Introduction to artificial intelligence, coding and chess – T2T course

Funded by
the European Union

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.
Project code 2022-2-IT02-KA210-SCH-000095953

License
This intellectual production was marked with CC BY-NC-ND 4.0
International license by: Centro Asteria-Istituto Suore di Santa Dorotea di Cemmo
You can write to us at the following address: cultura@centroasteria.it

Attribution-NonCommercial-NoDerivatives 4.0
International

T2T "Chess-Coding-AI" Course

Introduction

Welcome to the Erasmus+ project "Introduction to artificial intelligence, coding and chess for primary schools". This course for teachers gives you the qualification to give an introduction to chess, coding and artificial intelligence to 10–12-year-old primary school students. The course prepares you to teach 13 lessons combining the three subjects, all relevant to the STEM field. The part on artificial intelligence is inspired by the CSER-AI[1] course at the University of Adelaide, Australia, and several links to CSER were used for our course - mostly YouTube videos realized by CSER.

The Computer Science Education Research Group (CSER), based at the University of Adelaide in the School of Computer Science, aims to promote research and study in the field of computer science education. Link: https://CSERmoocs.adelaide.edu.au/about-us .

This course for teachers comprises **six modules**:

module 1:	general information on the project
module 2:	chess basics (referred to students manuals)
module 3:	the basics of programming
module 4:	the basics of AI
module 5:	learning scenarios (13 lessons)
module 6:	teacher feedback.

---

[11] *We have inspired many approaches to introduction to AI for primary school teachers and offer links to videos and other online resources for teachers and students suggested and/or produced by the CSER Group. This framework has been produced under a Creative Commons [CC BY-NC] licence.*

Funded by
the European Union

Overview

1.1 Course information
1.2 Teaching strategy and methodology
1.3 The curriculum
1.4 Evaluation and verification

2.1 Chess Basics, Part 1 (lesson one student's manual)
2.2 Chess Basics, Part 2 (lesson two student's manual)
2.3 Chess Basics, Part 3 (lesson three student's manual)
2.4 Chess Basics, Part 4 (lesson four student's manual)

3.1 Programming in Swedish and Italian primary schools.
3.2 History of computers and coding
3.3 How the computers work
3.4 The principles of programming: SARA - unplugged
3.5 Coding and robots in the classroom

4.1 Introduction to AI
4.2 Automatic Learning
4.3 Ethics
4.4 The future of AI
4.5 Biases

5.1 Overview of learning scenarios
5.2 Chess Lessons
5.3 Programming [*coding*] lessons
5.4 Lessons on AI

6. Teacher feedback
6.1 Focus groups
6.2 Feedback to experts

1.1 The project

The context

Chess has always been an area of interest in terms of technological development, partly because it has been the number one IQ [Intelligence Quotient] game and partly because it is a perfect testing ground as it is finite in nature, but at the same time with almost infinite possibilities. Both the struggle to create the first computer capable of beating a world chess champion (Deep Blue in 1997) and the first AI capable of beating the best chess computer (Alpha Zero in 2017) have driven digital development.

As digital development advances, the education system has an increasingly important mission to provide children with knowledge about how computers and artificial intelligence work. In a number of countries, this mission has been incorporated into school curricula, but efforts are still being made to find educational approaches that effectively inspire and transmit knowledge. In this case, we believe our mix could be particularly interesting, not least because our experience suggests that chess works well in the classroom, as everyone can participate on an equal footing.

The idea

With this project, we want to introduce artificial intelligence, coding and chess into primary school. We believe that the learning objectives of all three subjects (artificial intelligence, coding and chess) are - transversally and jointly - of absolute relevance to the STEM disciplines.

The course for teachers

As part of this project, we developed a course for teachers lasting a total of 20 hours, including 6 hours in the presence of one of the project experts and 14 hours of self-study through online activities.

The project is a pilot study preparing a larger project for the years 2024-2026. The feedback from teachers in this pilot project is very valuable for the development on a larger scale during the project to follow.

The project is funded by the European Union through the Erasmus+ programme.

Contact

Project Leader:

Sister Elisabetta Stocchi, Asteria Cultural Centre: direzione@centroasteria.it

Experts:

Alessandro Dominici: dominici@alfierebianco.com wingolab64@gmail.com (Italy)
Jesper Hall: info@jesperhall.se (Sweden)
Rolf Niemann: rolf.niemann@vattenhallen.lu.se (Sweden)
Ilaria Piccozzi: ilaria.piccozzi@gmail.com (Italy)

Project website:

https://www.centroasteria.it/

1.2 Teaching strategy and methodology:

The integration of STEAM subjects (science, technology, engineering, art and mathematics) into primary education offers students a well-rounded learning experience that promotes critical thinking, problem solving and creativity.

The aim of this project is to introduce primary school students to the fundamentals of chess, coding and artificial intelligence, while promoting critical thinking, computational thinking and collaboration skills.

The following **general learning objectives** guide the project:

Developing chess skills: Students will learn basic chess rules, terminology and strategies and practise chess training, fostering logical thinking, spatial reasoning and decision-making skills.

Introduction to programming concepts: Students will be introduced to basic *coding* concepts, such as algorithms, sequences, alternatives, repetitions and problem solving, to develop computational thinking skills.

Understanding the Principles of Artificial Intelligence: Students will explore the concepts of Artificial Intelligence (AI) and understand how it is applied to chess, delving into the role of algorithms and machine learning.

In particular, **the development** of **key competences** such as **critical thinking** (AI, Chess), **strategic thinking** (AI, Chess, Coding), **problem-solving** skills (AI, Chess, Coding), **sensitivity to the ethical risks associated with AI**, the ability to **embrace fair-play** (Chess) in a well-regulated competitive environment, **time management** (Chess) **mathematical logic** (Maths, AI, Chess), the **ability to concentrate (Maths, Coding**, Chess), **memory training** (Maths, Coding, Chess), the habit of **considering consequences before acting** (Chess), and the ability to **predict and plan** (AI, Chess, Coding) **are all essential skills to meet the challenges of today's digital world.**

The **teaching method is** based on *play*, *cooperation*, *chess* and *problem solving*. By stimulating children's curiosity and desire to learn, we want to open the door to learning about the digital revolution that has already begun and in which we live today.
Lessons are conducted with a mix of games, classical teaching and digital classroom teaching.

Game-based learning: Use chess as a basis for teaching various concepts. Introduce the game through interactive and engaging activities, such as mini-games or riddles, that allow students to explore the rules and strategies of chess.

Practical programming activities: Incorporate accessible and age-appropriate programming activities for primary school students. Use *'unplugged'* programming [without computers], which enables students to understand the principles of coding.

Collaborative learning: Promote teamwork and collaboration by assigning group exercises or pairing students with different skills. Encourage peer learning and problem solving through group discussions and activities.

Research-based approach: Encouraging students to ask questions, investigate and discover solutions independently. Provide opportunities for exploration and problem-solving related to chess, coding and artificial intelligence.

Technology integration: Use digital tools and resources to enhance learning experiences. Introduce software or online platforms for chess practice and analysis.

Formative assessment: Incorporate formative assessments during the project to monitor student progress. Use observations, class discussions and small quizzes to assess understanding and provide timely feedback.

Project Implementation

The aim of the project is to develop **13 lessons of 90 minutes each** for fifth grade students (approx. 11 years old) that provide students with *basic knowledge* on:

- *What a scientific approach means*
- *How humans, computers and artificial intelligence work*
- *The basics of programming*
- *The uses of AI today and in the near future*
- *The risks and opportunities of AI*

and to develop chess skills, and also the other skills that chess trains, and through this training provide insights into how human learning occurs.

Introduction to chess: start by introducing students to the basics of the game of chess, including the board, the pieces and their movements. Use visual aids, interactive demonstrations and stories to engage students and stimulate their interest in the game.

Programming Fundamentals: Introducing programming concepts through simple and interactive activities focusing on 'unplugged' programming (without the use of a computer).

Introduce the concept of AI in a simplified way. Discuss how AI is used in chess, highlighting its ability to analyse moves, suggest strategies and play against human opponents. Use age-appropriate examples and images to make the concept accessible.

Reflection: Conclude the project learning sessions with *student reflections*. Students can explain their programming choices and reflect on their learning journey.

1.3 AI in the curriculum in Sweden and Italy

In Sweden

Skolverket (the Swedish National Agency for Education) is responsible for the development and implementation of the national curriculum and educational guidelines in Sweden. As of September 2021, my closing date, Skolverket has not published any specific guidelines or frameworks dedicated exclusively to AI education in primary school. However, it does provide

Funded by
the European Union

general guidelines that encourage the integration of digital skills, computational thinking and *problem-solving* skills in various subjects, which can indirectly support AI education.
Skolverket emphasises the importance of incorporating digital skills in the classroom to prepare students for a digitised society. It promotes the integration of technology and computational thinking as tools for learning and creativity. This includes the use of digital resources, programming and *problem-solving* exercises to improve students' logical reasoning and algorithmic thinking skills.
Although there are no specific guidelines on AI education, Skolverket's focus on digital literacy and computational thinking provides educators with a basis for introducing AI concepts and related topics in primary school. Educators can use these guidelines to incorporate discussions on AI, ethics and related concepts into the broader scope of digital literacy and computational thinking.

1.4 In Italy

The Italian school system does not have a specific curriculum or guidelines dedicated exclusively to AI education in primary school. However, efforts are being made to integrate digital skills and computational thinking into the curriculum, which indirectly includes elements of AI education.

In the Italian school system, the use of digital technologies and ICT (Information and Communication Technologies) is encouraged in several subjects. The school curriculum promotes the development of digital competences, including basic computer skills, digital literacy and problem-solving skills. These skills form the basis for understanding and exploring AI concepts.

The National Indications of the Italian School System refer to the eight key competences of 2006/962/EC, including mathematical competence, technological competence and digital competence. The document emphasises the importance of mathematical practice being approached through problem-solving, to be understood as authentic and meaningful questions related to everyday life. As far as primary school is concerned, importance is given to the use of games, emphasising the crucial role they can play in communication, in education to respect shared rules and in the development of strategies suited to different contexts.

As far as 'Digital Citizenship' is concerned, Annex A of Law 2019/92 on the teaching of civic education in schools refers to the importance of developing in students (with appropriate and diversified strategies starting from pre-school) the ability to make conscious use of virtual media. It emphasises the importance of acquiring useful information and skills to improve the new way of being in the world and of making young people aware of the risks and pitfalls of the digital environment, while also considering the consequences on a practical level. Since it is not only a question of the knowledge and use of technological tools, but also of the type of approach to them, tackling digital citizenship education has become a professional commitment involving all the teachers in the class and has therefore been extended to all disciplines.

The Law's Guidelines provide for the identification of competence goals at the end of the first cycle of education and state that students must be able to distinguish between different devices and use them correctly, respect correct online behaviour and navigate safely. He/she must be

Funded by the European Union

able to distinguish digital identity from real identity and apply privacy rules, protecting himself/herself and the collective good.
They must be able to argue through different communication systems and become fully aware of digital identity as an individual and collective value to be preserved.

Regarding programming [*coding*] and computational thinking, the recent National Indications and New Scenarios (MIUR 2018) have given even more emphasis to the role they should play in today's schools. Indeed, Decree-Law No. 152 on the implementation of the National Recovery and Resilience Plan 2021 addresses the development of digital skills to foster and enhance learning. The National Plan for the Training of Teachers in Schools of All Sorts and Grades stipulates that, starting in the school year 2022/2023 and for a period of three years, the learning approach of computer programming (*coding*) and digital education will be identified as a national priority.

It is envisaged that, by the school year 2024/2025, a decree of the Minister of Education will integrate the specific learning objectives and competence targets of the National Indications in order to achieve, starting from the school year 2025/2026, in schools of all levels, the development of digital competences, also by encouraging the learning of computer programming (*coding*), within the existing teaching.

In addition, some schools in Italy have started to introduce coding and robotics as part of their STEM (Science, Technology, *Engineering*, and Mathematics) education initiatives. These programmes often deal with the basic principles of coding, logical thinking and problem solving, which are fundamental skills for AI education. It is important to note that educational policies and initiatives may vary between different Italian regions or provinces. Therefore, it would be useful to consult the official websites of the Ministry of Education and Merit and the *regional school authorities* for the most up-to-date information on AI education initiatives in primary schools.

Finally, given the dynamic nature of educational policies and the growing importance of AI, it is possible that specific guidelines or frameworks for AI education in primary schools will be introduced in the near future. Staying informed about updates from the Ministry of Education and engaging with professional networks and educational communities can help educators stay *up-to-date* with the latest developments in AI education, both in Italy and Sweden.

Funded by
the European Union

1.6 Evaluation and verification

We are planning to develop, together with the teachers involved, age-appropriate grids in line with the project objectives. Evaluate students on their understanding of chess rules and strategies, programming skills and incorporation of artificial intelligence concepts.

Encourage students to provide *constructive feedback* to their peers' group activities. This promotes active engagement and reinforces learning outcomes.

Provide opportunities for students to reflect on their learning experience. Encourage them to identify their strengths, areas for improvement and their understanding of the connections between chess, programming and AI.

Evaluation grids are proposed for each lesson, but only as guidelines for teachers: **quantitative evaluations will NOT be carried out in this project, as it is a pilot phase**.

1.7 Teacher Certificate

After completing the teacher training course (20 hours) and teaching most of the 13 lessons, you will obtain a certificate that can be used as part of your teacher training.

Funded by
the European Union

Module 2:

2.1 Chess Basics, Part 1 (lesson one student's manual)
2.2 Chess Basics, Part 2 (lesson two student's manual)
2.3 Chess Basics, Part 3 (lesson three student's manual)
2.4 Chess Basics, Part 4 (lesson four student's manual)

Funded by
the European Union

Module 3:

In this module, we will give you a brief introduction to computers and how coding works.

3.1 Programming in Swedish and Italian primary schools.

Programming and digital skills have become increasingly important in education around the world due to the growing importance of technology in various aspects of modern life. Recent years have seen a global push towards the introduction of programming and digital literacy in primary school curricula. Italy and Sweden are no exception, recognising the importance of equipping students with digital skills from an early age to prepare them for the challenges of the 21st century.

Here are some key points:

A. **Digital skills**: Both countries have placed a strong emphasis on promoting digital skills among their citizens. This includes the introduction of programming and coding in the primary school curriculum to prepare students for a future pervaded by technology.

B. **Curriculum integration**: Programming can be integrated into different subjects or taught as a stand-alone subject in primary school. For example, it can be combined with mathematics to illustrate concepts such as algorithms, or with design and technology to explore creative applications.

C. **Visual programming languages**: To make programming accessible to younger students, primary schools could use visual programming languages, such as Scratch, which allow students to create programmes using graphical blocks rather than traditional text-based programming.

D. **Problem-Solving and Computational Thinking**: Programming education often focuses on developing problem-solving and computational thinking skills. These skills can also be valuable for students who do not intend to pursue a career in computer science.

E. **Teacher training**: Primary school teachers may have received specialised training or resources to teach programming effectively, especially if they have no computer training.

F. **Extracurricular activities**: In addition to regular classroom lessons, schools could offer extracurricular activities, programming clubs or competitions to further engage students in programming.

