ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE
Paper ID #37526

Work In Progress: Reversi: A Platform for Teaching Programming Languages

Siwei He (Student)

Siwei is a fourth-year undergraduate student at University of Toronto. He works on research projects in file system, debugger, and machine learning application. Before that, he had experience in projects related to FPGA and worked as an FPGA software engineer intern at Intel. Siwei is an active contributor in the GNU GDB project. Github: @Louis-He

Hamid S Timorabadi (Assistant Professor, Teaching Stream)

Hamid Timorabadi received his BSc, MASc, and PhD degrees in Electrical Engineering from the University of Toronto. He has worked as a project, design, and test engineer as well as a consultant to industry. His research interests include the application of digital signal processing in energy systems and computer networks. He also has deep interest in engineering education and the use of technology to advance the learning experience of undergraduate students.

Afshin Poraria

Work In Progress: Reversi: A Platform for Teaching Programming Languages

Abstract

This paper introduces a new set of infrastructures and online interactive tools that can be employed to motivate students to learn programming languages. The tools were used to experiment in one of the introductory first-year engineering courses. The final project of the course requires implementing an AI program for a game called “Reversi". Reversi is a medium to hard level programming project that has been used in the course for several years requiring an immediate restructuring. Furthermore, due to COVID-19 and the restriction of in-person teaching, it has been a challenge for educators to excite, support, and encourage students. The new infrastructure provided an interactive platform for the students to familiarize themselves with the Reversi game project. It also provided a leaderboard, an interactive scoreboard, allowing students to compete with their classmates. The tools can instantaneously synchronize to students’ code submission to help students check their latest ranking among their classmates in real-time. This increased students’ level of engagement and learning. In addition, it allowed students to collaborate with their fellow classmates and discuss their algorithms. The tools and platform developed can also be employed in other courses as well other programming games. The result from students’ surveys and the active trend of the class online discussion forum indicates that the new online interactive system created a positive atmosphere and increased students’ motivation in learning programming languages.

Background

Programming language is a fundamental tool for engineering and computer science students to succeed both in their academic and career paths. Teaching introductory programming languages can be a challenging task especially if students are from many different engineering disciplines [1]. Only a few students find learning programming easy and indeed it is hard for instructors to teach fundamental programming languages [2]. Students at times may not become excited about programming languages [2]. From the teaching team’s perspective, instructors focus more on teaching programming language instead of teaching the application of the programming [3]. Students may not find programming courses intuitive, related to real life, or hard to engage with the course material.

During the pandemic period, the traditional face-to-face communication between students was disrupted. Students were in a more isolated studying environment than ever. A recent study shows that the impact of remote learning on students’ mental health is significant [4]. Remote learning not only impacts the logical/technical problem-solving ability of students, but also the engagement level of students in the material [5]. In this paper, we introduce an online platform along with a custom-build online judge that contributes to teaching an introductory course in C programming. The new platform transpires students to design their own game AI that can compete with other students’ AI. The platform is currently accommodating a design project that

runs over two practical lab sessions, nevertheless, the infrastructure can be used for other labs or in other programming courses. This report describes the fundamental structure of the platform by using a design project called Reversi. Reversi is a strategy board game invented over two centuries ago. The game involves two players, i.e., students who are asked to develop their own strategy by employing programming skills learned in the course as well as the use of AI strategy algorithms. The platform, then, allows students to evaluate their programming skills at various stages of game and code development and thus, become motivated to improve their designs by competing in a game environment.

Related Work

Learning programming can be arduous and at times dull for a novice programmer. Game-based concepts that can improve students' engagement in introductory programming courses are already discussed [6] [7] and many advantages and benefits are also reported [8]. The literature in [6] categorizes the existing game-based approach into three types, authoring-based approach, play-based approach, and visualization-based approach. All the analyzed games listed in [6] are either zero-player, one-player, or two players where the second player is an AI player machine. There are two multi-player games listed in [8], *Code Fights*, now renamed *Codesignal* (codesignal.com) and *Leek Wars* (leekwars.com). *Codesignal* has the capability of multiple players involved in a programming contest. Nonetheless, the game focuses more on the competition aspect instead of the game-promoting programming. It also does not have any gaming element and hence poses a limited effect on exciting or engaging students. *Leek Wars* is a multi-player game but the game itself is quite complicated for students in an introductory programming course. Students may become overwhelmed with all the rules and documentation of the game. This game is more suited for a more advanced level of computer science or engineering course. There appears no interactive game-based platform that encourages students in learning programming languages. And there is a need for designing hands-on and exciting practical labs for introductory programming courses. The following section provides technical details of a platform that assists students to learn programming languages faster and more effectively.

