This artifact is a proof of concept for a scalable customizable magic combat system. The idea behind the artifact is to create a system that allows players to customize magic spells by selecting from a list of available properties. But I also wanted to make sure the system is easy to expand by putting all the code in a single place. By making one object that holds a master list of all available spell properties and holds all the functions required for those properties to be executed, then adding to the system would theoretically be straightforward. I developed the concept over the course of my GAM-465 class, however the prototype was developed over three weeks during GAM-495.

I selected this artifact because it was the only artifact, I had made which was not based on a design document that I had to adhere to or derived from a tutorial. I felt obligated to develop the concept because I put a lot of work into refining the idea. Not only did I feel like it was a solid concept, but it also had the possibility to be one of the most complex programming challenges I had ever tackled. I am proud that the system works as I imagined it, the spellbook is designed to communicate with other instances of itself and it handles every aspect involved with casting magic and getting hit by magic. I can add properties to each of the categories quickly, and the changes to the spellbook actor component are automatically propagated to the menu used to customize the player’s equipped spell.

This artifact posed a unique challenge which I did not anticipate. The crux of the idea is based on the backend of the system, so naturally most of work was on things that cannot be seen when running the game. It was difficult developing a system that I could not visually confirm was working. In fact, when I did add the visual aspect of the system, I found several bugs that I had not noticed previously. I would have had an easier time developing the system if I had been more liberal with the debug text. Something else which popped up was that I needed to add radio buttons to the menu, but radio buttons are not an available widget in Unreal. What was surprising though was that the lack of radio buttons barely slowed me down. I did not even have to look up any resources for this problem, I just knocked out my own radio button widget for the player to choose from available options, as well as another unselectable radio button to show the player the options not yet available to them.