

# ARTIC

## PROJECT POST MORTEM



C. AUSTRIA &



M. BENSON

COSC 491

STEVEN KENEDY

# TABLE OF CONTENTS

## IN WHICH A PENGUIN IS DEEMED TO FLY

<i>The Reasons Behind the Project.....</i>	<i>4</i>
--	----------

## IN WHICH A GREAT DEAL OF PLANNING IS IN EFFECT

<i>Plans .....</i>	<i>7</i>
--------------------	----------

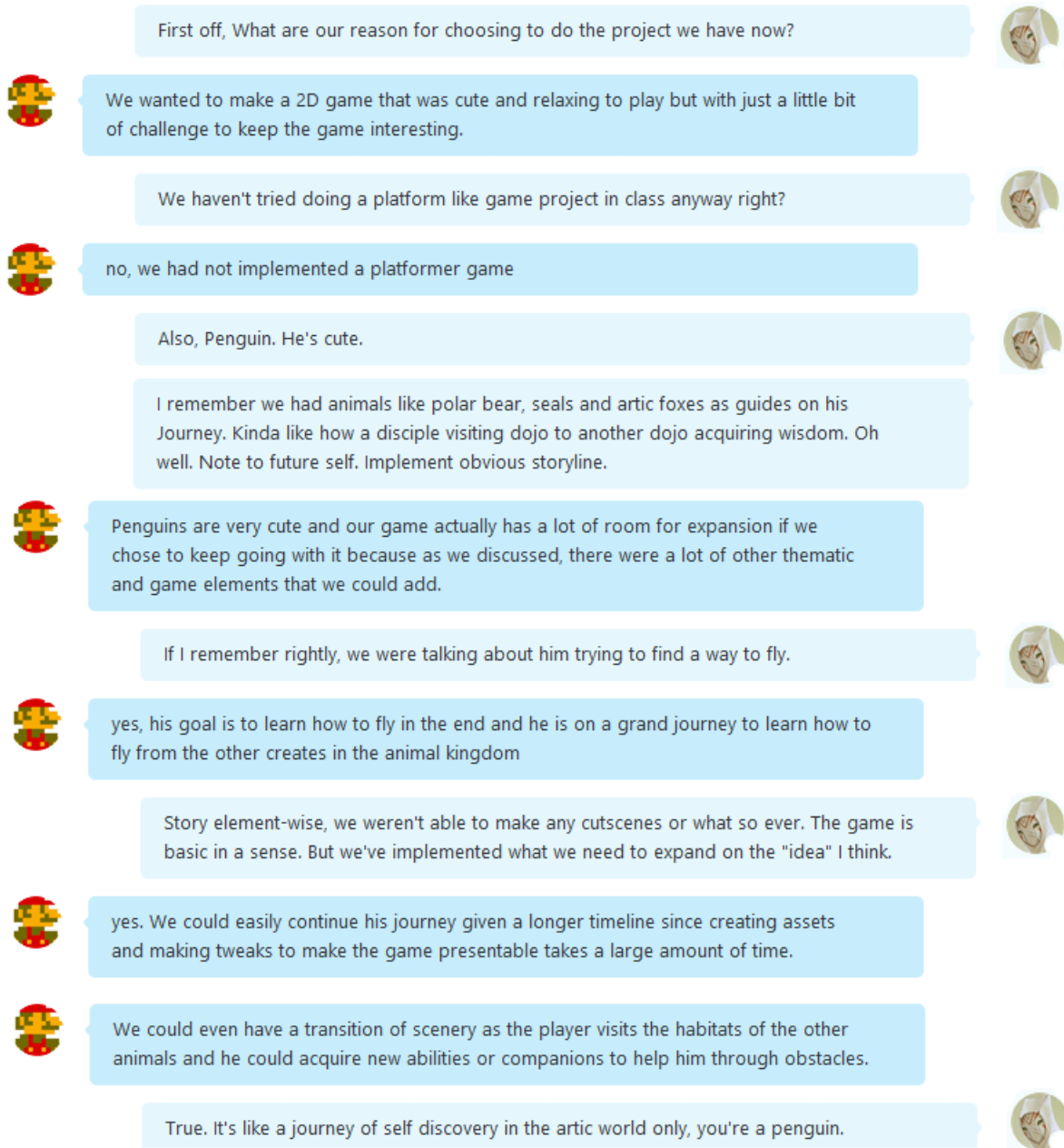
<i>Future Versions.....</i>	<i>8</i>
-----------------------------	----------

