Engaging Students in Undergraduate Research: Teaching Through Design, Development, and Collaboration

Ethan Widener
Mercer University
Macon GA, United States
widenere42@gmail.com

Soumik Kundu
Mercer University
Macon GA, United States
soumikkunduusa@gmail.com

Rajwol Chapagain
Mercer University
Macon GA, United States
rajwol.chapagain@live.mercer.edu

Prakriti Sapkota
Mercer University
Macon GA, United States
prakriti.sapkota@live.mercer.edu

Martin Q. Zhao*
Mercer University
Macon GA, United States
zhao_mq@mercer.edu

***Abstract*—Incorporating quality research in a college setting where undergraduate teaching is the primary goal is always challenging. Engaging undergraduate students in research-related activities using a research theme to arrange teaching materials and assignments can be beneficial. Students can be introduced to basic concepts in lower-level courses and later become effective research assistants. This paper presents such practices at Mercer University's Computer Science Department from the students' and faculty's perspectives. The focus is on how to plan for breaking down the needs of research projects among student teams in various courses and provide collaboration opportunities between faculty and student researchers. Activities related to a research project supported through a NSF grant show satisfactory results for both students and the faculty advisor.**

Keywords—Collaborative Learning, Teaching Pedagogies, Student Experiences, Incorporating Research in Undergraduate Teaching, Extracurricular Problem Solving

I. Introduction

The Computer Science Department at Mercer University is hosted in the College of Liberal Arts and Sciences. The department offers undergraduate courses through its Computer Science (CSC), Information Science and Technology (IST), Computational Science (CPS), and Cybersecurity (CYS) programs, with its CSC BS program accredited by ABET since 1998.

Students engaged with undergraduate research overseen by Mercer University's Department of Computer Science are part of an ongoing effort to enhance the student experience and provide students the opportunity to learn and expand their skill sets outside the classroom. A myriad of undergraduate research projects sponsored by department faculty have been made available to students at all points in their undergraduate careers, whether they are just beginning a Programming II class, or expecting to graduate following Software Engineering. These projects have been integrated into many of the courses offered by the department as a way to expose students to the opportunities available to them and to get them comfortable with working on large-scale projects. Dedicated and interested students are encouraged to continue work on these projects outside the scope of their regular coursework, building off of the contributions of previous students as progress is made towards each project's completion. During their time as research assistants, students are organized into fluid teams to facilitate collaboration and learning. Each team is advised to work closely with each other and borrow members of other teams to complete tasks and work through problems as needed. Students are encouraged to work on multiple sides of the project they are involved with. One week they may be involved with the software engineering aspects of a project, and then the next they are working with the user experience team to ensure that what has been produced is of quality. In this way, students are able to utilize their diverse backgrounds, skills, and interests to not only contribute to real-world projects but also to learn in an organic manner that transcends the standard restrictions of a classroom setting. Students are advised and guided by faculty but are given the freedom to experiment and learn on their own while balancing their coursework and other responsibilities. Through this method, undergraduate students flourish into competent problem solvers with an edge over their peers who lack the experiences that undergraduate research at the department provides.

II. Engaging Students in Undergraduate Research

The experiences and expectations of students involved with undergraduate research go beyond that which is offered in the classroom, encouraging them to tackle difficult real-world problems under the guidance and supervision of more experienced faculty. Through undergraduate research, students are given the opportunity not only to apply what they have learned through their courses beyond the scope of ordinary coursework but to also expand their skill sets and learn to solve problems without the more structured approach of the classroom. Additionally, by designing research projects around relevant and interesting problems, student engagement,

enthusiasm, and productivity reach new heights. Undergraduate research places students in an environment where they can collaborate and experiment with one another on projects they are passionate about. While standard coursework is necessary to train students to be problem solvers, students do not always engage with course assignments in a way that will allow them to learn and grow beyond the parameters of the coursework. In contrast, undergraduate research opportunities draw students into projects they find engaging. The curriculum at Mercer University's Department of Computer Science has shifted in recent years in response to the observed interest and engagement engendered through introducing real-world problems and projects into the classroom. As Tucker notes, computer science "education must evolve in both method and content as our discipline progresses" [1]. The evolution of Mercer University's approach to computer science education is a recognition of this discipline-wide need to innovate teaching methods as the field advances in scope and depth. In offering and allowing for the continuation of these projects outside the classroom, Mercer University has made strides in improving its computer science curriculum as part of an effort to encourage participation in undergraduate research and thereby equip students with the skills and experience needed to succeed in an evolving field.

