Gebze Technical University

Computer Engineering Department

CSE443 - Object Oriented Analysis and Design

Fall 2021-2022

Homework 2 Report

Murat YILDIZ

1801042004

Design of Character

- Character class has 5 fields which are style, color, strength, agility, health.

- There are 3 Type of Character. BlueCharacter, GreenCharacter and RedCharacter are inherited from Character class. They also additionally have characterFeaturesFactory field to create style that character will have.

- If someone wants to add a new Character type then he/she should inherit from Character class and must implement abstract initialize method and add as enum into Type enum class.

- That makes our job easy to add new type for the character.

- CharacterFeaturesFactory provide an interface and let subclasses to implement it.

- AtlantisCharacterFeaturesFactory, UnderwildCharacterFeaturesFactory, ValhallaCharacterFeaturesFactory are inherited from CharacterFeaturesFactory and each one implemented abstract createStyle method. With createStyle method, we decide which style character will have.

- If someone wants to add new style for the Character then he/she should inherit from Style class and implement its methods.

- That makes our job easier to add new style for the Character.

- AtlantisCharacterStore, UnderwildCharacterStore, ValhallaCharacterStore inherited from CharacterStore which has a createCharacter method to create a Character. It

takes type of the character as parameter (Type enum class) and according to type it creates it while giving it the right CharacterFeaturesFactory (according to store).

- After creating the rigth type of Character, we have initialize method in BlueCharacter, GreenCharacter and RedCharacter.

- Initialize method initializes the Character features such as style (calls its CharacterFeaturesFactory's createStyle method), strength, agility, health

- Calculating strength, agility, health fields are easy because of abstract factory pattern. All we do is multiply appropriate Character type's constants and style's multipliers.

Design of Enemy

- Enemy class has 5 fields which are style, color, strength, agility, health.

- There are 3 Type of Enemy. BlueEnemy, GreenEnemy and RedEnemy are inherited from Enemy class. They also additionally have EnemyFeaturesFactory field to create style that Enemy will have.

- If someone wants to add a new Character type then he/she should inherit from Character class and must implement abstract initialize method and add as enum into Type enum class.

- That makes our job easy to add new type for the Enemy.

- CharacterFeaturesFactory provide an interface and let subclasses to implement it.

- AtlantisCharacterFeaturesFactory, UnderwildCharacterFeaturesFactory, ValhallaCharacterFeaturesFactory are inherited from CharacterFeaturesFactory and each one implemented abstract createStyle method. With createStyle method, we decide which style Enemy will have.

- If someone wants to add new style for the Character then he/she should inherit from Style class and implement its methods.

- That makes our job easier to add new style for the Character.

- AtlantisCharacterStore, UnderwildCharacterStore, ValhallaCharacterStore inherited from CharacterStore which has a createCharacter method to create a Character. It takes type of the Enemy as parameter (Type enum class) and according to type it creates it while giving it the right CharacterFeaturesFactory (according to store).

- After creating the rigth type of Character, we have initialize method in BlueCharacter, GreenCharacter and RedCharacter.

- Initialize method initializes the Character features such as style (calls its CharacterFeaturesFactory's createStyle method), strength, agility, health

- Calculating strength, agility, health fields are easy because of abstract factory pattern. All we do is multiply appropriate Character type's constants and style's multipliers.