

# **READ ME**

## **Game Play**

1. Use the left arrow key to move the ship to the left.
2. Use the right arrow key to move the ship to the right.
3. Use the space bar to fire a laser.
  - You can only fire one laser at a time (the laser will not be fired until the previous laser has been destroyed).
4. Each alien that is killed will add 100 points to the score.
5. If you hit a missile with your laser, then the laser and that missile are removed. 1,000 points will be added to your score.
6. If you kill all of the aliens on the screen, you will advance to the next level.
7. Aliens and missiles move faster than the previous level.
8. The game is over if the Aliens reach the bottom of the playing area (top of the shields).
9. There are 5 rows of aliens each level, but higher levels have more columns of aliens. Each level, the aliens and missiles travel faster. Every odd level will add one life.
  - Level 1: 5 columns and 600 millisecond intervals. Start with 3 lives.
  - Level 2: 6 columns and 550 millisecond intervals.
  - Level 3: 6 columns and 500 millisecond intervals. Add 1 life.
  - Level 4: 7 columns and 450 millisecond intervals.
  - Level 5: 7 columns and 400 millisecond intervals. Add 1 life.
  - Level 6: 8 columns and 350 millisecond intervals.
  - Level 7: 8 columns and 300 millisecond intervals. Add 1 life.
  - Level 8: 9 columns and 250 millisecond intervals.
  - Level 9: 9 columns and 200 millisecond intervals. Add 1 life.
  - Level 10: 10 columns and 150 millisecond intervals.
10. To win the game, you must kill all of the aliens in level 10.
11. If you win the game, each remaining life you have will add 10,000 points.

## **Description of Program Coding**

1. Interfaces
  - a. `GameData`: This interface holds values for constants for easy use.
  - b. `Object2D`: This interface holds common methods for game objects.
  - c. `Shooter`: This interface holds a fire method for objects that can shoot.
2. GUI Component Classes
  - a. `InvadersGameFrame`: This `JFrame` holds all of the `JPanels` necessary for the game to show up properly. It has the `Timer` for the interval of updating the graphics.
  - b. `Stats`: This `JPanel` holds 3 `JPanels` each consisting of 2 `JLabels`.
    - i. `LivesPanel`: An inner class for displaying number of lives.
    - ii. `AlienPanel`: An inner class for displaying number of aliens killed.
    - iii. `ScorePanel`: An inner class for displaying the score.
  - c. `GameBoard`: This `JPanel` draws all of the `GameObjects` for the game within the game's board. This class implements the `KeyListener` for controlling the ship's movements.

## **READ ME**

- d. **Button:** This `JPanel` contains 3 panels, one for displaying the title, “Space Invaders,” one for the button to start/pause/resume the game, and one to display the level.
3. **GameLogic:** All of the game logic is in the `GameLogic` class. This class is responsible for updating the aliens’ position, missiles’ position, and laser’s position. This class also checks for intersections between objects. This includes: laser vs. aliens; missiles vs. ship; and laser vs. missiles.
4. **GameObject:** This abstract class has the implemented methods for game objects.
  - a. **Alien:** This class extends `GameObject` and has the `BufferedImages` of the aliens to draw. Since the alien must be able to shoot, it implements the `Shooter` interface. Within the method for updating position, it checks to see if it has reached the bottom of the aliens’ space.
  - b. **Ship:** This class extends `GameObject` and has the `BufferedImage` of the ship to draw. Since the ship must be able to shoot, it implements the `Shooter` interface.
  - c. **Shield:** This class extends `GameObject`. It has a health field to keep track of how many times it has been damaged. When it has 0 health, the shield is removed.
  - d. **Laser:** This class extends `GameObject` and has a method to update its position while checking to see if it intersects with a shield.
  - e. **Missile:** This class extends `GameObject` and has a method to update its position while checking to see if it intersects with a shield.

### **Extras**

1. As a little hint/cheat, you can press the ‘m’ key at any point during the game to clear all of the aliens’ missiles from the screen.
2. Aliens shoot missiles at the ship if the center of the alien is aiming at the ship.
3. Only the bottom-most alien shoots the missiles.
4. I have loaded an LCD font to be used by the `JLabels`.

### **Known Bugs**

1. There seems to be a malfunction with the lasers shooting after some time. If the ship stops shooting lasers when the space bar is pressed, then the `lasers ArrayList` has somehow gotten more than one laser in it. To fix this problem, just hit the ‘c’ key to clear the lasers. The ship should be able to shoot lasers again.