## IN WHICH EVERYTHING IS MADE

<i>Implementations.....</i>	<i>10</i>
-----------------------------	-----------

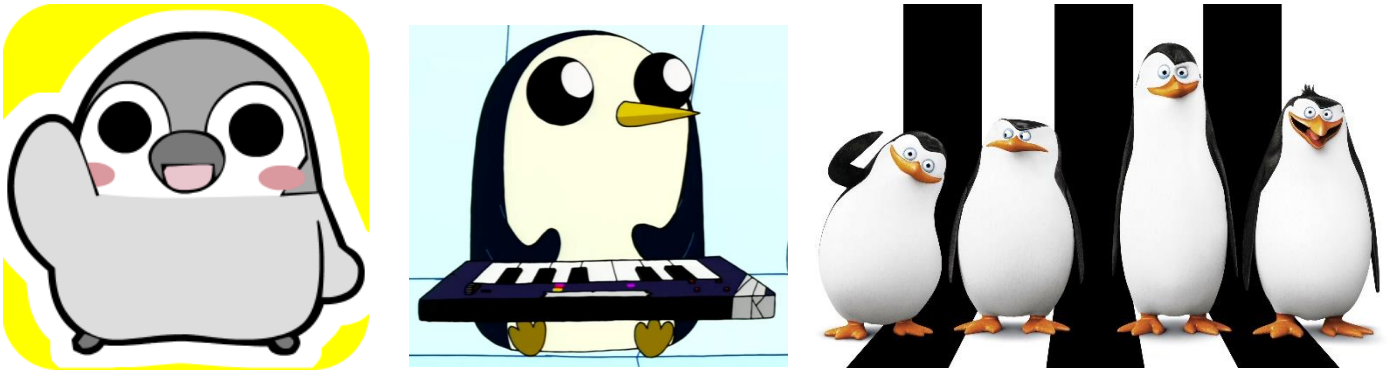
<i>Bugs.....</i>	<i>15</i>
------------------	-----------

# In Which a Penguin Is Deemed To Fly



# THE REASONS BEHIND THE PROJECT

It seems that there is no denying that we two developers like the idea this project will involve a penguin. These creatures do have a 'cute' factor to them and there is a certain amount of the general populace who are quite fond of them. Here are examples of popular penguins in either social media or pop culture in the west and in the east.

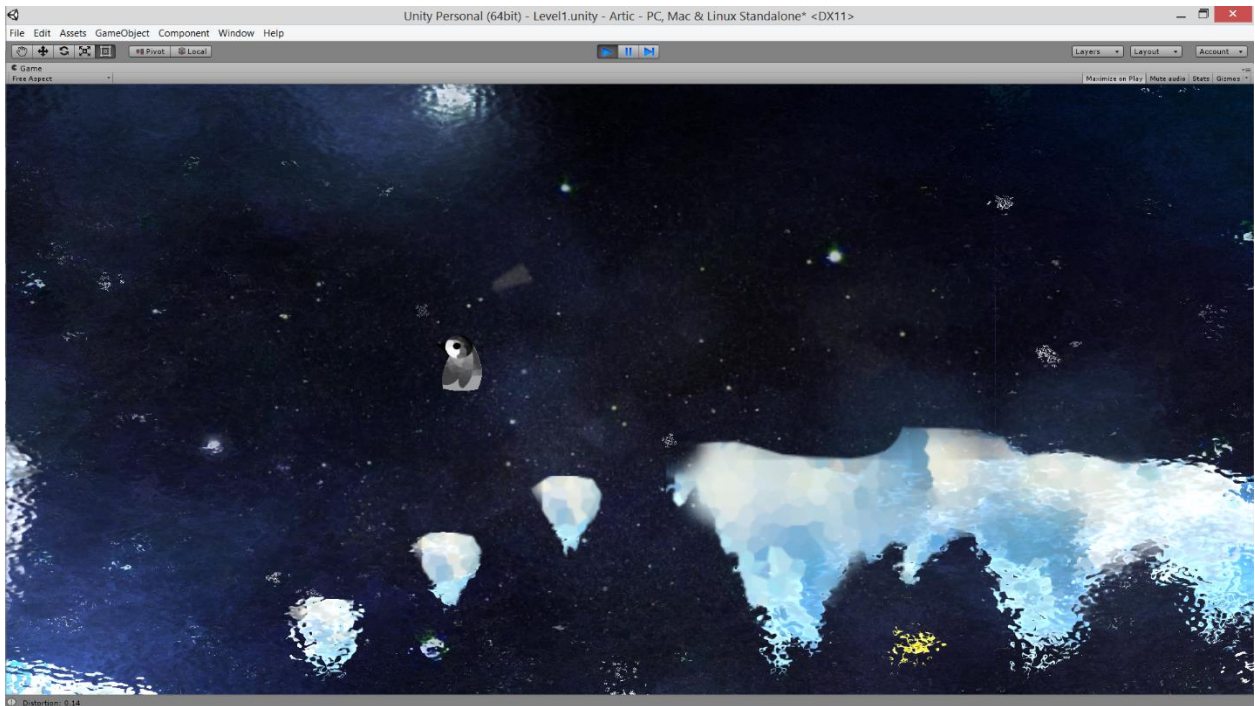


*Example 1.1: (From Left to Right) Pesoguin (Asia), Ice king's Penguin(Adventure Time), Penguins of Madagascar(Dreamworks)*

So there is an obvious appeal in the world for penguins. This is most likely why we chose to feature this animal in our game.

