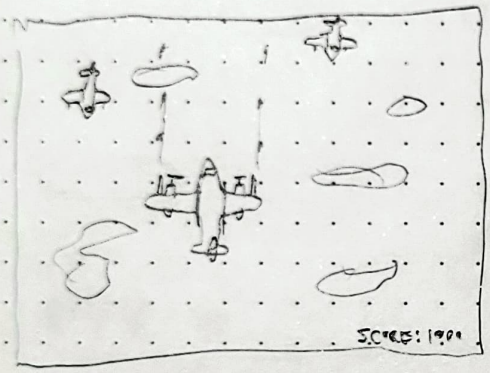
Game 10003 - Game Development Foundations - Assignment 3

Brainstorming

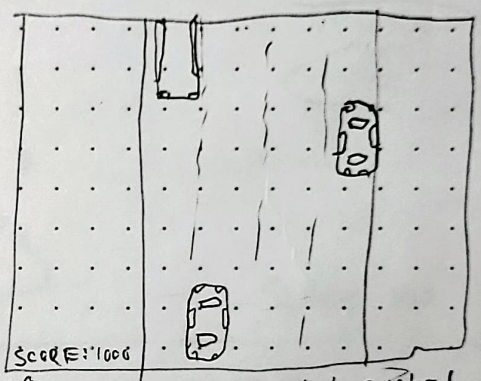
Games to clone (optional)

- Dinosaur Game (chrome T-Rex Game)
- Concentration (card game)
- Frogger
- Breakout
- Bubble Struggle (Pop & Avoid Bubbles)
- Space Invaders

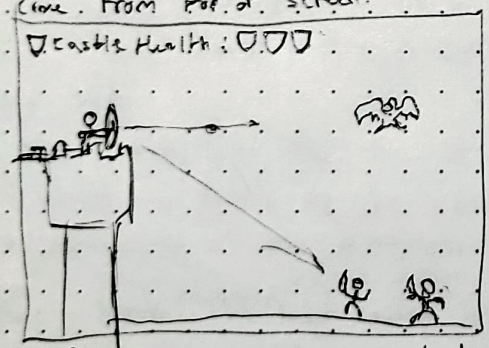
Size: 800 x 600 / 600 x 800



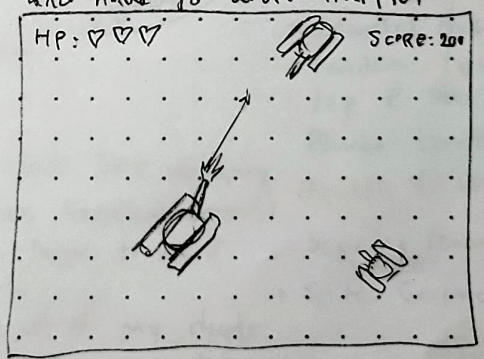
- Aerial combat game where you have to fight enemy planes that come from top of screen.



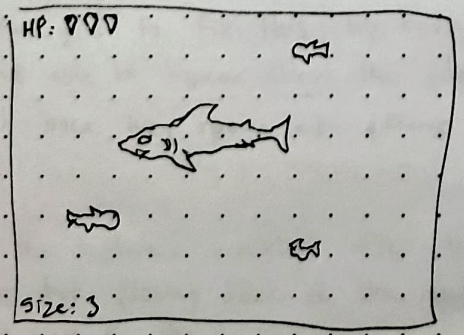
- A car game where you control a car driving down the highway and have to avoid traffic.



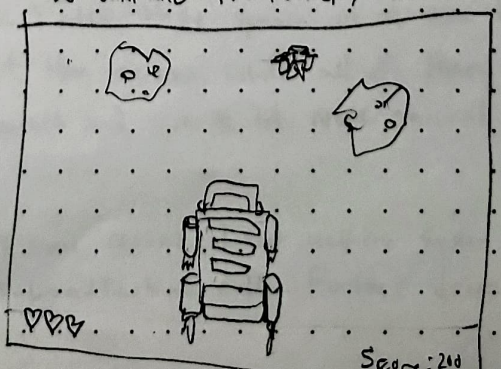
- Defense game where you control an archer defending a castle from monsters.



