Solvaris Prime

Game Overview

Solvaris Prime is a roguelike twin-stick shooter with a sci-fi theme set in space. The player must fight through 5 waves of progressively stronger enemies randomly picked from a bank. This ensures each run is slightly different each time. Due to the fast-paced nature of the game, the player will need to use precise movement to dodge enemy attacks and effectively use their abilities to survive. After each wave, players must select one of three randomized upgrade cards, each offering different effects. On wave four, the player must choose two of four cards allowing the player to have five total cards for the final boss. On the final wave, a boss is selected randomly to fight the player. To win, the player must defeat the boss. The player will lose if their health reaches zero. An easier difficulty option provides extra lives for a more accessible experience.

Genre & Inspiration

Like traditional roguelikes Solvaris Prime emphasizes procedural variation with enemy attacks and upgrades making each run slightly different with the core mechanics being the same. Drawing inspiration from similar games like Slay the Spire and Risk of Rain. Both games incorporate defeating randomly selected enemies of increasing difficulty with a final boss at the end and collecting upgrades along the way. I have drawn inspiration from Slayer the Spire by using a card upgrade system, while Slay the Spire is a deck builder where you attack with cards, my game will only use cards for upgrades. Both games use levels or floors to separate fights between upgrading. I have changed this idea and incorporated a wave-based system inspired by the co-op mode from the plants vs zombies garden warfare games called garden ops. A combination of inspiration from these games will be the foundation Solvaris Prime will be built from, creating a unique blend of combat and adaptive progression.

