Co-funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

AI-Powered Puzzles

AI4STEM

Introducing the 5 Big Ideas in Artificial Intelligence using Internet of Things in STEM education

T2.4 IoT Projects Design & Resources Development

29.08.2023 | ATERMON BV
PROJECT NUMBER: 2022-1-FR01-KA220-SCH-000085611

AI4STEM IoT Projects Design & Resources Development
Project: AI-Powered Puzzles

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Table of Contents

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

1.Introduction to the Project: AI in Games & Puzzles

1.1 The scope of the Project

This IoT project aims to introduce secondary education students (ages 12-15) to the world of Artificial Intelligence (AI) in the context of games and puzzles. It covers the Five Big Ideas in AI, which are Perception, Representation and Reasoning, Learning, Natural Interaction, and Societal Impact. The project is structured to include hands-on activities, discussions, and ethical considerations related to AI. Students will use BBC Micro:bit devices and coding environments to create AI-powered games and puzzles that address each of the Big Ideas in AI.

The project aims to achieve several key objectives:

- Introduce students to the core concepts of AI and its Five Big Ideas: Perception, Representation and Reasoning, Learning, Natural Interaction, and Societal Impact.
- Provide students with practical, hands-on experience using BBC Micro:bit devices to create AI-powered games and puzzles.
- Foster critical thinking, problem-solving skills, and ethical considerations related to AI in gaming and technology.
- Promote the responsible and inclusive development of AI-powered games and puzzles.

Each of the Five Big Ideas in AI is explored through dedicated activities, allowing students to understand the practical applications and implications of AI in games and puzzles. Students work with BBC Micro:bit devices, learning to code, program sensors, and develop AI-driven game mechanics. The project provides a range of resources, including activity sheets, code sheets, and assessment tools for teachers and students. These resources support learning, guide hands-on experiments, and evaluate student performance.

The project aims to have a lasting educational impact as it equips students with practical skills in coding and AI technology, it encourages students to think critically about the ethical, social, and cultural aspects of AI, it promotes awareness of AI's societal implications, fostering responsible and inclusive AI development, and lastly it inspires students to consider the role of AI in the broader context of their lives and future careers.

1.2 The target groups

The primary target audience for this project is secondary education students, specifically those aged 12 to 15. The project is designed to be engaging and educational, making AI concepts accessible to this age group. Secondary education teachers and educators can also benefit from the resources provided to implement the project effectively in the classroom.

1.3 The purpose of this document

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

The purpose of the project document is to serve as a comprehensive guide for both educators and students. It includes detailed instructions, explanations, and resources to facilitate the exploration of AI in games and puzzles. This document serves as a curriculum guide, providing a structured framework for teaching AI concepts in a hands-on and engaging manner. It is meant to:

1. Provide Structure: The document outlines the project's scope, objectives, and the specific activities that help students understand the Five Big Ideas in AI.
2. Educational Resource: It offers a valuable resource for educators, providing step-by-step guidance on how to introduce AI concepts in the classroom.
3. Engage Students: The document aims to make AI concepts engaging and accessible to students by incorporating practical activities, discussions, and examples.
4. Promote Ethical Considerations: It emphasizes the ethical, social, and cultural aspects of AI, encouraging students to think critically and responsibly about AI technology.
5. Facilitate Learning: The provided resources, including activity sheets, code sheets, and assessment tools, support students' learning and understanding of AI.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

2. Glossary of the Unit

Word	Definition
• IoT (Internet of Things)	A network of interconnected physical devices (things) embedded with sensors, software, and other technologies to collect and exchange data.
• AI (Artificial Intelligence):	The simulation of human intelligence processes by machines, especially computer systems, to perform tasks that typically require human intelligence.
• Big Ideas in AI:	Five core concepts that encompass key principles of artificial intelligence: Perception, Representation and Reasoning, Learning, Natural Interaction, and Societal Impact.
• BBC Micro:bit:	A pocket-sized programmable computer with various sensors, LEDs, and wireless communication capabilities, ideal for learning about IoT and coding.
• Perception	The process by which AI systems collect and interpret data from the physical world through sensors, cameras, or other input devices. It involves understanding and making sense of information to interact with the environment.
• Representation & Reasoning	The ability of AI systems to create internal models of the world and use these models for problem-solving, decision-making, and understanding complex relationships. It includes the use of data structures and algorithms to represent knowledge.
• Learning	The capacity of AI systems to improve their performance by recognizing patterns in data, adapting to new information, and making predictions. Machine learning, a subset of AI, is a key component of this big idea.
• Natural Interaction	AI's capability to communicate with users and machines in a way that is intuitive and similar to human interaction. This includes speech recognition, natural language processing, and gesture-based interfaces.

Co-funded by the European Union

• Societal Impact	The examination of AI's influence on society, culture, and individuals. It encompasses ethical considerations, fairness, accountability, and the broader implications of AI technology on people's lives and well-being.
• AI-Powered Game	A video game or puzzle that incorporates artificial intelligence algorithms to enhance gameplay, character behavior, and player experience.
• Procedural Content Generation	The use of algorithms to create in-game content, such as levels, characters, or items, dynamically and automatically.
• Bias in AI	The presence of unfair, discriminatory, or skewed results in AI algorithms, often resulting from biased training data.
• Personalization	he adaptation of content, experiences, or recommendations based on individual preferences and behavior.
• Social Interaction	Engagement and communication between individuals or groups in a virtual or online context, often facilitated by AI-driven features.
• Game Addiction	Excessive and compulsive gaming behavior that can negatively impact an individual's well-being.
• Cultural Representation	The portrayal of cultural values, norms, and identity in games, which can impact cultural perceptions.
• Ethical Dilemma	A situation that presents a choice between conflicting moral principles or values, often related to AI use in games and puzzles.
• Accessibility	Making games and technology available and usable by individuals with disabilities through adaptive features and technologies.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3. Introduction to the "AI-Powered Puzzles: An IoT Adventure"

3.1 Description

In this project, students will embark on a journey to create IoT-powered games and puzzles that encompass the Five Big Ideas in AI:

- 3.1 Perception: Students will explore how to use Micro:bit's sensors (e.g., accelerometer, temperature sensor, light sensor) to gather data about the physical world. They'll design games that react to changes in the environment, like tilting the Micro:bit to control a game character.
- 3.2 Representation and Reasoning: In this phase, students will learn how to represent and manipulate information. They'll design games that use decision trees, flowcharts, or simple algorithms to make choices based on player input, creating interactive and decision-driven game narratives.
- 3.3 Learning: Students will delve into the concept of machine learning, albeit in a simplified form. They can design puzzle games that adapt and become more challenging over time as the Micro:bit learns from the player's strategies, making the gameplay more engaging.
- 3.4 Natural Interaction: This aspect of the project will involve creating games that respond to natural inputs like gestures, voice commands, or visual cues. Students will use the Micro:bit's sensors, microphone, and LED display to build games that interact in intuitive ways.
- 3.5 Societal Impact: In this final phase, students will design games and puzzles that have a societal impact. This can involve creating educational games that teach players about environmental issues, ethics, or real-world problem-solving.

By the end of the project, students will not only have a deeper understanding of AI and IoT but also have developed their critical thinking, problem-solving, and coding skills. They will present their creations to their peers, promoting collaborative learning and sharing their innovative AI-powered games and puzzles.

3.2 Learning objectives & outcomes

Learning Objectives:

- Understand the fundamentals of AI, IoT, and their integration.
- Develop problem-solving and critical-thinking skills.
- Gain hands-on experience with the BBC Micro:bit.

Outcomes:

- Design and implement IoT-powered games and puzzles showcasing AI principles.
- Develop coding and debugging skills.
- Share and present their projects with peers.

3.3 Estimated duration of the Unit

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

This project is designed to be completed in approximately 10-12 days, with each Big Idea taking 2-3 days to explore.

3.4 Activity 1: Big Idea of Perception

3.4.1 Introduction -theory

In this activity, students will delve into the Big Idea of Perception, one of the foundational concepts in artificial intelligence. Perception, in the context of AI, refers to the ability of machines to sense and interpret their environment. It's about understanding and responding to data gathered from various sensors, much like how our senses (sight, hearing, touch) allow us to perceive and react to the world around us.

Perception in AI involves using sensors and data to make sense of the physical world. In this activity, students will explore how the BBC Micro:bit, equipped with sensors, can serve as a basic yet powerful tool for perceiving and reacting to the physical world. Specifically, students will focus on the accelerometer sensor, which can detect changes in tilt and acceleration.

Here's how the activity covers the Big Idea of Perception:

1. Sensors and Data Collection: Students will learn that perception in AI begins with sensors. They'll understand that sensors like the accelerometer in the Micro:bit can collect data related to tilt and acceleration. This data is essential for the Micro:bit to understand how it's positioned in space.

2. Interpreting Data: To make sense of the data collected by the accelerometer, students will delve into coding. They will learn to interpret the data and translate it into meaningful actions. For instance, they will understand that when the Micro:bit tilts in a particular direction, the accelerometer records these changes, and the code they write will interpret these changes as commands.

