# You Hacked My Program! Teaching Cybersecurity using Game-based Learning

Md. Hasan Tareque, Steven Deutekom, John Anvik, Maimoona Bashir *
[mdhasan.tareque,deutekom,john.anvik,maimoona.bashir]@uleth.ca
University of Lethbridge
Lethbridge, Alberta, CANADA

## ABSTRACT

As cyberthreats become more commonplace, the teaching of cybersecurity concepts at an introductory level is becoming increasingly important. However, teaching this subject in an engaging manner is challenging. This work investigates the use of a game-based learning approach to teaching cybersecurity concepts in the form of a card game called Program Wars. Within the game, players use cards to create a representation of a computer program while launching cyberattacks at their opponents and defending their own program. As the initial version of the game presented cybersecurity concepts at only a high-level, Program Wars *v.*2.0 was created to introduce players to eight common cyberattacks and the tools used to defend against them. The results of a user study show that after playing Program Wars *v.*2.0 a player's knowledge of cybersecurity concepts is improved, showing that our game-based learning approach provides an effective means for introducing cybersecurity concepts to those with little or no prior knowledge. As Program Wars is a freely available web-based game, it can easily be integrated into classes to improve a student's knowledge of cybersecurity concepts.

## CCS CONCEPTS

• **Applied computing** → **Interactive learning environments**; • **Social and professional topics** → *Computational thinking*; *CS1*; *Computing literacy*.

## KEYWORDS

Cybersecurity education; Game-based Learning; Web application

**ACM Reference Format:**
Md. Hasan Tareque, Steven Deutekom, John Anvik, Maimoona Bashir . 2024. You Hacked My Program! Teaching Cybersecurity using Game-based Learning. In *The 26th Western Canadian Conference on Computing Education (WCCCE '24), May 02–03, 2024, Kelowna, BC, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3660650.3660672

## 1 INTRODUCTION

Effective learning happens when teaching involves the combination of social and educational aspects [10]. This is especially true in a computer science curriculum where a vital role is played by experience-based learning. However, difficulties arise in applying an experience-based approach due to a scarcity of realistic learning environments, lack of proper tools and limited time. An approach to addressing these issues is the use of Game-Based Learning (GBL) in the form of serious games [9], where the usefulness of the entire gameplay in the context of learning is the focus. Already, this approach has been in use in CS education for teaching object-oriented programming [15], debugging [14] and software project management [8]. In other words, the use of serious games in a CS curriculum is not novel and has been shown to be effective [7].

The best methods for acquiring cybersecurity knowledge is an open question among cybersecurity experts. A leading theory posits that for efficiently training a cybersecurity incident response team (CSIRTs), the training should be designed as being directly applicable, with training activities that include role-playing, games, and practical exercises [5]. For example, the National Security Agency and the National Science Foundation jointly funded the GenCyber [2] program to stimulate K–12 students' interest in the cybersecurity field. Purdue University Northwest used GenCyber to teach 181 high school students across four summer camps, with a 51% of the students being from underrepresented minority communities (i.e. African American and Hispanic) and a male to female ratio of about 2:1 [11]. Similarly, the work of Karagiannis and Magkos [13] aspires to discover how simulation computer games could be transformed into virtual learning environments and enhance the learning process by increasing the motivation and engagement levels of undergraduate students in the topic of cybersecurity. Also, Silvestro et al. [17] proposed the use of Virtual Reality (VR) videogames to educate users about cybersecurity.

Program Wars[1] is a web-based card game created to investigate further the use of GBL for teaching programming language and cybersecurity concepts. It provides a learning environment meeting the attributes of role-playing, games, and practical exercises previously noted as being important for teaching cybersecurity [5]. Table 1 provides a summary of the various game elements and how they align with the *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering* [12] in the knowledge area of `Security (SEC)`.

Following a formal user study of Program Wars *v.*1.0 [6] and subsequent informal feedback, an updated version (Program Wars *v.*2.0) was created to address the identified limitations. Revising the user interface and gameplay was done to improve the game's effectiveness in meeting its educational goals. Although some changes

*We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2018-06004].



