

Mars Lapierre-Furtado

December 3rd, 2024

## Post-Reflection: Evolving Through Code

### **How I Evolved This Semester**

Looking back at my assignment at the start of the course, *Art Jam*, I focused on a simple variation of a basic program, using core p5.js functionalities like mouse interaction, `map()` for color gradients, and straightforward looping. My comfort zone was working with basic elements such as shapes and simple conditional logic, drawing on past experiences with p5.js. Most of my previous projects had been focused on creating interesting visuals influenced by simple interactions, and I had not pushed myself to deepen my skills or explore more interconnected programming elements. For *Art Jam*, I was inspired by the summertime and created visuals like gradients and mouse movement without delving into more complex game logic or interactivity, as I had limited experience in that area.

The *Mod Jam* project was the one of the first times I was asked to modify a pre-existing game to introduce complexity. While I had made simple games before, this assignment presented new challenges. At that stage, I didn't push myself to make more interconnected changes, and as a result, my project lacked complexity. Reflecting on this now, I would have approached it differently, making more significant and integrated modifications.

Since then, I've improved significantly. I decided that the *Variation Jam* assignment would be the way I tried to redeem myself by trying to push myself with more complex and more interconnected modifications. I am now a lot more comfortable introducing different game modes that require more advanced state management. I've learned to implement custom mechanics, using Flappy Bird as a foundation to experiment with unique gameplay elements. Through experimentation and a variety of resources, I've gained the ability to manage dynamic object creation, demonstrated by my management of a list of pipes with changing properties for

continuous gameplay. In *Variation Jam*, I felt a significant shift in my approach to coding. Introducing different game modes meant that I had to think beyond basic functions and consider the flow of the game itself—how different elements would interact with each other across multiple states. Learning how to manage dynamic objects was a key breakthrough for me, particularly when it came to creating pipes with changing properties in the game. I had to not only generate these objects but also ensure that they fit seamlessly into the game loop, requiring me to think much more about performance and real-time interaction. This project really pushed me to see the bigger picture, moving from isolated code snippets to thinking about how everything connects and functions together.

Looking back at the start of the semester, the biggest change in my coding practice has been my growing comfort with complexity. I feel more confident approaching projects that need me to focus on intricate interconnections between many different elements.

## **How Coding Expanded My Creative Vision**

Before this class, I had used coding to create artistically but honestly it was a means to an end a lot of the time when I had a specific vision in mind. I think I felt a little intimidated to think of how far I could push it, so I didn't, and some of me still is. The assignments and challenges of this class made me develop the way I saw the relationship between programming and creative practice. Programming allows for the creation of interactive and responsive work that isn't usually possible with typical media. Making a more complex code allows for a more dynamic project. These interactions can make the user feel more immersed and engaged in the work. With a deeper understanding of the tools that are available to me, I can see even more possibilities in my ability to custom-design visuals, animations and interactions based on my vision. Hopefully, with this great foundation, I feel even more confident exploring all the possibilities of Javascript and p5.js when it comes to creativity.

Initially, the idea of incorporating game mechanics or complex logic was intimidating because it introduced a layer of unpredictability and required a deeper understanding of how different elements interact with each other over time. With static visuals, the process was relatively straightforward: I created a set of fixed components, and once they were in place, they didn't need to be adjusted or changed. However, when it came to incorporating game mechanics, especially interactive elements like player inputs or random events, the stakes were higher, and the logic had to be more flexible.

Transitioning from static visuals to interactive, dynamic systems forced me to think about how to manage state transitions, game flow, and how objects or elements would interact with the player in real time. For example, for the *Variation Jam* project, I had to plan not just for visual effects, but also for how game objects (like the pipes in the Flappy Bird variation) would move, change properties, and continuously reappear in a way that felt seamless to the player. This meant thinking beyond individual pieces of code and starting to consider the relationships between them. How would one change affect the others? What would happen if certain conditions were met, or if the player interacted in unexpected ways?

This shift in thinking required me to challenge my earlier view of project planning. Before, I focused primarily on aesthetics and the user's immediate reaction to visual stimuli. Now, I needed to plan more comprehensively, considering not just the immediate visual experience but also the underlying systems that made the interaction feel fluid. For example, ensuring smooth transitions between different game modes, handling real-time updates to game objects, and ensuring the gameplay loop remained engaging were all challenges I hadn't anticipated in earlier projects.

To adapt to this new complexity, I had to take a step back and approach my projects in a more structured way. I began breaking the problem into smaller, more manageable pieces, mapping out how different parts of the code would work together. I also learned the importance of anticipating edge cases—unexpected events that might break the game's logic or disrupt the user experience. This kind of

forward-thinking was something I hadn't fully embraced before, but it was crucial for creating interactive, dynamic experiences. By developing a deeper understanding of how to organize my code and account for various possibilities, I've become more confident in my ability to tackle these kinds of challenges in future projects.

## **Growing Into a Creative Coder**

Looking ahead, I feel like I'm slowly moving closer to owning the role of a creative coder. This class, along with the other courses I've taken this semester, has helped me build both my skills and my confidence. Before, I was hesitant to tackle projects that involved complex game mechanics or dynamic systems. But now, with more practice and learning, I'm starting to feel more comfortable thinking about not just visuals, but how the interactions within a project can flow and evolve over time.

My understanding of creative coding has shifted, and I now see it as more than just a way to make things look interesting—it's about creating systems that can change, react, and engage. With this new perspective, I'm looking forward to pushing myself even further, trying out new ideas, and experimenting with more complex interactions. I'd like to work with things like procedural generation, or even bring more multimedia elements into my projects to create more immersive experiences.

There's still a lot I want to learn, and I know I'm not fully there yet, but I'm excited to continue developing my skills and ideas. This class has been a key step in growing as a creative coder, and I'm looking forward to seeing how far I can take these skills in future projects.