

CS Connections: Adapting a Popular Puzzle Game for Computing Concepts

Ethan Fong
University of Toronto
Toronto, Ontario, Canada
ethan.fong@mail.utoronto.ca

Michelle Craig
University of Toronto
Toronto, Ontario, Canada
mcraig@cs.toronto.edu

Jonathan Calver
University of Toronto
Toronto, Ontario, Canada
calver@cs.toronto.edu

Abstract

Students often find introductory computing courses challenging, possibly due to their difficulties with motivation and engagement. One established strategy instructors use to overcome these hurdles is to introduce games and puzzles. We developed CS Connections, a web-based application inspired by the New York Times Connections game. CS Connections presents students with a grid of elements to categorize based on shared characteristics. These elements can be text or code snippets, and their common links may involve runtime behavior, output, or conceptual relationships. By solving puzzles through correct grouping, students can review recently-learned material or actively explore deeper programming concepts.

We deployed the tool in two offerings of our university's CS1 course, and present our initial results on student retention, perceptions and ease of use. Our initial results suggest that CS Connections is favorably perceived and students find it useful, but student interest declines over the weeks as topics become more complex and the novelty of the tool wears off. Instructors found the interface intuitive, but had difficulty creating novel and relevant puzzles. However, by encouraging the exploration of programming concepts in an interactive format, the tool shows significant promise as part of the active learning paradigm.

Open-sourced for broader access, CS Connections is available online for educators in both CS and other subjects to integrate into their classrooms. This paper includes instructions for educators interested in using the tool.

CCS Concepts

• **Applied computing** → **Interactive learning environments**; • **Software and its engineering** → *Software creation and management*; • **Human-centered computing** → *Interaction design*.

Keywords

Game-based learning, Educational puzzles, Code classification, CS1, Student engagement, Web-based tools

ACM Reference Format:

Ethan Fong, Michelle Craig, and Jonathan Calver. 2025. CS Connections: Adapting a Popular Puzzle Game for Computing Concepts. In *The 27th Western Canadian Conference on Computing Education (WCCCE'25)*, April 28–29, 2025, Calgary, AB, Canada. 7 pages. <https://doi.org/10.60770/3jsj-m574>

1 Introduction

A major challenge that has persisted in introductory Computer Science (CS1) courses over many years is that of high dropout and failure rates [3]. While this can be attributed to many factors including teaching approaches or student learning preferences, one key reason is student motivation and engagement [8]. Researchers have demonstrated that incorporating games into the classroom enhances the learning experience by making abstract concepts more interactive and fun [8, 15]. Thus, in recent years, there has been growing interest in using games to boost student motivation in educational settings, including in Computer Science (CS).

With the intention to incorporate a popular puzzle game format in our CS1 course, we turned our attention to the New York Times (NYT) Connections puzzle [12]. Our goal was to explore whether this format could engage CS1 students and support their conceptual learning in a optional, supplemental activity that students could access without any grading or performance pressure. However, in order to create puzzles with this format, we found that we needed to create a web-based tool to do so, which we named “CS Connections”.

Evaluation of educational tools presents numerous challenges and requires a nuanced approach [19]. Due to the constraints in which we deployed the tool, we were unable to measure the effect of this tool on student learning outcomes and understanding. Instead, we focused this initial evaluation of our tool on three areas that serve as practical proxies for the value of educational tools:

- Student Retention
- Student Perceptions
- Ease of Use

While we used our tool with CS1 concepts, CS Connections can also be used in other CS courses or even other disciplines. This paper presents our reasoning for developing the tool, the initial evaluation of the tool being used in our classrooms, and instructions for future instructors looking to use the tool.

2 Background

2.1 Relevant Literature

Games can be an effective tool for learning computing concepts [13]. A 2016 study found that incorporating challenging games at the edge of students' skill level significantly boosts their motivation to continue learning and overcome challenges [7]. Puzzle games specifically, are a great way to build intrinsic motivation, as they provide a sense of satisfaction from the “Eureka moment” when the solution is discovered [5]. Kasmarik has shown that incorporating

puzzles, and by extension puzzle games, into a CS1 course increases students' interest and active participation [11].

