



Search For A Star 2022

MyNamesLex

[Concept Outline](#)

[Initial Design and Research](#)

[Source Control](#)

[Design Plan](#)

[External Tools](#)

[Production Notes](#)

[Problems and Difficulties](#)

[Small bug fixes](#)

[Conclusion](#)

[Assets Used](#)

Concept Outline

My aim for this project was to make a game that is a creative take on the classic concept of minesweeper by adding a layer of randomness which has a certain degree of control that can

be mastered and also be easily picked up by people who are new to the game. I've been interested in making more unique games as of late as shown in the other game jams I have been a part of recently.
















Initial Design and Research

My initial idea was to create a game inspired by the concept of minesweeper, the idea I decided on was to have a dice that has an effect on the game depending on which side of the dice it landed on. This makes the game has an inherent degree of randomness yet allow for the game to be mastered by trying to get the side which the player wants to have by nudging and hitting the dice to the favoured side.

This idea came about by coming up with a way to make the concept of minesweeper more random and beginner-friendly.

Source Control

Source control was very useful to have to try out new ideas without having a fear of breaking everything in the project by being able to easily revert changes that broke the project, luckily, I never had to revert any changes through bitbucket.

Author	Commit	Message	Date
 Lexingto	d87c1f0	Shark lose cutscene	2021-12-22
 Lexingto	52a13a0	detect mine function cleaned up + explanation of it + ...	2021-12-22
 Lexingto	253d4ff	Detect mine function + can win	2021-12-21
 Lexingto	16a83db	Edited mainmenu island prefab	2021-12-21
 Lexingto	fefacb7	more environment floaters	2021-12-21
 Lexingto	9a1f1bf	Environment improvements	2021-12-21
 Lexingto	80e1b7f	Accessibility Camera Shake Toggle	2021-12-21
 Lexingto	610b6d9	Implemented accessibility functions	2021-12-20
 Lexingto	6e2e516	First Accessibility Value Added	2021-12-20
 Lexingto	5e00c01	.	2021-12-20
 Lexingto	d18f683	Starter for accessibility toggles	2021-12-20
 Lexingto	07ce56e	Improved teaching players on how to play	2021-12-20
 Lexingto	4d2b8f4	polish	2021-12-12
 Lexingto	93359b5	UI polish	2021-12-12
 Lexingto	ecb1638	added vfx + easier to tell what effects are + dice nudg...	2021-12-11

Design Plan

There were several ideas that I wanted to implement which I did manage to implement in time successfully.

I wanted the game to be visually pleasing and feel like an alive world that exists. No matter how small that world truly is, I wanted to have several systems that are easily expandable and some may be small but I feel that the effect they have is quite large.

To fit the time constraints and the concept I decided that I wanted the player to be in the middle of the ocean with the world passing the player by as the player plays the game since they are in the middle of the ocean I wanted the feeling and vibe of the game to be serene, peaceful and relaxing.

External Tools

I used audacity to help with creating sound effects utilizing the effects panel to help create better sound effects.

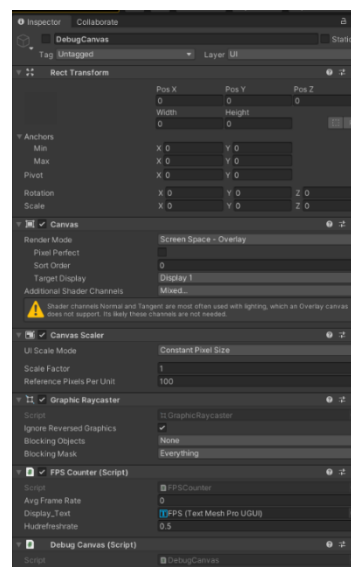
I used blender to create the dice model, I prefer using blender over other 3d modelling software because I find it easy to use and it has a wide range of plugins and effects to give the model that extra bit of polish.

I wrote down ideas I wanted to do on my whiteboard, I prefer this over using something like Trello when working solo because I find it easier to scribble down what I need to do when I think of it and have it remind me quite easily when I see it.

