

## System Requirements Specification for Space Invader Game

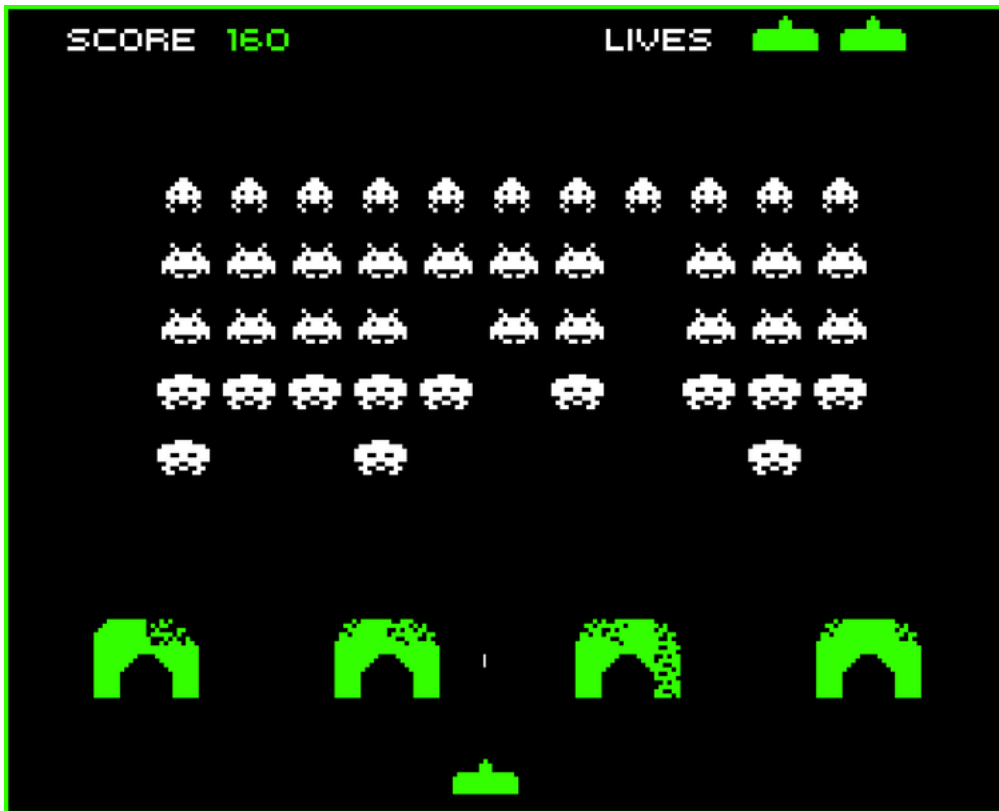
### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Basic Software Expectations</b>	<b>2</b>
BSR.1	.....	2
BSR.1.1	.....	2
BSR.2.2	.....	3
BSR.2	.....	3
BSR.2.1	.....	3
<b>3</b>	<b>Game Expectations</b>	<b>3</b>
GR.1	.....	3
GR.1.1	.....	3
GR.1.2	.....	3
GR.1.3	.....	3
GR.1.4	.....	4
GR.2	.....	4
GR.2.1	.....	4
GR.2.2	.....	4
GR.2.3	.....	4
GR.2.4	.....	4
GR.2.5	.....	7
GR.2.6	.....	7
GR.3	.....	8
GR.3.1	.....	8
GR.3.2	.....	10
GR.3.3	.....	10
GR.3.4	.....	11
GR.3.5	.....	12
GR.4	.....	12
GR.4.1	.....	12
GR.4.2	.....	12
GR.5	.....	12
GR.5.1	.....	12
GR.5.2	.....	13
GR.6	.....	13
GR.6.1	.....	13
GR.6.2	.....	13
GR.6.3	.....	13
GR.6.4	.....	13

GR.6.5 . . . . .	13
------------------	----

# 1 Introduction

This SyRS is written for the CISC 3140 semester project at Brooklyn College. This document is a work-in-progress, so nothing is concrete. The game is inspired by the famous game *Space Invaders*<sup>1</sup>. Here is what the game looks like for those that are not familiar:



To add our own twist on the game, instead of the aliens being enemies we will have City University of New York's Deans and Trustees, the spaceship will be the student, and the space bases will be the Professors, so the Brooklyn College professors will shield the students from the evil Deans and Trustees. A more detailed description will be given in later sections.

