



## Game Description

*Above The Clouds* is an action game set in a dark, ominous world of floating islands inhabited with a variety of terrifying enemies. However, you play as a goofy-looking, funny baby who struggles to run around and swing his weapons to kill things. This juxtaposition makes for a hilarious result that is fun to play. Your goal is to defeat enemies, collect weapons and items, and clear islands on your way to defeat the final boss. There are many cool technical features along the way that make this project something we are all proud of.

## Player Ragdoll Physics

The player character utilizes an active ragdoll to move in the game world. This gives the player character a humorous look and allows our game to stand out amongst other games that use non-physical animated models.



The ragdoll actually consists of two models: one that plays a predetermined animation and one that only moves based on a series of configuration joints. Each frame the joints on the physical model will apply forces to attempt to duplicate the transforms of the animated model. The movements of the physical model propels itself forward to match the input given by the player. The picture below shows the physical model on the left and the animated on the right.

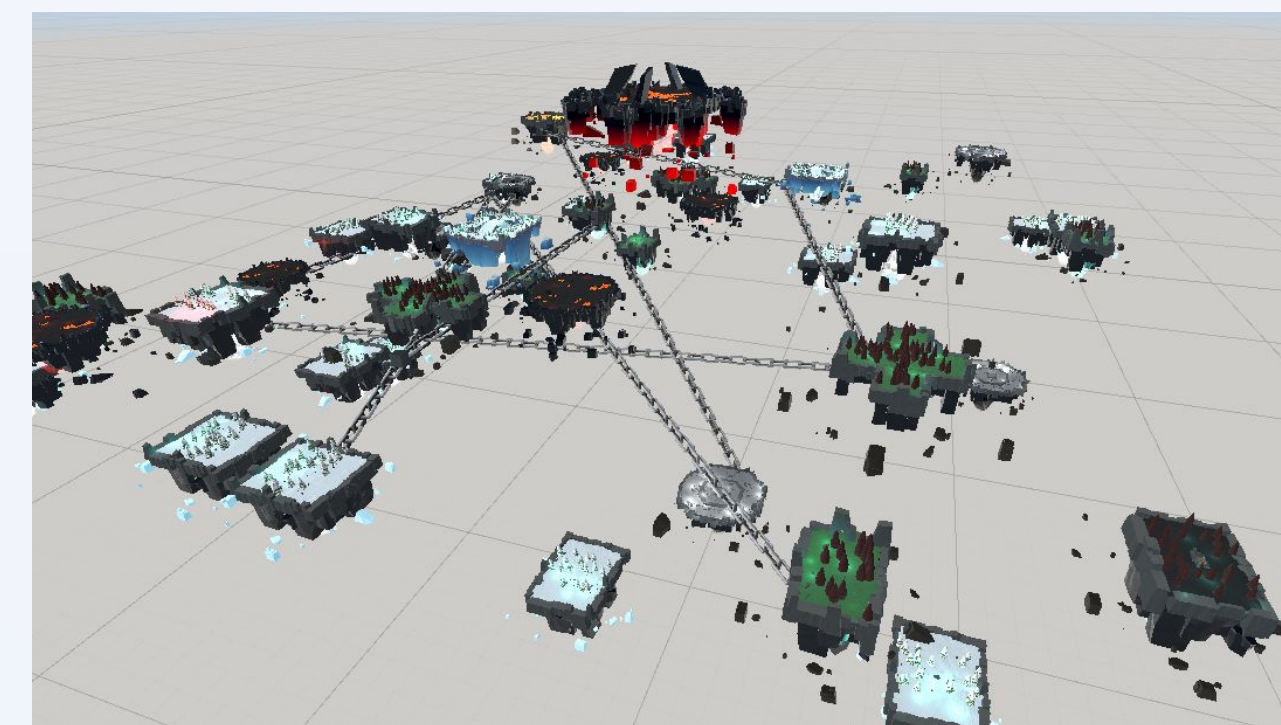
## Dodge/Jump

We created a dodge feature to allow players to quickly evade attacks from enemies for when the standard movement of the player isn't fast enough. When the player gives the dodge input, the player will be forced both upwards and in the current direction of movement. Dodging while still will act as a jump straight up into the air. To add humor and make the dodge animation more natural, the joints of the model stop applying external forces to the model when dodging, turning the player into a standard ragdoll. The player can toggle the active ragdoll and standard ragdoll with a button input.



