# Gamification method using Minecraft for training future teachers of computer science

Andrii V. Riabko[1], Tetiana A. Vakaliuk[2,3,4,5], Oksana V. Zaika[1], Roman P. Kukharchuk[1] and Inesa V. Novitska[5]

[1]*Oleksandr Dovzhenko Hlukhiv National Pedagogical University, 24 Kyivska Str., Hlukhiv, 41400, Ukraine*
[2]*Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine*
[3]*Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine*
[4]*Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine*
[5]*Academy of Cognitive and Natural Sciences, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine*
[5]*Zhytomyr Ivan Franko State University, 40, Velyka Berdychivska Str., Zhytomyr, 10008, Ukraine*

**Abstract**

The article examines the integration of the content of the educational program from "Informatics: programming practicum" of training future computer science teachers in a higher educational institution to the Minecraft education game process. The peculiarities and possibilities of Minecraft education regarding the gamification of the educational process, which are confirmed in many studies and publications, are considered. Methods of integration of individual topics from the programming workshop into the Minecraft gameplay are proposed. Examples of tasks, projects and creative tasks are given. The results of the research-experimental verification of the methods of integration of programming learning into the Minecraft game process are described, which confirmed its effectiveness based on the application of the non-parametric Mann-Whitney test. The perspective and wide possibilities of integration of other educational subjects into the gameplay of Minecraft are noted.

**Keywords**

gamification, Minecraft, learning, programming, building design, testing, integration

## 1. Introduction

In times of digital transformation of all areas of education, the issue of exploring the possibilities of gamification of education is gaining great interest. Many factors contributed to this, including the global pandemic caused by Covid-19 [1, 2] and Russian aggression against Ukraine [3]. One area of exploration lies in developing subject-specific curricula that seamlessly integrate some games, for example Minecraft's, into the learning process. Minecraft offers opportunities for students to develop technological skills essential for computer science teaching. They can learn programming languages such as Java through Minecraft's modding capabilities, allowing them to customize game mechanics and create interactive elements. Additionally, students can explore concepts like network protocols and cybersecurity within the game, gaining practical experience with technology-related topics. Students develop critical thinking and engineering skills vital for computer science teaching by working collaboratively on construction projects and overcoming design challenges.

Minecraft can be used to reinforce mathematical concepts relevant to computer science education. Students can explore geometric shapes, spatial reasoning, and mathematical modelling through construction projects and game mechanics. Additionally, they can develop computational thinking skills by solving mathematical puzzles and challenges within the game environment.

*DigiTransfEd 2024: 3rd Workshop on Digital Transformation of Education, co-located with the 19th International Conference on ICT in Education, Research, and Industrial Applications (ICTERI 2024), September 23-27, 2024, Lviv, Ukraine*

ryabko@meta.ua (A. V. Riabko); tetianavakaliuk@gmail.com (T. A. Vakaliuk); ksuwazaika@gmail.com (O. V. Zaika); kyxap4yk1@ukr.net (R. P. Kukharchuk); inesaserbin2601@gmail.com (I. V. Novitska)
https://acnsci.org/vakaliuk/ (T. A. Vakaliuk)
0000-0001-7728-6498 (A. V. Riabko); 0000-0001-6825-4697 (T. A. Vakaliuk); 0000-0002-8479-9408 (O. V. Zaika); 0000-0002-7588-7406 (R. P. Kukharchuk); 0000-0003-0780-0580 (I. V. Novitska)

## 2. Theoretical background

Many scholars have considered the issue of ICT [4, 2, 1] and gamification of education [5, 6, 7], and we will consider the key issues they have raised. C. Richardson's book [6] and D. Whale & M. O'Hanlon's book [8] provide a foundation for incorporating playback into Minecraft gameplay. Both books guide readers in developing programs using the Python programming language to control events within the Minecraft world.

We can consider many applications of Minecraft Education in the educational process from various educational disciplines [7]. The findings indicate that students' perceived learning outcomes were highly positive, and their overall learning experience was engaging and stimulating [9, 10]. It has been established that children's general media literacy increases while playing Minecraft [11].

The analysis reveals that students are better able to grasp intricate scientific concepts when presented through the game's interactive environment, enabling them to experience these concepts firsthand [5].

Minecraft Education Edition is a version of the popular video game Minecraft specifically designed for educational purposes. It is used in schools and other educational institutions worldwide to teach various subjects, including math, science, history, and language arts [12]. Minecraft Education Edition features a variety of tools and resources that make it easy for educators to create engaging and interactive learning experiences. Students can use Minecraft Education Edition to build virtual worlds, solve problems, and collaborate with their peers [13].

The most important aspects of Minecraft are student-centered learning and game-based learning [14]. Digital games are not just for entertainment; they can also be powerful tools for learning. The book "Impassioned learning and Minecraft. In Serious Play" explores how digital games can engage and challenge students, present complex representations and experiences, foster collaborative and deep learning, and enable curricula that connect with today's youth [15].

C. Tessler et al. present a groundbreaking lifelong learning system capable of effectively reusing and transferring knowledge from one task to another while preserving its established knowledge base. This knowledge transfer is facilitated by acquiring reusable skills, termed Deep Skill Networks, acquired by tackling tasks within Minecraft, a complex video game environment. These deep skill networks are subsequently integrated into a novel hierarchical deep reinforcement learning network (H-DRLN) architecture [16].

