integrating modern development practices and educational robotics

1st Raveanu
IT
HTL Wien West
Vienna, Austria
raveanu.r20@htlwienwest.at

2nd Lama
IT
HTL Wien West
Vienna, Austria
lama.n20@htlwienwest.at

3rd Petrovic
IT
HTL Wien West
Vienna, Austria
petrovic.n20@htlwienwest.at

***Abstract*—This paper delves into the integration of modern software development practices within educational robotics projects, highlighting our team's journey in enhancing collaboration and efficiency through specific tools and programming languages. We detail our transition to Python from traditional C programming, and our utilization of Git, GitHub, and Visual Studio Code Server on a Raspberry Pi. This shift not only facilitated a more collaborative development environment but also broadened the accessibility of STEM projects to a wider audience. Additionally, we provide a thorough examination of the Python Robotics Toolbox [1] as a potent educational instrument, discussing both its implementation challenges and its benefits for learning and project development.**

Keywords— Software Development, Educational Robotics, Python, Git, GitHub, Visual Studio Code Server, STEM

I. Introduction

The evolution of software development tools and methodologies has significantly impacted the efficiency and inclusivity of STEM education projects. In this context, our paper outlines our experience in integrating Git, GitHub, and Visual Studio Code Server on a Raspberry Pi into our Python-based educational robotics project. This narrative serves as a case study for modernizing development practices away from traditional C-based approaches and discusses the broader implications for STEM education.

II. Software Development Practices

A. Adopting Git and GitHub

Our project embarked on a path to incorporate Git and GitHub for version control and collaboration. This transition allowed our team to streamline our development process, enabling simultaneous coding efforts, easier integration of work, and a more efficient resolution of conflicts. We provide a detailed account of setting up our repository, managing branches for different features, and the adoption of a continuous integration/continuous deployment (CI/CD) pipeline to automate testing and deployment.

B. Visual Studio Code Server on Raspberry Pi

The installation of Visual Studio Code Server on the Raspberry Pi was a pivotal move towards enhancing our development workflow. [2] This infrastructure facilitated a seamless development environment that is accessible to all team members, irrespective of their physical location, and contributed significantly to the swift progression of our project.

III. Embracing Educational Robotics

Choosing Python over C was a strategic decision aimed at leveraging Python's simplicity and its vast ecosystem of libraries. The Python Robotics Toolbox [1] emerged as a central element in our project, offering advanced functionalities for robotics simulations and algorithm development.

IV. Engineering and STEM Projects

Our venture into educational robotics, while not primarily focused on engineering principles, naturally incorporated several key engineering concepts. This section details the interdisciplinary approach of our project, illustrating how software engineering practices were melded with robotics to achieve our educational goals. We discuss the impact of these practices on the project's success and the students' learning experience.

V. Conclusion

The integration of modern software development practices has significantly enhanced the collaborative and educational aspects of our STEM project. Through the adoption of Python, Git, GitHub, and Visual Studio Code Server [2], alongside a detailed exploration of the Python Robotics Toolbox [1], we have set a precedent for future educational robotics projects. Our experiences offer valuable insights and recommendations for the STEM community, aiming to foster a culture of innovation and collaborative learning.

Acknowledgment

The authors would like to thank Prof. Wurzer and HTL Wien West for their guidance and support in the Botball program.

References

[1] https://github.com/petercorke/robotics-toolbox-python

[2] https://www.digitalocean.com/community/tutorials/how-to-set-up-the-code-server-cloud-ide-platform-on-ubuntu-22-04