IKIGAI PROCESS DOCUMENT

Motivation:

Our main motivation for the project was making a 3-D game using Unity Engine. We wanted to create an Open World Explorer getting inspired from No Man's Sky and Minecraft. While looking into minecraft we got to know it used perlin noise to create realistic terrain. We wanted to create procedurally generated terrain using Perlin Noise. We then expanded the scope of the project to include different noise generation techniques such as OpenSimplex and a in-house random noise generator that generates realistic terrains.

A 2-D noise map was created using techniques such as Random Noise, Perlin Noise, Open Simplex and an in-house implementation. Using this noise map a mesh was generated using vertices based on the output of the 2-D noise map. The world was then populated with different kinds of clouds of varying shapes and scales. These clouds are placed randomly on the map. We added water based on the percentage of the land we want submerged under water. Trees were then added randomly on the mesh.

To traverse to different worlds we created a portal land in which different portals lead to different meshes associated with different underlying algorithms. Sound and menu was added on top. We used meditative sounds because of the explorative aspect of the game.

At last due to user feedback we added coin and a score to the game so that users have in interactive element apart from the exploratory aspect of the game.

Play Test 1:

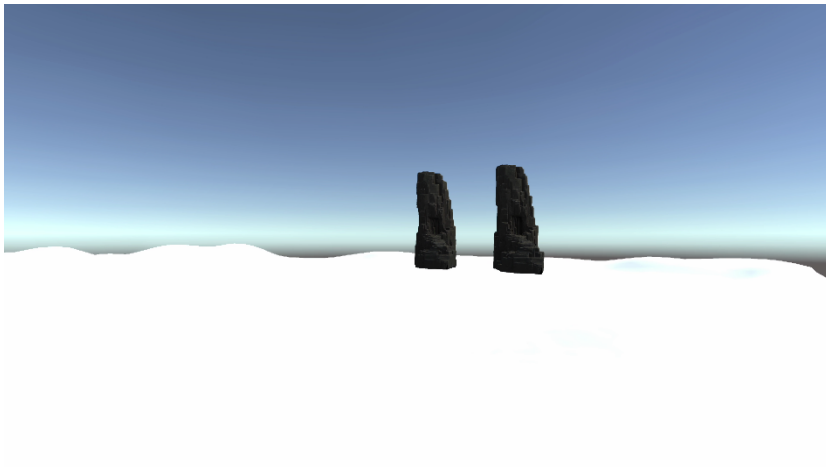
Feedback Notes:

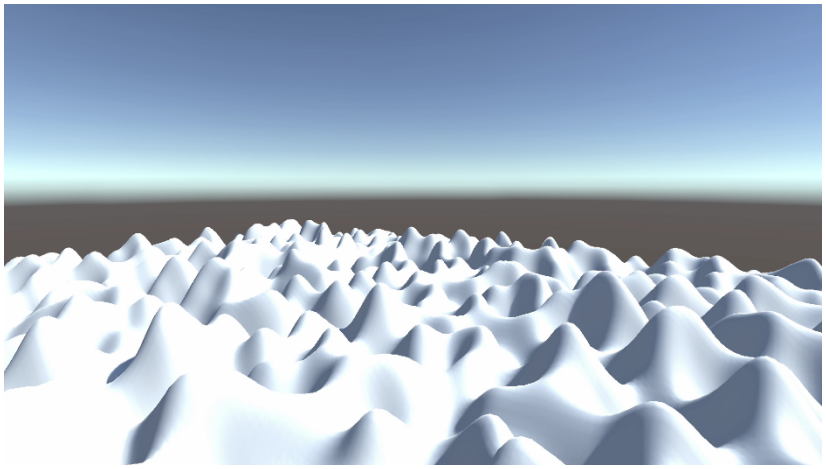
a. Give 3D motion
b. Idea of mentioning water, Mountains, clouds
c. Camera movement, Try to add movement in 3 axes.

Feedback / Learnings:

Overall the feedback was positive but a few critiques were related to aesthetic aspect of our game. In this playtest we got the feedback that our environment and terrain were very bland and barebones, there was no player object and our terrain had no features. Our portal landing was incomplete. The playtesters appreciated our idea of the generation of terrain using noise. They gave us feedback that we should add features to our terrain and also add a player object.

From this playtest we got to learn that users want a lot of aesthetic visual elements in any game they play.