Productions notes

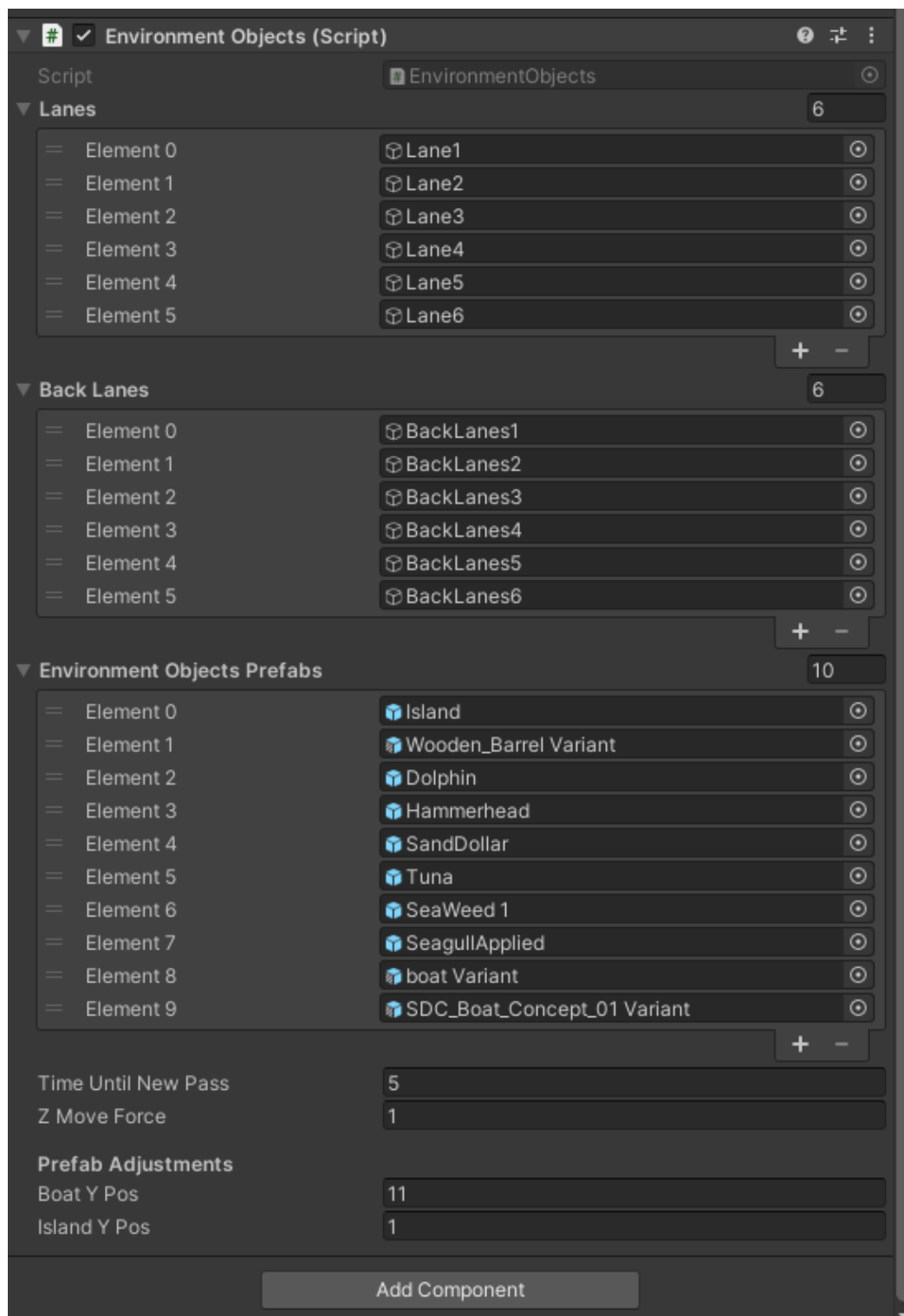
DebugCanvas

This is useful for debugging and currently only shows a frames per second counter and is not active in the main game. This script can be easily expanded to include other useful visual debugging tools when needed, this is primarily useful for playing the game when it is built and wanting to test out ideas to ensure they work.



EnvironmentObjects

This script is used to manage the gameObjects and spawn them at set spawns, this is used for the gameObjects that are not under the glass. This script works by randomly picking a gameObject from the Environment Object Prefabs and randomly selecting whether to set the transform to be in the Back Lanes or the front lanes and setting the move direction accordingly. I feel having different spawn locations and different objects spawning in different locations and moving through the scene helps the game feel alive.



