tr

Objective

Making a game like Candy Crush game that both AI and the player makes their turn and accordingly their turn, it causes damage to rival's characters or enemies. Construct the character and enemies with abstract factory design pattern.

There will be three types of color. Blue, Red and Green. A tile can deal more damage to characters or enemies based on the color of the tile and the color of the object it collides with.

	0.5x	1x	2x
Blue Tile ->	Green	Blue	Red
Red Tile ->	Blue	Red	Green
Green Tile->	Red	Green	Blue

A character or an enemy must have a type that is its color. Also, an enemy must have a style that can be Atlantis, Valhalla or Underwild. According to the character's style, it will have multipliers for strength, agility and health.

Types will be used to implement characters and, enemies' strength, agility and health attributes. Every color type it has own attributes.

Graphics

I used Stardew Valley 2D game's sprites for representing tiles, boards and background for the UI part I used what I used in first homework.

Implementation

I implemented all the animations for the tiles like changing tiles, shuffle and throw tiles to the characters and enemies when 3-row match happens.

I started the game without any matches that could happen on the board. So, when game starts or new game starts, it automatically shuffles the tile board.

I used two threads one for rendering and one for updating game status and one for calculate animations.

Strategy Design Pattern

I implemented abstract factory interface named as component factory abstract class. There are two concrete abstract factories which are named EnemyComponentFactory and, CharacterComponentFactory. These concrete factories are responsible for initializing Style and Type interfaces. I used Style and Type abstract classes as Products for Abstract Factory.

When a character or an enemy should be created, it should have taken the concrete factory is parameter. createComponents method is called inside of the constructor and, it calls to the factory to initialize Type and/or Style. After the creation of these objects, it assembles them and enemy or character object is ready to use.