## 2 Basic Software Expectations

Here we list requirements the Software must meet.

### BSR.1

The game should run smoothly (without any blatant bugs/software issues due to the developer) provided that the user has the required hardware to run the game.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Space\\_Invaders](https://en.wikipedia.org/wiki/Space_Invaders)

#### **BSR.1.1**

If the user's hardware is capable, the game should be able to run at 60 fps or higher.

#### **BSR.2.2**

The game should be able to read the user's inputs without too much of a delay. For example, if I move the spaceship to the right, there should be no noticeable delay.

### **BSR.2**

The game should run on multiple browsers/hardware.

#### **BSR.2.1**

Game must run on PC versions of Google Chrome, Firefox, Safari, and Internet Explorer.

## **3 Game Expectations**

As mentioned in the introduction, the game is based on Space Invaders. The game should play and behave just like the original Space Invaders. We will now list some requirements the game must meet.

### **GR.1**

Like the original game, there must be a 'start' screen that is displayed to the user upon loading the game.

#### **GR.1.1**

The start screen will display the name of the game. We have not yet decided on the title for the game, but it will be displayed in big bold font.

#### **GR.1.2**

Below the game title, the start screen should show the different types of aliens and the number of points each alien rewards when it dies. In our twist there will be no aliens—instead, there will be deans and trustees. We have not yet decided which dean or trustee will be displayed. Perhaps we can generate them randomly when the game loads, or we can allow the user to choose the deans and trustees they want to fight.

### GR.1.3

Below the deans and trustees there will be a text saying **PRESS ANY KEY TO BEGIN** telling the user that the game needs to be started by providing an input. We will want this text to be dynamic—that is, it will grow and shrink to get the user’s attention.

### GR.1.4

Finally, below the flashing text we will have the default keys and their function. We will have a picture of the arrow keys along with an equal sign saying that the keys are used for movement, and a picture of the space bar with an equal sign followed by “shoot”, telling the user that the space bar is used to shoot.

## GR.2

Once the user provides an input on the start screen, the main game will begin. We will now list some expectations of the main game.

### GR.2.1

When the game begins, we will show a short cinematic. The cinematic will just be a dialogue box that says “The evil Deans and Trustees of CUNY have come to attack the students. Help defend the students from the Dean and Trustee barrage!”. The text will not be displayed all at once. It will be displayed one word at a time. The rate at which each word is displayed has not yet been decided. We will then show the 4 evil Deans and Trustees from [GR.1.2](#).

### GR.2.2

Once the cinematic has finished, we will officially begin the game. We will now discuss the elements of the game. The game will be divided into three areas. See [Game Rectangle](#) on next page. We will now draw the game using abstract rectangles. Please note that these rectangles are an abstraction used to help think of the game.

### GR.2.3

The blue rectangle, the rectangle on top, will display the player’s score and the number of lives the player has. In the left corner of this rectangle will be the player’s score. It will be displayed exactly like the picture in the [Introduction](#). Similarly, the right corner of this rectangle should display the number of lives the player has.



Figure 1: Game Rectangle

#### **GR.2.4**

The red rectangle, the second rectangle, will contain the enemies. See [Enemy Rectangle](#) for more detail.

