Diverse data structures applied in the creation of console games as a result of a discipline's assessment project

Rúbia Eliza de Oliveira Schultz Ascari[1], **Eliane Maria De Bortoli Fávero**[1]

[1]Federal University of Technology - Paraná (UTFPR)
Pato Branco, Paraná, Brazil

rubia@utfpr.edu.br, elianedb@utfpr.edu.br

Abstract. *Project development is commonly used to assess disciplines in higher education courses linked to Computer Science. Data Structure is a subject of great importance for courses in this area, however, its contents are often considered complex and tedious by students. Looking for an attractive alternative to partially assess the students of a Data Structure 2 discipline of a Technology in Systems Analysis and Development course, a game-based approach was used for final project development. The students developed simple games, with interaction via console, using C program language and at least one of the Data Structures studied. Results are briefly described in this paper, and the hope is that sharing the creativity displayed by students could be helpful to educators.*

1. Introduction

Data Structure is a classic and essential discipline for Computer and related areas [Barbosa and Júnior 2013]. This discipline has plenty of content presented, which must be approached sequentially, theoretically, and practically using code execution tools, being generally offered in the first semesters of higher education courses linked to Computer Science [Ishikawa and Ralha 2015]. The Technology in Systems Analysis and Development (TSAD) course at the Federal University of Technology - Paraná (UTFPR), Pato Branco Campus, covers the Data Structure content in two disciplines, offered in the second and third periods of the course, which has six periods. Called "Data Structure 1" and "Data Structure 2" these disciplines require a high capacity for abstraction [Feitosa 2023] and some essential skills [Silva et al. 2011]: text interpretation, logical reasoning, good use of mathematics, and use of algorithmic structures. Lacking motivation to either carry out the exercises or participate in the classes actively commonly results in difficulties in the learning process, given the structural complexity of the cited disciplines. Student's poor performance in these subjects can compromise the rest of the course and is also a possible cause of university dropout in the area [Barbosa and Júnior 2013].

Several pedagogical proposals have been investigated to engage the students in learning, preventing them from being mere receivers of information and become proactive agents in constructing their knowledge [Ishikawa and Ralha 2015]. The use of play as a teaching resource for active learning is quite common and has shown significant results, as reported in studies of [Barros and Xavier 2022], and [de Souza et al. 2023].

Digital games contribute with a pedagogical strategy that allows the teacher to create playful learning contexts [Fardo et al. 2022]. Using a game environment to practice or learn new programming concepts is a viable and widely used approach to achieve student motivation; however, developing a game using programming knowledge demands

active learning. This is because it presents the students with opportunities for reflection and constructing meanings, which implies thinking, feeling, and acting; the students can create new knowledge in this way. In programming processes, knowing programming codes is insufficient for solving problems; having creative problem-solving skills (CPSS) is needed [Akar and Altun 2017]. CPSS encourages students to bring new perspectives and find new solutions. This is necessary for game development, as entertainment games could involve various elements such as adaptive challenge, curiosity, self-expression, discovery, clear goals, clear rules, player control, immersion, collaboration, competition, immediate feedback, and variable rewards, among others [Gurbuz and Celik 2022].

Considering that the development of projects is a common practice for assessing Data Structure disciplines and looking for an attractive alternative for the students, a game-based project was proposed as a partial assessment of the "Data Structure 2" discipline in the TSAD course. Using Digital game-based learning and challenging the students to put some data structures studied into practice, we hope to use active learning to stimulate proactivity and make the teaching-learning process more enjoyable.

2. Data Struct, the discipline and its challenges

For [Backes 2023], in computing, a data structure is a way of storing and organizing data to be efficiently used. Data structures and algorithms are essential subjects for Computer area students because their correct application considerably improves the efficiency of computational systems [Sedgewick and Wayne 2011]. Curricular Guidelines for Courses in the Area of Computing and Information Technology [Brasil 2001] indicate that the development of algorithms and the study of data structures should receive special attention when approaching the topic of programming. In the curricula proposed by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) for undergraduate courses in the area of Computer Science [Draft 2013], this importance is highlighted since the discipline is commonly a prerequisite for others.

Poor performance in algorithms compromises the entire course [Silva et al. 2011]. However, it is difficult for beginning students to develop a deep understanding of data structures and their algorithms. As highlighted by [Feitosa 2023], the literature cites several factors for student failure in the Data Structures discipline, the following being typical: (i) a heavy load of abstract concepts, (ii) learning failures, and (iii) lack of motivation. The acquisition of abstract knowledge and its application to reality through algorithms is already challenging in itself, so turning this process into something motivating for the students requires creativity by the teachers, and game-based approaches could be an option.

