

2hrs.  
don't use magic numbers.

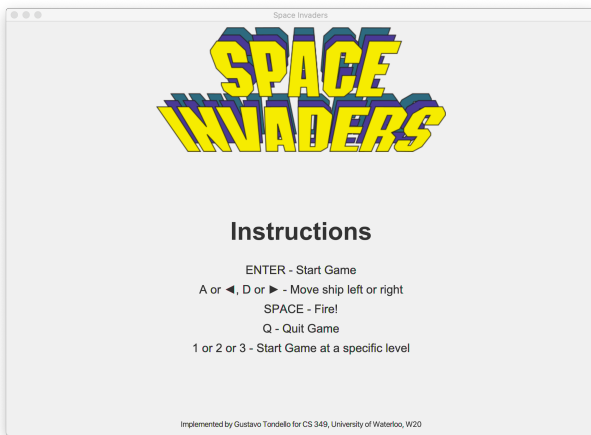
## A2: Event Loops and Drawing (Java) -- Space Invaders

### Synopsis

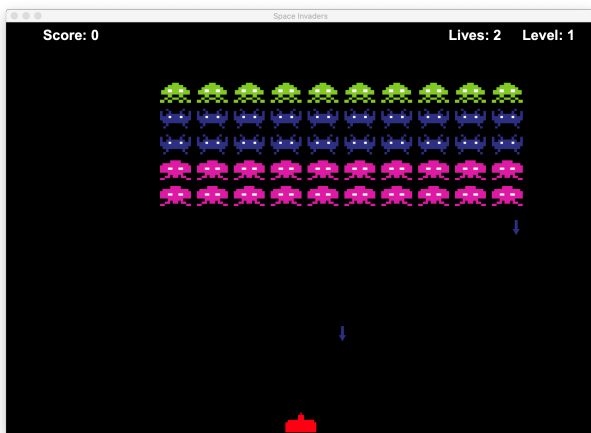
Space Invaders is a simple 2D video game, released by Taito in Japan in 1978 and the US in 1980. It was *hugely popular*, and helped to launch the commercial video game industry. The object of the game is to move a ship along the bottom of the screen and shoot up at aliens that are descending from the sky, without being hit in return. For this assignment, you will implement a fun, playable, version of Space Invaders in Java/Java FX.

For motivation, here's a YouTube video of the original US Atari release (<https://youtu.be/MU4psw3ccUI>) of the game, and a screenshot of the original game screen. (space-invaders-original.png)

We've also made our own version that matches the assignment specification. Here's a video for reference (space-invaders-demo.mp4), and screen shots are below. Your game should closely resemble this!



Title screen



Game screen

### Objectives

- Learn how to build an interactive application in Java.
- Use JavaFX to handle drawing and animation of simple graphics.
- Provide visual and audial feedback, and gameplay elements to make the game more enjoyable.

### Features

#### Starting the Game

When launched, you should display a title screen including (at a minimum):

- a. The title, "Space Invaders", at the top of the screen,
- b. Your name and student number, somewhere on screen,
- c. Directions on how to play the game (e.g. "Use A/D to move left/right and SPACE to fire. Press ENTER to start the game, or Q to Quit").

The player should be able to start or quit the game from this screen.

When the game starts, it should resemble the screenshot above, with standard gameplay elements:

- a. a game screen, including 10 columns and 6 rows of aliens,
- b. a player-controlled ship that can move along the bottom of the screen,
- c. a high-score indicator,
- d. a level indicator to show which level is in-play, and
- e. an indicator of the number of ships remaining.

Status information (high score, level, number of ships) should be visible during gameplay, without obscuring the board.

## The Aliens

The alien ships move together as a group. They initially move from their starting position towards the right-side of the screen. When they reach the edge of the screen, the following happens:

1. The ships all descend one row,
2. One of the ships fires at the player,
3. The ships reverse their direction and start moving in the opposite direction.

The ships continue moving until they reach the left-edge of the screen, and the pattern repeats and they start moving right again. They repeat this pattern, descending and alternating directions, until they reach the bottom of the screen, the player dies, or they are eliminated

Additionally, every time the player destroys an alien, the remaining aliens speed up. Effectively the game starts out fairly slow, and gets progressively faster as aliens are destroyed.

## The Player

The player has three ships at the start of the game: the ship they are using and two additional ships which are indicated on the screen. If the player has ships remaining and their ship is destroyed, you should remove one of the extra ships and respawn the player's ship so that they can continue playing. If they die for a third time (i.e. they've exhausted all ships), then they lose the game.

## Controls

The player interacts with the game by moving the ship left and right to avoid alien missiles, while attempting to return fire: 'A' moves the ship Left, 'D' moves the ship right, and SPACE fires missiles. The player may fire missiles as often as they wish (with the rate of fire not exceeding 2 missiles per second, and its possible for multiple missiles to be in motion). Missiles fly straight up until they strike an alien ship. Missiles will destroy any alien that they strike. Missiles that do not hit an alien, disappear when they reach the top of the screen. The player effectively has unlimited ammunition.

## Levels

The level continues until either all of the aliens, or all of the player's ships are destroyed. If all of the aliens are destroyed successfully, the next level is launched. Subsequent levels all have the same layout, but the ships and missiles move faster than previous levels. The level indicator should clearly indicate the level in play. There does not need to be any other visual changes.

You must have a minimum of three levels, and pressing "1", "2" or "3" from the start screen will launch the game at that level (this is not normal gameplay, but makes it easier for the TA to grade it!).

If the player succeeds in clearing all three levels, you should show a message telling them that they have won the game and display their high score. You should prompt them to restart or quit the game.

If the players ships are all destroyed, you should display a message telling them that they have lost the game, along with their high score. You should prompt them to restart or quit the game.