*WCCCE '24, May 02–03, 2024, Kelowna, BC, Canada*

ACM ISBN 979-8-4007-0997-5/24/05
https://doi.org/10.1145/3660650.3660672

[1] https://program-wars.firebaseapp.com

**Table 1: Alignment of features with Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering.**

| Concepts | Program Wars *v.*2.0 | Curriculum Guidelines |
|---|---|---|
| Cyberdefense/Safety | `Antivirus` & `Firewall` and `Computer Scan` cards | SEC.net.3 |
| Cyberattack | `Virus`, `Ransomware`, `Spyware` and `Trojan Horse` cards | SEC.net.1 |
| Hacking | `Buffer Overflow`, `Cross-site Scripting`, `SQL Injection` and `Denial of Service` cards | SEC.net.1 |

focused on addressing limitations in teaching fundamental programming concepts identified in the prior user study[2], this paper focuses on the changes made to enhance the learning of cybersecurity concepts and the evaluation of their effectiveness using a user study. Specifically, the `Malware` and `Hack` cards in Program Wars *v.*1.0 were each replaced by four specific cybersecurity threats, the `Computer Scan` card representing a one-time computer scan was introduced, and two cards deemed to not fit well with the cybersecurity learning objectives were removed.

The research question for this work is "*Do the refinements to the UI and gameplay of Program Wars* v.*2.0 improve a player's understanding of basic cybersecurity concepts?*" A user study was conducted to answer our research question. After playing Program Wars *v.*2.0, we found that most subjects' knowledge about cyberattacks and cyberdefense improved.

This paper proceeds with overviews of similar games to Program Wars that teach cybersecurity. Next, we describe the cybersecurity aspects of Program Wars *v.*2.0 before presenting the details and results of the user study. Finally, we conclude the paper.

## 2 RELATED WORK

Although several research works investigated efforts for teaching basic cybersecurity concepts, we will restrict ourselves to those where the medium used is a card game or a board game, as these are the most comparable to Program Wars.

Cyber Threat Defender [1] is a collectable cybersecurity card game. The game's goal is to build a network as quickly as possible so that it can do more business and gain more points. While doing so, a player has to remember to defend their network because the opponent is going to try and disrupt other player's systems and networks. There is a defence for every attack, and for every defence, there is a corresponding attack to get around it. The player with the complete set of security defences will be the one who is able to protect the critical systems and emerge victorious. For example, adding a wireless router without encryption will make the network vulnerable to attackers in the gameplay. For a player to successfully defeat their opponent, they must develop and implement a strategy for expanding and protecting their network. The main objective of the game is to make players familiar with basic and complex cybersecurity concepts.

Potato Pirates 2: Enter The Spudnet [3] is a board game that introduces players to various cybersecurity concepts. In the game, a player plays across a network of shipping ports where they must fulfill their five potato orders (i.e. send data packets) while playing cards to benefit themselves or harm others' shipments or structures. The board provides an analogy of a network map, and the map is organized into interconnected, coloured networks (shipping lanes) of nodes (ports), with each port having its own IP address. Players can place firewalls, which block travel to others, and play cards representing Trojans, Ransomware, and other cyberthreats. The gameplay can be competitive or cooperative, with each game style giving rise to its own strategies and approaches. As players move their potatoes across the shipping network, they will face various network hindrances such as navigating inconvenient firewalls and frustrating connection slowdowns when warehouse nodes get overloaded.

## 3 CYBERSECURITY ASPECTS OF PROGRAM WARS *V.*2.0

In Program Wars *v.*2.0, players build a representation of a computer program using cards that denote the use of instructions, methods, variables, and loops. Also, players can use cards that represent search and sorting algorithms to examine and manipulate the deck. Finally, players can launch a cyberattack on their opponent during gameplay or prepare their cyberdefense. The two-game modes added in Program Wars *v.*2.0 (Beginner and Standard) allow players to add different sets of cyberattack and cyberdefense cards into the deck. The combinations of cybersecurity cards in various modes are presented in Table 2.