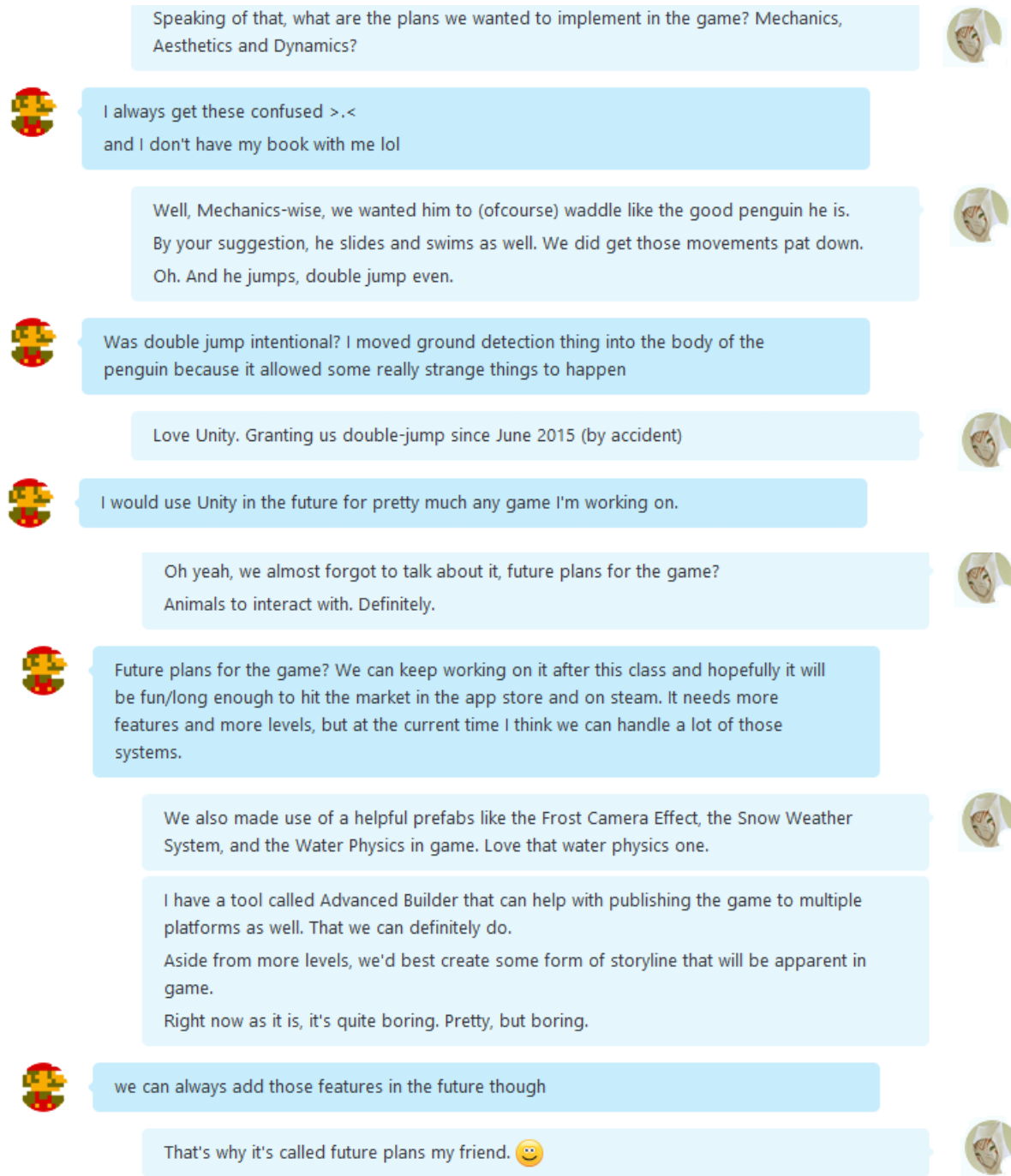
Another reason for choosing this character is for the Artic feel of the game. Since we knew we wanted to make a platformer, we had the idea to use ice and snow elements to effect gameplay. Using an artic theme would allow us both to incorporate various game dynamic possibilities that would open our consumers up to more challenging gameplay in the future. With that in mind, we set out with this thought process. What occurred to us along the game formation is the incorporation of a Sonic game loop-like obstacle that (in retrospect) would make sense for a penguin. Penguins are perfectly capable of high speeds on land as long

as they are on their bellies. Therefore, making that 360 loop will be fine. Swimming and Belly Sliding is another feature that we thought a penguin might need naturally. Finally we thought to make the penguin jump. Albeit the fact that real life penguins don't really jump high, we thought we might add this in for now until we find a better explanation as to why this is so.