3. Digital game-based learning

To [Fernandes 2021], for learning to be more effective and attractive, especially considering the profile of students of current generations, interactive and engaging resources similar to those that students are familiar with in their routines are necessary, making them show greater interest in the proposed content. Thus, Digital game-based learning aligns with current and future generations' needs and learning styles; it motivates because it is fun and incredibly versatile, which may be adapted to many disciplines [Prensky 2021].

Digital games are part of many student's daily lives. When used in an educational context as a teaching-learning strategy, they allow students an immersive and motivating

experience, providing rich moments for building learning. When this approach is used correctly, it is extremely effective [Prensky 2021]. For [Fernandes et al. 2022], we must explore the use and development of games in the classroom, as developing your own game allows students to acquire computational skills and explore content more meaningfully.

4. Methodological Procedures

This section describes the activity proposed to graduation students as a partial assessment for the "Data Structures 2" discipline of the TSAD course at the UTFPR - Pato Branco Campus. The complete discipline assessment involves two theoretical-practical tests, lists of practical exercises, and project development. In this discipline, students are often challenged to build algorithms and apply them to real problems by employing CPSS. The proposal described is the final project of the discipline since, at the final stage of the semester, theoretical and practical content on various Data Structures and related subjects has already been covered, such as complexity analysis and algorithm design paradigms.

The activity's objective is to evaluate the ability of academics to employ content related to the different data structures studied in a practical case study in the context of games. The requirements involved the creation of a program aimed at manipulating data from a digital game of the student's choice, using the programming language C, with interaction via console. The program should use abstract data types, encapsulation, and at least one specified data structure within a schedule of approximately 12 class hours.

The activity was proposed in two consecutive semesters. In the first semester, the data structures required were: Linked List, Linked Stack, Linked Queue, Linked Dequeue, Tree, or Graph. In the second proposal of the activity, in the subsequent semester, the students were directed to use a B-tree or Graph in their projects. This change in the second offering occurred to avoid plagiarism of previous works and force new game development. Furthermore, the discipline had been momentarily interrupted due to a strike at the institution. After the class returned, these two topics had been studied more recently during the replacement period. The specifications for the proposal activity were:

- Suitability of the digital game for use with the chosen structure.
- Correct implementation and use of the chosen data structure, according to the concept of abstract data type and encapsulation studied.
- Definition of quantity and type of data suitable for the context of the chosen game.
- Game correctly implemented and functioning.
- Correct presentation of the game's objective and the expected form of interaction.
- Submission of a document describing the project's context, justifying the choices.
- Presentation of the finished project to the professor for proof of authorship.

5. Results

The students' digital games are the activity proposal's primary results. The games were mainly developed using Portuguese; however, the images presented in this document were translated into English. A brief description of each game is presented below, divided into two sections, referring to the two offerings of the discipline, which occurred in two consecutive semesters. Students answered a questionnaire to assess their perception of the proposal, and the results, presented in this section, corrobored the expectations that digital game-based learning contributes to learning.

5.1. First proposal - Games using any studied data structure

In the first proposal of the activity, the students could choose any game and one of several suggested data structures as long as it met the requirements. Thus, many students searched the Internet for ideas, and the result was that most focused their efforts on reproducing board games, commonly developed for interaction via console, as described below.

- **Minesweeper game**: It is a game that challenges the user to choose available spaces in an area called a minefield without activating hidden bombs and choosing only safe spaces. For this game, the academic used a two-dimensional matrix to represent the game board, where each cell is accessed by its row and column coordinates. A linked list was used to store the position of mines randomly generated at the beginning of the game and revealed only after the game was over. Figure 1 (a) presents an example of the Minesweeper game interface.
- **Bingo game**: The player must repeatedly identify a drawn number on their card in various rounds. When completing the card with the minimum number of items necessary, the user can "knock" by calling "Bingo". The user who completes the card faster is considered the winner. A linked list was used to store the numbers drawn in each round. The game developed enables the user to play alone or with 1, 2, and 3 virtual players. Multiple players require interaction through relays. Figure 1 (b) presents the interface of the Bingo Game.
- **Super Trump**: It is a collectible card game in which players compete to win all their opponent's cards. The game is based on choosing the attributes of each card, refering to characteristics of dogs of different breeds, such as energetic, playful, protective, jealous, and defensive. A linked stack stores each player's card and the respective attributes per round. In each round, the program chooses cards, and the two players will have timed time to view their cards individually and choose the desired attribute. The game ends after comparing all the attributes of the selected cards. Figure 1 (c) presents the interface of the Trump Game developed without the console clearing that occurs during the interaction.