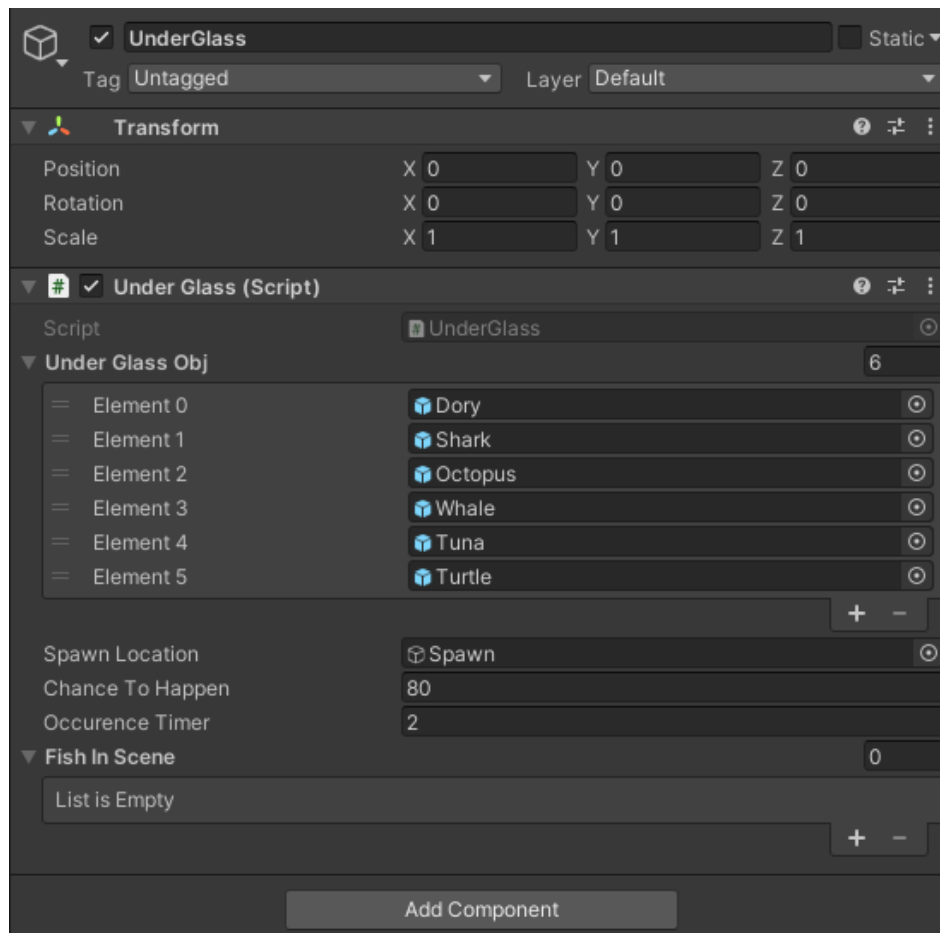
EnvironmentFloater

This script applies to the objects in the scene that move in the water and how it moves in the water, it is used by UnderGlass and EnvironmentObjects. I implemented this system because I wanted to make the environment of the game feel alive and having different movement systems and things moving around in the scene helps with this. I put the two different movement systems in the same script to make it simpler to add new prefabs which are required to move around in the game.



