

Alpha Release Report

Slippery Bash

Team Ice Guys

Akbar Suriaganda

Moritz Naser

Omar Ahmed

Tarek Elsherif

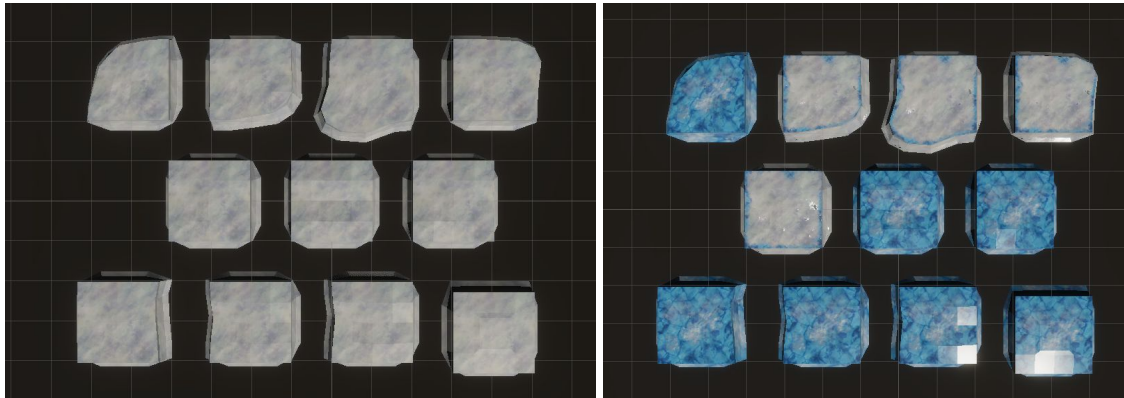
Alpha Release Report	1
Art	2
Character Animations	3
Platform Physics	4
More Tile States	4
Dynamic Buoyancy for Sub-platforms	5
Physical Materials	5
Sound	6
UI	7
UI Visuals	7
UI Controls	8
Camera	8
Parsec	9
Physics	9