W. H. Guss et al. present MineRL, a large-scale, simulator-paired dataset of human demonstrations. The dataset comprises over 60 million automatically annotated state-action pairs across various related tasks in Minecraft, a dynamic, 3D, open-world environment. They introduce a novel data collection scheme that enables the continuous introduction of new tasks and the acquisition of comprehensive state information suitable for a wide range of methods. Additionally, they demonstrate the complexity of the Minecraft domain and the potential of MineRL in developing techniques to address critical research challenges within it [17].

H. C. Lane et al. advocate for the relevance of Minecraft, an open-world, exploratory, and generative game, for both educators and learning science researchers. Gameplay fosters the acquisition of diverse skills and knowledge applicable to academic and real-world settings. Minecraft likely serves as an initial exposure to fundamental STEM concepts, such as engineering, agriculture, and biology. They emphasize the need to delve into the game's influence on children's STEM competency development, their perceptions of STEM fields, and their acquisition of metacognitive skills like planning and reflection [18].

O. Alawajee et al. propose game-based learning (GBL) as a promising and engaging approach to contemporary education. Minecraft, with its socially interactive and cooperative open-world gameplay, stands out as a potential tool for educational purposes. Research has demonstrated Minecraft's effectiveness in enhancing student motivation, language development, and academic learning across subjects like science and history [19].

J. García-Fernández et al. evaluated the potential of Minecraft, a popular video game, to serve as an effective tool for communicating and promoting engagement with culturally built environments. Through an analysis of videogames ability to motivate, immerse, and represent reality, Minecraft emerged as one of the most promising solutions. The researchers assessed Minecraft's capacity against established criteria of immersion, motivation, and fidelity to simulation, demonstrating its effectiveness in conveying built heritage environments [20].

The study by R. Andersen et al. revealed that incorporating Minecraft into subject-specific learning effectively fostered collaborative learning processes among students. Integrating Minecraft into math classes effectively promotes the development of 21st-century skills. Moreover, the study employs a novel research methodology that combines social network analysis (SNA) and intelligent automation (IA) to analyze the use of Minecraft for acquiring subject-specific skills [21].

A. Narayan-Chen et al. propose the development of interactive agents capable of collaborating with humans to solve tasks in realistic scenarios. They introduce a Minecraft-based collaborative building task where one player (A, the architect) is presented with a target structure and must guide the other player (B, the builder) in constructing it. They present the Minecraft Dialogue Corpus to facilitate this interaction, a comprehensive collection of conversations and game logs [22].

An indisputable advantage of Minecraft is the ability to create mods—special modifications to expand the game's capabilities. This is how an extension for learning Python programming was first made. One strategy to balance rising gamer expectations and development pressures is cultivating an active modding community around the game. In their paper, D. Lee et al. present the findings of an empirical study that examined 1,114 popular and 1,114 unpopular Minecraft mods from CurseForge, a prominent distribution platform for Minecraft mods. Their analysis revealed that popular mods typically feature high-quality descriptions and actively encourage community involvement [23].

Leveraging the widespread popularity of video games among teenagers, Minecraft Education (Minecraft Edu) emerges as a promising game-based learning tool (GBL) for science education. M. Nkadimeng et al. have demonstrated the effectiveness of Minecraft Edu in enhancing students' understanding of atomic structure [24].

V. Panja et al. investigated the potential of Minecraft Education Edition's chemistry features to foster an engaging and effective learning environment. They examined how Minecraft's game mechanics and Education Edition's chemistry-specific features could address the scarcity of effective game-based learning solutions [25].

Minecraft holds immense educational potential for fostering spatial thinking, imagination, and design skills. However, M. Opmeer et al. recommends introducing students to traditional design materials, such as paper and pencil, before delving into Minecraft. This initial exposure to traditional design tools can help students develop a solid foundation in design principles, making the transition to Minecraft's block-based building environment less restrictive [26].

## 3. Research methodology

The purpose of the study is to study the possibilities of integrating the content of the educational program from "Informatics: Programming Workshop" to train future informatics teachers in a higher educational institution in the game process of learning Minecraft.

Research objectives: 1) to consider the peculiarities and possibilities of Minecraft training about the gamification of the educational process in this educational discipline; 2) to offer practical ways of integrating individual topics from the programming practice into the Minecraft gameplay; 3) develop and implement tasks, projects and creative tasks; 4) perform a research-experimental check of the methodology of integration of programming learning into the Minecraft gameplay.

The study employs an experimental with a control group and an experimental group, each consisting of 12 students. The control group receives traditional programming instruction, while the experimental group performs programming tasks within Minecraft.

The experimental verification involves collecting data on student performance and learning outcomes. This includes assessing students' ability to develop programs to solve standard problems from the curriculum's topics and measuring their engagement and understanding of programming concepts.

The effectiveness of integrating programming learning into Minecraft gameplay is analyzed using the non-parametric Mann-Whitney test. This statistical analysis compares the performance of the control group and the experimental group to determine if there are significant differences in learning outcomes.