*(Above) Example 1.2 Sample In-Game Scene*

# In Which A Great Deal Of Planning Is In Effect





# PLANS

We had several goals in mind (initially) for this game project:

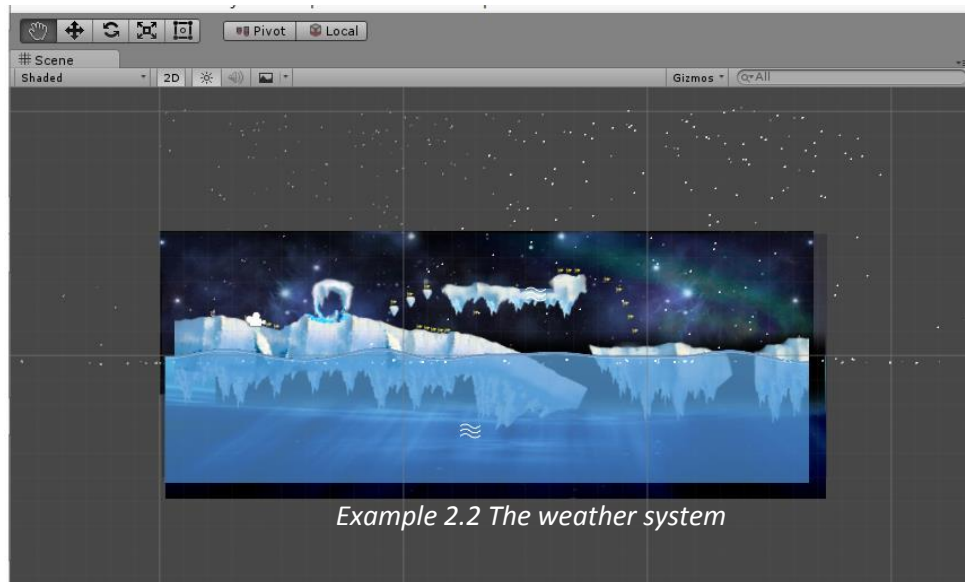
1. Establish several means of movement for the penguin.
2. Create a arttic platform level world



3. Implement a Sonic loop style obstacle
4. Easy to maneuver GUI and menu system.


So far we were able to achieve these and little bit of other side goals like:

5. Animated terrain and/or weather system




6. Smooth movement
7. Limiting the Camera's positioning (clamping)

# FUTURE VERSIONS



Oh yeah, we almost forgot to talk about it, future plans for the game?  
Animals to interact with. Definitely.




Future plans for the game? We can keep working on it after this class and hopefully it will be fun/long enough to hit the market in the app store and on steam. It needs more features and more levels, but at the current time I think we can handle a lot of those systems.

We also made use of a helpful prefabs like the Frost Camera Effect, the Snow Weather System, and the Water Physics in game. Love that water physics one.


I have a tool called Advanced Builder that can help with publishing the game to multiple platforms as well. That we can definitely do.

Aside from more levels, we'd best create some form of storyline that will be apparent in game.


Right now as it is, it's quite boring. Pretty, but boring.




we can always add those features in the future though




That's why it's called future plans my friend. 😊



Animation-wise, I forgot to animate the penguin. I animated the backgrounds and all that neat backdrop, but I forgot the penguin.




I totally forgot about penguin animations




There we go. It's on record. Both forgot. It's fine though, we looks like he waddles anyway. =))


\*he looks



he certainly appears to be waddling by just looking smooth and moving slowly



Something worth noting for future versions: Make penguin animations.



yes, that's a definite for the future



Our future plans include:

1. More levels for players to maneuver in.
2. Storyline for the penguin who searching for a way to fly.
3. Penguin animations.
4. More obstacles (complex) for the penguin navigate in.
5. Multiple characters to interact with.
6. More dynamic uses for the penguin's abilities. For example, if we are able to implement marine creatures, we can maybe have the penguin hitch a ride on those creatures. He can also maybe be lifted up by a water spout coming from the ground.
7. Added dynamic lighting and translucency for the scenes to add more ambience.
8. In-game cutscenes
9. End credits
10. More menu options like scene/level selection, collectibles etc.
11. Save function In-game like a save point.