3.2 History of computers and coding

The history of computers spans several centuries, beginning with the conceptualisation of computing devices and culminating in the advanced computers we use today.

Here is a brief overview of this fascinating journey.

3.2.1 Ancient calculus concepts (3000 BC - 1600 BC):

- Ancient civilisations, such as the Egyptians and Greeks, developed the first calculating devices, such as the abacus and the Antikythera mechanism.

- Mathematicians such as Blaise Pascal and Gottfried Leibniz made significant contributions to mechanical calculators during the 17th century.

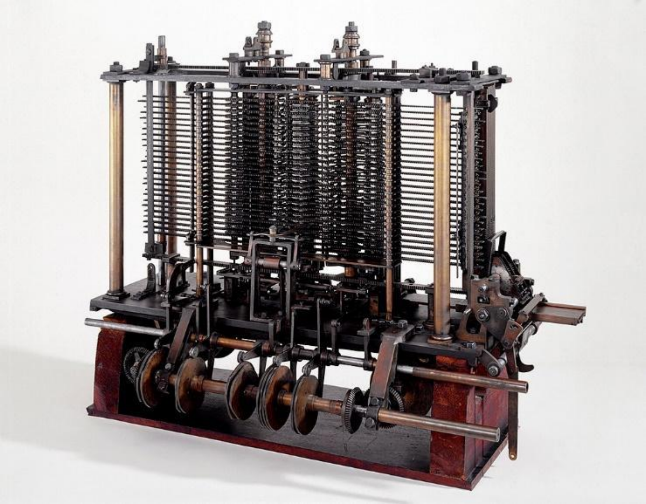
3.2.2. Mechanical computers (1600-1930):

- Charles Babbage, an English mathematician, designed the *Analytical Engine* in the 1830s, considered the precursor of modern computers. However, it was never completed.

- Mechanical calculators, such as the arithmometer and the Marchant calculator, were widely used during this period.

Charles Babbage (1791 - 1871) was an English polymath, mathematician, philosopher, inventor and mechanical engineer. Babbage originated the concept of the *programmable* digital computer.

Babbage is considered by some to be the 'father of the computer'. Babbage is credited with the invention of the first **mechanical** computer, the Difference Engine, which later led to more complex electronic designs, although all the essential ideas of modern computers can be found in Babbage's **Analytical Machine** (see image below, from Wikimedia Commons, the free media repository).



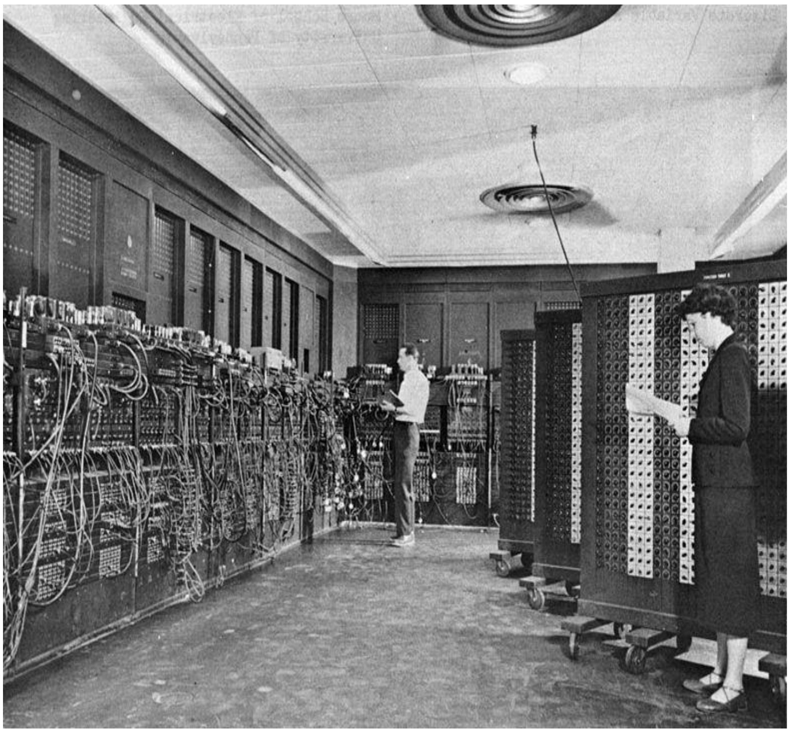
3.2.3 Electromechanical and valve computers (1930s-1940s):

The need to quickly calculate complex mathematical problems such as the ballistics of a projectile like a cannonball forced the development of ever more advanced mechanical machines. In the 1920s and 1930s, the invention of electrical components made possible the invention of the first computers as we know them today.

- In the 1930s, the invention of electromechanical computers, such as the IBM Harvard Mark I and the Atanasoff-Berry Computer (ABC), laid the foundation for digital computing.

- The 1940s saw the development of the first electronic computers, such as the Electronic Numerical Integrator and Computer (ENIAC), which used vacuum tubes for processing.

The ENIAC (electronic numerical integrator and computer), designed in 1943-46, was the first large-scale computer. (https://it.wikipedia.org/wiki/ENIAC)



The computer filled a room of almost 200 m^2 , weighed 30 tonnes and generated 130 kW of electrical power; programmers stood inside the computer room and programmed by moving cables to different positions.
ENIAC could perform 5,000 calculations per second. Compared to a human being, the computer was 5,000 times faster. Today we talk about GHz, which means 1,000,000 calculations per second.

3.2.4. Transistors and integrated circuits (1950s-60s):

- The invention of the *transistor* in 1947 revolutionised computing, replacing bulky vacuum tubes and leading to smaller, more reliable computers.

- The 1950s and 1960s saw the emergence of mainframe computers, such as the IBM System/360 series, capable of performing complex calculations and processing data.

3.2.5 Microprocessors and Personal Computers (1970s-1980s):

- The development of microprocessors, integrated circuits containing multiple transistors, enabled the creation of small computers.

- In 1975, the Altair 8800, an early microcomputer kit, was followed by the introduction of the Apple II and the IBM PC, marking the beginning of the popularity of personal computers.

The invention of the semiconductor transistor in the early 1960s was the starting point for the operation of today's computers.
In 1970, the personal computersth were designed, which represented the beginning of the use of computers both at work and at home.

Today, almost everyone has an advanced computer in their hands: the mobile phone.

A lot of information on the history of computers can be found on the Internet. https://it.wikipedia.org/wiki/Storia_del_computer ,
https://it.wikipedia.org/wiki/Storia_dell%27informatica
https://www.computerhistory.org/timeline/computers/

3.2.6 Advances in information technology (1990s-present):

- The 1990s saw the emergence of graphical user interfaces (GUIs) and the rapid growth of the Internet, which transformed computers into powerful tools for communication and access to information.

- The 2000s brought significant advances in processing power, storage capacity and connectivity with the advent of laptops, smartphones and cloud computing.

- The current focus is on artificial intelligence, machine learning, quantum computing and the development of more efficient and specialised information technologies.

This overview highlights some milestones in the history of computers, but it is important to note that over the years there have been countless innovations and contributions from numerous individuals and organisations that have shaped the evolution of computing as we know it today.

A worksheet (*the history of computers*) is attached.

3.2.7 Information on programming and *coding*

The history of programming dates back to the dawn of computing and evolved in parallel with the development of computers. Here is a brief overview of its milestones:

3.2.7.1. Early programming concepts (1800-1940):

- Ada Lovelace, in the mid-1800s, is credited with writing the first algorithm for Charles Babbage's analytical machine, thus becoming *the world's first woman programmer*.

Ada Lovelace is often recognised as the first programmer in history, when she designed the first algorithm for C. Babbage's analytical machine in 1843.

- Early programmes were executed using machine language, which involved direct manipulation of the computer hardware through binary instructions.
- In the 1940s, the *assembly* language was developed, which provides mnemonic symbols to represent machine instructions and makes programming more accessible.

3.2.7.2. High-level programming languages (1950s-70s):

- The 1950s saw the birth of high-level programming languages such as *Fortran* (Formula Translation), which introduced a more human-readable approach to programming.
- In the 1960s, programming languages such as *COBOL* (Common Business-Oriented Language) and *ALGOL* (Algorithmic Language) were developed, dedicated to specific domains.
- The 1970s brought notable programming languages such as *C*, which offered greater control over hardware, and *BASIC*, designed for beginners.

3.2.7.3. Structured programming and object-oriented programming (1960s-1980s):

- The paradigm of structured programming, popularised by languages such as *ALGOL, Pascal and C*, emphasised the use of control structures and modular programming techniques.
- In the 1980s, object-oriented programming (OOP) became established with languages such as *Smalltalk*, *C++* and later *Java,* which allow code reuse, encapsulation and inheritance.

3.2.7.4. The rapid development and the Internet era (1990-2000):

Funded by the European Union

- The 1990s saw the rise of rapid application development (RAD) tools, such as *Visual Basic* and *Delphi*, which enabled faster software development.

- The Internet boom led to the development of web-centred programming languages, such as *JavaScript* and *PHP*, which allow the creation of dynamic web pages and *server-side scripting*.

- Object-oriented scripting languages, such as *Python* and *Ruby*, have gained popularity due to their simplicity and readability.

3.2.7.5. Modern programming paradigms (2000s-present):

- The 2000s saw the emergence of new programming languages, frameworks and paradigms. *Python, Ruby and Java* continued to be widely used, while languages such as *Go*, *Rust* and *Swift* gained popularity.

- Functional programming languages such as *Haskell and Scala* have gained ground, emphasising immutability and *declarative programming*.

- Domain-specific languages (DSLs) and scripting languages have become prevalent because they respond to specific tasks and allow automation.

- In addition, advances in *cloud computing*, big data, machine learning and artificial intelligence have led to the development of specialised tools and libraries for these areas.

This overview highlights some significant milestones in the history of programming. As the field of programming is vast and constantly evolving, numerous languages, frameworks and paradigms have emerged that have shaped the way software is developed and used in various fields.

Computers only understand machine code, often in binary notation ('1' or '0'), which is difficult for humans to understand. In 1952, Grace Hopper developed the first 'compiler', a translation from a much simpler programming language to machine code.

Since then, many different programming languages have developed. Today, there are a couple of thousand languages. Some examples are *Python* and *Java*.

On the history of coding you can find some interesting facts here:
https://en.wikipedia.org/wiki/Computer_programming [Eng].
https://it.wikipedia.org/wiki/Programmazione_(informatics) [Ita].

Here is a short illustrative document that can be used in class to talk about the history of computers and programming: .ppt **Time Line of Computers**.

3.3 How does a computer work?

Here is an explanation of how computers work, designed especially for primary school teachers to share with their students.

A computer is a complex machine that can do many different things, such as help us learn, create and play. Let's explore the main components and how they work together:

3.3.1. **Input devices**: Just as our senses allow us to interact with the world, a computer needs *input devices* to receive information. Examples of input devices are the keyboard, mouse and touch screen. When we press the keys on the keyboard or move the mouse, the computer receives these signals as input.

3.3.2. Central Processing **Unit (CPU)**: The CPU [*Central Processing Unit*] is like the brain of the computer. It receives input from input devices and *processes* (or *reasons* about) the information. It performs calculations, makes decisions and remembers things. The CPU is made up of small electronic parts that work together to perform these tasks quickly and accurately.

3.3.3. **Memory**: computers have a memory, just like our brain. It is where they store and retrieve information. There are two main types of memory: RAM (Random Access Memory) and Storage Memory. RAM is like a short-term memory that the computer uses to work quickly with information. Storage memory, such as a hard disk or solid-state drive, is a long-term memory in which the computer stores files and programmes for later use.

3.3.4. **Output devices**: After processing the information, the computer must display the results. *Output devices* help in this respect. Examples of output devices are the monitor (or screen), speakers and printers. The monitor displays images and text, the speakers play sounds and the printer can create hard copies of what we see on the screen.

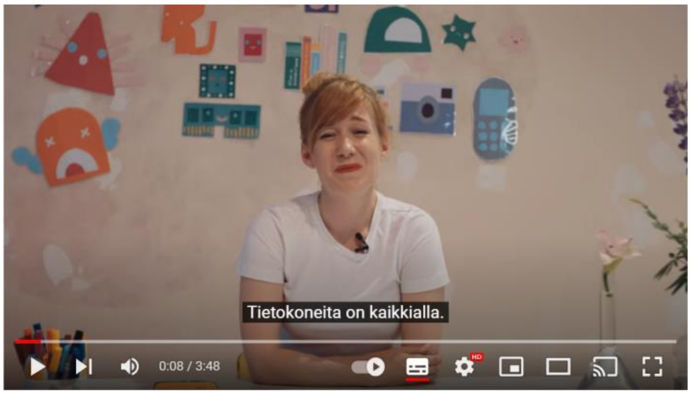
3.3.5. **Software**: Computers work because of software, which is *a set of instructions or programmes that tell the computer 'what to do'*. Just as we follow instructions to complete tasks, computers follow software instructions. Software includes operating systems (such as Windows or macOS) and the programmes we use, such as word processors, web browsers and educational software.

3.3.6. **Data and information**: Computers work with data, which is raw information such as numbers, text, images and sounds. When the computer processes data, it turns it into meaningful information that we can understand. For example, if we enter numbers in an operation, the computer can perform the calculations and give us the result of the operation.

In general, computers work by receiving input from devices, processing it in the CPU, storing and retrieving information from memory and presenting the results through output devices. They rely on software programmes to perform various tasks and transform data into useful information.

Remember that this is only a simplified explanation of how a computer works. Computers are incredibly complex machines, with many intricate parts and processes. But understanding these basics can help us appreciate the technology we use every day!

A very nice film recommended on YouTube by Linda Luikas. It talks about "what is a computer?" in child-friendly language. Link : https://www.youtube.com/watch?v=9lVKcTNVD1Q



Tietokoneita on kaikkialla.

What is a computer?

The computer uses numbers and each number has its own meaning. For example, the number 42, in binary form, is written 00101010.

If you want to learn more about binary form, we provide a simple worksheet - **'binary numbers'.** Feel free to use it in your maths lessons.

We could also return to mathematics by studying the number of seeds that can be placed on a chessboard by starting with one seed on the first square and doubling the number at each step, a classic problem that is often cited.

3.4 How a Chess Computer Works

A *chess computer* is a special type of computer designed to play chess. It uses its programming and algorithms to make moves and play against human opponents.

3.4.1 Chessboard representation: A chess computer starts with a representation of the chessboard and the pieces in its memory. It uses a special system that assigns a unique value to each square on the chessboard and each chess piece. This representation allows the computer to understand the 'current' state of each game.

3.4.2 Move generation: The chess computer uses algorithms to generate possible moves based on the current position. It analyses the positions of all the pieces and calculates all the legal moves each piece can make. This process helps the computer create a list of potential moves to consider.

3.4.3 Evaluation: After generating a list of possible moves, the chess computer evaluates the potential outcomes of each move. It considers factors such as capturing an opponent's piece, protecting its own pieces, controlling important squares and creating threats. The computer assigns a value to each move based on its evaluation.

3.4.4 Search algorithms: To choose the best move, the chess computer uses *search algorithms*. These algorithms explore different moves and their consequences by looking at future moves in the game. They simulate possible variations of the game and use a scoring system to estimate the desirability of different moves.

