

Ofir Ben Shoham

Id: 208642496.

## ReadMe File

In this text I will explain in which interfaces and classes I used in my code in order to programmer this game, with extending the former game – Arknoid.

I used the **interface: Shooter**.

This interface has a method that has the ability to shoot balls. It has a void function that get GameLevel object as a parameter and create a new ball. Later adding his to the game (GameLevel).

In our new game, we should have the ability of the paddle & the Alien to shot. Therefore, both classes implement the interface Shooter, and have the ability to shot new ball (adding new balls to our game).

The new classes I wrote are as the following:

**Alien class:** this class represents an Alien in our game. This object has the ability to move and change his velocity to the other side. It makes implement in Shooter, because the Alien should has the ability to shot new ball. Moreover, the Alien class extends Block. I did this approach in order to save duplicate code. For my opinion, this inheritance is makes sense since each Alien consist of a block, and his image is an image of an Alien.

In the new game I supposed to move a group of Aliens together. In order to achieve this, I created new class that called: **AliensCollection**. Also this class implements the Shooter interface. Because we need to choose randomly with Alien in our group should shoot new ball.

In addition, the AliensCollection class has List<List<Alien>>, in order to represent all the group of Aliens together. So, it has some methods: Shoot ball randomly, find the most right Alien in our group, and find the most left & the lowest. This needed in order to change their velocity to the other side when the most right/left hit our border or hit our shields.

When the player loses a life, the alien's collection should return to their beginning. Therefore, it has a method that returns them to the default places.

About the **shields**: for my opinion, it doesn't make sense if I were made the shield class to extend the block. Since, shield consists of a lot of little blocks. In order to make this approach, each Shield has the attribute of: List<List<Block>> and upper left point. We have 3 parts of shield in the game. So in GameLevel object we have list of

shields with three indexes, each one for one shield object, after initialized him with the initialize method that shield has. In addition, the shield object has a method that adds him to our game. This is very important since our blocks hit and disappear just if they part of our game, according our last implement when we created the game Arknoid.

How did implement shot by aliens? As I wrote the AliensCollection has a method that shoot a ball randomly. According the instructions we were required that every 0.5 seconds an alien shoots a ball. So, in order to check the time was passed, in GameLevel there is a variable that counts how much time was passed that starts from zero. Each 0.5 seconds it shoots a ball from a randomly alien, and then the variable returns to be zero and continue counting.

The approach to the player shoots is very close. In the paddle class I added a counter that counting the time was passed and every 0.35s passed from the last shot it has the ability to shot more one ball. After the next shoot, it comes to zero again.