## Game Features

Your game screen should be no larger than 1600x1200 pixels (1280x800 or a similar resolution is recommended). Your game window should NOT be resizable.

Your game should use audio feedback/sounds for key game elements (e.g. movement of the aliens, when missiles are fired, when an alien ship is destroyed).

You should use images to represent the game elements: you can overlay images directly over Nodes.

You have a number of choices for sound and image assets.

- This Space Invaders fan site has the original image files and sound clips (<http://www.classicgaming.cc/classics/space-invaders/>). You may use these, but you MUST just attribute them in your README please.
- Gustavo modified the originals for use in the demo (above). You are welcome to use our sound and image files. (space-invaders-assets.zip)
- You can use anything that you find on the web IF you know that it is licensed to allow for public use e.g. Creative Commons License. If you do this, you MUST attribute the URL and license in your README file.
- You can create them yourself!

## Hints!

Here are some suggestions to get you started.

- Consider having multiple scenes: one for the introduction screen, one for level 1, one for level 2 and so on. The "SwitchScenes" sample demonstrates how to do this.
- You will want to animate things on the screen so that they move at a smooth and consistent rate (30-60 times per second is a good target rate). Avoid the temptation to have a single while... loop that does this. Instead, use a Java Timer or a JavaFX AnimationTimer to fire periodically and animate things on the screen e.g. moving the aliens, moving the player's ship, drawing a missile. The "SimpleAnimation" and "SimpleAnimationTimer" examples demonstrates how to do this.
- There are a large number of "magic numbers" that you will need to play around with, including player ship speed, alien speed, distance to move and so on. Consider defining constants at the top of your class, so that they're defined in one place. e.g.

```
// Speeds
public static final double PLAYER_SPEED = 3.0;
public static final double PLAYER_BULLET_SPEED = 6.0;
public static final double ENEMY_SPEED = 0.5;
public static final double ENEMY_VERTICAL_SPEED = 10.0;
public static final double ENEMY1_BULLET_SPEED = 4.0;
public static final double ENEMY2_BULLET_SPEED = 5.0;
public static final double ENEMY3_BULLET_SPEED = 6.0;
```

- Use a keyboard handler attached to the scene to capture keyboard input from the user. The "EventHandlers" samples demonstrate how to do this.
- Finally, look into some of the specialized classes in JavaFX for sound and graphics. For example, Media (<https://openjfx.io/javafx/11/javafx.media/javafx/scene/media/package-summary.html>) and Image (<https://openjfx.io/javafx/11/javafx.graphics/javafx/scene/image/package-summary.html>) classes may be useful.

## Technical Requirements

- You must use Java 11 and IntelliJ 2019.3 or later. Java FX 11 may also be used as appropriate.
- You are free to use any sample code or examples provided in class. You may copy or modify this code freely.
- You may also use the image and sounds assets identified above, but you may not use source code from any other source without explicit written permission from the instructor.
- Your main class should be named "SpaceInvaders" and reside in a file named `SpaceInvaders.java`. Running this class should execute your application.

## Submission

Your directory structure for your assignment submission should be in subdirectories under your a2/ directory in your Git repository. It should use a standard file layout for an IntelliJ project. For example, it could resemble this (with possibly different class, image and sound file names of course).

```

a2/
├── a2.iml
├── out/
├── readme.md
├── src/
│   ├── sounds/
│   │   └── fastalien.wav
│   ├── images/
│   │   ├── alien.jpg
│   │   ├── barrier.jpg
│   │   └── ship.jpg
│   ├── Alien.java
│   ├── Ship.java
│   └── SpaceInvaders.java

```

Your submission needs to include:

- All source code and resources required to build and execute your program.
- An IntelliJ project that compiles everything using the specified JDK and Java FX versions, and which has a Run target that will execute your program.
- A `readme.md` file with any details required to help the TA grade your assignment.

Your `readme.md` file needs to include, at a minimum:

- Your name,
- Your student number,
- The version of Java that you used,
- Your operating system,
- The source for your image and sound assets,
- Anything else needed to run your assignment (e.g. undocumented hotkeys).

For example:

```

Jeff Avery
12345678 j2avery
openjdk version "11.0.2" 2019-04-16
macOS 10.14.6 (MacBook Pro 2017)
* Used sound and image files distributed in-class.
* No changes from specification.

```

## Assessment

Your submission will be assessed roughly as follows:

**5%**

Complete submission. Code compiles and runs. Attribution in README file.

**5%**

When the application is launched, the game displays a Start screen containing required information and instructions for the user. The user can launch or quit the game from this screen.

**10%**

When the user starts the game from the Start screen, the game screen displays all required elements (level information, number of ships, game screen with aliens and player ship).

**20%**

Alien ships move across the screen (left-right) and descend periodically to threaten the players ship.

**10%**

Aliens fire missiles that can destroy the player's ship. If the player's ship is struck by a missile, it is destroyed and a new ship is spawned (if ships remain).

**20%**

The player can maneuver their ship using A/D keys and fire using SPACE. Missiles that are fired by the player destroy alien ships when they strike.

**10%**

When the level is cleared, the game progresses to a new level with faster movement. There are at least three levels (and 1/2/3 from the Start screen will launch them).

10%

When the game ends, the appropriate message will be displayed (i.e. "you win" or "you lose") and the player will be prompted to restart or quit the game.

10%

Compelling gameplay that includes graphics and sound effects!

## Versions

1.0 Jan 27, 2020. Initial release.

---

CS 349 User Interfaces (Winter 2020)

Cheriton School of Computer Science

University of Waterloo

- ship 2x - 1x 从中间开始  
- alien - 一直 shoot, high chance of higher level  
- ship fire missiles not longer the 2 / second