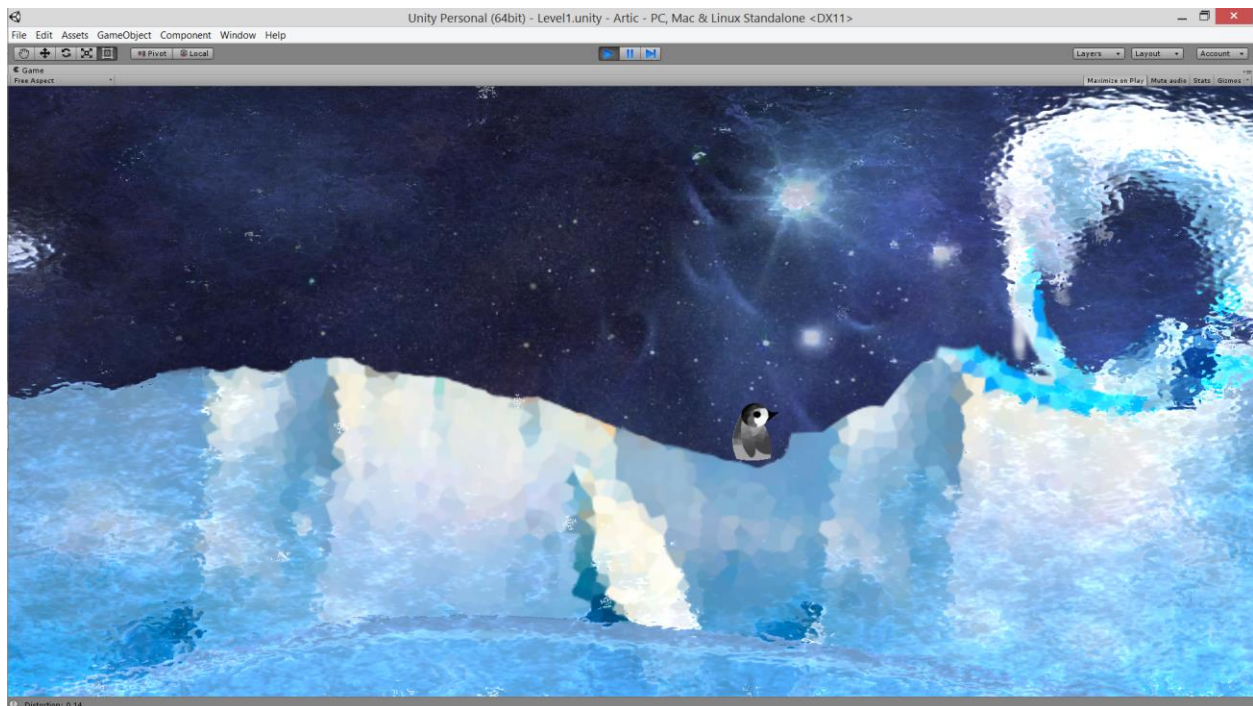
The characters we wanted the penguin to interact with came from the inspiration of Inuit sculptures such as:



# In Which Everything is Made IMPLEMENTATIONS

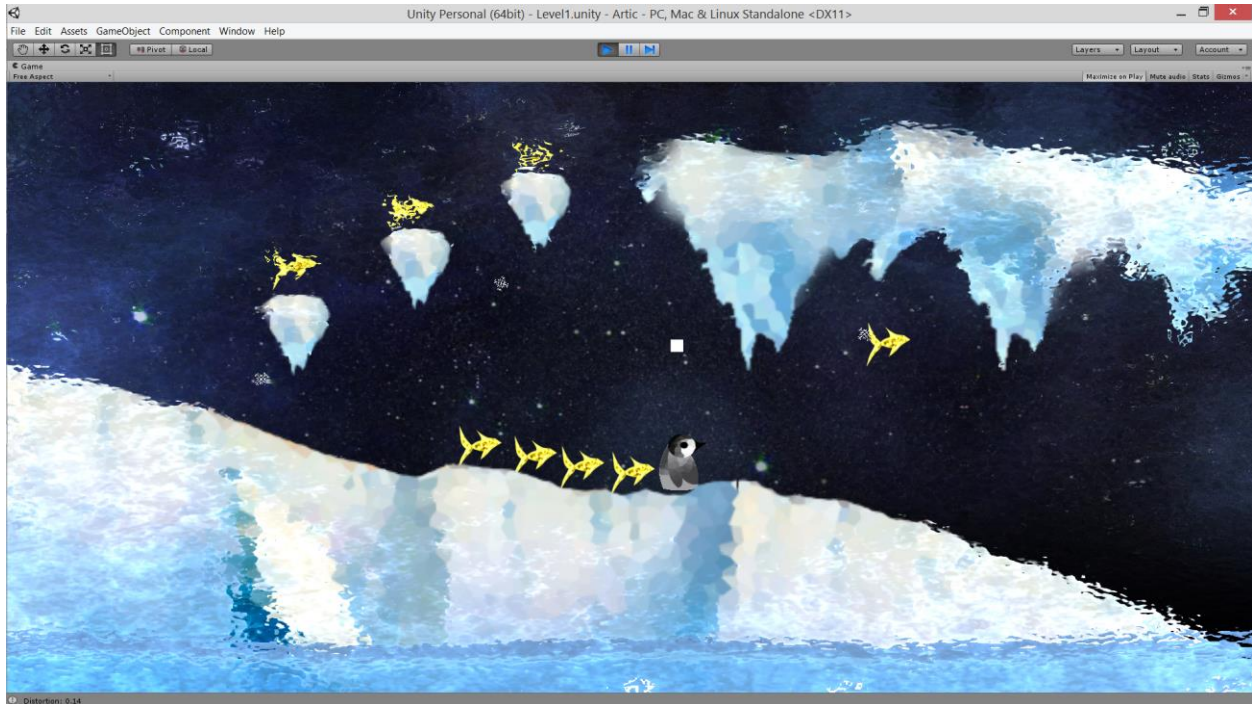
In movement, we used a system that included a GroundCheck gameObject that would see if the penguin is on the ground. This lets the penguin move through the edge collider we placed on the ground more freely (this prevents him from going haywire due to movement). If the penguin wants to jump or swim, we turn this off in order for him to execute the movement. Putting this GroundCheck on the penguin unintentionally gave him an extra double jump ability. We are both working to correct this in the future.

The level design was made while thinking of possible ways to walk through the player using the terrain. At the beginning, a player will be able to walk forward but will soon reach a ditch. This will prompt the player to want to 'jump' over the ditch, introducing s/he to the jump dynamic.



