Assignment 7 Readme

Name: Dmitri Zinkevich

ID: 317301596

Username: zinkevd

Class Changes:

Bullet

Previously the “Ball” class, named changed to be more fitting.

Constructor logic has changed to include a color parameter (red for enemy bullets, blue for friendly ones)

moveOneStep logic simplified since we don’t need to calculate the location of the hit anymore, only that it hit the colllidable somewhere.

The bullet also makes a comparison and when the bullet is red and the target is an alien, it will simply disappear to avoid friendly fire.

Ship

Previously the “Paddle” class, also renamed for a more fitting name.

Constructor logic changed to hold references to the game environment and level the ship was created in (so we can manipulate and use its location)

Added space key to the sensor with a timer that checks whether 0.35 seconds have past since the last shot. First time it checks for the epoch time set on the constructor, and then on the following iterations it evaluates by changing the epoch time to the current time and checking the time difference in milliseconds again.

Added the “shoot” method which shoots bullets towards our targets in a straight line and has a 10% chance to shoot two additional bullets to the sides.

The ship also holds an Image field that lets us display the ship as an image rather than a simple block. (It is extracted using the ExtractImage class from the “ship.png” file).

ExtractImage

New class that extracts an image from a passed on file name through its constructor, and returns it via a getter to the class that created it.

AlienGroup

New class that is in charge of the logic of the aliens, their formation, their shooting capabilities, their movement, creating them and resetting their positions. In depth explanation in the following section.

EndScreen, MenuAnimation, PauseScreen, HighScoresAnimation

All have been changed and received a graphic upgrade, consisting of images and text enhancements

GameLevel

Receives more responsibilities since we're only dealing with one looping level. In charge of resetting and initializing the level and its components and running them.

Specific Implementations:

A. Aliens Formation

The formation is implemented using the AlienGroup class, which holds a list within a list to arrange the aliens into columns (List of Columns => each Column holds a list of Aliens).

It creates the aliens, adds their listeners and adds them to the game regularly, but also creates the aforementioned list of lists.

This allows us to iterate through each column separately, and when we invoke the timePassed method (which is in charge of moving the aliens), if our current alien speed is positive, we'll scan the columns from the right most, meaning the end of the list, to the first column, and vice-versa when our dx is negative.

If the first column that has a living alien hits the border, we'll iterate through the aliens in all columns, multiply their speed by -1 and move them downwards 10 pixels. Otherwise we simply move them according to their speed and dt.

The timePassed method then checks, with the same logic as the Ship shooting explained previously, if 0.5 seconds have passed. If they have, it picks a random column using java's random package and scans its aliens until it finds the lowest one and makes him shoot a bullet.

B. The Shields

The implementation is very simple. We create three clusters of blocks in three different locations in the GameLevel class when it is initializing the level (about one third of the screen's width in x difference).

The clusters we'll be created from simple 12x5 blocks, 3 blocks long and 10 blocks wide. We'll add the usual blockRemover and ballRemover listeners (to remove the block hit and the hitting bullet). Since the player can also hit the shields there's no additional special logic needed.

C. Shots by Player

As previously briefly explained, the player shoots from his ship in intervals of minimum 0.35 seconds. We check that by comparing epoch times to current system times.

The shot itself is created on the spot in the shoot method by creating a new Bullet, giving it velocity and adding it to the game.

I also added a small feature that allows the players ship to sometime shoot 3 bullets (two from each side, 45 and 315 degrees and one regular upwards) if he rolls a 1 out of 10 chance. (Although it's pretty mundane considering the shields usually block the diagonal bullets anyway, but sometimes it does help the player).