Space Game Shooter:

Report

By: Javier Ramirez

Table of Contents:

Game Concept:

The aim of Space Game Shooter was to produce a fun, and addictive game for the player to enjoy destroying multiple waves of enemies that get thrown at them while also avoiding their projectiles from damaging you.

Introduction:

The idea behind my game design began with writing my ideas down and brainstorming the implementations I would need to add in this game. Game setting would take place in space while you controlled a spaceship, I knew I would need multiple items added to my game so I created a rough sketch of how this game would be made and followed it as my design progressed. With huge inspiration from the game "space invaders" I was able to create an idea of what my game needed to have as its main components before I built the game. More specifically "Space Invaders 1978" was the game I studied and wanted to originally replicate but decided to add my own components when it came to building this game. Essentially I would need a few basic things to begin my creation process: Ship, Enemy Ship, Ship Bullets, Enemy Bullets, and the main function tab. With the game mechanics of space invaders being that the movement was limited to a horizontal position at the bottom of the screen I wanted to switch things up. The concept of enemies being able to shoot at you and damage you was interesting so I attempted to add that to my game. I decided that the player would have freedom to maneuver through the game play screen as he pleased. Thanks to the many tutorials shown in class asteroid game peaked my interest the most originally I was going to create a Unity game where you would be able to shoot different waves of mobs and your score would increase, since asteroid game was something I could reference and use to make my own game I decided to use Processing to start my game. My first idea or concept was a zombie-like shooter where the player would fend off waves of zombies. I changed my idea to make a space game shooter that earned a score from destroying enemies that spawned.

Timeline of Game:

The game started from a sketch of what I wanted it to eventually be the player Ship would be created as the player controlled object this object would be able to shoot bullets at Enemy ships. First I began writing code to create the space this would all occur in, next was to set a background to black and now I had a blank space to work with. I created ellipses as the place holders for the user controlled object and enemies for now until later implementations of images that would be added to these objects. Secondly was to make these ships move which I will explain in further detail in the game mechanics section, once movement was implemented to the ship, I needed to make it shoot a bullet out of it. Once I had a moving ellipse with shooting capabilities I had to replicate this for the enemy ship but make its movement random instead of player controlled. Lastly I had ships and enemies with shooting capabilities but no collision was implemented yet so in order to get this done I used the distance function that basically detected if 2 objects reached each other they would then both get added to an array list of objects for deletion.

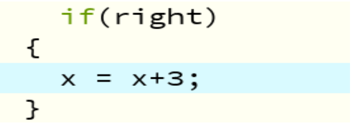
Game Overview/Initial Design Phase:

Character **Ship**: Basic movement and shooting capabilities, ship can shoot and maneuver around the game world shooting is limited to upwards vertical position replicating space invaders and movement is up, down, left, right. User shoots with a spacebar and creates a new bullet to be added to an array list where it would be drawn to shoot upwards at enemies from ship position.

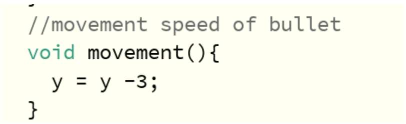
Character **Enemy**: No player controlled movements instead the enemy was randomly created using a function that would spawn 10 enemies at first at random positions moving right and left without leaving the game window boundary. A random bullet is to be created and added to an array list where it would then shoot out of a random enemy position during that frame using the random function in processing. **Bullets/Enemy Bullets**: basic idea was to create 2 seperate tabs for writing their class variables. In this tab I would only calculate the size of the bullet, and the speed of which they travel on the y-axis. Essentially they worked similarly but their speed was slightly different in order to give the player a slight advantage. **Collision**: when ship hit an enemy this meant the bullet that hit the enemy and the enemy itself would mean they needed to be deleted since array lists helps dynamically change this adding the to one for deletion was necessary.