Technical Design

The design is a multi-user interactive online platform based on a custom-made online judging system. This system has three main components, a frontend, a backend server, and a judger server. Figure 1 shows the context diagram of this system. The backend server and the judger server are housed in one server. One subsystem from the frontend component is used to handle students' submissions and leaderboard generation. The other subsystem is used to host an interactive platform for students to play with the reference solutions provided by the teaching staff. Another subsystem manages the submission of students' solutions and then interconnects students' solutions to play against each other. Furthermore, the judger server needs to run in isolation for security reasons. As it is possible for malicious code to find its way into the system

via students' code submissions. It is crucial to implement the judger in a separate server independently. Hence, the decision was to install the judger code in an isolated part of the server. This ensures security and additionally allows the backend server and the judger server to be installed on the same server reducing the complexity of managing two servers as well as cost.

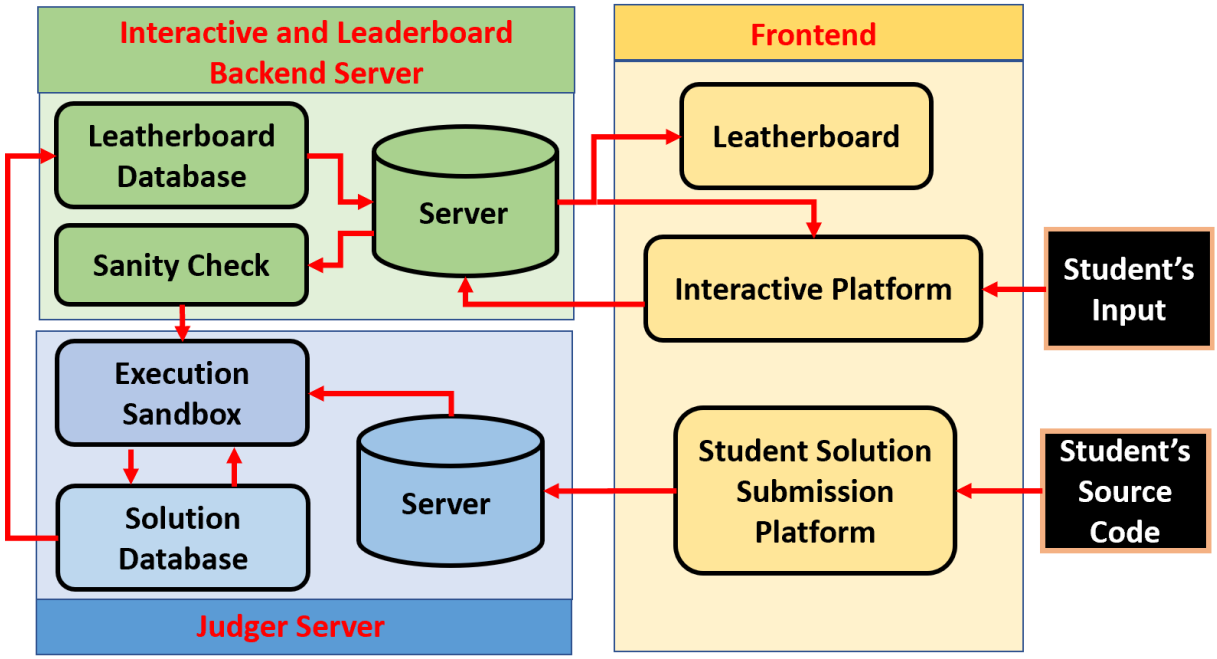

Figure 1: Context System Diagram for student submission and leaderboard subsystems.

As shown in Figure 1, the frontend is where students start submitting their code. The frontend then passes the code to the solution database of the backend and then to the execution sandbox in the judger server. The judger server runs the code and compares it against some public testing cases. The result of the tests sends feedback to students through the frontend. If the submitted students' code passes all the public test cases, then the solution will be added to the solution database. In the meantime, the judger runs the solution against other students' solutions and adds the result to the leaderboard.

The interactive session subsystem allows students to initially select another student, or player that they wish to compete against. This includes the reference solution and a sandbox scenario where students can play with themselves to familiarize with the game rules. After students selected the competing opponent, they can use the interactive session to play against the opponent as well as examine different algorithmic strategies. This is facilitated via a built-in function in the interactive platform which can show students all the legal next moves of the current run of the game. The intention of this design is to aid students to understand the rules of the Reversi intuitively and quickly. The backend receives its input from the frontend then performs a sanity check to filter out any invalid input from students and finally passes the request to the judger server. The sanity check is to lessen the processing pressure on the judger server.