## 4. Results

In our study, the curriculum was developed for the disciplines "Informatics: programming practicum" of the "Secondary Education (Informatics)" educational program with the integration of learning into the Minecraft gameplay. Let us present the characteristics of the discipline. The scope of the discipline is 5 credits (150 hours), of which laboratory classes – 46 hours, independent work 104 hours. The subject of study is the practical mastering of the basics of programming using the language of modern programming languages (using the Python language as an example).

The goal of the discipline is for students to acquire practical skills in creating software to solve typical problems; students acquiring practical skills in creating application programs using a high-level programming language, development of logical thinking, development of technical culture, development of analytical thinking, development of scientific skills, logical and algorithmic thinking.

The content of the educational discipline includes questions: 1) Input (output) of data; 2) Operator if; 3) Mathematical calculations; 4) Operators for; 5) String data; 6) Operator while ; 7) Lists; 8) Function and recursion; 9) Arrays; 10) Multiple data types; 11) Dictionaries; 12) Construction of graphic primitives.

Control measures – testing, control work. The knowledge and skills acquired during the study of the discipline are used in the study of the disciplines of information and communication orientation and the school course of informatics and methods of teaching informatics.

An example of the integration of the educational program into the Minecraft education gameplay is given below (1 and 2).

In the first lesson, we explain to students what Minecraft Education is and how it is used in education. We show students who are not familiar with the game how to play. We allow students to play Minecraft Education to learn the game's basics. Consider installing and configuring the game, Java, and the necessary mods. We offer them interesting tasks that lead them to think about the effectiveness of using framing in Minecraft education gameplay. We use various resources to introduce students to Minecraft Education, such as websites, videos and tutorials.

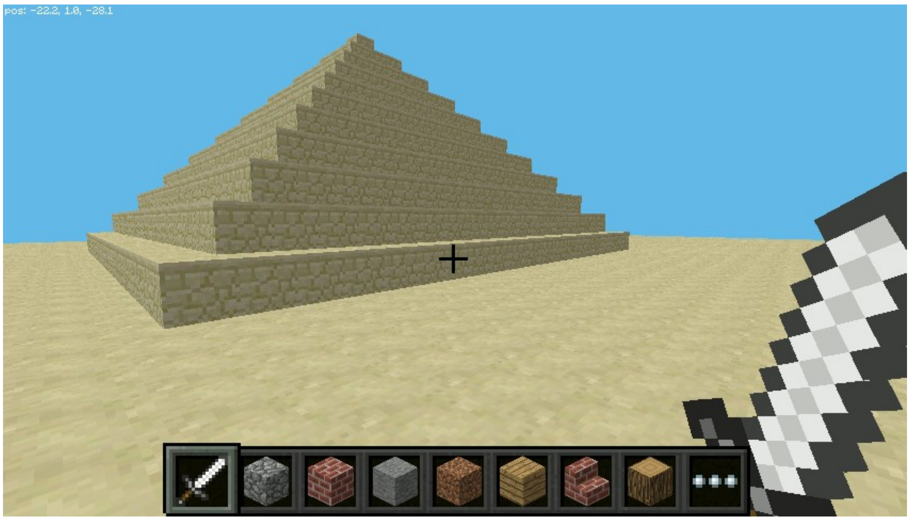


**Figure 1:** Pyramid in Minecraft education.

Examples of rapid construction especially impress students. Their experience convinces them that the ability to program is a skill that gives an undeniable advantage over players who do not know how to program.

Let us consider an example of the construction of a pyramid.

**Table 1**
Integration of the curriculum of the discipline “Informatics: programming practicum” into the Minecraft game process education (part 1).

| № | Topic | Examples of tasks | Example of an educational project |
|---|---|---|---|
| 1 | Input (output) of data | Create a program to track player movement and output the data to a file. Develop a system to collect information about the types of blocks players are placing. Design a mechanism to record the frequency of item usage. | Design and build a data collection system for Minecraft Education. This system could collect data on various aspects of the game, such as player location, block placement, and item use. The data could then be analyzed to identify trends and patterns in player behavior. |
| 2 | Operator if | Create a program to control the opening and closing of a door based on the presence of a player. Develop a system to activate a trap when a player steps on a pressure plate. Design a mechanism to display a reward when a player solves a puzzle. | Implement a conditional branching system in Minecraft Education. This system could be used to control various aspects of the game world, such as the behavior of NPCs, the activation of traps, and the appearance of rewards. |
| 3 | Mathematical calculations | Create a program to add, subtract, multiply, and divide two numbers. Develop a system to solve quadratic equations. Design a mechanism to calculate the volume of various shapes. | Build a mathematical calculator in Minecraft Education. This calculator could perform basic operations such as addition, subtraction, multiplication, and division. It could also be used to solve more complex equations. |
| 4 | Operator for | Create a program to build a pyramid block by block. Develop a system to collect all the wood from a forest. Design a mechanism to craft a stack of diamonds from rough diamonds. | Implement a loop system in Minecraft Education. This system could be used to automate repetitive tasks, such as building structures, collecting resources, and crafting items. |
| 5 | String data | Create a program to display a message when a player enters a specific area. Develop a system to trigger a dialogue with an NPC when a player clicks on it. Design a mechanism to display clues or instructions to solve a puzzle. | Design and build a text-based adventure game in Minecraft Education. This game could involve exploring different areas, interacting with NPCs, and solving puzzles. |

```
import mcpi.minecraft as minecraft
import mcpi.block as block

def build_pyramid(mc, x, y, z, base_size, block_type):
# Set the starting position of the pyramid
start_x = x - base_size // 2
start_y = y
start_z = z - base_size // 2

# Build the pyramid layer by layer
for level in range(base_size):
# Calculate the current layer width
width = base_size - level * 2

# Build the current layer
for offset_x in range(width):
for offset_z in range(width):
mc.setBlock(start_x + offset_x, start_y + level, start_z +
```