3. Real-world Interaction: Students will recognize that once the Micro:bit interprets the data from the accelerometer, it can interact with the real world. This is a key aspect of perception in AI. In the case of this activity, the Micro:bit can control a game character's movement based on the tilt gestures it "perceives."

4. Responsive Systems: The activity demonstrates how perception allows AI systems, like the Micro:bit, to be responsive. It can react to changes in its surroundings. In this specific context, the Micro:bit's game character moves in response to tilting, creating an interactive and responsive experience.

5. Perception as a Building Block: Students will understand that perception is a fundamental building block in AI. It is the means by which machines gather information from their environment, and it's the basis for more advanced AI functionalities like decision-making and learning. By mastering perception through this activity, students lay the foundation for exploring more complex AI concepts in later activities.

In summary, this activity effectively covers the Big Idea of Perception by introducing students to the core concept of how machines collect and interpret data from their environment to interact with it. It

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

highlights the importance of sensors and real-world interaction in AI, setting the stage for further exploration of AI principles in subsequent activities.

3.4.2 Hardware

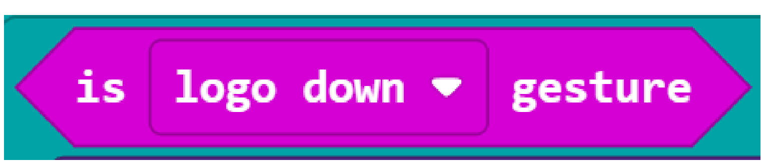
- BBC Micro:bit
- Computer with USB cable for Micro:bit connection
- MakeCode coding environment

3.4.3 Setup

- Begin by introducing students to the BBC Micro:bit, a pocket-sized computer equipped with various sensors and LEDs.
- Have each student or group of students connect the Micro:bit to a computer using a USB cable.
- Instruct students to install the necessary coding environment. For this project, you can use MakeCode, a block-based coding platform designed for the Micro:bit. You can explain the setup process as follows:
    1. Open a web browser and go to the MakeCode website (https://makecode.microbit.org/).
    2. Connect the Micro:bit to your computer using a USB cable.
    3. The Micro:bit should appear as a drive on your computer.
    4. In MakeCode, select "New Project."
    5. Students can drag and drop code blocks to create their program.
    6. To upload the program to the Micro:bit, they can click "Download."

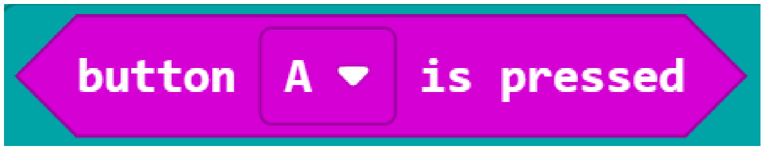
3.4.3.1 Code

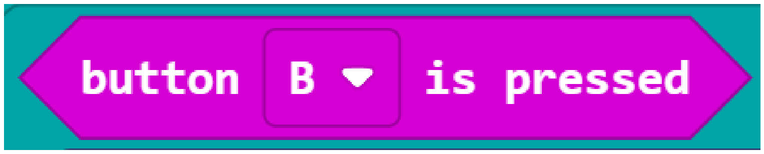
Provide a sample code that utilizes the accelerometer sensor to detect tilting. In this code, you can introduce students to the concept of event-driven programming. Use the following example as a starting point:



Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.





Explain that the code listens for specific gestures and button presses and executes certain actions or events when those actions are detected. This will provide students with the basics of creating a responsive program based on data from the Micro:bit.

3.4.4 Exercise / Experiment 1

In this experiment, students will apply their understanding of perception by creating a game where tilting the Micro:bit controls a game character. The game can be designed as follows:

Creating the Maze:

1. Design the Maze:
    - Begin by introducing the concept of a maze to the students. Show them examples of mazes and discuss the challenge of navigating through them.
    - Instruct students to design a simple maze on paper or using digital drawing tools. The maze should consist of walls, a starting point, and an endpoint. The walls can be represented as lines, and the starting and endpoint can be marked with distinct symbols.

Programming the Micro:bit:

Walk students through creating a new MakeCode project. Explain that they will use the Micro:bit to control a game character's movement within the maze. Use the Micro:bit's LED grid to represent the game character. This can be a simple lit-up LED. Encourage students to use the example blocks provided earlier as a foundation. These blocks can be a starting point for controlling a game character's movement based on gestures. Students can use variables to keep track of the character's position on the grid. Define variables for the character's X and Y coordinates. Provide students with code blocks to move the character based on gestures. Here's a sample code snippet to get them started:
Use the following blocks to program the game's behavior:

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Figure 1 Beginning of program in MakeCode

First, you need to create a few variables. Recall that variables function as containers that store information. In this case, two variables are necessary to monitor the player's location. One is designated to record the player's x position, while the other is dedicated to tracking the player's y position.

Additionally, you need a variable to monitor the maze level, allowing for the possibility of multiple levels. Another variable is necessary to track the game's status, indicating whether it is active or if it has concluded.

The initial values are set to start at level 1, and gameOn is initialized as True. This is because, upon powering on the Micro:bit, the intention is to commence the game immediately. While the starting point for the player's location can be chosen arbitrarily, it needs to be recalled later when configuring the maze level to ensure the player does not begin inside a wall. For this example, the player is initiated at x=0 and y=0.

Note

The complete set of coordinates **x,y** for the grid that micro:bit offers are presented in the table below.

Table 1 X, Y coordinates for micro:bit grid

(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

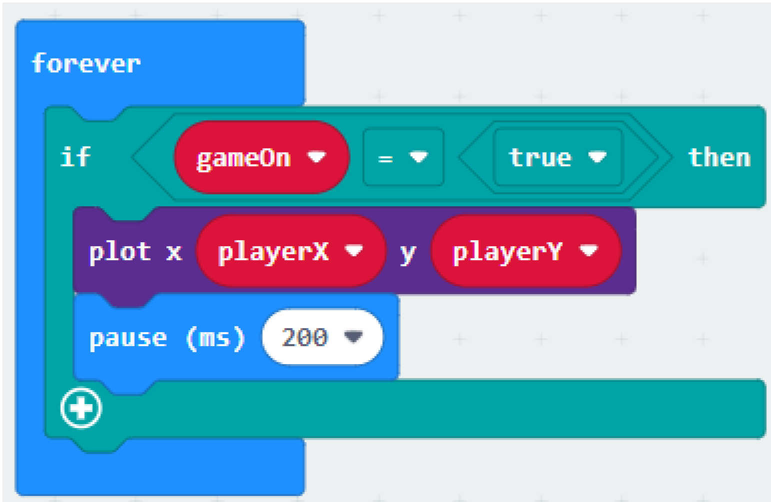


Figure 2 First forever loop

Now that the initial variables are in place, ensure the player is displayed on the Micro:bit screen!

To achieve a distinctive blinking effect for the player, you need to employ the 'plot x y' block alternated with the 'pause' block within a forever loop. The intention is for the player to continuously blink on and off. When maze walls are introduced, the Micro:bit will overwrite the player each time it draws the walls. By incorporating a pause block here, we ensure that the player won't be immediately re-plotted, resulting in the desired blinking effect.

The utilization of the playerX and playerY variables created earlier is crucial. Why? If numerical values were directly inputted here, it would limit the flexibility to make the player move. The use of variables enables you to modify the values of playerX and playerY, allowing the forever loop to plot the player's new location.

It's essential to note that the pause block operates in milliseconds (e.g., 200 ms = 0.2 seconds), and the blinking speed can be customized by adjusting the duration of the pause.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