In recent years, professors in the Computer Science Department have begun to adapt segments of their research projects into their courses, exposing students early in their careers to the opportunities available to them should they wish to pursue more rigorous and rewarding work. In the classroom, these projects serve to educate students on concepts relevant to the course while simultaneously showing them how what they are learning can be applied to solve real and meaningful problems. Martin Zhao, one of the largest proponents of this approach to teaching students in the department, has designed several of his courses, including Programming II, Database Systems, and Software Engineering, around this method of engagement. Among the projects he has made available for students to work on both in and out of the classroom are the CSViewer, Chinese Learning Application, and Super Bowl Simulation projects.

A. *CSViewer*

The project referred to as "CSViewer" is a National Science Foundation funded software system designed to help researchers studying the Cayo Santiago rhesus monkey population have an integrated tool for visualizing their data. The software presents information about the population in expandable matrilineal and patrilineal trees and provides other interfaces for organizing and plotting data into charts and tables. Working on the multi-year CSViewer project allows students with a myriad of backgrounds to contribute to an intricate project and apply their skills, advancing the goals of the project and laying the foundation for the next group of students to build off their work. The CSViewer project engages students with a development process that more closely mimics the development of a product than any mock course project can provide, allowing students to communicate and collaborate with an actual client to produce software solutions with real and immediate applications. Students engaged with this project work closely with faculty and researchers to design a tool that will help researchers in their daily work. They receive feedback from them on initial designs and prototypes, allowing students to learn from and adjust for client feedback in real-time. This project is an invaluable opportunity for students to apply what they are learning in the classroom and expand their skill sets in the process. Fig. 1 showcases development progress on the graphical user interface portion of the software in version 1.0.2.

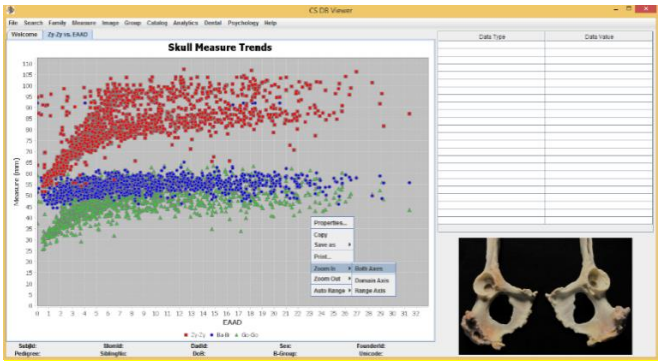

Fig. 1. A Screenshot of CSViewer v1.0.2

B. *Chinese Learning Application*

The Chinese learning application is another multi-year project that students have the opportunity to work on both during specific courses (such as Software Engineering) and outside of the scope of their regular coursework. The application's focus is to provide an interface for individuals to learn how to read, write, and understand Chinese symbols. Students working on this project have their efforts spread out over several parallel versions, including those designed to be desktop applications and web interfaces. This application encourages cross-curriculum learning, as those involved with its design and implementation not only engage with the software components of the project but also learn about Chinese culture and language. This project has the advantage of appealing to students who have interests in the humanities and other areas beyond the scope of computer science, allowing them to explore their interests while working on a meaningful application that can in turn be used to teach others.

C. *Super Bowl Simulation*

The Super Bowl simulation project is the newest addition to the extracurricular projects available to students in the computer science department. This project has students gathering, analyzing, and using data about a popular sporting event to create a visualization of that data that can be easily understood by the average person. Students engaged with this project use available data to "simulate" Super Bowl games on a virtual field, presenting the game play-by-play along with statistics and other information a user might find interesting or enlightening about the event. Like the Chinese learning application, this project draws students with interests outside of pure computer science and encourages them to contribute to a project that may be more relevant to them and their interests than standard coursework. Fig. 2 exhibits the in progress GUI component of this project.

III. Collaborative Learning

Undergraduate students engaged with extra circular research opportunities through the Mercer Computer Science

Department are afforded the advantage of being able to work closely with their peers and faculty. Students are encouraged to apply the skills learned through their coursework and to learn new skills not only from the challenges presented by these research opportunities but also from each other. The projects utilized by the department are designed to foster an environment of cooperation and collaboration in which students can share their ideas and make discoveries with one another. Regular and frequent meetings between faculty and research assistants help to facilitate the research and development processes while allowing for a more organic workflow as compared to the more rigid structure of the classroom. This approach follows and supports Ben-Ari's assertion that computer science "learning must be active: the student must construct knowledge assisted by guidance from the teacher and feedback from other students" [4]. Students involved with research in the department have their efforts attuned to several key areas, with many students operating in more than one area at a time.