3.4.5 Decision-making process: Based on the evaluation and search algorithms, the chess computer selects the best move from all those analysed. It takes into account factors such as the potential point gains, risks and strategic advantages of each move. The computer aims to choose the move that gives it the best chance of winning or playing a strong game.

3.4.6 Execution of the move: once the best move has been determined, the computer updates the chessboard representation to include the move it intends to execute. It also communicates the move to the user, whether it is displayed on a screen or communicated through a voice or text interface.

3.4.7 Human-computer **interaction**: Chess computers are often equipped with a user interface that allows human players to interact with them. This can be a physical chess board with sensors that detect human moves or a graphical interface on the computer screen.

In summary, a chess computer uses algorithms, programming and evaluation techniques to analyse the situation on the board, generate and evaluate moves and select the best move based on its calculations. In this way, it can play chess against human opponents and offer a challenge to players of all levels.

Chess computers have evolved over time, becoming more powerful and sophisticated in their decision-making processes. However, in their essence, they follow these basic principles for playing chess.

3.5. Artificial intelligence in chess is a computer programme designed to play chess against human opponents**.** It uses algorithms and advanced techniques to make strategic moves and play the game.

3.5.1 Data and learning: A chess artificial intelligence needs a lot of data to learn from. It uses historical chess games, grandmaster games and other sources to understand different chess strategies and *patterns.* This data helps the artificial intelligence understand how strong players execute moves and develop winning strategies.

3.5.2 Machine learning: Chess artificial intelligence often uses a technique called *machine learning* to improve its skills. It uses algorithms that allow the AI to learn from collected data. Through this learning process, the AI analyses *patterns*, evaluates moves and identifies winning strategies.

3.5.3…Evaluation functions: Chess artificial intelligence also uses evaluation functions to assess *the strength of a* chess *position.* These functions consider various factors such as material advantage (number of pieces captured), centre control, piece activity, king security and pawn structure. The artificial intelligence assigns a higher or lower numerical value to each position to estimate how favourable or unfavourable it is.

3.5.4 Search algorithms: To decide on the best move, chess artificial intelligence uses search algorithms. These algorithms help the AI explore different moves and their potential consequences by looking ahead several moves into the future of the game. The AI simulates various game scenarios, evaluates each position and estimates the most promising moves.

3.5.5 Deep neural networks: Some modern artificial intelligence *models* for chess use deep neural networks, which are advanced mathematical models inspired by the human brain. These networks can process large amounts of data and learn complex *patterns.* They help artificial intelligence to analyse chess positions, evaluate moves and make strategic decisions.

3.5.6 Decision-making process: Based on evaluation functions, search algorithms and neural networks, the chess artificial intelligence selects the move it considers strongest. The goal is to choose the move that offers the best chance of winning or playing a *strong game*, based on its calculations and learned *patterns*.

3.5.7 User interface: Chess AI often has a user interface that allows human players to interact with it. This can be a graphical interface on a computer screen, a mobile application or an online platform where players can make moves and see the AI's response.

To summarise, a chess AI learns from data, uses machine learning algorithms, evaluation functions and search algorithms to analyse chess positions and make strategic moves. It harnesses the power of advanced mathematics and computer science to play at a high level.

Artificial intelligence in chess is constantly improving thanks to research and technological advances. While it can be a great challenge, it also offers human players the opportunity to learn and improve their chess skills.

3.6 The Basics of Programming (SARA)

What is programming?
When you write a programme you tell the computer '*what to do*'. It can be compared to a *recipe for a cake*. When executing a code, the computer simply follows the description.
As already mentioned, there are several programming languages. We will start with 'unplugged' programming, in which simple symbols or words are used, without using a computer.

The basics of programming: SARA
SARA stands for **sequence, alternative, repetition and abstraction**. Good comparisons can be made with the basic operations of arithmetic (plus, minus, multiplication, division).

To begin with, we need commands.
Caution. Since all programming languages use English, we will also use commands in English. Instead of saying "stand, jump, turn, wave, say *'Hello'*, sit", we will use "stand, jump, turn, wave, say *Hello*, sit".

With these commands we write our first programme:

```
stand
jump
turn
wave
"Hello"
sit
```

It is a **sequence of** commands.

The next one is about **alternatives**: If you have black shoes, you have to jump, otherwise you have to turn.

Funded by the European Union

```
If (black shoes) {jump} else {turn}
```

One possible programme could be this:

```
stand
wave
"Hello"
If (black shoes) {jump} else {turn}
sit
```

In the case of many repetitions, it is better to use a new command: **repeat**

```
jump
turn
jump
turn
jump
turn
```

It will become:

```
repeat(3) {jump;turn}
```

And a programme can be:

```
stand
repeat(3) {jump;turn}
wave
"Hello"
sit
```

Abstraction is used when you want to use the same code several times.
For example, if we simultaneously

```
wave
"Hello"
```

It can be called that:
Greeting

```
Def greeting = {wave; "Hello"}
```

Now the computer *understands* when you write a greeting in the code and executes it.
Without abstraction:

```
Stand
wave
"Hello"
repeat(3) {jump;turn}
wave
"Hello"
sit
```

With abstraction:

```
Def greeting = {wave; "Hello"}
Stand
greeting
repeat(3) {jump;turn}
greeting
sit
```

In *coding* lessons, we will examine these basics and practise them in different ways, especially on the blackboard.

This part can be found in the following ppt.
SARA - unplugged [without computer].

You can find some useful videos on YouTube:

Information on the use of control boards: https://youtu.be/IWJNMF7p-fM/IWJNMF7p-fM

Sequence with stand, jump, wave,..: https://youtu.be/RDbaOvE3DmE

The commands forward, right, left,... https://youtu.be/_ec25bEjuL0

How to move on the chessboard: https://youtu.be/y8N55vivQQY.be/y8N55vivQQY

3.7 *Coding* and robots in the classroom

Only *unplugged programming* will be performed during the lesson. If you want to *go further* in class, you can introduce programming in KOJO (link https://kojo.in/ for download) Kojo is an open source learning environment available as a desktop application for Linux, Windows and Mac and as a web application that runs on any modern Internet browser.

Download *Kojo's challenges* in other languages:

- **Challenges with Kojo** (pdf 2 MB, new tab) [in English].
- **Challenges with Kojo** (pdf 2 MB) [in Italian].

There is a web application where you can use the commands *forward*, *left*, *right* to draw a turtle.
Many other commands can be used in the desktop version.
If you write: //#include/chess.kojo in the first line, you can move on a chessboard.

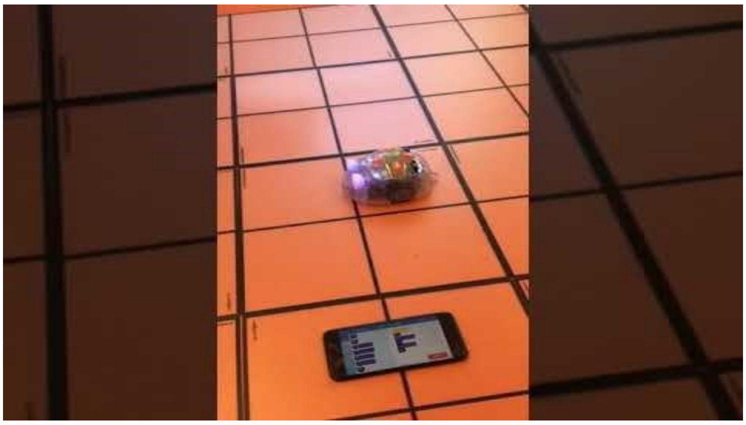
If you like **using robots in the classroom, there is** the popular ***bluebot*** with the same commands we use for unplugged programming: *forward*, *left*, *right*, *backward*.
Extras: Bluebots, robot mice or similar robots on the blackboard. This is *optional* for our course.

All parts where *forward*, *left*, *right* and *backward* commands are used can be executed with a Bluebot and the robot mouse.

NOTE: the size of the board must be adapted to the movement of the mouse or Bluebot's robot (*forward* = one square).

Watch the film https://youtu.be/6jo007uhh1Q.be/6jo007uhh1Q
For an explanation of how the Bluebot works.

The film shows an example with the ***blueblocs*** **app.** https://youtu.be/FaaXX_BUd5Qtu.be/FaaXX_BUd5Q



You can use the ***blueblocs application*** to work with mobile phones.

Film on how to make your own bluebot chessboard:
https://www.youtube.com/watch?v=Up37AdIXVlA?v=Up37AdIXVlA

(* in the pictures one square equals *two times forward* for the robot-mouse)

For inspiration, take a look at these robots on a chessboard:
https://www.youtube.com/shorts/4RCPpIpFoFg

Funded by
the European Union

Important note

In our Erasmus+ project 'Introduction to Artificial Intelligence, Coding and Chess for Primary School', we had the support of the University of Adelaide *(our project stakeholder)* ***participating with its CSER*** *[usually pronounced 'Caesar']* ***Group: Computer Science Educational Research Group.***

CSER has produced several MOOCs aimed at both primary and secondary school teachers on different topics worth exploring and considering for all teachers.

We have inspired many approaches to introduction to AI for primary school teachers and offer links to videos and other online resources for teachers and students suggested or created by the CSER Group. This framework has been developed under a Creative Commons [CC BY-NC] licence.

4.1 Introduction to AI for primary school teachers

In today's rapidly changing world, technology plays a crucial role in our lives and one of the most interesting and impactful technological fields is Artificial Intelligence (AI). As educators, it is very important to understand AI and its potential impact on society and education. This brief introduction aims to provide a basic understanding of AI and its relevance in the primary school setting.

AI refers to the development of computer systems capable of performing tasks that usually require human intelligence. It is about creating algorithms and systems that can learn from data, make decisions and perform tasks autonomously. AI has the potential to transform several aspects of our lives, including education.

In the context of primary education, AI can be a powerful tool for enhancing teaching and learning experiences. It can provide personalised learning opportunities, adaptive feedback and intelligent tutoring systems tailored to each student's needs. AI-powered educational platforms can help identify students' strengths and weaknesses, enabling teachers to design targeted interventions and provide personalised support.

In addition, artificial intelligence can assist teachers with administrative tasks such as correcting assignments, generating lesson plans and managing classroom resources. By automating some routine tasks, artificial intelligence can free up teachers' time, allowing them to focus more on personalised teaching and promoting students' critical thinking skills.

However, it is important to approach AI with a critical mindset and consider its ethical implications. An understanding of bias, transparency and privacy is crucial for the integration of AI in the classroom. As primary school teachers, we have a responsibility to guide our students to develop a balanced understanding of AI and its ethical implications, fostering a sense of responsibility and digital citizenship.

By embracing AI as a tool for innovation and improvement, we can empower our students to navigate and succeed in a technology-driven world. It is essential for us as educators to stay informed about the latest developments in AI and explore ways to thoughtfully integrate it into our teaching practices, always keeping in mind the specific needs and capabilities of our primary learners.

On this journey, we can collaborate, share experiences and learn from each other to harness the full potential of AI in primary education. Together, we can create engaging and inclusive learning environments that prepare our students for the challenges and opportunities of the future.

Artificial intelligence has become an important field and will have a very significant impact on human society. If you have already tried Chat-GTP[2] , you can imagine how classroom activities will change.

The section on AI will give you a brief introduction to what AI is and an overview of how it works.

During the course, some examples on the use of AI, a simple explanation of neural networks and some stimulating classroom activities will be provided.

4.1.1 What is AI?

The definition of AI has been discussed and modified over the years. It was invented in 1950 by Alan Turing and John McCarthy. Turing tried to answer the question *"Can machines think?"* and introduced the Turing Test to determine whether a computer can demonstrate the same intelligence (or results of the same intelligence) as a human being.

The CSER group[3] at the University of Adelaide uses the following definition:

AI is the creation of machines or computer programmes that imitate human capabilities, such as training a machine *to see* (recognise objects in an image) and *hear* (interpret and analyse sounds). In a traditional computer programme, our algorithm tells the machine exactly what to do (step by step), whereas in an AI programme, the machine is programmed to learn and make decisions on its own. Artificial Intelligence is a broad term that covers a range of specialisations and subsets, such as ***computer vision*** and ***natural language processing***.

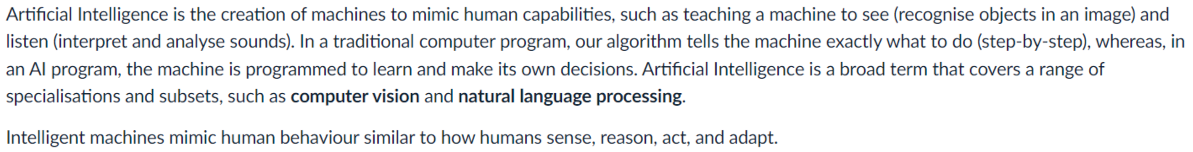
Intelligent machines mimic human behaviour by imitating humans in feeling, reasoning, acting and adapting.

---

[2] Requests can also be made in Italian

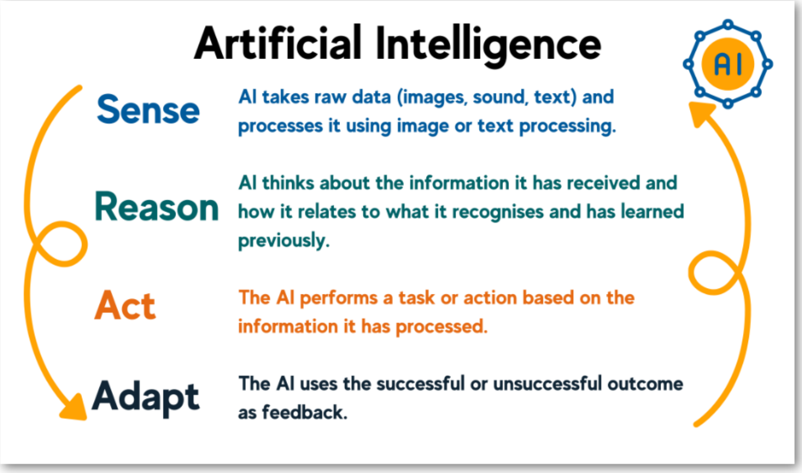
[3] *In our Erasmus+ project 'Introduction to Artificial Intelligence, Coding and Chess for Primary School', we had the support of the University of Adelaide (our project stakeholder) participating with its CSER [usually pronounced 'Caesar'] Group: Computer Science Educational Research Group.*

Funded by
the European Union



Artificial Intelligence is the creation of machines to mimic human capabilities, such as teaching a machine to see (recognise objects in an image) and listen (interpret and analyse sounds). In a traditional computer program, our algorithm tells the machine exactly what to do (step-by-step), whereas, in an AI program, the machine is programmed to learn and make its own decisions. Artificial Intelligence is a broad term that covers a range of specialisations and subsets, such as **computer vision** and **natural language processing**.

Intelligent machines mimic human behaviour similar to how humans sense, reason, act, and adapt.



Artificial Intelligence

Sense AI takes raw data (images, sound, text) and processes it using image or text processing.

Reason AI thinks about the information it has received and how it relates to what it recognises and has learned previously.