**Table 2**
Integration of the curriculum of the discipline “Informatics: programming practicum” into the Minecraft game process education (part 2).

| № | Topic | Examples of tasks | Example of an educational project |
|---|---|---|---|
| 6 | Operator while | Create a program to keep a door open as long as a player is standing on a pressure plate. Develop a system to continuously check for the presence of a specific item and light up a redstone lamp when it is found. Design a mechanism to keep a mob chasing a player until it catches them. | Implement a while loop system in Minecraft Education. This system could be used to monitor conditions in the game world and trigger actions accordingly. |
| 7 | Lists | Create a program to add items to a grocery list. Develop a system to mark items as purchased as they are found. Design a mechanism to display the remaining items on the list. | Build a grocery shopping list system in Minecraft Education. This system could allow players to create lists of items they need to purchase and track their progress in acquiring them. |
| 8 | Function and recursion | Create a function to build a house with different parameters for size and style. Develop a function to generate a random number between two specified values. Design a recursive function to calculate the factorial of a given number. | Implement a function system in Minecraft Education. This system could be used to encapsulate reusable code blocks, making programs more modular and efficient. |
| 9 | Arrays | Create a program to store items in an array and display them on a scoreboard. Develop a system to retrieve items from an array based on their index. Design a mechanism to sort items in an array alphabetically. | Build a storage system for items in Minecraft Education. This system could allow players to store and retrieve items using arrays, keeping their inventories organized and accessible. |
| 10 | Multiple data types | Create a program to store player names, inventory items, and scores in a database. Develop a system to calculate the average | Implement a data type system in Minecraft Education. This system could allow players to store and manipulate different types of data, such as numbers, strings, and booleans. |

```
offset_z, block_type)

# Create a Minecraft connection
mc = minecraft.Minecraft.create()

# Set the desired pyramid parameters
base_size = 5
block_type = block.GOLD_BLOCK

# Build the pyramid at the player’s position
player_pos = mc.player.getPos()
mc.postToChat("Building a pyramid...")
build_pyramid(mc, player_pos.x, player_pos.y, player_pos.z, base_size, block_type)
mc.postToChat("Pyramid built!")
```

This code will build a pyramid with a base size of 5 blocks using gold blocks. You can change the base size and block type to your liking.

Another example. This code will create a simple maze in Minecraft with dimensions 10x10x1. The maze will be generated randomly, and the starting position will be at the origin.

```
import mcpi.minecraft as minecraft
```

