Assignment 7- README file

Username: shenkmr

ID: 311130777

Name	Type	Purpose and Responsibilities
BallRemover	Class	A hit listener for removing balls after a hit.
BlockRemover	Class	A hit listener for removing blocks, after a hit by removing algorithm it holds.
CollisionInfo	Class	An object which holds a collision point and object which the ball collided with (used for checking the ball's movement).
Counter	Class	A object that counts with integers, can be decreased and increased and used for scores, lives, number of balls and number of blocks.
GameEnvironment	Class	This class holds all the collidable objects in the game, used by the ball the find out which object it is going to collide with.
ScoreTrackingListener	Class	A hit listener for counting the score of the game (increases its counter whenever a hit with alien occurs).
Velocity	Class	An object that holds the movement details, can be in form of angle and speed or just DX and DY. Each ball holds its own velocity.
Ball	Class	A circular object, moving around the screen and hitting blocks, changing its velocity when a hit occur and can be shot by pressing space (to destroy aliens).
Block	Class	An object that holds a rectangle, has method to set its location and is used for many purposes (shield, paddle, alien, borders) when it gets hit by a ball it returns a new velocity for the ball, also can be removed from game according to it's purpose.
EndScreen	Class	An animation which is shown after the game has ended, shows the score.

HighScoresAnimation	Class	An animation which holds a high score table, used after the game has ended or when ‘h’ is pressed in the menu, shows the high scores.
LevelBackground	Class	A sprite which holds an image or a color, used for the level background.
LevelNameDrawer	Class	A sprite which holds a string, used for drawing the name of the level.
LivesIndicator	Class	A sprite which holds a counter of lives, used for drawing the left lives.
MultAlienColumn	Class	A sprite which holds a list of alien columns, manages all the movement of the aliens and initializing the columns.
Paddle	Class	A sprite and collidable, a block which holds a keyboard sensor (can be moved by user) and its purpose is to keep balls from falling down.
ScoreIndicator	Class	A sprite which holds a counter of score, used for drawing the score of the user.
Shield	Class	A block object (extends the block class) for creating a shield.
SpriteCollection	Class	A collection of sprites, using this for notify all sprites that time passed and drawing all the sprites.
AlienColumn	Class	This class manages a column of aliens, creates the aliens, and moving them by inputted speed and orders.
Line	Class	This class features a line from 2 points, has an intersecting method (used for checking intersection of ball and other objects which holds a line)

Point	Class	This class features a point made from 2 coordinates (x and y).
Rectangle	Class	This Class features an Rectangle (made from 4 lines).
HighScoresTable	Class	This class features a high score table, which allows adding a new high score (according to an adding algorithm) and saving and loading a table.
ScoreInfo	Class	This class stores information about an high score (name and score).
Animation	Interface	This interface featues an animation which has a do one frame method and should stop boolean.
Collidable	Interface	This interface is used to describe the basics of collidable object (an object the ball can collide with).
HitListener	Interface	This interface is used for the Listener pattern to describe an object that gets notified when a hit occur.
HitNotifier	Interface	This interface is used for the Listener pattern inform the HitListener object that it there was a hit.
IsPaddle	Interface	This interface is extended by Collidable interface, it's purpose is to know if the object we collided with is paddle.
LevelInformation	Interface	This interface features a level information, that holds all the information required to start a game.
Menu	Interface	This interface features a generic menu, allowing to add tasks and sub menus and get its current status. It also extends animation

Sprite	Interface	This interface is used to describe the basics of a sprite object (an object the game can draw and make changes to its visibility).
Task	Interface	This interface is used to describe a task, that is being operated from the menu (when the user presses a certain button).
SingleLevel	Class	This class features a single level object, implements level information and holds all the needed information for level to run and also multAlienColumn sprite (for controlling the aliens).
AnimationRunner	Class	This class features an animation runner object, which run an animation object (like a high score animation)
CountdownAnimation	Class	A countdown animation, shown every start of level or after the user loses life (can select time and number to start from).
GameFlow	Class	This class operates the game (in an endless loop until the user dies), it initializing a level according to the current level (start with 1) and then creates a GameLevel object with this level and run it (playOneTurn method).
GameLevel	Class	This class holds a single level and counters, it manages the game (runs the level, in charge of creating balls, check when player lose) and decides when do stop running the game (when player lost or won- if he won the GameFlow will immediately create a new GameLevel and start all over again)
KeyPressStoppableAnimation	Class	This class features an animation which stops by pressing a specific key.
MenuAnimation	Class	This class features a menu animation, used for managing tasks and let the user decide which task to operate.
PauseScreen	Class	This class features a pause screen animation object. If the user presses “p” in the middle of the game it operates the pause screen and has to press space to resume.

ExitGameTask	Class	This class features a exit game task, which closes the gui and shuts down the program.
RunGameTask	Class	This class features a run game task, which creates game flow and uses it to run the level information list it gets.
ShowHiScoresTask	Class	This class features a show high scores task, shows the high score table by using high score animation class.
Ass7Game	Class	This class operates a menu, and getting the status of the menu and running it's tasks.

The aliens formation

The alien formation is controlled by **MultAlienColumn** class, which holds a list of columns, a column is an **AlienColumn** which is a column of aliens (an aliens is actually a block).

The **MultAlienColumn** controls the speed and direction of the columns (by checking it's the most right column and most left column) and also checks if the columns reach bottom, after every time the user loses life and still has life, the **MultAlienColumn** class returns all the AlienColumns to the starting point (unless the user hit all blocks in the most left columns- it will move the right columns to the left).

Each **AlienColumn** has a Boolean if it touched the left side or the right side or bottom, and I can easily control their movement from the **MultAlienColumn** class.

Each alien block changes its position by a SetUpperLeft method, which operated by its **AlienColumn**, which is being operated by **MultAlienColumn.**

The shields

Shield is a new class, which extends the **block** class.

The shields are created in the level (when initializing level), and after the level is finished it recreate them.

Each shield (3 of them) contains 60 blocks (each block is in size 5X5).

Shots by aliens

The game level checks if half a second has passed from the last alien ball lunch (by using System.currentTimeMillis() and a private long alienShootBallTime, if so:

The game level generates a random number (from 1 to 10) and picks a column by that number (from **MultAlienColumn** class), if the column has no aliens, it select a number again (repeating the process).

After finding a column with aliens in it, the game level creates the ball that the aliens shoot, the game level finds the lowest alien and lunch the ball from the below center of the alien (and updating

Shots by player

The game level checks if half a 0.35 second has passed from the last user ball lunch (by using System.currentTimeMillis() and private long paddleShotTime, if so:

The game level creates the ball that the user shoots, and lunch it from the center of the paddle.