

Looking back on this semester, I've realized how much time I spent learning programming fundamentals like syntax, structure, and key concepts, versus actually writing code. At first, I felt completely overwhelmed trying to figure out how everything fit together. For example, the `setup()` and `draw()` functions in P5.js seemed simple enough, but truly understanding how they interacted with other parts of the program, like arrays or loops, took time to grasp. Honestly, it was a slow start. What helped me get over the initial hurdles was watching YouTube videos. Seeing programming concepts broken down step-by-step and applied to real examples made things so much easier to understand. Following along and seeing immediate results on my screen gave me the confidence to keep going.

On top of that, I spent a lot of time outside of class practicing. While class time gave me a foundation, it wasn't enough for me to feel confident on its own. I worked on small side projects, rewatched tutorials, and debugged my code endlessly to reinforce what I had learned. Those extra hours really paid off when it came time to tackle my final project. I felt ready to take on more creative challenges instead of just focusing on making things work. This growing interest in programming pushed me to sign up for a coding course over winter break outside of NYU so I can keep improving. I'm also excited to take more programming classes next semester as part of my coursework. It feels good to know programming is becoming a bigger part of my education and interests.

One of the things I'm most proud of in my final project is the animated circle. It became the centerpiece of the project and tied perfectly into the Interstellar theme, symbolizing the loops of time and gravity central to the movie. Its constant motion and dynamic transitions gave the project a sense of life and energy. But getting it to work wasn't easy. At one point, the transitions felt clunky, and the animations didn't align properly. I spent hours tweaking frame calculations and debugging until I finally smoothed out the timing. That moment, when it all came together and reflected the theme so well, was incredibly rewarding.

This semester also helped me understand the difference between object-oriented programming (OOP) and procedural programming. Procedural programming is like writing a step-by-step recipe for the computer to follow. It's straightforward and works fine for simple tasks. But for bigger projects like this one, OOP became essential. Using classes to manage elements like the animated circle and particles made my code more organized and easier to scale. At first, OOP felt overly complicated, and I struggled to understand why it was necessary. But as the project grew, I saw how grouping data and behaviors together made everything more modular and manageable. OOP wasn't just about making the code "fancy"; it made complex interactions way easier to handle.

This project also helped solidify programming concepts I struggled with earlier in the semester. Managing arrays and loops, which felt intimidating at first, became second nature by the time I was working with hundreds of particles. I also got much better at using P5.js-specific functions like `dist()` for collision detection and `push()/pop()` for managing transformations. Debugging, as always, played a huge role in the process. There were moments where things didn't work as expected, animations glitched, variables didn't reset, or parts of the design overlapped awkwardly. Watching debugging tutorials taught me new strategies, like isolating parts of my code and testing small sections individually. Debugging can be frustrating, but the satisfaction of solving a stubborn problem is hard to beat.

One unexpected challenge I ran into during the final project was the layout. My original goal was to create an interactive and visually engaging experience with an animated circle and a Morse code input system. While I hit that milestone, I realized the layout felt cluttered, especially with stars overlapping the text in the center. Fixing this required redesigning parts of the project and adjusting the logic for how stars and other elements were displayed. This extra step wasn't planned, but it made the final result so much cleaner and more polished. Another part of the project I'm really proud of is how the animated circle tied into the Interstellar theme. It wasn't just a design element, it symbolized the fabric of time and gravity, just like the movie's concepts. I wanted the circle to feel

dynamic and alive, and seeing it come together after so much tweaking felt amazing. It added meaning to the project, and I was proud of how it turned out.

Moving forward, I'm excited to keep building on what I've learned this semester. Signing up for a coding course over winter break is one way I'm pushing myself, and I can't wait to take more programming classes next semester. I see programming as a skill I want to carry into other areas of my education, whether it's creating more projects like this or applying it to problem-solving and design in other classes. This semester has shown me that programming is challenging but incredibly rewarding, and I'm excited to see where it takes me next.