One type of puzzle seen in CS is the code classification puzzle. Code classification puzzles are a type of challenge where students identify and organize code segments according to their function. These puzzles require learners to analyze, classify and match code snippets to corresponding categories or structures. In the BRACElet project [17], novice programmers were asked to classify and connect code segments to understand their relationships as part of a larger program. This significantly improved students' comprehension of the code and retention of programming concepts. Current studies on code classification tend to focus on traditional assessment formats [17] and lack investigation into game-based approaches. This context motivated us to develop a tool that enables instructors to transform code classification into an interactive, game-based activity.

To guide the evaluation of evaluation our tool, we draw on Self-Determination Theory (SDT), a psychological framework that emphasizes autonomy, competence, and relatedness as essential for motivation [14]. We choose this framework as it has been commonly applied to the design and evaluation of games in HCI research [18]. In educational contexts, tools that allow students to make their own choices (autonomy), build skills through manageable challenges (competence), and feel connected to others or to the learning environment (relatedness) are more likely to foster intrinsic motivation and sustained engagement [14]. Therefore, we selected three evaluation criteria that we could both measure and reflect these motivational principles:

- **Student Retention** can provide insight into whether the tool helps support autonomy by observing whether students choose to re-engage with it voluntarily. Although other factors may influence students' repeated use of the tool, the fact that some students chose to engage with it voluntarily over multiple weeks suggests they saw value in the activity. This kind of self-directed engagement aligns with conditions that support autonomous motivation, where students participate more because they find the experience personally meaningful or enjoyable.
- **Student Perceptions** can reveal whether the tool supports competence and relatedness by indicating if students found the puzzles valuable, appropriately challenging, and relevant to their learning experience. Perceptions of usefulness suggest support for competence, while mentions of collaboration or shared experience can reflect relatedness.
- **Ease of Use** supports both competence and autonomy by minimizing technical barriers that might otherwise lead to frustration or disengagement. When students can easily navigate and use the tool, they are more likely to feel capable of succeeding (competence) and in control of their learning experience (autonomy), both of which are essential for fostering intrinsic motivation according to SDT.

2.2 NYT Connections

The NYT Connections game [12] challenges players to organize a grid of 16 words or phrases into four groups of four based on shared themes or categories. Players must find the common link

between items in each group such as synonyms, colors, or pop culture references while avoiding red herrings designed to fit multiple categories. Once all the items in a group are correctly identified, the game reveals the category. Figure 1 presents a sample game. One category, called "Slimy Animals", has already been found and grouped. The puzzle is solved once all four categories are revealed, but only if the player does not make four mistakes while trying to group the categories.

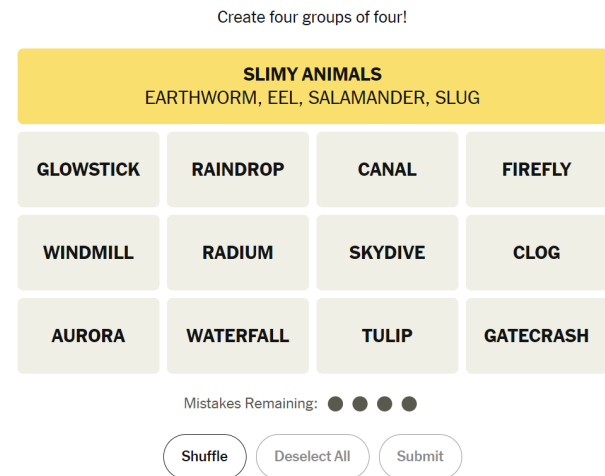


Figure 1: Sample Game of NYT Connections (link)

Despite Connections' widespread appeal as a recreational puzzle, its use in educational contexts is largely unexplored. A recent study adapted the format in the context of medical education [16], suggesting broader educational potential. To our knowledge, Connections-style puzzles have not been used to support CS1 concept learning. This represents a valuable opportunity to explore how this familiar, classification-based puzzle format might foster engagement and conceptual thinking in introductory programming courses. Using a familiar format like Connections could make learning feel less intimidating to students and help present or review content in a more accessible way.