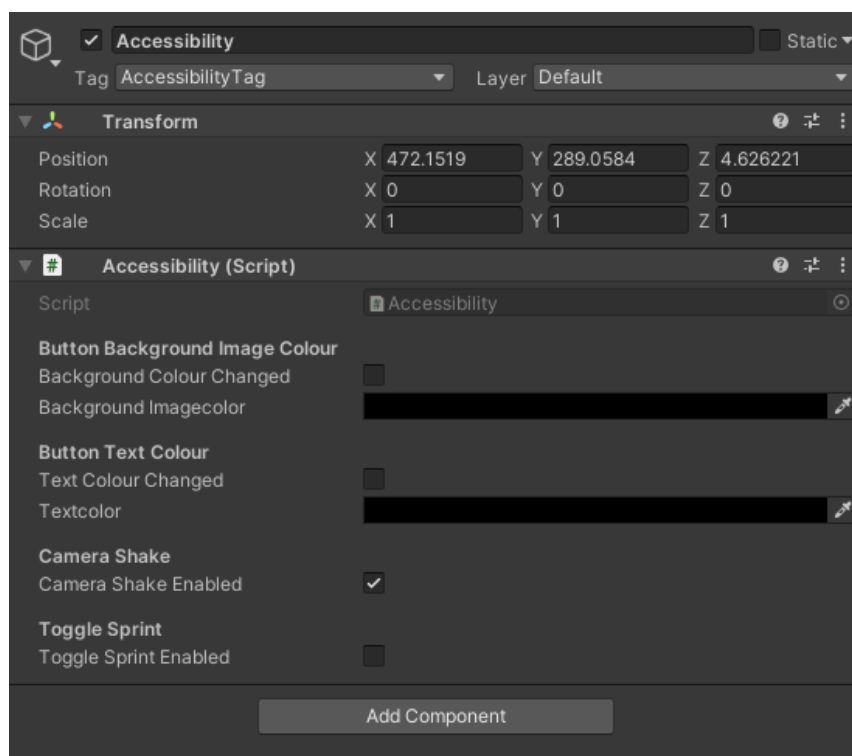
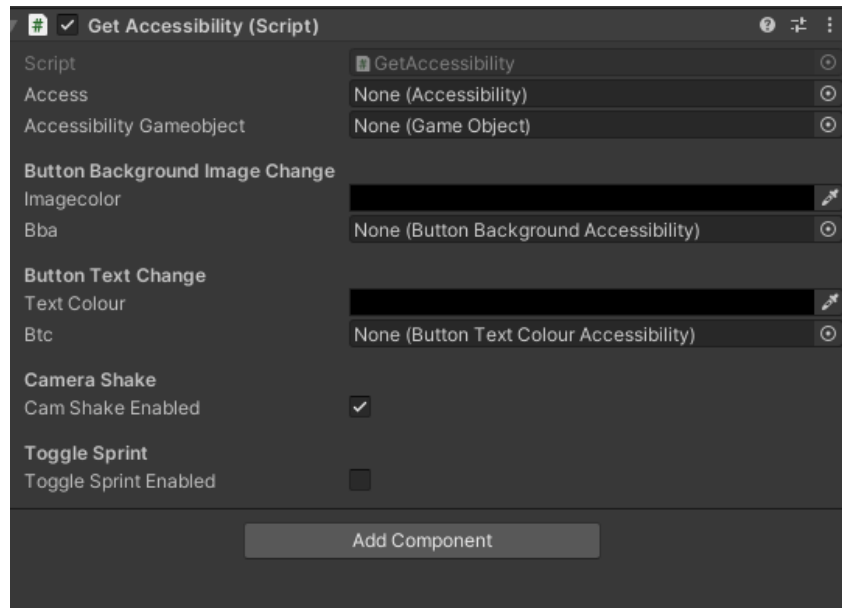
UnderGlass

UnderGlass is the script for the system to manage the fish which spawn and move underneath the glass. This system is in place to add to the ‘aliveness’ of the scene and make it feel like it is its own world by spawning a select few fish. The fish spawned are added to the list FishInScene and this list is used for checking if the amount of fish in the scene are at 50 or exceeding 50. If this is true, it stops spawning fish for performance reasons and also visual reasons. If there are too many fish in the scene with no limit and the player plays the game for a prolonged period of time, this can result in lag induced by the volume of fish. This also won’t look too good having too many fish below the player.



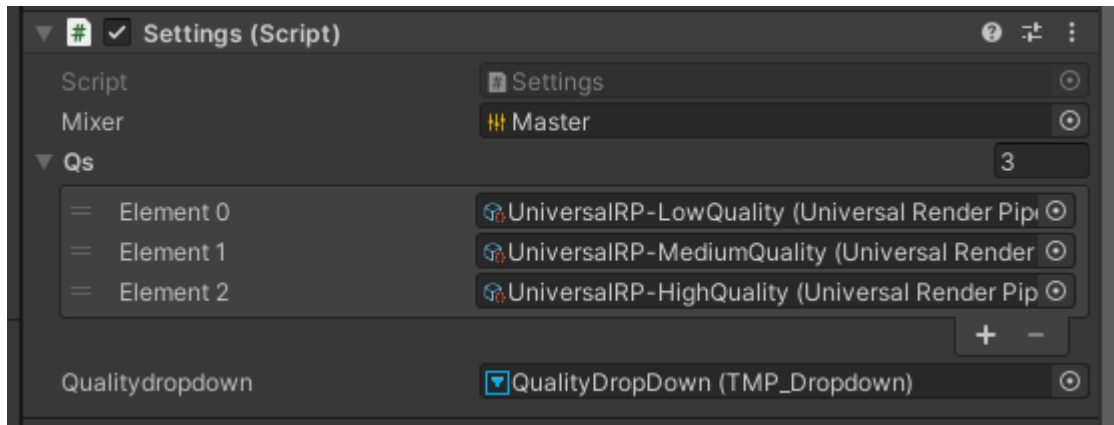
GetAccessibility/Accessibility

GetAccessibility is used for the individual scene to find accessibility and accessibility relating scripts apply accessibility effects by checking the DontDestroyOnLoad script Accessibility to see which accessibility options to apply. I implanted accessibility options to the game to allow for more people to play and enjoy the game.