Act The AI performs a task or action based on the information it has processed.

Adapt The AI uses the successful or unsuccessful outcome as feedback.

(From the CSER- AI course:

https://CSERmoocs.adelaide.edu.au/available-moocs#teaching-ai-in-the-classroom)

To explore this topic further, many explanations of AI can be found on YouTube:

https://www.youtube.com/results?sp=mAEB&search_query=ai+explained

Perhaps you have already tried ChatGPT or at least heard a lot about it? Maybe some of your students are already using it to write their assignments. You can find ChatGPT at: https://chat.openai.com/

When asking ChatGPT.

"Explaining artificial intelligence to primary school students". The answer will be similar to this:

"AI, or Artificial Intelligence, is a technology that helps computers learn and do things that look smart. It is like teaching a computer to think and make decisions, just like humans do!"

Computers are very 'good' at following instructions and performing tasks quickly, but they usually need humans to tell them *exactly what to do*. With artificial intelligence, computers can learn from data and experiences and make decisions on their own, without always being told what to do.

One way in which AI is used is through ***machine learning***.

Machine learning involves computers learning many examples and using this knowledge to make predictions or solve problems. For example, you may have seen apps that can recognise your face in photos or suggest videos you would like to watch. These apps use artificial intelligence to learn from many images and videos to do these things.

Another way in which AI is used is in ***natural language processing***. This is when computers understand and interact with human language, such as when we talk to virtual assistants like Siri or Alexa. They can understand what we say and respond to our questions or commands.

AI is also used in many other areas, such as ***self-driving cars,*** where computers learn to drive by analysing and reacting to their surroundings. It is also used in the ***healthcare*** sector to help doctors analyse medical images or predict diseases.

But it is important to remember that AI was created by humans and still has certain limitations. It is always supervised by people who make sure it works properly and ethically. So even though AI can do extraordinary things, it is important that we humans understand how it works and use it responsibly.

In conclusion, AI is a technology that helps computers learn and make decisions autonomously. It can perform many tasks that seem intelligent, such as recognising faces, understanding human speech and even driving cars. It is an exciting field that is shaping our future! "

4.1.2 Chronology of AI

To begin with, we provide a historical overview.

History of artificial intelligence: Dates and key names

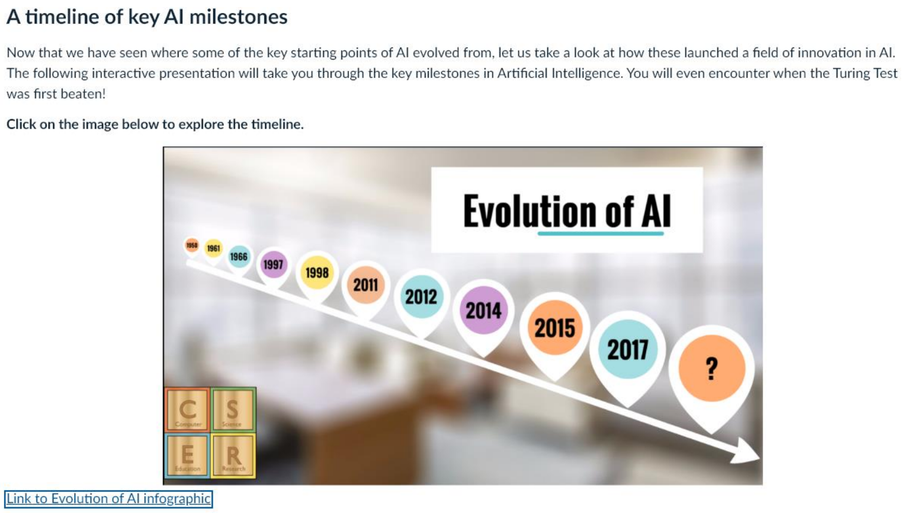
The idea of a *'thinking machine'* dates back to ancient Greece. But since the advent of electronic computing (and in relation to some of the topics discussed in this article) important events and milestones in the evolution of artificial intelligence include the following:

- **1950:** Alan Turing publishes *Computing Machinery and Intelligence.* In the article, Turing, famous for deciphering the Nazis' ENIGMA code during World War II, proposes to answer the question *"Can machines think?"* and introduces the Turing Test to determine whether a computer can demonstrate the same intelligence (or results of the same intelligence) as a human being. Since then, the value of the Turing Test has been debated.
- **1956:** John McCarthy coined the term 'artificial intelligence' at the first conference on artificial intelligence held at Dartmouth College. (Later that year, Allen Newell, J.C. Shaw and Herbert Simon create Logic Theorist, the first functioning artificial intelligence programme).
- **1967:** Frank Rosenblatt builds the Perceptron Mark 1, the first computer based on a neural network that 'learns' by trial and error. Just a year later, Marvin Minsky and Seymour Papert publish a book entitled *Perceptrons*, which becomes both the reference work on neural networks and, at least for a time, an argument *against* future neural network research projects.
- **1980s:** Neural networks using a *back-propagation algorithm* for training become widely used in artificial intelligence applications.
- **1997:** IBM's Deep Blue beats the then world chess champion Garry Kasparov in a chess match (and rematch).
- **2011:** IBM Watson beats champions Ken Jennings and Brad Rutter at *Jeopardy!*, [a TV quiz game show in the US].
- **2015:** Baidu's Minwa supercomputer uses a special type of deep neural network, called a *convolutional neural network*, to identify and classify images with a higher rate of accuracy than the average human.
- **2016**: DeepMind's AlphaGo programme, powered by a deep neural network, beats Lee Sodol, the world champion Go player, in a five-game competition. The victory is significant considering the huge number of possible moves during the game (over 14.5 trillion after only four moves!). Subsequently, Google bought DeepMind for 400 million dollars.
- **2023**: The rise of large language models, or LLMs, such as ChatGPT, creates a huge shift in the performance of AI and its potential for business value creation. With these new generative AI practices, deep learning models can be pre-trained on large amounts of raw, unlabelled data.

From: https://www.ibm.com/topics/artificial-intelligence

Watch the CSER presentation[4] on the evolution of AI as prezi (https://prezi.com/dashboard/next/#/all)

A good overview of the history of AI can be found here: https://prezi.com/view/GmFxO9w6KS8vhEAcaSC6/



A timeline of key AI milestones

Now that we have seen where some of the key starting points of AI evolved from, let us take a look at how these launched a field of innovation in AI. The following interactive presentation will take you through the key milestones in Artificial Intelligence. You will even encounter when the Turing Test was first beaten!

Click on the image below to explore the timeline.

Link to Evolution of AI infographic

4.2 Automatic Learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. It is inspired by the way humans learn from experience and adapt to new information.

Machine learning is based on the training of computer systems capable of automatically analysing and interpreting data, recognising patterns [*patterns*] and making predictions or taking actions on the basis of that analysis. This process begins with feeding the machine learning algorithm with a large data set, which serves as a training set. The algorithm then learns from this data to identify *patterns* and relationships, extracting meaningful insights and building a model.

There are different types of machine learning algorithms, including *supervised learning*, *unsupervised learning* and *reinforcement learning*:

- Supervised learning involves training the model using labelled data, where each example in the training set is matched with the corresponding correct answer. The model learns to generalise from these examples to make accurate predictions or classifications on new, unseen data.

---

[4] *We have inspired many approaches to introducing AI to primary school teachers and offer links to videos and other online resources for teachers and students suggested or created by the CSER Group. This framework has been addressed with a Creative Commons [CC BY-NC] licence proposal.*

- **Unsupervised learning,** on the other hand, deals with unlabelled data, in which the algorithm tries to identify *patterns*, relationships or groupings in the data without knowing the correct results in advance. Unsupervised learning is often used for clustering tasks, dimensionality reduction or anomaly detection.

- **Reinforcement learning** consists of training an agent [programme] to interact with an environment and learn through trial and error. The agent *receives feedback* in the form of rewards or penalties based on its actions, allowing it to learn to make decisions that maximise the cumulative reward over time.

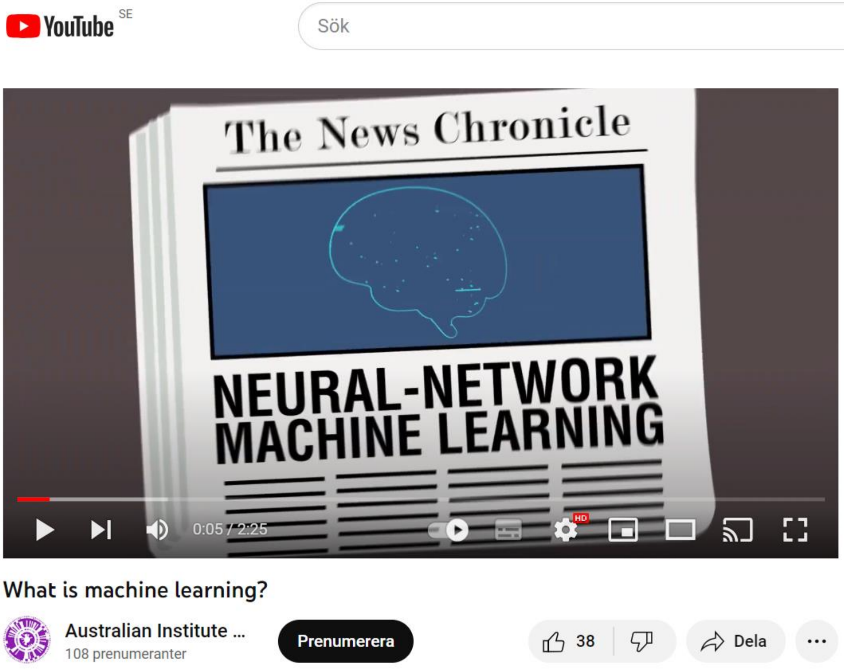
Machine learning has found application in various fields, including healthcare, finance, marketing, robotics and others. It has been used for tasks such as image and speech recognition, natural language processing, *advice* systems, fraud detection and autonomous vehicle control.

As educators, understanding the basics of machine learning can be invaluable as it becomes increasingly integrated into various aspects of our lives. The opportunity to explore how machine learning can be harnessed to enhance teaching and learning experiences while promoting a critical understanding of its limitations and ethical considerations.

4.2.1 What are AI and machine learning?

This film on YouTube:

https://www.youtube.com/watch?v=ob_p9qARjJom/watch?v=ob_p9qARjJogives a brief explanation of AI and machine learning, including some examples (2:20 min)



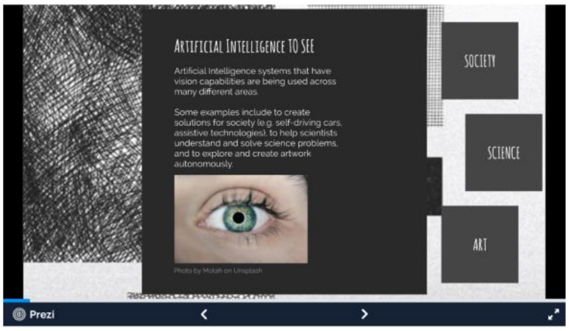
What is machine learning?

Watch CSER's prezi presentation https://prezi.com/p/dxnouruask3k/ai-examples/p/dxnouruask3k/ai-examples/ on AI examples. They cover the whole range of AI: self-driving cars, assistive technology,

Funded by
the European Union

health science, teaching robots to draw, AI for communication: reading and writing, speaking and listening, AI that makes recommendations, AI and games, supporting humans, agriculture.

We have created a Prezi presentation for you to explore that highlights some fascinating examples of AI. In some cases, we have included optional videos to view if you would like to learn more about AI technology. You may work through this resource in as much or as little time as you like. **No sign-in is required to view.**



ARTIFICIAL INTELLIGENCE TO SEE

Artificial Intelligence systems that have vision capabilities are being used across many different areas.

Some examples include to create solutions for society (e.g. self-driving cars, assistive technologies), to help scientists understand and solve science problems, and to explore and create artwork autonomously.

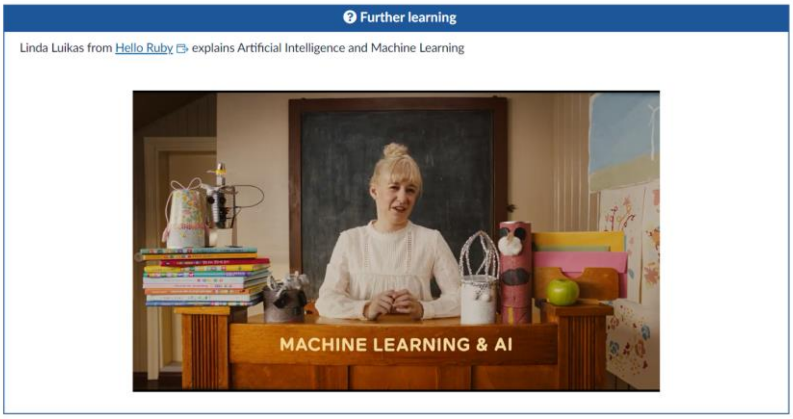
Photo by Motah on Unsplash

Examples of AI Licence: CSER Group, Creative Commons Share-Alike, Non-Commercial.

4.2.2 AI and machine learning for primary school:

We recommend watching Linda Luikas' film (about 7 minutes) on AI and machine learning:

https://www.youtube.com/watch?v=Wm1Id-vEX3U&t=197sch?v=Wm1Id-vEX3U&t=197s



Further learning

Linda Luikas from Hello Ruby explains Artificial Intelligence and Machine Learning

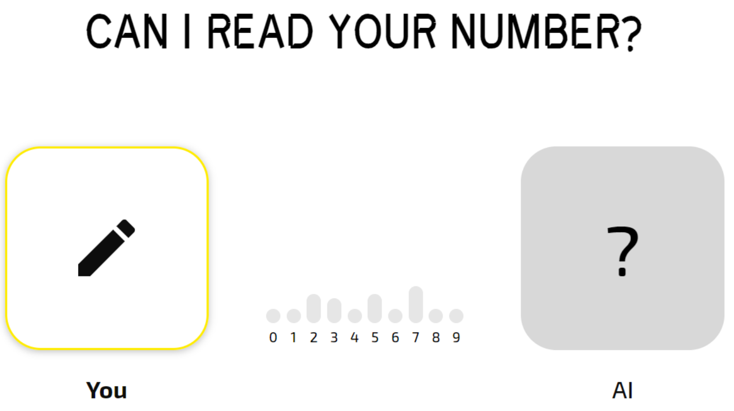
Let's take a look at machine learning with the help of "I AM AI" https://www.i-am.ai/ which is an open-source project of "IMAGINARY open mathematics".

(Link: https://about.imaginary.org/)

You can find much more information on AI on their website.