3 Adapting NYT Connections for CSEd

Tools to create and share NYT-inspired Connections puzzles can be found on the internet [2, 4]. However, through our own usage, we found that the existing tools lacked key features needed to use the puzzles in a CS1 context. Critically, they did not support using code snippets as game elements. Table 1 summarizes the main features that we found were lacking and the associated rationale for needing these features.

Therefore, recognizing available Connections creation tools did not meet our needs, we developed our own web-based application that has built-in support for code snippets as game elements. This allowed us to implement the features that we could not find in other Connections tools and gave us control over the data that is collected from these games.

Required Feature	Reasoning
The ability to customize grid size (M categories of N)	Allows instructors to fit larger individual elements in a game and supports more flexible game creation
Support for lowercase letters	Coding languages like Python are case-sensitive, so capitalization is critical to represent code clearly
Support for special characters such as commas	These are essential for representing valid code syntax as elements
Use of characters such as newlines and tabs	These are essential for allowing longer code snippets as elements
Prerequisite information field	Helps present variable initializations or assumptions concisely
Data collection on guesses and categories	Provides insights into student performance and common challenges

Table 1: Feature Requirements and their Rationale

3.1 User Interface and Game Example

The interface of the game is based on an open source rendition of the NYT Connections game written in React.js and available on Github [1]. To tailor it specifically for CSEd, we made several enhancements to the original interface. Features such as syntax highlighting allow players to easily distinguish between code snippets, while responsive scaling ensures the game works seamlessly across devices. See Figure 2 for an example CS Connections game, with one category revealed.

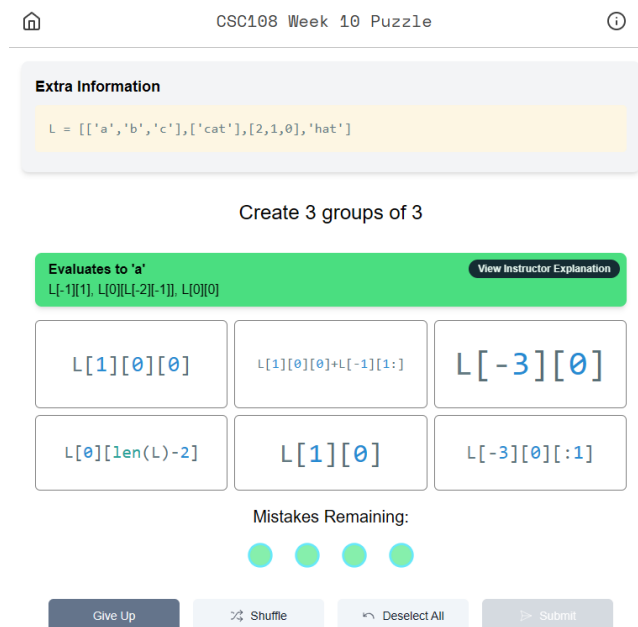


Figure 2: Example Game on List Indexing

Students can also create their own Connections games using a game creation form on the CS Connections webpage. All puzzles are organized on a centralized landing page, grouped by course for easy navigation. Instructor-generated puzzles are distinguished from student-generated puzzles, allowing students to quickly find instructor-generated puzzles while still being able to browse student puzzles.

3.2 Instructor Interface of the Tool

The instructor interface of CS Connections includes authentication and a dashboard for managing games in their courses. After logging into the platform, instructors can create new puzzles through the game creation form. The form includes a built-in preview feature, allowing instructors to visualize and test how the puzzle will appear and behave for students before publishing it. Each puzzle has a JSON representation which allows for the storing and sharing of puzzles. The JSON representation of any puzzle can be downloaded from the homepage of the app and used in game creation allowing instructors to reuse and edit past puzzles.

3.3 Data Collection and Statistics View

One advantage of using our CS Connections tool over other existing Connections tools is the data collection that is integrated within the app.