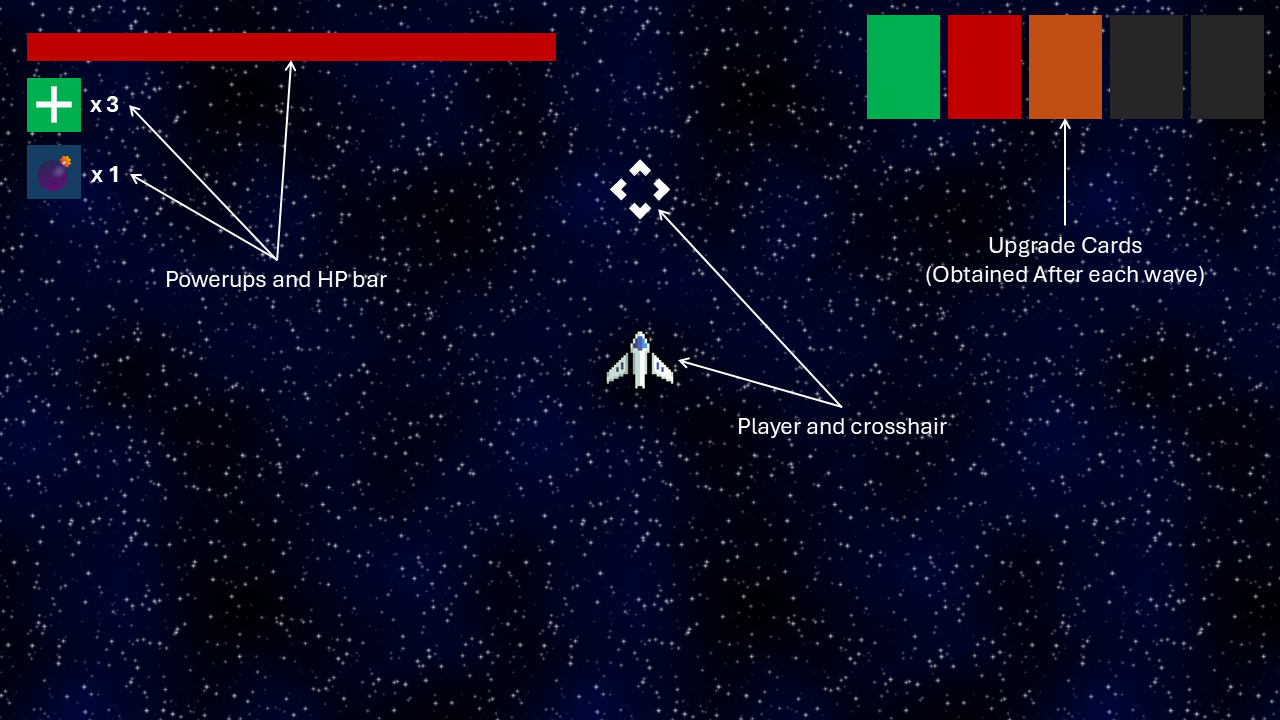
Intended Player Experience

Solvaris Prime delivers fast-paced gameplay that encourages players to think strategically. The roguelike nature of the game ensures every run is different, meaning a player cannot remake the same build every time. To ensure survival, the player must balance attacking with defensive movement which is the core challenge of bullet hell games which the idea also inspires from. The randomly picked card upgrades allow the player to make a different build every time to encourage different playstyles and ensure no two runs are the same.

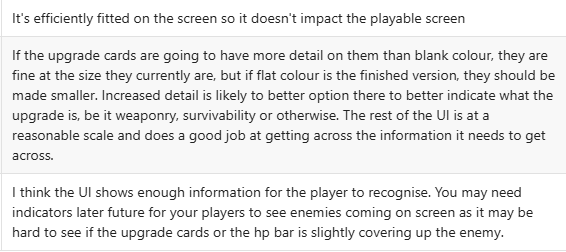
Tools

Planning

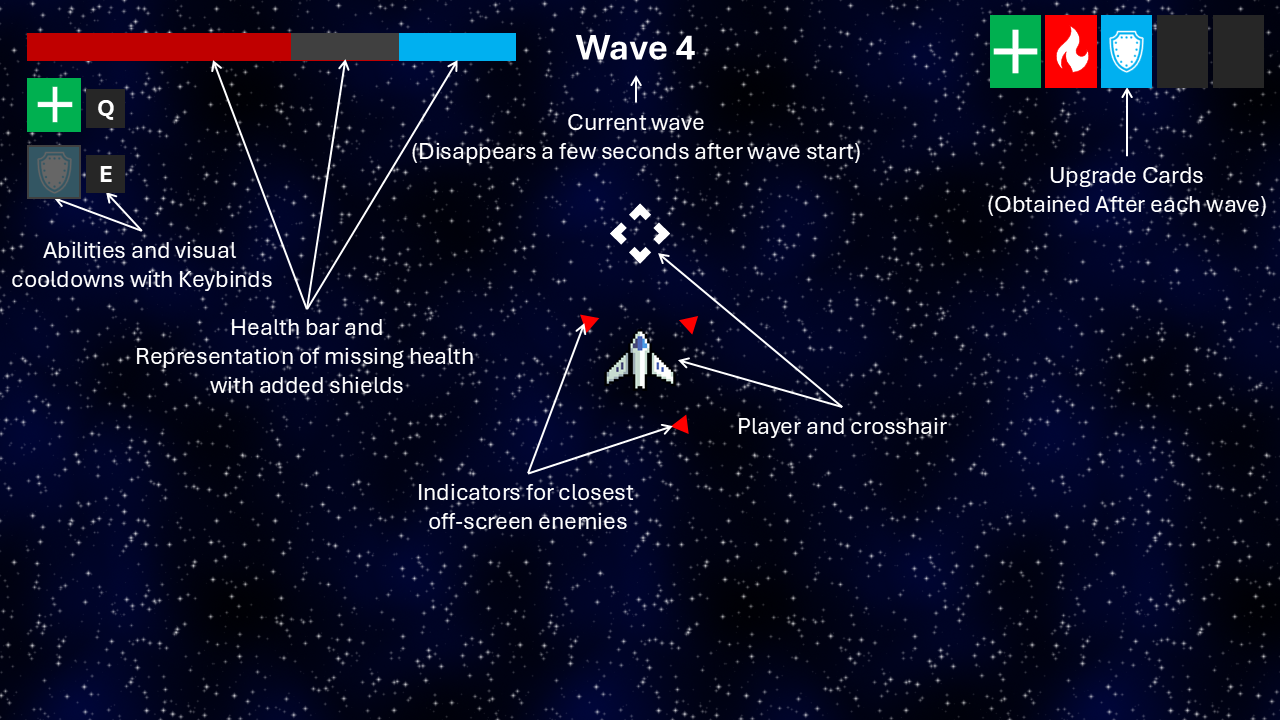
Due to the simplicity of the idea of the game, I wanted to make sure the UI had the correct information without taking up too much of the player's screen. I first came up with this iteration of the UI and put it in a form to get feedback on.



