

Project 5 - Space Invaders

Students will implement a SNAP version of the classic arcade game *Space Invaders*.

Overview

Originally released in 1978, Space Invaders was one of the modern video games and is often credited with popularizing the video game industry. In addition to being a vastly popular arcade game, the 1980 version for an early Atari system helped game consoles achieve the mainstream status they maintain today. Because of this, the characters and gameplay style of Space Invaders are often used to represent the video game industry as a whole. You can play an online version of the game at <http://www.pacxon4u.com/space-invaders/>. Our game will have some differences from the classic version, but will maintain the key aspects of gameplay.

Details

1. Behavior

i. Sprites

Your implementation of Space Invaders will include at least three main sprites: the **player**, **invaders**, and **projectiles**.

- The **player** moves along the bottom of the stage attempting to destroy invaders by firing projectiles. The player can move left and right only (controlled by the arrow keys), not up and down. The player fires when the space bar is pressed. The player begins with three lives and loses a life each time an invader touches him or reaches the bottom of the stage.
- **Invaders** begin at the top of the stage and slowly move towards the bottom in a grid formation. Invaders move from side to side across the stage, bouncing off the left and right edges. Each time an invader bounces off an edge, it should move down by the height of one invader. When an invader is hit by a projectile, it is destroyed, and should disappear from the stage.
- **Projectiles** are fired by both the player and the invaders. The player fires a projectile when the user presses the space bar, while the invaders fire randomly. A projectile should originate at the position of the sprite that fired it and move quickly up or down the stage until hitting either an invader, the player, or the edge of the stage. If the projectile hits an invader or player, that invader or the player should be destroyed. Upon hitting either an edge of the stage or another sprite, the projectile should then disappear. Only one invader may fire a projectile at a time, and another invader may not fire until the projectile has disappeared. The player has an unlimited supply of projectiles, but may only shoot *three* at a time.

ii. Starting the Game

When the green flag is clicked, the game should initialize and a welcome screen should be displayed to the user (think of this as inserting a coin into an arcade machine). The welcome screen should include a “Start Game” button that, when clicked, begins the actual game. The welcome screen should also display some basic information about the game.



iii. **Lives and Game Over**

The player begins the game with three lives, and loses a life each time an invader either touches the player or reaches the bottom of the stage. When the player loses a life, the game should halt momentarily. All invaders and projectiles should disappear, and there should be a visual indication that the player has lost a life (message, change in costume, etc.). After a brief delay, the game should restart with all invaders at the top of the stage and the player having one fewer life. When the player loses its final life, the game ends and a “Game Over” message should be displayed. A “New Game” button should then be displayed that, when click, restarts the game from the beginning.

iv. **Levels**

Each time the player manages to destroy all the invaders on the screen, he or she completes a level. When a level is completed, a “Level Cleared” message should be displayed. Then, after a brief delay, the next level should begin. Each successive level should be made more difficult by increasing the speed at which the invaders advance down the stage and the frequency at which they fire projectiles.

v. **Score**

Each time the player destroys an invader, points should be earned. Scoring should start at 100 points per invader in the first level, and increase by 50 points per invader in each successive level. So, for example, each invader in level 2 is worth 150 points, each in level 3 is worth 200 points, at so on. The player’s score should be displayed on the stage throughout the game and be featured on the “Game Over” screen.

2. Implementation Details

i. **Documentation and Style**

As with all previous projects, your program must be well-written, well-documented, and readable. This includes, but is not limited to:

- organizing your scripts so that they can be read and comprehended easily
- giving your sprites meaningful names
- naming and using your variables, lists, and custom blocks well
- including comments to describe the structure of your program and any particularly complex or unintuitive pieces of code
- separating master sprites from clones, and cleaning up clones that are finished (see below)

ii. **Cloning**

The invaders and the projectiles should be implemented using prototyping. A single master sprite should be implemented for each type and clones should be created each time a new instance of the sprite is needed. The master sprites should be hidden throughout the program and should not take part in gameplay.



Introduction to Computer Science

iii. Required SNAP Elements

Your program must include the following SNAP elements:

- At least three variables
- At least two custom blocks, at least one of which must take an argument

3. Required Checkpoints

- Have the player and a single invader moving correctly
- Be able to clone invaders and have the entire group move correctly
- Be able to fire projectiles from both the player (when the space bar is pressed) and the invaders (randomly)

Grading Scheme/Rubric

Functional Correctness (Behavior)	
Welcome screen containing game info displayed green flag clicked	1 point
Start button displayed and game starts when clicked	1 point
Keyboard control of player (including shooting)	1 point
Multiple invaders	2 points
Invaders attack in formation	2 points
Invaders die when hit by projectile	1 point
Player earns points when an invader is killed	1 point
Player fires projectiles correctly	2 points
Invaders fire projectiles correctly	3 points
Player loses life and level resets when player or bottom of stage touched by invader	2 points
Game ends and new game screen appears when player is out of lives	1 point



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

Functional Correctness (Behavior)	
Level changes when all invaders are destroyed	1 point
Invaders speed up in each level	1 point
Point values increase each level	1 point
<i>Total</i>	<i>20 points</i>
Technical Correctness (Implementation)	
Program is well-documented and shows good style	2 points
Program shows creativity and effort	2 points
Invaders and projectiles are implemented using cloning	3 points
Program operates correctly under all normal conditions (aka “No bugs”)	3 points
Program uses at least three variables	2 points
Program includes two custom blocks, one of which accepts arguments	3 points
Checkpoint 1	5 points
Checkpoint 2	5 points
Checkpoint 3	5 points
<i>Total</i>	<i>30 points</i>
<i>Total</i>	<i>50 points</i>