Play Test 2:

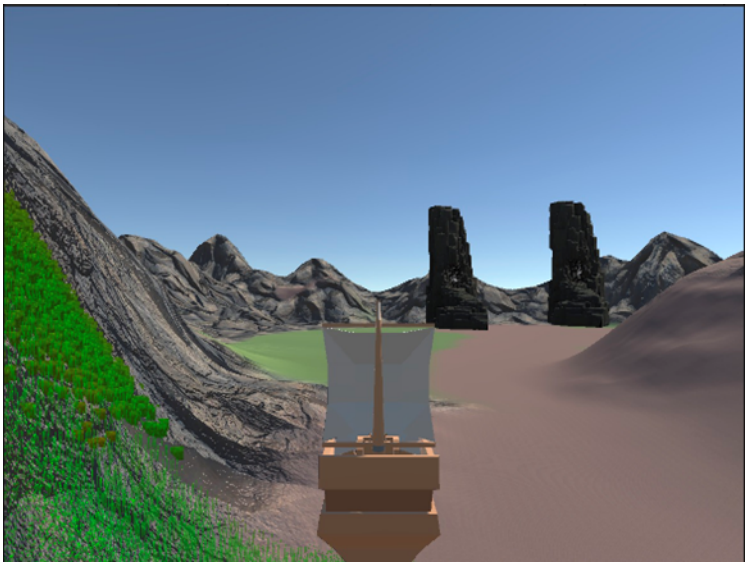
For this playtest we improved on our gameplay after taking feedback from playtest1. We improved the portal landing page. We added features like water and added colors to our terrain. We added a player object (a pirate ship).

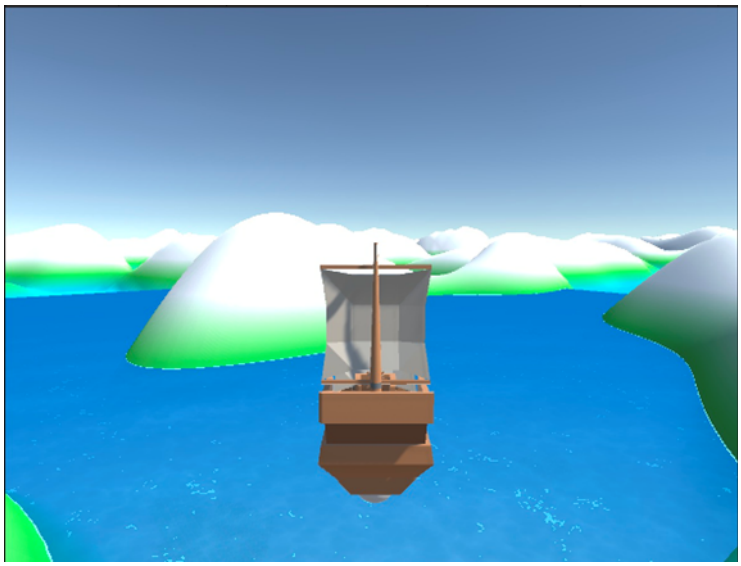
Feedback Notes:

a. Add brown color to land
b. Add other components.

Feedback / Learnings:

In this playtest the feedback we got was that we should add brown color to the mountains. We were also told that we should add some objects to the terrain to improve the gameplay experience. The learnings from this playtest were that users want some objects in the game like trees and clouds.





Play Test 3:

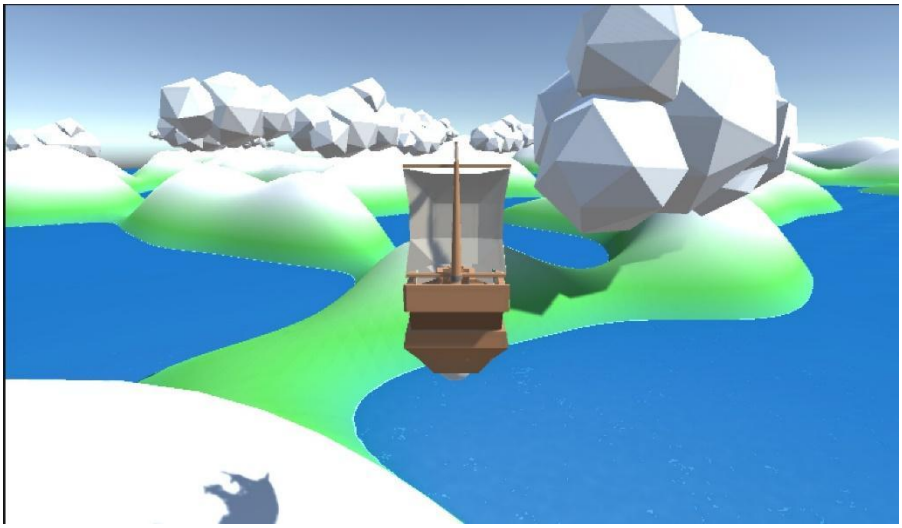
For this prototype we took feedback from the last playtest and added clouds. The clouds were such that player object could go through them.