After the judger server completes its execution, it then returns the result back to the backend and the backend passes the data to the frontend.

Figure 2 shows the front page of the Reversi platform. The front page allows students to access their personal pages and a sandbox. The Reversi platform automatically assigns each student a unique ID number. Only the assigned student knows if that particular ID belongs to him or her. This ensures anonymity and allows students to participate regardless of their programming background or feeling uncomfortable as a result of their rankings. The matrix board shows circles with black, white, and yellow colors. In the example shown in Figure 2, the student plays black and the opponent plays white. The hint can be turned on or off assisting student with the next move. In Figure 2 the hint is turned on and the suggested moves are shown in yellow.

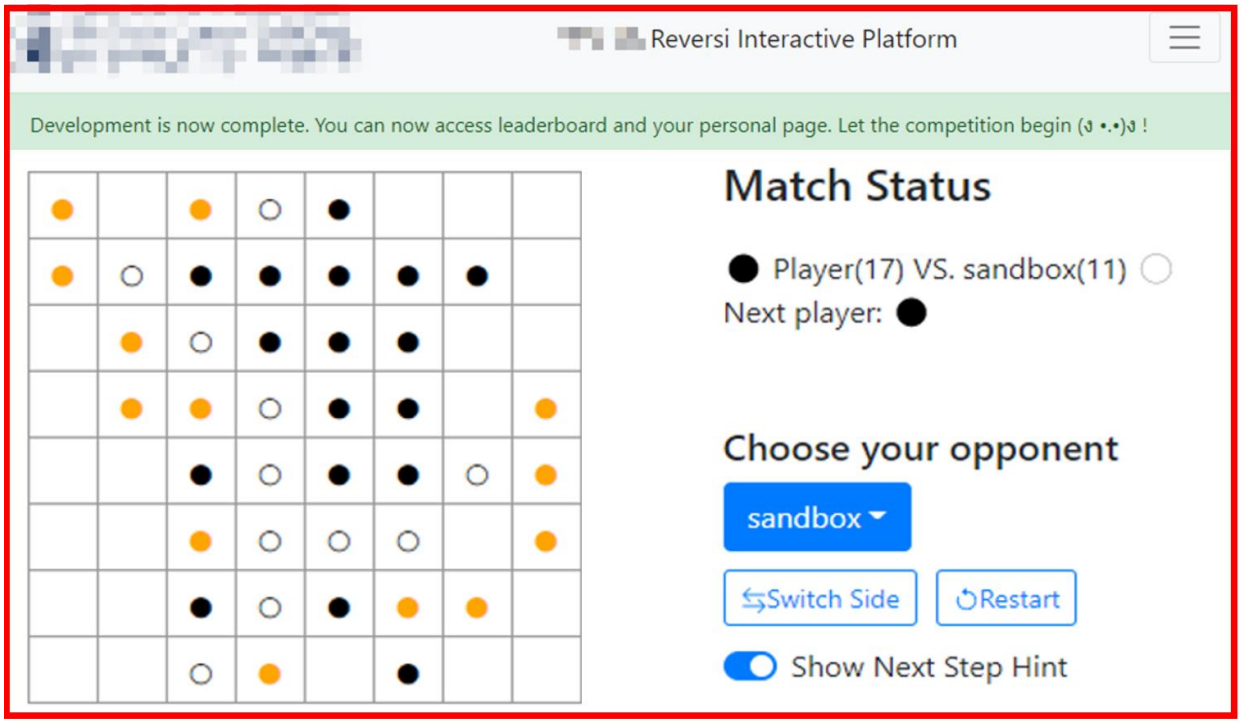

Figure 2: Reversi platform illustrating the frontpage in sandbox mode with next move hint turned on.

Figure 3 shows the personal page of the Reversi platform. The personal page provides historical ranking along with scores that student has achieved including student's current ranking, e.g., in Figure 3 the ranking is 15. Any time limit information and any illegal or invalid moves that might have occurred are also reported to the student on the personal page. This information is captured by the judger that runs a series of tests on the student's code. The ranking and scores information can be an incentive for the student to improve his/her solution, i.e., improve the algorithm or the solution performance. The time limit information or invalid move cases can also be employed to reconsider the strategy more thoroughly, consequently, allowing the student to improve his/her programming skills.

