Shani Herskowitz 321658387
ass7

classes and interfaces used:

Aliens: Controls aliens movement, Is in charge of resetting aliens into formation. Also is assigns the correct alien to shoot.

InFormation: Is a member of Aliens class, controls the movement in a group, down, left, right. Resets the aliens in formation.

MovesFormation: Is a member of aliens class. checks when to move left or right.

Shooter: Is a member of GameLevel. Is called in do one frame. controls shooting both from aliens and from space ship. Gets correct alien to shoot from Aliens class. Measures time between shots.

Level: The basic level of the game. creates the shields by adding them to block list. Holds a list of aliens (array list of blocks), gets speed and level name from constructor.

Level Information (Interface): changed the interface to receive to lists of blocks, one of the shields and one for the aliens that way i can defreinciate between them. Also added Speed of aliens.

Implementation:

(a) the Aliens formation :

Using the classes described above. I used InFormation to control to movement by looping over all blocks in the aliens list and moving the rectangle based on the speed of the game. I used a boolean to confirm when i needed to move left, right. whenever i did i went down one.
In order to reset i used a counter to count the movement down and then measured the space from the left most alien to the border and moved all aliens by that amount.

(b) the shields :

I added shields to the level information interface and had it hold a list of very small blocks.

(c) shots by aliens :

The Aliens class is in charge of finding the random alien from the bottom row. I did this by looping over my aliens and taking the first row of aliens into an array.
I then looped over all the aliens and compared the elements in my array, if an alien with the same x value was found i swapped them, that way i always end up with the alien at the bottom of each column. I used a random func on the bottom row array and sent a point back from the function (the middle point from the alien that needs to shoot)
My Shooter class then created the ball and shot it from the correct location.

(d) shots by player :

The Shooter class has a shoot function that receives the paddle and keyboard and checks when to shoot. In the Shooter class i have two timers that check when its okay to shoot from the player and when from the aliens. I do this using an int and counting down based on the frame rate.