# Pledge : Surya Prasad Parajuli

Introduction

In Plato’s Meno, Socrates poses the fundamental questions: how do we know what we know, and how do we recognize that we need to learn more? These questions are especially meaningful when I think about my experience in Game Programming with Python, a course I’m taking as part of my first year as an international student at Lyon College. Throughout this course, I worked on creating a simple number-guessing game using Python and Google Colab. The process of learning to program has been both exciting and challenging, and it has allowed me to reflect on the idea of knowledge in the same way that Socrates and Meno discuss in the dialogue. This essay will explore how I’ve learned through programming, where I experienced struggles, and what I still need to learn, all through the lens of Socratic questioning.

Learning Through Game Programming with Python

When I first started Game Programming with Python, I had little to no experience with coding. The idea of using a programming language to create something interactive seemed both exciting and intimidating. The first project we worked on was a simple number-guessing game, where the computer picks a random number, and the player has to guess it correctly. It wasn’t a complicated game, but for me, it was a huge learning experience.

I remember feeling very uncertain at first. I had to learn how to write Python code, understand basic logic, and figure out how to use Google Colab, which was new to me. At first, the syntax seemed like a jumble of strange words and symbols, but slowly, I began to make sense of it. For example, I learned how to use the ‘`input()`’ function to take guesses from the player and the ‘`random.randint()`’ function to generate the random number. These concepts felt like small victories because I could see the result on the screen once the program ran.

Working on this simple project was my first experience of turning abstract ideas into something tangible. I could see how the program worked, step by step, and this made me realize that programming is about breaking down problems into manageable pieces. This felt very much like what Socrates describes in *Meno*, when he helps Meno’s slave boy figure out a geometric problem by questioning him step by step. In a similar way, by following the logic and structure of the game, I could see how each piece of the code worked to solve a problem.

Experiencing Growing Pains

Despite the excitement of learning, the process was not without its challenges. I found myself struggling with bugs and errors, especially when the game wouldn’t behave as expected. In one instance, after entering a guess, the game wouldn’t stop even after the correct answer was entered. It was frustrating to see the game running endlessly, but this is where I began to understand the value of debugging. I had to go back to my code, check each line, and figure out why the program wasn’t working as it should.

This process of identifying and fixing mistakes reminded me of the way Socrates helps Meno understand that knowledge isn’t something that can be handed to us, but rather something we have to work through ourselves. Socrates doesn't give Meno the answers but instead guides him to find the right solutions. Similarly, when my program wasn’t working, I had to ask myself the right questions, such as: “Did I use the correct syntax?” or “Did I make sure the `if` condition for winning the game was properly set up?”

Another challenge I faced was understanding the structure of the code. For example, I had to figure out how to properly loop the game so that the player could keep guessing until they found the right number. Learning about loops and conditional statements was a key moment for me. It wasn’t enough to simply understand the logic of the game; I had to know how to control the flow of the game using while loops and if statements. This was like learning the rules of a new language or understanding a new system, and at times it felt overwhelming. But over time, I began to see how all the pieces fit together, and that made the experience much more rewarding.

Recognizing New Learning

One of the best ways to know you’ve learned something new is when you can explain it to someone else or apply it in a new context. After completing the number-guessing game, I felt confident enough to explain the basic code to my friends. I could show them how I used the `random` module to generate numbers and how I used loops to keep the game running. This was a clear sign that I had absorbed the information and could apply it beyond the classroom.

In Meno, Socrates teaches Meno that knowledge is not simply something you acquire, but something you recognize through experience and inquiry. In my case, I didn’t just learn how to write a number-guessing game; I also learned how to solve problems on my own. I realized that programming isn’t just about typing code; it’s about thinking logically, breaking down problems, and finding solutions. This was my "aha" moment, when I could see how far I had come in understanding how a computer processes information.

What I Still Need to Learn

While I’ve learned a lot, I know there’s still so much more to discover. My number-guessing game was only the first step, and I have many more challenges ahead. For example, I’d like to move beyond simple text-based games and create games with graphics and more complex interactions. I still need to learn more about using libraries, handling user input, and learning about advanced features like artificial intelligence or game physics.

I also want to become more proficient with Python itself. While I’ve learned the basics, I know I still struggle with certain parts of the language, such as more complex data structures or understanding the best way to structure a program. I also want to become better at debugging my code. Even though I can fix simple bugs now, I still find myself confused when more complex errors occur.

I know I have more to learn because every time I encounter a new problem, I find myself Googling solutions or reading tutorials to understand it better. This is similar to Meno’s realization that learning is a continuous process. As Socrates guides Meno to recognize that knowledge is a never-ending journey, I, too, recognize that I am only at the beginning of my programming journey.

Conclusion

Both Meno and my experience with Game Programming with Python have taught me valuable lessons about knowledge and learning. In the same way that Socrates uses questioning to help Meno discover knowledge, I’ve learned that programming involves breaking down problems and asking the right questions. The number-guessing game was a small but significant step in my journey of learning to program, and while I’m proud of what I’ve accomplished so far, I know there’s still much more ahead. Just as Meno’s journey towards understanding is not complete, my own journey of learning programming is ongoing. With each new challenge, I gain a deeper understanding, and I look forward to continuing the process of discovery and growth.