Gebze Technical University
Computer Engineering

CSE 443 - 2021 Spring

HOMEWORK 1 REPORT

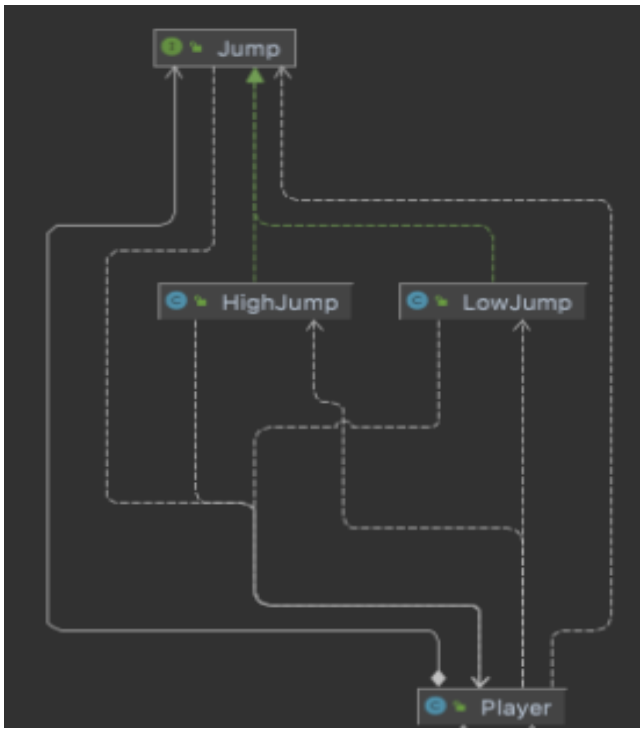
Dilara Karakaş

171044010

Introduction

Our game is a 2D single player game. There are monsters and obstacles that the player must jump over. In my design, the obstacle is a missile. There are powerups that the player must collect in order to increase the points received. The player can run by pressing the d key on the keyboard and jump by pressing the space key. One of the boosters changes the jump type. If the player is doing a low jump, it converts it to a high jump. Or if it is high jumping, it converts it to low jump.

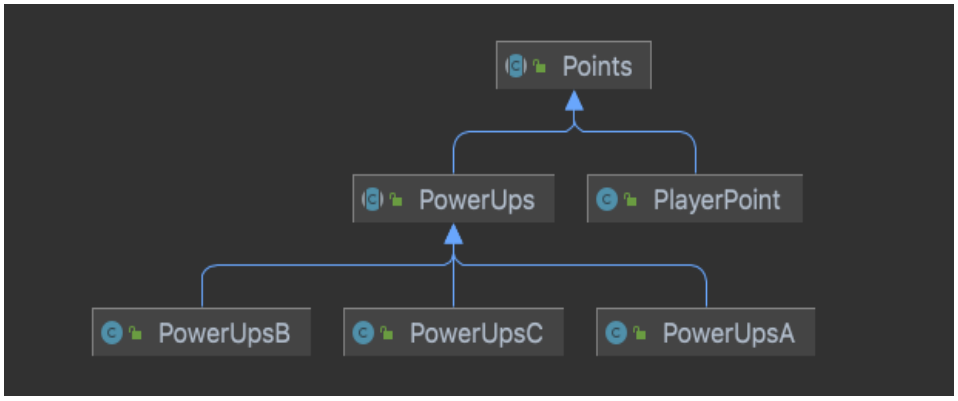
Design



- ***Strategy Pattern***

This pattern is used in the jump action. This allowed us to change the type of jump in the runtime when the player received the Boost D. A jump interface was made for this. Two classes implemented from this interface were created: High Jump, Low Jump. The difference between these two classes was to change the y coordinate of the player while jumping. A jump interface object is kept in the Player class. It was possible to change according to the jump type that came in the run time. By default, the player starts the game with a low jump.