We begin with an overview of Program Wars *v.*2.0. Then, we describe the specific cybersecurity cards in Program Wars *v.*2.0.

### 3.1 Overview of Program Wars *v.*2.0

The player, which we will refer to as Stone, starts by going to https://program-wars.firebaseapp.com/ to begin a game. Stone can play against another human on the same computer (i.e. hot-seat play) or against provided computer opponents. In this case, Stone chooses to play against a computer opponent (`n00b_b0t`). Next, Stone chooses between two modes of gameplay: *Beginner* and *Standard*. The two gameplay modes were created in response to feedback from the Program Wars *v.*1.0 user study [4], where participants commented that "once I got the hang of the basics, there wasn't much room to improve". Stone chooses the *Beginner* mode and the *Malware 1* card set to start the game. By selecting the *Malware 1* card set, `Spyware` and `Ransomware` cyberattack cards are added to the deck, as well as an `Antivirus` card for each player.

The basic Program Wars deck is comprised of `Instruction`, `Method`, `Repeat`, `Variable`, `Search`, `Sort` and `Computer Scan` cards. Program Wars is played in a series of rounds in which each player

[2]Details regarding these changes and their evaluation via a user study can be found in [16].

**Table 2: Game modes and Card sets in Program Wars *v.2.0* .**

| | | Beginner | | | | Standard | | | |
|---|---|---|---|---|---|---|---|---|---|
| Card Type | Card Name | Malware 1 | Hack 1 | Malware 2 | Hack 2 | Malware | Hack | Combined 1 | Combined 2 |
| Safety | AntiVirus | ✓ | | ✓ | | ✓ | | ✓ | ✓ |
| | Firewall | | ✓ | | ✓ | | ✓ | ✓ | ✓ |
| Malware | Spyware | ✓ | | | | ✓ | | ✓ | |
| | Ransomware | ✓ | | | | ✓ | | | ✓ |
| | Virus | | | ✓ | | ✓ | | ✓ | |
| | Trojan | | | ✓ | | ✓ | | | ✓ |
| Hack | Buffer Overflow | | ✓ | | | | ✓ | ✓ | ✓ |
| | Cross-site Scripting | | | | ✓ | | ✓ | | |
| | DoS Attack | | ✓ | | | | ✓ | | ✓ |
| | SQL Injection | | | | ✓ | | ✓ | ✓ | |


**Figure 1: An annotated user interface for Program Wars *v.2.0* after eight turns of a game.**

takes turns to either play a card, discard a card, or draw a new hand. Each player's hand consists of five cards, with a sixth card drawn at the start of their turn. The play area of the current player and the player's currently selected card are highlighted in green in the game's UI. On a player's turn, they can build their program, launch a cyberattack at an opponent, or prepare their cyberdefense.

To win, Stone will need to create a program that reaches the goal number of points, where points represent the number of instructions that would be executed based on the cards played. Programs are built by creating stacks of `Instruction`, `Method`, `Repeat` and `Variable` cards. Each stack of cards represents a portion of the player's program. Above each stack is shown how many points the stack contributes to the player's score. The total score for a player is the sum of all of their stack scores. Stacks with green scores indicate that the stack is complete[3] and no more cards can be added to it. If a player reaches or exceeds the goal number of points, the game finishes at the end of the current round. In *Beginner* mode, the player's instruction score determines the winner, and players can tie. In *Standard* mode, bonus points are awarded for reaching specific objectives, such as having played an `Antivirus` or not being affected by any cyberattacks by the end of the game. If both players have the same total score, these bonus points are used to break ties. Clicking on the "Bonus" tab in the UI shows a series of *if-then* statements that specify the bonus objectives. The statement

[3]A complete stack has two `Repeat` cards played on it with `Repeat-X` cards only counting if they are paired with a `Variable` card.