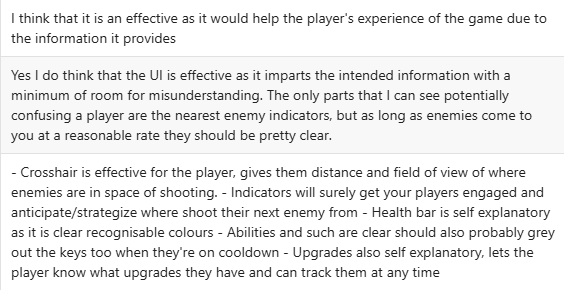
This was the feedback I got on the first iteration. The responses I got were happy with the size of the UI and the amount of information it displayed. There were some issues with it however, these were not having enemy indicators in case enemies were behind the UI and the play couldn’t see them and some missing details like the upgrade cards which only had colours and were a little too large.



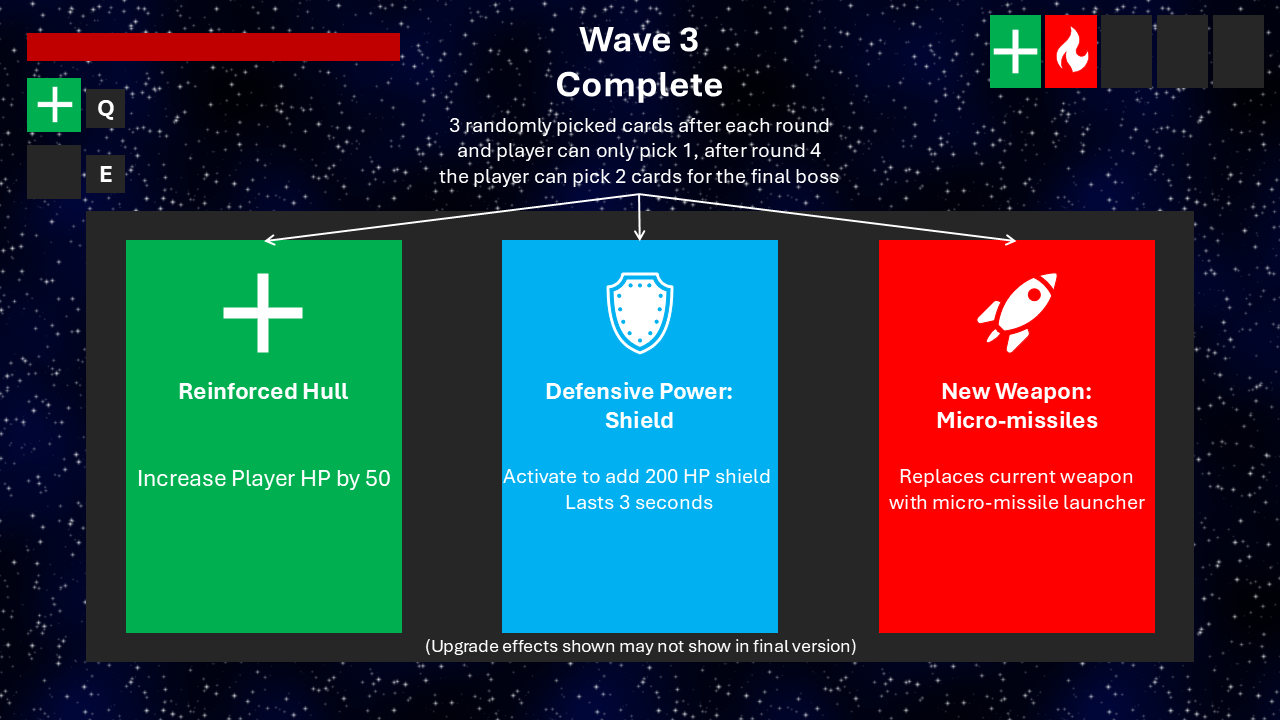
Based on the feedback, this is the second iteration of the UI. I added enemy indicators along with making some UI elements smaller like the upgrade cards and I also added symbols to them, so the player knows which is which. I also changed the powerups to be active abilities given by the upgrade cards. When an ability is on cooldown, it is greyed out a little along with the key bind to show its unavailable. In the final version of the game, there will also be a timer on top of the ability to show how long left until it is able to be used again. Along with this, I added text to show what wave the player is on which will fade out a few seconds after the wave starts and I shortened the health bar and added visual indicators for shields and missing health.