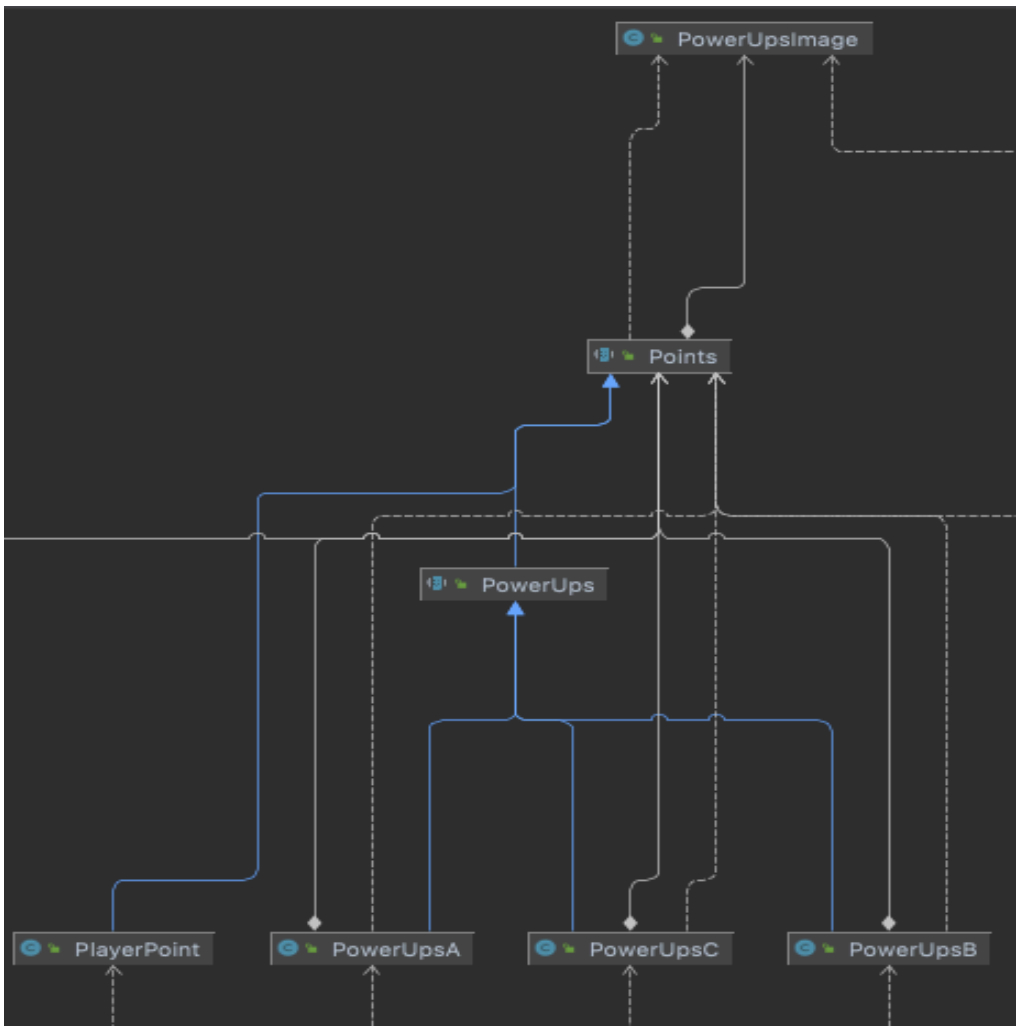
- ***Decorator Pattern***



This pattern was used for boosters. Points abstract class was created for Component. Created the PoweUps abstract class for Decorator. PowerUpsA, PowerUpsB, PowerUpsC classes were created for Concrete Decorators. Our ConcreteComponent class is also our PlayerPoint class. As for the relationship between them, there is the PlayerPoint class and the PowerUps abstract class that have been extended from the Points class. Boosters are also extended from these PowerUps classes.

Our PowerUpsA, PowerUpsB, PowerUpsC classes contain the Points object and take input as a variable in the constructor structure. In this way, every time a new object is created, it is wrapped with the previous object. The values wrapped here are power times values. On the other hand, point values are kept in the Board class.