Feedback Notes:

a. Idea of landscaping is unique.
b. Objects with diversity can be added

Feedback/Learnings:

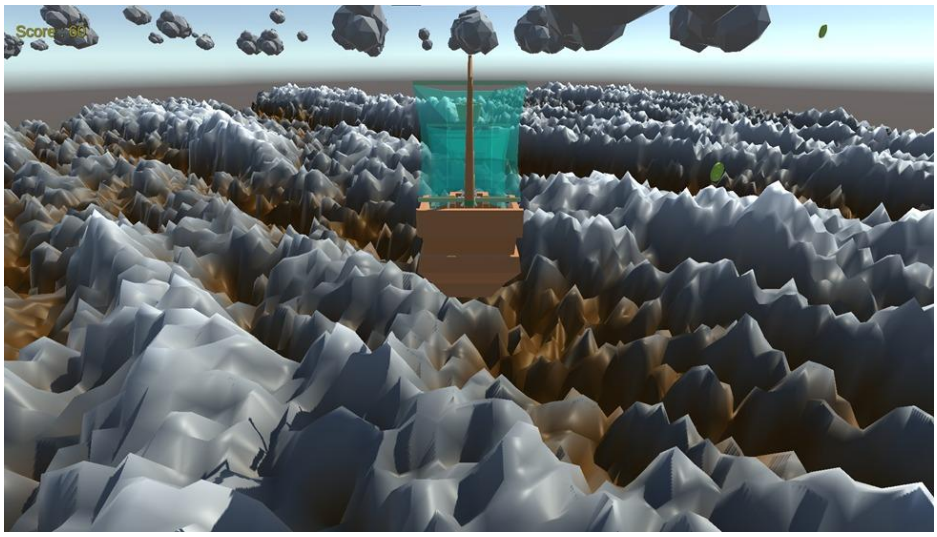
The feedback we got was that we should add different objects.



Final Prototype:

From all the feedback we received we wanted to add trees and interactive elements like gold and a scoring system. We came across the issue of not being able to add trees and gold with terrain. We decided to switch to meshes.

We added randomly placed trees and added interactive elements (gold with a scoring system).



Questions:

1. Look back at your 3 goal tiers from the project plan (minimum viable product, meets-expectations, and exceeds-expectations). Where does your final prototype land with respect to them?

A: With respect to our 3 goals our final prototype lands in the “exceeded expectations”. We added interactive elements (gold and score tracker). We implemented X procedurally generated noise algorithms:

   a. Random Noise: Totally random heights for an initial terrain to built upon (this is the worst possible terrain and we wanted it to be shown to shed light upon the power of better techniques such as Perlin noise)
   b. Restrictive Random Noise: Our own random noise implementation which aims to generate better noise than random noise.
   c. Restrictive Random Noise: Another version of restrictive random noise(our own implementation) with steeper slopes
   d. Perlin Noise: Procedurally generated gradient noise.
   e. Perlin Noise with Octaves: Octaves were used to generate Perlin noise for better noise maps.
   f. OpenSimplex: We took an off the shelf OpenSimplex implementation to drive our mesh generation algorithm

2. What unsuccessful designs did you explore, and what made you decide not to pursue them?

A: At the beginning, we wanted to carry out the noise to heightmap implementation for which we used terrains in Unity. After implementing it on terrains we found that placing objects (adding elements like trees and gold) was extremely difficult using terrain. Hence, we switched to meshes, which are superior to terrain since we can control every vector in terrain.

3. What did you learn from working with play testers?

A: Working with play testers gave us a lot of insight into what players expect when playing an explorative game. Players give a lot of importance to the visual fidelity of the game as well as some interactive playable elements. The game needs to be visually appealing and interesting to play. We took inputs from them and added a player object (a pirate ship), and gave features to our terrain (trees, water, gold). We also added interactive play by adding gold and a scoring system.

4. What technical challenges did you face, and how did you overcome them?

A: Getting familiar with Unity was the first challenge we faced. Unity is a complex ecosystem and getting the basics right is quite a time-consuming task. The next challenge was the C# language, all of us were new to the language. The next challenge we faced was object placing. Originally, we had used terrains in our project, but we could not get around to object placements using terrains. As a solution to this problem, we used Meshes.

5. How does your final prototype differ from your initial concept?

A: Our initial concept was very barebones and bland (according to play testers) in that we focused on generating a 3-D terrain using a 2-D noise map using at least one noise generation algorithm. We took inputs from all the playtests. In our final prototype, we have a player object with a 3rd person view, our terrain has features like water, trees, etc. We have also added interactive gameplay (collecting gold coins).