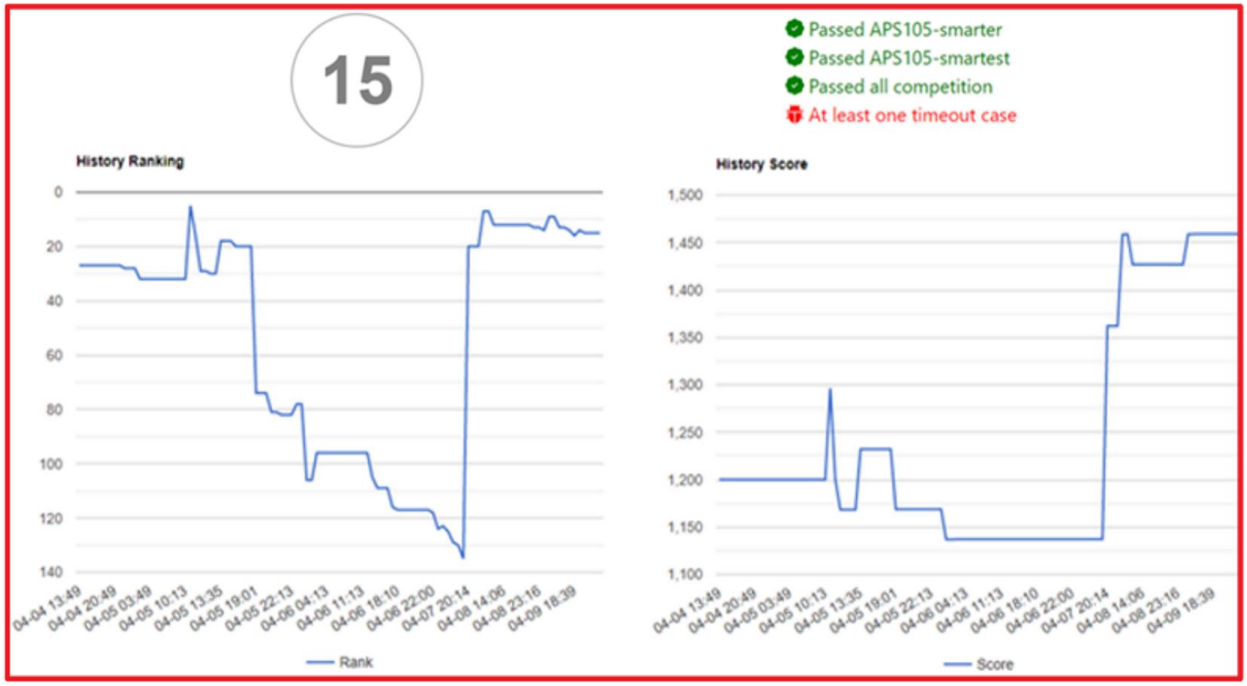

Figure 3: Reversi platform showing the personal page where student achieved rank of 15.

Functionality Testing and Verification

For the backend of the system, we verified our implementation by gathering all the past year's students' solutions and treating them as the input to the system. The number of testing samples is over 500 solutions. For the frontend of the system, we open-sourced our project and welcomed all the students who took the course to help improve and maintain the project. We also asked the students to play against reference solutions and solutions from fellow classmates. There was no major bug reported involving security or functionality during the semester.

Survey and Feedback Results

The analysis of whether the new system had a positive impact and excited students in learning programming languages examines two different components: a survey and statistics from the class online discussion forum. These components are discussed as follows.

Survey Result

The survey was conducted among a first-year programming course. Appendix B – survey provides a complete list of questions asked in the survey. Of the total participating students in the survey, 76% had no or very limited experience with programming when the term started. This, 76%, is the exact percentage of the students who found the complexity of the Reversi project challenging. For the general feedback on the interactive platform, 89% of the students visited the website several times during the Reversi project run whereas 27% of these 89% students were visiting the website more than 3 times a day. For the part where students can compete against reference solutions or create a sandbox game, 92% of the students thought that the interactive platform helped them understand the Reversi rule. In addition, 84% of the students gave positive

feedback on whether the interactive platform helped them to complete their Reversi project and come up with a better algorithm. Figure 4 presents the percentage distribution of students who found the new site helpful.