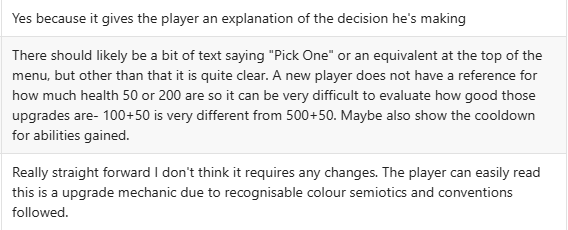
This is the feedback I got on the updated UI. Overall, the feedback was happy that the previous issues have been fixed. The only issues pointed out in this feedback were very minor and I fixed those for the final version.



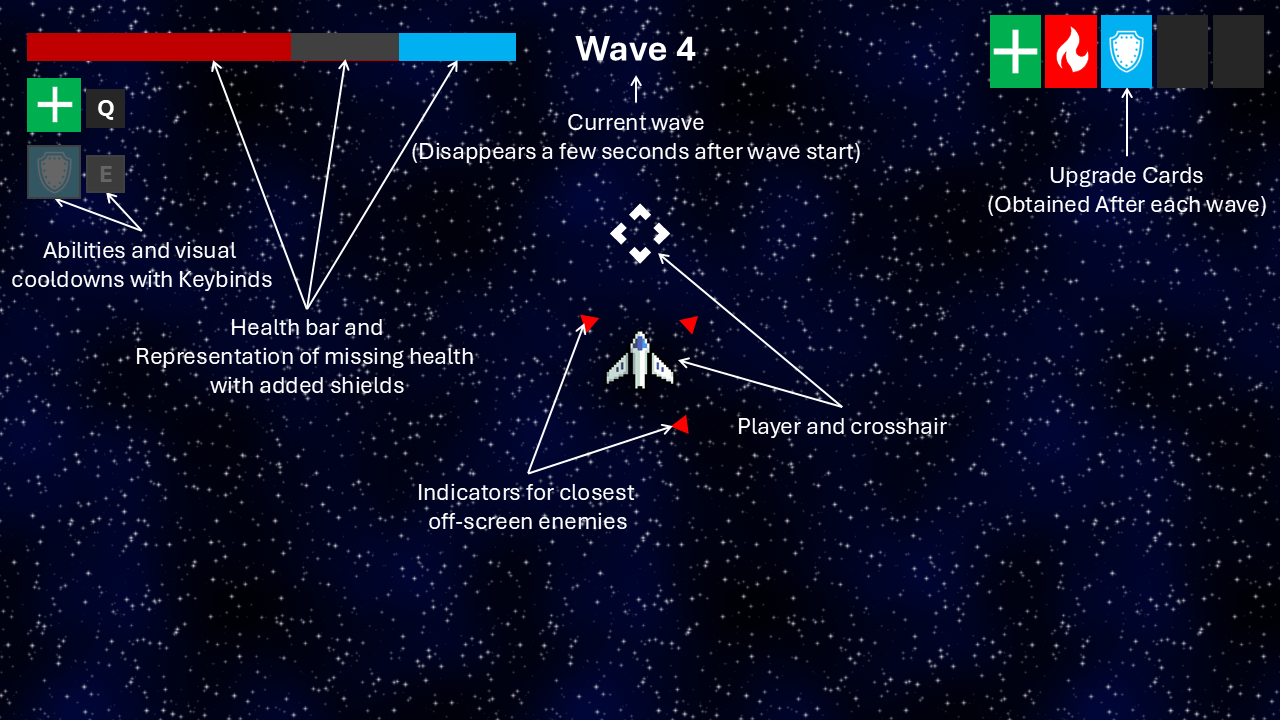
This is also when I created the UI for selecting a new upgrade card at the end of a round. I designed this part of the UI, so it did not clash with other parts of it like the abilities. I also showed what abilities look like when a slot is empty.