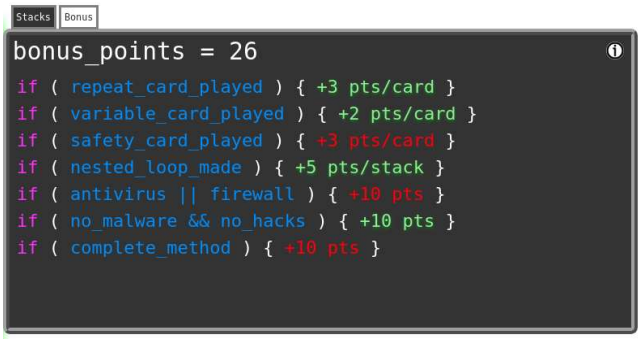


**Figure 2: An example of a player's bonus objectives.**

will be shown in green if the player has met the objective and in red otherwise. This is how conditional statements are presented in Program Wars *v.*2.0. Figure 2 shows an example where a player has used at least two `Repeat` cards, a `Variable` card, has a nested loop, and has no cyberthreats affecting their program.

## 3.2 Cyberattacks and Cyberdefense during Gameplay

Figure 1 shows an example game of Program Wars *v.*2.0 after eight turns. The bottom left shows that Stone has played `Instruction` cards in their *Method Stack* area (Area 1a), giving a value of 9 to their `Method` cards, and that they have played a `Repeat`-3 card on a `Method` card to make a 'program' of 27 instructions (Area 2a). Similarly, the bottom right of Figure 1 shows `n00b_b0t`, the computer opponent, has played `Instruction` cards to a total of 9 in their *Method Stack* and created a nested loop using a `Repeat`-3 and `Repeat`-2 cards to create a 'program' of 54 instructions.

Stone launched a cyberattack on their fourth turn by playing a `Spyware` card against the computer. Active cyberthreats are visible in Area 3a of the UI, along with the number of turns they will be active in the top left corner of the icon. Stone can now see the computer's hand by pressing the "Spy Hand" area (3b in the figure) for the next five turns. This will help Stone play more strategically by anticipating his opponent's moves.

On their sixth turn, Stone launched another cyberattack by playing a `Ransomware` card. The effect of this attack shifts 10 points from the computer's total to Stone's total. This effect is reflected in Stone's total, which is 37 even though their program totals 27 instructions. Similarly, (`n00b_b0t`) shows an instruction total of 44 instead of the 54 indicated in their play area.

Finally, fearing reprisal from the computer for his two cyberattacks, Stone played an `Antivirus` card to protect themselves from future cyberattacks. The effect and the icon are visible in the "Threat Prevention" area (4a of Figure 1).

Gameplay continues from the point shown in Figure 1 with the two players continuing to build their programs, attacking each other or defending themselves until there is a winner.

## 3.3 Cyberattack and Cybersecurity Cards

This section presents the eight cards representing the cyberattacks taught by Program Wars *v.*2.0. To defend and guard against these cyberattacks, three cards represent scanning the computer for threats, running an antivirus program in the background, and using a firewall.

*3.3.1 Cyberattacks.* The cyberattack cards of Program Wars *v.*2.0 are divided into two categories: *Malware* and *Hacking*. The *Malware* category contains cards representing four of the most common types of malware (`Spyware`, `Ransomware`, `Virus` and `Trojan Horse`), and the *Hacking* category contains representations of three types of code injection attacks (`SQL Injection`, `Cross-site Scripting`, and `Buffer Overflow`) and a denial of service attack.

*Malware.* Spyware is used to gather and send information to another party without the target's consent. In the context of the game, when the `Spyware` card is used, a button labelled "Spy Hand" appears beside the opponent's name in the top portion of the screen for five turns allowing the attacker to see the affected player's hand.

The `Ransomware` card's effect reflects the real-world event where an attacker blocks access to a target's files, such as encrypting them, and threatens to publish or delete them unless a ransom is paid. When a player plays a `Ransomware` card, the targeted player loses 10 points from their total score and these points are added to the attacker's score.

