

Created by Tomohiro Nishikado GDD written by Dylan Rampersad

**Table of Contents**

* Game Concept
* Characters
* Obstacles
* Javascript

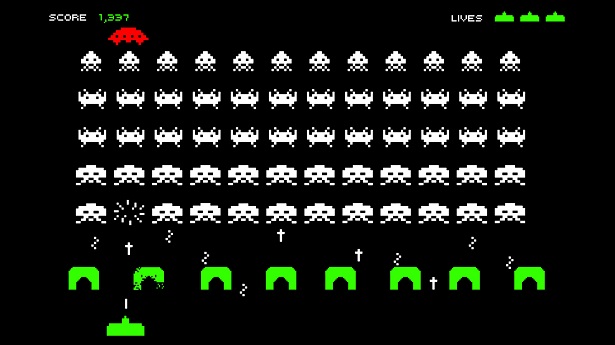
**Game concept**

Space Invaders has you as Earth’s only defense against an alien invasion.

Fighting through waves of enemy aliens your goal is to destroy all the aliens before they get to you and avoid the lasers being shot at you.

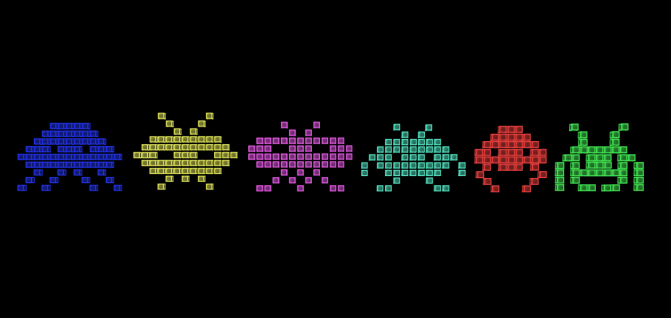
The speed at which the aliens move and shoot is based on how far in a level you progress. The more you destroy the faster they become.

There will be five rows of eleven aliens moving side by side. When the reach the far ends they move down and toward your ship.

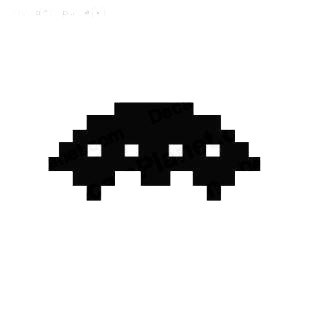


**Characters**

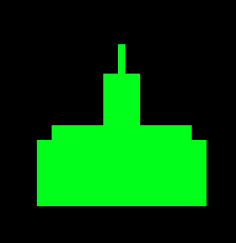
The standard enemies in the game move simultaneously and fire at you.



The alien space ships move side to side and have a higher rate of fire.

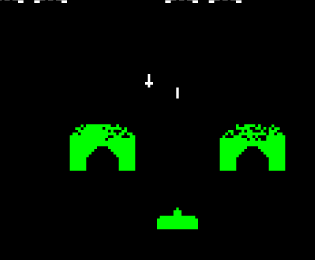


The ship you operate in the game can move only side to side and fire at the enemies.



**Obstacles**

The only obstacles in the game are the walls that can block incoming shots however they also block outgoing shots therefore the player cannot shoot through one however they can be destroyed overtime.



**Javascript**

Javascript will be used as the language for this game.

public class SpaceInvaders extends JFrame implements Commons {

public SpaceInvaders() {

initUI();

}

private void initUI() {

add(new Board());

setTitle("Space Invaders");

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setSize(BOARD\_WIDTH, BOARD\_HEIGHT);

setLocationRelativeTo(null);

setResizable(false);

}

public static void main(String[] args) {

EventQueue.invokeLater(() -> {

SpaceInvaders ex = new SpaceInvaders();

ex.setVisible(true);

});

}

}

This will be used for the title screen.

public class Player extends Sprite implements Commons {

private final int START\_Y = 280;

private final int START\_X = 270;

private final String playerImg = "src/images/player.png";

private int width;

public Player() {

initPlayer();

}

private void initPlayer() {

ImageIcon ii = new ImageIcon(playerImg);

width = ii.getImage().getWidth(null);

setImage(ii.getImage());

setX(START\_X);

setY(START\_Y);

}

public void act() {

x += dx;

if (x <= 2) {

x = 2;

}

if (x >= BOARD\_WIDTH - 2 \* width) {

x = BOARD\_WIDTH - 2 \* width;

}

}

public void keyPressed(KeyEvent e) {

int key = e.getKeyCode();

if (key == KeyEvent.VK\_LEFT) {

dx = -2;

}

if (key == KeyEvent.VK\_RIGHT) {

dx = 2;

}

}

public void keyReleased(KeyEvent e) {

int key = e.getKeyCode();

if (key == KeyEvent.VK\_LEFT) {

dx = 0;

}

if (key == KeyEvent.VK\_RIGHT) {

dx = 0;

}

}

}

This will represent the player’s ship and how it responds to controls.

import javax.swing.ImageIcon;

public class Alien extends Sprite {

private Bomb bomb;

private final String alienImg = "src/images/alien.png";

public Alien(int x, int y) {

initAlien(x, y);

}

private void initAlien(int x, int y) {

this.x = x;

this.y = y;

bomb = new Bomb(x, y);

ImageIcon ii = new ImageIcon(alienImg);

setImage(ii.getImage());

}

public void act(int direction) {

this.x += direction;

}

public Bomb getBomb() {

return bomb;

}

public class Bomb extends Sprite {

private final String bombImg = "src/images/bomb.png";

private boolean destroyed;

public Bomb(int x, int y) {

initBomb(x, y);

}

private void initBomb(int x, int y) {

setDestroyed(true);

this.x = x;

this.y = y;

ImageIcon ii = new ImageIcon(bombImg);

setImage(ii.getImage());

}

public void setDestroyed(boolean destroyed) {

this.destroyed = destroyed;

}

public boolean isDestroyed() {

return destroyed;

}

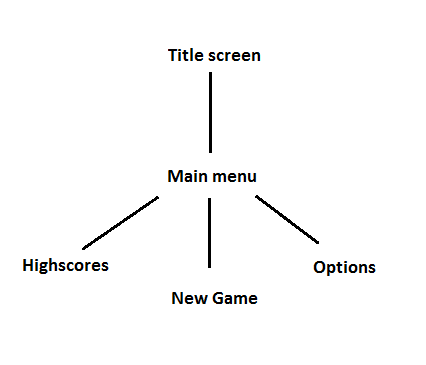
}

}

This code represents the Aliens and how they act in game.

**UML**

The UML will follow a standard format going from the title screen to the main menu to the game itself.



By: Dylan Kishan Rampersad