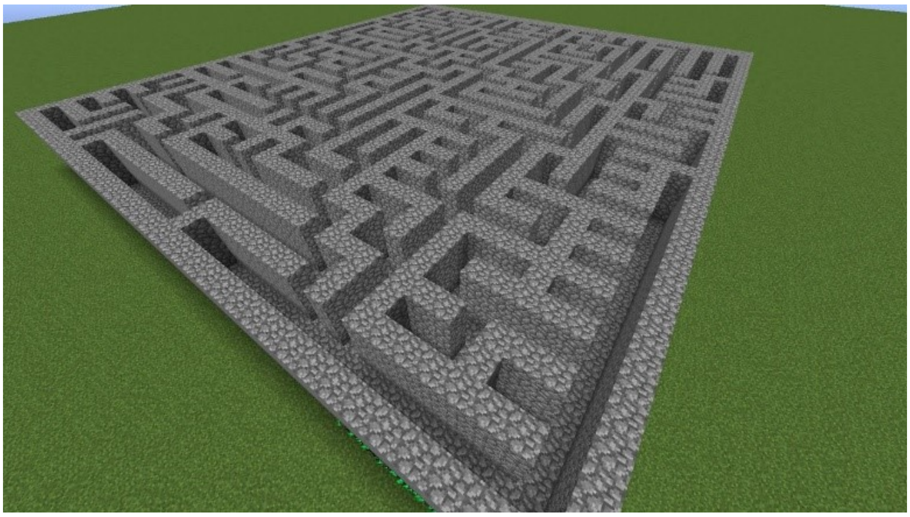


**Figure 2:** Maze in Minecraft education.

```
import mcpi.block as block

# Connect to Minecraft server
mc = minecraft.Minecraft.create()

# Set up variables
width = 10
height = 10
start_pos = mc.player.getPos()

# Create a list to store the maze
maze = []

# Initialize the maze with all walls
for x in range(width):
maze.append([])
for y in range(height):
maze[x].append(block.STONE)

# Define a function to check if a position is within the maze boundaries
def is_within_bounds(pos):
x, y, z = pos
return 0 <= x < width and 0 <= y < height and 0 <= z < 1

# Define a function to carve a passage from the start position
def carve_passage(start_pos):
# Get the current position
pos = start_pos.clone()

# Carve a passage while staying within the maze boundaries
while True:
# Choose a random direction
direction = random.choice([block.AIR, block.AIR, block.AIR, block.AIR])
```

```
# Move in the chosen direction
if direction == block.AIR:
new_pos = pos.clone()
new_pos.add(-1, 0, 0)
if is_within_bounds(new_pos):
maze[new_pos.x][new_pos.y] = block.AIR
pos = new_pos
elif direction == block.AIR:
new_pos = pos.clone()
new_pos.add(1, 0, 0)
if is_within_bounds(new_pos):
maze[new_pos.x][new_pos.y] = block.AIR
pos = new_pos
elif direction == block.AIR:
new_pos = pos.clone()
new_pos.add(0, -1, 0)
if is_within_bounds(new_pos):
maze[new_pos.x][new_pos.y] = block.AIR
pos = new_pos
else:
new_pos = pos.clone()
new_pos.add(0, 1, 0)
if is_within_bounds(new_pos):
maze[new_pos.x][new_pos.y] = block.AIR
pos = new_pos

# Call the function to carve a passage from the start position
carve_passage(start_pos)

# Place the maze blocks in the Minecraft world
for x in range(width):
for y in range(height):
for z in range(1):
mc.setBlock(x, y, z, maze[x][y])
```

It is common knowledge that the basic training of pupils and students in IT and programming can be very different. For this purpose, creative tasks have been prepared for students with a fast pace of knowledge acquisition, including creating.

We encourage such students to use Python to create mods or plugins for Minecraft. This task is more complex but can also be very rewarding. Mods and plugins can add new features to the game, such as new blocks, items, or game mechanics.

Let's give examples of creative tasks of increased complexity in Python programming in the educational game environment Minecraft (3):

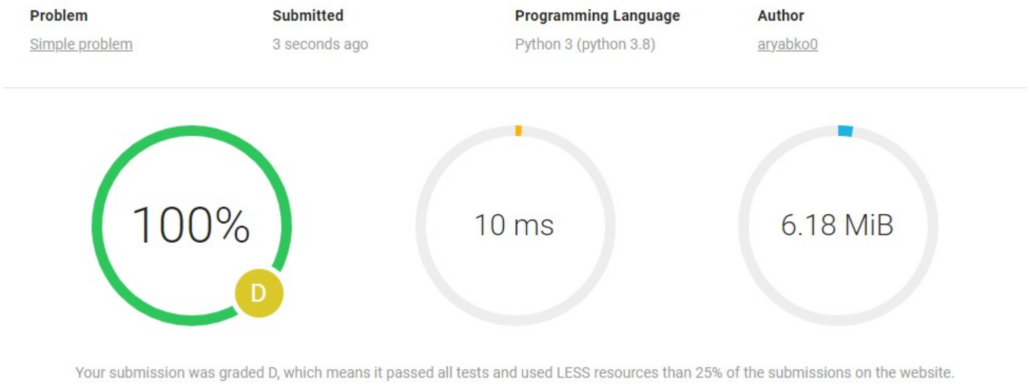
## 5. Experiment

The experimental verification of the results of implementing the specified integrated gaming curriculum in programming was carried out in the control group (12 students) and the experimental group (12 students).

The measuring tool was a particular test using the automated verification system of problems on the website https://www.eolymp.com, which consisted of developing programs for solving standard problems from the curriculum topics listed in 1.

**Table 3**
Examples of creative tasks of increased complexity in Python programming in the educational game environment Minecraft.

| Examples of creative tasks | Description |
|---|---|
| Procedural World Generation | Create a Python program to procedurally generate a Minecraft world with diverse biomes, terrain features, and structures. Explore algorithms like Perlin noise to generate realistic landscapes and incorporate conditional logic to place blocks based on elevation and biome type. |
| Interactive NPC Characters | Develop Python-based NPC (non-player character) characters that can interact with players in various ways. Implement dialogue trees, quests, and conditional interactions to create engaging and immersive experiences. Utilize text-based or voice-based interactions to enhance the gameplay. |
| Dynamically Changing Environments | Design a Python program that dynamically modifies the Minecraft environment based on player actions or in-game events. Implement mechanisms to trigger environmental changes like day/night cycles, weather patterns, or natural disasters. Utilize game events and timers to control the timing and behaviour of these dynamic elements. |
| Educational Minigames | Create interactive minigames in Minecraft using Python to teach various educational concepts. Develop engaging gameplay mechanics, incorporate scoring systems, and provide feedback to enhance learning. Utilize Minecraft's diverse block types and items to represent educational concepts visually. |
| Collaborative Programming Challenges | Design collaborative programming challenges that require players to work together to solve complex programming problems within the Minecraft environment. Utilize shared code repositories and in-game communication tools to facilitate collaboration. Implement scoring mechanisms and leaderboards to encourage teamwork and competition. |
| Utilize Python to create artistic installations or narrative experiences within Minecraft | Generate intricate patterns, construct sculptures or craft interactive storybooks. Explore Minecraft's creative tools and block variations to express artistic ideas and convey storytelling elements. |



| Problem | Submitted | Programming Language | Author |
|---|---|---|---|
| Simple problem | 3 seconds ago | Python 3 (python 3.8) | aryabko0 |

100% D

10 ms

6.18 MiB

Your submission was graded D, which means it passed all tests and used LESS resources than 25% of the submissions on the website.