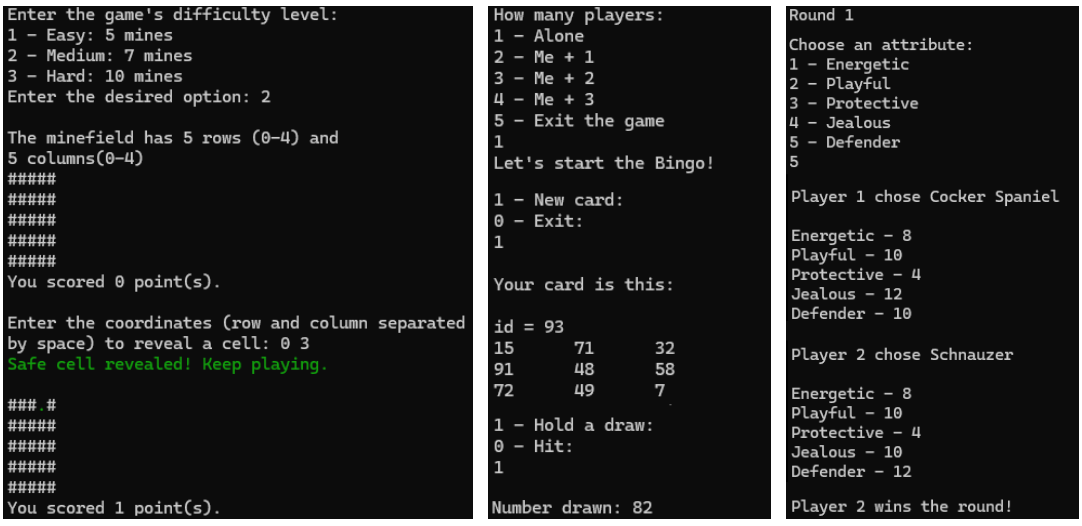
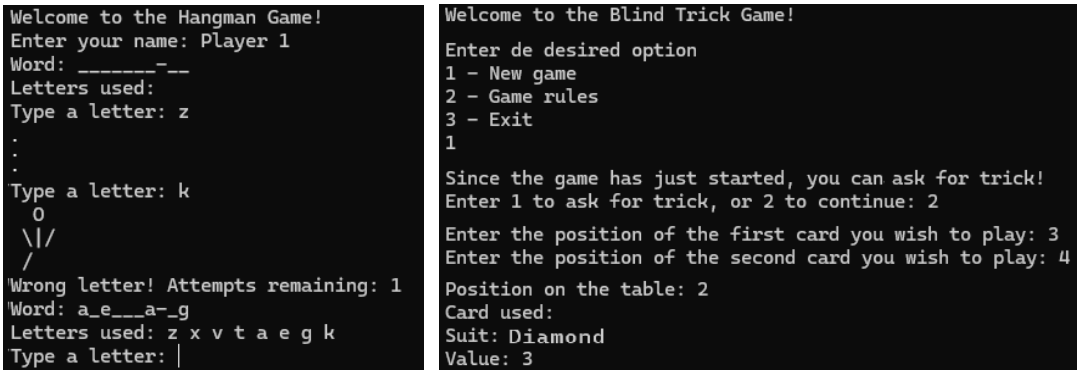


Figure 1. Interfaces of (a) Minesweeper (b) Bingo and (c) Super Trump games.

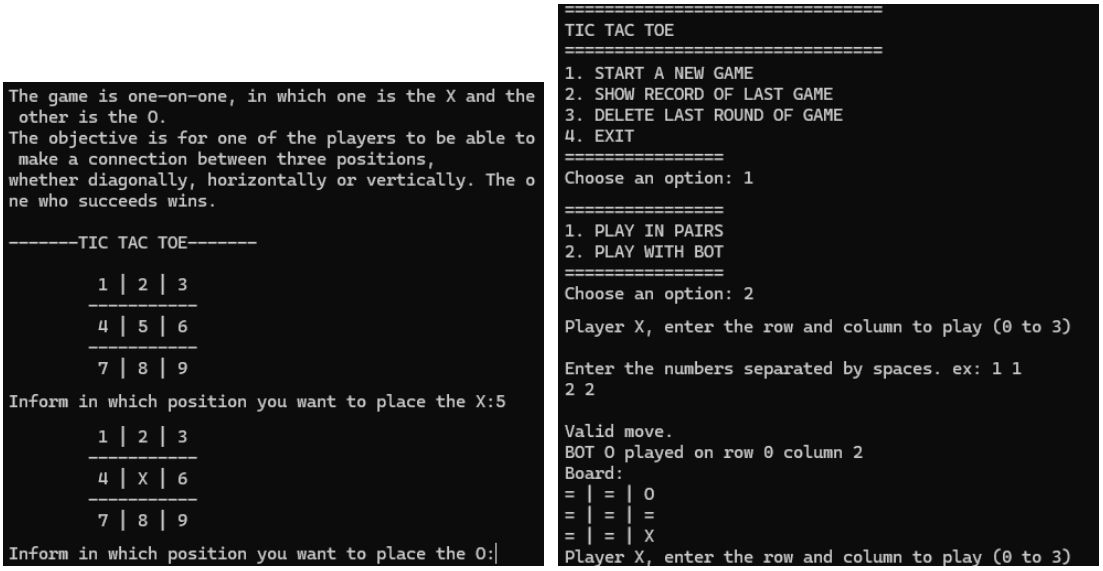
- **Hangman game**: In this game, the player tries to guess a randomly chosen word from among those stored in a vector composed of names of soccer teams from the Brazilian Championship Series A. Whenever the given letter is in the selected word, it is revealed in the hidden word. Otherwise, a part of the gallows drawing (a doll with a head, body, and limbs) is drawn, and the number of remaining attempts is displayed. The game continues until the player guesses the word correctly or misses the number of attempts (six), resulting in the complete printing of the doll, simulating its death, and game over. A linked list stores information about the players (name, number of attempts, game number). For each correct word, the player's data are inserted at the beginning of the user list. The list of people who played and guessed the correct words is printed after the user choose to stop playing. Figure 2 (a) presents the game's interface.
- **Blind trick game**: A simplified trick game was created, using linked lists to store cards from a deck, with information regarding the values and suit of each card. A list containing all the cards is generated when the game starts. Then, these cards are shuffled using a random selection process and inserted into a new list. The program separates the cards used in the round and asks the user to choose whether to ask for a trick. A series of validations occur to decide the round winner based on the user's choice of cards. The process is repeated until one of the players reaches 12 points. This was a challenging project due to the nature of the chosen game and the many rules needed to identify the winner of each play and the match as a whole. The game developed was functional but did not meet all the rules existing in the original game. Figure 2 (b) presents the interface of the Blind trick game.