*Example 3.3: (Above) The ditch area in-game.*

Another area feature here is the fork beyond the loop de loop. This area shows the player three features s/he can take/make. First off is the idea of multiple paths to take symbolized by the layout of the coins. Second, the idea of a 'double jump' signified by a fish coin that is higher than all the other fish coins. Lastly, the possibility to use the Penguin's slide ability because of the downward slope of the platform at the bottom end of the shot leading to the water. The water area also lets the player 'swim' through the area if they choose to take it.



*Example 3.4 (Above) Forked area in-game.*

The assets for the game were actually rendered from manipulated photoshop instances using these two images.

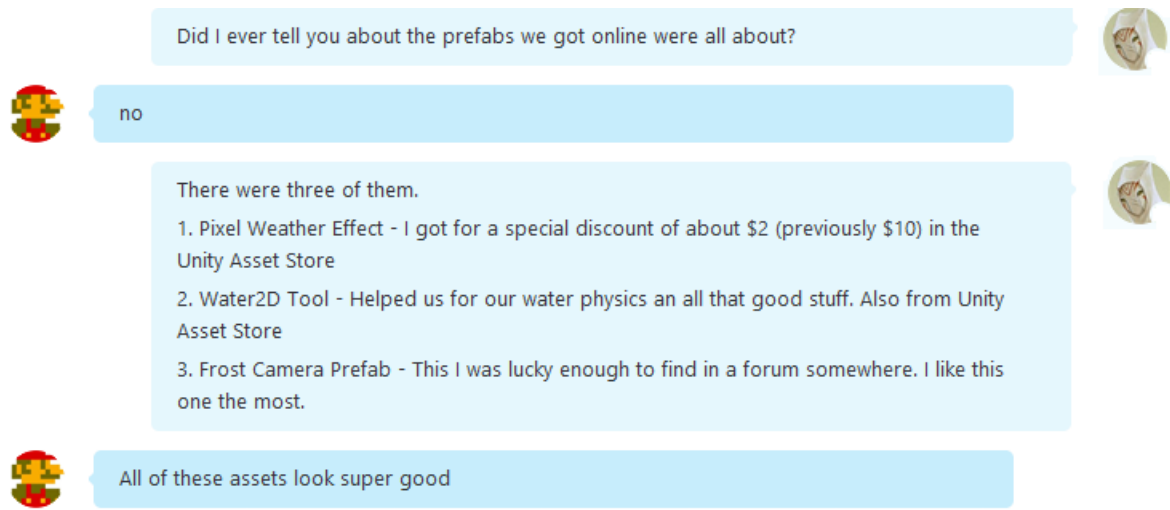


*Example 3.5 Asset Inspiration #1*



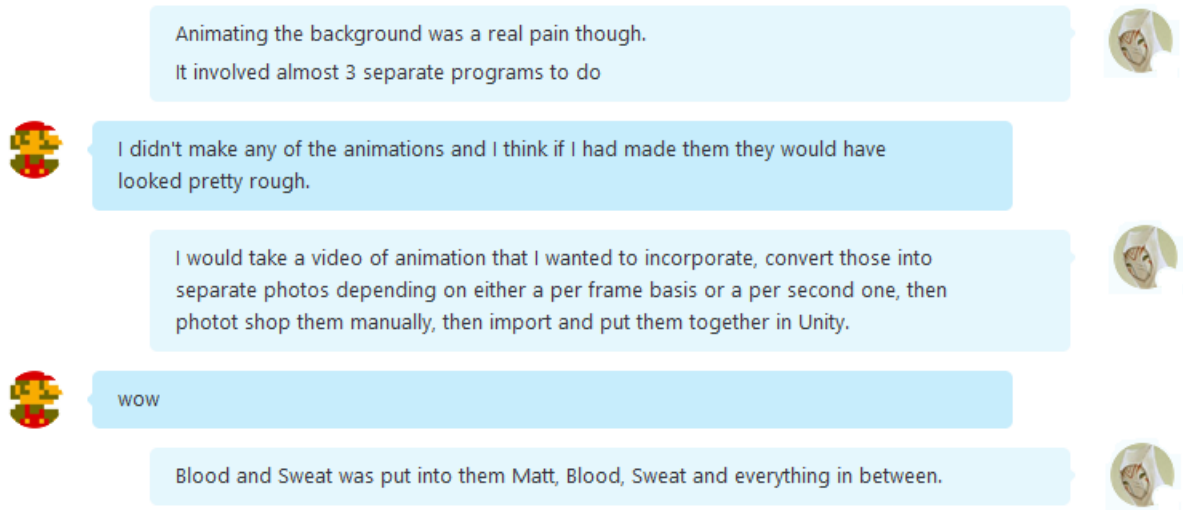
*Example 3.6 Asset Inspiration #2*

The prototype of the Penguin was created using gradients and crystalized. The whole game world was also crystalized (ice effect) and pointalized (snow effect) in photoshop to complete the artic effect.

