TealPlay: A Media Center for Experiential Learning

Michael A. Murphy
Megan Hickman Fulp
mmurphy2@coastal.edu
mlhickman@coastal.edu
Coastal Carolina University
Conway, South Carolina, USA

ABSTRACT

TealPlay is an MIT-licensed open source media center application that is designed to be accessible to lower-level undergraduate computing students. TealPlay provides an opportunity for these students to work on a large program early in their academic careers, instead of focusing exclusively on small example problems, which could build student confidence and improve motivation and satisfaction. This application is designed to solve a pedagogical chicken-and-egg problem in which students could benefit from experience with a larger application, but larger applications tend to be inaccessible without experience. TealPlay can provide opportunities to practice communications, project management, tool use, and testing skills.

CCS CONCEPTS

• **Software and its engineering** → *Maintaining software*; **Open source model**; • **Applied computing** → **Interactive learning environments**; • **Social and professional topics** → *Computational thinking*; **Software engineering education**; *Information technology education*; *Computer science education*; *CS1.*

KEYWORDS

experiential learning, large application, project management, development tools

ACM Reference Format:
Michael A. Murphy and Megan Hickman Fulp. 2025. TealPlay: A Media Center for Experiential Learning. In *Proceedings of ACM Mid-Southeast Conference.* ACM, New York, NY, USA, 3 pages.

1 INTRODUCTION

Undergraduates who elect to major in the computing disciplines often begin their postsecondary education with little to no prior experience in programming or software engineering. To teach programming, faculty normally create self-contained exercises that are small enough to assign and grade efficiently. As a result, students may not have an opportunity to practice on a large-scale application prior to graduation unless they acquire an internship or engage in undergraduate research. As a consequence, new graduates from the computing disciplines often lack both technical and soft skills that industry desires.[10] A significant number of undergraduates also lack the confidence necessary to apply for industry internships, where they would otherwise develop these skills.[3]

Open source software development provides an opportunity for students to develop professional skills within an academic setting.[8] While students could join existing projects in the broader open source community, their ability to make significant contributions may be impaired until they have gained enough experience to understand the code, which presents a chicken-and-egg problem. TealPlay is an open source media center application designed as a vehicle for teaching basic programming and software engineering skills. Its purpose is to serve as a relatively large code base to give entry-level students hands-on experience in a controlled yet realistic scenario, allowing students to learn while simultaneously addressing confidence and motivation issues.

2 RELATED WORK

There is a dichotomy between the academic need to teach basic computing fundamentals from the ground level and industry expectations for new graduates, particularly with respect to communications skills, project management ability, use of software tools, and testing both software and hardware systems.[10] One potential solution is to create a dedicated course that explains how to prepare a development environment, navigate through an existing large code base, make modifications, and communicate changes effectively through a "Cognitive Apprenticeship" approach.[11] While this technique may be appropriate for computer science and information systems majors, deficiencies have also been observed in graduating information technology majors.[10]

Individual students can gain missing skills and obtain competitive advantages by participating in industry internships prior to graduation. Computer science students find internships rewarding for skill development and industry-standard practice, especially when their work impacts real users.[5] In spite of these benefits, fewer than 40% of computer science and computer engineering majors at a public research university have been found to pursue internships, and these numbers would not be expected to be better at comprehensive institutions. Of the students who do not pursue internships, half feel unprepared, believe it's not yet relevant at their class standing level, or lack the confidence to apply. The other half report other difficulties, including family responsibilities, a prior bad experience with rejection, health problems, and so forth.[3] In the near future, students may become even more pessimistic if companies reduce expenditures on internships and entry-level positions in order to invest in Artificial Intelligence (AI).[7]

One way for academia to provide an alternative experience for non-interning students – and to improve the competitiveness of students who do seek internships – is to provide opportunities to work on larger applications that appeal to a broad user base. Open Source Software (OSS) provides one method for exposing students to applications that transcend the "toy projects" typically found

ACM Mid-Southeast Conference, November 13–14, 2025, Gatlinburg, TN
2025.

in classroom assignments.[8] Surveys of industry professionals indicate that prior participation in OSS development is beneficial during the hiring process, as it indicates that students with such experience have the ability to work on larger applications.[4] Several approaches to incorporating OSS development are found in the literature, at least with respect to software engineering and development. Students with sufficient programming ability can be guided to contribute to established open source projects, where they learn tools and techniques through experience.[8] Alternatively, professors can create courses around OSS development skill sets, employing hands-on laboratory exercises to teach specific related skills.[2] One other approach, which can align well with traditional academic research, is to engage interested students to work on an open source project that is developed at their university for academic credit in lieu of taking one or more traditional courses. This approach enables students to work on OSS development for longer periods of time, although the participation rates may be low.[6] TealPlay is designed to broaden the appeal of working on an OSS project to reach a larger number of students who might otherwise be intimidated. Since today's students have grown up consuming multimedia, a media center application has been chosen for its intuitive functionality.