**Figure 3:** An example of the result of an automated task check.

The system of automated verification of tasks gives the result in the form of the result of passing 10 control tasks on a 100-point scale (figure 3). In addition, the originality and creativity of the proposed solution to the problem with the involvement of experts (department teachers) were evaluated. In addition, the task completion time is measured.

10 standard tasks were proposed for each topic according to 1. Let's give an example of a task for topic 2 List.

Create a Python program allowing users to create and manage a grocery list. The program should:

1. Prompt the user to enter an item to add to the grocery list.
2. Add the entered item to the grocery list.
3. Allow the user to continue adding items or exit the program.
4. Display the complete grocery list when the user exits the program.

Grocery List: apples, milk, bread.

Solution (Python):

```
grocery_list = []
while True:
item = input("Enter an item to add to the grocery list: ")
grocery_list.append(item)
choice = input("Add another item (y/n)? ")
if choice.lower() != "y":
break
print("\nGrocery List:")
for item in grocery_list:
print(item)
```

An example of a task from topic 6 While loop operators.

Write a Python program that generates a random password. The password should be at least eight characters long and contain at least one uppercase letter, one lowercase letter, one digit, and one unique character. Example Output: Password: d35%Tr0ub4dor.

Solution:

```
import random
import string
def generate_password():
password = ""
while len(password) < 8:
character = random.choice(string.ascii_letters + string.digits +
string.punctuation)
if character.isupper():
password += character
elif character.islower():
password += character
elif character.isdigit():
password += character
elif character in string.punctuation:
password += character
return password
# Example usage
password = generate_password()
print("Password:", password)
```

In the study, the control and experimental groups were subjected to differing methodologies to assess the integration of programming learning into the Minecraft game process.

For the control group, the educational approach followed a more traditional format, centred around lectures, theoretical coursework, and practical exercises within the confines of a conventional classroom setting. Students might have engaged with programming concepts through traditional teaching methods such as textbooks, written assignments, and some hands-on computer coding exercises. The experimental group experienced a innovative approach to learning, integrating Minecraft as a platform for exploring and applying programming concepts.

n the result, was ranked the grades and wrote out separate ranks for sample A (control) and sample B (experimental).

Was find $\sum RA$, which is the sum of ranks for students in sample A, and respectively $\sum RB$ for sample B. For sample A, the sum of ranks is $\sum RA = 106$. For sample B, the sum of ranks is $\sum RB = 194$ (4).

**Table 4**
Results of final testing.

| Scored points in the control group. | Scored points in the experimental group. | Control group (ranks) | Experimental group (ranks) |
|---|---|---|---|
| 758 | 678 | 18 | 17 |
| 339 | 677 | 6 | 16 |
| 446 | 445 | 8 | 7 |
| 672 | 807 | 15 | 21 |
| 555 | 962 | 11 | 24 |
| 287 | 584 | 4 | 12 |
| 795 | 548 | 20 | 10 |
| 244 | 447 | 3 | 9 |
| 134 | 956 | 2 | 23 |
| 109 | 663 | 1 | 14 |
| 333 | 870 | 5 | 22 |
| 662 | 777 | 13 | 19 |
| | | $\sum RA = 106$ | $\sum RB = 194$ |
| | | UA = 116 | UB = 28 |

Then the value of U for sample A was calculated:

$$U_A = n_A \cdot n_B + \frac{n_A(n_A+1)}{2} - \sum R_A \tag{1}$$

For sample B:

$$U_B = n_A \cdot n_B + \frac{n_B(n_B+1)}{2} - \sum R_B \tag{2}$$

For sample A, $U_A = 116$. For sample B, $U_B = 28$ (4).

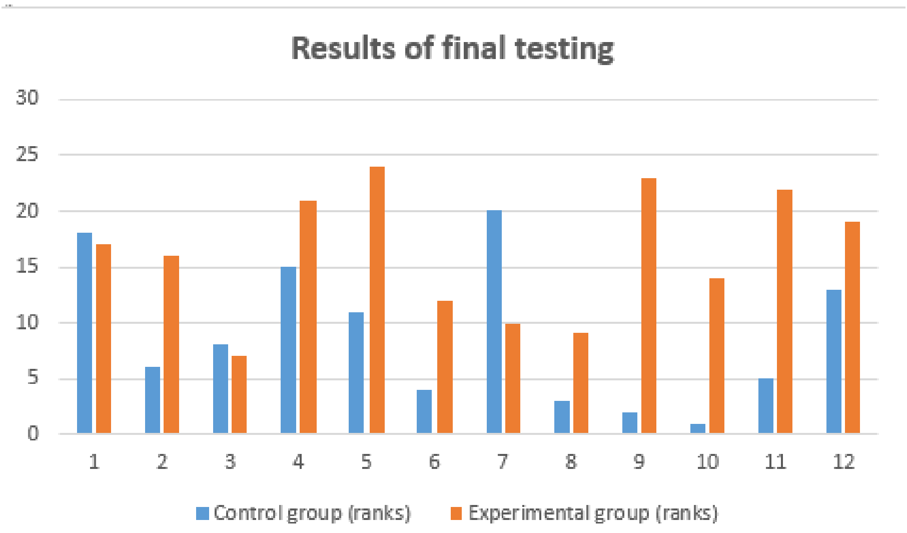


**Figure 4:** Results of final testing.

Mann-Whitney U test is 28. The critical value of the Mann-Whitney U test for a given number of compared groups is 37.

