Şeyda Özer - 171044023

CSE443 – OBJECT ORIENTED ANALYSIS AND DESIGN

HOMEWORK 2

REPORT

My Design Decision Explanations:

- Firstly, I filled the 6x9 square tiles randomly. The tiles are created according to a random color (in blue, red, green). I encapsulated creating tiles with simple tile factory.
- I also randomly created player's and computer's entities. They were created according to a random type (red/fire, blue/ice, green/nature) and style (Atlantis, valhala, underwild). I encapsulated creating random entity with simple entity factory.
- If there is no match when the player or computer makes a move, I put the tiles back in their places.
- I there is a match when the player or computer makes a move, I store the owner of the last move. I needed this because after one match there can be many matches. Even if the turn passes to the next, the matches that occur will not damage the owner of the last move. The other one takes damage.
- If the tiles match, they will disappear, and the tiles below them will be shifted upwards to fill their place. During the shifting, neither the player nor the computer can switch the tiles. In addition, the fps value of the game decreases in order to see the tiles easily during this shifting. It increments when the shifting finished. When the fps value decreases, Mouse inputs for the pause and resume buttons may be gets late.
- For each tile, the damage calculated according to the entity it will attack and the xyz rule, when the tiles match.
- I save the position of the first click to switch the positions of two adjacent tiles clicked. If the second click is adjacent to the first, the tiles can be switched with each other easily.
- I did not implement shuffle the tiles when there are no moves left. I did not control this.
- I marked the res directory as the resource. The res directory contains my game images. I uploaded the images in the res file with getResource in the ImageLoader class.
- The Launcher class is my main class. It is launch the game.