Game Mechanics/Game Play:

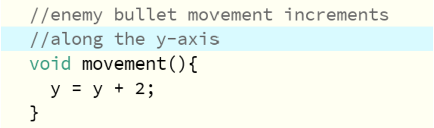
How did the game operate, what were the necessary functions used and how did everything work? Creating a screen size of 500x600 with a black background as my starting template for the game, I created multiple tabs besides the main game tab, ship, bullets, enemy bullets, enemy ship tabs were all implemented and would be called from the main game tab when their classes were to be implemented. I began with building the ship class under the ship tab for the game since this was a good start. I set an ellipse as its placeholder for now until I managed to change it to a loaded image from my data folder. Ship tab initialized x position and y position for the ship to be spawned at in the screen, a few boolean variables were implemented for right, left, up, down and set to false for later use. Movement speed was controlled by if statements that controlled how fast the ship moved according to x,y-axis using keyPressed(), and keyReleased() function and some more if statements that would set the previous boolean variables to true if the user had a certain up, down, left, right key pressed or released. To better understand this, if statements controlled if the user pressed a key and if he did the ship would move and position would update for the ship, each frame the ship moves the position of ship updates for later implementation of bullets I would use this position for ejecting the bullets out of the ship. I started by having a moving ellipse now it was time for bullets to shoot out of my ship's position but I knew I needed to create a dynamically changing array that will store everything for addition and deletion of bullets, and enemies throughout my game this meant I would create a few array list objects “<Bullet>, <Enemy>, <Bomb>" to indicate my intention to fill ArrayList with bullets, enemies, bomb objects. Now my bullet tab contained class bullet where I would create an ellipse for it as a placeholder for now and it's movement was controlled by the following: if ship shot a bullet it would travel from whatever position y upwards so if it was shot out of y = 450 it would decrement 450 by -3 and continue frame by frame until it exited the screen and vice versa for the enemy bullets tab.



```
   if(right)
 {
   x = x+3;
 }
```

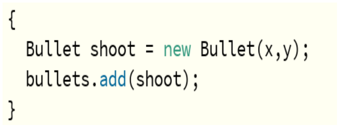


```
//movement speed of bullet
void movement(){
  y = y -3;
}
```

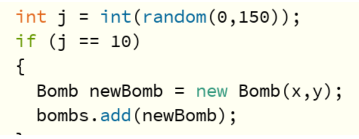


```
//enemy bullet movement increments
//along the y-axis
void movement(){
  y = y + 2;
}
```

Left Image gives an idea how movement worked for the ship when pressed right under the ship tab , 2 images on the right explains how bullets would travel across y axis one shoots up one shoots down. Movement of ship and bullet movement was implemented now it was time to make the space button add new bullet objects to the array list bullets in order to do this under ship tab the following code was used to add new bullets to array list from position x,y of ship during the frame it was in at the time: This would work similarly for enemy bullets where enemy bullets would add new bullets to it's array list and then be drawn in the main tab window.



```
{
  Bullet shoot = new Bullet(x,y);
  bullets.add(shoot);
}
```

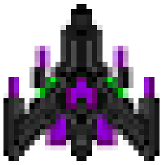


```
int j = int(random(0,150));
if (j == 10)
{
  Bomb newBomb = new Bomb(x,y);
  bombs.add(newBomb);
```