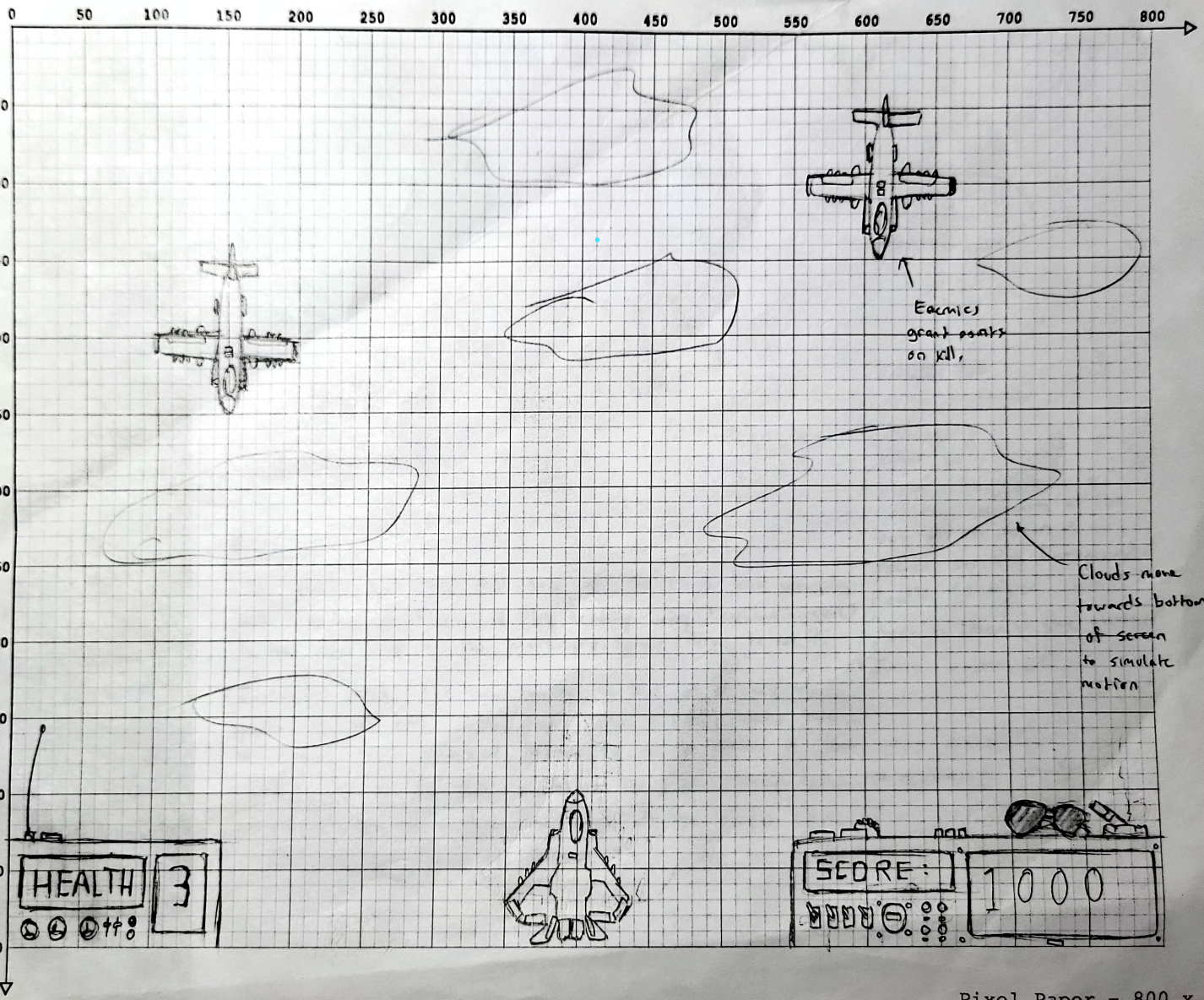
- Tank combat game where you have to survive as long as you can. Mouse to aim and fire turret, wasd to move tank



- Feeding Frenzy style game where you control a fish that changes into bigger and bigger fish as you eat fish. Fish that are too big deal damage.



- A top down version of Galactic Cleanup where you have to avoid asteroids and collect garbage.