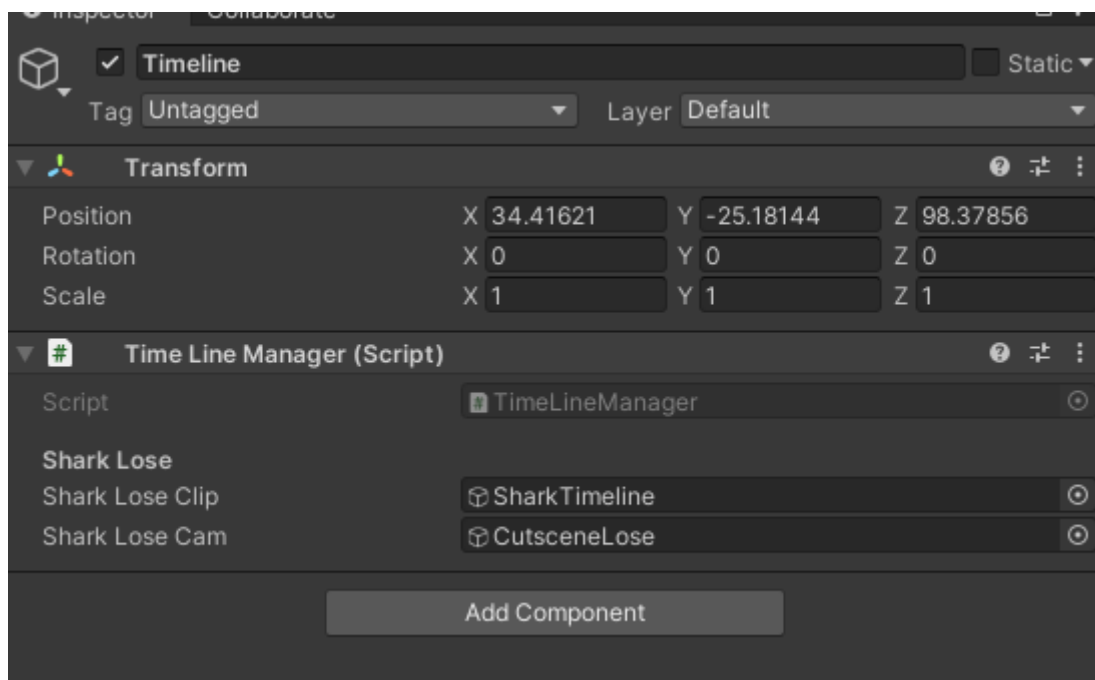
Settings

This script holds all the functions regarding the settings for the game including audio changes, fullscreen toggles and visual quality controls. I feel that this is fundamentally important to the game because it allows for more people to tailor the game experience to how they desire it to be by allowing this degree of control.