## Floating Island Generation

We created a system to generate levels based on existing prefabs being placed randomly within a specific area. To do this Islands are given a size of length and width proportional to their actual size. A two by two empty array is then used and filled up using the sizes of islands. A specified number of attempts are made to place islands down into the array only placing islands that fit in the empty spaces. This allows for islands to be randomly placed without clipping into one another. Finally, the height at which islands are placed is randomized to get the level generation you see below.



## Island Traversal with Cannon

Since some islands are far apart and not connected to each other, we wanted a fun way for the player to travel island to island. We decided to create a *Cannon* that the player can use to launch themselves from island to island.

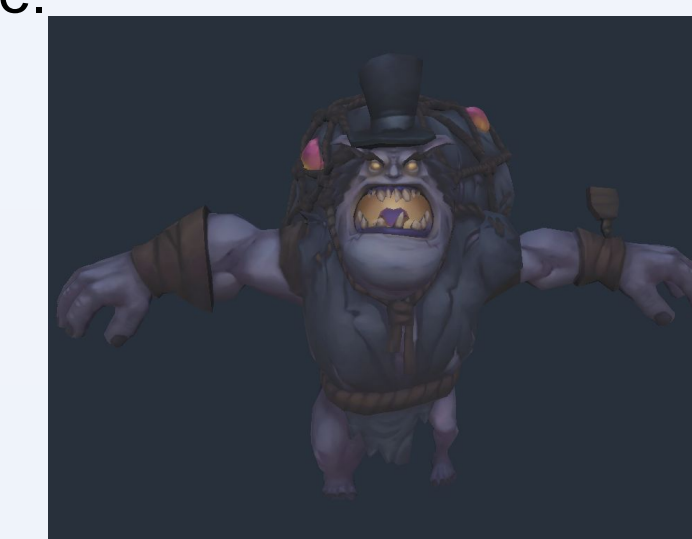
We added a trajectory line for the cannon so that the player can know where they will land. Since you can shoot yourself across such a far distance, it can be difficult to see where you'll actually land. To solve this problem, we made it so the trajectory line turns **green** when it's hitting an island's surface, turns **yellow** when it's hitting a different part of an island, and turns **red** when it's not hitting anything.



## Enemy AI

Our normal monsters use the Unity's NavMesh for pathfinding. Since we use procedural island generation, we needed runtime NavMesh building and implemented it by using the provided component in Unity's experimental AI navigation package.

We created 4 normal monster types (Basic ghoul, Bomb thrower, Self-destructer and Mine planter) and they basically have 3 states: 'Patrol', 'Chase', 'Attack'. While patrolling, monsters try to find the player by checking intersected colliders using `OverlapSphere()` method. When they detected the player, they turn into chase state and follow the player. If the player is in attackable distance, monsters turn into attack state and begin to attack. At the end of attack animation, `EndAttack()` method is called and they turn into chase state again. On the other hand, the Boss monster does not use navmesh. It always know the player's location and it shows different behavior depending on the distance between the player. When the player is too far, it turns into Ready state and then performs dash attack. If the player is nearby, it performs spin attack and then faints for a short time.



## Combat System

Our combat system makes use of *Inverse Kinematic* arms controlled with the mouse. The player can hold a weapon in either hand and swing their arm by holding either mouse button and moving the mouse. The player can then simply kill enemies by hitting them with the swinging weapons.

Since both *Looking around* and *Combat* are controlled by moving the mouse, we had to lock the camera in place while the player is using weapons. To improve this, we created a *Targeting System*, where the player can press a button to lock the camera onto a specific enemy, which is then highlighted.



## Shaders

We created many shaders with Unity's Shader Graph to enhance the visual experience of the game. Our *Blood and Guts* shader was created to visually show how damaged the player or an enemy is by connecting it to their *Health* property.

Our *Dissolve* shader was created to get rid of an Island when the player has cleared it. When the player leaves an island after clearing it, it will dissolve and fall away into the void.

Several other shaders were created to enhance the look and feel of the game including flowing water, lava, a force field, an outline/highlight, etc.

