

# Lab 3 - Taylor Cowley and Andrew Okazaki

---

## Chapter 1: Space Invaders Overview

- Section 1.1: Space Invaders History
  - Section 1.2: Game Play
  - Section 1.3: Game Details and Specifications
- 

### Section 1.1: Space Invaders History

Two good-sized paragraphs are required for full credit.

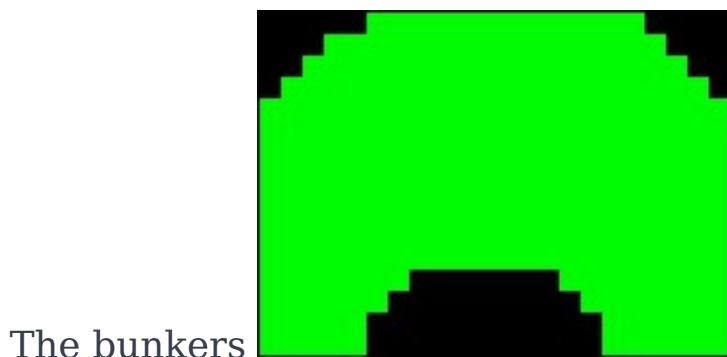
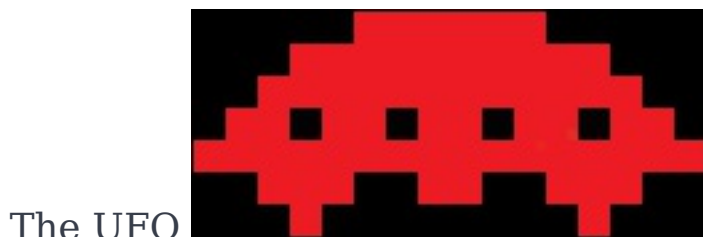
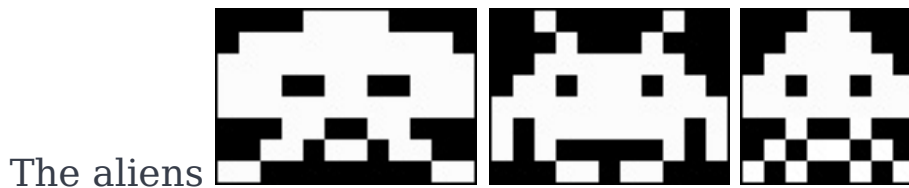
Space Invaders is a video game, developed by Tomohiro Nishikado, released in 1978 by Taito. Development was hindered by lack of technology at the time- Nishikado was required to design his own arcade board hardware capable of running it. Using an Intel 8080 CPU and a Texas Instruments sound chip, he was still reportedly disappointed in the limitations.

Released as an arcade machine in Japan and the United States, Space Invaders became the best-selling video game at the time, and is still the second best-selling arcade game of all time, bested only by Pac-Man. Space Invaders is credited with expanding video games into a real industry, and inspiring future video-game legends, such as Shigeru

Miyamoto (Nintendo, Mario) and Hideo Kojima (Castlevania, Metal Gear).

---

## Section 1.2: Game Play



The objective of space invaders is earn the most points by destroying aliens.

To destroy the aliens the user will control a tank which can move left and right as well as shoot. There is only one tank bullet aloud on the screen at one time. If the bullet hits an alien the alien and the bullet from the tank will be destroyed. The tank only has three lives, if the

user spends all three lives the game will be over.

The tank will lose a life if it is hit with a bullet from the aliens. There are three different types of aliens with a total of fifty five aliens on screen. However there is only allowed to be a maximum of four alien bullets on screen at one time. The bullets will fire at random from random aliens. The tank Needs to destroy all of the aliens in a allotted time period becasue as the aliens move on screen and reach the ends of the screen the aliens will shift forward towards the tank. If the tank fails to destroy the aliens it will become over run and the usser will lose the game.

The tank has the option of hidding from the aliens behind bunkers. There are 4 bunkers on screen. These bunker will degrade if they are hit by a bullet or come into contact with an alien.

To add another level of complexity there is an alien saucer that will cross the screen at random intervals behind the aliens. If the user is able to hit the saucer they will be able to earn bonus point.