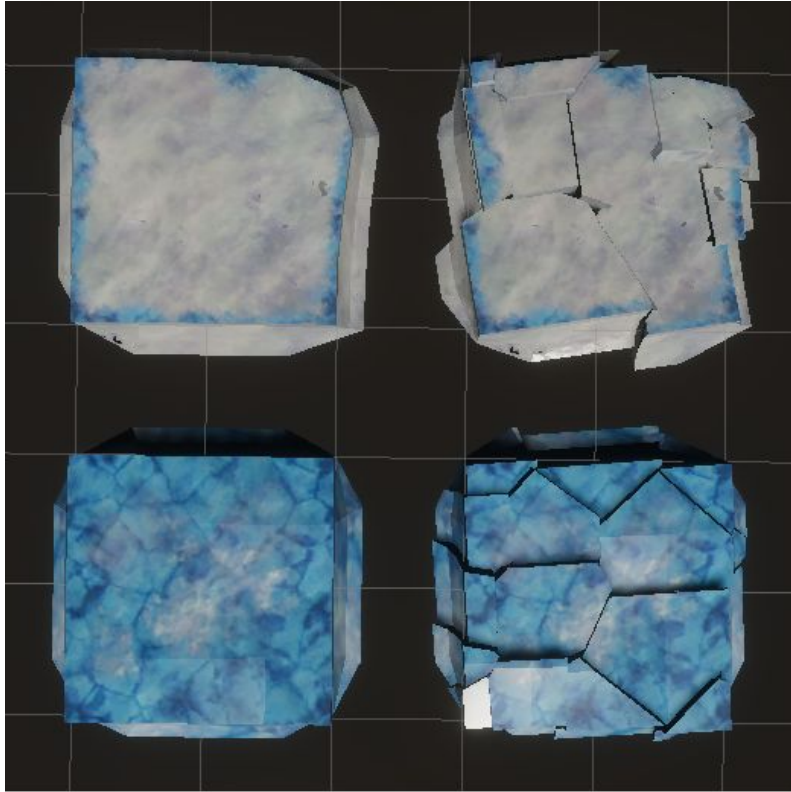
Overall Progress

Art

Level Design

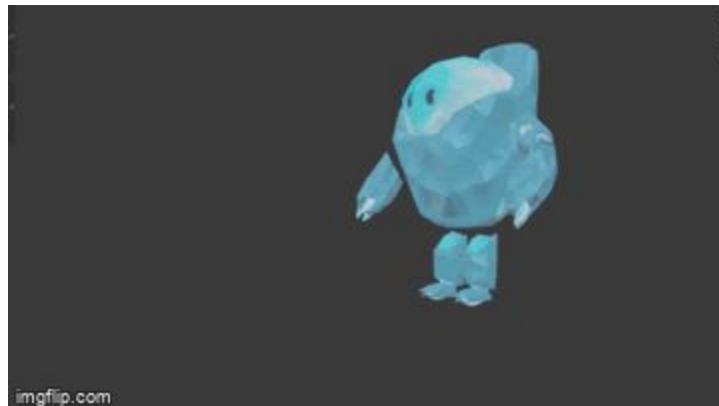
According to our paper prototype we planned to separate our island into small tiles which are then either snow or ice. That makes it easier to apply the slipperiness effect to the player character and due to the tiles system the island can be separated easily. However it was quite a challenge to have the tiles to have the snow/ice tiles blend nice into each other and also to make the tiles a little bit diversified. To accomplish this, we implemented a modular design for the tiles where we have 3 different types of tiles that are in the middle of the island with no contact to the ocean, 4 different types for each corner and edge tile. Each tile is individually fractured in blender. The random assigned material of the tile at the start of the game is then assigned to all broken tiles as well.





Character Animations

While testing the game loop we recognized that the hit animation was too slow and the impact of the animation was too weak. So we redone the animation completely and come up with the following:



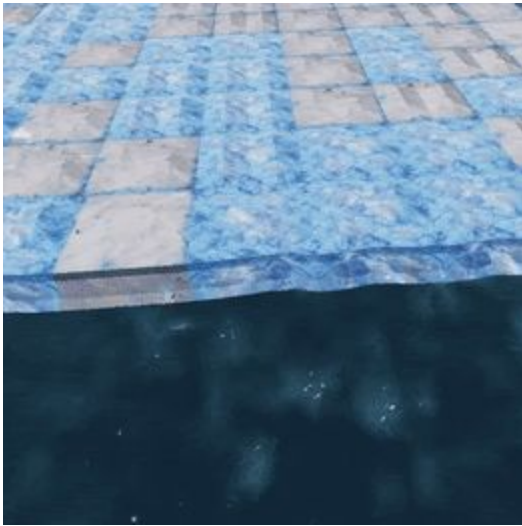
Platform Physics

More Tile States

One more state was added to the tiles of the Island platform. There are three states now:

1. Normal tile
2. Cracked tile
3. Broken tile

All the tiles start as normal tiles and they take up to two hits. The first hit will crack the tiles.