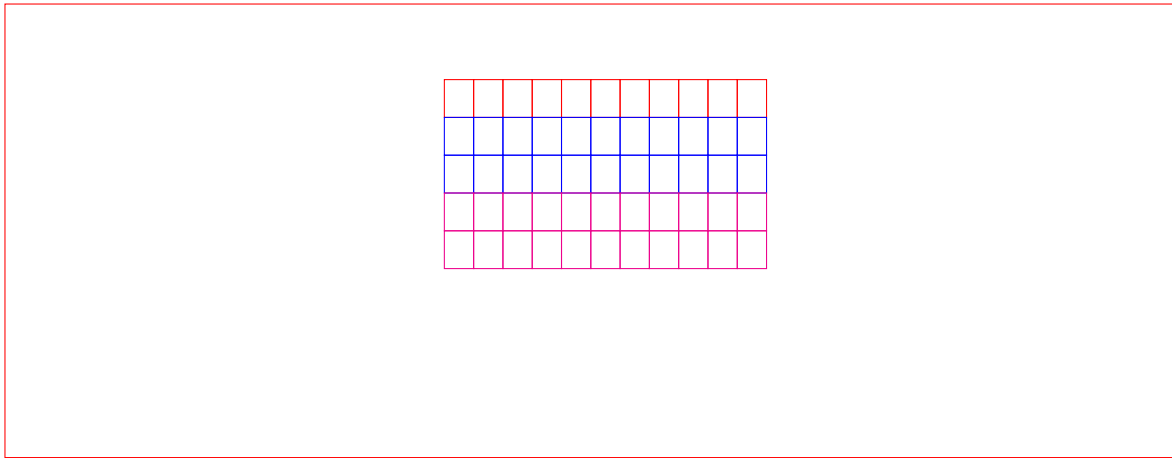


Figure 2: Enemy Rectangle

As you can see by the different colors of the small rectangles, there will be 3 types of enemies as mentioned in [GR.1.2](#). Killing the enemy denoted by the small red rectangle will reward 40 points, killing the enemy denoted by the small blue rectangles will reward 20 points, and killing the enemy denoted by the small magenta rectangles will reward 10 points. The enemies will move across the red rectangle until they reach the end of the rectangle. When the aliens reach the following position:

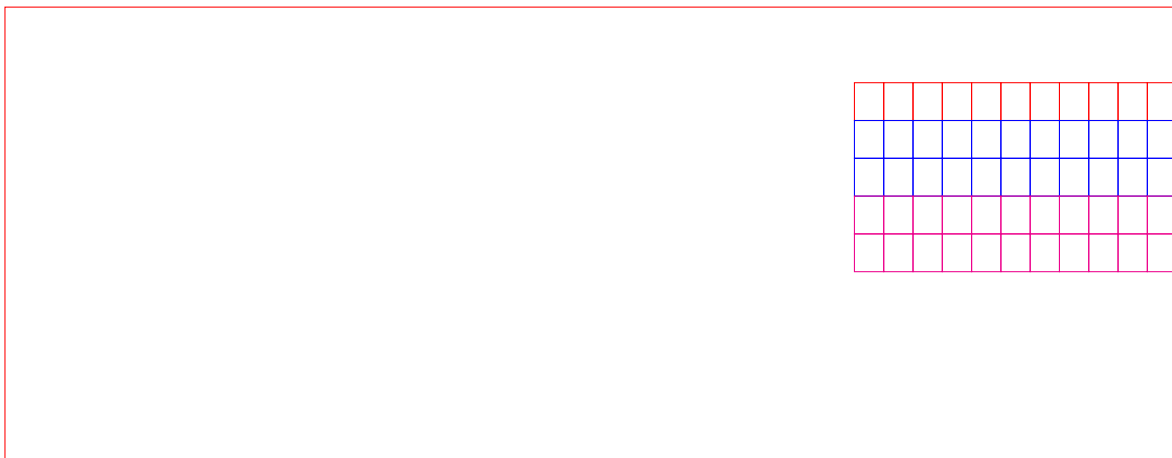


Figure 3: Enemies Reach Border

The enemies will move down vertically:



Figure 4: Enemies Move Down

Then the enemies will continue moving horizontally to the left until they reach a position analogous to **Enemies Reach Border** in which case the enemies will move down vertically analogously to **Enemies Move Down**. This process will repeat until the enemies reach the green rectangle from **Game Rectangle**; if an enemy is in the green rectangle, the game ends and the player loses.

### GR.2.5

Now we discuss the speed at which the aliens move. The aliens will begin the game by moving once every second. The number of times the aliens move per second will scale with time. For every second that passes, the number of times the aliens move per second will be increase by 0.1. For example, say that 10 seconds have elapsed since the game started: the aliens should move twice every second, and if 25 seconds have elapsed, the aliens should move 3.5 times every second—but since 3.5 movements per second does not make sense, we will floor the number to get 3 movements per second.