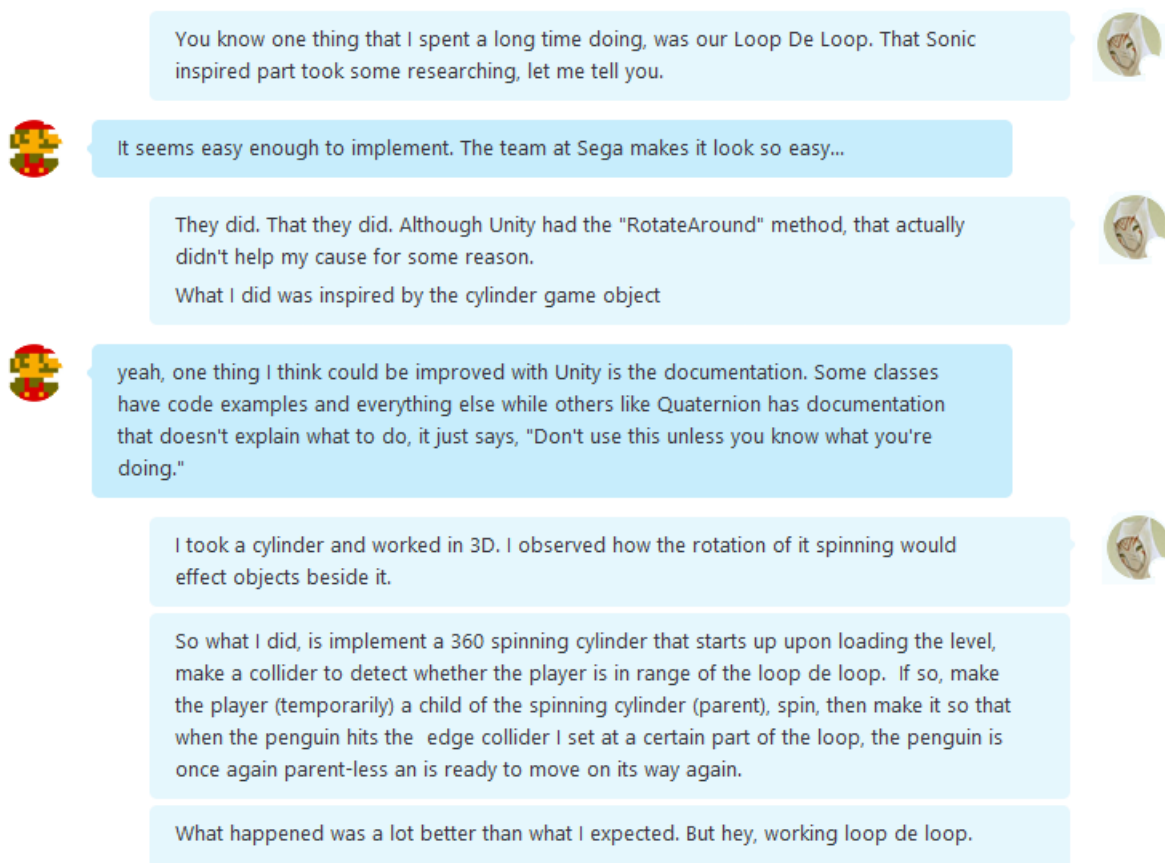


Our team was also able to utilize some Unity assets we were able to fetch online. The Pixel Weather was is responsible for the falling snow flakes and snow in the game. The Water2D helped us immensely on manipulating the water physics in game. We still had to manipulate variables and play with it. The tool was ultimately incredibly helpful and also added to our knowledge about how (for example) Sine waves are used on the top lining of the water edge to create a 'wave' effect for the water. The Frost Camera prefab is used to create the frosted glass look for the main camera. We really liked this one because you can directly control how much frost is applied to the main camera. This gives us multiple possibilities to manipulate this to expand on the dynamics of future gameplay. For example, we could make it seem as if the penguin is completely frozen by the cold by increasing the frost on camera etc. The main camera script also handles what