$28 \leq 37$, therefore the differences in the level of the trait in the compared groups are statistically significant ($p < 0.05$). We can reject H0 because the data provide sufficient evidence to conclude that there is a significant difference between the two samples. Hypothesis H0 is rejected at the significance level of 0.05 (or 5%). This indicates that the scores in the control group are systematically lower than in the experimental group.

Research-experimental verification of the developed methodology for integrating the curriculum of the discipline "Informatics: programming practicum" into the Minecraft game process proved its effectiveness and is recommended for implementation in the educational process.

## 6. Conclusions

The study underscores the potential of integrating educational content into popular gaming platforms like Minecraft to enhance learning outcomes, particularly in challenging subjects such as programming. The findings of the study demonstrate that by leveraging the Minecraft, educators can effectively engage students in learning complex programming concepts. The experimental verification of integrating programming curriculum content into Minecraft gameplay confirms its efficacy in enhancing student understanding and retention of programming principles.

These results suggest that Minecraft's widespread popularity and versatile educational applications offer a promising avenue for teaching various disciplines within computer science teacher training programs.

In conclusion, successfully integrating programming curriculum content into Minecraft gameplay underscores the potential of gamified learning environments to revolutionize education. By leveraging the engaging and immersive qualities of popular games like Minecraft, educators can create dynamic and effective learning experiences that empower students to master complex subjects and prepare for future challenges in the field of computer science.

## References

[1] D. Dymek, S. Didkivska, M. Grabowski, G. Paliwoda-Pekosz, T. A. Vakaliuk, Students' perception of online learning during the COVID-19 pandemic: Polish and ukrainian perspectives, in: E. Ziemba, W. Chmielarz, J. Watróbski (Eds.), Information Technology for Management: Approaches to Improving Business and Society - AIST 2022 Track and 17th Conference, ISM 2022, Held as Part of FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022, Extended and Revised Selected Papers, volume 471 of *Lecture Notes in Business Information Processing*, Springer, 2022, pp. 127–147. doi:10.1007/978-3-031-29570-6_7.

[2] I. S. Mintii, T. A. Vakaliuk, S. M. Ivanova, O. A. Chernysh, S. M. Hryshchenko, S. O. Semerikov, Current state and prospects of distance learning development in Ukraine, in: S. H. Lytvynova, S. O. Semerikov (Eds.), Proceedings of the 4th International Workshop on Augmented Reality in Education (AREdu 2021), Kryvyi Rih, Ukraine, May 11, 2021, volume 2898 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 41–55. URL: https://ceur-ws.org/Vol-2898/paper01.pdf.

[3] S. O. Semerikov, T. A. Vakaliuk, I. S. Mintii, S. O. Didkivska, Challenges facing distance learning during martial law: results of a survey of ukrainian students, Educational Technology Quarterly 2023 (2023) 401–421. URL: https://acnsci.org/journal/index.php/etq/article/view/637. doi:10.55056/etq.637.

[4] V. Osadchyi, I. Krasheninnik, O. Spirin, S. Koniukhov, T. Diuzhykova, Personalized and Adaptive ICT-Enhanced Learning: A Brief Review of Research from 2010 to 2019, in: O. Sokolov, G. Zholtkevych, V. Yakovyna, Y. Tarasich, V. Kharchenko, V. Kobets, O. Burov, S. Semerikov, H. Kravtsov (Eds.), Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume

II: Workshops, Kharkiv, Ukraine, October 06-10, 2020, volume 2732 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 559–571. URL: https://ceur-ws.org/Vol-2732/20200559.pdf.

[5] A. Bile, Development of intellectual and scientific abilities through game-programming in Minecraft, Education and Information Technologies 27 (2022) 7241–7256. doi:10.1007/S10639-022-10894-Z.

[6] C. Richardson, Learn to program with Minecraft: Transform your world with the power of Python, 2015. URL: http://surl.li/nlvnm.

[7] D. Bar-El, K. E. Ringland, Crafting Game-Based Learning: An Analysis of Lessons for Minecraft Education Edition, in: Proceedings of the 15th International Conference on the Foundations of Digital Games, FDG '20, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3402942.3409788.

[8] D. Whale, M. O'Hanlon, Adventures in Minecraft, John Wiley & Sons, 2014. URL: http://surl.li/nlvnm.

[9] S. Bourdeau, T. Coulon, M.-C. Petit, Simulation-Based Training via a "Readymade" Virtual World Platform: Teaching and Learning With Minecraft Education, IT Professional 23 (2021) 33–39. doi:10.1109/MITP.2021.3062935.