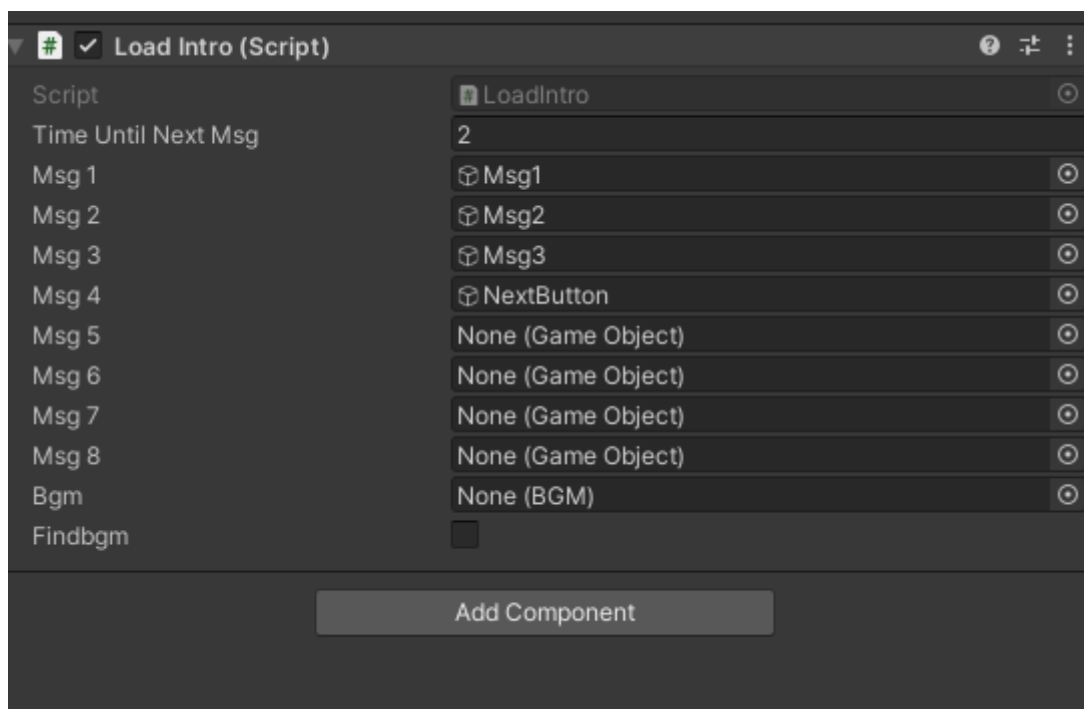
TimelineManager

This system is rather simple and involves the use of Unity Timeline. It only currently manages one cutscene since there is only one cutscene in the game but this is easily expandable. It enables the camera and other objects relating to the cutscene. I implemented this using unity timeline to add an extra layer of polish to the game.