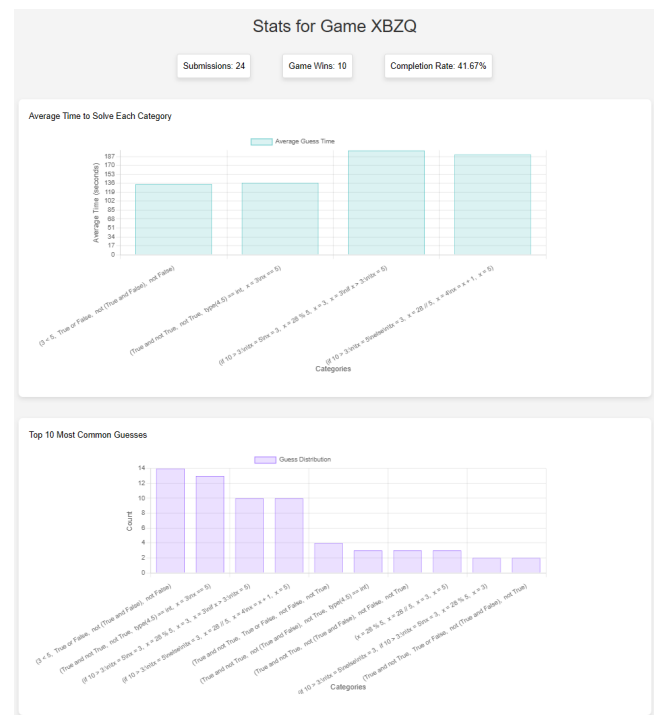


Figure 3: In-game Statistics View

Our CS Connections app collects gameplay data for each game played, capturing details about the cumulative time it took to make each guess, and the top 10 most common groupings made by the students as shown in Figure 3.

When the game is completed or the student presses the “Give Up” button, the solution to the puzzle is revealed, and the gameplay data is sent to the backend. The collected data includes guess distributions, time taken for each grouping, and whether the game was successfully completed.

Gameplay information is automatically visualized in a dedicated statistics view for each game. This enables instructors to identify trends in student performance, such as commonly missed categories or categories which students took longer to categorize. These insights help instructors address areas where students are struggling and create games at the right difficulty level.

4 Results

In this section, we present our results from the use of CS Connections in two offerings of our university’s CS1 course, which is taught in Python. In both the Fall 2024 and Winter 2025 semesters, we presented weekly puzzles in the 10 minutes before lecture. The CS1 course meets for three hours of lectures each week and follows an active learning approach that includes activities such as worksheets during lectures, making the inclusion of these puzzles a natural extension of the course’s interactive format. Some of the CS1 topics and design choices have also been documented [6].

4.1 Evaluation Criteria

Because adoption of this tool is still in its early stages, we did not aim to directly measure student learning outcomes, which would require more controlled conditions and long-term assessment. Instead, we selected three evaluation criteria based on our interpretation of SDT and used the following data sources to evaluate our tool:

- Student retention via gameplay statistics ($n = 674$)
- Student perceptions via end-of-term survey responses ($n = 22$)
- Ease of use via student feedback ($n = 22$) and instructor reflections ($n = 2$)

4.2 Student Retention

For the Fall 2024 offering of our CS1 course, there were 674 games played across 11 weekly puzzles. The data shows that the initial games were played much more than those from subsequent weeks. The first week the app was introduced was the most popular, with 220 games played that week. The number of plays is visually represented in Figure 4. Note that the graph does not include statistics from the first week as we were still using other online tools to create the puzzles thus we could not collect any gameplay data.

Student engagement initially starts out strong but trends downwards over time. This suggests that initial interest in the game may have been driven by its novelty, while sustained engagement likely required more incentives or integration into the course structure. This suggests that the tool in the way that we presented it may not have been conducive to autonomous motivation. However, it is important to note that a range of external factors (such as midterms, declining attendance, and exam preparation) may have also contributed to the declining participation. Future iterations of the game could incorporate direct incentives, such as having a similar grouping problem on a test or by using “marketing” techniques such as weekly email reminders designed to advertise the puzzles which

has been shown to improve engagement with other educational tools [10].