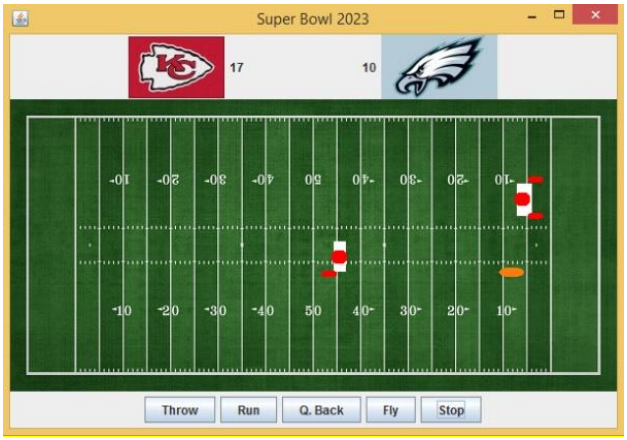

Fig. 2. Super Bowl Scoreboard under development

A. *Software Engineering*

The research opportunities provided by the computer science department all have an emphasis on the design and development of software systems that will be utilized to solve real-world problems. Undergraduates who choose to contribute to any of these projects are exposed to software engineering and design principles beyond the scope of what is taught in the classroom. Due to the nature of these projects as multi-year endeavors, students are encouraged to learn about and implement agile software development strategies where the progress made on a daily and weekly basis is tailored to the evolving needs and scope of the projects. Many of the research projects offered by the department are not intended to be completed by a single group of students moving through the computer science program but rather are designed for each group of students to learn from and build off of the work of their predecessors.

As part of the onboarding process, new undergraduate research assistants are exposed to the practices of the students who have worked on the project previously, allowing them to take advantage of what has already been done, adding their own contributions and insights to the development process over the course of their time working on the department's various ongoing projects. Typically, students will evaluate the needs and goals of the project on a weekly basis, adjusting the nature and scope of their contributions in a way that is organic and responsive. This practice gives students the ability to work on areas of each project that interests them or that they may have expertise in. Students coordinate with one another to ensure that while the overall objectives of the project are being worked towards, each individual gets to work on the areas that most interest them. Encouraging this sort of collaboration facilitates student learning in that students can share what they are passionate about with one another. It has been observed by the department that students learn best when allowed to pursue their interests and share their findings with their peers with minimal interference from faculty. Faculty members serve in the capacity of ensuring the projects stay on course but allow their students to be the driving force behind the progress made. This approach more accurately reflects real-world scenarios involving development teams and project managers than the traditional student-instructor relationship found in the classroom. Under this model, students get to experience actual software development practices and are exposed to an environment that translates more readily into real-world development than that found in a classroom setting.

Students engaged with the software side of the department's research projects do so primarily through the GitLab platform, a DevSecOps tool that utilizes Git to facilitate collaboration. While the platform is also used in many of the computer science courses taught by the department, students are able to become more intimately familiar with the tool through undergraduate research opportunities. Tools such as GitLab not only help students communicate and collaborate efficiently but also prepare them for their careers where they may use similar tools as part of a development team. Students are encouraged to manage project repositories themselves, with faculty oversight, so that they are exposed to the project management side of software engineering. Students are expected to maintain repositories in a professional manner as they learn how to work with one another through industry-standard tools such as GitLab.

The undergraduate students working on the software components of the department's projects do not do so in isolation, as collaboration with students engaged with other segments of the projects is encouraged. Students communicate with and support one another throughout the life cycle of a project to ensure that the final product meets expectations. For instance, the students involved with the software engineering side of the CSViewer project work closely with the students handling the data analytics of the project. Each team communicates its needs and expectations to the other, learning from each other's practices in doing so. After the software team implements a new feature, the analytics team ensures that the integrity of the data has been maintained and that any visualizations of the data are accurate and meaningful.