A computer virus is a computer program that replicates itself by modifying other programs. In the game, this translates to having the `Virus` card reduce the points contributed by a set of cards representing computer instructions. If the card is an `Instruction`, the contributing points become 0. If the card is a `Method`, the contributing points are reduced by 50%. This difference is to encourage players to consider how modularity can improve the security of the software.

In the context of computing, a "Trojan horse" is a computer program that misleads users as to its real intent. When a `Trojan Horse` card is played against an opponent, a random card in the opponent's hand is replaced with one that mimics it. Although players can see that a `Trojan Horse` is played on them, they cannot tell which card has been replaced. The actual effect of the mimicked card depends on which card is replaced. Playing a `Instruction` or `Method` card activates the `Buffer Overflow` effect on the player. Playing a `Repeat` or `Variable` card adds a `Virus` card where the card was played. Cyberattack cards activate a `Cross-site Scripting` attack on the player instead of attacking the opponent. Playing a cyberdefense or algorithm card activates a `Ransomware` on the player.

*Hacking:* Hacking is an intrusion into a computer system, and Program Wars *v.*2.0 has cards to represent four common ways whereby a computer system is compromised: causing a memory buffer overflow, a cross-site scripting attack, a denial of service attack, or injection of malicious SQL code.

Code injection attacks, such as memory buffer overflows, cross-site scripting and SQL injection, result in malicious code being run and disrupting a computer's regular operation. In Program Wars *v.*2.0, these three code injection attacks are represented by cards that have the effect of either preventing a player from playing

cards or corrupting a player's program. The `Buffer Overflow` card prevents the playing of an `Instruction`, `Repeat`, `Variable` or `Method` card for two turns. The `Cross-site Scripting` card prevents a player from playing any algorithm or cyberattack cards for two turns. The `SQL Injection` card corrupts an opponent's program by reducing the total of each of their `Method` cards by two points. The effect of the `SQL Injection` card lasts until removed by a `Firewall` or `Computer Scan` card.

A denial-of-service attack occurs when a networked computer system is intentionally flooded with fraudulent requests so that the system can no longer handle legitimate requests. The `Denial of Service` card prevents a player from drawing a new card for three turns, thus simulating an unresponsive computer system.

*3.3.2 Cyberdefense:* Program Wars *v.*2.0 provides three cards for cyberdefense. Two of the cards are *permanent* cards, meaning that they continue to protect the player until the end of the game. The first of these cards is the `Antivirus` card which prevents any *Malware* cards from being played on a player. The second is the `Firewall` card which protects against the *Hack* cards. The final card is the `Computer Scan` card, which represents the action of a user explicitly scanning all of their files to find any infected items using an antivirus tool. If the player is under the influence of a cyberattack card, then the `Computer Scan` card allows the player to remove a single effect. The `Antivirus` and `Firewall` will remove all relevant effects when played. There are several `Computer Scan` cards in the deck, but only one `Firewall` and one `Antivirus` card in the deck per player.

## 4 USER STUDY OF PROGRAM WARS *V.*2.0

To assess changes in cybersecurity knowledge, we chose to conduct a 'within-subjects' study. In other words, subjects were asked questions about cybersecurity concepts both before and after playing Program Wars *v.*2.0, and their responses were compared.

The user study for Program Wars *v.*2.0 was conducted online and without any supervision due to COVID protocols present at the time.[4] The user study had three phases. First, the subject completed a pre-game survey consisting of demographic questions and questions to assess the subject's prior knowledge and experience with cybersecurity. Next, they were asked to play Program Wars *v.*2.0 until they felt they understood the game's concepts (minimum of three times, maximum of ten times). Finally, they completed a post-game survey with knowledge-testing questions similar to those of the pre-game survey, and opportunity to provide feedback.

To participate in the study, a subject must have been 18 years of age or older with little to no cybersecurity knowledge. Subjects were recruited from within the university community. Participation in the study was expected to take 60 ∼ 90 minutes, depending on the number of times the game was played. Unique study IDs were used to link the pre- and post-game responses.

