Ass7Game:

New Classes added:

All new classes are in a package named invaders , the rest are the same classes from ass6.

The invaders level specifications are being read from space_invaders_level txt file like in ass6.

Class InvadersGameLevel extends GameLevel:

*Upon construction gets the enemy formation initial move speed (each wave will get a 10% increase) and the current wave number to display, sets this game to remove shoots that collides with the game level screen blocks.

*responsible for the game level logic and initializing and creating the game components like:

Player(paddle), player shots (if space was pressed and with a 0.35 seconds delay between shots),screen limit blocks, the game shields , the enemy formation (columns and enemies).

responsible to run the level animation and stop the level when stopping conditions are met, as long as the player has lives and he didn't clear this level enemy wave ,the player loose one life if the player was hit by enemy shot or the enemy formation reached the shields, resets the remaining enemies to the formation original start point.

Class InvadersGameFlow:

This class is similar to the normal Game flow with one change to run the same level with increase of the enemy formation initial speed of 20% each new wave, will run as long as the player still have lives, when the player have no lives , if the player score deserve to be added to the game high-scores, the player will be asked for his name and be added to the game high-scores, finally the game high-scores are displayed.

Class Formation:

*This class is responsible for creating the enemy columns and coordinate their movement and the random column shooting every 0.5 seconds.

The whole enemy movement is handled by the formation which call all its columns which in turn call all its enemies (same for the movement speed).

The whole enemy shooting is handled in the formation (with a timer), it chooses a random column among its columns and make a shot from the last enemy in that chosen column.

*Given a list of enemies (blocks) divides them and creates a column with 5 enemies and adds it to this formation column list.

*This class keeps a list of all enemies shots created in order to clear them once the player turn ends.

*uses a timer with a delay of 0.5 seconds with a task that changes a Boolean value formationCanShoot to true in order to allow the formation to randomly shoot.

*shoot() – this method will get a random column among the formation columns and will create and add a shot from the lowest enemy in the column(changes the Boolean value formationCanShoot to false).

*has a method when called resets the whole formation to its starting position with the initial move speed.

*has a method to increase the whole formation move speed by 10%.

*has a method to check if the formation reached the shields.

* has a method to remove an enemy from the formation and its respected column.

When the formation hits the left or right side of the screen, the aliens in the formation (a) go down; (b) change direction; and (c) increase their speed by 10%. The formation is considered to hit the side of the screen when the left-most (or right-most) alien in the formation hits the side of the screen.

When the lowest alien in the formation reaches the height of the shields (or where the shields were, if all shields are destroyed), the player looses a life.

Class ColumnFormation:

This class responsible to coordinate movement and shots for enemies, enemies that belong to the same column are those with the same starting x position.

Each column have 5 enemies.

*the column keeps a list of the enemies inside it ,and a member of the formation it belongs to.

* when adding the first enemy to the column its top_left point is kept,in order to reset it to the same position when the turn ends.

*has a method to change the whole column initial movement speed.

*has a method to remove an enemy from the column.

*has a method to return the lowest(last) enemy in column.

*has a method to move the whole column to given x,y parameters.

*has a method to reset the whole column to its starting position.

*has a method to increase the whole column movement speed by 10%.

*has a method to shoot from its lowest enemy (possible only every 0.5 seconds).

Class Enemy extands Block:

This class is a block with the ability to move and shoot.

Upon construction the enemy keeps the formation and column formation it belongs to, in order to coordinate its movement and shots.

*has a method to change the enemy initial movement speed.

*has a method to check if this enemy is the last enemy in its respected column.

*has a method to return its bottom middle point.

*has a method to move the enemy to given x,y parameters.

*has a method to reset the enemy to its starting position.

*has a method to increase the enemy movement speed by 10%.

*has a method to shoot from its bottom middle point (possible only every 0.5 seconds).

*upon notifying a hit this enemy is removed from its column and formation.

Class player extends Paddle:

*This class is an extension of paddle , with the added ability to fire a shot every 0.35 seconds.

*This class has a method shoot() for adding a new player shot from the middle of it, a shot is fired if space key is pressed, a new shot can be fired every 0.35 seconds.

*This class keeps a list of all player shots created in order to clear them once the player turn ends.

*This class hit method removes the hitter (shot) upon impact and inform of the player being hit because its one of the player turn stopping condition.

Class Shot extends Ball:

*This class is an extension of Ball, and it represents a shot ,both for player and enemy.

*Upon construction of a new shot it gets a Boolean value , true for an enemy shot false for player shot.

*Has a method to add a collideable Enemy to this shot uncollideWith list, each enemy added upon impact will remove this shot without harming the enemy itself.
*Has a method to increase the shot movement speed by 10%.
*the moveOneStep method will let the shot move as long as there is no impact, upon impact will check if the collided object is among his uncollideWith list (and if the shot is an enemy shot), if it is the shot is removed from the game without harming the enemy, otherwise the collided object and the shot are removed from the game.

Class shields :

*A shield is no different then any other block except that it will not be counted upon removal among the needed number of blocks to clear the level.

*This class has one method that construct and returns a list of blocks shields with specific size and locations, 3 shields are created each with 4 rows of 5 by 5 blocks .