Furthermore, the user experience team ensures that the tools designed by the software team are easy to use and visually pleasing. The user experience team takes the raw functionality produced by the software team and drafts designs for how the final product should look and feel for the user to ensure a smooth experience. The software team then takes the feedback from their peers and redesigns the software as needed. This loop of design, implementation, feedback, and redesign comprises the short development cycles of each project, allowing them to develop both swiftly and organically. This practice encourages regular collaboration between teams, allotting students the freedom to learn in a collaborative manner without the constraints generally imposed by standard coursework.

B. Data Analytics

The data analytics team is responsible for gathering, sanitizing, analyzing, and evaluating the data related to each project before communicating their findings to the other teams. Of the three previously mentioned ongoing projects, the CSViewer project has been the most challenging and engaging project for the team to contribute to. Undergraduates working as part of the data analytics team on the CSViewer project have combed over a variety of data gathered by researchers on over 11,000 distinct animals located at the Cayo Santiago Rhesus Colony. Students involved with this project analyze the data in hopes of discovering patterns in the reproduction and birth rates of the specimens, as well as search for other existing patterns and relationships among the data that could interest researchers. By linking the collected data into a meaningful representation of the colony, the data analytics team is able to communicate to the software and user experience teams how best to visually represent the data. Through this project and others, students are learning how to work with large data sets and derive meaningful conclusions from them with the oversight and direction of faculty. This experience goes far beyond what students are exposed to in the classroom as these datasets are comprised of real-world data which students are expected to parse through and make sense of in the context of the overall project and endeavor.

Undergraduate students are also involved with data analysis for the Chinese learning app and Super Bowl simulation where they are expected to organize available information into a format digestible by the average person. Students working on the Chinese learning app maintain a database of information on Chinese characters such as meaning and stroke order, which is used to present information organized into lessons in the desktop and web interfaces. Likewise, students involved with the Super Bowl simulation take publicly available data about the event to create an easily followable simulation of the game (complete with statistics and information relevant to each game) that is designed to help users understand how each game played out in a visual manner. The data analytics teams involved with these projects are challenged to find the best way to present the information they have at their disposal. These teams work closely with the user experience team to ensure that the data is presented in a smooth and digestible way.

C. User Experience

The user experience team is responsible for tying together the work of the other teams in such a way that the results and conclusions of each project are presentable, clean, and accurate. For the software side of each project, this means ensuring that the graphical user interfaces are easily navigable and that the functionality of the software is clearly communicated to the user through these interfaces. The user experience team ensures that the raw functionality provided by the software engineering team is tailored to make the client's experience as smooth and efficient as possible. The user experience team works closely with the software engineering team to provide feedback and assist with redesign. Additionally, the user experience team is responsible for designing layouts and graphics for the software, balancing the functional needs of the client with an aesthetically pleasing design.

Furthermore, the user experience team provides testing and quality assurance for each version of the software. The user experience team is expected to ensure that the software is robust and to take note of any bugs found. Suggestions are then made based on these findings for a redesign in the next iteration of the project. Additionally, the user experience team is responsible for guaranteeing that the software is easily accessible and usable on a variety of platforms, including Mac and Linux. The team has collaborated in the past with the software engineering team to produce command-line installers for these systems and to write user manuals for the software. Overall, the user experience team is an integral part of each research project that polishes and perfects the work done by the other teams.

IV. Balancing Research and Coursework

A. Faculty Perspective

The department offers BS and BA degrees in its major academic programs (CSC, IST, CPS, and CYS) and a minor in Data Science (DSC). It also provides courses for majors from other academic programs at Mercer University, including Engineering, Business, as well as CLAS.

Mercer enjoys its reputation for being one of the leading universities in the U.S. for best undergraduate teaching [10]. Teaching accounts for 60% of the faculty's workload. Conducting quality research; seeking external funding; engaging undergraduate students in research; these meaningful goals all compete for time and effort with normal teaching loads and service commitments.

One way that has been carried out by MQZ is to incorporate research goals into the courses assigned. These courses are applied in nature, including intermediate programming, database systems, software engineering, and data science topics. The research and development tasks can be categorized as

- Data quality analysis and preprocessing;
- Data modeling and database design;
- Graphical user interface design and development;

- System architectural design and programming;
- System testing and user experience/training; and
- Data analytics for bioinformatics.

When breaking down the tasks and arranging the activities along the courses assigned, it becomes easy to prepare course materials and assignments in relevant courses. Therefore, research and teaching are not competing parties for faculty time and effort. Time management becomes easy and the two aspects can be beneficial to each other.