If you are able to destroy all of the alien the aliens, a new wave of aliens will be generated. However the bunkers that the tank hid behind will not be regenerated and will continue to degrade as they are hit.

---

## **Section 1.3: Game Details and Specifications**

### **Tank Movement:**

The tank can move left and right but will not exit the playing field which is bound by the ends of the screen. To move the tank left pixels outlining the left outside of the tank are drawn and pixels on the right inside of the tank are blacked out. This maneuver will shift the tank one pixel to the left. To shift the tank right the same operation occurs however the right outside pixels are drawn and the left inside pixels are blacked out. We decided to execute the code in this manor because it is much faster changing a few pixels than blacking the image out and re-drawing the image. This is faster because the program no longer has to loop through each pixel as well as limiting the amount of times it needs to read and write to memory. Storing this data of the tank in a struct makes it organized and easy to access. The data stored includes tank position, which is a column and row number and an integer that keeps track of how many lives the tank has left.

## **Alien Movement:**

There are fifty-five aliens located in five rows of eleven on the screen. There are three different type of aliens the first type of alien is located on the first row, the second type of alien is located on rows two and three, The third and final type of alien is located on rows four and five. As the aliens move left or right the alien will cycle between two different frames adding some animation to the aliens. As the far left or far right column of aliens reach the end of the screen all aliens will shift one row down towards the tank then continue to move in the opposite direction. Each alien type is stored as a struct which contains a Boolean telling if the alien is alive or dead as well as an alien row and column number. These Struct's are built into an array, so that each

individual alien is represented. Knowing the position of each alien allows the program to over write the aliens with black pixels and re draw the next frame of alien to the left or right when the aliens are moving. As the aliens become hit by the tank the Boolean with in each alien struct will move to dead. When an alien is dead the program will no longer continue to draw the alien leaving a blank spot. If the far left or right column becomes obliterated and no aliens exist then the alive aliens still travel to the far end of the screen. To execute this we loop through the columns until an alive alien is found and return its location to push the alien hoard to the end of the screen.

## **Bunker:**

As the bunkers begin to be hit with the alien bullets or the tanks shells they will begin to degrade. There are four bunkers and each bunker is split into ten different sections. These sections will degrade at different rates depending on how many times they have been hit. Each section will only be able to take four hits, each time showing the degradation. There are four degradation templates that will overwrite the bunker section every time a hit is registered. Once a bunker section is gone bullets will travel through uninterrupted.

## **Bullet movement:**

Both Tank and Aliens will be able to fire bullets. However there is only one tank bullet aloud to be on screen where aliens are allowed to have four bullets on screen. The tank built is a single column of pixels that move linearly. The aliens have two different types of bullets that are chosen at random using the random number generator. The first type is a cross which will oscillate the arms of the cross between front

middle and back. The second type will flip between a less than operator and a greater than operator as it travels. To move the bullets the program stores each bullet as a struct keeping track of if it is alive and its location. With the location we are able to erase the trailing pixels and draw the incremented pixel causing the bullet to move. The variable keeping track of its alive status, will be continuously checked before a bullet is fired to confirm there would not be too many bullets on the screen.

## **Game interface:**

The game will end when the tank has been hit three different times and all of the lives have been used. As the tank destroys the aliens it will earn points which will accumulate and display at the top left of the screen.

---

## **Bug Report**

The first error that we ran into was when trying to input a character we would run out of memory. To fix this error we ran our program from the DDR. However we forgot to update the starting location in DDR and outputted many random pixels at the top inch of the screen. To fix the error we needed to change the starting location.

The error that was most common was changing the in-correct bits. This happened many times from the erosion of the bunkers to the moving of the aliens. The fix for this was simply correcting the value to represent the correct bit location. This was easily accomplished with trial and error techniques. By first changing the pixel color then

updating the value of the location until it was located in the correct position.

The final error that plagued us was getting stuck in infinite loops. A few of our functions when we erode the bunker are recursive. So in some case we would get stuck in an infinite loop. To fix this error we simply needed to add a base case that would exit before entering the infinite loop.