Reflective Report - Project 1

Nicolas C. Panopoulos
C5059389
CSC-1034

During the development of the "Walking Panda" project for CSC1034, I had the opportunity to apply and understand several fundamental concepts of software engineering. This project was not only about writing Python code, but also about learning how to structure, document, and manage a small software project from start to finish.

At the beginning, I was slightly confused about how to set up the project correctly. I learned how to use GitHub to create a repository from a template, clone it into PyCharm, and make commits and pushes to keep my code up to date. This was my first experience managing a project using version control properly, and I now understand how useful Git is for tracking changes and maintaining organized work.

Another important part was organizing the Python files into packages and modules. I had difficulties with imports and understanding how Python recognizes directories as packages. After learning about the purpose of the __init__.py file and avoiding name conflicts between files and folders, I managed to fix these issues and structure the project correctly. This helped me see how important clean architecture is in programming.

Installing dependencies through the requirements.txt file also helped me understand how software can be made reproducible across different systems. I used the Panda3D library to create a 3D animation of a walking panda, which was new for me. Running the 3D environment and seeing the animation work correctly gave me a real sense of progress and achievement.

One of the biggest lessons from this project was how important documentation is. I wrote a clear README.md explaining how to install dependencies and run the program, and I included a license file as required. I also learned that writing good commit messages and keeping files organized are part of being a professional software engineer, not just a programmer.

Overall, this project taught me much more than expected. I learned to use GitHub confidently, to organize a Python project using modules, to install and use external libraries, and to write technical documentation. Even when facing errors, such as import problems or missing files, I learned to read error messages carefully and solve issues step by step.

In conclusion, this project helped me connect programming with real software engineering practices. I now understand that successful software development involves not only coding, but also structure, documentation, and continuous reflection about how to improve.