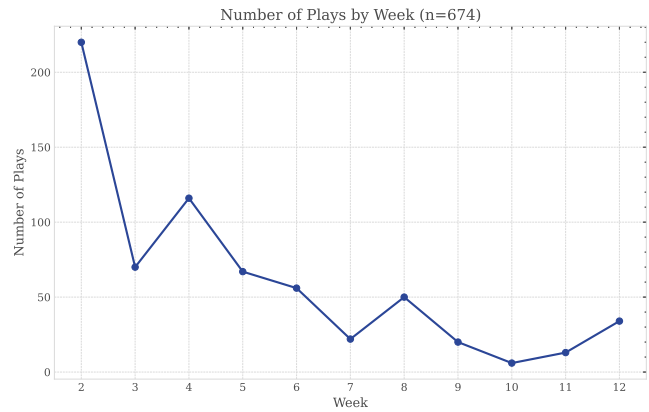


Figure 4: Game Submissions by Week

4.3 Student Perceptions

In order to assess student opinions on the game, we applied for and received ethics approval from the University Ethics Board (Protocol Reference #: 00047439) to conduct a survey on student perceptions of the game. This survey was distributed to the Fall 2024 CS1 cohort of over 400 students. There were a total of 22 responses and the survey was structured into three sections; the first aimed to gauge student profiles, the second measured perceptions of the game and the third collected open-ended feedback.

4.3.1 Student Profiles. To better understand the context of the survey responses, we first gathered information about the students who responded. See Table 2 for a summary of the results.

Survey Category	Option	Count (%)
Intend to study CS	Definitely	2 (9.1)
	Most likely	5 (22.7)
	Most likely not	7 (31.8)
	Definitely not	3 (13.6)
	Undecided	5 (22.7)
Gender	Man	12 (54.5)
	Woman	9 (40.9)
	Gender variant/non-conforming	1 (4.5)
Played NYT Version Before	Yes	16 (76.2)
	No	5 (23.8)
Individual or Group Play	Always individually	6 (28.6)
	Mostly individually	9 (42.9)
	Sometimes in a group	2 (9.5)
	Mostly as a group	3 (14.3)
	Always in a group	1 (4.8)
Play At the Beginning of Lectures	Frequently (6-10 times)	6 (28.6)
	Occasionally (3-5 times)	10 (47.6)
	Rarely (1-2 times)	5 (23.8)
Play Outside of Lecture	Very frequently (more than 10 times)	1 (4.8)
	Rarely (1-2 times)	5 (23.8)
	Never	15 (71.4)
Opened the Instructor Explanations	Every time	5 (23.8)
	Most of the time	6 (28.6)
	Half the time	3 (14.3)
	Less than half the time	4 (19.0)
	Never	3 (14.3)

Table 2: Survey Results for Student Profile

From the survey, we see that a large majority of students never or rarely play the games outside of lecture time, suggesting that the in-class engagement with the game may be sufficient for its intended purpose. Alternatively, this could also indicate that students view the game as primarily a classroom activity rather than an independent learning tool.

Another insight is that students tend to complete the game individually, consistent with findings from the use of a Connections game in a medical context [16]. Although our tool was designed to support relatedness by encouraging social interaction, this outcome suggests that students may naturally default to individual play. To better support relatedness and foster a sense of community as outlined in our SDT-based evaluation criteria, instructors might consider explicitly encouraging collaboration to promote more social engagement with the puzzles.

4.3.2 Likert Data. The second part of the survey used a Likert scale which provided students with options ranging from strongly disagree to strongly agree. The results are shown in Figure 5.

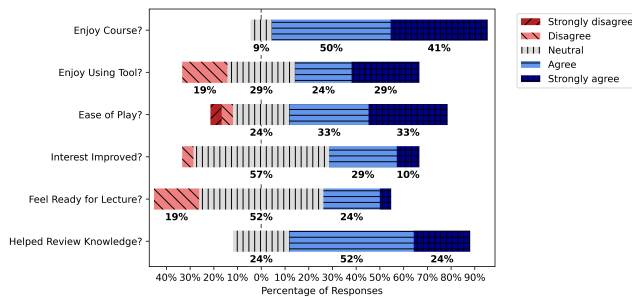


Figure 5: Likert Chart of Survey Responses

Students expressed a positive sentiment towards the use of the tool with 53% of students respectively choosing "Agree" or "Strongly agree" when asked if they enjoyed using the tool. Similarly, students largely agreed that the games helped to review knowledge with 76% of students choosing "Agree" or "Strongly agree" options. This suggests the tool supports students' sense of competence by providing meaningful practice that challenges students at an appropriate level. However, students were notably less positive when asked whether the tool helped them feel ready for lecture. This contrast may indicate that while the tool reinforced past content, this was not enough to make students feel comfortable with learning upcoming material.