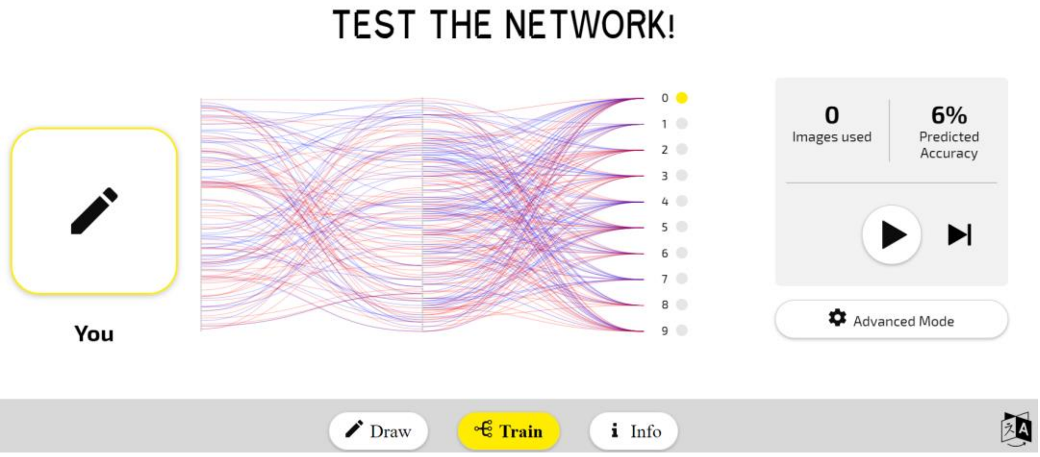
4.2.3 Neural numbers :

Start with https://www.i-am.ai/apps/neural-numbers/#intro



How does the artificial intelligence recognise the number? It is necessary to train the programme with many numbers and provide feedback. How to understand the training process?

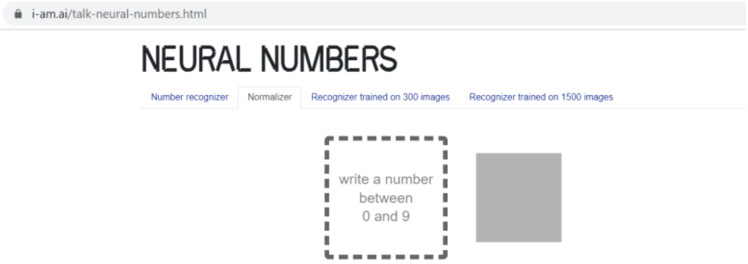
https://www.i-am.ai/apps/neural-numbers/#training



Read more: https://www.i-am.ai/apps/neural-numbers/#how-does-it-work

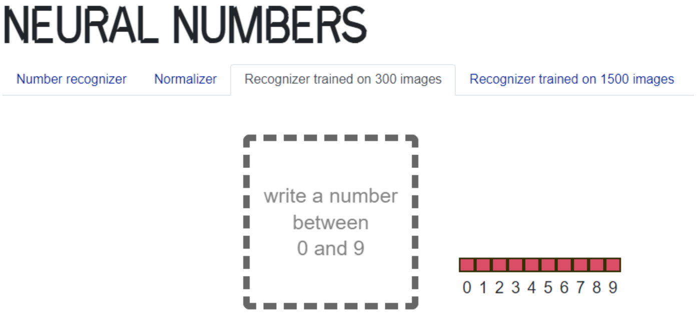
The next activity shows the importance of the training process:

https://www.i-am.ai/talk-neural-numbers.html

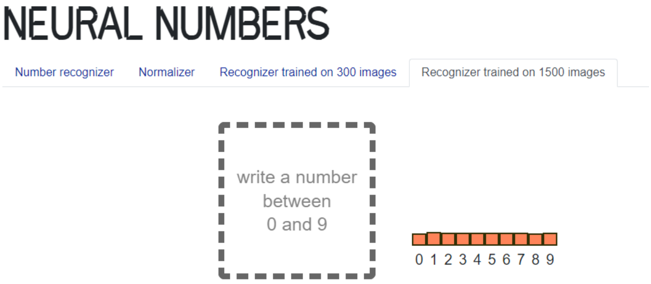

i-am.ai/talk-neural-numbers.html
NEURAL NUMBERS
Number recognizer
Normalizer
Recognizer trained on 300 images
Recognizer trained on 1500 images
write a number between 0 and 9

Start with the normaliser. The number is converted into an image of 25x25 pixels. The artificial intelligence only needs to know whether a pixel is activated (black) or not (grey) and will learn.

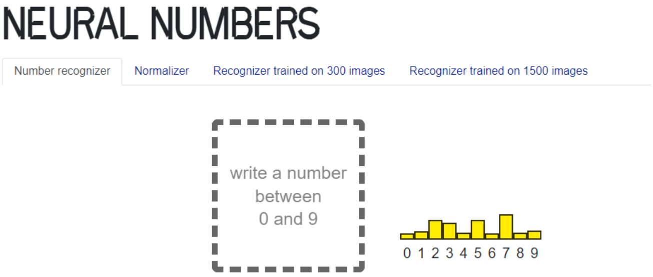
System test trained on 300 images.


NEURAL NUMBERS
Number recognizer
Normalizer
Recognizer trained on 300 images
Recognizer trained on 1500 images
write a number between 0 and 9
0 1 2 3 4 5 6 7 8 9

And out of 1,500 images


NEURAL NUMBERS
Number recognizer
Normalizer
Recognizer trained on 300 images
Recognizer trained on 1500 images
write a number between 0 and 9
0 1 2 3 4 5 6 7 8 9

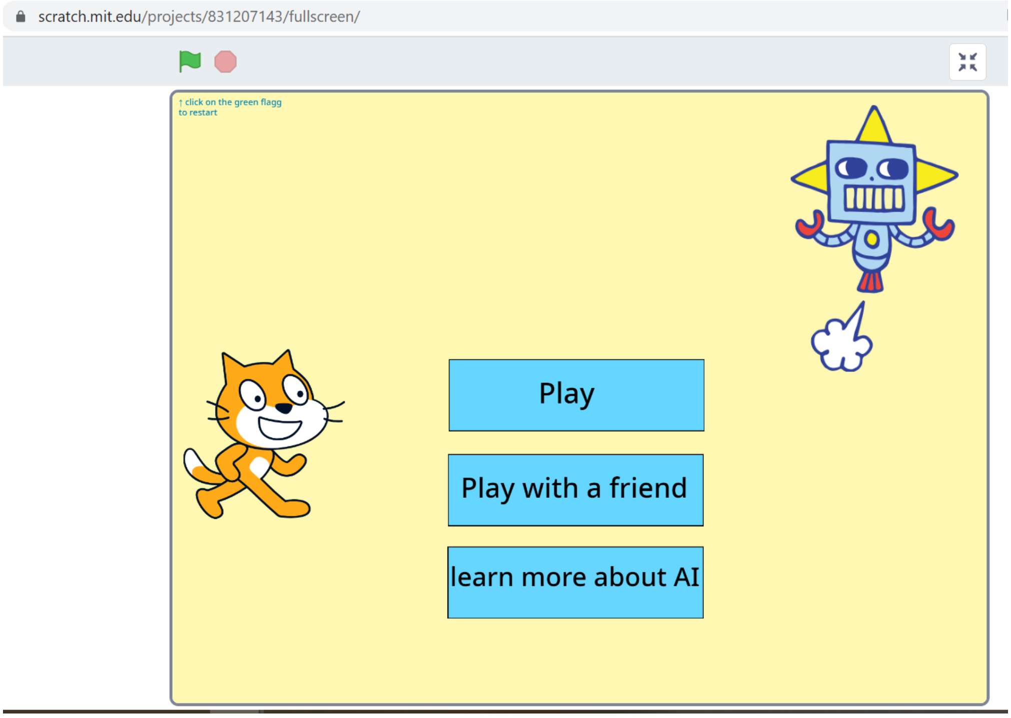
And out of 70,000 images


NEURAL NUMBERS
Number recognizer
Normalizer
Recognizer trained on 300 images
Recognizer trained on 1500 images
write a number between 0 and 9
0 1 2 3 4 5 6 7 8 9

4.2.4 AI - learning process with the Scratch application

To understand the learning process, we made a *Scratch application*:

https://scratch.mit.edu/projects/831207143/fullscreen/


scratch.mit.edu/projects/831207143/fullscreen/
↑ click on the green flagg
to restart
Play
Play with a friend
learn more about AI

Start with 'learn more about AI', then play and also try playing with a friend.

Do it together with a colleague.

If you are familiar with *Scratch*, take a look *inside*. Feel free to make changes.

Of course, there is much more to be said about AI.

In the next unit you will learn how to recognise *patterns*.

4.3 Pattern recognition [*pattern*]

Pattern recognition is a special ability that computers and even humans have to recognise and understand *patterns*. Do you know what *patterns* are? They are things that repeat themselves or follow a certain order. Think for example of the stripes of a zebra or the colours of the rainbow. Those are *patterns*.

Now, imagine if *a computer* could be *taught to recognise patterns*. It could help us in many ways! For example, it could help us sort things faster or make decisions based on what it sees. Just as we can look at a chessboard and know what the best move is by recognising *patterns* on the board, so too can a computer learn to do this.

Funded by
the European Union

In a computer, *pattern* recognition occurs through special algorithms, which are like a series of instructions. These algorithms help the computer observe its surroundings, learn about different *patterns* and make decisions based on what it sees. It is like teaching the computer to see the world in a similar way to how we see it.

For example, let us say we want the computer to *recognise shapes.* We can show it many images of different shapes, such as circles, squares and triangles. The computer will learn to identify the *patterns* that make each shape unique. Then, when we show it a new image with a shape, the computer will be able to tell us what shape it is.

Pattern recognition is a useful skill because it helps us understand and make sense of the world around us. For computers, it is a *kind of superpower* that helps them perform many tasks more efficiently. So, the next time you see a pattern, remember that computers can also learn to recognise and understand it, just like we do!

Pattern recognition plays a crucial role in chess, both for human players and computer programmes. Chess is a game that involves strategic thinking and decision-making based on *patterns* and positions on the board.

In chess, *patterns* are recurring formations or configurations that occur on the board. These *patterns* can include different types of openings, mid-game positions, tactical patterns and endgame scenarios. Recognising these *patterns* is essential to understanding the current state of the game and making informed moves.

For human players, *pattern* recognition in chess requires years of experience and study. Experienced players are able to quickly identify familiar patterns and apply their knowledge to make strategic decisions. They can recognise common opening moves, such as the Sicilian Defence or Ruy Lopez, and understand the typical plans and ideas associated with such openings. Likewise, they can identify tactical *patterns* such as *nailing*, *double-attacking* or *threading*, which can lead to the conquest of material or the checkmate of the opponent.

Computer programmes also rely heavily on *pattern* recognition in chess. They utilise sophisticated algorithms and large databases of games to recognise and analyse *patterns* on the board. By comparing the current position with known *patterns* from previous games, chess engines can suggest strong moves and assess the strengths and weaknesses of the position. This allows them to play at a high level and challenge human grandmasters.

Pattern recognition in chess is not limited to specific positions or moves. It also extends to strategic concepts such as *pawn structures*, *piece co-ordination* and *positional advantages*. Experienced players and chess programmes are able to recognise these *patterns* and use them to develop long-term plans and positional manoeuvres.

An interesting aspect of *pattern* recognition in chess is the ability to visualise the future consequences of moves and to identify potential *patterns* that may emerge. Strong players are able to predict how a certain move will affect the overall position and what *patterns* might emerge as a result.

In summary, *pattern* recognition is a fundamental skill in chess. It allows players to understand position, identify common *patterns* and tactics, and make informed decisions. Both human players and computer programmes rely on *pattern* recognition to analyse the board and play strategically. Developing *pattern* recognition skills is a key aspect of improving in chess and can greatly increase one's playing strength.

Pattern recognition, as the name suggests, involves the process of recognising *patterns* through the use of machine learning algorithms. It involves classifying data based on existing knowledge or statistical information extracted from *patterns* and their representations. *Pattern* recognition has wide application potential and is used in various fields, such as speech recognition, speaker identification, multimedia document recognition and automated medical diagnosis.

In a typical *pattern* recognition application, raw data is processed and transformed into a machine-friendly format. This process involves the classification and grouping of *patterns*. Classification involves assigning an appropriate class label to a *pattern* based on an abstraction generated from a set of training *patterns* or domain knowledge. It is used in supervised learning. On the other hand, clustering generates a partition of the data, facilitating decision-making in unsupervised learning.

Features, which can be *continuous*, *discrete* or *discrete binary variables*, serve as quantifiers of significant characteristics of an object. For example, in the case of a face, the eyes, ears, nose and so on are features of the face. When these features are combined, they form a feature vector. In speech, Mel's frequency cepstral coefficient (MFCC) represents spectral features and the sequence of the first 13 features forms a feature vector.

Pattern recognition has several features, including the ability to recognise familiar *patterns* quickly and accurately, to identify and classify unfamiliar objects, to recognise shapes and objects from different angles, to identify partially hidden patterns and objects, and to recognise *patterns* quickly and effortlessly with automaticity.

Training and learning are essential aspects of *pattern* recognition. Learning allows a system to adapt and function accurately based on the data provided. The data set is typically divided into two categories: *the training set* and the *test set*. The training set is used to build the model by applying training rules and algorithms. It contains part of the data set, usually about 80 per cent, and is used to extract relevant information and obtain results. The test set, which comprises about 20% of the data set, is used to check the accuracy of the system after training. Recall of neural number activity.

***Pattern* recognition has applications in various fields:**

1. **Image processing, segmentation and analysis**: Provides machines with human-like recognition capabilities required for image processing.

2. **Computer vision**: *Pattern* recognition helps to extract meaningful features from images and videos, contributing to applications in fields such as biological and biomedical imaging.

3. **Seismic analysis**: *Pattern* recognition methods help discover, imagine and interpret temporal patterns in seismic array recordings.

Funded by
the European Union

4. **Radar signal classification/analysis**: Signal recognition and signal processing techniques are used for the classification of radar signals, including the detection and identification of AP mines.

5. **Speech recognition**: *Pattern* recognition paradigms have achieved significant success in speech recognition algorithms by treating larger units such as words as *patterns* instead of using phoneme-level descriptions.

6. **Fingerprint identification**: *Pattern* recognition approaches are widely used in fingerprint recognition technology, which is a dominant sector in the biometric market.

In conclusion, pattern recognition plays a crucial role in understanding and classifying data in various fields, enabling machines to recognise and analyse patterns effectively.

The goal of *pattern* recognition is to understand and automate the complex decision-making processes of human beings, which often involve the recognition and interpretation of *patterns*. For example, in a game of chess, the next move is determined by analysing *patterns* on the board, while buying or selling shares is based on identifying *patterns* within financial information. The goal of *pattern* recognition is to unravel these intricate decision-making mechanisms and enable computers to perform these functions automatically.

Pattern recognition is defined as the study of how machines can observe their surroundings, learn to distinguish various *patterns* of interest from the background and make logical decisions about the categories of these *patterns*. Recognition involves assigning certain objects to specific categories during the recognition process. Being an ever-evolving and expanding field, there are several definitions of *pattern* recognition. A first definition characterises it as "classification of input data through the extraction of important features from a large number of noisy data" (Thomas Gonzalez, 1978). Another definition describes *pattern* recognition as a scientific discipline focused on the categorisation of objects into numerous classes, which is an integral part of artificial intelligence decision-making systems (Sergio Theodoridis, 2003). In general, *pattern* recognition can be understood as a process of reducing, mapping or labelling information. In computer science, it refers in particular to the correspondence between incoming data and the information stored in a database according to its attributes.