If we show a more comprehensive diagram ;



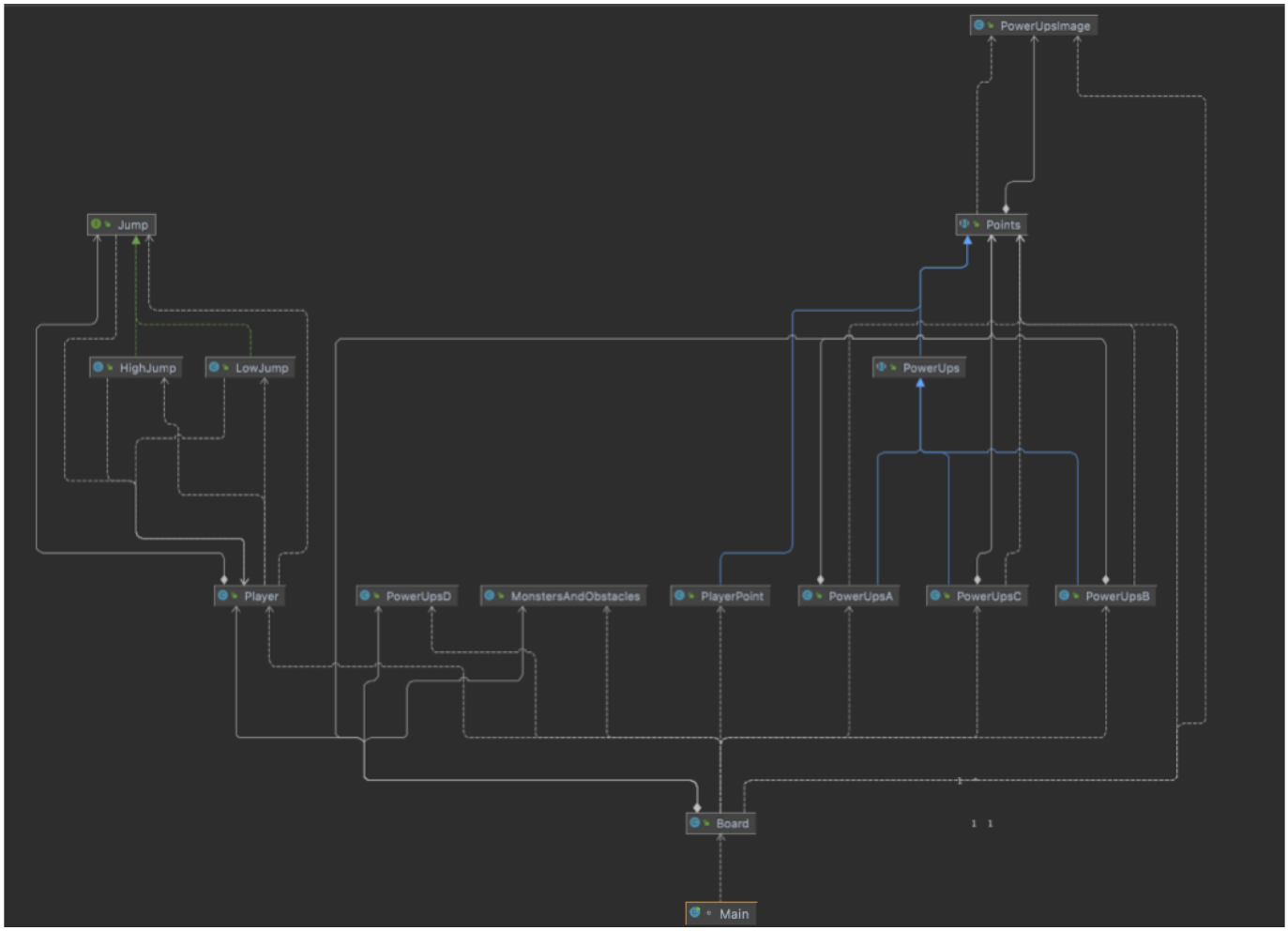
Class Diagram and Other Classes

Let's talk about classes and techniques used outside of patterns.
We have one Board class. It has extended from the JPanel class. In addition, it has been implemented from the ActionListener and KeyListener interfaces. Our game is generally played here. It holds all game-related objects in it.

We have one Player class. The player performs operations such as pictures, pressing and holding the jump object.

Our PowerUpsImage class provides images of powerups and their suppression, movement. It keeps the coordinates and, when generated, ensures that the x coordinate comes in random ureteral different places.

Our MonstersAndObstacles class is our obstacle and monster class. This class is used to randomly generate monsters and obstacles. It holds the coordinates of the pictures of the objects in it. Scrolling takes place here.



Desired but not realized in the game

- End the game when player life reaches 0
- Menu
- FPS display on screen