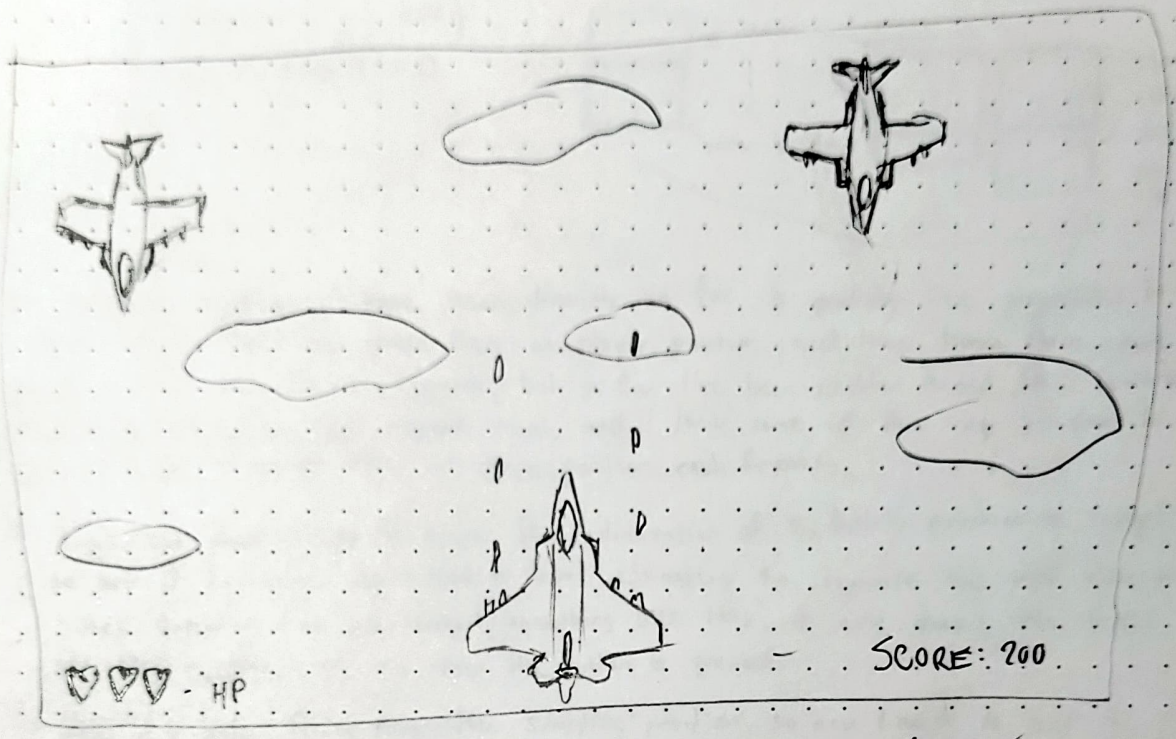


Pixel Paper - 800 x 600

Graph Paper for Screen Coordinates

Plane Game:

- In this game, the idea is that you are controlling a fighter jet and have to shoot down enemies that come in from the top of the screen. Enemies grant score when killed. You have 3 health, and running into enemies or (maybe) getting hit by their weapons takes off one HP. Run out of HP and it's game over.



Controls:
- W/A/S/D = Move plane
- Spacebar = Fire guns
- Mouse = Using/Navigating Menu

Features: DONE!
- Moving ✓
- Shooting ✓
- Enemies that spawn in random locations at the top of the screen ✓
- Pseudo-scrolling background ✓
- Health (3 HP) ✓
- Scoring (Enemies give points) ✓
- Sprite Graphics ✓

Notes:

- One issue I had was that none of the clouds were wrapping and I ended up fixing it by actually using .KeepCloudsOnScreen() in the same loop as the movement. I guess I forgot to use it.
- Another issue I've been having is that all of my clouds spawn in at once at the top, when I really want them to spawn in a staggered pattern, so as to make it look more natural.
- Managed to fix this by having the Y value they spawn in at also be randomized, and able to appear above the edge of the screen. While all of them still spawn in at once, they spawn at different Y values and look a lot more natural.

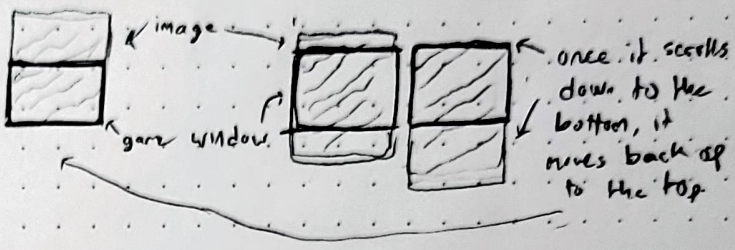
- Got the textures working after having some issues. I kept getting System.AccessViolationException errors, but placing some of the Graphics.LoadTexture calls further down in Update() fixed most of it.