[10] O. G. Glazunova, N. Morze, B. Golub, O. Burov, T. Voloshyna, O. Parhomenko, Learning Style Identification System: Design and Data Analysis, in: O. Sokolov, G. Zholtkevych, V. Yakovyna, Y. Tarasich, V. Kharchenko, V. Kobets, O. Burov, S. Semerikov, H. Kravtsov (Eds.), Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kharkiv, Ukraine, October 06-10, 2020, volume 2732 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 793–807. URL: https://ceur-ws.org/Vol-2732/20200793.pdf.

[11] M. Dezuanni, Minecraft and children's digital making: implications for media literacy education, Learning, Media and Technology 43 (2018) 236–249. doi:10.1080/17439884.2018.1472607.

[12] N. Klimová, J. Šajben, G. Lovászová, Online game-based learning through minecraft: Education edition programming contest, in: 2021 IEEE Global Engineering Education Conference (EDUCON), 2021, pp. 1660–1668. doi:10.1109/EDUCON46332.2021.9453953.

[13] J. Kuhn, Minecraft: Education Edition, CALICO Journal 35 (2018) 214–223. URL: https://journal.equinoxpub.com/Calico/article/view/7633. doi:10.1558/cj.34600.

[14] J. Sajben, N. Klimova, G. Lovaszova, Minecraft: Education Edition as a Game-Based Learning in Slovakia, in: EDULEARN20 Proceedings, 12th International Conference on Education and New Learning Technologies, IATED, 2020, pp. 7686–7693. doi:10.21125/edulearn.2020.1946.

[15] M. Dezuanni, J. O'Mara, Impassioned Learning and Minecraft, 2017, pp. 36–48. URL: https://www.routledge.com/Serious-Play-Literacy-Learning-and-Digital-Games/Beavis-Dezuanni-OMara/p/book/9781138689411.

[16] C. Tessler, S. Givony, T. Zahavy, D. Mankowitz, S. Mannor, A Deep Hierarchical Approach to Lifelong Learning in Minecraft, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 31, 2017. URL: https://ojs.aaai.org/index.php/AAAI/article/view/10744. doi:10.1609/aaai.v31i1.10744.

[17] W. H. Guss, B. Houghton, N. Topin, P. Wang, C. Codel, M. Veloso, R. Salakhutdinov, MineRL: A Large-Scale Dataset of Minecraft Demonstrations, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 2442–2448. URL: https://doi.org/10.24963/ijcai.2019/339. doi:10.24963/ijcai.2019/339.

[18] H. C. Lane, S. Yi, Chapter 7 - Playing With Virtual Blocks: Minecraft as a Learning Environment for Practice and Research, Academic Press, San Diego, 2017, pp. 145–166. URL: https://www.sciencedirect.com/science/article/pii/B9780128094815000079. doi:10.1016/B978-0-12-809481-5.00007-9.

[19] O. Alawajee, J. Delafield-Butt, Minecraft in Education Benefits Learning and Social Engagement, International Journal of Game-Based Learning 11 (2021) 19–56. doi:10.4018/IJGBL.2021100102.

[20] J. García-Fernández, L. Medeiros, Cultural Heritage and Communication through Simulation

Videogames—A Validation of Minecraft, Heritage 2 (2019) 2262–2274. URL: https://www.mdpi.com/2571-9408/2/3/138. doi:10.3390/heritage2030138.

[21] R. Andersen, M. Rustad, Using Minecraft as an educational tool for supporting collaboration as a 21st century skill, Computers and Education Open 3 (2022) 100094. URL: https://www.sciencedirect.com/science/article/pii/S2666557322000222. doi:10.1016/j.caeo.2022.100094.

[22] A. Narayan-Chen, P. Jayannavar, J. Hockenmaier, Collaborative Dialogue in Minecraft, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5405–5415. URL: https://aclanthology.org/P19-1537. doi:10.18653/v1/P19-1537.

[23] D. Lee, G. K. Rajbahadur, D. Lin, M. Sayagh, C. Bezemer, A. E. Hassan, An empirical study of the characteristics of popular Minecraft mods, Empirical Software Engineering 25 (2020) 3396–3429. doi:10.1007/S10664-020-09840-9.

[24] M. Nkadimeng, P. Ankiewicz, The affordances of Minecraft education as a game-based learning tool for atomic structure in junior high school science education, Journal of Science Education and Technology 3 (2022) 605–620. doi:10.1007/s10956-022-09981-0.

[25] V. Panja, J. Berge, Minecraft Education Edition's Ability to Create an Effective and Engaging Learning Experience, Journal of Student Research 10 (2021). URL: https://www.jsr.org/hs/index.php/path/article/view/1697. doi:10.47611/jsrhs.v10i2.1697.

[26] M. Opmeer, E. Dias, B. de Vogel, L. Tangerman, H. J. Scholten, Minecraft in Support of Teaching Sustainable Spatial Planning in Secondary Education - Lessons Learned from the Marker Wadden-Project, in: Proceedings of the 10th International Conference on Computer Supported Education - Volume 2: CSEDU, INSTICC, SciTePress, 2018, pp. 316–321. doi:10.5220/0006764403160321.