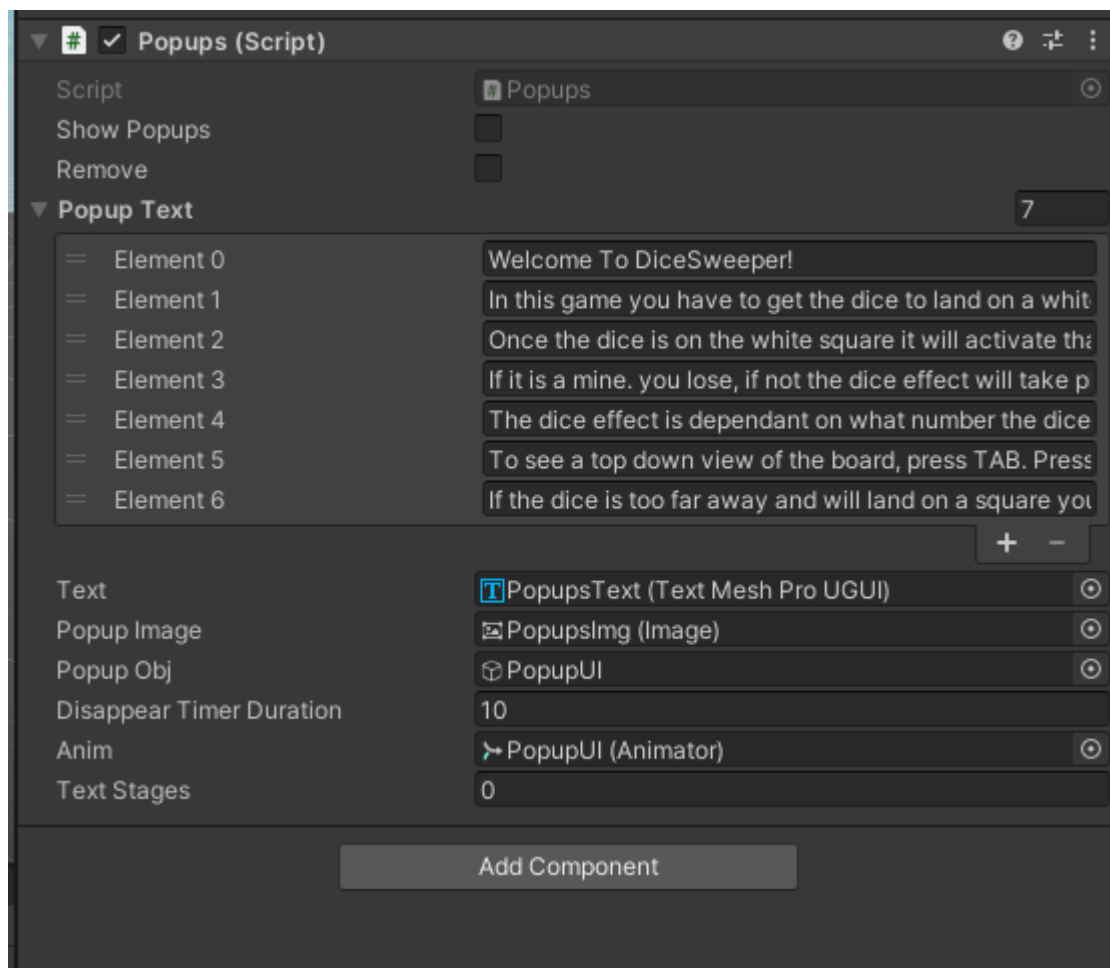
LoadIntro

This system allows for if the player has seen the intro before while the application is running through the boolean SeenIntro, if its set to false it will show the intro and then after showing the intro set that Boolean to true, If it is set to true it will skip this sequence, the sequence explains the game to the player and staggers the text to make it more readable. I could have made it so it writes to the system so that if the game has *ever* been played it will detect that through a file written to the system but I decided against this. This is because If the player has not played the game for a while they may forget a few things so to counter this I made it so that the system only detects if the game has been played once while the application is running, this is forgotten if the application is closed.



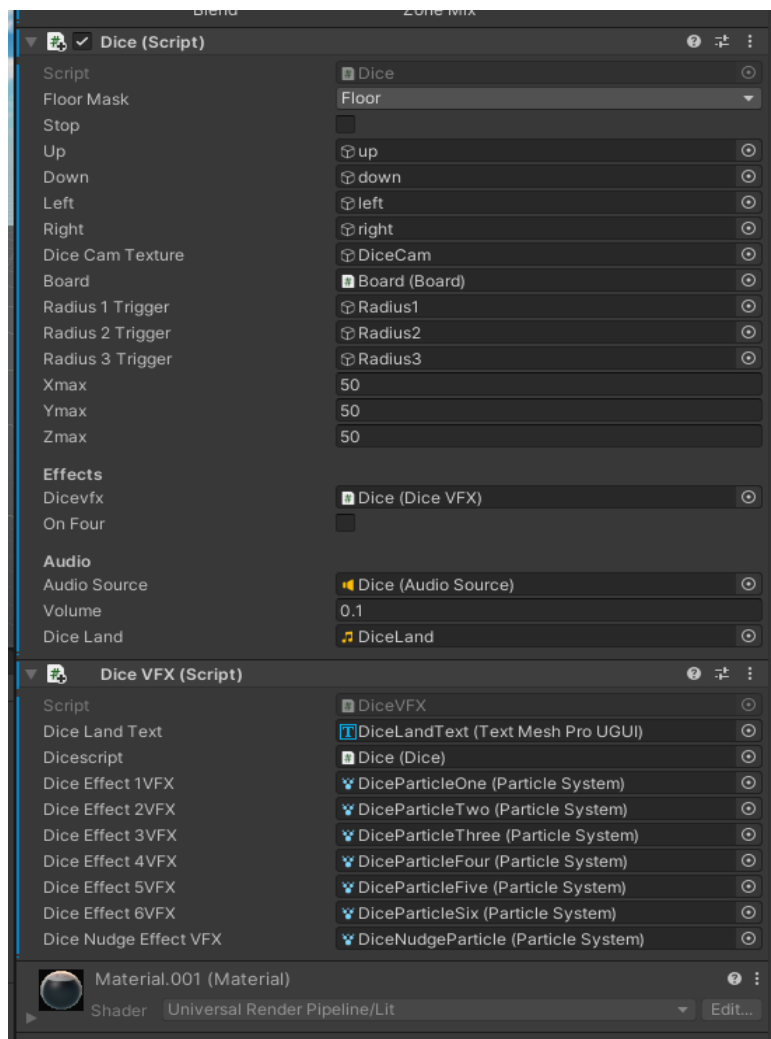
Popups

This system is similar to the Intro system. If the game has been played, it will set the HasPlayedSinglePlayer boolean to true, if not it will be false. If the game has not been played and if the game is loaded, it will show the popups further explaining the game. I wanted the game to be as easy to understand as possible which is why I have two systems explaining the game. The big block of text approach the intro has is not a good way to explain things but the player will likely remember some of the information due to the sentences being staggered, This is why this system is involved in the game to ensure they know how to play by having small blocks of text with eye-catching animations for showing and removing the popup from the screen.