### GR.2.6

Finally, we discuss the player rectangle. The player rectangle should have the following form:



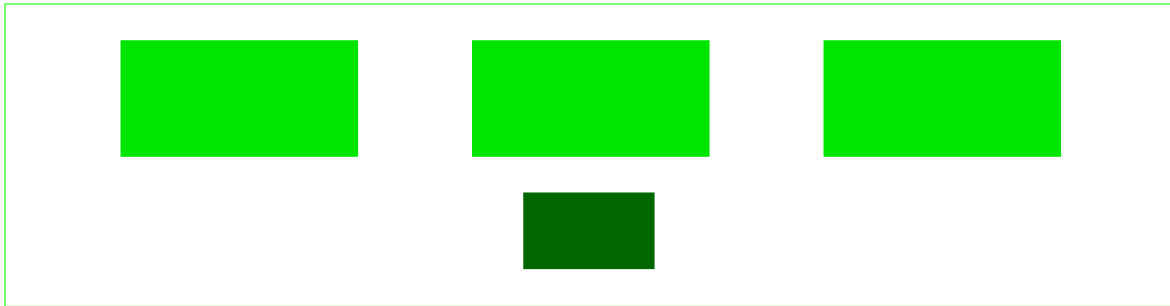


Figure 5: Player Rectangle

The three green rectangles on top represent our allies that will shield us from the enemies assault; they are static and do not move, they do however take damage—this will be discussed in detail later. The dark green square represents the player. The player is not static and can move left and right; player movements and actions will be discussed in detail later.

## GR.3

In this section we discuss the ability for red enemies, enemies furthest from the player, and the player to shoot.

### GR.3.1

We begin by discussing enemy behavior. As discussed in [GR.2.4](#), the enemy furthest from the player, which we denoted by a red rectangle in [Enemy Rectangle](#), is the only enemy that can shoot. We discuss here what it means for an enemy to shoot. When an enemy shoots, this is what we expect:

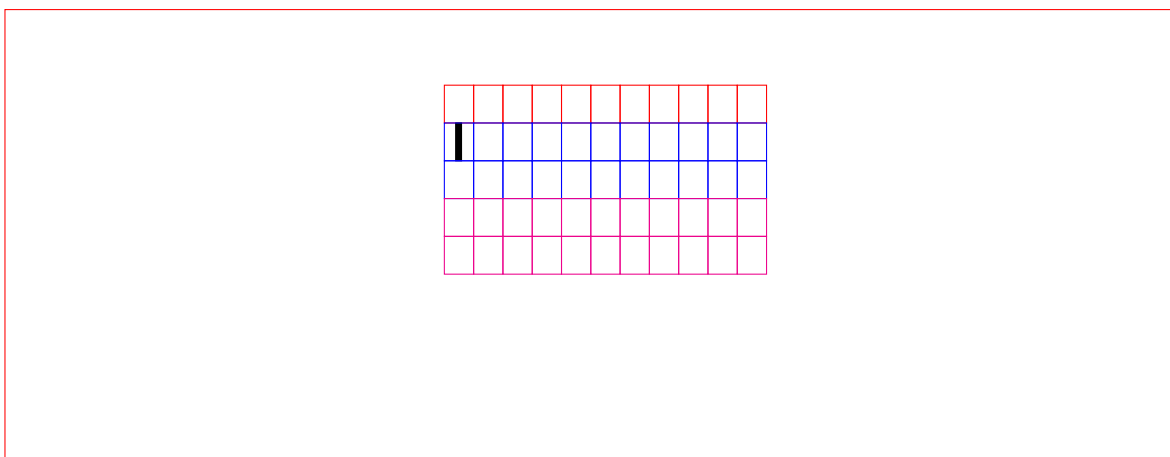
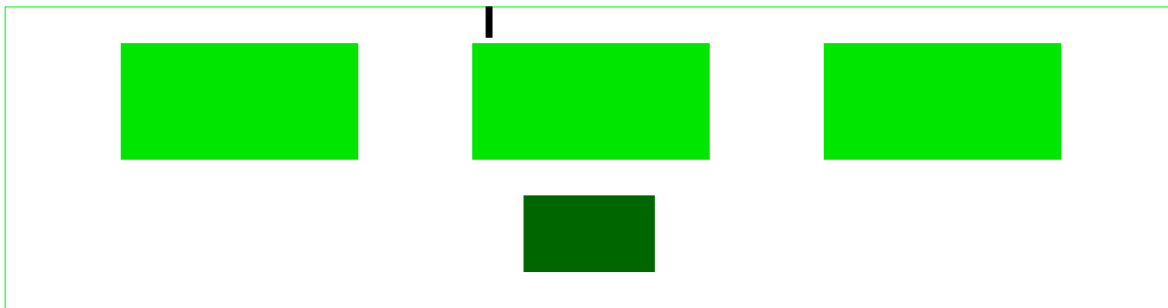