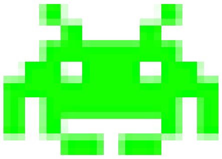
Enemy bullets would use the random function for each frame if the random function = the number 10 for a specific frame then it would enter the if statement and add a new bomb that would get shot out from the enemy. The image on the left would allow me to create a new bullet everytime spacebar was pressed to shoot out of position x,y of my ship or ellipse at the time while moving around the game screen. Next was to implement my enemy under enemy tab since this was not a player controlled object the idea was to randomize their x,y position on the top of the screen and have them not exit the play area I wanted these enemies to move right and if they hit the boundary move left and work vice versa for the boundary on the left, the boundary was set up with if statements that said if the enemy reaches the left boundary of 500 then move left a certain amount of frames and set the right boolean for the enemy to false if enemy reaches left boundary of 0 set right to true and the enemy will move to the right. Since the bullets for the enemy worked in a similar fashion to the ship bullets they would randomly be added to enemy objects at random positions of enemies and shot out randomly from the enemies I spawned in the main game tab. Now I had a ship, enemies, and bullets with movement working and I needed to somehow implement an array list of objects to eliminate enemies dynamically if a bullet collided with it, remove it and the bullet it collided with as well as reduce the health of the ship if enemy bullets were to hit it. Using a distance function implemented I was able to remove both objects that collided based on if their distance reached a certain number or essentially got close to each other delete them by adding them to an array list for deletion. Furthermore I finally had a working prototype for a ship that was maneuverable and was able to shoot down enemies. My next implementation was to add a float score, float health, and waves of enemies once the score would reach a certain amount. Float was used as global variables in order to update it throughout the game, using if statements under the function where enemies were to be deleted I added a score increment so every time an enemy was deleted score would increase and displayed on the gameplay similar to score if enemy bullet hit my ship I added a float health and set it = to 100 meaning the ship has 100 health points and when hit by a bullet the ship would decrement health by a certain amount. This worked but in order to add more enemy waves I needed a for loop to know when score reached a certain number lets say 10 it would need to add 20 more and then more on top of that when score reached 20, making the game progressively difficult as you continued playing. I had to implement different screens for start screen, game over screen, and win screen these screens each had to have their own prompts for displaying a text of a certain size the start screen had the tutorial displayed and if the key “s” was pressed it would redirect you to the start of the game in the game if your health points dropped to 0 then you will be redirected to the game over screen, however if you win the game by killing 50 enemies then the game would end and the user would redirect to the win screen where they could see their final score with an option to quit the game. Depending on the game state the game was set to if the game state changed then that would redirect the user to certain screens.

Gameplay Visuals:

For my gameplay I used multiple free .png images and used the masking function in order to display them in my game for them to work smoothly and would display no errors in order to do this first I found a ship.png, enemy.png, and created my own laser bullets using paint on windows these pictures are shown below. Note: I had to adjust the size of these images exponentially in order to have them fit well within my game which is why the resolution is low quality as shown.



A.Ship

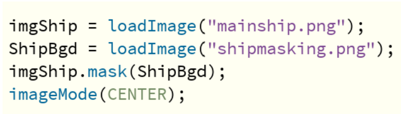


B.Enemy

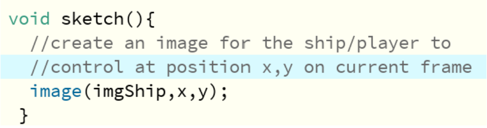
EnemyBullet

ShipBullet

These images all contained a masking I created with windows paint, basically a black background with an outline of these images shown above. To better explain this, I referenced processing’s website: “Masks part of an image from displaying by loading another image and using it as an alpha channel. This mask image should only contain grayscale data, but only the blue color channel is used. The mask image needs to be the same size as the image to which it is applied.” Once the image was loaded using the following function below it would mask over the original photo and I just needed to remove the ellipse placeholder and replace it with Image() function where ever the position of x,y of the ship,enemy,bullets, enemy bullets was located during the current frame, and it would then be drawn in the main game tab.

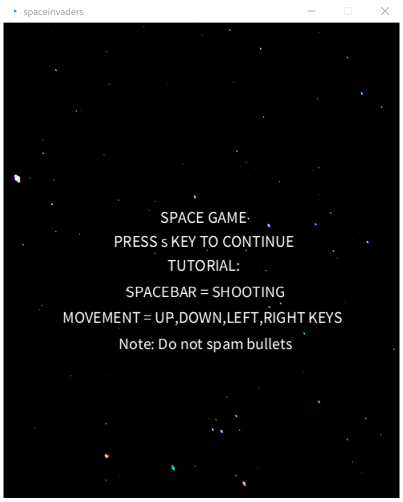


```
imgShip = loadImage("mainship.png");
ShipBgd = loadImage("shipmasking.png");
imgShip.mask(ShipBgd);
imageMode(CENTER);
```