Pattern recognition examples and projects

- Image recognition
- Video recognition
- Stock market forecasts
- Optical character recognition
- Text *pattern* recognition
- Handwriting recognition
- Face recognition and visual search
- Voice or speaker recognition
- Emotion recognition systems
- Advantages of *pattern* recognition

Further information and links: (could be not necessary to translate in Italian)

An example from the CSER course: https://youtu.be/YZTkoaMkgKE (approx. 5 min.)

Read more: https://viso.ai/deep-learning/pattern-recognition/ (site)

a YouTube clip 'how computer vision is finally taking off, after 50 years'
https://www.youtube.com/watch?v=eQLcDmfmGB0atch?v=eQLcDmfmGB0 (video, 8 min)

From google cloud tech:
https://www.youtube.com/watch?v=OcycT1Jwsnsube.com/watch?v=OcycT1Jwsns (video, approx. 7 minutes)

4.4 Biases

Just like human beings, AI systems can be subject to biases that can influence their decision-making processes. Bias in *AI refers to the presence of unfair or discriminatory biases in the data, algorithms or results produced by AI systems*. These biases may emerge from various sources, including the data used to train the AI system, the algorithms themselves, or human biases inadvertently incorporated into the design of the system.

In the context of schooling and education in general, biases in AI can have profound consequences.

For example, if an AI-powered educational platform is trained on data that disproportionately represents a particular demographic group, it could inadvertently perpetuate existing inequalities. Students from underrepresented groups might receive biased recommendations or limited access to resources, hindering their educational opportunities.

It is essential that educators are aware of the potential biases present in AI systems and their impact on students' experiences. By critically examining and questioning the design, training data and decision-making processes of AI systems, we can help ensure that our students are not adversely affected by biased algorithms.

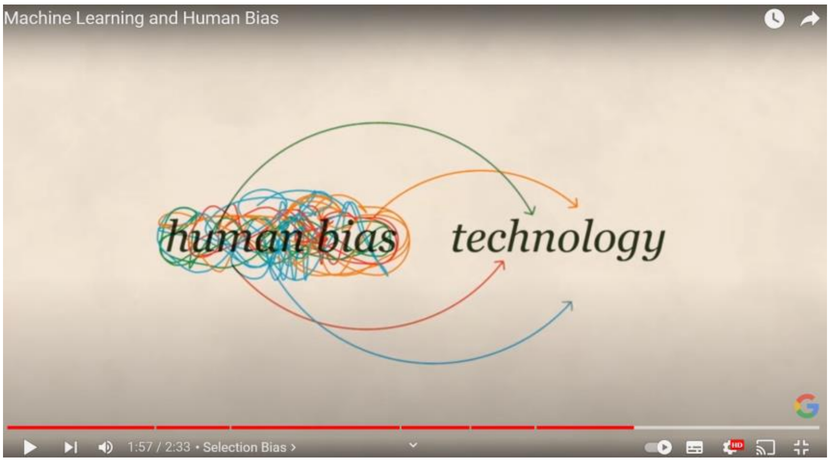
Furthermore, as primary teachers, we have a responsibility to educate our students about AI biases. By promoting a critical mindset, we can empower them to question the decisions made by AI systems. It is crucial to teach our students to recognise that AI is only as impartial as the data and human input it receives and should never be considered infallible.

Addressing biases in AI requires the collaboration of educators, researchers, policymakers and technology developers. By supporting transparency, diversity and ethical practices in AI development, we can work to create more equitable and inclusive AI systems.

Study the following film

https://www.youtube.com/watch?v=59bMh59JQDocom/watch?v=59bMh59JQDo

As an introduction to prejudice. (2:40)

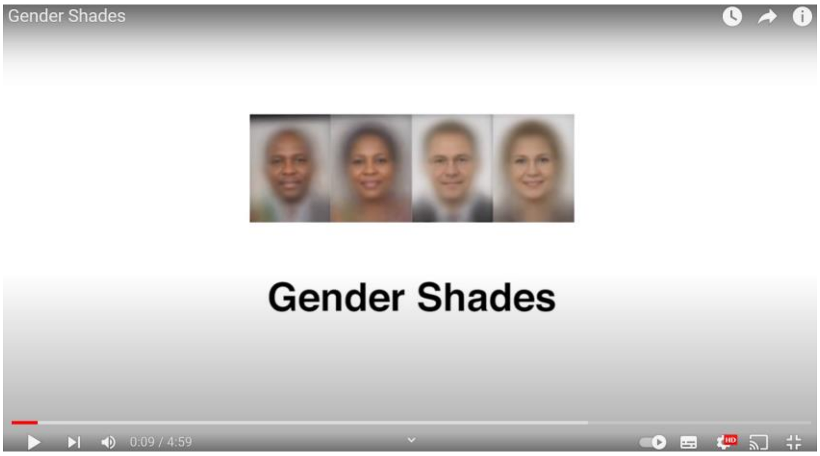

Machine Learning and Human Bias
human bias
technology
1:57 / 2:33 • Selection Bias ›

Suggest other examples on:

- Human prejudices
- Interaction bias
- Latent prejudice
- Selection bias

Watch the following film to learn more about the subject. (5 minutes)

https://www.youtube.com/watch?v=TWWsW1w-BVowatch?v=TWWsW1w-BVo



Read more about prejudice: https://www.youtube.com/watch?v=TihWPgUVCKw

(2:10 min)

Funded by
the European Union

4.5 The future of AI:

Overview of the future of AI

The future of Artificial Intelligence (AI) has immense potential to transform our lives in various areas. AI is constantly advancing, thanks to rapid technological advances, increased computing power and the growing availability of large data sets. Here is a brief overview of potential future developments in AI:

a) **Enhanced automation**: artificial intelligence is poised to further revolutionise automation. As algorithms and robotic systems become more sophisticated, they will take over repetitive and mundane tasks, freeing up human resources for more creative and complex work. This automation could lead to greater efficiency, productivity and economic growth.

b) **Personalised experiences**: AI has the potential to personalise our experiences in various areas, such as education, healthcare and entertainment. By harnessing user data, AI algorithms can tailor content, recommendations and services to individual preferences, resulting in more engaging and personalised experiences.

[Every coin has its reverse: in doing so, AI systems push us into (more or less) homogeneous groups, dividing society into opposing groups that are continuously reinforced by societies, thus reinforcing oppositions. Is this a good thing? Or is it?]

c) **Advances in robotics:** Artificial intelligence and robotics are closely interlinked. Future developments in robotics will allow for more advanced robots capable of performing complex tasks, collaborating with humans and navigating dynamic environments. This paves the way for applications in areas such as healthcare, manufacturing and space exploration.

d) **Autonomous vehicles:** Self-driving cars and autonomous vehicles are an area of great interest and research in the field of AI. The future promises safer and more efficient transport systems, in which AI-powered vehicles can navigate the roads, make real-time decisions and communicate with each other to avoid accidents and congestion.

e) **Innovations in healthcare:** Artificial intelligence has the potential to revolutionise healthcare, contributing to disease diagnosis, drug discovery and personalised treatment plans. Machine learning algorithms can analyse large amounts of medical data, identify patterns and help healthcare professionals make more accurate diagnosis and treatment decisions.

f) **Ethical considerations:** As AI becomes more pervasive, ethical considerations become increasingly important. Issues such as algorithm bias, transparency, privacy and the potential impact on jobs and socio-economic disparities need to be addressed. Ongoing research and discussions are essential to ensure responsible development and deployment of AI technologies.

g) **AI and creativity:** Although traditionally associated with logic and reasoning, AI is making great strides in creative fields such as art, music and writing. Generative models and creative AI systems are being developed to create original works and collaborate with human artists, opening up new possibilities for creative expression.

[We also reflect on the 2023 all-out strike by screenwriters and other workers against Hollywood production companies].

h) **Human-machine collaboration**: The future of AI is not to replace humans, but rather to enhance their capabilities. Collaborative systems that combine human skills with AI capabilities are likely to become increasingly common, allowing humans and machines to work together synergistically and achieve more than they could individually.

Watch the film about humanoid robots:
https://www.youtube.com/watch?v=PLBAbmETEmooutube.com/watch?v=PLBAbmETEmo
concerning the robot Grace (2021) developed during the pandemic.
About Nadine (communication robot)
About AI-da (2019) robotics art generator,
Jia Jia, communication
Friend, communication
Sofia , 2016, (10' 4")

Here are two more videos on humanoid robots:

a) https://www.youtube.com/watch?v=DbXXdE_GqRY (6' 32")
more about the future, examples with humanoid robots, this is s cience fiction!
b) Interview with the sexy and realistic robot named Sophia CNBC (5' 4")

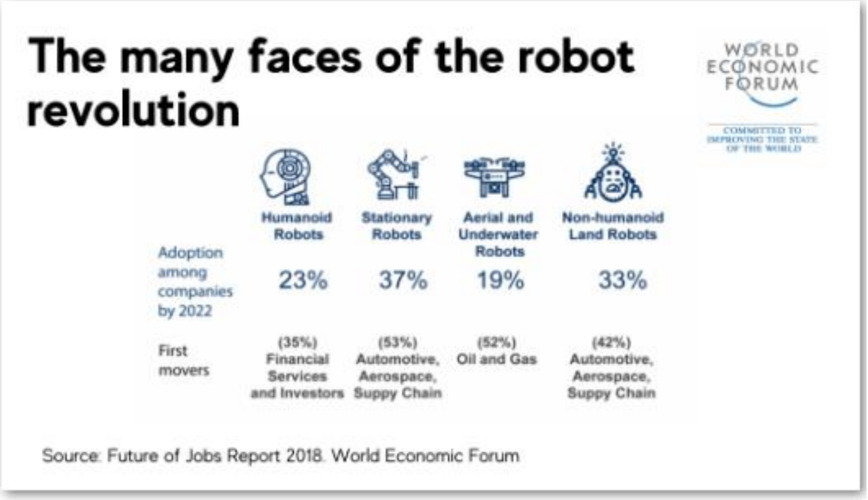
...and more:

- A fun, *unrelated* activity you can try with colleagues or students at **CS Unplugged** in New Zealand is based on the Turing test and discusses the concepts behind it. https://classic.csunplugged.org/activities/the-turing-test/

- A test to determine whether a poem was created by a computer or a human being: can you tell the difference? https://www.npr.org/sections/alltechconsidered/2016/06/27/480639265/human-or-machine-can-you-tell-who-wrote-these-poems

Many wonder whether robots or artificial intelligence are replacing our jobs. Although they may replace some of our most repetitive tasks, they are also helping us to do our work faster and better and are creating new jobs and opportunities.

A World Economic Forum report on the future of work found that although machines will replace some human jobs, they will also create 133 million new roles and free up time for human activities to complete other tasks. In 2018, in 12 industries, 29% of tasks were completed by machines, and by 2022 it is estimated that this percentage will be 42%.
Below is an example of the use of some of these machines.



The many faces of the robot revolution

	Humanoid Robots	Stationary Robots	Aerial and Underwater Robots	Non-humanoid Land Robots
Adoption among companies by 2022	23%	37%	19%	33%
First movers	(35%) Financial Services and Investors	(53%) Automotive, Aerospace, Suppy Chain	(52%) Oil and Gas	(42%) Automotive, Aerospace, Suppy Chain

Source: Future of Jobs Report 2018. World Economic Forum

benefits and risks

Watch the CSER film:

Link: https://www.youtube.com/watch?v=9qGao-3Hrc8&t=12s (6' 34")

Dangerous places - challenging places - repetitive tasks - low errors - assistance to humans - invasion of privacy - social manipulation - autonomous machines used for evil -

A possible lesson on the future of AI proposed by ChatGPT

Title: ***Exploring the future of artificial intelligence: Unlocking the potential***

Duration: 80 minutes

Purpose: The purpose of this lesson is to introduce primary school students to the exciting possibilities and potential impact of Artificial Intelligence (AI) in the future. The lesson will encourage curiosity, critical thinking and ethical considerations about AI technologies.

Materials:

1. *Interactive whiteboard or projector*
2. *Internet access.*
3. *Worksheets and pens/pencils*
4. *Optional: Visual aids or videos illustrating AI technologies and advances.*

<u>Educational plan</u>

Introduction (10 minutes)

1. *Start the lesson by engaging students in a discussion about technology and its evolution over time.*
2. *Ask students if they have heard of AI or artificial intelligence. Encourage them to share their ideas and prior knowledge.*
3. *Explain that AI refers to computer systems capable of performing tasks that typically require human intelligence, such as problem solving, decision making and learning.*

Exploring applications of artificial intelligence (20 minutes)

1. *Introduce students to a range of applications of AI and how they are currently used in various fields, such as healthcare, transport, education and entertainment.*
2. *Show visual aids or videos to highlight real examples of AI technologies, such as autonomous vehicles, virtual assistants or robotic companions.*
3. *Encouraging students to think about how these technologies can improve our lives and address challenges in different areas.*

Imagining the future (30 minutes)

1. *Divide the students into small groups and provide them with worksheets and pens/pencils.*
2. *Ask each group to brainstorm and imagine potential future applications of AI. They can consider various sectors such as education, transport, healthcare and entertainment.*

3. Encouraging students to think creatively and imagine how AI could shape our lives in the future, both from a practical and imaginative perspective.

4. Each group should select an AI application that they find most interesting or impactful and create a short presentation outlining its features and potential benefits.

Group presentation and discussion (15 minutes)

1. Each group has the opportunity to present their chosen IA application to the class.

2. After each presentation, encourage a discussion by asking questions such as:

- *How could this AI application improve people's lives?*
- *What challenges or concerns might arise from its use?*
- *Are there ethical considerations to be taken into account when using this technology?*

Reflection and ethical considerations (15 minutes)

1. Engage students in a class discussion to reflect on lessons learned about the future of AI.

2. Discuss the potential benefits and risks associated with the increasing use of AI technologies.

3. Encourage critical thinking and ethical considerations by asking questions such as:

- *How can we ensure that AI technologies are used responsibly and ethically?*
- *What safeguards can be put in place to address potential risks or prejudices?*
- *What role can individuals play in shaping the future of AI for the better?*

Conclusion and closure (5 minutes)

1. Summarise the key points discussed during the lesson.

2. Encourage students to continue exploring and remaining curious about AI and its future developments.

3. Thank the students for their participation and commitment to the lesson.

***Note**: adapt the lesson plan according to the age of the students, their prior knowledge and the dynamics of the class. Incorporate interactive activities, videos or demonstrations to enhance student engagement and understanding. It is important to strike a balance between highlighting the exciting potential of AI and addressing any concerns or ethical considerations about its use.*

5.1 Overview of learning scenarios

The aim of our project is to develop 13 lessons of 90 minutes each for fifth grade students (approximately 11 years old) that will provide students with the basic knowledge of:

- What a scientific approach means
- How humans, computers and artificial intelligence work
- the basics of programming
- uses of AI today and in the future
- the risks and opportunities of AI

and to develop chess skills, and thus the other skills that chess trains, and through this training provide basic insights into how human learning occurs.

We developed a total of 13 lessons combining chess, coding and artificial intelligence.