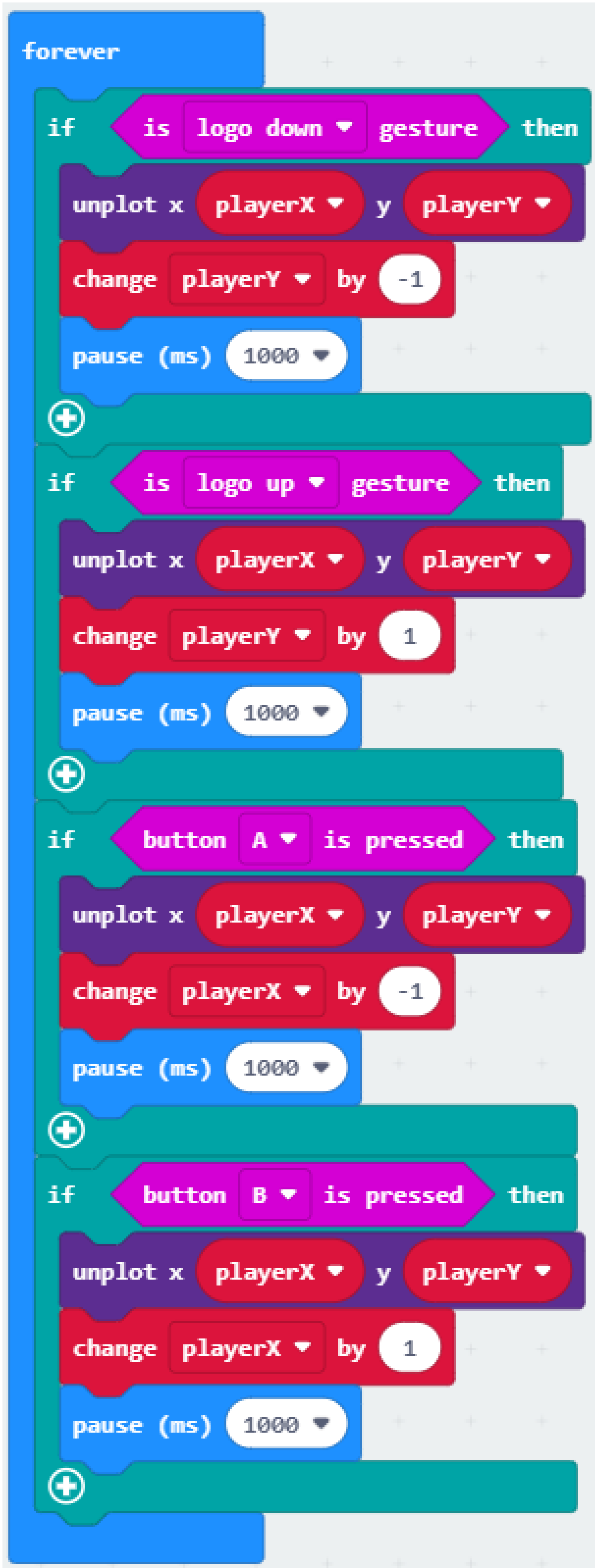


Figure 3 Second forever loop

Now you need to setup the player's movements (left, rights, up, and down). You need to use the two integrated buttons and the logo swipe function.

Set the logo up gesture to move up, the logo down gesture to move down, the button A to move left, and the button B to move right.

To accomplish this, utilize if statements. These statements assess whether a condition is true; if it is, any blocks inside the if block are executed. When you embed an if statement within a forever loop, you continually check if the condition is true.

For player movement, modify the playerX or playerY variables. It's crucial to remember that decreasing or increasing playerX causes left or right movement, respectively, while decreasing or increasing playerY results in upward or downward movement, respectively. Given that we consistently plot the player's location using these variables, any changes automatically reflect the player's new position.

It's worth noting that a brief 300ms pause is added after each button press. This prevents the Micro:bit from moving the player across multiple spaces rapidly with each button press, as the code runs quickly without the pause.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

stly, displaying the maze walls

Figure 4 Third forever loop

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

on the LED screen; secondly, continually checking if the player collides with a wall (indicating game over); and thirdly, perpetually assessing if the player successfully completes the maze level.

A forever loop is employed. Within this loop, an 'if' statement is used to verify if the level variable equals 1. Consequently, this code segment will only execute when the level variable equals 1. If you want to add more levels, then make sure that this variable changes accordingly.

Inside the 'if' statement, the maze walls are displayed using the 'show leds' block. LEDs are illuminated to represent walls, while unlit LEDs denote the maze paths. Caution must be exercised to ensure that the player's starting position, set earlier at x=0, y=0, does not coincide with a maze wall.

The subsequent task involves checking if the player collides with a wall. This is achieved through additional 'if' statements, verifying if the playerX and playerY variables align with the coordinates of a wall in the 5x5 LED grid.

Lastly, the code checks if the player successfully navigates through the maze. In this example, the maze's end is at x=1, y=4. If these conditions are met, a successful melody plays, the player’s position is reset to the beginning of the maze, and a smiley face appears on the Micro:bit. If you have added additional levels, then you also need to change the variable level by 1.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

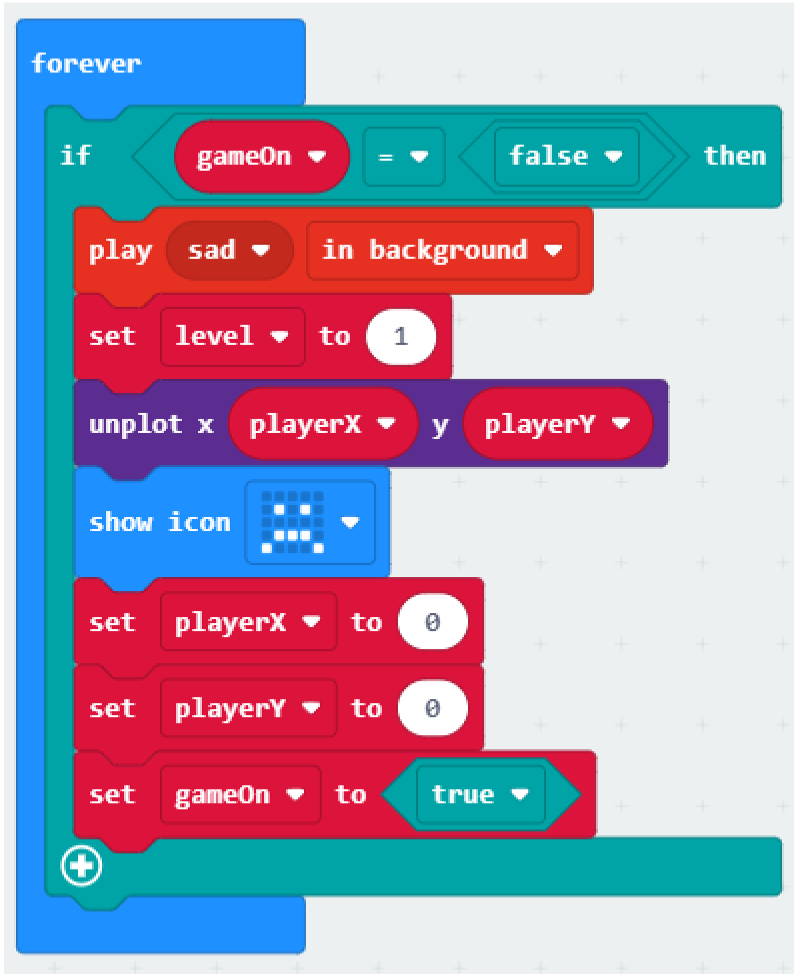


Figure 5 Fourth forever loop

In case of a game over, implement an action triggered by the 'gameOn' variable indicating a collision with a wall.

Within a forever loop, an 'if' statement is used to assess the value of the 'gameOn' variable. If it equals 'false,' the game over code is executed.

In this instance, a sad melody plays in the background, the 'level' is reset, the player LED is unlit, a sad face is displayed, and the game starts from the beginning.

This experiment not only introduces students to the concept of perception but also provides them with a hands-on opportunity to apply this knowledge in a creative and interactive manner.

The complete activity and code are on the pdf called: Activity 1 – create a maze game.pdf

Transferring the code to Micro:bit:

Click on the "Download" button on the bottom left of the coding environment and follow the instructions in order to transfer your code to the Micro:bit.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.4.5 Questions

Multiple choice

How can the Micro:bit receive the code in blocks?
a) By connecting it to a printer.
b) By plugging it into a power outlet.
c) By connecting it to the MakeCode environment and downloading the code.
d) By shaking the Micro:bit.

True/False

The accelerometer sensor can detect tilting and acceleration. (True/False)

3.5 Activity 2 - Big Idea of Representation & Reasoning:

3.5.1 Introduction – Theory

In this activity, students will delve into the Big Idea of Representation and Reasoning, which plays a crucial role in artificial intelligence. Representation and Reasoning refer to how AI systems model and interpret information to make informed decisions. This activity will provide students with hands-on experience in understanding and implementing effective information representation and decision-making in the context of creating an AI-enhanced puzzle solving game on the BBC Micro:bit.

Here's how the activity covers the Big Idea of Representation and Reasoning:

1. **Data Representation:** In AI, data representation is crucial as it determines how information is encoded and stored for processing. In the AI Number Guesser game, the Micro:bit represents the target number and the player's guess using variables (targetNumber and guess). This simple form of data representation is foundational in AI, where more complex systems might use sophisticated data structures to represent knowledge and information.
2. **State Representation:** The game's current state (i.e., the current guess and whether it is higher, lower, or equal to the target number) is a simple example of state representation. In more complex AI systems, state representation is key to understanding the environment and making decisions.
3. **Feedback as a Form of Representation:** The feedback given by the Micro:bit (arrows indicating the direction to adjust the guess, and the checkmark for a correct guess) is a form of representing information to the user. This is analogous to how AI systems might receive and interpret feedback from their environment to adjust their actions or decisions.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

4. Decision Making Based on Feedback: The core of the game involves the player making decisions (guesses) based on feedback from the Micro:bit. This process mimics how AI systems use reasoning to make decisions or solve problems based on the information available to them.
5. Iterative Reasoning Process: The player engages in an iterative process of refining their guess based on feedback, which is a fundamental aspect of reasoning in AI. AI systems often use iterative processes (like in machine learning algorithms) to gradually improve their performance or converge on a solution.
6. Problem-Solving Strategy: The player uses a problem-solving strategy to guess the number, which involves understanding the feedback and reasoning about the next best action. This strategy is similar to how AI systems use algorithms and heuristics to solve problems.