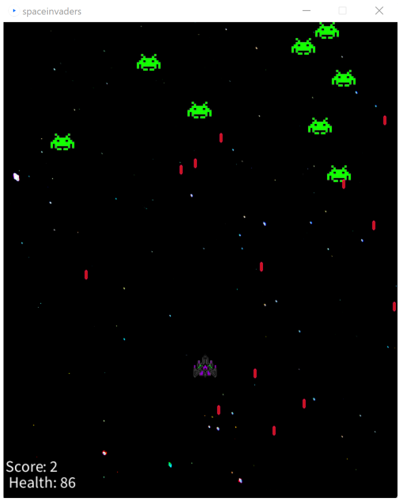


```
void sketch(){
   //create an image for the ship/player to
   //control at position x,y on current frame
   image(imgShip,x,y);
 }
```

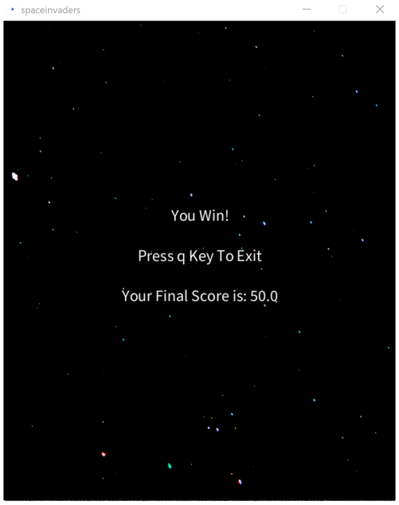
Furthermore implementing a background for my game was different since I was originally using a black background so I searched for a space image since my game was a space shooter the image below was implemented the following way: first creating a PImage start screen and then loading it in the setup function if the gamestate was set to 1 it would load it to the start screen once the s key was pressed to continue to the game I would load it to all screens that I had created.



SPACE GAME
PRESS s KEY TO CONTINUE
TUTORIAL:
SPACEBAR = SHOOTING
MOVEMENT = UP,DOWN,LEFT,RIGHT KEYS
Note: Do not spam bullets



Score: 2
Health: 86



You Win!

Press q Key To Exit

Your Final Score is: 50.0

The above images are visual representations of the game as you progress it once the images are loaded like stated in game mechanics all I had to do was assign them to each stage of the game and the objects ship, enemies, and bullets and their x,y position so they can display it in the game screen. Lastly, where do you store all these images that get drawn for these objects, these images should be saved under a new folder called data where it's placed in the same location where your program is located and its various tabs.

Bugs/Errors/Problems:

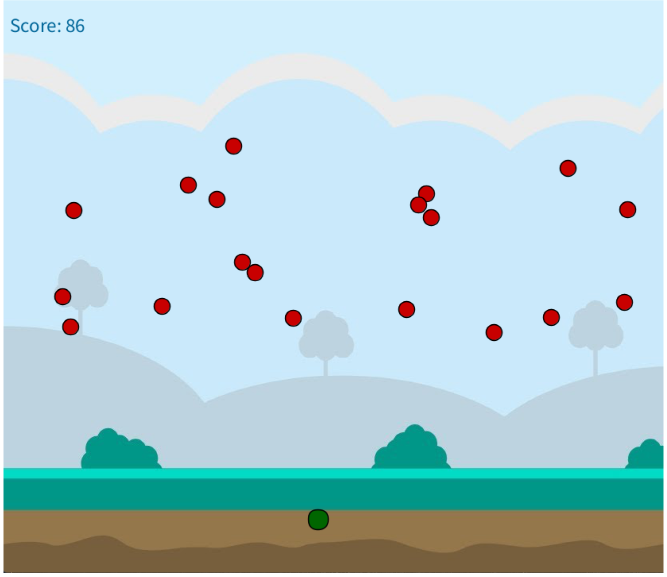
The game as of now contains a bug but not necessarily one that will affect the whole program and not have it compile, I was able to discover this bug at late stages of implementation so fixing this will be a future implementation for this project. I plan on fixing this specific error and making this game as smooth as possible. As of now the bug you can encounter is: if you spam too many bullets too quickly before score reaches 10 you will get a bug preventing any progression. Essentially the game will bug out your score since my gamestate conditions are if you get a certain score then more mobs will spawn in, this bug stops the second and third wave of mobs from spawning in unless the player takes his time to kill the mobs with 1 shot only. If the last mob will give you a score of 10 and you kill that mob with one bullet then this bug does not happen, but if you spam bullets at it there is a bug that occurs where your score will bug out having you be unable to get to the win screen page where your final score is displayed since I used the distance to calculate if objects hit each other this is probably why I am getting this bug the bullets of the ship are traveling too fast if I were to slow the amount of bullets that get shot out of the player ship then this bug will not happen so often. Potentially a fix for this is adding a delay to where the bullets get shot out of giving the user less freedom with the amount of bullets he can shoot at a certain time frame meaning he will have to kill each mob with 1 shot then wait a certain amount before he can take a second shot and prevent him from bugging the score out. This will work and multiple waves of enemies are able to spawn in after another thanks to a for loop that spawns more in. In continuation a fix I encountered was that enemy bullets and ship bullets would not delete after exiting the boundary, why did this need to be fixed? In order for no future lag or slowness of gameplay this meant I needed to add the bullets that left the game screen to an array list of objects for deletion once the position of the bullet was greater than my size of the screen > 500 or < 0 for enemies. Lastly enemy bullets sometimes don't register to give damage to the player ship. This is because my ship moves at a faster pace than enemies and bullets. If the ship is moving forward too quickly then the ship can bypass the health being reduced. I will need to tweak this in order to have these bullets work more smoothly than what it is now, when using distance of two objects I will need to adjust accordingly to make my game slower paced since objects are moving too quickly and giving me these errors.