(a) (b)

Figure 2. (a) Hangman game with doll design printed. (b) Blind Trick game.

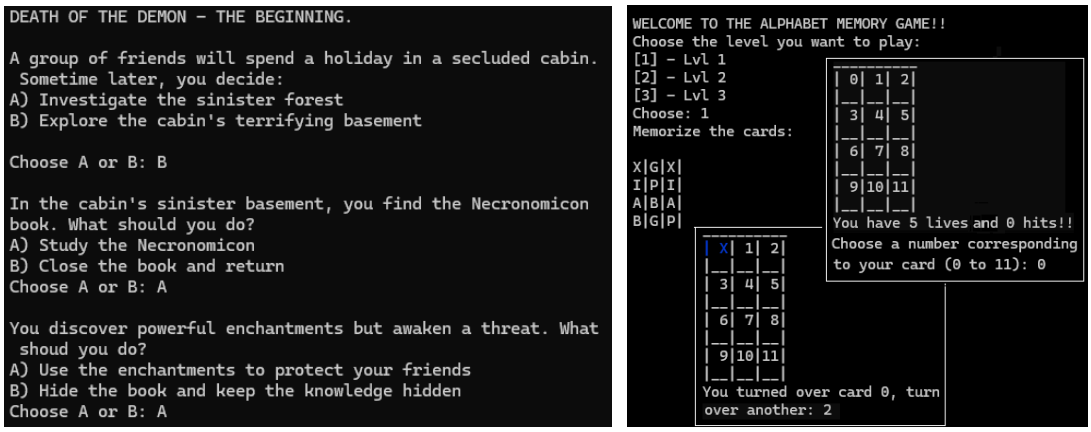
- **Tic Tac Toe**: This is a strategy game in which two players take turns placing pieces on a square board to form a horizontal, vertical, or diagonal line with their pieces. Two games were developed with this objective. In the first one, a linked stack was used to store the moves made in each play and the player identification, making it possible to undo the last moves or even the last round of the previous game. It is possible to play in pairs or with a robot (BOT), where the BOT will generate random numbers to play, validating the number and the field chosen. A second Tic Tac Toe project was developed, and in this case, a linked list was used to store the movements performed, inserting X's or O's in the positions that were informed. Figure 3 presents part of the interface of the Tic Tac Toe games developed.



(a) (b)

Figure 3. Partial interfaces of the two Tic Tac Toe games developed.

- **Interactive storytelling**: This game allows users to make choices and simultaneously tell a horror story, defined at runtime. The data structure used was the Binary Tree since each choice has two options, thus forming a story (Figure 4 (a)).
- **Memory game**: This game demands memorizing pairs of cards with alphabet letters printed randomly. The player selects positions to find the corresponding pairs by remembering their location. It has three challenging levels. For example, the first level has 12 cards and three lives in six possible hits. The player should match the six pairs of cards in at least three attempts. When the player misses a card, one life is reduced. The player wins if he identifies all the pairs without running out of lives. A Doubly Linked List was used to store symbol values in random positions. Figure 4 (b) presents the game interface.



(a) (b)

Figure 4. Interfaces of (a) Storytelling and (b) Memory games.

