Tournamental Postmortem
Tim Obenauf

This month, I was tasked with implementing two mechanics to the tile[based, puzzle game, Tournamental. I needed to come up with a unique mechanic, implement it, then implement a unique mechanic designed by one of my peers to complement my own. Through this, let me talk about the good, as well as the bad, things that happened over the course of this project.

What Went Right:

1. **The Mechanics –** The overall success was that I was able to get both my peer's and my mechanic working completely successfully, with no immediate bugs or anything that would hinder the performance and gameplay. This came from many hours of trying to figure out where things were happening in the code, so that I could incorporate my own code allowing it to function properly.

2. **The Blueprints –** For part of my mechanic, I had to introduce new blueprints to my project for it to work. I created a pressure plate, a gate, as well as a clone that I could call from the main player if certain criteria are met. I was able to add only what I needed. Because of this, my project didn't need too much space.

3. **The Design –** The project came with pre-completed functionality. With it, I was able to create the blueprints so they fit into the same design of the project. Nothing feels out of place, and if it is, it makes sense why it would be. This is most prevalent in the pickup blueprint I made, since nothing else was really a sphere so the player would already be drawn to it and want to try interacting.

4. **Troubleshooting –** There were some challenges in the coding that I needed to solve, and with use of the development tools like Print String, I was able to narrow down issues in my code to figure out what was going wrong. I was able to find out issues where my movement would override the tile entities, which would break the player movement. Troubleshooting this took some time, but overall it was still a great success.

5. **Peer Mechanic Implementation –** As mentioned, the mechanics all went well. For my peer's mechanic, I needed to modify it significantly for it to fit into my code. With hours of testing, coding, more testing, and more coding, I was able to get it all to function with my mechanic completely. Taking their mechanic, which only allowed toggling between two characters, I was able to modify it so it would work with any number of characters in the level.

What Went Wrong:

1. **Loss of Work –** The biggest issue with my project was loss of work. I thought I was saving frequently enough, but at one point I lost power at home due to a small storm. Because my project was on a Desktop, it was not saved when the computer shut down. This caused me to lose about 8 hours of work that I could not recover. Some of this code is still missing, but I was able to restore enough of what I had for it to be functioning properly.

2. **Incorrect Code –** There were a few occasions as I was working on Tournamental where I was able to get the code functioning with certain code such as calling an array with certain variables passing into the indexes. After continuing to work, these nodes would not work with the new developments, so I had to completely strip down what I made. This also happened when using the timeline in my gate blueprint, since I was not accurately interpolating the vectors.

3. **Calling the Clones –** As part of my main mechanic, I had the player clone themselves which would set them on the map. When I would try swapping with my peer's mechanic, it would lock up the clones and I couldn't switch back to them. This continued to break my project for some time, but I was eventually able to identify the issue and correct it by calling possession from an array rather than direct actor references.

4. **Improper Materials –** Part of my mechanic would affect the player directly by changing the color. If the player had clones, they would glow blue. This did not work from possessing the clones that I place, as it would override the material to the default color. Ultimately, I had to remove the glowing color, as I could not fit it in the scope of what I needed to finish to submit my project for review.

5. **Arraying Blueprints –** Swapping the player's pawn functions off of an array that is set any time the player spawns a clone. I had issues with this to begin with as I couldn't figure out how best to set the array. After extensive troubleshooting, researching, and review, I did get it to work. This could have been improved if I reached out for help, but I chose to do this project solely on my own.

Conclusion: Overall, the project went well. I resolved any issues I had with the project, and am in a place where I can continue to expand on this for other unique features to make a unique and fun game. With many hurdles, there was a lot to learn from this, with the most important being reflection from where I started. I need to be prepared to seek help if my research doesn't solve my concerns, and will strive to do that moving forward.