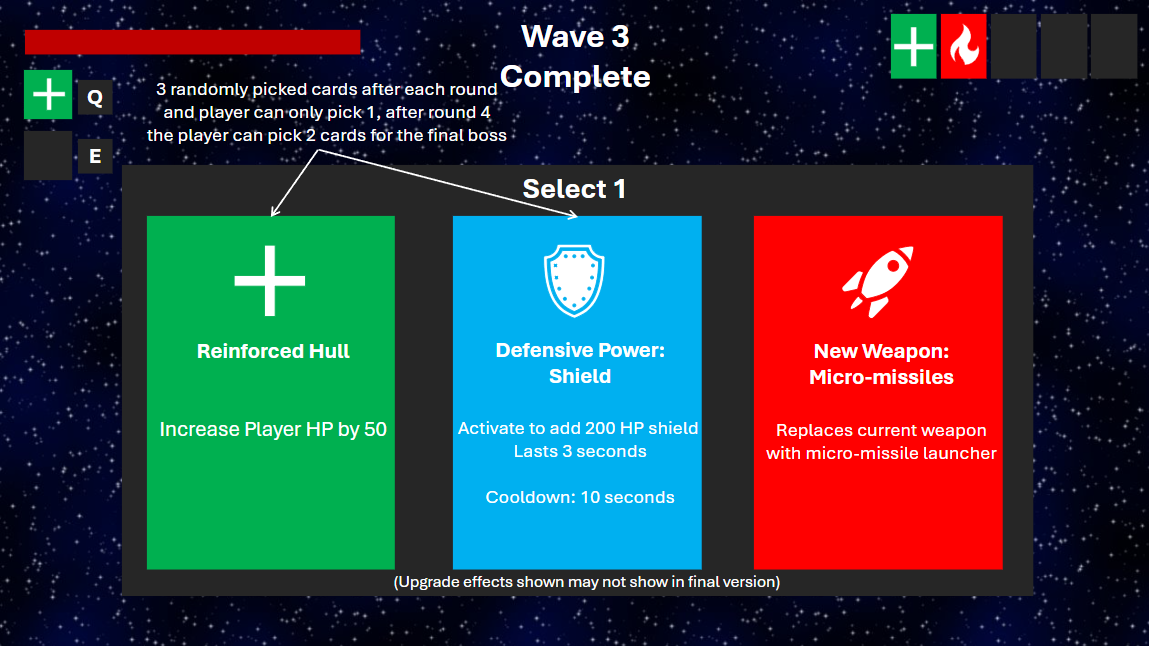


This is the feedback I got on the upgrade UI. The feedback was already happy overall and so I only needed to make very minor changes like adding text to say how many upgrades can be selected and adding text to show the cooldown of active abilities. The main issue pointed out was with the green upgrade, as I used an arbitrary number of 50 it did look misleading. This will be fixed in development of the game once I have found reasonable values and an upgrade like this may be changed to a percentage value rather than a generic number. This may make the upgrade look better as a percentage increase could look better than a set number.



From the changes of the feedback, this is the final iteration of the plan for the UI. This is what I will base the in-game UI from when it comes to development of the UI in production.





I have also considered the level design of the game, but I have not made a plan for it as the level will stay the same and will be mostly empty aside from invisible enemy spawn points. However, I will make sure in the project that the level is of reasonable size for the number of enemies on screen and in each wave. When the final boss spawns, it will always be in the centre of the level and will have a warning to the player to move out of the way to prevent any issues along with teleporting the player out of the way if they don’t move.

Throughout the project I used github to keep track of versions and updates and keep my project on the cloud so I could access files at home.

A screenshot of a computer

AI-generated content may be incorrect.

This is the script that allows the player to shoot, it takes in the prefab for the player and bullet along with some other public variables that can be changed in the editor to make it easier for testing and balance. This script also makes use of co-routines for attack speed, and it takes in a number which can be modified to change rate of fire.

A screenshot of a computer program

AI-generated content may be incorrect.

This is the script for player movement, this script allows the player to move in a top-down style and it also rotates the player towards the cursor so the player can aim to shoot.

A screen shot of a computer program

AI-generated content may be incorrect.

This is the code used by both enemies, they use the same script, but different numbers are used in the editor to change their damage and health along with move speed. The smaller enemy is faster but does less damage and the bigger enemy is slow and does more damage.

A screen shot of a computer program

AI-generated content may be incorrect.

If I were to improve on this project, I would add the scrapped ideas back in that had to be cut because of knowledge and time constraints.