5.2. Second proposal - Games using data structures B-Tree or graph

In the second proposal of the project, students were required to develop projects using exclusively B-Tree or Graph, and the developed games are briefly described below.

- **Mining game**: In this game, a B-tree was used to store gemstones obtained by the player during mining activities. B-tree printing presents the inventory of what was already found, showing the gemstones in ascending order of accumulated value. Each mining activity has a cost, and some losses can occur related to damage to tools or insect infestation, for example. The print of the B-tree implemented in class is in a format rotated to the left so that the root node is on the left, the corresponding child nodes on the right, and so on recursively, as shown in Figure 5 (a). In this way, to print the items that the user has, starting from the items with the lowest value to the highest, the student removed the original indentation used in the print of the tree and then walked through the tree from left to right. Figure 5 (b) presents an example of the Mining game interface.

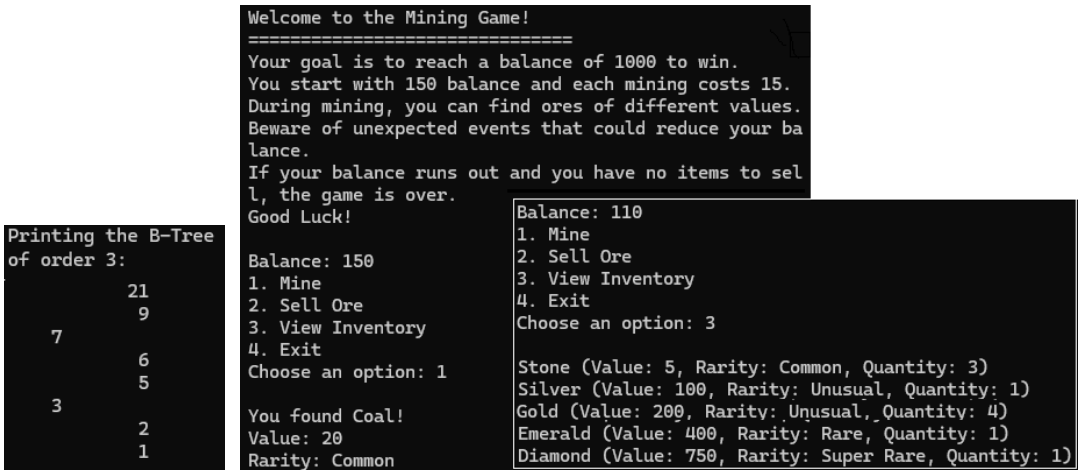
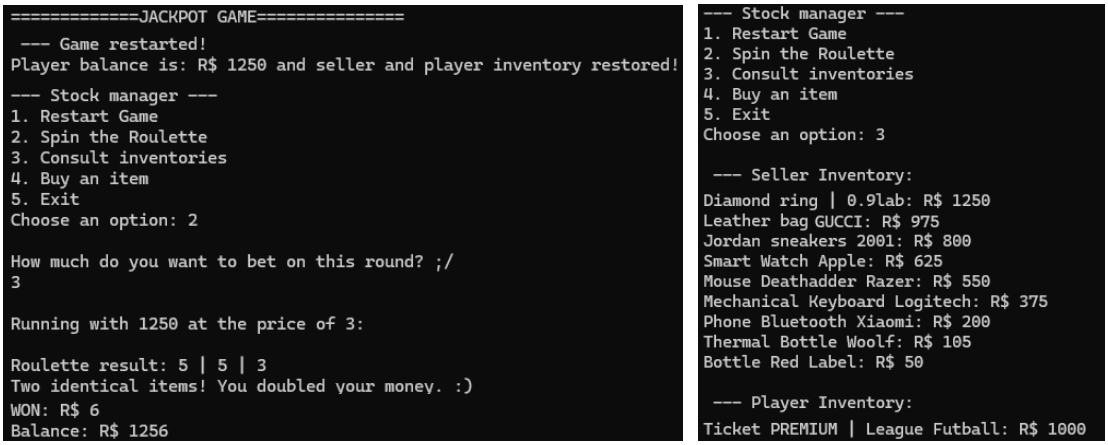


Figure 5. (a) B-tree rotated to the left - a node with two keys on the left and three child nodes on the right. (b) Mining game's interface and inventory.