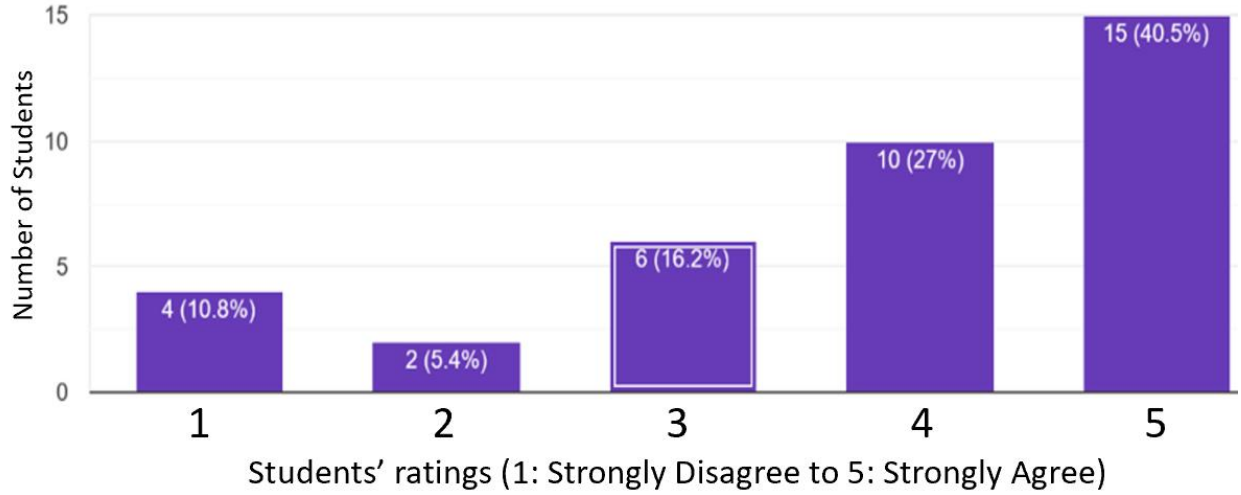

Figure 4: Survey result to the question of: **The site helped me in coding my project.**

For the part where students could compete against their fellow classmates' solutions, 78% of the students found that the ability to compete against their classmates helpful in improving their solutions. More importantly, 78% of the students have developed more interest in coding after doing the project using the new platform system. Figure 5 illustrates the impact of the project on students' interests in learning programming.

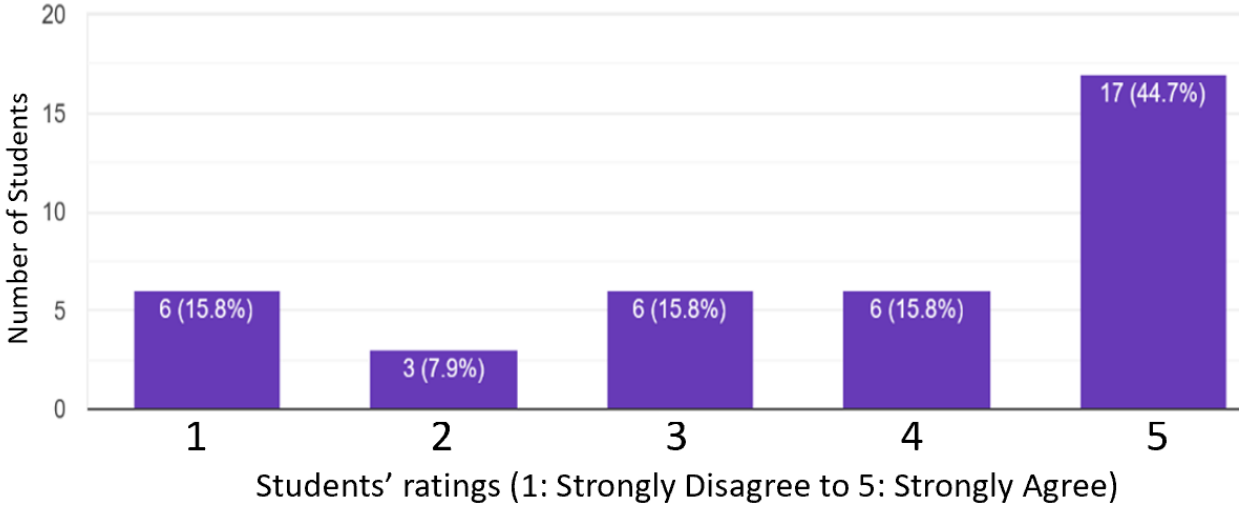

Figure 5: Survey result to the question of: **The Reversi Platform has increased my level of interest in coding.**

In the open question part where the survey asked about any downside and shortcomings about the site, most of the survey participants expressed their need to know more details about the time limit violation and illegal move case when the judger runs their solution against other students' solutions. The initial consideration on this part is to let students try to find the bug inside their

code based on the testing result. However, the survey shows that more details are needed to let the students learn more from their submissions.