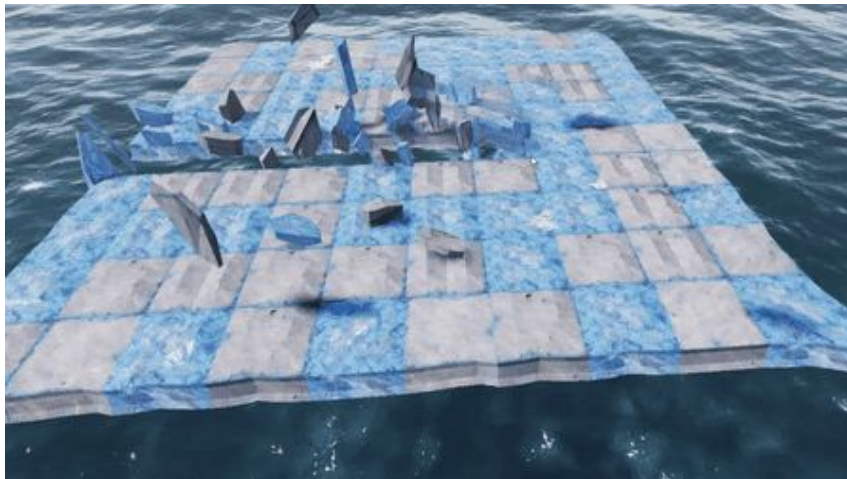
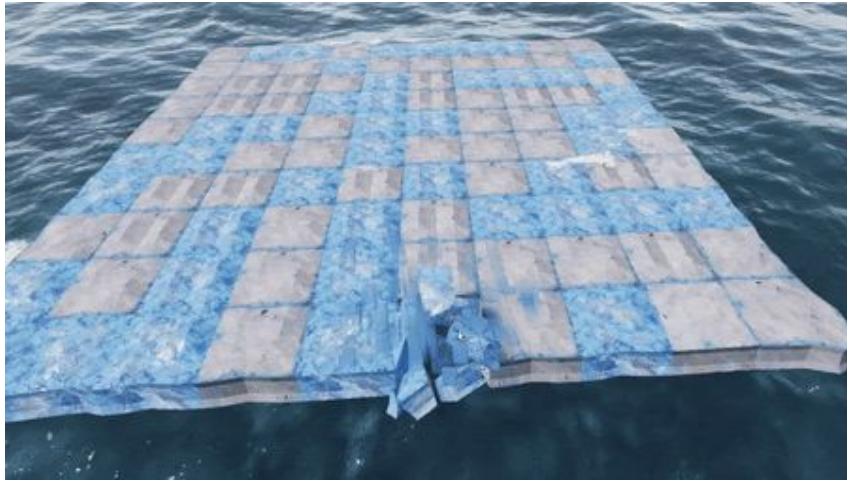


The second hit will break the tile into pieces and add an explosive force to them.



Dynamic Buoyancy for Sub-platforms

The game now supports dynamic destruction with separated buoyancy behavior for each sub-platform. Upon breaking any tile, the connection of the tiles is checked to see if there is any group of tiles that is separated from the main platform. In case there is, the separated tiles will be grouped under a new sub-platform and the sub-platform will be given its own buoyancy behavior. With this, a more realistic destruction system is achieved.

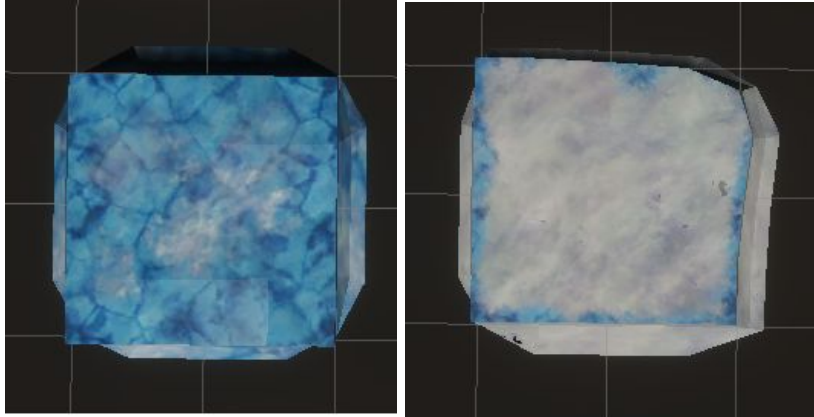


Physical Materials

Upon the instantiation of the platform, the tile type is randomly generated. There are two type of tiles:

1. Icy Tile
2. Snowy Tile

Each type of tile has its own visual and physical material which changes its behavior when the player interacts with it. The Icy tile is slippery and will cause the player to easily slide on it, while the snowy tile is more rough and gives the player more control when moving on it.



Sound

For the Alpha release, we added sound (music and sound effects) that should help with the user's immersion in the experience.

The sounds currently available are:

- Background music (that plays when no events are happening)
- Event-specific music that indicates the current event happening
- Player sound effects for the different action for clarity and immersion

For the playtesting phase, we hope to gather feedback regarding

- The mixing of different tracks (i.e. attenuation levels)
- The quality of individual tracks (should certain tracks be replaced?)
- The placement of sounds (should something be removed? Is there something missing?)