Dice VFX

This system allows for an individual visual effect for when the dice lands on a side. Although, most of these are duplicates and I did this to simplify the game more. I made it so red colours means the square is a mine and white colours means the square is not a mine, this helps the game be more beginner-friendly. I also added a DiceNudgeParticle to add that extra layer of polish when the dice is nudged, it is a small effect but I feel it is a nice addition to the game.



Problems and Difficulties

Some problems I had with the project include an issue I had late into the project with the game crashing, I didn't know why it was crashing as it caused Unity to crash as well so I didn't have any debug help. I had to go into the LocalLow for the project in %AppData% to see the log's which helped me realise what was causing it to crash. It was a `NullReferenceException` for the `Radius1Trigger` which is used by the Dice when the dice lands on 5. This was due to the now-deleted `Timeout Coroutine` which set the `gameObject` to no longer be active which resulted in the `gameObject` being passed to the `Box.OnClick` to be null due to the script sending the information to the `Box.OnClick` no longer being active. I fixed this by removing the `Timeout Coroutine`.

Small bug fixes

Some small bug fixes I had to do included the boat spawning above the water which made me have to include prefab adjustments for the Environment Objects to adjust the y value's and potentially any other value would be easy to implement with the system In place.



Another small bug fix was regarding the UnderGlass fish objects not colliding with the points so they no longer moved to another point and waited until the timer run out to move. This resulted in the fish being stacked on one another in clumps where the points are located. I fixed this by using a BoxCollider trigger for the fish and a box collider with no trigger for the points.



Conclusion

I am very happy with this project as I feel it accomplished my concept of building on the concept of minesweeper and adding a layer of randomness and making it more beginner-friendly. I feel that this game is unique and fun to play.

Some aspects are not as well done but could be changed if I have more time. One of these things are is that there is only one environment, If I had more time I would have liked to do different environments and areas. There could have been a system to randomly select from a couple of made environments and play in that set environment when the user selects Start Game.

There also could have been different effects for different numbers landed on, for example, if the dice landed on one there could have been 5 different effects to be randomly chosen from. Adding more variety to the game and more randomness to the game. Also unique effects depending on the environment which would have been randomly chosen.

Assets Used

I used Unity's Universal Render Pipeline to have an increased graphical quality and also more control over the graphics qualities to allow for the player being able to lower or raise the quality in the settings

I used Milkshake's camera shake engine to create a camera shake easily
(<https://assetstore.unity.com/packages/tools/camera/milkshake-camera-shaker-165604>)

I used a water shader for the ocean (<https://assetstore.unity.com/packages/2d/textures-materials/water/simple-water-shader-urp-191449>)

I used Unity Timeline to do cinematics

I used TextMeshPro to alter the text more

Audio Assets Used:

Abstract Glitch Ambient by Ivymusic on Pixabay

Beauty Chill Future Bass by Ivymusic on Pixabay

Fashion Chill Future Bass by Ivymusic on Pixabay

Jump - <https://soundbible.com/1898-Spin-Jump.html>

Visual Assets Used:

Sand Texture - <https://www.sharetextures.com/textures/plaster/brown-sand-plaster/>

Grass Texture - https://www.sharetextures.com/textures/ground/grass_1/

Tree - <https://www.gameassetsgarden.com/assets/3d-model/nature/trees/maple-tree-1-green/maple-tree-1-green-v1.php>

Wooden Cart - <https://www.3dmodelscc0.com/model/wooden-cart>

Barrel - <https://www.3dmodelscc0.com/model/wooden-barrel>

Marine Animals Pack - <https://rkuhlf-assets.itch.io/aquatic-animal-models>

Skybox - <https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-series-free-103633>

"1 seagull" (<https://skfb.ly/6WYK7>) by vicente betoret ferrero is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).

"boat" (<https://skfb.ly/onQ8B>) by Damco is licensed under Creative Commons Attribution (<http://creativecommons.org/licenses/by/4.0/>).

Font:

Logo font [https://www.dafont.com/retrogaming.font?l\[\]=10](https://www.dafont.com/retrogaming.font?l[]=10)

(Thank you for your consideration and thank you for looking into my game)