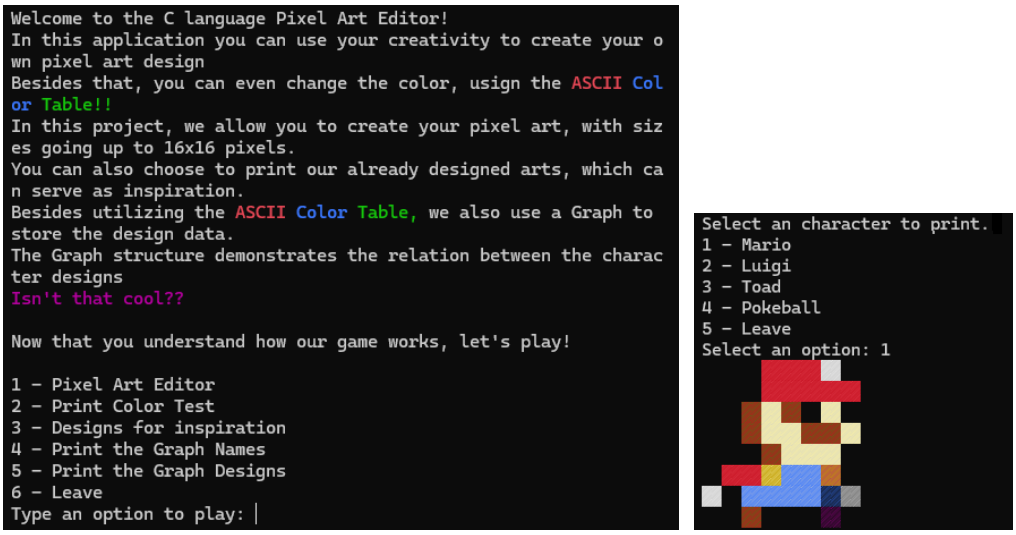
- **Jackpot game**: Jackpots are accumulated prizes common in casino games, such as slots, poker, and roulette. In this game, the user uses the menu option to restart the game and then receives money to realize bets utilizing the desired amount. Then, the player can realize a bet, spin the roulette wheel, and verify whether he received a prize or lost the amount bet. With the accumulated money, the player can buy available items for sale and make up his inventory in descending order. Using logic similar to the Mining game, a B-tree was used to store items the user purchased, as the player's inventory. Figure 6 shows the Jackpot game interface.
- **PixelArt Editor**: This program allows the editing of Pixel Art images via console, providing the colors and coordinates of each pixel to make up an image. The users can choose three sizes: 4x4, 8x8, and 16x16 pixels. The ASCII table was used to print the colored blocks in C, and eight standard colors were provided for the user to edit. It is also possible to print existing designs as inspiration. The program's author made some images of games's characters, which were stored in a graph, whose edges represent the relationship between the characters. Figure 7 (a) and



(a) (b)

Figure 6. (a) Jackpot game's interface and (b) Printing inventory stored in b-tree.

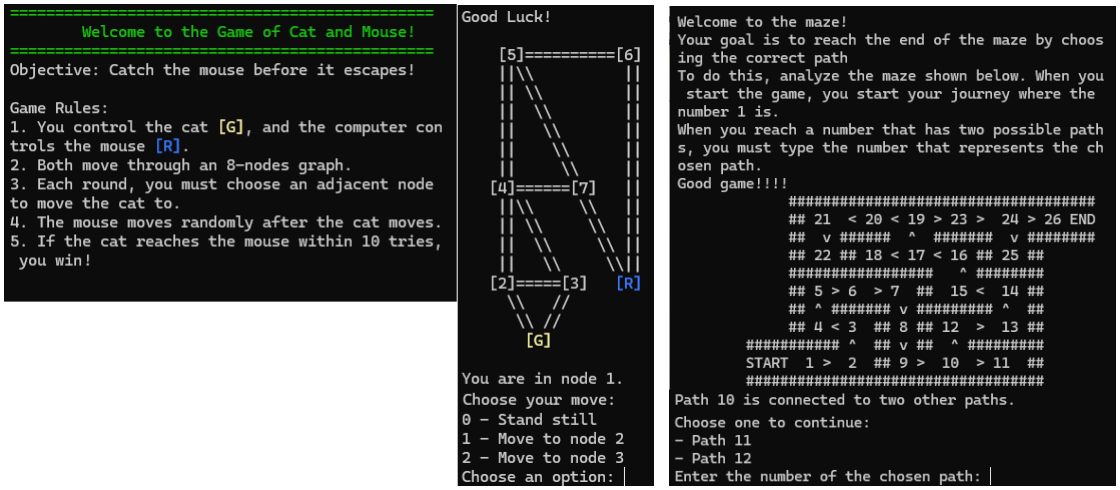
(b) present, respectively, an example of the Pixel Art Editor interface and one of the images suggested by the program, stored in a graph node.



(a) (b)

Figure 7. (a) PixelArt Editor's interface and (b) Example given by the program.