- Got the fonts working, and it was surprisingly simple. It's essentially the same process as the 2D textures.

- I want to have a scrolling background to simulate movement, but I'm not exactly sure how. Perhaps I could just make one long image to scroll through?

  ↳ Managed to figure this out. I ended up going with a tiled background that I made, and moved it down the screen. Once it hits the top of the image, (bgPosition.Y = 800), I set it back to the starting point, where it continues this process indefinitely. It ends up looping smoothly, and doesn't look choppy.

```
if(bgPosition.Y == 800)
{
    bgPosition.Y = 0
}
```



- The biggest challenge I have been having so far is getting the projectiles to work. I have been trying to spawn them on player position and then have them move upwards from where they spawn, but so far I've been unable to get this to work. They keep staying on their spawn point, and I think some of this may be due to how it keeps drawing them at player position each frame.

  ↳ Turns out that I had to move the initialization of the bullet's position to setup() to get it to work. I also added a bool isShooting to separate the input from the bullet drawing function. Without something like this, it only draws the bullet for like a frame or so when the button is pressed.

  ↳ Now it is only firing from the starting position, so now I need to get it to fire from wherever the player currently is.

  ↳ Managed to fix this by adding bulletPosition = position; to the Shoot() function, so that it correctly sets the starting location when you press the fire button.

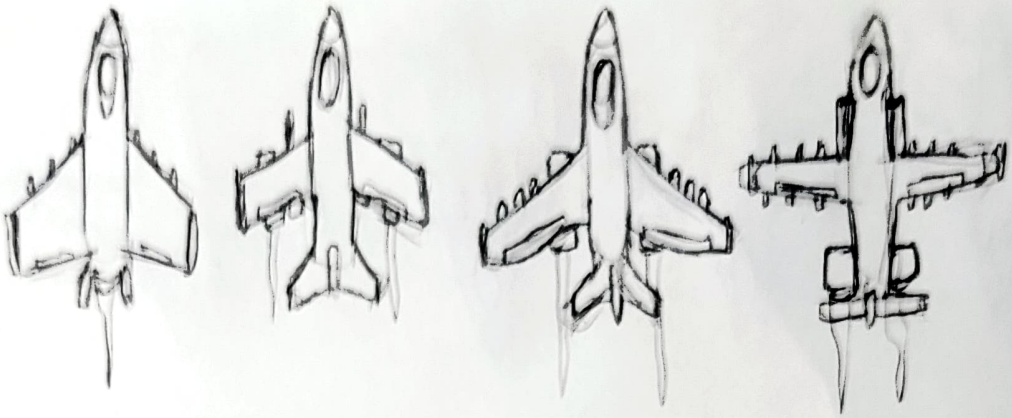
  ↳ After this, the next challenge is to figure out how to get it to shoot repeatedly. Currently it's just set to fire once, so I have to add a loop somewhere.

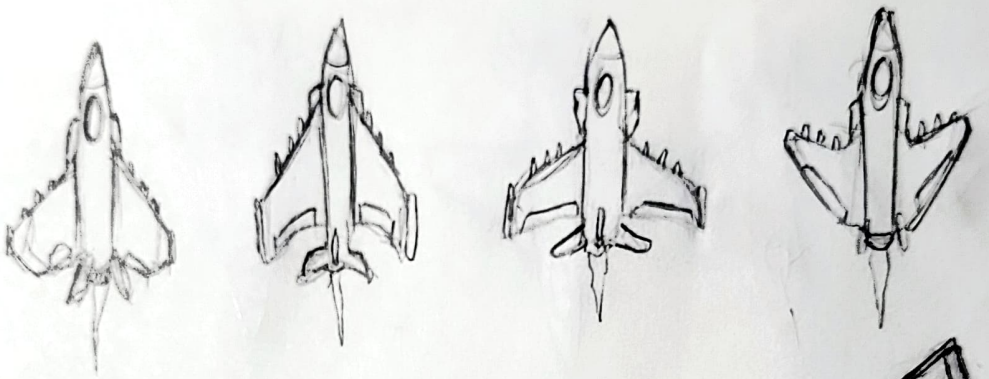
  ↳ I ended up adding a loop in Game.cs under the loop for enemy collision. It's very similar to the one for the player, but applied to the bullet instead. I originally tried doing this within the player class, but I could only get it to work properly within Game.cs.

