



Game Proposal →

Inspiration →

Systems →

Gameplay →

Level Design →

Prototype →

Case Study →



GAME DESIGN CASE STUDY SPRING 2024

Game Designer: Jack P Smith @ SUNY WCC ITECH

Project Title: Galactic Vortex

Prototype: <https://jackpsmith.com/wp-content/uploads/2024/11/GalacticVortex.zip>

Course: ITECH 200 Game Design – Fall 2024



DESCRIPTION:

Galactic Vortex is a single-player, 3D, third-person arcade space shooter where the player must take down an endless wave of enemy ships in order to survive. The target audience is casual PC gamers of all ages. The game has a futuristic, space-themed UI and the environment is set in outer space.



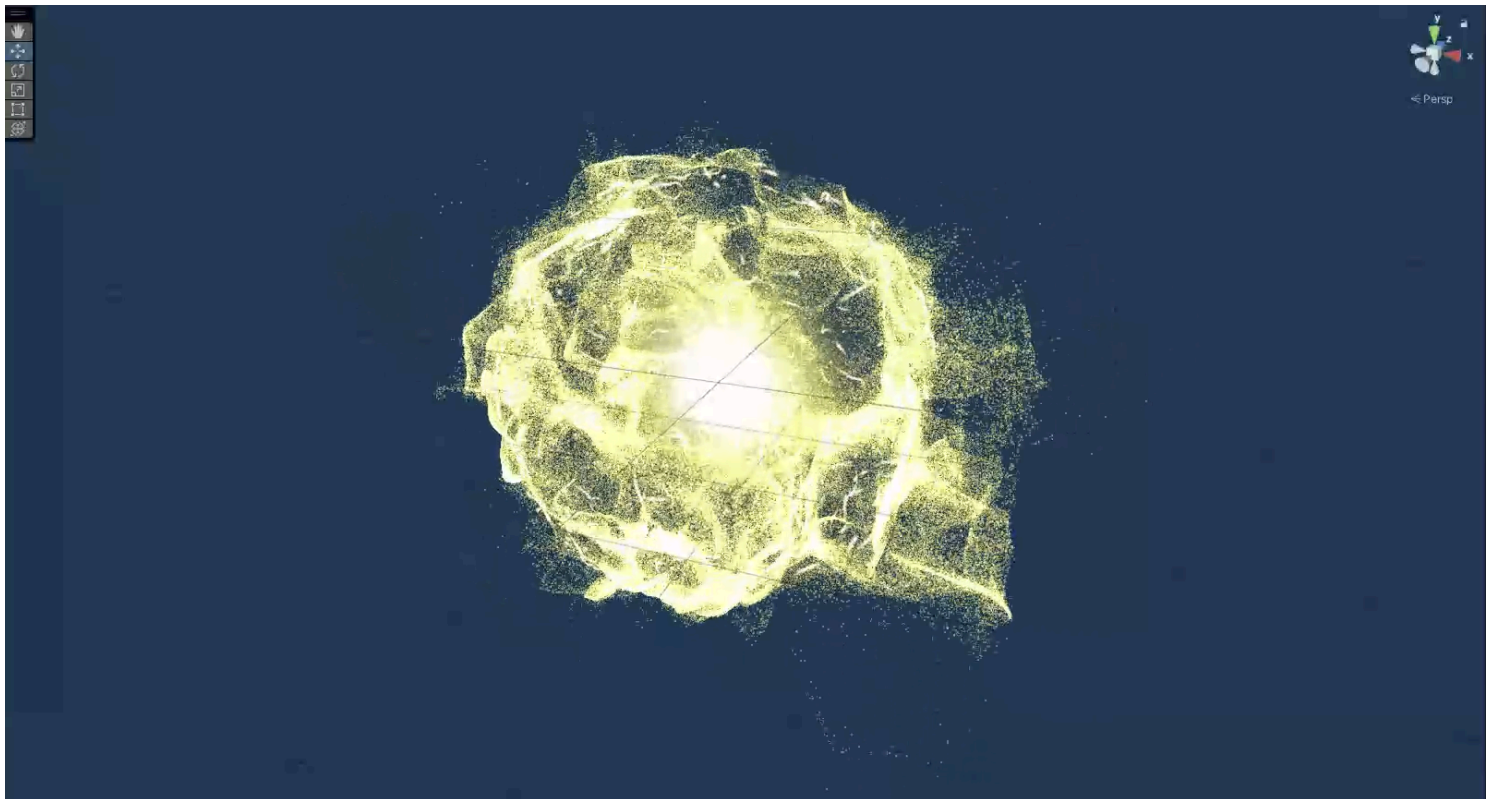
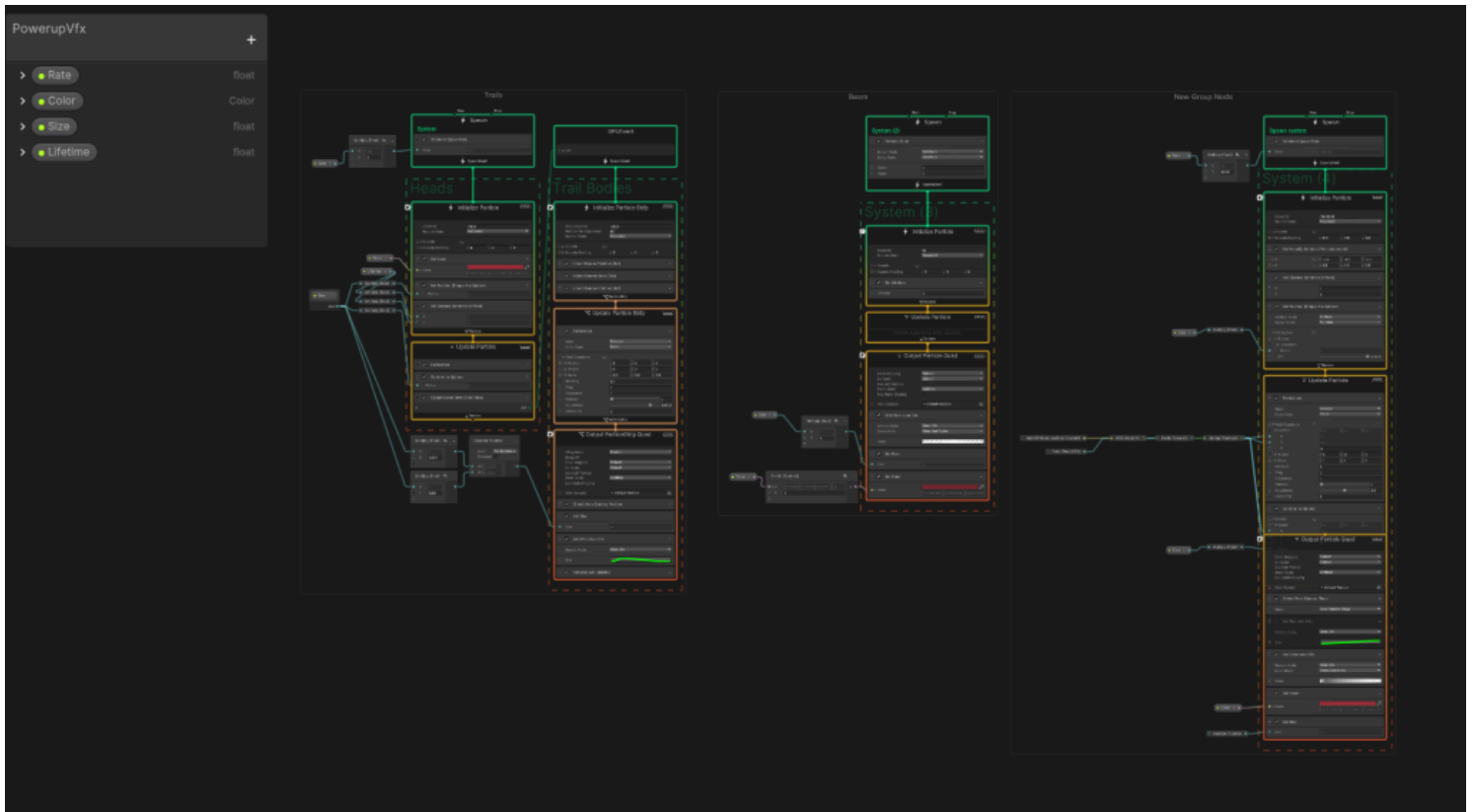
RESEARCH:

Galactic Vortex takes inspiration from several other space games including Rockfish Games' *Everspace 2* (2023), EA's *Battlefront 2* (2017), and Squad's *Kerbal Space Program* (2015). I found that one of the most important goals for my game was to make flying the ship feel fun for the player. Players spend so much time moving through the game world that it became one of the most important focuses for development.

CHALLENGES:

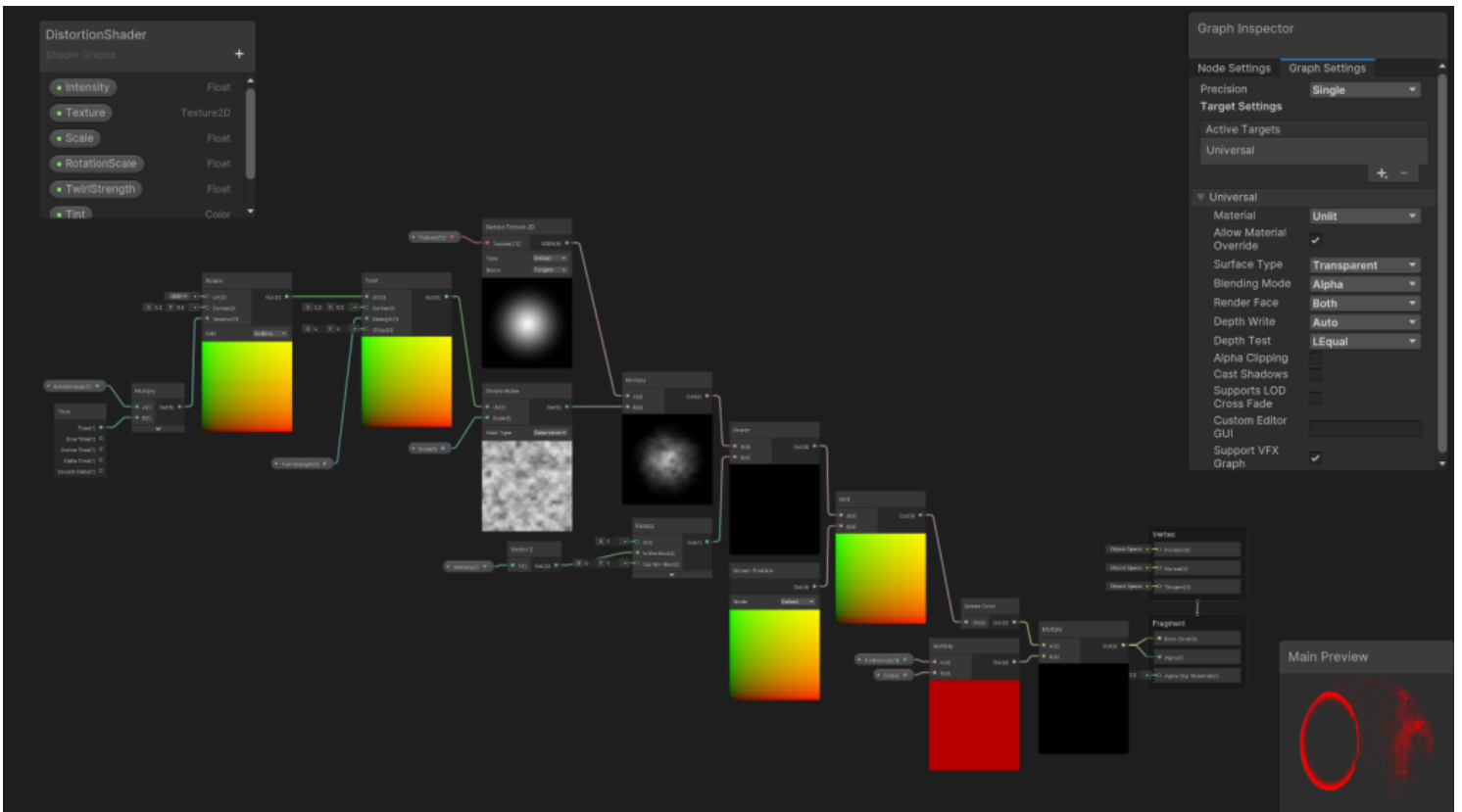
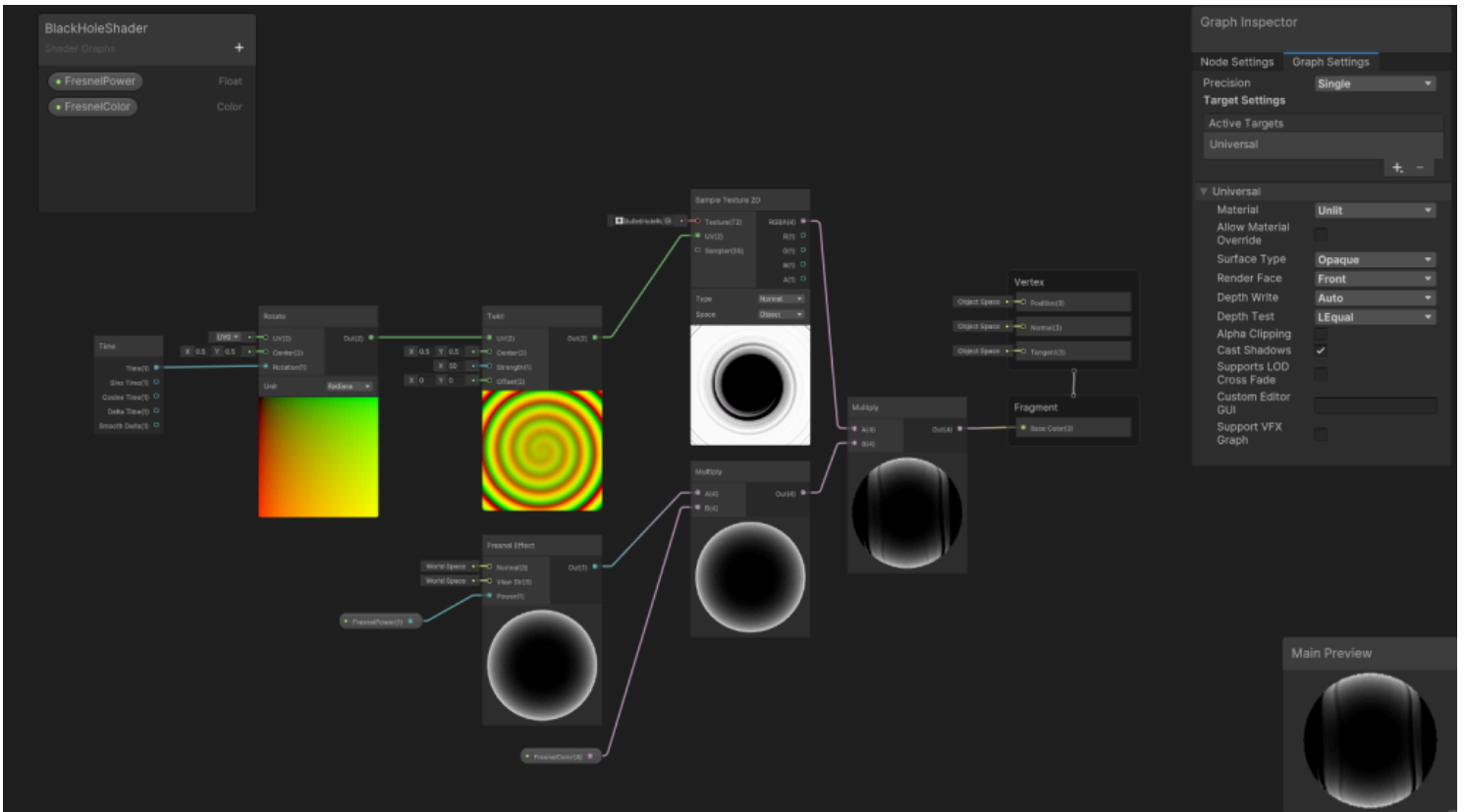
One challenge was designing the visual effects. Galactic Vortex includes several visual effects and custom shaders for the different powerups and celestial objects. I used Unity's Shader Graph and VFX Graph tools to easily create powerful visual effects.