It is also worth noting that students who reported enjoying the tool may already have a predisposition towards puzzle-solving, which could influence their positive perceptions. Overall, we see that the tool was able to generate positive perceptions related to students' sense of competence, but it did not appear to foster relatedness, which might have emerged more strongly if the games were played in a social context.

4.4 Student Ease of Use

In addition to questions about learning value and enjoyment, the survey also asked students to evaluate the tool's ease of use. As

shown in the corresponding row of Figure 5, the responses suggest that while many students found the interface accessible, there is still room for improvement. Specifically, 66% of students agreed or strongly agreed that the tool was easy to use, whereas 10% reported difficulty and 24% remained neutral.

4.4.1 Vignettes of Student Feedback. The survey also collected student feedback on the games. Table 3 summarizes some relevant feedback points and takeaways, which may serve as guidance for future directions of the app.

Table 3: Student Feedback and Possible Adjustment

Feedback	Adjustment
I didn't have time to play the game before each class as my phone wasn't able to read the QR codes and it took too long to access the games. I played the games at the end of the course while studying for the final exam and found them extremely educational.	Allow more time for the activity. In the 2024 Fall CS1 course, the QR code for the game was only visible for 10 minutes before lectures start.
I might have missed the goal of your project at the beginning of the course. It would be helpful for the professor to go over it with us during the first class.	Since this project was developed as the course was progressing, there could have been clearer communication on the goals of the project.
It would be helpful to provide a short description of the game objective for each week.	We could add an explicit game objective when presenting the puzzles.
(Instructor explanations are) mostly helpful but sometimes not detailed enough; could use more step-by-step breakdown of the solutions.	In addition to stating the category, the instructor explanation should explain the process to arrive at the correct grouping.

Overall, the results of the survey indicate that students enjoy using the tool but there is room for improvement in making the games understandable for all students. Ease of use was also a challenge seen in another use of Connections games in an educational context [16] and is a common challenge in adopting game-based learning [9]. Therefore, in future iterations of the game, it would be beneficial for instructors to explain the goals of the puzzle, and possibly provide an example of how the game works to help address students' concerns over ease of use.

4.5 Instructor Ease of Use

We gathered feedback from two instructors who used the CS Connections tool for two offerings our University's CS1 course. Here we present a summary of their experiences.

4.5.1 2024 Fall CS1. At first, the instructor found it challenging to create good puzzles. She was making the puzzles too hard by assuming students already deeply understood the concepts (rather than including easier categories that could be selected without deep

understanding). After a few weeks, she requested the instructor-explanation feature to address this issue. She reflects however, that by this time many students had stopped participating because of the initial challenging puzzles. Nevertheless, there was still a small group of students that continued to engage with the puzzles and had very positive attitudes towards the games.

On the topic of creating puzzles, this instructor was surprised at the variety of concepts she could explore with the puzzles. During the term she felt that it was important to keep consistent with the presentation of these puzzles to cater to the students that already have built a habit of playing them. Note that she only presented puzzles in a passive way on the pre-class lecture announcements. In post-term reflection she wondered if instead of being presented weekly, one or two puzzles could be used judiciously to replace a few of the in-class active-learning worksheets.

4.5.2 2025 Winter CS1. For the Winter semester, a different instructor was able to re-use puzzles from the previous offering of the course which allowed the instructor to focus on student engagement rather than content creation. The instructor reflected on the need to be mindful when creating puzzles, noting that it is easy to design puzzles that lack a "fun" factor. Maintaining variety across category types was identified as a key design principle. For example, it would be tempting to create categories that are all variations on "outputs [BLANK] in the terminal." Once students encounter a few similar categories, the puzzle becomes repetitive, detracting from the puzzle. Being intentional about the inclusion of red herrings (elements designed to fit into multiple categories) and incorporating overlapping categories also helped to create opportunities for eureka moments, which are critical for an effective puzzle experience.

