

Post-Reflection

Significant Changes in My Programming Knowledge

1. Understanding JavaScript Logic

At the start of the course, my understanding of JavaScript was limited to basic ideas about how code functions. I could not think in JavaScript or through JavaScript, in the sense that I didn't understand how the program structures and processes information, including different datatypes such as constants, conditional statements, arrays, and p5 functions. Through class discussions and debugging experiments, I gained a clearer grasp of how to work with the language to achieve specific outcomes. I also grew an awareness of its limits and how to navigate around them. Finally, I learned how to bridge JavaScript with HTML and CSS.

2. Improved Debugging Skills

Debugging has been one of the most significant areas of improvement. With my previous minimal coding experience, my lack of debugging strategies was the biggest chokehold to progression. I was very reliant on forking other people's code and making minimal value changes to create my projects, and often gave up on adding complexity when I encountered errors. Through this class, however, I learned how to use the console effectively, interpret error messages, and apply `console.log()` to track when functions are being called. I also became familiar with common issues such as inconsistent naming, incorrect syntax, uninitialized variables, and attempting to use p5 functions outside their expected structure. As a result, I became much more efficient at solving issues as they arose.

3. Introduction to Datatypes

In this class, I was introduced to many new datatypes such as conditional statements, arrays, events, and p5 functions. For example, I employed arrays in my Mod Jam to store text strings and in my Variation Jam to store a large amount of circle data. I later called upon these indexed circles and changed their variable value (color and position). I was also introduced to basic gamification elements such as scoring of which I used in my Mod Jam by inputting scoring results in conditionals to trigger different effects (i.e. lightening screen). In all of my project jams, I repeatedly used `dist()` for detecting overlap and `mouseClicked()` to create buttons which changed screens or screen elements. While I have acknowledged my improved skills in these aforementioned areas, I do believe I am not fully confident in other concepts such as timing mechanisms, while loops, and sine waves as I didn't implement them in my projects.

Programming and My Creative Practice

1. Interactive Art and Audience Engagement

Before this course, my work was mostly visual and non-interactive. In my photography and illustrative work, I focused on leveraging different elements of art and principles of design in order to create a certain impact on the audience. However, through learning to code my approach to art has shifted entirely. I can now make work that reacts directly to the audience, where their actions change the experience and visual outcome. I have experimented with how events such as `mouseClicked`, `keyPressed`, `mouseX` and `mouseY`, and `acceleration` can create different audience engagement. This requires a different mode of thinking, as my piece isn't fixed or stagnant. Now the focus is on what moments can I create for the user and how will that push them to engage differently with the work. An example of this knowledge in use is in my Art Jam, I used `mouseX` and `mouseY` in certain areas to bend the design and push users to click a button which would change the screen and bring them to a new page.

2. Experimentation and Iteration

Coding has changed the way I approach experimentation. In my previous artistic practice, jewelry design, there was a large emphasis on extensive planning and precision. This level of precision comes from working with expensive materials and material limitations, for example, due to the difference in hardness opals cannot be set in white gold. While this precision is necessary it also limits experimentation. With programming on the other hand, there are no high consequences to mistakes and experimentation, I can iterate quickly and also include functions such as `random` to include chance as a part of the work. Changing one variable can also produce a completely new result instantly. This has made me less attached to what the final outcome "should" be, and more open to unexpected results. In my Variation Jam I have been especially interested in playing with how `random` can change element position and fill color. I've also enjoyed playing with loops and seeing what patterns I can create with changing variables. With these new conditions, I now try to work through a program, meaning I break down all its components and test all their possibilities before settling on a final product.

3. Working in a Digital Environment

Coming from practices grounded in material and physical constraints, the shift into a digital environment opens so many possibilities. With code, I can create systems and visual effects that don't need to obey gravity, material tolerances, or any real-world limitations. This includes giving odd movements to elements such as using `acceleration` and `position` limits to create a bouncing ball effect or in my Mod Jam adding a buzziness index to make an element shake as it moves. The complexity I can achieve digitally is also often beyond what would be possible in a physical medium, for example I can create a mandala design with a loop in which you can nearly infinitely scroll into the design. This opens up a wide range of new artistic possibilities that I couldn't access before.

Future Directions as a Creative Coder

1. Giving Thanks to Pippin

One of the most important outcomes of this course has been developing a more open understanding of what coding can be. I appreciated learning in an environment that did not reinforce narrow expectations of what coding should be. The open-ended structure of the projects encouraged experimentation rather than directing me toward a predetermined form. This pushed me to think creatively and focus on taking small ideas and pushing them to their limits.

2. Artistic Direction

Definition - Traditional Game Design: Game design which often relies on defined goals and fixed rules.

I've realized that my interest in creative coding sits less with traditional game design and more with building interactive visual systems. What draws me in is the potential to use iterative processes to generate unexpected patterns and evolving forms. I'm particularly interested in how mathematical functions (sine and square root) and looping logic can be pushed to produce complex, visually compelling patterns that respond to user interaction.

3. Interdisciplinary Aspirations

Creative coding has also opened up several interdisciplinary directions that connect to my existing artistic practices.

Printed Work and Visual Collections

One project I hope to pursue is a printed book of generative visuals created in p5.js, paired with the code that produced them. I am also interested in integrating artistry in the code's comments specifically through poetry or reflective writing, a concept I first encountered through Tasha's approach to commenting.

Narrative Video Experiments

A more ambitious idea involves combining video with eye-tracking data. I am interested in using a viewer's eye movements as a kind of paintbrush, producing visual traces over video footage. Applying this to old personal videos could produce a compelling contrast between the original camera framing (representing the past perspective) and the eye-tracking layer (representing the present one). For example, imagine a video taken years ago of a group of friends hanging out when your first love was just an acquaintance. In the original recording your first love would appear only as a side character, but the eye-tracking layer would show your gaze snapping to them every time they enter the frame, lingering until they disappear again. In addition, sound such as synthesized tones could be added that reflect emotional responses deepening the narrative quality.