This activity empowers students to explore the concepts of Representation and Reasoning by creating decision-driven games, fostering their critical thinking and decision-making skills in an engaging way. By engaging with this game, students learn to think about how AI systems represent information and use that representation to reason and make decisions. The game provides a tangible example of how feedback can be used to guide decision-making, a concept that is central to many AI applications, such as reinforcement learning. The AI Number Guesser game, though simple, encapsulates key concepts of representation and reasoning in AI. It offers a hands-on experience that helps students understand these concepts in a practical and accessible way. This understanding forms the basis for exploring more complex AI topics, such as machine learning, natural language processing, and robotics, where representation and reasoning play crucial roles.

3.5.2 Hardware

- BBC Micro:bit
- Computer with USB cable for Micro:bit connection
- MakeCode coding environment

3.5.3 Setup

- Ensure students have their Micro:bit devices and USB cables ready.
- Confirm that the MakeCode coding environment is installed on their computers.
- Connect the Micro:bit to the computer using the USB cable.
- Open the MakeCode editor for Micro:bit in a web browser.
- Create a new project and choose the Blocks programming interface.
- Create the code for the game.
- Download the code onto the Micro:bit.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.5.3.1 Code

Sample code for creating a puzzle-solving game:

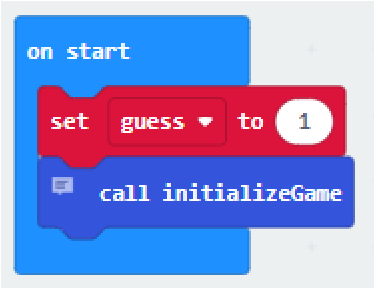


Figure 6 Game initialisation

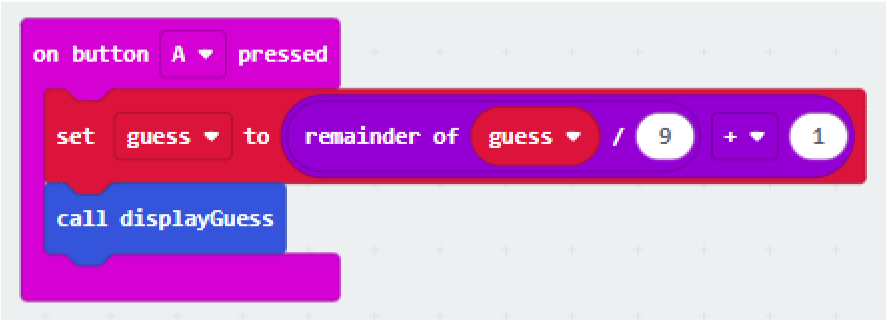


Figure 7 Setup button A

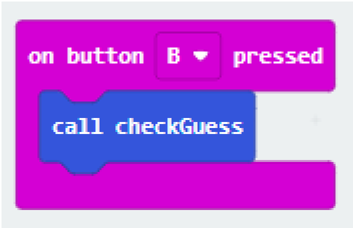


Figure 8 Setup button B

Figure 9 Display the guessed number on the LED array of the Micro:bit

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Figure 10 Function to start the game

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

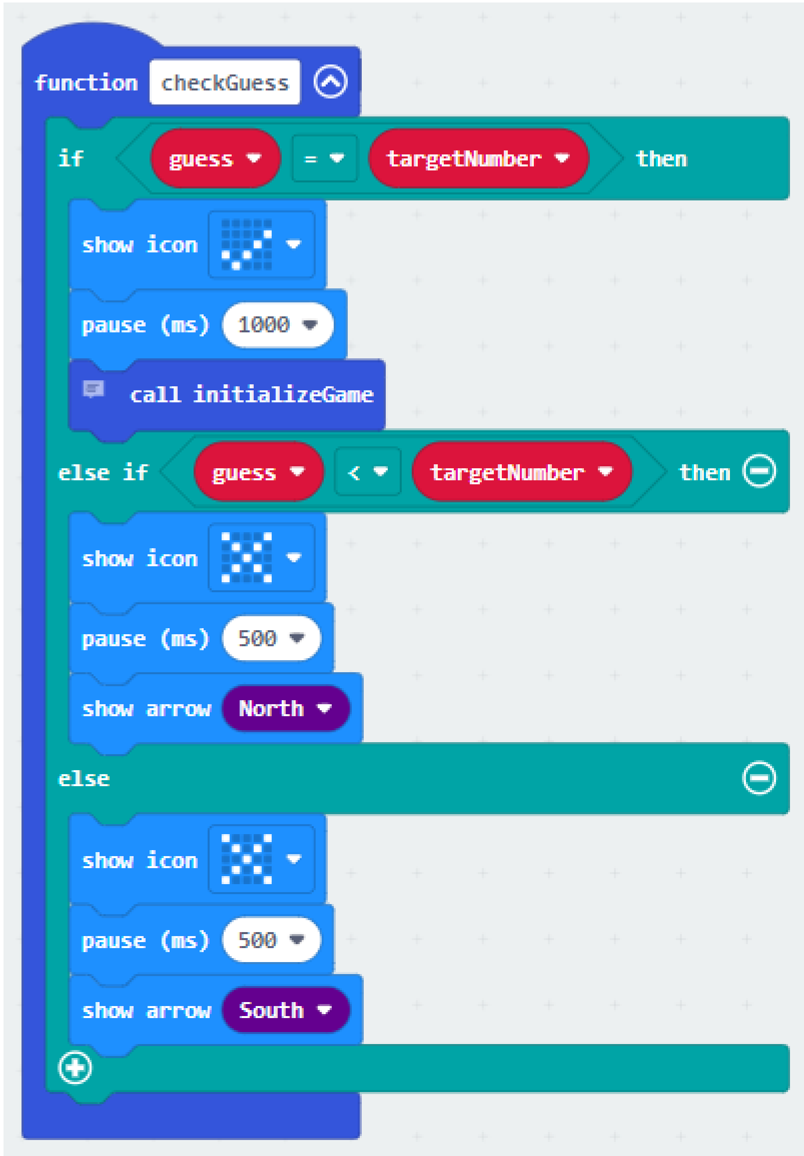


Figure 11 Function to setup the number-guessing mechanic

Explain how the code uses decision trees to guide players through a text-based adventure game. Students will choose between options A and B, affecting their score and health.

3.5.4 Exercise / Experiment 2

In experiment 2, students will develop a puzzle solver game in order to understand how AI processes information and how by using this information it can make informed decisions. Here's how the experiment is structures:

- The Micro:bit randomly selects a number between 1 and 9.
- The player guesses the number by pressing button A to increase their guess and button B to submit the guess.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

- The Micro:bit provides feedback through its LED display: a checkmark for a correct guess, an up arrow for a guess that is too low, and a down arrow for a guess that is too high.
- The game resets automatically after a correct guess, allowing continuous play.

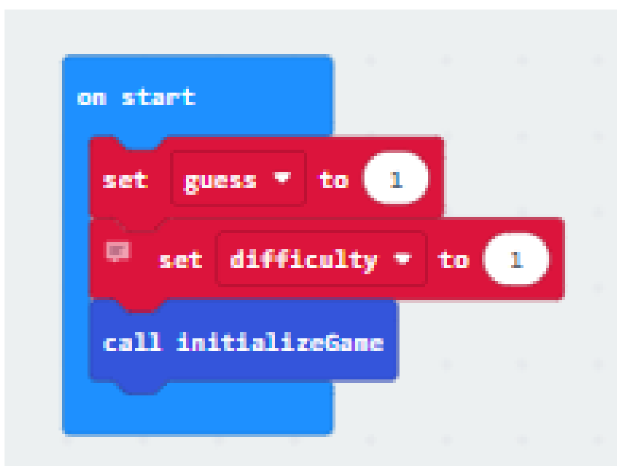
Code explanation:

- Variable Initialization: targetNumber is the randomly selected number, and guess is the player's current guess.
- Display Function (displayGuess): Shows the current guess on the LED display.
- Guess Checking Function (checkGuess): Compares the guess to the target number and provides feedback. Resets the game if the guess is correct.
- Button Press Handlers: Button A increases the guess, and button B submits the guess and checks it.

This project helps students understand how AI can represent information and reason based on feedback. It illustrates the concept of iterative improvement based on feedback, a key aspect of many AI algorithms. The game encourages problem-solving skills and logical thinking, as players must deduce the correct number based on limited information.