Powerup Visual





Another challenge was creating a complex enemy AI for tracking the player ship and attacking the player without overwhelming them. While I am happy with my initial implementation, I think that with more time I could improve it even further. Below is the existing enemy AI script:



```

using UnityEngine;

namespace GalacticVortex
{
    public class EnemyShipAI : MonoBehaviour
    {
        [SerializeField] private Ship _ship;
        private Transform _playerShip;
        private Vector3 _rotDir;

        private void Awake()
        {
            _rotDir = new Vector3(
                Random.Range(-1, 1),
                Random.Range(-1, 1),
                Random.Range(-1, 1));
        }

        private void Update()
        {
            if (GameManager.Instance.GetIsRunning())
            {
                if (_playerShip == null)
                {
                    _ship.GetShipPhysics().Accelerate(2);
                    return;
                }

                UpdateRotation();

                float distanceFromPlayer = GetDistanceFromPlayer();
                UpdateProjectiles(distanceFromPlayer);
                UpdateAcceleration(distanceFromPlayer);
            }
        }

        private float GetDistanceFromPlayer()
        {

```



```

        return Vector3.Distance(_playerShip.position,
        _ship.transform.position);
    }

    public void SetPlayerShip(Transform playerShip) => _playerShip =
    playerShip;

    private void UpdateAcceleration(float distanceFromPlayer)
    {
        if (distanceFromPlayer > 50 && distanceFromPlayer < 300)
        {
            _ship.GetShipPhysics().Accelerate(5);
        }
        else if (distanceFromPlayer > 300)
        {
            _ship.EnableTurbo();
        }
        else
        {
            _ship.DisableTurbo();
        }
    }

    private void UpdateProjectiles(float distanceFromPlayer)
    {
        if (distanceFromPlayer < 200)
        {
            var shootChance = Random.value;

            if (shootChance < .01)
            {
                ShipProjectiles shipProjectiles = _ship.GetShipProjectiles();
                var subChance = Random.value;

                if (subChance < .33f)
                {
                    shipProjectiles.FireLeftProjectile();
                }
                else if (subChance > .66f)
                {

```



```

        shipProjectiles.FireRightProjectile();
    }
    else
    {
        shipProjectiles.FireLeftProjectile();
        shipProjectiles.FireRightProjectile();
    }
}
}

private void UpdateRotation()
{
    transform.LookAt(_playerShip);
    transform.RotateAround(_playerShip.position, _rotDir, .01f);
}
}
}

```

STRATEGY:

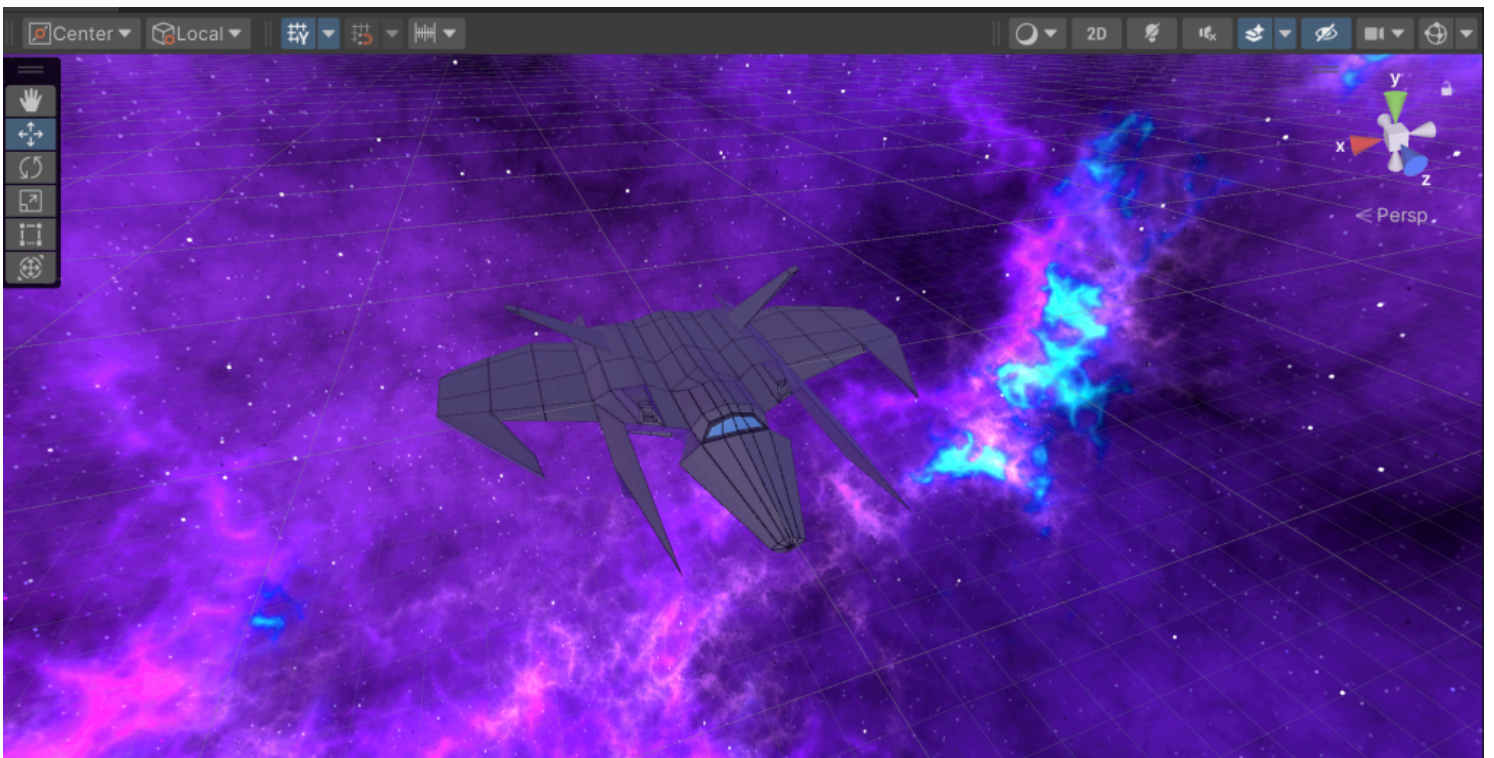
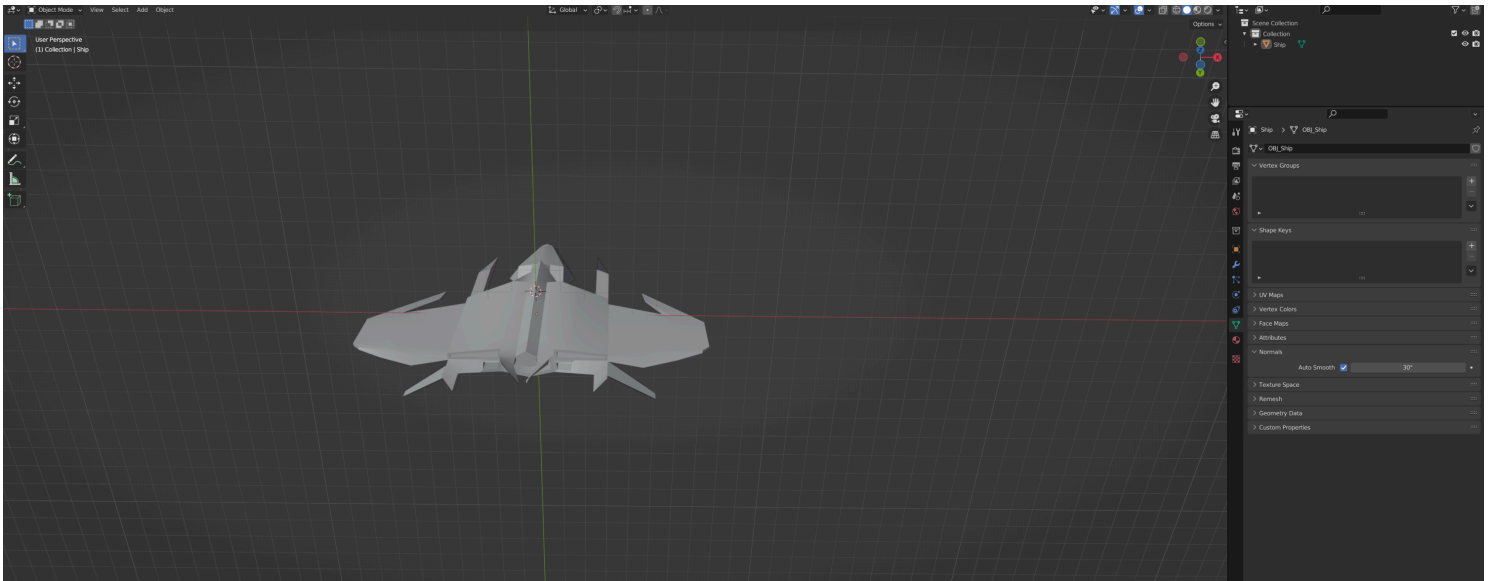
Strategy – The player’s goal is to survive for as long as possible while defeating as many enemies as possible. Both of these things will increase the player’s score, which will increase their rank on the leaderboard. The player may want to collect powerups along the way that help them in achieving these feats.