Cybersecurity knowledge questions were asked in both the pre- and post-game surveys. Subjects were asked two different cybersecurity questions. Given the description of a real-world scenario, the subject was asked to identify which cybersecurity threat was described. One set of questions described a *Ransomware* cyberattack, and the other described a *Spyware* cyberattack. Figure 3 shows an example question; the post-game question for *Ransomware*. Also, in the post-game survey, subjects were asked to identify the cyberdefense technique for a specific cyberattack (Figure 4).

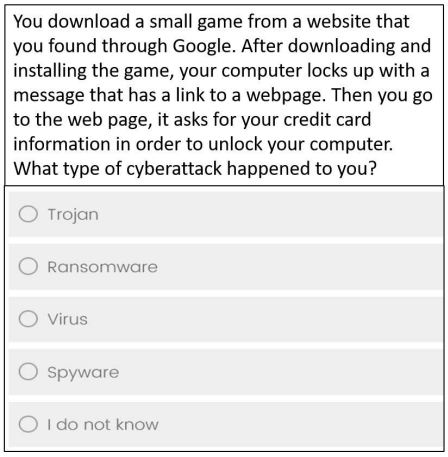


**Figure 3: Ransomware post-game survey question.**

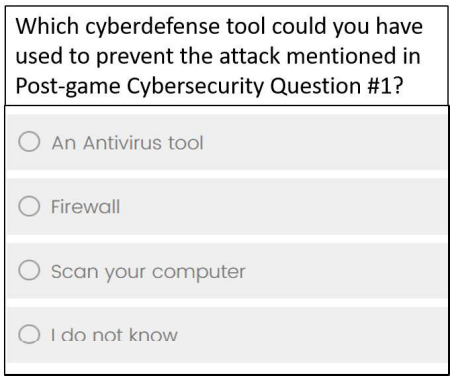


**Figure 4: Cyberdefense post-game survey question.**

### 4.1 Results

Twenty-six subjects met the participation criteria and completed all of the stages of the study. Of the 26 subjects, 23 were from the 17 to 24 age group, and three were from the 25 to 34 age group. Regarding their education, 22 reported completing their high school degree, 2 had their bachelor's degree, 1 had a graduate level, and 1 had an associate (2-years) degree. Regarding cybersecurity experience, 14 subjects reported no experience with cybersecurity, and 12 reported being novices. All subjects reported playing the game at least three times in the post-game survey.

*4.1.1 Changes in Cybersecurity Knowledge.* We completed a 'within-subject' analysis whereby we categorized a subject's knowledge improvement into one of four categories. If a subject got both the pre-and post-game questions for a concept correct, they were considered to have had *Prior Knowledge* of that concept before playing the game. If a subject answered a concept's question(s) correctly

[4]The study procedures were reviewed and approved by the university's Human Subject Research Committee. Specific details of the study procedure can be found in [16].

**Table 3: Knowledge changes from playing Program Wars *v.*2.0.**

| Knowledge Improvement | Pre-Game | Post-game | Ransomware | Spyware |
|---|---|---|---|---|
| Prior Knowledge | Correct | Correct | 4 | 1 |
| Unclear | Correct | Incorrect | 3 | 4 |
| Improved | Incorrect | Correct | 14 | 15 |
| No Change | Incorrect | Incorrect | 5 | 6 |

in the pre-game survey but incorrectly in the post-game survey, then their knowledge improvement was considered as *Unclear* as they seemed to have lost knowledge. If a subject answered a concept's question(s) incorrectly in the pre-game survey and correctly in the post-game survey, we considered this a demonstration that *Knowledge Improved*. Finally, if a subject got a concept's questions incorrect in both the pre-and post-game survey, this was considered to be *No Change in Knowledge*.

The results for the concept areas of *Ransomware* and *Spyware* are shown in the last two columns of Table 3. We see that 14 subjects improved their knowledge about the *Ransomware* cyberattack, and 15 subjects improved their knowledge about the *Spyware* cyberattack.