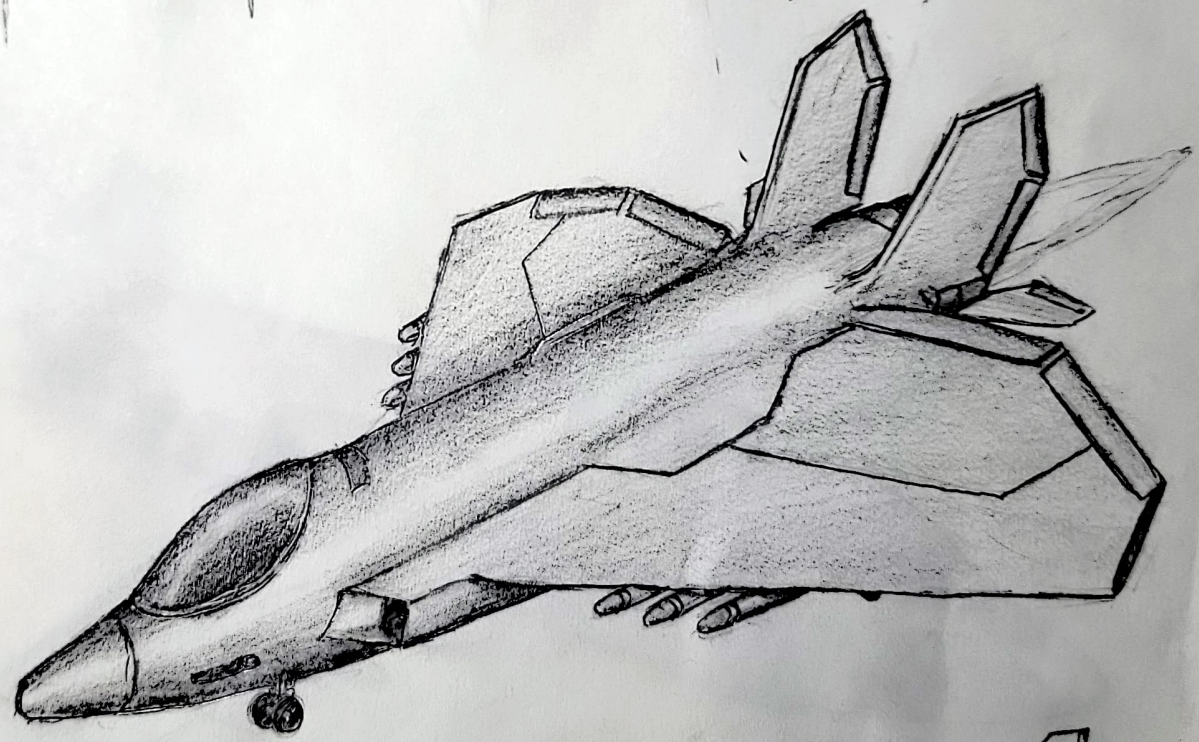
  ↳ Now I am trying to work out the collision for the bullets. I am going to essentially re-use the code for the player collision, and make it so that hitting an enemy despawns the projectile, grants score and hides the enemy.

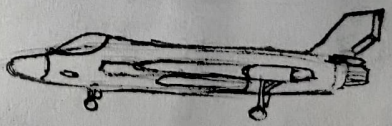
  ↳ I originally had the loop using bullets.Length for the condition, but that wouldn't work. What worked was putting in activeEnemyCount though that had the side effect of making the enemies too fast. To fix this, I just halved the Vector2 that was being added to the enemy's position each frame. This fixed the speed and didn't ruin the collision code the way that changing the movement vector2 from (0, 6) to (0, 3) did.