Mechanics – The game’s mechanics revolve around moving a spaceship through 3D space with few restrictions. The player can accelerate and change direction with the mouse. They can move the ship within the viewport with the WASD keys. They can fire laser beams, do barrel rolls, collect powerups, and do much more with just a few simple inputs.

DESIGN APPROACH:

I decided from the start of the project that I wanted the game to be arcade style (procedural generation with repeated attempts at the same level) and that I wanted it to be space-themed. I designed my game around the code, allowing the visual assets available to me to guide the visual direction. The mechanic of moving the spaceship is the centerpiece of the game’s design, as the most time and thought went into the ship movement code.





SUMMARY | RESULTS:

I've learned a lot about the game design process through this project, particularly about the importance of prototyping and making changes incrementally in order to maintain the intended scope of the project. I had several ideas for this game that unfortunately had to be set aside, however this allowed me to explore the core mechanics to their fullest in order to set a solid foundation for something that I can build upon further. I most enjoyed writing the code to implement the systems that I wanted to include. My next goal in the field of game design and development is to continue strengthening my programming skills. While I will continue to work with software such as Unity and Blender, I plan on resuming development on my own game engine in C++.

PROTOTYPE:

Initial Prototype Gameplay Video: <https://www.youtube.com/watch?v=0AhW1WhGCcY>



Final Executable: <https://jackpsmith.com/wp-content/uploads/2024/11/GalacticVortex.zip>

Jack P Smith

INTERACTIVE MEDIA

✉ jackpsmith@jackpsmith.com

✉ js99564p@pace.edu

HOME

[ABOUT](#)

[PROJECTS](#)

[CATAclysm](#)

[CONTACT](#)

[RESUME](#)

BLOG

[MAY 2024](#)

[APRIL 2024](#)

[MARCH 2024](#)

[FEBRUARY 2024](#)

[NOVEMBER 2023](#)

[OCTOBER 2023](#)

[SEPTEMBER 2023](#)

SOCIALS

[GITHUB](#)

[VIMEO](#)

[LINKEDIN](#)