A paired samples t-test was performed to compare the improvement between pre- and post-game scores for the two cyberattacks. There was a significant difference in scores between pre- ($M = 0.27, SD = 0.20$) and post-game ($M = 0.69, SD = 0.22$) for ransomware; $t(24) = -3.070, p = 0.003$. Similarly, there was a significant difference in pre- ($M = 0.19, SD = 0.16$) and post-game ($M = 0.62, SD = 0.25$) scores for spyware; $t(24) = -2.848, p = 0.004$.

When asked in the post-game survey how to prevent such an attack, 14 subjects and 16 subjects correctly identified the cyberdefense techniques for *Ransomware* and *Spyware*, respectively.

Therefore we conclude that **"Program Wars does improve a player's knowledge of cybersecurity"**.

As the previous study of Program Wars [6] did not assess its learning effects for cybersecurity, we cannot compare the two game versions for this aspect.

*4.1.2 Subject Feedback.* Almost all subjects (23) reported that they liked playing Program Wars *v.*2.0, and 19 agreed that they would suggest playing Program Wars *v.*2.0 to their friends. Regarding self-assessment of learning, many noted they were not sure that playing Program Wars *v.*2.0 improved their cybersecurity knowledge. Perhaps this results from not providing feedback to subjects on whether they answered questions correctly. When asked whether they will play Program Wars in the future, 11 subjects said that they would play Program Wars *v.*2.0 in future to improve their knowledge.

When asked for additional comments, some subjects indicated it was a 'good game' and they had fun playing it. A couple of subjects found the UI and game rules to be complex. However, this comment is not surprising, considering that for any new game, the initial couple of rounds are typically a challenge as players learn the rules of that game. One of the subjects remarked that the game has *"A very good concept to teach [...] cybersecurity basics and ideas, The game is really enjoyable, and it enhanced my knowledge regarding basic [...] cybersecurity."*

## 4.2 Threats to Validity

As the study was conducted asynchronously online due to COVID restrictions, this situation may have affected the validity of the results as subjects were not able to ask clarifying questions, and researchers were not able to make direct observations. This is likely most evident from the subjects' comments regarding the complexity of the game. Had the subjects been observed while completing the study, clarifications could have been provided, and further insights regarding Program Wars *v.*2.0 may have been possible. Although Program Wars does provide in-game access to web pages describing the gameplay and cards, it is unknown if subjects used this information.

To judge changes in a subject's knowledge of a cybersecurity concept, we compared a subject's answers in their pre- and post-game surveys. It is possible that some subjects guessed answers. For example, those who were considered to have *Had Prior Knowledge* may contain some subjects that correctly guessed answers in both surveys or those with *Unclear* may have correctly guessed the answer in the pre-game survey and incorrectly in the post-game survey. As subjects were not observed when completing the study, we cannot determine if this behaviour existed and, if so, how much it impacted the results.

As the cards in the game are dealt randomly, and the players choose what cards to play, it is possible that some subjects may have had little to no experience with the `Ransomware` and `Spyware` cards during their games. This situation could result in the subject being asked questions in the post-game survey, which they could not answer fairly. We believe this threat is unlikely as a minimum of three games were played.

# 5 CONCLUSION

We have presented an evolution of the cybersecurity concepts, game elements, and user interface originally introduced by Program Wars *v.*1.0 [6] to support the use of GBL in teaching basic cybersecurity concepts to those with little to no experience in this area. Specifically, the cards which represented the general concepts of *Malware* and *Hacking* were each replaced with cards that represent specific common cyberthreats.

To assess the effectiveness of Program Wars in teaching basic cybersecurity concepts, a user study of Program Wars *v.*2.0 was conducted. As more than 50% of the subjects demonstrated improved knowledge of at least two cyberthreats, we conclude that the game's application of GBL towards teaching cybersecurity is effective.

We plan to continue investigating how playing Program Wars affects players' knowledge about programming and cybersecurity with a cross-institution study or a study at the K-12 level.

# REFERENCES

[1] [n. d.]. Cyber Threat Defender. https://cias.utsa.edu/ctd.php. [Online; accessed 15-Mar-2022].