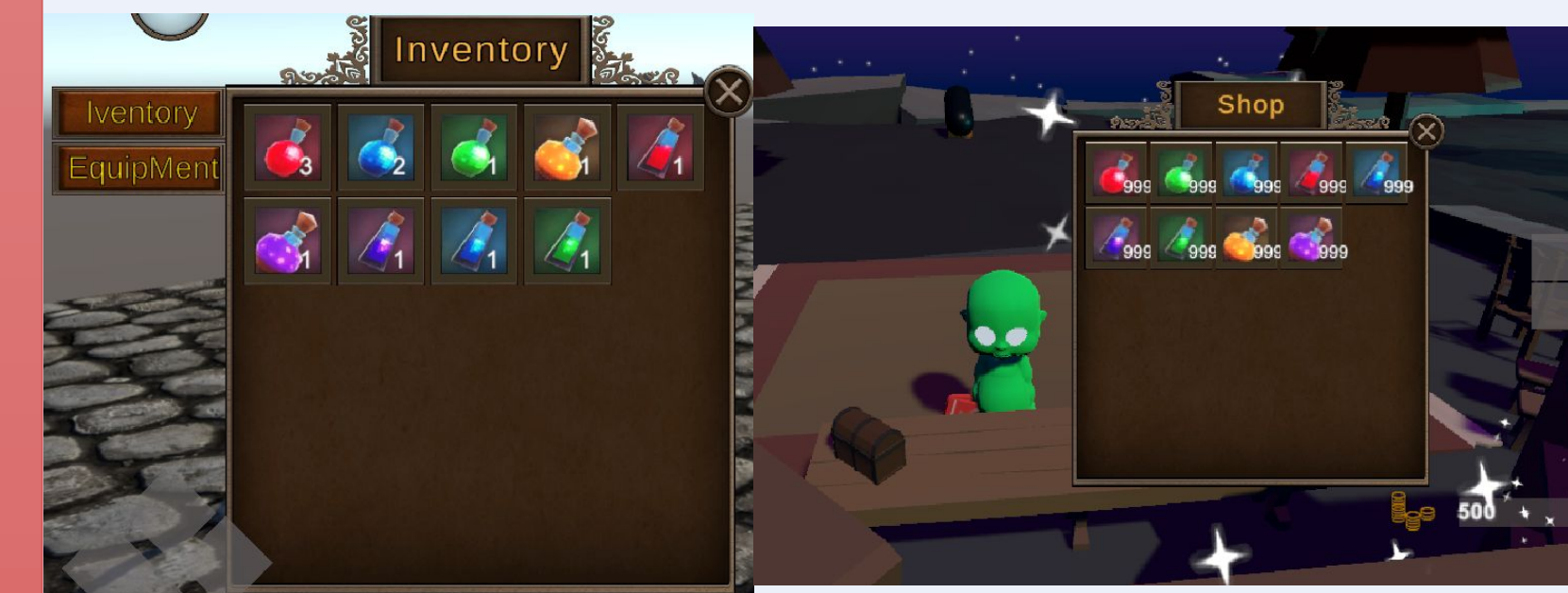


## Shop & Inventory System

We created an inventory system and equipment system to allow the player to collect items and weapons. In addition, a shop system was added for the player to buy new items. The player can kill the enemies to get the coins, and they can use the coins to buy items in the shop. You can also kill the boss which drops a chest containing random items and coins.

You can stack multiple items at the same time, because we have a number menu for items. For the inventory system, we also added an item cooldown, so the player cannot continue using items, which increases the difficulty of the game. We also added a cool effect of using different items, like a *size* potion can make the player larger.

For the Equipment System, we allow the player to equip two weapons at one time, and the player can switch between these two weapons in combat. Of course, you can also change what two weapons you want anytime, but you have to open the Equipment System to change. In addition, there is a weapons bar in the HUD to show which two weapons you have equipped now.





## Game Description

*Above The Clouds* is an action game set in an dark, ominous world of floating islands inhabited with a variety of terrifying enemies. However, you play as a goofy-looking, funny baby who struggles to run around and swing his weapons to kill things. This juxtaposition makes for a hilarious result that is fun to play. Your goal is to defeat enemies, collect weapons and items, and clear islands on your way to defeat the final boss. There are many cool technical features along the way that make this project something we are all proud of.