Discussion Forum Statistics Result

The Reversi project took place between late March 2021 and mid-April 2021. The number of posts on the class online forum indicates a significantly increased activity during the time when the Reversi project was in progress. There were another two peaks in the chart, the one during early March is due to the online Q&A during the midterm test and the one during the end of May is due to the final exam Q&A. Other than these two peaks, the average number of posts per day throughout the term is 91.3 with a range of 1 to 289 posts. During the time when the Reversi project was in progress, the average number of posts per day increased to 198.17, ranging from 127 to 258 posts that if compared is approximately 2.17 times more active than the normal studying period. Figure 6 shows the number of posts on the course discussion forum from January to April 2021.

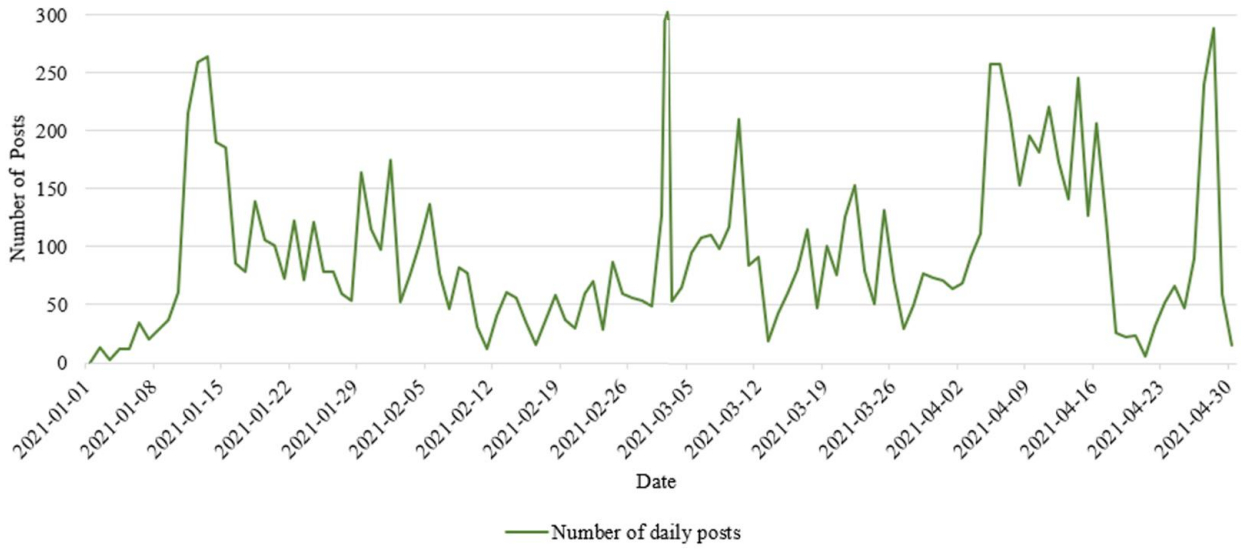

Figure 6: Daily Active Trend plotted based on number of posts on the online Discussion Forum.

Impacts

The new interactive platform created the possibility of interaction and communication among students in a new way during the online remote learning. The platform guides students to improve their solutions to the game-based lab in a visualized way. Students can check their improved version of their solutions on the platform in near real-time. The platform also allows students to use their skills to practice or compete with an AI machine or other students in a relaxed environment. This can increase students' engagement through active learning while improving their problem-solving skills via analysis of data available to them. For an introductory course in programming, it can also increase students' interest in programming. It should be noted that the use of the platform and the competition within the course was completely optional or had no marking weights assigned to it. Further and intentionally the Reversi platform was designed

to automatically assign each student with a unique ID number that only the assigned student could relate this ID to herself or himself. This ensured anonymity and provided an environment for students to participate regardless of their programming knowledge, skill, or any other background. The new platform makes it plausible to extend the system by supporting more games as part of course laboratories. Nevertheless, this platform is not limited to introductory courses. It can be extended to advance programming courses as well. For instance, an advanced machine learning-related course may find the platform useful in allowing students to improve their coding and machine learning skills by introducing a game/agent strategy laboratory.

Future Work

The feedback from the students' survey showed that most of the participants have positive feedback on the platform. However, there are some improvements that can be considered and executed in the current implementation of the system. Other than the improvement on the current Reversi project interactive platform, the current infrastructure can add more games, and projects to the platform. The necessary parts for such system improvements have already been implemented and are part of the frontend, backend, and judge server. The initial design idea of the system focuses on an introductory level of programming courses; however, game-based project design is also beneficial for advanced courses. Problems that involve more advanced algorithms like game tree, introductory AI problems, travelling salesman problems, NP-hard questions with different heuristics, etc. can be redesigned into a game-based project and added to the current infrastructure. One potential next step is to implement more games with low and high levels of difficulties on the platform. As a result, the platform with more difficult games could be extended to advanced courses while the less difficult games could attract novice programmers.