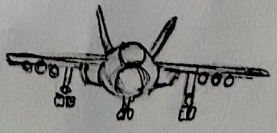
2D Game Project Concept Art - Plane



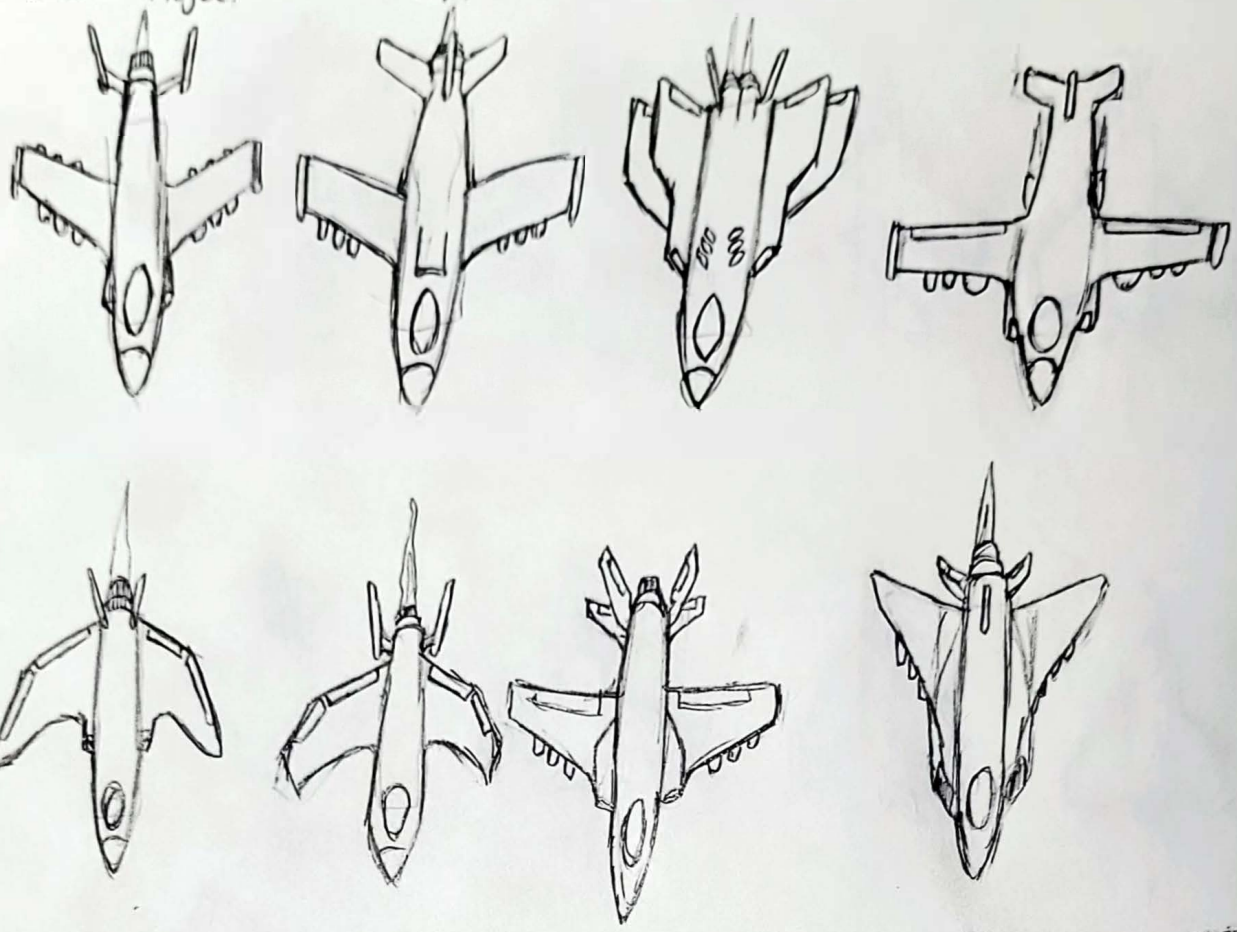


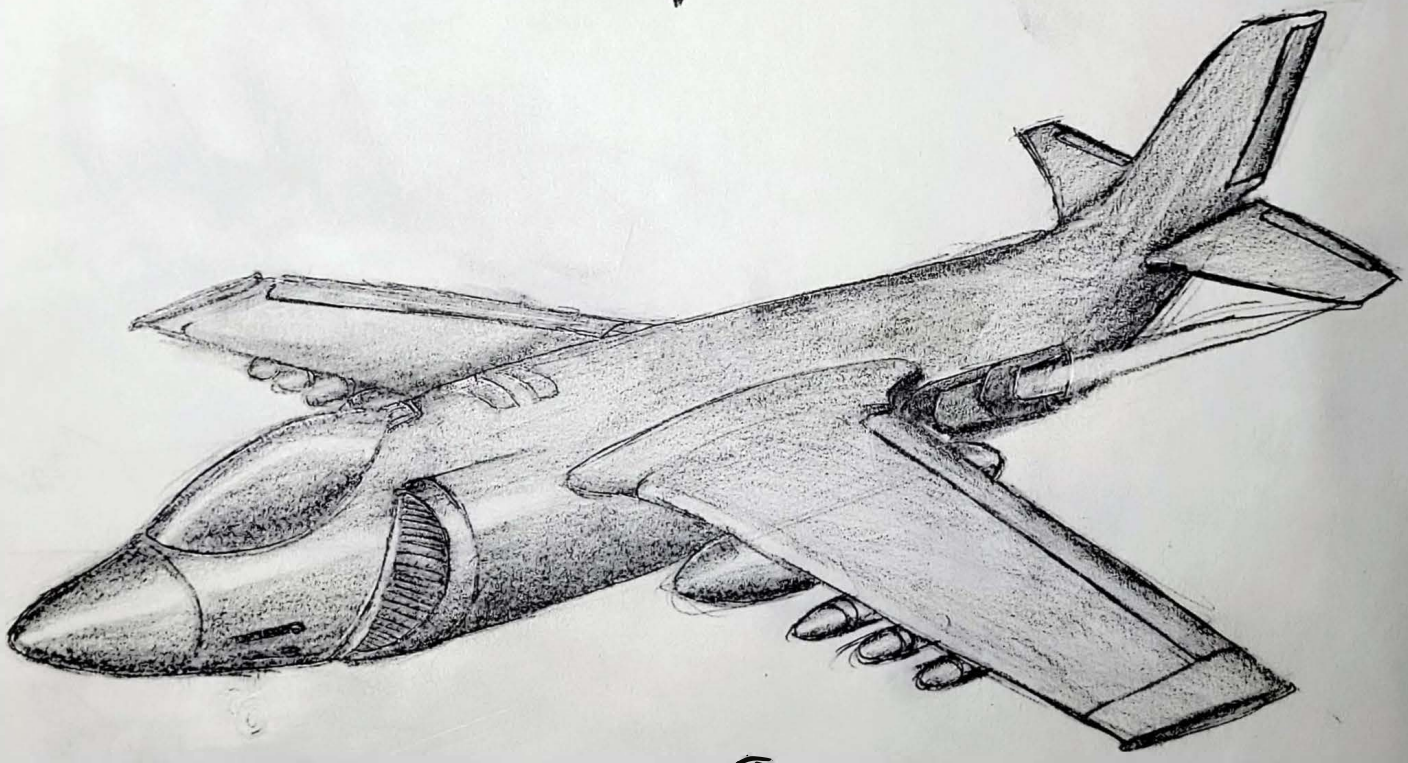