In the Computer Science Department, we believe that software and database development topics can be best learned by applying design principles and programming techniques in practice; and ideally, in real-world projects. The case studies and project topics used in MQZ's classes evolved from workshops [5] extended from popular textbooks [3], to externally funded research and development projects [7][8][9].

Starting this semester, specific plans have been made to train students in intermediate programming with skills that can help with research tasks, as well as to hire qualified students as research assistants in the CS rhesus monkey database project funded by a NSF grant [6].

B. Teaching Pedagogies

To better balance teaching and research efforts, we have found using a research theme to help organize teaching materials and assignments is the best approach.

Recent examples of teaching/learning activities were given in courses involved the NSF database project include:

1. Database implementation and essential queries have been assigned to student teams in several database and software engineering classes since the NSF grant was awarded in 2019.
2. The recent CSViewer release was refactored and extended from the codebase finished by a team in the latest Software Engineering I (CSC 480 of fall 2022). The student project was based on CSV files listing animal entries sorted by father/mother in chronicle order that were prepared by MQZ and a layered software architectural template introduced in class.
3. Analytical result visualization (using the JFreeChart API) and related selection features were then added to form version 1.0. As a member of the CSC 480 team, EW continues to contribute to source control and redesign for v1.1 while taking research and internship courses.

Providing DB and GUI design and codebase to student teams with specific tasks seems to be more effective. Incremental development process and source and document control can help establish a foundation for securing continued feature building across many classes during subsequent academic terms. Students can contribute through class assignments, volunteer and paid commitments in multiple (software engineering, database systems, data science, research and internship) courses.

C. Student Experiences

Students engaged in undergraduate research do so in addition to their normal coursework, and as such have the added challenge of balancing their schedules and managing their time efficiently around their contributions to ongoing projects. Students have found it easier to manage this task due to the nature of each of these research projects as team-oriented endeavors. Students work closely with one another to divide up tasks on a weekly basis, allowing students with a particularly busy week due to coursework and other obligations the ability to pass some of their responsibilities off to their teammates in the short term. In this way, each team helps share the additional strain undergraduate research can put on students, making it not only a manageable experience but an enjoyable one. Furthermore, students are encouraged to communicate with their professors about their schedules and obligations so that an appropriate research schedule and pace can be maintained. Department faculty involved with undergraduate research understand the needs and abilities of their students, working with them to find ways for them to contribute without overwhelming them. Overall, dedicated students have found undergraduate research to be a fantastic opportunity for learning even if what is expected of them goes above and beyond the expectations of the classroom. Support from faculty and team members makes undergraduate research in the department an accessible opportunity for interested students despite the additional challenges.

V. Results of Student Collaboration

The practice of using real-world problems in teaching has been adopted in the Computer Science Department at Mercer University for a long time and has recently been enhanced, especially with R&D needs driven by the CS rhesus monkey database project supported through a NSF grant.

Student teams in several courses offered at the Department in the past few years have been involved in various aspects of building these real-world applications in an incremental and iterative approach, utilizing a "stepwise" pedagogical method that Nager argues is one of the strongest aspects of effective computer science education [2]. Table 1 shows the course offerings and numbers of students involved in the CSViewer project, while Tables 2 and 3 display the same information about the Chinese learning application and Super Bowl simulation project respectively.

Both students and faculty alike have found the research projects offered by the department to be beneficial and an overall enhancement of the department's curriculum. Not only have these projects allowed students to experience an additional facet of the learning process and faculty to innovate their teaching strategies but they have also been met with an abundance of satisfaction and enjoyability by all parties involved.

TABLE I. CSVIEWER PROJECT INVOLVEMENT

Courses	Classes	Students
IST 220 - Intro to Databases	1	4
CSC 312 - Database Systems	1	3
CSC 480 - Software Engineering	4	20
CSC 485 - Data Sci Special Topics	1	1
Total	**7**	**28**

TABLE II. CHINESE LEARNING APPLICATION INVOLVEMENT

Courses	Classes	Students
CSC 480 - Software Engineering	4	22
CSC 485 - Data Sci Special Topics	1	1
Total	**5**	**23**

TABLE III. SUPER BOWL SIMULATION INVOLVEMENT

Courses	Classes	Students
CSC 205 - Programming II	1	12
CSC 312 - Database Systems	1	9
Total	**2**	**21**

Student satisfaction can be categorized into the following areas:

- Understanding why they are learning certain concepts or principles and how they are used in solving real-world problems.
- Seeing how a "real" working system is structured and being a part of the development team to add new features to it.
- Getting familiar with industry-standard tools and practices ahead of entering the workforce

From the faculty perspective, the following aspects are most advantageous and enjoyable for incorporating research-related materials in teaching:

- Balancing research and teaching loads in a way that one is supporting the other.
- Showing the latest and best knowledge to students based on new results from research at a level the students can understand and participate in.
- Gradually building the projects and forming a team that can help with progressing the projects.