In each lesson (5 to 13) there is a riddle to solve, which extends to all lessons. If you do not follow all the lessons, adapt to the riddle.

In the first four lessons, we focus on learning the basics of chess. For those who already have chess within one or more subjects (chess in schools), it is possible to go further and start with programming lessons. We recommend taking at least one chess repetition lesson to refresh the basics.

In lessons 5 and 6 we learn the basics of coding. (SARA) with *unplugged programming*. In lessons 7 and 8 we focus on AI, in lessons 9-12 we discuss chess and AI and in lesson 13 we discuss Conclusions: the future of AI, benefits, risks and biases.

An overview:

1. Chess 1
2. Chess 2
3. Chess 3
4. Chess 4
5. Programming 1
6. Programming 2
7. AI 1
8. AI 2
9. Chess and AI 1
10. Chess and AI 2
11. Chess and AI 3
12. Human prejudices
13. Conclusions on AI: Future, Benefits and Risks

Assessment sessions (according to the teacher's autonomous choices)

Prepare these lessons by doing each activity on your own. Adapt to the learning situation in your class. Make a list of materials needed and prepare worksheets by copying the right number of photocopies. Some activities require Internet access. Enquire in advance with the IT staff at your school. In some activities, students will need to be able to use computers, I-pads or mobile phones. Be prepared for this as well.

Funded by
the European Union

5.2 Chess Lessons (#1 to #4)

Chess Lesson 1

'Chess and AI - 1 Pre-chess'

A. chess lesson 1: possible learning objectives:

STEM (Science, Technology, Engineering, Mathematics) Lesson Learning Objectives

1. Understanding the history and cultural significance of chess:

- Identify the origins of chess and its historical importance as a game invented more than 1,400 years ago in India to train generals in war strategies.

- Recognise the impact of chess on various cultures and its reputation as a game of intelligence and skill, devoid of luck and chance.

2. Mathematical concepts: Coordinate system:

- Learn the chess coordinate system, which involves associating letters and numbers with each square on the chessboard.

- Practice using the coordinate system to locate and name specific squares on the blackboard accurately.

3. Problem solving and critical thinking skills:

- Analysing and understanding the movement *patterns* of the different chess pieces (king, rook, bishop, queen, knight and pawn) and their unique abilities.

- Apply strategic thinking to make informed decisions during the minigame 'Crossing the board' and during chess play in general.

- Use critical thinking skills to identify the best moves and tactics during the game.

4. Spatial awareness and visualisation:

- Develop spatial awareness by understanding the layout of the board, including lines, ranks and diagonals.

- Improve visualisation skills by visualising potential moves and strategies before executing them on the board.

Funded by
the European Union

5. Concentration and attention:

- Cultivate a sense of calm and concentration during chess lessons to foster better learning and a better playing experience.

- Show respect for the other players, maintaining concentration and refraining from disruptive behaviour during the lesson.

6. Sportsmanship and ethics:

- Stress the importance of treating opponents with respect and courtesy during the game, as chess is a game for gentlemen and gentlewomen.

- Familiarise yourself with the '6 Golden Rules' of chess, which define appropriate behaviour and conduct during the game.

7. Integration of chess and artificial intelligence (AI):

- Briefly introduce the concept of chess and AI, emphasising how computers have influenced playing strategies and tactics.

8. Teamwork and collaboration:

- Encourage a sense of teamwork and cooperation among students, who see themselves as teammates rather than opponents during chess lessons.

9. Connecting chess to STEM subjects:

- Discuss the connections between chess and mathematics, highlighting how the game involves strategic thinking, problem solving and mathematical concepts such as coordinates and *patterns*.

10. Encourage lifelong learning:

- Promoting interest in chess as a lifelong learning activity, inspiring students to continue exploring the complexities and strategies of the game beyond the class they attend.

By incorporating these STEM learning objectives into the lesson, students can develop critical thinking skills, improve their mathematical understanding, increase spatial awareness and acquire valuable life skills such as concentration, sportsmanship and ethical behaviour. In addition, relating chess to STEM subjects can stimulate curiosity and interest in learning more about these fields.

Funded by
the European Union

Possible evaluation grids for Lesson 1 - Pre-chess'.

Assessment grids are useful tools for evaluating students' performance and progress in achieving specific learning objectives. Assessment grids related to STEM subjects for chess lessons in primary school are proposed here:

Chess Lesson 1: (decide with Jesper about what to say in the next for lessons: Italian or Swedish path? Both)

1. Demonstrating an understanding of chess pieces and movement

(teaching at the demonstration chessboard):

- Advanced: The student carefully explains the movement and unique characteristics of each chess piece (king, queen, rook, bishop, knight, pawn) on the demonstration board and can answer detailed questions about their movements.

- Competent: The student demonstrates a good understanding of most chess pieces and their movements on the demonstration board, but may have some minor inaccuracies in the explanations.

- Development: The student struggles to explain the movements of some chess pieces on the demonstration board and may require further clarification.

- Beginning: The student shows limited understanding of the movements of the pieces on the demonstration board and requires significant guidance.

2. Application of chess rules during the mini-game "Crossing the board" (game):

- Exemplary: The student consistently follows the '6 Golden Rules' while playing the 'Cross the Board' minigame, demonstrating excellent sportsmanship and respect for opponents.

- Competent: The student mostly respects the '6 golden rules' during the mini-game, but may occasionally require a gentle reminder of correct behaviour.

- Partially competent: The student struggles to consistently follow the '6 golden rules', causing some interruptions during the minigame.

- Limited: The student often ignores or forgets the '6 golden rules', causing interruptions and misbehaviour during the mini-game.

3. Understanding the chessboard coordinate system

(teaching on the demonstration chessboard):

- Outstanding: The student demonstrates a thorough understanding of the chessboard co-ordinate system, consistently identifying squares and answering questions correctly.

- Satisfactory: The student shows a good command of the chessboard coordinate system, but may occasionally make small errors in identifying squares.

- Development: The student struggles to understand the coordinate system of the chessboard and often makes mistakes in identifying the squares.

- Limited: The student has difficulty understanding the chessboard coordinate system and requires significant assistance.

4. Performance in the 'Crossing the board with kings' minigame

- Skilled: The student successfully employs strategic thinking and planning during the minigame 'Crossing the board with kings', showing effective use of pawns and kings to win the game.

- Adequate: The student demonstrates some strategic thinking during the mini-game, but may overlook opportunities to effectively advance pawns or kings.

- Need for improvement: The student shows limited strategic thinking and may have difficulty progressing in the minigame 'Crossing the board with kings'.

- Ineffective: The student lacks strategic thinking and does not make significant progress in the minigame 'Crossing the board with kings'.

5. Participation and summary of the lesson Conclusion:

- Active participation: The student actively participates in the concluding discussion, providing thoughtful answers and questions related to the content of the lecture.

- Active participation but limited engagement: The student participates in the concluding discussion but makes only minimal contributions.

- Passive participation: The student is present during the conclusions, but does not actively participate or contribute to the discussion.

- Disengaged: The student shows disinterest or distraction during the visit and does not actively participate in the discussion.

These evaluation grids can be used by the teacher to provide feedback to students, assess their progress and identify areas for improvement. It is essential to ensure that the grids are in line with the learning objectives of the lesson and are clearly communicated to students in advance to promote a sense of transparency and fairness in the assessment process.

Funded by
the European Union

Chess Lesson 2

‘Chess and AI - 2 Pre-chess'

Possible learning objectives

Learning objectives for the primary school chess lesson:

1. Understanding chess pieces: Students will demonstrate a clear understanding of the movement patterns and unique characteristics of each chess piece (pawn, rook, bishop, knight, queen) and be able to identify their positions on the demonstration board.

2. Application of chess strategies: Students will apply strategic thinking by playing several chess mini-games (Rook vs. 5 pawns, Bishop vs. 3 pawns, Queen vs. 8 pawns and "The Duel in the Club") and demonstrate their ability to effectively utilise the strengths of each chess piece.

3. Knowledge of the chessboard co-ordinate system: Students will be able to identify and name the squares on the chessboard using the co-ordinate system (letters and numbers) and apply it to communicate effectively during play.

Evaluation grids for lesson 2 - Lesson 'Chess and AI - 2 Pre-chess'

1. Understanding chess pieces:

- Advanced: The student accurately explains the movement and unique characteristics of each chess piece and can accurately demonstrate movements on the board.

- Competent: The student demonstrates a good understanding of most chess pieces and their movements on the demonstration board, but may have some minor inaccuracies in the explanations.

- Development: The student struggles to explain the movements of some chess pieces on the demonstration board and may require further clarification.

- Beginning: The student shows limited understanding of the movements of the pieces on the demonstration board and requires significant guidance.

2. Application of chess strategies:

- Skilled: The student shows strategic thinking and effectively utilises the strengths of different chess pieces during mini-games, making wise moves to achieve game objectives.

- Competent: The student demonstrates some strategic thinking during the mini-games, but may overlook opportunities to optimally use some chess pieces.

Funded by
the European Union

- Partially competent: The student struggles to apply strategic thinking and may not use chess pieces effectively, resulting in limited progress in mini-games.

- Limited: The student lacks strategic thinking and does not make significant progress in mini-games.

3. Knowledge of the chessboard coordinate system:

- Examples: The student accurately identifies and names the squares on the board using the coordinate system on the demonstration board and answers the related questions with confidence.

- Satisfactory: The student shows a good command of the chessboard coordinate system, but may occasionally make small errors in identifying squares or answering related questions.

- Development: The student struggles to understand the coordinate system of the chessboard and often makes mistakes when identifying squares or answering questions about them.

- Limited: The student has difficulty understanding the chessboard coordinate system and requires significant assistance.

These evaluation grids will help the teacher assess students' performance and progress in achieving STEM-related learning objectives during chess lessons. By providing clear and specific feedback, students can better understand their strengths and areas for improvement in chess skills and knowledge.

Introduction to artificial intelligence, coding and chess – T2T course

Funded by
the European Union

Chess Lesson 3:

Chess and AI - 3 Pre-chess'

Learning objectives for the lesson 'Special rules and checkmate' in primary school:

1. Understanding *the special rules*: Students will demonstrate an understanding of the special rules of chess, including pawn promotion, en passant and castling. They will be able to apply these rules during play and recognise their importance in improving strategic options.

2. *Identifying checkmate*: Students will be able to identify and explain checkmate in chess. They will understand the concept of checkmate as a condition of victory and will be able to recognise and create images of checkmate during play.

3. *Strategic thinking and problem solving*: Students will begin to develop strategic thinking skills by engaging in exercises and games that require them to analyse positions, plan moves and make decisions to achieve checkmate or victory in *'losing chess'* (mini-game).

Evaluation grids for Lesson 3:
Special rules and checkmate

1. Understanding the special rules:

- Advanced: The student demonstrates a clear understanding of the special rules of chess (pawn promotion, en passant and castling) and applies them effectively during play. He/she is able to explain the strategic advantages of using these rules.

- Competent: The student shows a good understanding of most of the special rules of chess and is able to apply them in most situations. May present some minor inaccuracies in explanations or play.

- Development: The student struggles to grasp some special chess rules and may not apply them consistently during play. He needs further support and practice in the use of these rules.

- Beginning: The student does not have a basic understanding of the special rules of chess and needs significant guidance to apply them correctly.

2. Identify the checkmate:

- Competent: The student demonstrates a thorough understanding of checkmate and can accurately identify checkmate positions during drills and play. Is able to effectively explain the concept of checkmate to others.

Funded by the European Union

- Competent: The student shows a good understanding of checkmate and can identify most checkmate positions, but may occasionally overlook some situations.

- Partially competent: The student struggles to identify checkmate positions consistently and may require additional guidance to fully understand the concept.

- Limited: The student has difficulty identifying checkmate positions and requires significant assistance to recognise and understand checkmate.

3. Strategic thinking and problem solving:

- Strategic Master: The student consistently demonstrates strong strategic thinking skills, effectively planning moves to checkmate and making sound decisions in loser's chess. Demonstrates creativity in creating pairing pictures.

- Competent strategist: The student shows good strategic thinking skills, but may occasionally overlook certain possibilities or miss opportunities in exercises and games.

- Strategy development: The student struggles to apply strategic thinking consistently and may need support in planning moves and understanding the best approaches in different scenarios.

- Limited strategy: The student has difficulty thinking strategically and solving problems and needs significant assistance to make progress in exercises and games.

These evaluation grids will help the teacher assess students' progress in achieving STEM-related learning objectives in the chess lesson 'Special Rules and Checkmate'. By providing specific feedback, students can better understand their strengths and areas for improvement in chess knowledge and strategic thinking.

Chess Lesson 4

Chess and AI - 4 Pre-chess'

Learning objectives for the lesson 'How to make a checkmate' in primary school:

1. ***Understanding checkmate patterns***: Students will learn and recognise common checkmate *patterns*, including kissing checkmate and barber's checkmate. They will be able to identify the key pieces involved and understand the tactics used to checkmate.

2. ***Application of checkmate strategies***: Students will practise checkmate using different pieces and combinations, such as two rooks against a king. They will demonstrate the ability to plan and execute checkmate strategies effectively.

3. Defence against *checkmate*: Students will learn how to defend against the most common threats of checkmate, in particular the so-called *'barber's check'.* They will demonstrate the ability to perform strategic moves to protect their king and avoid checkmate.

4. Chess position analysis: Students will develop skills in analysing chess positions, assessing threats and making informed moves to checkmate or avoid being checkmated.

Evaluation grids for Lesson 4 'How to make a checkmate'.

1. Understanding checkmate patterns:

- Advanced: The student demonstrates a thorough understanding of different checkmate *patterns* (checkmate, barber's check) and can accurately identify the key pieces and tactics involved. Is able to clearly explain concepts to others.

- Competent: The student shows a good understanding of most checkmate *patterns* and can identify them correctly in most situations. May present some minor inaccuracies in explanation or play.

- Development: The student struggles to grasp some checkmate *patterns* and may not identify them correctly. He needs further support and practice to recognise these *patterns*.

- Beginning: The student does not have a basic understanding of checkmate *patterns* and needs meaningful guidance to recognise and understand them.

2. Apply checkmate strategies:

- Competent: The student consistently demonstrates strong skill in the application of checkmate strategies, such as the effective use of two rooks to achieve checkmate. Is able to plan and execute checkmate moves with confidence.

- Competent: The student shows good ability to apply checkmate strategies, but may occasionally overlook some possibilities or make small errors in the execution of moves.

- Development: The student struggles to apply checkmate strategies consistently and may require support in planning and executing checkmate moves.

- Limited: The student has difficulty applying checkmate strategies and needs significant assistance to execute effective moves.

3. Defence against checkmate:

- Strategic defender: The student consistently demonstrates effective defensive skills against checkmate threats, particularly Scholar's mate. He is able to execute strategic moves to protect his king and avoid checkmate.

- Competent defender: The student shows good defensive skills, but may occasionally miss some defensive moves or overlook threats.

- Developing defender: The pupil struggles to defend against threats of checkmate consistently and may require support in identifying and executing defensive moves.

- Limited defender: The student has difficulty defending himself against threats of checkmate and needs significant assistance to protect his king.