Conclusion

This study reports on the deployment of the interactive platform in an introductory programming course. The platform was employed to house a game called Reversi which was traditionally used in the course. The platform allowed students to compete with each other and become more motivated in improving their programming skills. The study is both observational and experimental. Both studies indicate that the platform alongside Reversi game assisted students in improving their programming skills.

References

[1] A. Robins, J. Rountree and N. Rountree, "Learning and Teaching Programming: A Review and Discussion," *Computer science education,* pp. 137-172, 01 06 2003.

[2] H. Gao, Z. Qiu, D. Wu and L. Gao, "Research and Reflection on Teaching of C Programming Language Design," in *Intelligent Computation in Big Data Era. ICYCSEE 2015*, Berlin, 2015.

[3] A. Gomes and A. J. Mendes, "Learning to program - difficulties and solutions," in *International Conference on Engineering Education – ICEE 2007*, Coimbra, Portugal, 2007.

[4] C. Baumann, H. Rousseau, C. Tarquinio, M. Batt, P. Tarquinio, R. Lebreuilly, C. Sorsana, K. Legrand, F. Guillemin and S. Bourion-Bédès, "Effect of the COVID-19 outbreak and lockdown on mental health among post-secondary students in the Grand Est region of France: results of the PIMS-CoV19 study," *Health and quality of life outcomes,* vol. 19 (1), pp. 265-265, 2021.

[5] S. Asgari, J. Trajkovic, M. Rahmani, W. Zhang, R. C. Lo and A. Sciortino, "An observational study of engineering online education during the COVID-19 pandemic," *PLoS ONE*, no. 16(4): e0250041, 2021.

[6] F. Li and C. Watson, "Game-based concept visualization for learning programming," in *Proceedings of the third international ACM workshop on multimedia technologies for distance learning*, 2011.

[7] S. Combéfis, G. Beresnevičius and V. Dagienė, "Learning Programming through Games and Contests: Overview, Characterisation and Discussion," *Olympiads in Informatics,* vol. 10, pp. 39-60, 2016.

[8] D. Dicheva, C. Dichev, G. Agre and G. Angelova, "Gamification in Education: A Systematic Mapping Study," *Educational technology & society,* vol. Vol.18 (3), pp. 75-88, 2015.

[9] S. Wasik, M. Antczak, J. Badura, A. Laskowski and T. Sternal, "A Survey on Online Judge Systems and Their Applications," *ACM Computing Surveys,* vol. 51, no. 1, p. 1–34, 2018.

Appendices

Appendix A - Leaderboard Site Screenshot

This appendix provides a screenshot of the leaderboard site in April 17, 2021. The Reversi platform automatically assigns each student with a unique ID number that only the assigned student knows that the particular ID belongs to him/her. This ensures that student's identity is kept confidential, consequently, no one in the course can relate a ranking to a certain student. Figure 7 shows the leaderboard where students' ID has been intentionally blurred. Figure 7 also illustrates provides the ranking of students and their scores where the score is calculated based on Elo rating. The indicator shows 100% for the highest scored achieved. Status field shows the history with respect to any violation such as Time Limit Exceeded (TLE) or Invalid Move (IM). If there is no violation in any of the game settings then PASS is shown.