- **Cat and Mouse game**: This is a cat-and-mouse game based on a graph, where the player controls the cat, and the mouse is controlled by the computer, moving randomly. The cat and the mouse move through a graph, where the nodes represent the possible positions and the edges represent the paths between these positions. The game's goal is to catch the mouse before it can escape (Figure 8 (a)).
- **Maze game**: A graph was used in this maze game. Nodes represent the positions in the maze, and the program presents the available motion options to the user. The idea of the maze came when thinking about a directed graph in which a search is made using the Dijkstra algorithm, as it works in a way like a maze, where starting from an initial node, it will travel through the graph along the shortest possible path to reach the final node. Figure 8 presents the game's main interface.



(a) (b)

Figure 8. (a) Example of Cat and Mouse and (b) Maze game's interface.

- **StoryTelling**: A storytelling theme using graphs was chosen by five students this semester since that data structure inherently facilitates the organization of different available environments, represented by nodes and the connections between them through edges. The game interfaces generated are similar to the Figure 4 (a).

5.3. Evaluation of the activity proposal

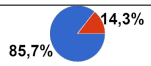
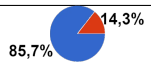
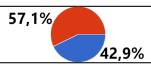
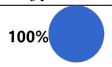
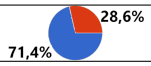
After the end of the second proposal, 22 participating students were invited via email to answer an electronic questionnaire to evaluate the activity. Table 1 presents the questions used and answers obtained from seven participants who agreed to participate through a free and informed consent form. Most respondents (85%) considered the approach positive, and 100% said they better understood the chosen data structure with the game's development. This indicates that it can be better explored in the teaching-learning process of different programming disciplines, aiming to obtain overall positive results for the course. 42.9% of students used Artificial Intelligence resources to assist in some aspects of development, which did not interfere with knowledge acquisition. Some challenges were mentioned, mainly related to applying the data structure in the game and controlling the interaction of two players. Interesting perceptions regarding the activity were related.

6. Discussion and Conclusion

Motivation is a critical element in the teaching-learning process, and using games for this purpose is an undeniably attractive option [Ascari et al. 2021]. Hoping to stimulate proactivity and make this process more enjoyable, this paper shares the results of a partial assessment applied to the "Data Structure 2" discipline of the TSAD course at the UTFPR, Pato Branco Campus, where a game-based approach was used. The students were challenged to build a game employing creativity to use some data structures studied, and the results were exciting. A summary of the functional games developed was described in this document. The perception of the students who evaluated the experience, despite the small number of respondents, was, in general, optimistic. The students presented their games to their colleagues and interacted with the games developed, providing a moment of socialization and knowledge sharing. The positive results observed encourage the use

of this approach. They might be helpful to educators interested in applying project-based assessments and contributing to the training of professionals in the Computer area. Furthermore, other teachers could take advantage of the creativity displayed by the students to generate game-based educational resources for teaching specific data structures.

Table 1. Questions used in the evaluation questionnaire for the proposed activity.

Question	**Answers** Yes No Maybe
1) Do you consider it beneficial to explore game development in assessments or final projects for subjects focused on teaching programming?	
2) In the case of the Data Structures 2 discipline, in which you developed a game as a final project using one of the data structures studied, did the fact that the topic was a game make any difference in your motivation or interest in completing the work?	
3) Did you use artificial intelligence resources, such as ChatGPT, to develop your game?	
4) If yes to the previous question, describe how the tool was used. *."For micro-bug fixes, theory research, and game testing for verification."* *. "To help define what could be done according to the activity specifications."* *."There was a method I hadn't noticed that fell into an infinite loop. With IA, it was easier to identify it."*	
5) Do you consider that the development of the proposed game contributed to a better understanding of how the chosen data structure works?	
6) Did you have any difficulties or challenges developing the final project?	
7) If yes to the previous question, describe the difficulties or challenges. *. "The main difficulty was in the design of a game based on a B-Tree or Graph data structure. However, I received the necessary help to continue with the project."* *. "I would like to point out a challenge: adapting a linked structure to the Minesweeper game, which was a requirement. I used a linked list that stored the mines randomly generated each time the game was played. However, this functionality could have been implemented more simply with another data structure."* *. "I asked IA to review my code for the above-mentioned difficulty, and the problem became clear."* *. "The hardest part is seeing the data structure in the game and, therefore, how to implement it."* *. "My project was very simple, it was a Super Trump, one of the problems I had was how to make it so that one player couldn't see the other's choices, but nothing like Google to help me kkk."*	
8) Do you have any other information you want to add or describe? *. "The fact that the theme of the final project was a game motivated me to work more intensely on the project because I was able to use my creativity and personal tastes to apply my technical knowledge, unlike a project with a defined theme, for example: "Pet Shop Management System" - a theme that does not interest me, and since I do not have the possibility of choosing, it would discourage me from developing it. In short, the theme being a game, in my opinion, is an excellent idea because it allows students to work with their imagination."* *. "Creativity made it possible to make the game without creating a single function since the functions created in tests and previous work did everything for a game to run."* *. "The development of software focused on games is in line with the experience of many students on the course since the number of students who play or have played virtual games is very high. In this sense, the work on game themes sparked my interest in developing something beyond what was just proposed, as I was very concerned with creating features that would facilitate and instruct the player."* *. "I believe that game development is the best way to learn any programming content, as it is cool to see the final result, and as the project is carried out and tested, you learn and have fun simultaneously."* *. "I believe that implementing a game is cool for development, but it is interesting with a game you have affinity or like; this human factor will make a total difference when you work."* *. "It was a challenge, unexpected, and caught me off guard, but it was a project I was very interested in. I think that for future classes, this could be seen as something that will keep their attention on the subject more."*	

7. Acknowledgments

The authors thank CAPES, CNPq, and the students who contributed to generating results.

