Introduction:
This project was a collaboration between level designers to create a seamless 3D platformer game by combining our multiple levels and linking them via transitions. After multiple iterations to our blockmesh, we began combining our levels through perforce and began replacing blockmesh with finalized assets.

What Went Well:

1.) Water and Water planes

One of the main worries of mine was that the water inside of the cave wasn't going to stand out enough from the other structures and would seem too flat. To counteract this, once we added water into our levels, I decided to use a rectangular light with a hue of light blue to create this blue lagoon type water effect. This dramatically highlighted the water, but didn't overpower the scene and didn't overpower any other aspects of the scene I was trying to differentiate. One of my biggest fears with approaching this by adding a blue light was that this would cause reflections and color bleed onto other areas of my main cave area. Luckily, I decided to place the light facing downward on the Z-axis instead of upward, so most light was absorbed by the rough, dull ground instead of shining upward and reflecting off of unwanted surfaces.

2.) Landscaping in a pinch

With just days before the final build of our project I had to replace all of my trees. I later went into this in the "What Went Wrong" section. In a pinch, I still wanted to add landscaping, so I decided to place a landscape plane on top of my level and went to work. I had previous experience landscaping in Unity, but had very little experience using the landscaping tools in Unreal Engine. I was able to pick up on these tools very quickly and soon I found a great workflow between using the erosion tool to lower spots, then flattening to even out a chunk of the landscape. This allowed me to power through the landscape and create a landscape I was very proud of. I was able to fix the water planes and other areas where I felt it didn't blend well in the landscape, and I was able to later use this landscape to blend the mountainous, rocky walls into the ground.

3.) Leading Players Eyes

Just recently learning how to use linear interpolation, I was nervous to apply this throughout my level so much. I started off very simple by adding levers, door animations, and other simple lerps. I eventually felt confident enough to add things like the bird and fish in my level that help guide the player and lead their eyes to the proper locations around the map. While working on the bird it kept looking a little bit off, so to resolve this I decided to do a little bit of research and look up how many flaps per second a bird's wing takes (2.4/s if you're curious). After doing some simple math and translating this information into visual scripting, I was able to make my birds movements more realistic and believable.

4.) Cinematics

Having no previous experience doing cinematics in Unreal Engine, I was skeptical at adding many of these into my gameplay. After adding the first cinematic sequence, when the cave crumbles in and closes off the entrance, I saw how much it added to my level. Although timing was an issue with my sequences and cinematics, at first, I soon learned various tricks and began developing a workflow that helped me create cinematics in a rush and frame scenes that would create foreshadowing and excitement for players. These cinematics added extra impact to gameplay and showed the importance of interacting with levers, or showing that players are now stuck within an area of the map and need to find a way out.

5.) A Notable Difference

One aspect that I didn't expect to make such a difference in gameplay was the addition of notes. These notes added that little bit of added story and background that helped with player immersion and setting the scene without adding numerous assets scattered about. I first added these notes to direct players to their next playable area and to keep players on track with what was to come in the level. As I progressed I noticed that the addition of these notes helped with immersion vastly, so I decided to make the final note a trigger for the final sequence when the floor collapses under the player. I was hesitant at first when deciding to do this. but soon after discovered that this helped ensure players would fall through the floor when it collapsed and helped players gather the information needed to understand why the floor was collapsing. The notes made a major difference in my level's gameplay and I am very happy with how the final results turned out.

What Went Wrong:

1.) Too Many Tris!

One major setback for me was using mega scan trees for a majority of my level before I had a full understanding of LODs, reducing tris and vertices, and optimizing my meshes for performance. After spending days replacing blockmesh with finalized assets, rotating trees, spacing them properly, and setting up properly framed shots with trees, my team couldn't build. In order to fix this we cut down the amount of primitives in our levels. The most prevalent source of this was the megascan trees and prior to deleting these from our game I hadn't optimized assets and reduced their tris. After deleting all the trees out of my level we were able to build again and I later used different tree assets that we had in our content. From this I did learn how to better optimize my assets. I eventually looked into how to reduce tris and vertices and lowered my rocks LODs by 75% to optimize performance.