3 IMPLEMENTATION

TealPlay is written in Python using Qt for Python[9] for the graphical user interface (GUI) toolkit. The initial implementation contains more than 14,000 lines of source spread over approximately 55 files. Within the main package are 12 top-level modules, where each module corresponds to a single Python source file. The remaining source files are organized into 8 sub-packages, several of which have their own descendant packages.

Despite the size and complexity of this application, TealPlay is logically divided into modules and packages with descriptive names. For example, the `constants` module stores project-wide constant values, while the `playlist` package manages playlists. This organization is designed to help students locate implementation files for major features. Each file uses descriptive class, method, and function names, with Python documentation strings explaining their purpose.

Figure 1 depicts the main window of the TealPlay graphical user interface (GUI). The interface organizes content into three areas, with the video player and media playlist at the top, playback controls in the middle, and library browser at the bottom. Each part of the interface (except for the video) may be hidden by means of dropdown menu items or corresponding shortcut keys. By selectively displaying sections, TealPlay can function as either a media center or as an ordinary video player.

The application includes additional windows, including a settings window for customizing library and player options, a dialog window to add randomly selected library items to the playlist, and a help window. Settings are stored in a human-readable INI file at a platform-appropriate default location. Help documents, written in a subset of HTML, offer one possible avenue of participation for students more interested in system support than software development.

To make the project more accessible for testing, debugging, and development activities, TealPlay uses Python's standard logging facility, which been extended to provide an in-application log window to display the log contents. The logging verbosity may be set at application start time using command-line options, and it may be changed during the course of execution via the log viewer window. When the log level is set to "Debug," the origin package name, module name, and source line is shown with each log message. In addition, the log viewer window provides a text search capability. These features are intended to guide students to the correct source file and aid in tracing application behavior.

The scope of debugging statements in the released version is designed such that critical sections of the code that are likely to contain unintentional bugs are easy to trace, while less critical sections are left uncovered. For example, the library scanning feature has more complete coverage in the released implementation due to risks of coding mistakes and file system issues. In contrast, the help system lacks any logging functionality, as its code path poses fewer risks, making it an ideal place for freshmen to add statements.

Other areas of TealPlay that have been left for future student projects include the addition of user profiles, audience filtering, and library browser improvements. TealPlay currently supports a single library that can span multiple directories on the file system. User profiles would permit portions of the library to be restricted by user, which would enable audience filtering for parental controls. Since TealPlay's GUI design generally follows the Model-View-Controller (MVC) pattern, multiple user interface improvement projects could be assigned. For example, students could work to replace the tree-based library browser with a more visual library similar to those found on popular streaming services. Additional development projects could involve supporting alternative playlist formats, music visualization, remotely loaded subtitles, and dynamic file type detection. Projects suitable for information technology majors could include documentation, support exercises, platform-specific deployments, packaging, and usability testing. Moreover, students may identify their own features or contributions beyond this initial set of ideas.

4 CONCLUSIONS AND FUTURE WORK

TealPlay is a relatively large, open source media center that is intentionally designed to be accessible to computing majors early in their academic careers, giving them hands-on experience in areas that are important to industry. Participation in projects related to this "real" code base may help to inspire confidence, improve motivation, and deliver a sense of satisfaction that would be more difficult to achieve with smaller programming problems.

Future research work will include gradually integrating TealPlay into freshman-level programming courses. Instructor experience is likely to identify unexpected pedagogical challenges, which may yield new research problems. Additional work will involve utilizing TealPlay in software engineering courses, since it can provide a hands-on platform for teaching testing, project management, and tool use. In order to measure the broader impacts of TealPlay across different student populations, the software is released under the open-source MIT license as a means of welcoming collaborators.[1]