UI

UI Visuals

The visuals have been updated that it fulfills the initial design. To fit the look of the game, textures that are used for the platform are also used for the UI panels. Furthermore, larger texts use a stylized image to look more “icy”. These text images are created via GIMP with a method described in this tutorial: <https://www.saxoprint.co.uk/blog/ice-text-effect-in-gimp>.



Additionally to the Pause UI, a vignetting and desaturating post-process effect is used to show that the time has been stopped.

UI Controls

At the end of this milestone, the UI controls seem stable and all users can access the UI independently (i.e. opening/closing the pause menu), but there is still a minor case of 'not sure why it works'. Sometimes the UI control map seems to be enabled when only the Player (movement) map should be enabled, but it does work. For the playtesting phase we will try to clean up the UI further and look out for any inconsistencies the players notice.

Camera

In this milestone, we made the camera move dynamically during the game as the environment also changes. The position and distance of the camera now depends on the important object positions. These objects include:

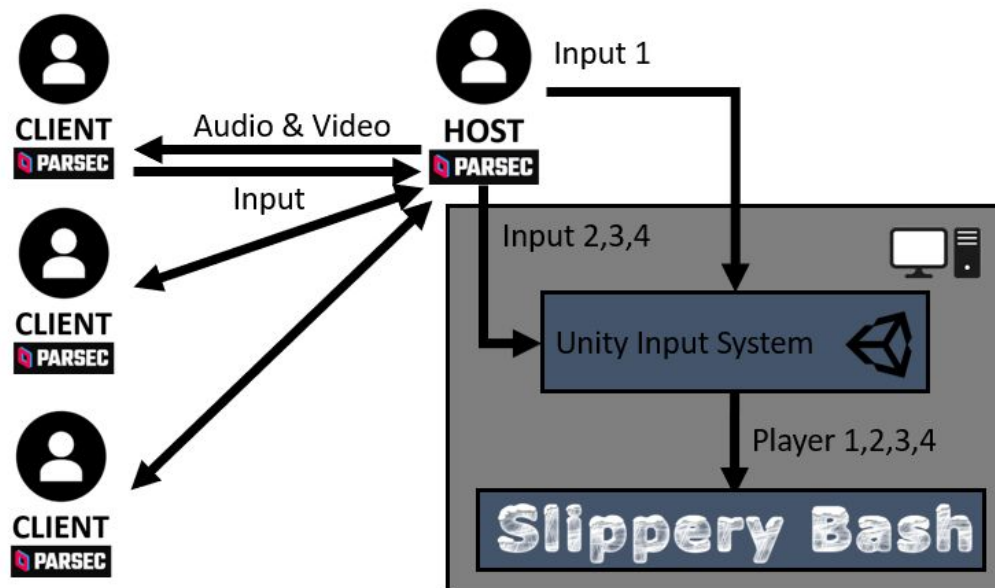
- Center of the platform
- Players
- Titanic
- Lightning Strikes

A bounds object is updated in each frame that encapsulates all of these objects. The camera is then moved smoothly so it focuses to the center of the bounds. The distance of the camera depends on the width or the length of the bounds, whichever is larger. This method is taken from this tutorial: https://www.youtube.com/watch?v=aLpivrPvlB8&ab_channel=Brackeys.

During the lobby, the camera is fixed so the environment can be seen on the right. And during the end of the game, the camera zooms in to the remaining player if there is any.

Parsec

Multiple players are mandatory for future playtesting. This is not possible during the current Corona situation which is why we needed an online multiplayer feature. Instead of adding an online feature to our game, we used Parsec. It is a platform that easily enables us cloud gaming without additional code necessary. One player who has the game hosts their own PC, on which other players can join. The others get audio and video streams and share their input to the host. The host interprets these inputs as their own so the game itself behaves as if there were multiple local players on the PC. Combined with the local multiplayer already implemented in our game and thanks to the new Input System of Unity, we could instantly test our game together.



Physics

With the addition of Parsec, we were able to do a minimal amount of playtesting, and already figured out a few things that we are not happy with in regards to the physics. This will be one of the key factors addressed in the playtesting phase.

We will be looking to improve:

- Swimming (at the moment, this can get much too quick)
- Punch interaction in general (can feel slightly underwhelming)

Other values and interactions might also be adjusted depending on the playtesting feedback.