2.) Apex Destruction doesn't render materials during build

I started by using a bridge that had planks that would self destruct and shatter after a second of exiting the hit box around each plank. I eventually used this for my transition to the next level by adding it to the floor in my final vault where players would fall through the shattering floor. The problem occurred when I went to build and noticed each of these meshes only had a base gray material on them instead of their intended material. I was looking for solutions throughout the entire project. I eventually upgraded my bridge to a wonderful asset I found that used simulated physics to give the rope bridge more realism. I later found the solution to this by changing all destructive meshes materials to the material I wanted to utilize. This solved the problem and ended up rendering just fine.

3.) GPU Performance and high GPU delay

As stated before, I haven't dabbled in profiling, performance, and optimization as much. My level was having major performance issues throughout the merge of our projects. After diving more into profiling and learning how to properly optimize each aspect of my games performance, I was able to figure out it was a GPU issue after using the stat unit command during my playthroughs. I noticed a major delay in the GPU throughout my entire level. After profiling the GPU by using the command profile GPU I noticed there was a directional light that hadn't been deleted upon merge that was causing shadow rendering issues. This also led me to optimizing shadows on grass and other secondary environmental assets. This greatly improved performance for my game and resolved my GPU delay.

4.) How hard is it to break a floor?

How hard was it for me to break a floor? Turns out it was harder than expected. I ran into issue after issue trying to get the results I wanted when creating my collapsing floor destructible mesh blueprint. First, I ran into an issue with the scale of my destructible mesh. I created the mesh by using a singular destructible mesh block, and when I would scale this up there was seemingly nothing wrong, but once I triggered the destruction event everything went wrong. The breaks were stretched and it was noticeable that scale was a major factor that caused the floor to not shatter and instead create multiple elongated planes that would launch players to the ceiling rather than having them drop through the floor. To resolve this I scaled up a block to the size of the floor and added a very tiny block in one corner that would be hidden in a wall. After creating these I combined them into a static mesh, turned this into a destructible mesh, and replaced this with the current mesh in my collapsing floor blueprint. This resolved my scale issue, but now we were building our game nightly and reiterating, so I began noticing that the materials would render in the engine, but not in the builds. This posed a major issue because having a gray floor that collapses takes away from immersion. As mentioned above, I later resolved this issue by applying the materials in every spot I could find to apply materials for this object. After resolving the rendering issue, I noticed that again, it’s not very immersive when you have a floor collapse and the objects sitting on the floor are just floating above the player. This led me to my final iterations of this object and its sequence. Although this collapsing floor issue caused me hours of wasted time researching solutions and tons of failed build attempts, I was able to resolve this in the end and come up with a collapsing floor I was quite proud of.

5.) Close Quarters Mantling

With the game's core mechanic being that players can grab ledges and climb them, I wanted to showcase this in my level and allow players to utilize this mechanic. I soon discovered a major bug with this mechanic that interfered with my level's design. When players are going for these ledge grabs and the player character is close to a wall, this causes the player character to become suspended in the air under the ledge and that player will lose the input function. While the solution was simply just moving the ledge grab further away from the wall, this led me to rethink many areas of my maps design. Since my level was primarily focused around the cave, I had a majority of the ledge grabs in the cave area and since the close quarter mantling bug would happen if areas were too tight this led me to opening up many of the areas of the cave. This alone set me back some time and caused me to rethink areas of my playthrough that caused this bug. One of the major areas where this was prevalent was the caves exit, where players jump and mantle a series of rocks in this half cylindric cave exit. I had to redesign this area and its jumps a handful of times to ensure each of them were usable and this bug wouldn’t stop players from exiting the cave.

Conclusion:

This project was a great learning experience for me and helped me better understand performance and optimization. I was also able to utilize many new tools within Unreal Engine and in the end I am very happy with the final results of my level.