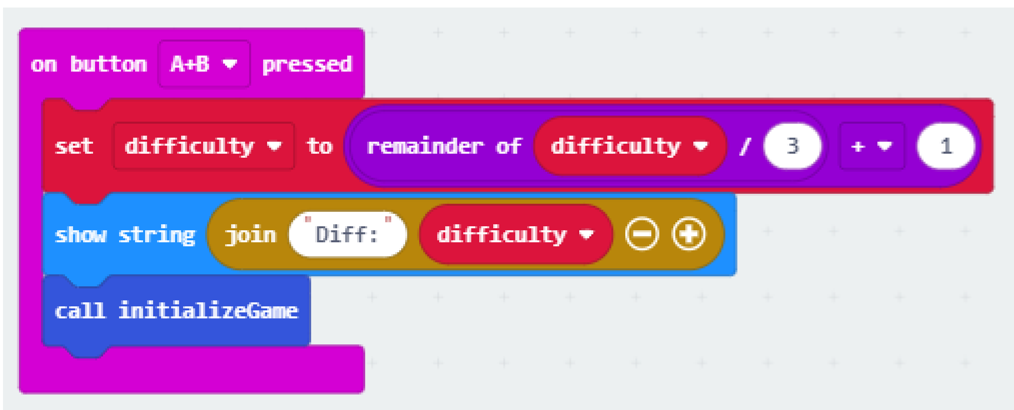
Some optional extension ideas can involve introducing difficulty levels by increasing the range of number, implementing a scoring system based on the number of guesses taken to find the correct number and adding a timer to see how quickly the player can guess the correct number.

Foe example, you can introduce the following blocks in order to include the aforementioned features:



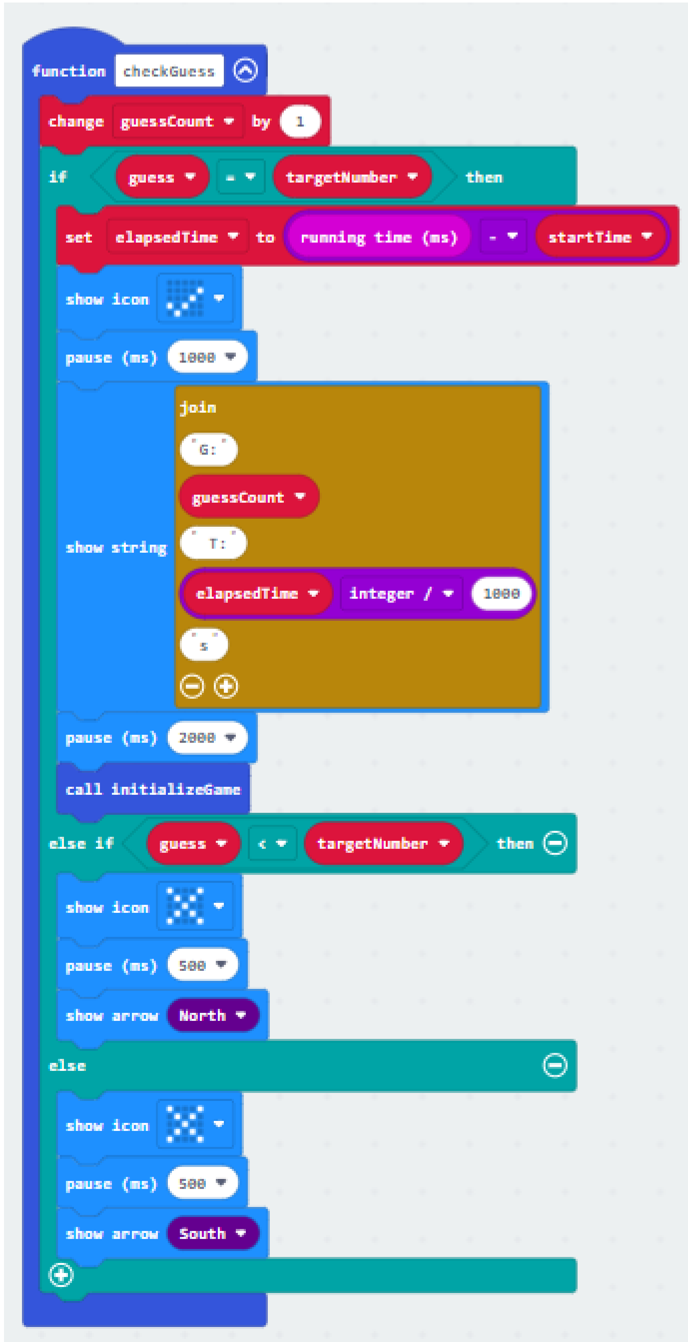
Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.



Code Explanation

- targetNumber: Stores the randomly selected target number for the player to guess.
- guess: The current guess of the player.
- difficulty: Represents the difficulty level of the game (1: Easy, 2: Medium, 3: Hard).
- guessCount: Tracks the number of guesses the player has made.
- startTime: Stores the time when the guessing round starts.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

initializeGame Function

- Sets targetNumber to a random number within the range based on the selected difficulty level.
- Resets guess to 1 and guessCount to 0 at the start of each round.
- Sets startTime to the current running time for the time challenge.
- Calls displayGuess to show the initial guess.

displayGuess Function

- Clears the LED screen and displays the current guess.

checkGuess Function

- Increments guessCount each time the player submits a guess.
- Compares the guess to targetNumber and provides feedback:
    - Shows a checkmark if the guess is correct.
    - Shows an up arrow if the guess is too low.
    - Shows a down arrow if the guess is too high.
- If the guess is correct, calculates the elapsed time, displays the score (number of guesses) and time taken, and resets the game after a pause.

Button Event Handlers

- Button A: Increases the guess and wraps around based on the maximum number for the current difficulty level. Calls displayGuess to update the display.
- Button B: Calls checkGuess to submit the current guess and receive feedback.
- Buttons A + B: Cycles through the difficulty levels (1 to 3), displays the current difficulty, and restarts the game at the new difficulty level.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.5.5 Questions

Multiple choice

Which of the following best describes the concept of representation in the context of the AI Number Guesser game?

a) The process of the player guessing the number.
b) How the game uses variables to store the target number and the player's guess.
c) The method by which the Micro:bit generates a random number.
d) The way the player changes the difficulty level.

How does the AI Number Guesser game illustrate the concept of reasoning in AI?

a) By allowing the player to change the difficulty level of the game.
b) By displaying the number of guesses and the time taken to guess correctly.
c) Through the use of a random number generator to select the target number.
d) Through the player's use of feedback to adjust their guesses.

True/False

Representation and Reasoning in AI involve how machines model and interpret information to make decisions. (True/False)

3.6 Activity 3 – Big Idea of Learning

3.6.1 Introduction – Theory

In this activity, students will explore the Big Idea of Learning in AI, which involves the ability of machines to learn from data and adapt their behavior based on that learning. This concept is often referred to as machine learning or artificial intelligence. The key components of this activity are as follows:

- Concept Overview: Learning in AI refers to the ability of an AI system to improve its performance over time by gaining experience or being exposed to new data. It often involves recognizing patterns, making predictions, and adjusting behaviours based on feedback.

- Types of Learning: In AI, there are various types of learning, such as supervised learning, unsupervised learning, and reinforcement learning. Each type has its own methodologies and use cases.

- Pattern Recognition: The core of the AI Shake Detector game is pattern recognition. The Micro:bit uses its accelerometer to detect shaking motions and categorizes them into different levels of intensity. This process mimics how AI systems learn to recognize patterns in data.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

- Feedback Loop: The game involves a basic feedback loop where the player shakes the Micro:bit, and the device responds with a corresponding LED pattern. This immediate feedback allows players to understand how their actions (shaking patterns) are interpreted by the system.
- Simulating Learning: While the Micro:bit has limited capabilities for advanced machine learning, the project simulates basic learning principles by using algorithms to classify shaking patterns. The game can be seen as a simplified model of how AI systems learn from and adapt to new data.

The AI Shake Detector game provides an accessible and engaging way for students to explore the Big Idea of Learning in AI. Although the Micro:bit's capabilities are limited compared to more advanced AI systems, this project effectively demonstrates key learning concepts, such as pattern recognition and feedback loops. It serves as a foundation for understanding more complex machine learning and AI topics, making it an ideal introductory project for students new to AI. Through hands-on interaction and experimentation with the Micro:bit, students can observe how AI systems might process and learn from sensory data, providing a tangible example of AI learning in action.

The project also encourages students to think critically about how AI systems interpret data, the importance of accurate data representation, and the role of feedback in shaping AI behavior. By experimenting with different shaking patterns and observing the Micro:bit's response, students engage in a basic form of AI training, helping them grasp the concept of how AI systems learn from experience.

3.6.2 Hardware

- BBC Micro:bit
- Computer with USB cable for Micro:bit connection
- MakeCode coding environment

3.6.3 Setup

- Connect the Micro:bit to a computer using a USB cable.
- Open the MakeCode coding environment.

3.6.4 Exercise / Experiment 3

In this experiment, students will have the opportunity to practically apply the concept of machine learning by training the BBC Micro:bit to recognize a specific movement or action. They will then program the Micro:bit to respond in a particular way when this learned action is detected. This hands-on exercise will help students grasp the fundamental principles of machine learning and its practical applications. The objective of this exercise is to teach students the concept of machine learning by training the Micro:bit to recognize shaking and responding with a message.