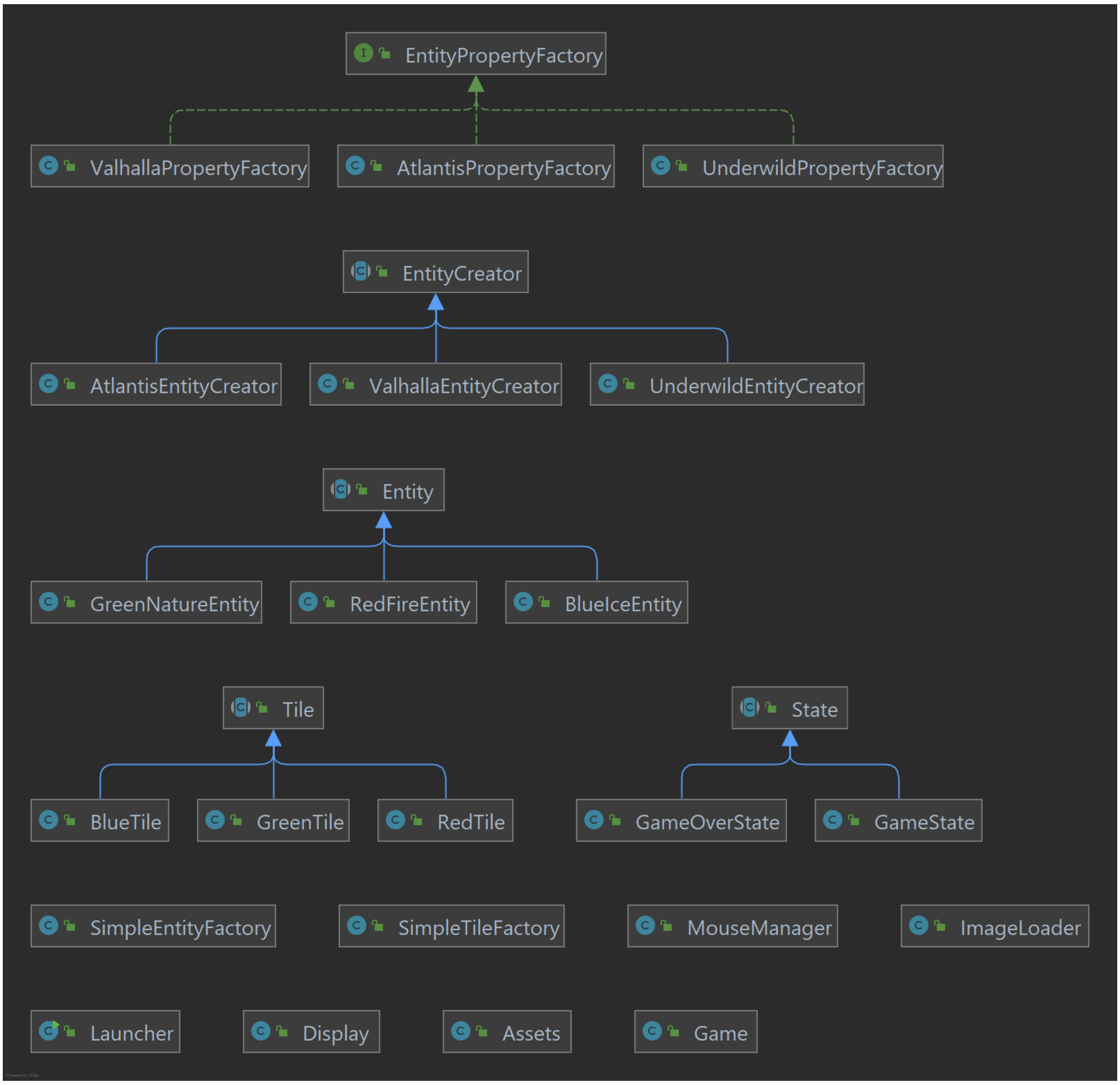
Şeyda Özer - 171044023

Abstract Factory Design Pattern:

An entity has three properties: strength, agility, health. There is no need to create classes or interfaces for these properties because they have numeric values. I used double for them. The main types for entities are blue, red and green. The product is the Entity and concrete products are red entity, blue entity and green entity. We need the style factory to calculate the properties of the entity. These property factories are Atlantis property factory, Valhalla property factory and Underwild property factory. These blue, red and green entities are creates using the property factories. The creators are Atlantis entity creator, Valhalla entity creator and Underwild creator. These creators create the red, blue and green entities with using own factory.

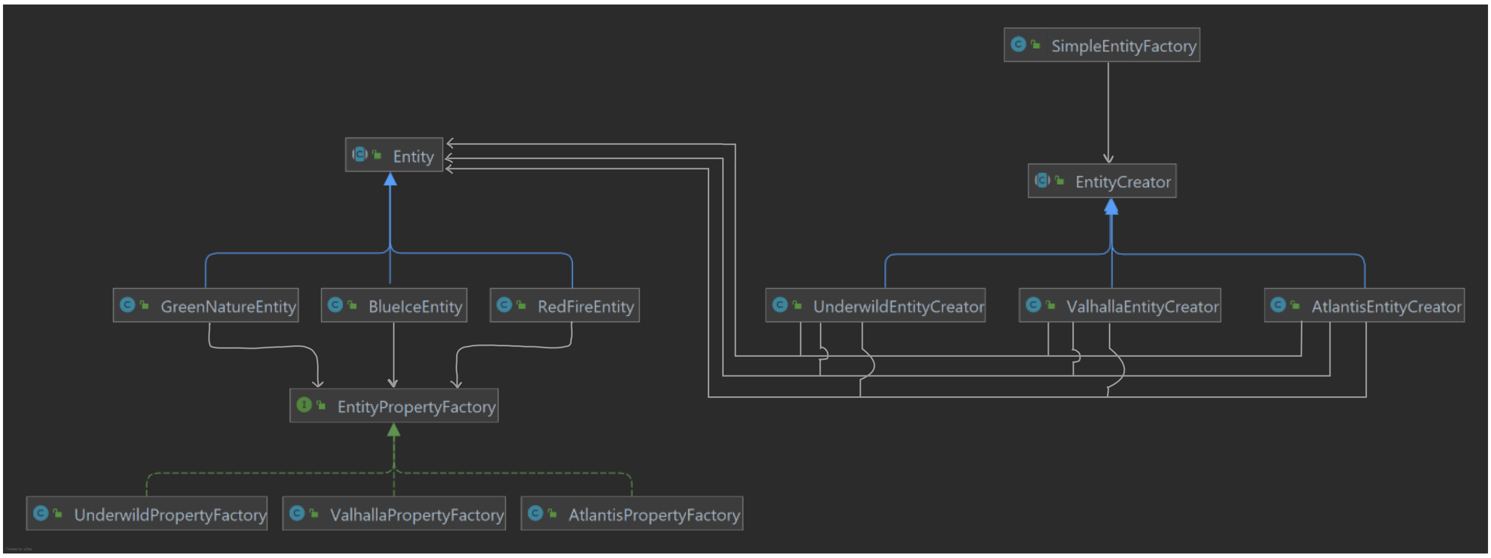
Class Diagrams:

All Classes:

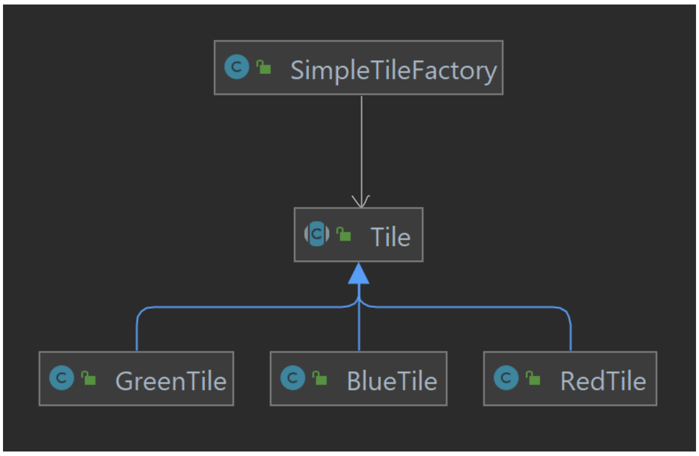


Şeyda Özer - 171044023

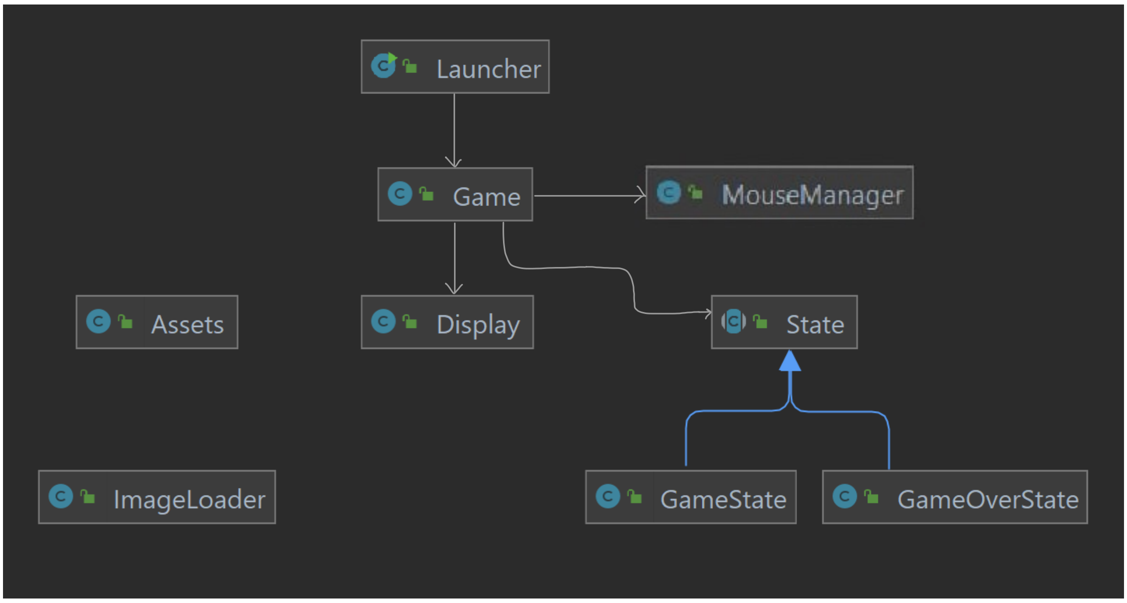
Entity Factory:



Tiles:



Game:


Launcher
Game
MouseManager
Assets
Display
State
ImageLoader
GameState
GameOverState