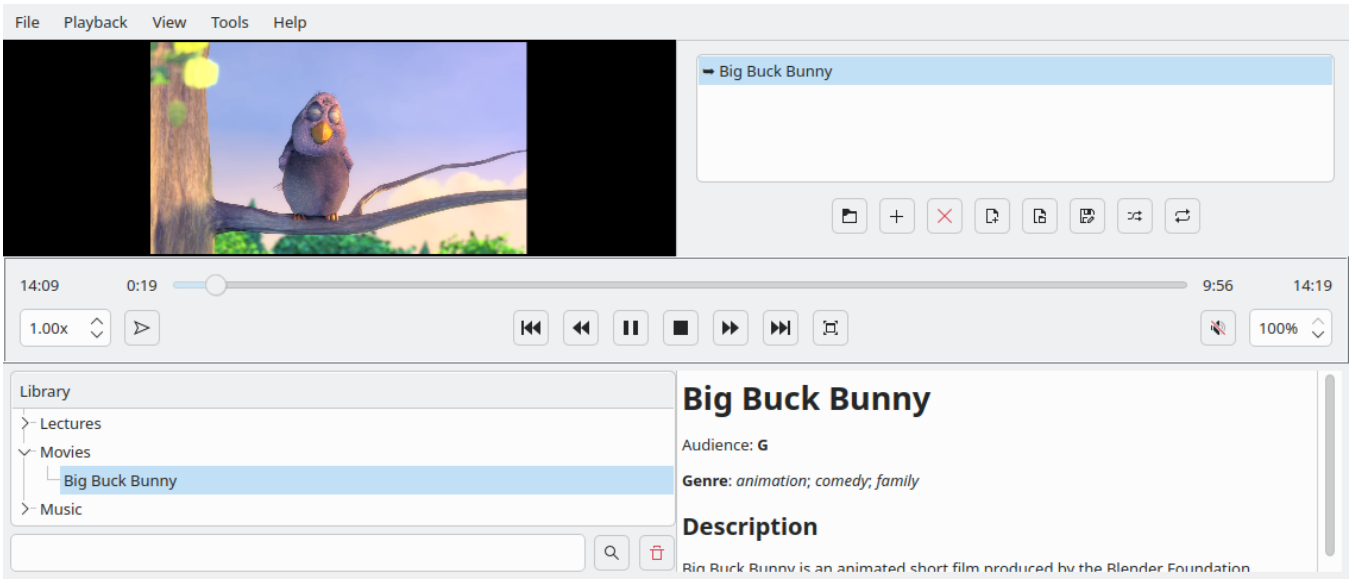

Figure 1: The TealPlay main window displays the video, playlist, controls, and media library.

REFERENCES

[1] Coastal Carolina University. 2025. TealPlay. https://code.coastal.edu/open-source-lab/tealplay

[2] Mohsen Dorodchi and Nasrin Dehbozorgi. 2016. Utilizing open source software in teaching practice-based software engineering courses. In *IEEE Frontiers in Education Conference (FIE)* (Erie, Pennsylvania).

[3] Amanpreet Kapoor and Christina Gardner-McCune. 2020. Barriers to Securing Industry Internships in Computing. In *ACE'20: Proceedings of the Twenty-Second Australasian Computing Education Conference* (Melbourne, Australia). 142–151.

[4] Ju Long. 2009. Open Source Software Development Experiences on the Students' Resumes: Do They Count? - Insights from the Employers' Perspectives. *Journal of Information Technology Education: Research* 8, 1 (1 2009), 229–242.

[5] Mia Minnes, Sheena Ghanbari Serslev, and Omar Padilla. 2021. What Do CS Students Value in Industry Internships? *ACM Transactions on Computing Education* 21, 1 (3 2021), 1–15.

[6] Matthias Müller, Christian Schindler, and Wolfgang Slany. 2019. Engaging Students in Open Source: Establishing FOSS Development at a University. In *Proceedings of the 52nd Annual Hawaii International Conference on System Sciences* (Grand Wailea, Hawaii).

[7] David Unyime Nkanta. 2025. New Study Indicates AI Might Already Be Displacing Entry-Level Tech Positions. *International Business Times UK* (5 2025). https://www.ibtimes.co.uk/new-study-indicates-ai-might-already-displacing-entry-level-tech-positions-1734619

[8] Gustavo Pinto, Clarice Ferreira, Cleice Souza, Igor Steinmacher, and Paulo Meirelles. 2019. Training Software Engineers Using Open-Source Software: The Students' Perspective. In *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)* (Montreal, Quebec).

[9] Qt Company. [n. d.]. Qt for Python. https://www.qt.io/qt-for-python

[10] Alex Radermacher and Gursimran Walia. 2013. Gaps between industry expectations and the abilities of graduates. In *SIGCSE '13: Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. Denver, Colorado, 525–530.

[11] Anshul Shah, Jerry Yu, Thanh Tong, and Adalbert Gerald Soosai Raj. 2024. Working with Large Code Bases: A Cognitive Apprenticeship Approach to Teaching Software Engineering. In *SIGCSE 2024: Proceedings of the 55th ACM Technical Symposium on Computer Science Education* (Portland, Oregon), Vol. 1. 1209–1215.