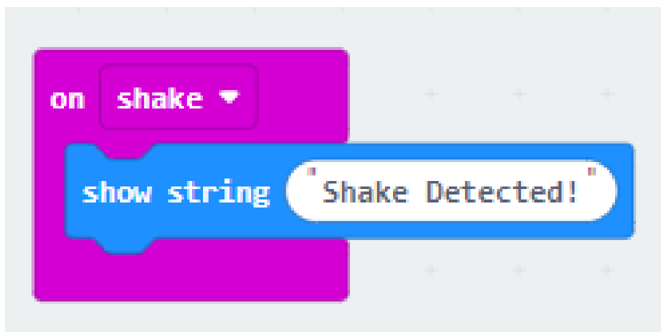
Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Machine learning is a branch of AI where machines learn from data and adapt their behavior based on that learning. Students are going to teach their BBC Micro:bits to recognize a specific action – shaking. Then, make them respond in a certain way when they 'see' this action. By the end of this experiment, students will understand the basics of how machine learning works.

Each student or group should be provided with a Micro:bit, a USB cable, and access to the coding environment. They should make sure the Micro:bit is connected to the computer.

Training the Micro:bit: Students should be guided through the process of training the Micro:bit to recognize the chosen action. This can be done using the accelerometer sensor to collect data when the Micro:bit is subjected to the action. For example, when students want to teach the Micro:bit to recognize shaking, they can physically shake the Micro:bit and record the accelerometer data associated with that action. Create a function in the MakeCode environment that will allow to collect this data. In the MakeCode environment, students should select 'Input' from the blocks palette. Then, drag and drop the block 'on shake' into the code editor.



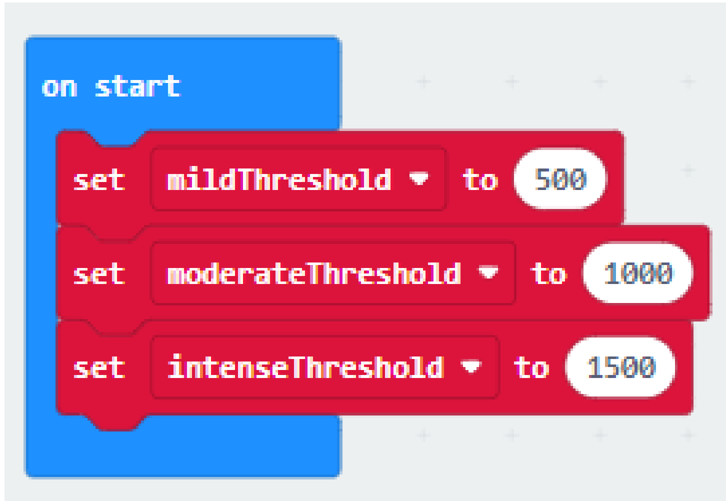
This block tells the Micro:bit to pay attention when it's shaken, and then collect the data from the accelerometer. Shake the Micro:bit while the function is active. If you add a message using the 'show string' block, the message "Shake Detected" will appear each time you shake the Micro:bit.

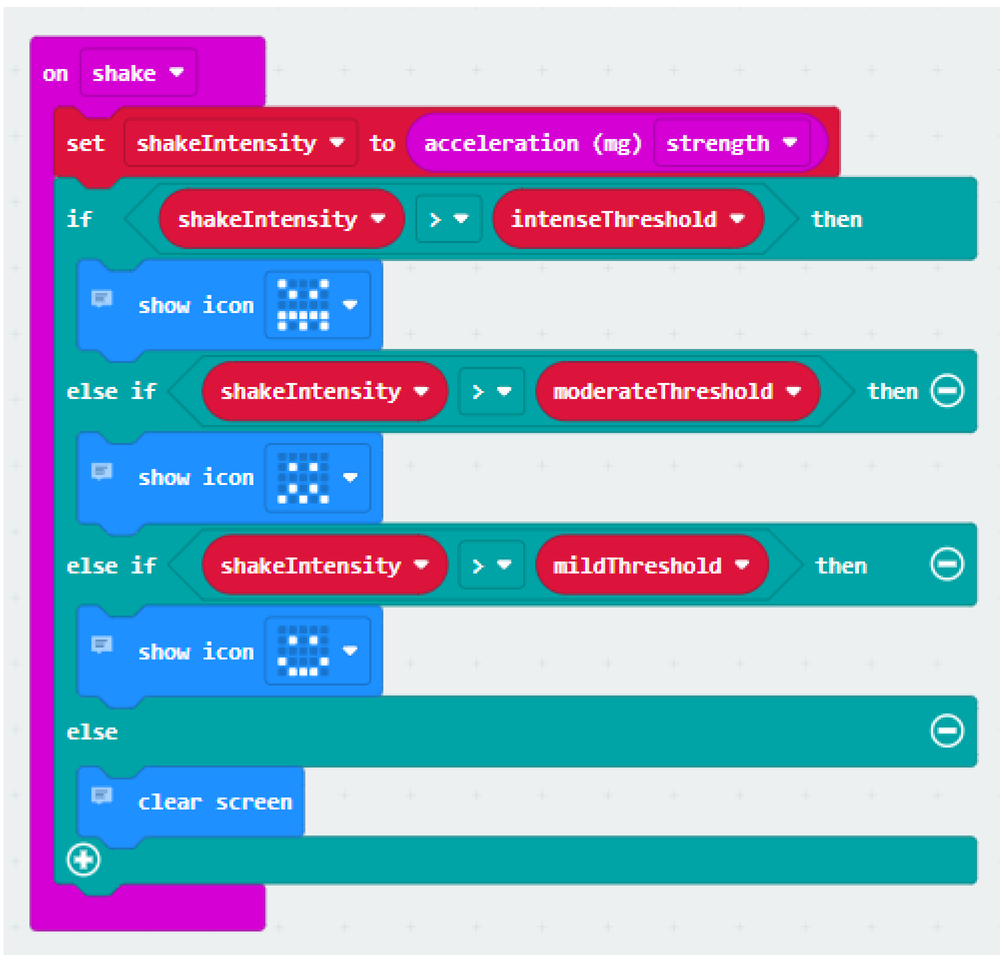
Now you want to teach the Micro:bit to understand different levels of intensity and respond accordingly. The game will categorize shaking into mild, moderate and intense and the Micro:bit will respond with different LED patterns based on the detected shaking intensity. Students can experiment with different shaking patterns to see how the Micro:bit categorizes them.

Programming the Response: Students should program the Micro:bit to respond when it detects the action. They can set specific responses, such as displaying a message on the LED grid, playing a sound, or triggering an event. In this project, the Micro:bit will show different LED patterns.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.





After adding the response code, it's time to test it. Download the code to your Micro:bit and shake it to see if it responds with the message you programmed. If everything is set up correctly, you should

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

see your response on the LED grid. You can practice and fine-tune your response by adjusting the code..

Testing: Students will test their Micro:bits to see if they have successfully trained them to recognize and respond to the action. They can do this by performing the action and observing the Micro:bit's response. Each student or group should take turns testing their Micro:bit by performing the chosen gesture. For example, if the Micro:bit is trained to recognize shaking, give it a good shake. Then, students should observe the response on the LED grid. Does it display the message it was programmed to display or respond in the way it was intended? If the response isn't as expected, students can go back to their code and adjust it.

Discussion: The teacher should lead a class discussion where students share their experiences and observations. Discuss the significance of training machines and how the Micro:bit's response is based on learned data. Emphasize the real-world applications of machine learning in devices.

1. How did your Micro:bit respond when you performed the gesture? Was it accurate and responsive?

2. What did you learn about the process of training machines or devices to recognize specific patterns or actions?

3. How could this technology be used in real-life applications?

4. What were some challenges you faced, and how did you overcome them?

Students explored the concept of machine learning using their Micro:bits. They personalized their Micro:bits by programming them to recognize gestures and respond. This is just the beginning of what can be achieved with AI and machine learning.

General Teacher's Tips:

- Encourage students to experiment with different actions. For example, they can teach the Micro:bit to recognize shaking, tapping, or any other specific gesture.

- Discuss the importance of data quality in machine learning. The more diverse and representative the training data, the better the model's accuracy.

- Encourage students to think about how this technology is used in everyday life, such as in devices like smartphones that respond to gestures.

This experiment provides a practical understanding of machine learning and its applications using the Micro:bit as an accessible platform. It allows students to see firsthand how machine learning can be used to train devices to respond to specific actions or patterns.

A complete activity sheet and sample code are on the pdf called: Activity 3 – gesture recognition game.pdf

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.6.5 Questions

Multiple choice

What do the mildThreshold, moderateThreshold, and intenseThreshold variables represent in the AI Shake Detector game?

a) Different game levels that the player can choose.
b) The number of shakes required to complete the game.
c) Threshold values for categorizing the intensity of shakes.
d) The duration of time the Micro:bit needs to be shaken.