At present, the efforts of the students involved with these projects have shown promising results. The CSViewer project in particular has moved from a conceptual phase to an experimental one. Versions of the software have been made available to researchers involved with the data collection and analysis process for feedback. Early feedback from these individuals has been positive, indicating that they believe that a full version of this software will facilitate their daily work with the data. Similar results are hoped for with the Chinese learning application and the Super Bowl simulation as they continue to be developed; the former to provide individuals with an interest in learning Chinese with an intuitive tool to do so, and the latter to help the public understand and breakdown a complex sporting event. Each of these projects hosted by the department aims not only to be an in-house learning tool for students via the design, development, and collaboration processes, but to also serve as learning tools for individuals in the general public interested in each project's subject matter via the final products.

VI. FUTURE WORK

Software systems with real-world applications have been built by involving students in relevant courses offered at the Computer Science Department. Efforts will be made to host these applications in the department to make them accessible to researchers and the general public. Regarding the CSViewer project in particular, infrastructure will need to be implemented in the department to make the data available to interested parties. The maintenance and daily operations of the servers used to host the data will be handled by undergraduates interested in the project, allowing them to gain practical experience with database and server management while performing a service for the department and the public.

The department intends to continue to incorporate students to build on top of existing codebases and train skillful developers through using R&D-related content in classroom teaching and assignments.

Additionally, efforts will be made to seek external funding to support developing and hosting these applications. New funding sources can be used to hire students to take up system administration and maintenance tasks.

ACKNOWLEDGMENT

The CSViewer app is developed as a part of a project supported by a NSF grant to M.Q.Z. (#1926402). The datasets used in CSViewer are based on the CPRC Skeletal Collection, supported by the National Institutes of Health NIH contracts NIH 5P40OD012217; and provided by Qian Wang and George Francis through a NSF grant (#1926601).

Robert Allen, Jesse Sowell, and the students at Mercer University's Computer Science Department provided their support and contributions.

REFERENCES

[1] A. B. Tucker. "Strategic directions in computer science education." *ACM Computing Surveys (CSUR)* vol. 28 no. 4, pp. 836-845, 1996.

[2] A. Nagers and R. D. Atkinson. "The case for improving US computer science education." *Available at SSRN 3066335,* 2016.

[3] C. Horstmann, Object-Oriented Design and Patterns, 2nd Edition, Wiley, 2005.

[4] M. Ben-Ari. "Constructivism in computer science education." *Journal of computers in Mathematics and Science Teaching,* vol. 20 no. 1, pp. 45-73, 2001.

[5] M. Q. Zhao and L. White, "Engaging Software Engineering Students Using a Series of OOAD Workshops," Proceeding of ASEE, Chicago, IL, June 18-21 2006.

[6] M. Q. Zhao, E. R. Widener, G. Francis, and Q. Wang, "Building a Knowledge Model of Cayo Santiago Rhesus Macaques: Engaging Undergraduate Students in Developing Graphical User Interfaces for a NSF Funded Research Project," ICR '23, Madrid, Spain, 2023.

[7] M. Q. Zhao, "Knowledge Models for SA Applications and User Interface Development for the SITA System," Final Report, ARFL/RI, Rome, NY, July 15, 2011

[8] M. Q. Zhao, M. Maldonado, T. B. Kensler, L. A. P. Kohn, D. Guatelli-Steinburg, and Q. Wang. "Conceptual Design and Prototyping for a Primate Health History Model," In: Arabnia HR, Deligiannidis L, Tinetti FG, Tran Q-N (Editors). Advances in Computer Vision and Computational Biology, New York, Springer, p.511-522, 2021.

[9] M. Q. Zhao, R. Allen, "Training Problem-Solvers by Using Real World Problems as Case Studies," Poster Paper, ICR '23, Madrid, Spain, 2023.

[10] US News college ranking: Best Undergraduate Teaching 2022, https://www.usnews.com/best-colleges/rankings/national-universities/undergraduate-teaching. Visited April 10, 2023