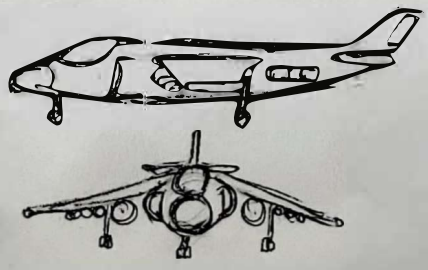




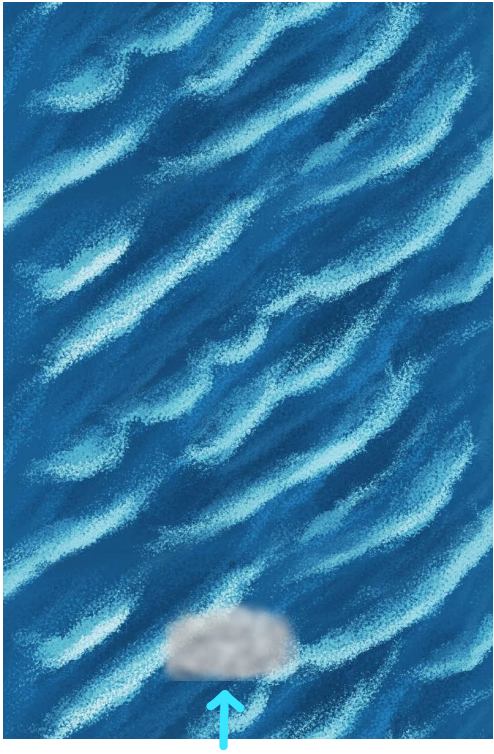
2D Game Project - Concept Art - Enemy Planes







UNUSED BACKGROUND



UNUSED CLOUD TEXTURE