Both instructors emphasized that the most significant challenge in using the platform was not the interface itself, but the effort required to create original, high-quality puzzles. Designing puzzles that are both effective and engaging takes time and creativity, especially without prior examples to draw from. To improve ease of use, we aim to continue expanding the shared repository of puzzles. As more instructors adopt the tool and contribute their puzzles, it will become easier to create new games by drawing inspiration from existing ones. This growing knowledge base can lower the barrier to entry and make the tool more approachable to other instructors.

5 Adopting CS Connections

Since CS Connections is a web-based tool, instructors do not need to download or run any additional software to use the tool. Instructors can choose to use our online version of the app or to utilize the source code and host the app themselves. While self-hosting requires additional setup, it offers full control over the data collected. The source code of the app is available at <https://github.com/ethan-fong/cs-connections>, and our online version can be accessed at <https://cs-connections.app>. Regardless of the hosting method, follow these steps to create the first game.

1. Create a Course on the Platform. Instructors interested in using the platform can request a course to be created on the platform using the provided contact information on the homepage of CS Connections.

2. Receive Credentials to Upload Games. After having a course created, they will also be provided with credentials to log into the instructor interface to upload and manage games.

3. Create a Puzzle and Upload it. Instructors can create a game using the game creation form and once the form is submitted, the app will generate a four letter game code and the game will be visible for students to play online.

4. Present Puzzles to Class. Present the link to the class to allow students to play the games. One way to share the link is by displaying a QR code linking to the puzzle so students can access the games from their mobile devices.

6 Future Directions

CS Connections has been designed as an open-source tool to ensure accessibility for educators and students for use in the classroom or for further research. Over time, CS Connections continues to be iteratively improved, based on needs expressed by instructors. The CS Connections tool is being tested in a CS2 course and a Data Structures and Algorithms course, which has expanded the tool to be used with Java and a language-independent theory course on advanced data structures.

The code classification format of the NYT Connections puzzle was selected because prior research has shown that these types of exercises improve beginners' understanding of code. However, while this pedagogical foundation informed the tool's design, further research is needed to assess whether the specific implementation of the game leads to measurable improvements in students' learning and conceptual understanding. Future studies could explore whether playing the games before lectures helps students better grasp key programming concepts. Evaluating these outcomes would help determine the educational value of the tool beyond engagement and motivation.

7 Conclusion

CS Connections introduces an interactive, game-based learning activity that instructors can easily integrate into their classrooms. By adapting the familiar structure of the NYT Connections game, the platform adds an interactive element to the classification puzzle. Results from its use in a CS1 course show promise in initial student engagement, although sustained interest retention remains a challenge. Survey results indicate that students are enjoying the game and find it useful as a review tool, but more can be done to encourage the students to treat this as a social activity. Instructors' experiences emphasize the difficulty in creating original and relevant puzzles, a challenge that should be alleviated as the knowledge base of puzzles expand through continued instructor use. Another insight was to intentionally thinking of a "fun" factor for each puzzle such as by using red herring elements and varying category types.

Ultimately, CS Connections represents one component of the active learning toolkit, offering educators a convenient and enjoyable addition to their teaching arsenal. In this way, the platform can serve as a robust tool for fostering both engagement and deeper learning in CS and other subjects.