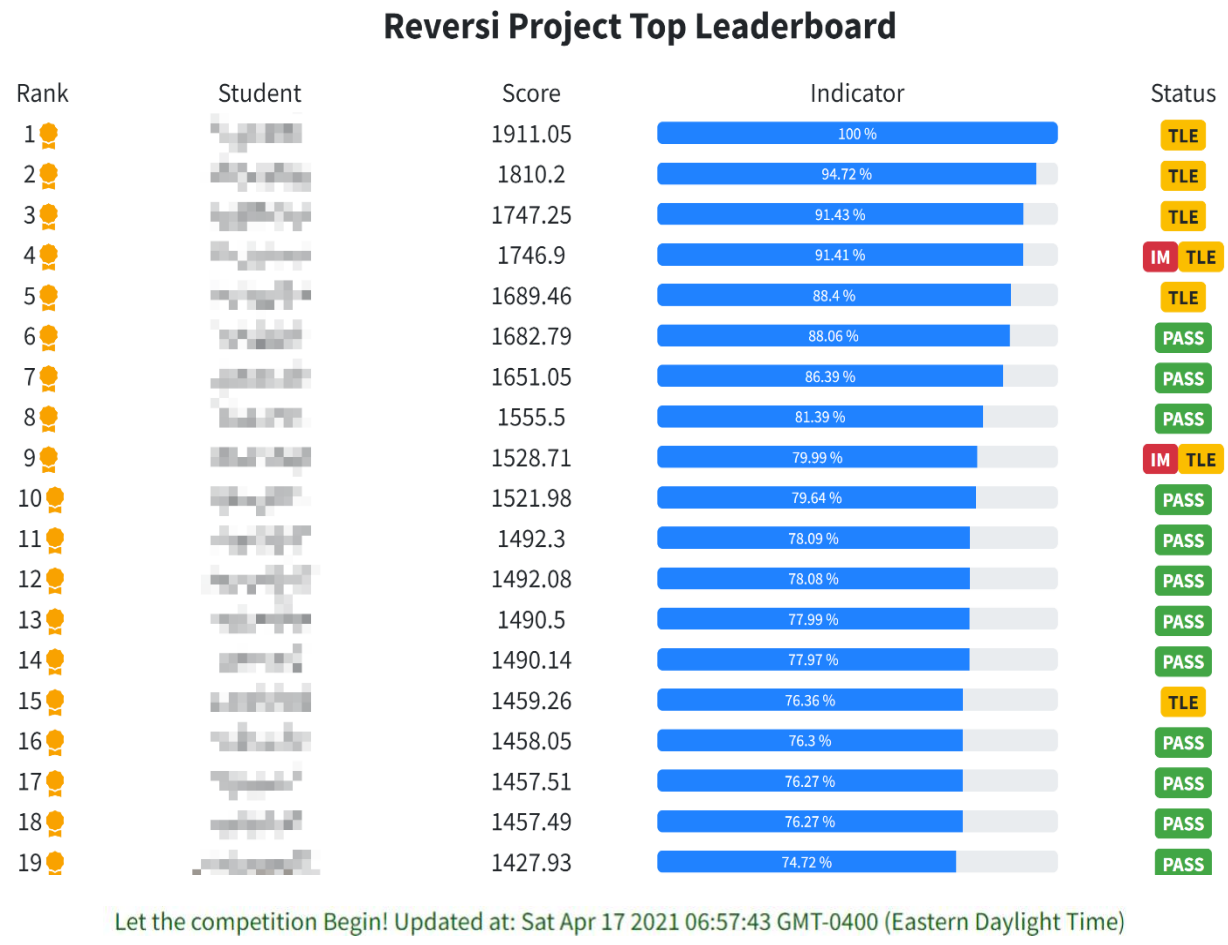

Figure 7: Screenshot of the online Leaderboard.

Appendix B – Survey

The following is the type of question that were asked students in the conducted study survey.

1. How useful is the site in helping you understand the project's tasks?
2. How useful is the site in helping you understand the rules of the game?
3. How much do you think the site has helped you in coding your project?
4. Is the site useful in improving the overall competitiveness of your code?
5. Is the site's Hints for next possible moves helpful in designing your code?
6. Is the site's capability to allow you to play against another competitor useful in helping you design better strategy?
7. Do you think the possibility provided on the site to find your ranking is helpful in motivating you to improve your code?
8. Are the site's evaluation results (PASS, IM, and TLE) useful to debug and improve the efficiency of your code?
9. How many times have you accessed the website to play the game?
10. How many times have you accessed the website to check your ranking?
11. How helpful do think it would be if the site is accessible from the start of the term?
12. How would you rate having a leaderboard on the site?
13. How would you rate having an all-time high-ranking list?
14. [Text Question] What don't you like about the site? (Explain)
15. [Text Question] What would you like to be added to the site? (Explain)
16. Do you think the leaderboard scoring scheme needs to be presented on the site?
17. How would you rate the complexity of the project?
18. Do you think providing solution library for the first stage of the project is helpful to you in designing a more competitive second stage code?
19. Is the lab handouts and seminars in line with the game on the site?
20. How would you rate the difficulty level of the project compared with the material covered in the course?
21. How fair is the marking allocation for the two stages of the project?
22. How would rate the impact of this project in increasing your interest in coding?

Course Related Questions:

23. How would you rate yourself in terms of coding capability when you started the term?
24. How do you rate the complexity of materials covered before the midterm?
25. How do you rate the complexity of materials covered after the midterm?
26. How would you rate yourself in terms of coding capability after the term?
27. Was the material presented in the course in line with the labs and assignments?