

Carael Bandojo  
Pippin Barr  
CART 253 A 2242  
December 5, 2024

## Post Reflection

Though it was only a short few months, I do believe this is probably the best programming class I've had out of the three in my life. Not for any flattery reasons, but because somehow my previous teachers never felt the need to explain why we were writing the code that we were writing. The "learning to think like a computer" component of the course was by far the most beneficial knowledge I gained from here. Now I can say I know why I do things a certain way.

At the start of the course, I had taken two courses in Java, both of which required us to use odd IDEs that were either not powerful or outdated. The first 10th grade class I had was solely using Replit, and it was alright for the things we were learning, but it was always a laggy site and I never understood the information given by our teacher. In the 11th grade we only used Eclipse, which has an insanely saturated UI. But our teacher told us to just figure it out, so I never really got to learn how it works. Overall not a very fun learning experience. I felt very out of my element for both of those semesters, because the class was very clearly geared to people already experienced in programming that wanted to get an easy grade. Even though I made some friends in that class, they'd tell me that too. That they were only there because they already knew what they were doing, and that it was another easy grade to add to their portfolio for university.

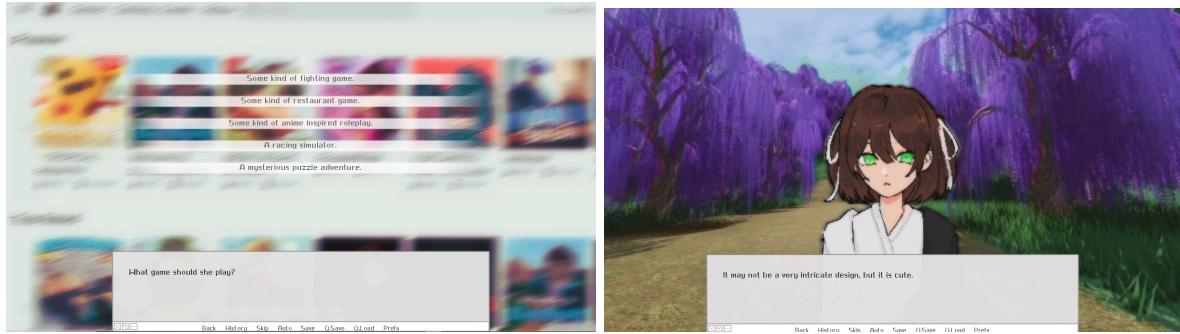
That course never let us really explore anything fun. We were literally made to make programs that make complex calculations or things like that. I think the most fun I had in that course was making a sketch for a UI of a group project we made from scratch. It was really cool to see the designs I made become interactive, and I felt really bad that my group mates had to carry most of the coding because I just didn't know how to do it. Now I can say that I do, and it's so very fun to implement weird ideas into games! I've also found that I actually do enjoy coding when I know what I'm doing. It feels like a combination of drawing and solving a puzzle. I think the funnest project was Mod Jam, since at that point I was reminded of most of the things I learned in my prior years but with a better understanding of them, and with a good understanding of the library. I remember struggling for a bit with the random function, since I'm really used to using Math.random. Figuring it out felt like getting kicked in the face, so from then on I started paying more attention to the documentation. I also forgot how useful functions and trying to figure out modular code can be! I remember a friend telling a group of



people that he wrote a program that was 1000 lines and they just laughed at him. At the time I didn't understand why, but now I do! Programming is one of the few times in academics where you want to make it as short as humanly possible. Hence fun stuff like loops that will do repetitive calculations for you. Plus the ability to just make a function that says [if {(number % 2) = 0 } //it is even}], so you don't end up like that guy. Though I do understand the usage of for loops for things that are really just repeated sequences for however long your loop is, I would like to understand while loops better. !

Like with every brushstroke I have to solve an equation to get my desired outcome. That enjoyment really opens my world up to the possibilities of making more interactive art. Design is something I'm really comfortable with, so now I feel like I'm just trying to get my coding up to the same speed. I'd like to be able to just sit down, think of an idea, and put it out onto an IDE without stopping like I can with a drawing.

Realistically though, that is a very long term goal. In the meantime, I'll be trying to make manageable projects for myself, somewhat akin to the weekly exercises in class but on a bigger scale. Recently, I decided to make visual novel for my friend's birthday using the software Ren'py. I personally really enjoyed the process. Decisions are just if statements that call functions to bring up different parts of the story you wrote. There's two choices in the game that branch into four unique choices, and then eight different endings for each of those ending choices. I didn't know python before that project either, but I'm slowly learning how to use a very simple version of it with the Ren'Py library. Doing that really made me wonder what I could do now that I feel like I have a good grasp on some fundamentals, so I'm going to challenge myself to make a much longer visual novel on the side during 2025, with maybe some kind of minigames, since I know the software is capable of handling any sort of python.



```

menu game_choice_1:

    "What game should she play?""

    "Some kind of fighting game." if not fighting_exp:
        $ fighting_exp = True
        "She opens the what appears to be a Poku no Zero Macadamia game, some cheap spin off."
        "She spends the day doing that, which somehow makes her feel more confident in her combat knowle
        "{b}<This will probably be useful later. >{/b}"
        "{b}<SKILL ACHIEVED: CRP>{/b}"
        jump game_choice_1

    "Some kind of restaurant game." if not restaurant_exp:
        $ restaurant_exp = True
        "She opens what appears to be a game about working at a pizza shop. It's a bit dull."
        "It feels like a waste of time after an hour."
        "So without gaining anything, she moves swiftly onto the next experience."
        jump game_choice_1

    "Some kind of anime inspired roleplay.":
        jump demon_slayer_1

    "A racing simulator." if not racing_exp:
        $ racing_exp = True
        "She fires up a racing game, spending hours meticulously tuning virtual cars."
        "The precision of adjusting gear ratios and suspension feels oddly satisfying."
        jump game_choice_1

    "A mysterious puzzle adventure." if not puzzle_exp:
        $ puzzle_exp = True
        "An indie puzzle game catches her eye. The cryptic narrative and challenging mechanics draw her
        "Hours pass as she unravels complex mysteries and solves intricate puzzles."
        "{b}<Her problem-solving skills might have improved.>{/b}"
        "{b}<SKILL ACHIEVED: Puzzle Solving>{/b}"
        jump game_choice_1

return

```

I'm not sure if I can own the role as a creative coder quite yet. I believe anyone who draws at all is an artist, so technically I feel like I'm already a creative coder, just probably not one at a very high level of experience. Practice will surely make me feel more confident, even if it's not something overly ambitious and is just a visual novel. I'm very inspired by works like DDLC by Team Salvato. The game uses a lot of sound effects and effects provided by the library in Ren'Py, so I'm really looking forward to exploring that and seeing what I can come up with. Also thinking of exploring different ways to branch out the story for the player, like perhaps storing

variables for how much a character likes the player, or even going as far as messing with how the save and load functions work. These are things I wouldn't have thought of if I didn't understand the code I was doing. I think the design and writing aspects of it will allow me to easily bridge myself into wanting to work on it more consistently, and hopefully from that I'll be able to challenge myself more technically.