[2] [n. d.]. INSPIRING THE NEXT GENERATION OF CYBER STARS. https://www.gen-cyber.com/. [Online; accessed 15-Mar-2022].

[3] [n. d.]. Potato Pirates 2: Enter The Spudnet. https://potatopirates.game/products/enter-the-spudnet-board-game. [Online; accessed 15-Mar-2022].

[4] [n. d.]. Program Wars. https://program-wars.firebaseapp.com/. [Online; accessed 15-Mar-2022].

[5] Giddeon Angafor, Iryna Yevseyeva, and Ying He. 2020. Game-based learning: A review of tabletop exercises for cybersecurity incident response training. *Security and Privacy* 3 (07 2020). https://doi.org/10.1002/spy2.126

[6] John Anvik, Vincent Cote, and Jace Riehl. 2019. Program Wars: A Card Game for Learning Programming and Cybersecurity Concepts. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 393–399. https://doi.org/10.1145/3287324.3287496

[7] Alex Baker, Emily Oh Navarro, and André van der Hoek. 2005. An experimental card game for teaching software engineering processes. *Journal of Systems and Software* 75, 1 (2005), 3–16. https://doi.org/10.1016/j.jss.2004.02.033 Software Engineering Education and Training.

[8] Alejandro Calderón, Mercedes Ruiz, and Elena Orta. 2017. Integrating Serious Games as Learning Resources in a Software Project Management Course: The Case of ProDec. In *2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM)*. 21–27. https://doi.org/10.1109/SECM.2017.3

[9] G. Bente J. S. Breuer. 2010. Why so serious? On the Relation of Serious Games and Learning. *Eludamos, Journal for Computer Game Culture* 4, 1 (2010), 7–24.

[10] Molly Stewart LawlorKimberly A. Schonert-ReichlJenna Whitehead Jacqueline E. Maloney. 2016. *A Mindfulness-Based Social and Emotional Learning Curriculum for School-Aged Children: The MindUP Program.* Springer, New York, NY.

[11] Ge Jin, Manghui Tu, Tae-Hoon Kim, Justin Heffron, and Jonathan White. 2018. Game Based Cybersecurity Training for High School Students. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 68–73. https://doi.org/10.1145/3159450.3159591

[12] IEEE Computer Society & Association for Computing Machinery Joint Task Force on Computing Curricula. 2015. Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. https://www.acm.org/binaries/content/assets/education/se2014.pdf. [Online; accessed 16-Oct-2020].

[13] Stylianos Karagiannis and Emmanouil Magkos. 2021. *Engaging Students in Basic Cybersecurity Concepts Using Digital Game-Based Learning: Computer Games as Virtual Learning Environments.* Springer International Publishing, Cham, 55–81. https://doi.org/10.1007/978-3-030-41196-1_4

[14] Michael A. Miljanovic and Jeremy S. Bradbury. 2017. RoboBUG: A Serious Game for Learning Debugging Techniques. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (Tacoma, Washington, USA) *(ICER '17)*. Association for Computing Machinery, New York, NY, USA, 93–100. https://doi.org/10.1145/3105726.3106173

[15] José María Rodríguez Corral, Antón Civit Balcells, Arturo Morgado Estévez, Gabriel Jiménez Moreno, and María José Ferreiro Ramos. 2014. A game-based approach to the teaching of object-oriented programming languages. *Computers & Education* 73 (2014), 83–92. https://doi.org/10.1016/j.compedu.2013.12.013

[16] Md. Hasan Tareque. 2021. *Updating a Web-based Card Game to Teach Programming, Cybersecurity and Software Development Life Cycle Concepts.* Master's thesis. University of Lethbridge.

[17] Silvestro V. Veneruso, Lauren S. Ferro, Andrea Marrella, Massimo Mecella, and Tiziana Catarci. 2020. *CyberVR: An Interactive Learning Experience in Virtual Reality for Cybersecurity Related Issues.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3399715.3399860