Future Work/Implementations:

Due to time constraints I was not able to fix bugs, or add important features I wanted such as having coins randomly spawn as the game progressed I wanted to implement multiple waves of enemies and have different enemies that would take more bullets to kill. Furthermore a highscore system that will store new players then display their score in the credit screen. Another good implementation I had in mind was to create a level format for my game having a boss level where this boss has a float value that was set to 100 health points or greater than 100 points, I know that this will need to be added as a separate class altogether and import a separate .png image for the boss' appearance itself and better experience for the actual fight. A boss level would need to have the position of where it will spawn in, the maneuverability of the boss making it was slower than the normal mobs that you destroy in the main game, this boss will need to move at a different pace as well as a different angle such as being able to move up down left right randomly to best avoid your bullets. If this boss takes damage then his health points will decrease but also have his health points displayed on the screen and decrement it as he gets shot. This boss I had in mind will have different shooting capabilities and possibly its own bullets that are faster and bigger than the normal mobs having that difficulty when facing off with him. My future implementations will probably be different mobs instead of just the normal ones and the boss I just explained. In addition, creating a way for me to not use distance of 2 objects for collision detection will be a better outcome for my game, since objects sometimes travel way too quick they can produce errors which are able to be fixed with delays but a game will run much smoothly without it. Furthermore I would like to create an economy for my game when killing a mob you receive a coin. This coin can be used later on in a game menu where if the game menu contains a store, this store will have upgrades to your ship potentially for increased speed, larger bullets, or multi-bullet upgrades. I could also add new ships so users can change their appearance if the default ship isn't what they enjoy. Once a user has collected a large amount of coins from beating levels and destroying mobs he is given the option to buy the ending screen which means once he collects enough coins and buys this feature he will be able to exit the program and end the game. Overall this game needs replayability aspects which means maybe add randomness to the game or even easter eggs like rare ships, enemies, upgrades that will peak the players interest when they are first discovering these items for example a laser beam upgrade that would bring excitement to the user when discovering it. Another concept I had in mind was to add various sounds that would trigger for example if the enemy was shot down the audio will play if the bullet hit the ship the audio will play as well as having a background sound while in the main game window not only this but when shooting bullets out of the ship have it make laser beam sound to add sound over the whole game. In addition potential obstacles that could be added for the player to hide behind explosion effects when an enemy gets shot down by the bullet. Lastly this game needs to have the enemy mobs attack the user maybe by shifting their position down the y-axis and eventually reaching the player if he doesn't kill them in time, again much like the Space Invader Game of 1978.