Figure 6: Enemy Shooting

We expect a projectile to emanate from one of the red enemies, the enemies that can shoot. As the projectile leaves the enemy, it should travel in a straight line downwards until it collides with something or if it leaves the **Game Rectangle**. If it collides with something it should disappear. If the projectile collides with one of the player's allies, a portion of the ally should disappear. Here's what we expect to happen to the player's ally if a projectile collides with it:

Before Collision:



After Collision:

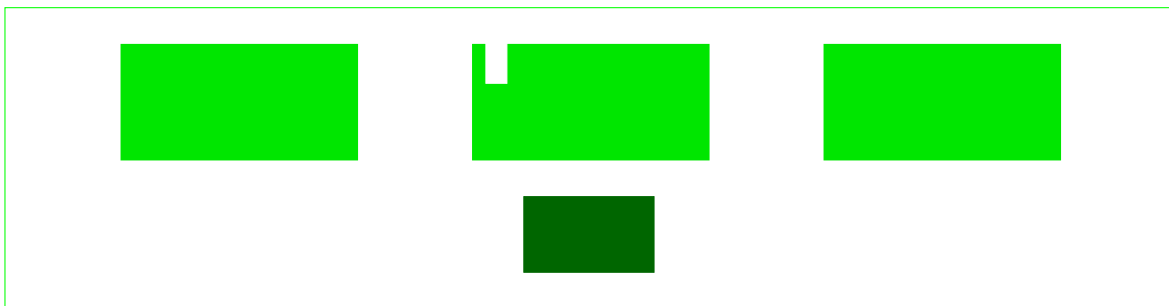
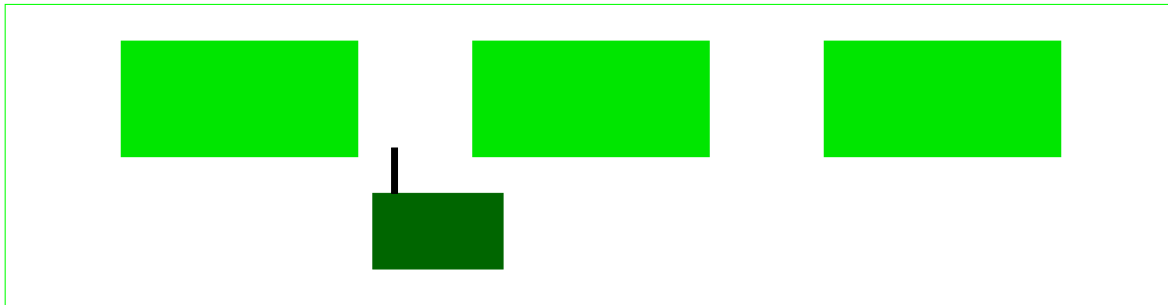


Figure 7: Ally Collision

The above figure depicts what happens when an ally is hit with an enemy's projectile. There are two main takeaways from this figure: firstly, the area where the projectile hits the ally disappears, and secondly the impact area is larger than the projectile. Now let us consider the case where the player is hit with the projectile:

Before Collision:



After Collision:

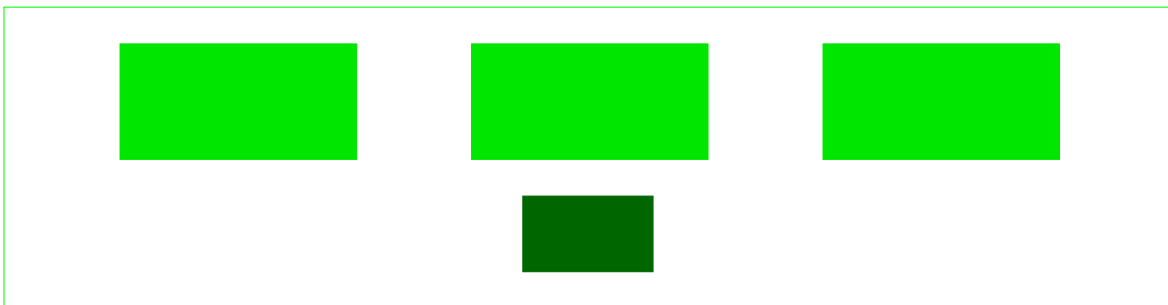


Figure 8: Player Collision

Here is what's happening in the figure: when the player is hit with the projectile, they die and lose a life. Since they die, they must re-spawn. Hence, their position is reset to the default position.

### GR.3.2

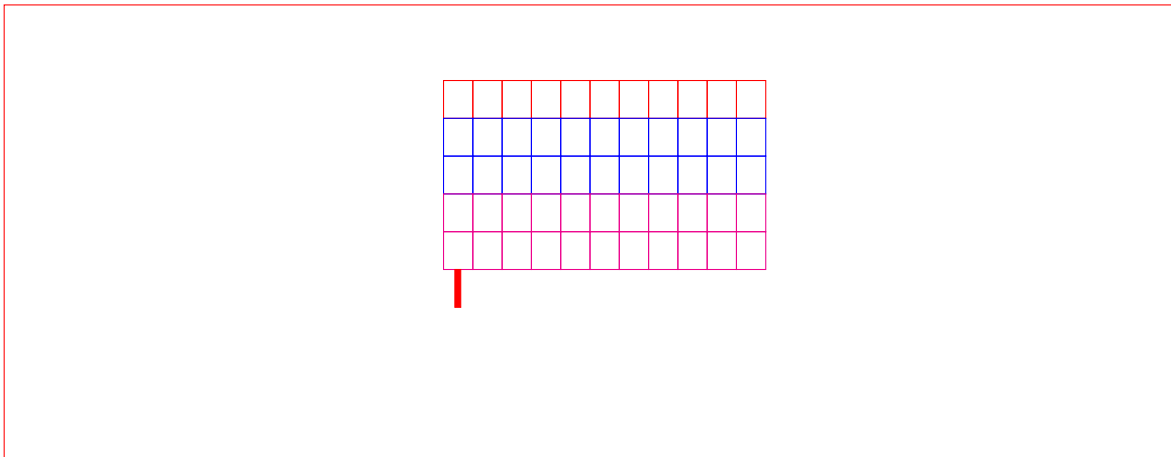
Now we discuss the details of the enemy missile. As discussed earlier, only the red enemies, the enemies furthest from the player are allowed to shoot. The red enemy which shoots will be chosen at random. At the beginning of the game, the number of red enemies which can shoot at a time is 1. The red enemies will shoot every 5 seconds. As time progresses, the number of red enemies that can shoot at a time will increase; up to a maximum of 9. Every 10 seconds the number of red enemies that can shoot will increase by 1. The enemy missile speed will stay constant throughout the game. Missile speed has not yet been decided.

### GR.3.3

The player has the ability to shoot as well. The player shoots a projectile similar to the enemy's projectile discussed earlier. The player's projectile can only impact the enemies. The projectile emanates from the player and is identical to the enemy's

projectile. The projectile moves in a straight line and goes upward, away from the player. If the projectile does not collide with anything, it will disappear as soon as it enters the blue rectangle from **Game Rectangle**. If the player's projectile hits one of its allies, nothing will happen: the projectile will just disappear. Now if the projectile hits the enemy, the enemy should disappear. Here is what we expect to happen if an enemy is hit with the player's projectile:

Before Collision:



After Collision:

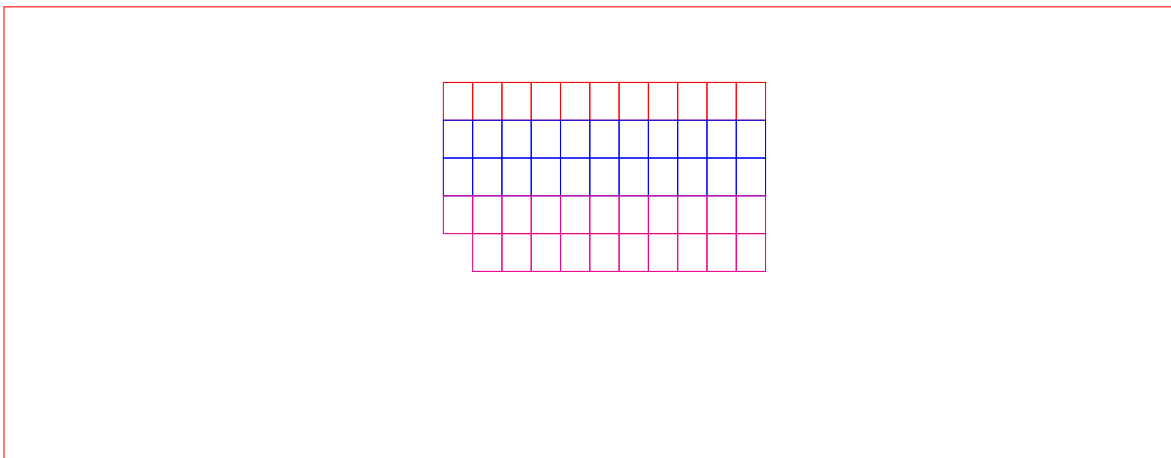


Figure 9: Player Shooting

It should be obvious what the player's projectile does to an enemy. It simply destroys the enemy hit. When an enemy is killed there should be a large explosion that leaves a bloody mess. The enemy should also yell out an obnoxious screech as it dies.

### GR.3.4

The player projectile will be the same speed as the enemy's projectile speed, and will be constant throughout the game. After the player shoots, the player must wait 0.5 seconds for the projectile to recharge. When the projectile has recharged, the player is free to shoot once again.

### GR.3.4

As mentioned in [GR.2.1](#) there are 4 aliens, but we have only mentioned 3 so far. The fourth alien is a bonus alien that randomly spawns throughout the game. The alien awards a large amount of points if it is killed. The alien appears directly below the user-interface rectangle. The alien spawns randomly either on the left or on the right, and continues to move horizontally across the screen. When the alien leaves the screen, it disappears and is no longer part of the game.

## GR.4

There are levels in this game. We discuss them here.

### GR.4.1

The player progresses to a new level if they destroy all of the enemies. The player is moved to the next level as soon as the last enemy on that level is destroyed. When the player moves to a new level, the player's lives and their allies are reset—that is, the player's lives are set to 3 and the player's allies are restored to their original state without any destruction due to enemy missiles. The player's position is also reset to the position in [Player Rectangle](#). The enemies are also restored and start in their default position from [Enemy Rectangle](#). The number of enemies that can shoot at a time and the number of times the enemies can move per second does not reset. They continue to increase as described by their respective rules. Even though it should be obvious, the player's score is not reset. However, if the player is waiting for their missile to recharge, they start out with their missile recharged—that is, they can shoot as soon as the level begins.

### GR.4.2

[GR.4.1](#) repeats until the player loses. Conditions for losing are discussed in the next section.

## GR.5

Conditions for the game ending.

### **GR.5.1**

The game ends if the player gets hit by an enemy projectile while having 0 lives.

### **GR.5.2**

The game ends if an enemy reaches the player rectangle from **Game Rectangle**.

## **GR.6**

The game should have a “game over” screen.

### **GR.6.1**

The game over screen should show the player’s score.

### **GR.6.2**

The game over screen should show the player’s level reached.

### **GR.6.3**

The game over screen should show the number of enemies killed.

### **GR.6.4**

The game over screen should have the text that says “Press any key to play again” or “Click anywhere to play again” depending on the device, just like **GR.1.3**.

### **GR.6.5**

In total, the game over screen should look something like:

Game Over!  
Score: x  
Level: y  
Enemies Killed: z  
Press any key to play again!