What is the purpose of the calibration mode in the AI Shake Detector game?

a) To turn off the Micro:bit.
b) To allow the player to adjust the sensitivity of shake detection.
c) To change the LED patterns displayed by the Micro:bit.
d) To count the number of shakes detected.

True/False

Machine learning involves training machines to recognize patterns based on data. (True/False)

3.7 Activity 4 – Big Idea of Natural Interaction

3.7.1 Introduction – Theory

In this activity, you'll explore the Big Idea of Natural Interaction in the context of AI-powered puzzles and games. Natural interaction involves using intuitive and human-like ways to communicate with machines and AI systems. You will apply this concept to create an interactive, maze game that respond to light intensity sensing. The user will use light from a flashlight or other light emitting source and guide a player through a maze. The objective is to understand how AI can be integrated into games and puzzles for a more interactive and engaging experience.

Natural Interaction refers to human-computer interaction methods that are intuitive and mimic natural human behaviors. It aims to create interfaces that users can interact with in a natural manner without requiring specialized knowledge.

In AI, natural interaction involves processing and responding to inputs that are naturally occurring or intuitive, like gestures, speech, and environmental changes (e.g., light intensity). This project uses changes in light intensity as a natural and intuitive form of interaction.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.7.2 Hardware

- BBC Micro:bit
- Computer with USB cable for Micro:bit connection
- MakeCode coding environment
- A flashlight or other light emitting source

3.7.3 Setup

- Connect the Micro:bit to a computer using a USB cable.
- Open the MakeCode coding environment.

3.7.4 Exercise / Experiment 4

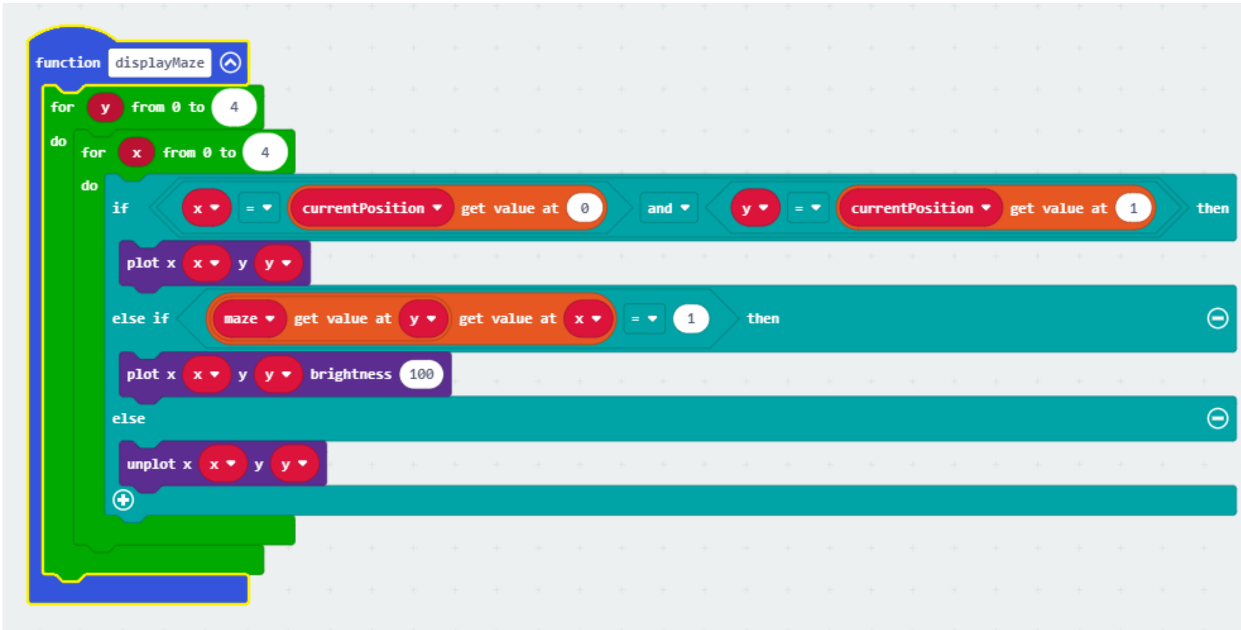
In this experiment, the students will understand the Big Idea of Natural Interaction in AI. Natural Interaction involves enabling technology to understand and respond to human input in ways that feel intuitive and human-like. To grasp the significance of Natural Interaction, consider the everyday applications of voice assistants like Siri, Alexa, or Google Assistant. They can answer questions, control smart devices, such as lights, or play music based on your commands. These examples showcase how Natural Interaction enhances our daily lives by making technology more accessible and engaging. This experiment aims to create a light-following puzzle game using a BBC Micro:bit. It will challenge the students to design a maze that responds to light intensity using the onboard light sensor. The goal of the game is to move towards the brightest light source.

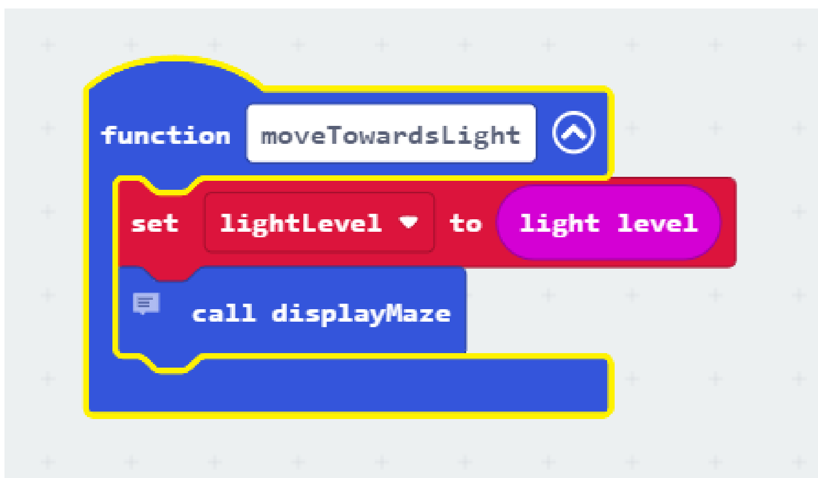
The Micro:bit displays a maze layout on its 5x5 LED grid. The player is represented by a lit LED. The user uses a flashlight to guide the virtual character through the maze while Micro:bit's light sensor detects changes in light intensity and direction, moving the character accordingly.

Each student or group should have a BBC Micro:bit, a USB cable for connection, and access to the MakeCode coding environment. Verify that all Micro:bits are connected and functioning. Use the following code:

Co-funded by the European Union

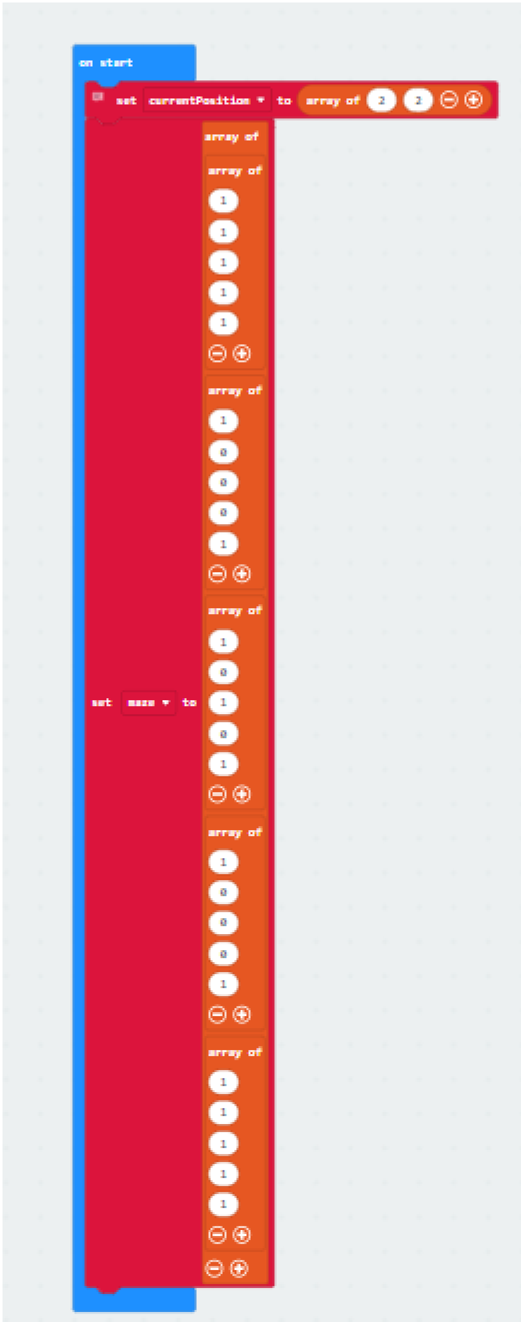
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

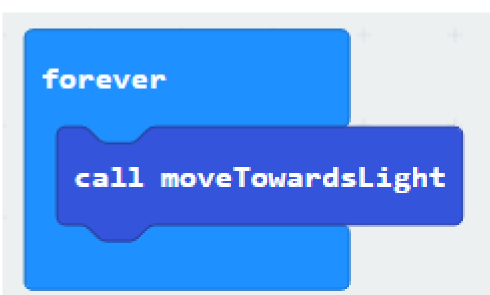




Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.