References

- [1] AndComputers. 2024. *React Connections Game*. <https://github.com/and-computers/react-connections-game>
- [2] Anthony Kenzo Salazar. 2024. *Swellgarfo Connections*. <https://connections.swellgarfo.com/>
- [3] Jens Bennedsen and Michael E. Caspersen. 2007. Failure rates in introductory programming. *SIGCSE bulletin* 39, 2 (2007), 32–36. <https://doi.org/10.1145/1272848.1272879>
- [4] Claire Wang. 2024. *Custom Connections*. <https://custom-connections-game.vercel.app/>
- [5] Karolína Dočkalová Burská, Vít Rusňák, and Radek Ošlejšek. 2022. Data-driven insight into the puzzle-based cybersecurity training. *Computers Graphics* 102 (2022), 441–451. <https://doi.org/10.1016/j.cag.2021.09.011>
- [6] Ethan Fong, Michelle Craig, and Jonathan Calver. 2025. Crafting Interesting Puzzles with CS Connections. In *Proceedings of the 2025 Conference on Innovation and Technology in Computer Science Education V. 2* (Nijmegen, Netherlands) (*ITiCSE 2025*). Association for Computing Machinery, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3724389.3731261>
- [7] Juho Hamari, David Shernoff, Brianno Collier, Jodi Asbell-Clarke, and Teon Edwards. 2016. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior* (08 2016). <https://doi.org/10.1016/j.chb.2015.07.045>
- [8] Maria-Blanca Ibáñez, Ángela Di-Serio, and Carlos Delgado-Kloos. 2014. Gamification for Engaging Computer Science Students in Learning Activities: A Case Study. *IEEE Transactions on Learning Technologies* 7, 3 (2014), 291–301. <https://doi.org/10.1109/TLT.2014.2329293>
- [9] Elina Jääskä and Kirsi Aaltonen. 2022. Teachers' experiences of using game-based learning methods in project management higher education. *Project Leadership and Society* 3 (2022), 100041. <https://doi.org/10.1016/j.plas.2022.100041>
- [10] Abdullah Karaksha, Gary Grant, Shailendra Anoopkumar-Dukie, S. Niru Nirthanan, and Andrew K. Davey. 2013. Student Engagement in Pharmacology Courses Using Online Learning Tools. *American Journal of Pharmaceutical Education* 77, 6 (2013), 125. <https://doi.org/10.5688/ajpe776125>
- [11] Kathryn Kasmarik. 2010. An Empirical Evaluation of Puzzle-Based Learning as an Interest Approach for Teaching Introductory Computer Science. *Education, IEEE Transactions on* 53 (12 2010), 677 – 680. <https://doi.org/10.1109/TE.2009.2039217>
- [12] New York Times. 2023. *New York Times Connections*. <https://www.nytimes.com/games/connections>
- [13] Marina Papastergiou. 2009. Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers Education* 52, 1 (2009), 1–12. <https://doi.org/10.1016/j.compedu.2008.06.004>
- [14] R M Ryan and E L Deci. 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *Am. Psychol.* 55, 1 (Jan. 2000), 68–78.
- [15] Bianca L. Santana and Roberto A. Bittencourt. 2018. Increasing Motivation of CS1 Non-Majors through an Approach Contextualized by Games and Media. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE Press, 1–9. <https://doi.org/10.1109/FIE.2018.8659011>
- [16] Shane N. Stone and Steven Kirshblum. 2024. PMR connections: A pilot study evaluating the adoptability and impact of a serious educational game on trainees at a Physical Medicine and Rehabilitation Program. *American Journal of Physical Medicine and Rehabilitation* (Sep 2024). <https://doi.org/10.1097/phm.0000000000002620>
- [17] Errol Thompson, Jacqueline Whalley, Raymond Lister, and Beth Simon. 2006. Code Classification as a Learning and Assessment Exercise for Novice Programmers.
- [18] April Tyack and Elisa D. Mekler. 2024. Self-Determination Theory and HCI Games Research: Unfulfilled Promises and Unquestioned Paradigms. *ACM Trans. Comput.-Hum. Interact.* 31, 3, Article 40 (Aug. 2024), 74 pages. <https://doi.org/10.1145/3673230>
- [19] Megan Venn-Wyherley, Ahmed Kharrufa, Susan Lechelt, Rebecca Nicholson, Kate Howland, Abrar Almjally, Anthony Trory, and Vidya Sarangapani. 2024. The Realities of Evaluating Educational Technology in School Settings. *ACM Trans. Comput.-Hum. Interact.* 31, 2, Article 26 (Feb. 2024), 33 pages. <https://doi.org/10.1145/3635146>