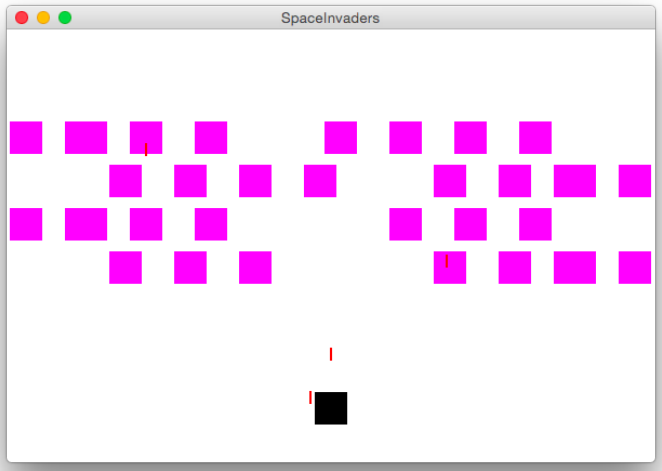
Project 2: Space Invaders (Due Nov. 23 at midnight)

Note: You may work in a group of up to three students for this project, but there must be someone who you have not worked with in the group. Every group (triples, pairs, or individuals) needs to send me an email saying who they are working with.

Project Description



Space Invaders is an old arcade game about destroying alien spaceships. In the game, you control a lone projectile-shooting spaceship, and must destroy an enemy fleet before they overrun your position or shoot you down. Many variations of the game exist; you can play a Flash clone of the original here, and a simplified JavaScript version here.

This project asks you to create your own version of Space Invaders. You have a lot of creative freedom in how the game works, but your game must do the following:

- The player must control a ship that moves left and right at the bottom of the screen, that can fire projectiles at the enemy.
- There must be a grid of aliens that move left and right, and that moves down towards the player over time.
- The aliens must randomly fire projectiles at the player.
- If a player-fired projectile touches an alien, the alien must be removed.
- If an alien-fired projectile touches the player, the player loses.
- If an alien touches the player, the player loses.
- If an alien gets past the bottom of the screen, the player loses.
- If all aliens are destroyed, the player wins.
- There must be end screens when the player wins and loses.

Outside of these requirements, you can do whatever you want. Are you controlling an X-wing to shoot down TIE fighters? Controlling EVE to shoot down AUTOs? You could draw that. Should there be a boss fight? Levels? You can try coding all these possibilities once you are done with the basic game.

Code Walkthrough

The project has two starter classes, and you will need to create more to represent the player, the aliens, etc. The two starter classes are:

- `GraphicsObject` is almost exactly the same class as the one you used for the animation homework. The only difference is that we will use doubles for the position and speed, to get more precision. You will need to convert (*cast*) the doubles to ints when drawing.

- `SpaceInvaders` is based off the `Animation` class. I have made some changes to get rid of the flickering, but the general idea is the same. All of the following components of the game are marked with `FIXME`s in the code.

    - Instead of having a generic ArrayList of GraphicsObjects, you will directly have member variables for the player, the aliens, etc. You must create these member variables yourself and initialize them in the constructor.

    - You must also write the code that updates each member variable in `update()`.

    - Instead of a `paint()` method, I have created three separate methods, `paintGameScreen()`, `paintWinScreen()`, and `paintLoseScreen()`. You must write the code for these methods, which will paint the screen during normal gameplay, when the player has won, and when the player has lost, respectively.

    - There are two new methods `hasWonGame()` and `hasLostGame()`. You must write the code for these methods, which will determine if the game should end. I use the return values from these methods to decide which paint method to call.

    - Finally, there are three more new methods: `keyPressed()`, `keyTyped()`, and `keyReleased()`. These functions allow the program to respond to key presses. The only one you need to write code for is `keyPressed()`; the other two can be left empty. I have created some code that detects the left and right arrows and the space bar, but you will have to write the code for what happens.

Hints

To help you get started on a project of this scope, here is a recommendation for what to tackle first:

1. Draw the player
2. Make the player move based on key presses
3. Draw the aliens
4. Make the aliens move
5. Create and draw player-fired projectiles based on key presses
6. Remove aliens when a player-fired project touches them
7. Make aliens randomly fire projectiles
8. Detect a win and show an appropriate end screen
9. Detect a loss and show an appropriate end screen

Submission

When you are done, fill out the evaluation and email all your .java files to me.