References

Akar, S. G. M. and Altun, A. (2017). Individual differences in learning computer programming: A social cognitive approach. *Contemporary Educational Technology*, 8(3):195–213.

Ascari, R. E. S., Silva, L., and Pereira, R. (2021). Playing games via personalized gestural interaction. In *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 717–720. SBC.

Backes, A. R. (2023). *Algoritmos e estruturas de dados em linguagem C. (translated) Algorithms and data structures in C language*. LTC, Rio de Janeiro.

Barbosa, W. A. and Júnior, P. A. P. (2013). Um mapeamento sistemático sobre ferramentas de apoio ao ensino de algoritmo e estruturas de dados. (translated) A systematic mapping of tools to support teaching algorithms and data structures. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 24, page 406.

Barros, A. T. and Xavier, K. A. (2022). Jogos didáticos para o ensino de zoologia: Uma revisão bibliográfica. (translated) Educational games for teaching zoology: A bibliographic review. *Revista Electrónica de Enseñanza de las Ciencias*, 21(2):356–373.

Brasil, M. (2001). Diretrizes curriculares de cursos da área de computação e informática. (translated) Curricular guidelines for courses in the area of computing and information technology. *MEC–SESU (Secretaria de Educação Superior). CEEInf–Comissao de Especialistas de Ensino de Computação e Informática. Brasília*.

de Souza, J. A., Feliciano, S. M., and Teles, R. N. (2023). Gamificação: uma agordagem inovadora no ensino da matemática. (translated) Gamification: an innovative approach to teaching mathematics. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 9(9):1969–1978.

Draft, S. (2013). Computer science curricula 2013. *ACM and IEEE Computer Society, Incorporated: New York, NY, USA*.

Fardo, M. L., Anastácio, B. S., Ramos, D. K., and Mattar, J. (2022). O uso de jogos no processor de ensino e aprendizagem. (translated) the use of games in the teaching and learning process. *Revista Interacções*, 18(63):1–4.

Feitosa, A. C. (2023). Iniciativas para o ensino de estruturas de dados em universidades brasileiras: uma revisão sistemática da literature. (translated) Initiatives for teaching data structures in brazilian universities: a systematic literature review. Final paper, Centro Multidisciplinar de Pau dos Ferros.

Fernandes, K. T. (2021). *Game Criativo: desenvolvendo habilidades de pensamento computacional, leitura e escrita através da criação de jogos. (translated) Creative Game: developing computational thinking, reading and writing skills through game creation*. PhD thesis, Universidade Federal do Rio Grande do Norte.

Fernandes, K. T., Aranha, E. H. d. S., Lucena, M. J. N., Fernandes, G. L. d. S., and Araújo, W. (2022). Do texto ao jogo: uma análise da produção textual motivada pela criação de jogos digitais em sala de aula. (translated) From text to game: An analysis of textual production motivated by the creation of digital games in the classroom. In *Anais do XXVIII Workshop de Informática na Escola*, pages 01–13. SBC.

Gurbuz, S. C. and Celik, M. (2022). Serious games in future skills development: A systematic review of the design approaches. *Computer Applications in Engineering Education*, 30(5):1591–1612.

Ishikawa, E. and Ralha, C. (2015). Ensino de sistemas de informação em cursos de computação: relato de experiência com uso de abordagem prática em tic. (translated) Teaching information systems in computer science courses: experience report using a practical approach in ict. In *Anais do XI Simpósio Brasileiro de Sistemas de Informação*, pages 579–586. SBC.

Prensky, M. (2021). *Aprendizagem baseada em jogos digitais. (translated) Digital game-based learning*. Editora Senac São Paulo.

Sedgewick, R. and Wayne, K. (2011). *Algorithms*. Addison-wesley professional.

Silva, G. C., Ré, R., Kawamoto, A., and Schwerz, A. (2011). Uma experiência na aplicaçao de práticas de apoio no ensino-aprendizado de algoritmos. (translated) An experience in the application of support practices in the teaching-learning of algorithms. In *Anais do XVII Workshop de Informática na Escola*, pages 1378–1381. SBC.