Explanation of the Code

The maze array represents a simple maze, with 1 indicating a wall and 0 indicating a free path.

- 'currentPosition' holds the x and y coordinates of the player's position in the maze.
- The 'displayMaze' function updates the LED display to show the player's position and the maze layout.
- The 'moveTowardsLight' function uses 'input.lightLevel()' to read the ambient light intensity. The function then determines the direction of increased light and updates 'currentPosition' accordingly.
- The 'basic.forever' loop continuously calls 'moveTowardsLight', making the game responsive to changes in light.

However, given the limitation on the Micro:bit's capabilities this project might not work properly in all settings and environments. It is important though to understand how a machine can interact with the natural world through a series of sensors, and then execute commands based on the received input.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.7.5 Questions

Multiple choice

How does the simplified light-following puzzle game on the Micro:bit represent the character's position?

a) By changing the color of the Micro:bit's LEDs.
b) By using a specific LED pattern on the grid to indicate the character.
c) By playing different sounds for different positions.
d) By sending messages to an external display.

In the context of AI, what does the concept of reasoning refer to in the simplified light-following puzzle game?

a) The ability of the Micro:bit to produce sound.
b) The player's ability to solve the puzzle.
c) The way the game uses light intensity to make decisions about character movement.
d) The LED patterns used to represent the maze.

True/False

Micro:bit's light sensor can accurately detect the direction of a light source, making it suitable for complex light-following navigation tasks. (True/False)

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

3.8 Activity 5 – Big Idea of Societal Impact

3.8.1 Introduction – Theory

Overview of the societal impact of AI in the context of games and puzzles:

Artificial Intelligence (AI) is a transformative force with significant societal impact. When applied to games and puzzles, AI serves as a microcosm of its broader influence on society as a whole. It affects various aspects of human life, including entertainment, learning, social interaction, and cognitive processes. Games like "The Elder Scrolls" or "Grand Theft Auto" use AI to populate their worlds with NPCs who exhibit unique behaviors and adapt to player actions. This enhances immersion and engagement.

Incorporating AI into educational games and puzzles has far-reaching implications for learning. AI can personalize content, adapting it to individual learning styles and paces. This facilitates more effective learning experiences. Platforms like Duolingo use AI to create language-learning apps that adapt to users' language proficiency levels, making language acquisition more efficient.

Multiplayer games, especially online ones, foster social interaction and community building. AI-driven matchmaking systems, chatbots, and virtual worlds contribute to the formation of online social circles. Online multiplayer games like "Fortnite" create communities of players who collaborate, communicate, and form social connections, extending the gaming experience beyond the screen.

AI can also influence how we think, solve problems, and make decisions. Puzzle-solving games, for instance, can challenge cognitive skills and strategic thinking. Games like Sudoku or chess leverage AI to generate puzzles, adapt difficulty, and even offer hints, enhancing players' problem-solving skills.

As AI technology infiltrates gaming and puzzle experiences, it shapes culture, influences behavior, and, in some cases, raises ethical and social questions. The societal impact of AI in games extends beyond mere entertainment; it affects our perception of the world and our engagement with it. Games like "Pokémon Go" encouraged players to explore their surroundings, impacting urban culture, increasing foot traffic at local landmarks, and even raising safety concerns and discussions about urban planning.

In essence, AI in games and puzzles is not confined to the boundaries of screens and boards; it's a dynamic force that extends into how we live, learn, connect, and think. Understanding the societal impact of AI in this context is essential for responsible and informed engagement with AI technology in the gaming and puzzle domain. It encourages critical thinking, ethical consideration, and awareness of AI's broader influence on society.

AI does not only affect technology but also affects society, culture, and individuals:

AI, in the context of games and puzzles, transcends its role as a technological tool. It transforms into a powerful societal agent, shaping various aspects of our lives. AI contributes to cultural shifts in gaming and entertainment. Different games reflect cultural values, norms, and stories. AI algorithms can impact how characters are represented in games, affecting cultural perceptions. The

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

representation of different cultures in games, such as the portrayal of Japanese culture in "Ghost of Tsushima," can influence how players perceive and appreciate cultural diversity.

Multiplayer games and social platforms in gaming enable global interactions. AI plays a pivotal role in the creation of virtual communities and the facilitation of social connections. Massively multiplayer online games (MMOs) like "World of Warcraft" have given rise to guilds, clans, and in-game social structures, influencing players' online and offline social lives.

AI-driven game mechanics influence player behavior and engagement. Game design, guided by AI analytics, can lead to game addiction and concerns about well-being. "Candy Crush Saga" employs AI to design levels and adaptive difficulties, influencing how players engage with the game, sometimes to the point of addiction.

Games and puzzles often reflect and reshape how players perceive reality. AI's role in procedural content generation and character behavior has implications for how players view the world. Games like "Minecraft" allow players to construct and reshape entire virtual worlds, influencing their perception of creativity and environmental dynamics.

AI can impact players' ethical decision-making and morality. Choices presented in games can challenge players' moral compass and influence their thinking. Games like "The Walking Dead" force players to make ethical decisions that reflect their personal values and impact the game's narrative, raising ethical dilemmas.

In essence, AI in games and puzzles extends its influence beyond technology, permeating culture, social interactions, individual behaviors, and even the way people perceive the world. It acts as a societal mirror, reflecting and shaping the values and concerns of the society it exists within. Understanding these broader implications of AI in gaming and puzzles is vital for both game developers and players, as it encourages ethical considerations, cultural sensitivity, and informed engagement with AI-powered experiences.

The Broader Impact of AI:

AI is not solely about enhancing convenience and efficiency; it carries profound implications for society, culture, and individuals. These implications extend to various aspects of AI in games and puzzles. AI algorithms used in games and puzzles may exhibit biases, such as gender or racial bias. It's essential to question the fairness of AI-driven game mechanics and ensure that everyone has an equal and unbiased gaming experience. If an AI-driven puzzle game consistently provides hints or easier levels to one group of players based on their profile data, it raises fairness concerns.

Discussing AI's accountability is vital. When AI influences gaming outcomes, it's crucial to understand who is responsible for the design, performance, and consequences of AI-driven game features. If a game's AI algorithm promotes aggressive in-game behavior that leads to negative player experiences, accountability questions may arise.

AI's role in games has implications for the overall harmony of society. Games can either promote positive, cooperative behavior or reinforce negative stereotypes and competitive attitudes. Games like "Animal Crossing" promote social cooperation and harmony within a virtual community. Discussing

Co-funded by the European Union

these ethical aspects is essential for informed decision-making. Game developers, players, and society at large must be aware of the ethical dilemmas and considerations related to AI in games and puzzles.

Ethical and cultural considerations directly impact the user experience. Games that are sensitive to ethical issues and cultural diversity tend to offer a more enjoyable and inclusive experience. Games that include diverse and culturally accurate characters and storylines often receive positive feedback from players who appreciate representation.

AI-powered games can shape and reflect cultural and societal trends. They have the potential to challenge stereotypes, encourage inclusivity, and contribute to meaningful conversations. Games that explore complex social issues, like "This War of Mine," encourage players to reflect on the human cost of war, promoting discussions about empathy and social responsibility.

In summary, the ethical, social, and cultural aspects of AI in games and puzzles are integral to responsible development and inclusive gaming experiences. Discussions in these areas are essential for ensuring fairness, accountability, and societal harmony and for making informed decisions about the role of AI in gaming. It's crucial to emphasize that the impact of AI goes beyond technology and extends into the realms of ethics, culture, and society.

3.8.2 Exercise / Experiment 5

This exercise is a group discussion where students will explore and debate the societal impact of AI in games and puzzles. Students should be divided into small groups to discuss and share their thoughts on various aspects related to AI in games and puzzles.

Learning Objectives & Outcomes:

- Understand the societal implications of AI in games and puzzles.
- Develop critical thinking skills by analyzing ethical and cultural aspects of AI.
- Encourage students to express their opinions and engage in constructive discussions.

Topics:

Divide the discussion into the following key topics, and assign each group a different topic to explore:

- Ethical Considerations: Discuss the ethical dilemmas of AI in games, such as AI behavior in multiplayer games, fairness, and responsible AI design.
- Cultural Influence: Explore how AI-driven games can reflect or impact different cultures and traditions.
- Accessibility: Consider how AI can be used to make games more accessible to individuals with disabilities.
- AI in Education: Discuss the use of AI in educational games and its impact on learning.
- AI in Game Development: Explore how AI is used in the development process of games and its implications for the gaming industry.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

After group discussions, bring the class together for a whole group discussion. Each group can briefly present their findings and insights to the entire class. Encourage open dialogue, questions, and cross-topic discussions.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.