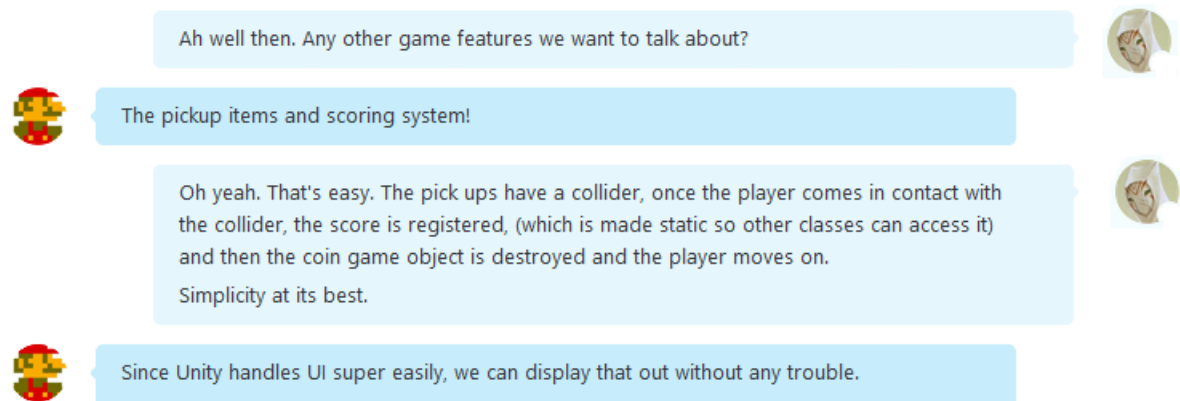




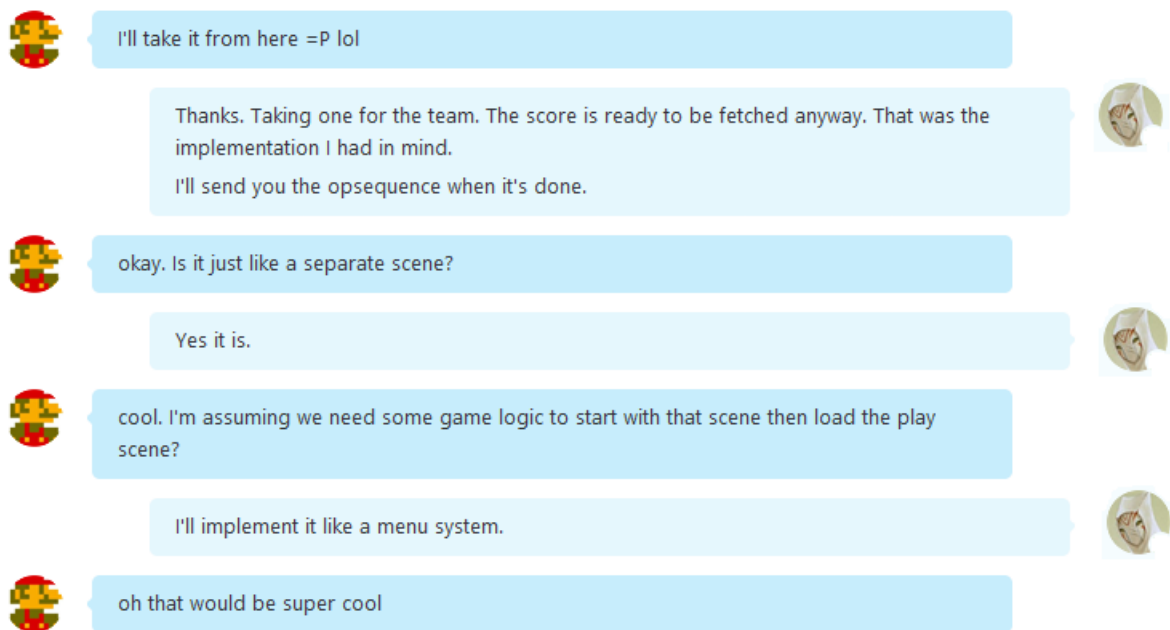
As shown in the chat image, the animated backdrop used a video representation of the scene, converted that into frames and then animated into Unity.



The loop de loop of the game we thought to initially render using the RotateAround method in Unity. We ran into problems with this and scraped things this out in the end. We ended up implementing a continuously rotating cylinder (rotated to a certain angle) towards the player that will make the player its child when the player comes in contact with the loop and then release the player after s/he was able to make a successful 360 degree loop.



The scoring system is simple in the sense that when the player comes into contact with the fish coin prefab, it registers a point and then destroys the prefab.



The menu system is made a separate scene that leads to the first level of the game when Start is clicked. Exit leads to the application terminating.



# BUGS



Unity is a good platform by all means. I had it crash and so I reported the crash and opened a support ticket. By the time the support team contacted me, the issue had already been resolved and I only needed to update Unity.

You know it crashed for me as well. Good ol' online community forums told be to delete the Library folder of the project, and lo and behold, Unity fixed itself.



In this area we talk about what we encountered working with the engine. Matt and I played Unity's dangerous game of crashing when we least expect it. Matt had been working mainly on our movement system and Unity decided it was time for a break. The crash made him loose some Project assets and settings but thankfully the scripts were saved.

I (Cat) encountered a different situation when I was importing large assets into the game. This was the time I was animating and editing our in-game effects when Unity froze on me. It recovered after a while but Unity kept bugging me about an error about the wrong timestamps for some of our assets. The program ultimately failed and would not let me see my changes on the game screen. Unity crashed thereafter and before I decided to submit a ticket, I researched on the net if this was a common occurrence. Sure enough, it was and the temporary solution was to delete the Library folder in the project's folder. I proceeded with this solution with a Hail Mary and this fixed my problem.

Another notable aspect about documentation both Matt and I found was the lack of updated samples and explanation from the documentation site. As you read in some of the chat images, Matt and I would encounter some methods or classes that didn't fully explain the capabilities of it in the explanation provided in the site. Work-arounds were in order if not for forum browsing, stack overflow and the occasional Youtube tutorial.

In Matt's case, the Unity dev team was quite helpful and was able to assist him in his problem via updates. Unity overall is still an excellent platform we would both chose to use for future game development projects.