Motivation for the Game/Prior Ideas/Brainstorming:

The main motivation for my game came from years of playing shooter games. I always had an interest in shooting games, especially one that lets you collect a score depending on how many enemies you destroy. I had an idea of a shooter game which was a zombie-like game, where hordes of zombies would approach your position x,y and you would have to rotate like the asteroid game and shoot these mobs. My original idea was to have waves of mobs spawn in and create an increased difficulty for the player/user. I had thought to use unity to design this game but changed my mind to a different game idea altogether. Using processing I had created a simple game prior to the project I have now that aided to the development of my current project in the project shown below waves of bombs/spikes would randomly be dropping from the top of the y axis using the random function and if that spike hit my player controlled object's ellipse he would lose and his score would display at the end but if it didnt the game progressively would get harder as well as keeping track of score and incrementing as waves were added. This game used the mouseMoved() function in processing allowing the mouse to control the movement of the green ellipse to avoid the obstacles the user would move the mouse and gain a score for surviving the waves of spikes that spawned.



Originally this was the project I was thinking of using and implementing further with game menus, replayability, and lives with win/loss conditions. After seeing the asteroid game example I decided to get more creative and create a different game altogether, one that I enjoyed working with and learning how the various functions worked. I had an idea of how everything worked in processing for movement and collision so the rest was implementing it all at. Lastly I had attempted to create a Unity game where the player was able to move around in a specific play area with the player being able to shoot around with this mouse and arrow keys and click movements were used as shooting mechanism everytime he would click the bullet would shoot and if it reached a object on the screen it would explode the game was viewed as 2D of the character from overhead position

Experience Earned:

When it comes to experience that I earned for this project I was able to become extremely familiar with array lists when creating an array list of objects to work behind the scenes to dynamically add/remove items from it. This was one of the main functions that I researched online for help as well as tutorials on how exactly they worked for adding bullets, enemy bullets. Furthermore I learned how movement worked, how to increase or slow down movements for example an object can be slowed down or increased. In addition I experienced how to load images properly for my game to run smoothly. I created different scenes for my game and even added a background for each game scene. I became more familiar with class functions and how collision works with distance as well as the disadvantages to using it in-game. Lastly being able to experiment with Unity and Processing gave me the ability to get an idea of how games are created while using them you get to understand the backbone of these games and their implementations of how they actually work from movements to screens being created to deletion of objects or even the keypressing of the arrow keys has to be all well considered and implemented correctly for the user to have a positive experience in your game. I was able to overcome multiple errors that would ultimately make my game collapse and be exposed to how processing and unity works.

Conclusion:

Overall I am extremely satisfied with the outcome of my game although it did contain a bug the game works well. When first starting out with processing I had little to no knowledge of how everything worked, processing is beginner friendly and I was able to quickly grasp many of its concepts since they are similar to other programming languages I have been exposed to. I started with an idea of having a game like “Space Invaders 1978” but quickly changed it around to make it more unique that targeted the sci-fi shooter audience. This game is a shooting game that presented many challenges I had to overcome from errors that would make my game slower to errors that bugged out the win condition/score. With these challenges presented towards me I was able to learn many new things and experience how many of the functions worked by tweaking their variables and playing enough with the numbers to make things change within the game. Not only did I learn a majority of processing’s documentation but thanks to the references they provided in their website it was easy to implement into my own game. With enough tweaking and future implementations I plan on building more on this game to have a smooth working project that has replayability functionalities and exciting game modes/levels. I can truly say that this whole process of learning how processing works improved my knowledge more since some of the functions are very familiar to me.

References:

https://processing.org/reference

https://www.youtube.com/watch?v=2VLaIr5Ckbs&list=PLzJbM9-DyOZyMZzVda3HaWviHqfPiYN7e&ab_channel=TheCodingTrain

https://stackoverflow.com/questions/6510837/distance-between-two-points-in-processing-app

asteroidGame Example, Sprite Example, Tutorials about sprites, Distance formula using processing to measure the hit boxes between bullets and ships.