Enhancement One: Software Design/Engineering Narrative

By Sam Blanton

CS-499

July 21, 2001

Brief Description of the Artifact

The artifact I selected for enhancement is an Out-of-Box Experience (OOBE) automation script initially written in Python. This script was created for Innovative Computer Solutions to automate the initial setup of computers, bypassing the need for manual configuration. The original implementation involved a client-server setup where a microcontroller would communicate with a server running the automation script. The script leveraged computer vision to detect UI elements and interact with them accordingly. The purpose was to reduce manual intervention in the setup process, which had been problematic with traditional imaging tools on our standard Dell Optiplex machines.

Justification for Inclusion in ePortfolio

I chose this artifact because it demonstrates my proficiency in both Python programming and computer vision, key skills in the realm of software development. The original script showcased my ability to solve real-world problems through automation and innovative use of technology. By converting this script to C++, I aimed to enhance its performance and scalability, aligning it more closely with industry standards for standalone applications. This conversion also highlights my versatility and ability to work with different programming languages and paradigms.

Meeting Course Outcomes

This enhancement aligns with the course outcomes by demonstrating:

1. **Algorithmic Principles and Computer Science Practices (Outcome 3):**
    - Designing and evaluating a computing solution (OOBE automation) that solves a specific problem.
    - Managing trade-offs involved in design choices, such as moving from a client-server model to a standalone application for simplicity and scalability.
2. **Innovative Techniques and Tools (Outcome 4):**
    - Utilizing advanced C++ programming techniques.
    - Integrating machine learning models (YOLO) for object detection.
    - Ensuring robustness, maintainability, and scalability in the software engineering process.

Reflection on the Enhancement Process

The enhancement process involved several key steps and learning experiences:

1. **Model Conversion:**
    - The initial challenge was converting the YOLO model from a .pt file to an ONNX format compatible with C++. This required in-depth research and understanding of model conversion tools and formats.
2. **Image Processing and Detection:**
    - Implementing code to read images and display detected objects using OpenCV. This step involved translating Python-based computer vision logic into efficient C++ code.
3. **Screen Capture and Display:**
    - Integrating OpenCV to capture screenshots and display detected elements on the screen. This was crucial for visual feedback during the automation process.
4. **Handling Coordinate Scaling:**
    - A significant hurdle was managing Windows scaling, which affected the accuracy of x and y coordinates for clicks. The solution involved making the program DPI-aware to handle scaling properly.
5. **Implementing Input Simulation:**
    - Transferring the logic for clicking buttons and typing text based on detected elements. This required precise control over input simulation to ensure the automation was reliable.
6. **Compiling and Dependency Management:**
    - Ensuring all dependencies and DLLs were correctly included for the program to run on a fresh Windows installation. This step highlighted the importance of thorough testing and dependency management in software development.

Challenges and Learning

Throughout this enhancement, I faced several challenges, particularly with coordinate scaling and dependency management. Overcoming these challenges required persistence and problem-solving skills. I learned the importance of understanding system-specific nuances, such as DPI settings, and the necessity of thorough testing across different environments. The process also reinforced the value of research and continuous learning in software development, especially when dealing with cross-language implementations and integrating machine learning models into applications.

This enhancement not only improved the original artifact but also expanded my skill set, making me more efficient at tackling complex software engineering problems. The successful conversion of the Python script to a standalone C++

application demonstrates my ability to deliver efficient and scalable solutions, a critical competency in the field of computer science.