4. Analysis of chess positions:

- Strategic Analyst: The student demonstrates a strong ability to analyse chess positions, assess threats and make informed moves to checkmate or avoid being driven out.

- Competent analyst: The student shows good analytical skills, but may occasionally miss some possibilities or make small errors of judgement.

- Developing Analyst: The student struggles to analyse chess positions consistently and may require support in assessing threats and making strategic moves.

- Limited analyst: The student has difficulty effectively analysing chess positions and needs significant assistance in understanding threats and planning moves.

These evaluation grids will help the teacher assess students' progress in achieving the STEM-related learning objectives in the chess lesson 'How to Checkmate'. By providing specific feedback, students will better understand their strengths and areas for improvement in chess knowledge, strategy and analysis.

Funded by
the European Union

5.3 Programming Lessons (#5 and #6)

Chess and AI - 5 *Coding*

"SARA - Sequence"

First *coding* lesson

The lesson contains 4 different units (plus the first part of the riddle):

1) History
2) Programming disconnected, sequence
3) Programming on a chessboard
4) Worksheets
5) THE ENIGMA - part 1

STEM-related learning objectives for primary school
For Lesson 1 on programming:

1. **Understanding the basics**: Students will learn the basics of programming and coding, focusing on the concept of sequences in programming. They will understand the importance of following a specific order of commands and how this affects the result.

2. **Historical context**: Students will gain an awareness of the history of computers and programming. They will learn about milestones and key figures, fostering an understanding of the evolution of technology.

3. **Problem solving**: Students will engage in *problem solving* activities by creating and following sequences of commands. They will learn to think *logically* and *strategically* to complete specific tasks, such as moving from one position to another in the classroom or on a whiteboard.

4. **Collaboration and communication**: Students will work in small groups and pairs to develop and share their sequences. They will practise effective communication and teamwork, explaining their ideas to their peers and discussing possible solutions to the challenges posed.

5. **Mathematical logic**: Students will develop their logical thinking and mathematical skills while planning and executing sequences of commands. They will understand the importance of precision and accuracy in programming.

6. **Concentration and memory training**: Students will improve their ability to concentrate and remember sequences of commands by actively participating in programming activities. They will exercise their memory and attention to detail.

Funded by the European Union

7. **Foresight and planning**: Students learn the importance of considering the consequences of their actions before executing them. They will develop forecasting and planning skills by thinking ahead and anticipating the results of their sequences.

8. **Strategic thinking**: Students will begin to use strategic thinking, analysing different approaches and determining the most effective sequences to complete specific tasks. They will consider alternative solutions and make informed decisions.

9. **Application of concepts to chess**: students will apply their understanding of sequences to the game of chess. They will learn to use specific commands to move chess pieces on a chessboard and develop logical thinking skills through chess-related programming exercises.

10. **Programming Principles**: Students will learn about the SARA principles (Sequence, Alternative, Repetition and Abstraction) in programming. In this lesson, they will focus on the sequence aspect and understand its importance in the creation of programmes.

11. Reflective thinking: Students reflect on their programming experiences and discuss what they have learnt from the activities. They will write down their observations, conclusions and key points from the lesson.

Note: these learning objectives can be modified according to the specific level, students' prior knowledge and the time available in the classroom. Ensure that activities are engaging and age-appropriate so that students can actively participate and explore programming and sequencing concepts.

Evaluation grids for the first programming lesson:
"SARA - Sequence"

1. Understanding the history of computers and programming:

- The student demonstrates knowledge of milestones in the history of computing and programming by accurately marking them on the timeline.

- The student effectively communicates the importance of these milestones and their impact on technological advances.

2. Sequences and programming [without computer]:

- The student actively participates in movement activities, following and giving commands to move from one place to another.

- The student collaborates with peers to plan and execute sequences of commands, demonstrating an understanding of the concept of sequence in programming.

- The student accurately writes down the sequence of commands used to guide a companion from one place to another.

- The student reflects on the assignments and provides a clear statement on his or her learning from the activities.

3. Chessboard movement and programming:

- The student demonstrates understanding of the co-ordinates of the chessboard and the movement of the rook by accurately executing the given commands to move the rook on the chessboard.

- The student collaborates with a partner to create a sequence of commands to move the tower from one specific box to another.

- The student writes precisely on the blackboard the sequence of commands used to move the tower.

- The student accurately determines the final square where the tower ends based on a sequence of commands.

- The student successfully solves the challenge of moving the tower to a specific box by writing down the correct sequence of commands.

4. Problem-Solving and critical thinking:

- The student effectively applies problem solving skills to determine the sequence of commands needed to move the tower to a specific box.

- The learner demonstrates logical thinking and reasoning in determining the correct sequence of commands for various scenarios.

- The student analyses and evaluates his own sequences and those of his peers to identify errors or improvements.

- The student effectively communicates his solutions and reasoning on the challenges posed by the lesson.

5. Collaboration and communication:

- The student actively engages in group activities, communicating and collaborating effectively with peers to complete tasks and solve problems.

- The student demonstrates active listening skills and respectfully considers the ideas and suggestions of others during group discussions.

- The student effectively communicates his or her ideas, solutions and reflections, using appropriate language and terminology.

6. Technology literacy:

- The student demonstrates the ability to use command cards, chess boards and other materials or tools used during the lesson.

- The student effectively applies technological skills, such as the use of computer presentations or digital resources, to enhance learning and understanding.

7. Self-reflection and evaluation:

- The student reflects on his or her own learning experiences and evaluates his or her understanding of the concepts relating to the sequence, planning and movement of the chessboard.

- The student identifies connections between lecture content and real-world applications or personal experiences.

- The student effectively communicates their reflections and self-assessments, providing insights into their learning process and areas of growth.

Note: the assessment grids may be adapted or modified according to specific learning objectives, educational level and classroom context.

Funded by
the European Union

Chess and AI - 6 *Coding*

'*Unplugged* Programming - Alternative and Repetition'.

Lesson objectives: A and R in SARA.

Students learn to use repetition and alternatives to execute codes more efficiently. They learn to work together and discuss possible solutions for a given task.

STEM-related learning objectives for primary school. for the lesson "Programming 'unplugged' - Alternation and repetition".

1. **Understanding repetition**: Students will learn to use the "repeat" command in programming to execute a specific set of commands several times. They will understand the concept of repetition and its significance in creating efficient and concise code.

2. **Application of repetition**: Students will practise using the 'repeat' command to simplify and optimise code. They will apply the concept of repetition to solve problems, such as moving a chess piece on a board or navigating through a given path.

3. **Problem-solving**: Students will engage in problem-solving activities that require the use of repetition. They will analyse situations, identify repetitive *patterns* and develop strategies to optimise their code.

4. **Decisions based on '*alternatives*'**: Students will learn the concept of *alternatives* in programming and how to make decisions based on certain conditions. They will understand how to use conditional statements (if-else) to choose between different options in their code.

5. **Implementation of alternatives**: Students will practise using conditional statements to create alternative paths in their code. They will apply the concept of alternatives to solve problems, such as avoiding obstacles or making decisions based on specific conditions.

6. **Collaboration and communication**: Students will work in pairs or small groups to discuss and solve programming challenges. They will engage in discussions, explain their code to others and evaluate different solutions.

7. **Strategic thinking**: Students will develop strategic thinking skills while planning and executing sequences of commands. They will consider alternative approaches, make informed decisions and analyse the results of their code.

8. **Logical** Reasoning: Students will improve their logical reasoning skills by identifying *patterns*, predicting outcomes and analysing the logic behind their code. They will develop critical thinking skills when *debugging* and improving their programmes.

9. **Mathematical logic**: Students will strengthen their understanding of mathematical logic by applying concepts such as *repetition* and *alternatives* in programming. They will explore the relationship between mathematics and programming.

10. Reflection **and self-assessment**: Students will reflect on their programming experiences, evaluate their code and identify areas for improvement. They will develop a growth mindset and the ability to learn from their mistakes.

11. **Familiarity with chess terminology**: Students will become familiar with chess terminology, including the movements of the pieces and the chessboard concept. They will apply their understanding of chess movements to create programmes and solve programming challenges.

Note: these learning objectives can be modified according to the specific level, students' prior knowledge and available classroom time. Ensure that activities are engaging and age-appropriate to allow students to actively participate and explore the concepts of *repetition* and *alternatives* in programming.

Evaluation grids for Lesson 6:

Unplugged' Programming - Alternative and Repetition

Category 1: Understanding and applying repetition

1. **Demonstrate understanding of repetition**:

- The student gives examples of situations where repetition is useful in programming.
- The student explains the purpose and meaning of using the 'repeat' command to execute a specific set of commands several times.
- The student identifies repetitive *patterns* and tasks in the code examples provided.

2. **Application of repetition**:

- The student accurately uses the 'repeat' command to simplify and optimise the code in different scenarios.
- The student demonstrates the ability to determine the appropriate number of repetitions needed according to the given problem.
- The student explains his or her thought process and reasons for choosing the number of repetitions.

Funded by
the European Union

Category 2: Understanding and applying alternatives

1. Demonstrate understanding of alternatives:

- The student defines the concept of alternatives in programming and gives examples of situations where alternative paths or decisions are required.

- The student explains the purpose and functionality of conditional statements (if-else) in programming.

- The student recognises the importance of considering different conditions and making decisions based on them.

2. Application of alternatives:

- The student effectively uses conditional statements (if-else) to create alternative paths in their code.

- The student identifies and correctly implements the *appropriate conditional statement* according to the given conditions or requirements.

- The student demonstrates the ability to modify the code according to different conditions and results.

Category 3: Collaboration and Problem Solving

1. Collaboration:

- The student actively participates in discussions and collaborates with peers during pair or group activities.

- The student listens to and respects the ideas and perspectives of others.

- The student effectively communicates his or her thoughts, ideas and solutions to peers.

2. Troubleshooting:

- The student approaches programming challenges with a problem-solving mindset.

- The student analyses problems, identifies *patterns* and develops strategies to optimise the code.

- The student demonstrates resilience and persistence in finding solutions and debugging their code.

Category 4: Reflection and growth mindset

1. Reflection:

- Students reflect on their learning experiences in programming and articulate their understanding of repetition and alternatives.
- The student identifies areas for improvement and growth in their programming skills.
- The student is open to feedback and willing to learn from mistakes.

2. Growth mindset:

- The student welcomes challenges and setbacks as opportunities for growth.
- The student shows a positive attitude towards learning and actively seeks to improve his or her programming skills.
- The student demonstrates resilience and a willingness to take risks in his or her planning efforts.

Note: These evaluation grids may be adapted or modified according to specific learning objectives, instructional level and classroom context. Consider incorporating additional criteria specific to lesson objectives and activities.

Funded by the European Union

5.4 Lessons IA (#7 - #12)

Lesson IA 1:

Students will learn: What AI is, know the history of AI, what AI can be used for, understand the basic concept of machine learning. They will learn how to work together and discuss possible solutions for a given task.

STEM-related learning objectives for primary school for Lesson 1 on AI:

1. **Understanding AI**: Students will learn the definition of Artificial Intelligence (AI) and understand how AI mimics human capabilities and behaviour.

2. **Exploring examples**: Students will explore various examples of AI applications and technologies. They will identify and discuss real-world examples of AI use, such as self-driving cars, assistive technology, healthcare, educational robots, recommender systems and more.

3. **Chronology of AI**: Students will learn the historical chronology of AI, understanding its development over time. They will explore the most important milestones and discoveries in AI research and technology.

4. **Application fields of AI**: Students will explore different fields and sectors in which AI is applied. They will discover how AI is used in various sectors, including healthcare, education, transport, gaming, agriculture and others.

5. **Introduction to Machine Learning**: Students will acquire a basic knowledge of machine learning, a subset of AI. They will learn that machine learning algorithms enable computers to learn from data and make predictions or decisions without being explicitly programmed.

6. **Collaboration and discussion**: Students will work in pairs or groups to discuss AI concepts, share examples and complete worksheets. They will engage in meaningful discussions, exchange ideas and learn from the perspectives of others.

7. **Critical thinking**: Students will develop critical thinking skills by analysing the capabilities and potential of AI. They will evaluate the benefits and challenges associated with AI technologies and consider the ethical implications of their use.

8. **Technology literacy**: students will improve their technological literacy by using computers, iPads or mobile phones to explore interactive AI applications and tools. They will gain practical experience with neural number systems, understanding how AI processes and normalises data.

9. **Communication skills**: Students will practise effective communication skills by presenting their ideas, discussing AI concepts and explaining their understanding of AI to their peers. They will develop clarity in expressing their thoughts and ideas about AI.

10. **Ethical considerations**: Students will be introduced to ethical considerations related to AI, such as privacy, bias and fairness. They will begin to understand the importance of responsible development and use of AI.

11. Reflection **and self-assessment**: Students will reflect on their learning experiences and assess their understanding of AI concepts. They will identify areas for improvement and consider how AI can have a positive impact on their lives and society.

Note: these learning objectives can be modified according to the specific level, students' prior knowledge and available class time. Ensure that activities are engaging and age-appropriate, allowing students to actively participate and explore AI and machine learning concepts.

Evaluation grids

Evaluation grids for the lesson Part 1: Introduction to AI

Category 1: Understanding AI

1. **Knowledge of AI**:

- The student accurately defines IA based on the definition provided by CSER.
- The student demonstrates an understanding of AI as the creation of machines that mimic human capabilities.
- The student identifies and explains the differences between their own definition of IA and that provided by CSER.

2. **AI examples**:

- The student provides relevant examples of AI based on his prior knowledge.
- The student contributes effectively to the class discussion by sharing examples and ideas.
- The student demonstrates an understanding of how AI is present in various fields and applications.

Category 2: Chronology and AI fields

1. **Understand the chronology of AI**:

- The student accurately interprets and discusses the AI chronology presented in the CSER Prezi presentation.

- The student identifies milestones and developments in the field of AI.

- The student demonstrates knowledge of the historical progression of AI.

2. **Knowledge of AI fields**:

- The student watches the videos and presentations provided on AI fields.

- The student actively engages with the content and demonstrates an understanding of the different fields and applications of AI.

- The student completes the worksheet on AI fields, demonstrating an understanding of AI applications.

Category 3: Neural numbers and reflection

1. **Understanding neural numbers:**

- The student demonstrates understanding of the concept of neural numbers as presented by the teacher.

- The student explains the process of normalising neural numbers and its importance in AI.

- The student actively participates in the class activity and engages in writing and interpreting numbers for the demonstration of neural numbers.

2. Reflection on IA:

- The student engages in the individual or paired activity of testing neural numbers on their own devices.

- The student reflects on his or her own experience and shares what he or she has learnt about neural numbers and their relationship to AI.

- The student demonstrates an understanding of the practical application of neural numbers and their significance in AI.

Note: These evaluation grids may be adapted or modified according to specific learning objectives, instructional level and classroom context. Consider incorporating additional criteria specific to lesson objectives and activities.

Funded by
the European Union

Lesson IA 2

Lesson IA 2

Students will learn more about what AI is, what AI can be used for, understand the basic concept of machine learning and in particular the training process. They will learn how to work together and discuss possible solutions for a given task.

STEM-related learning objectives for primary schools for Lesson 2 on AI

1. **Understanding AI**: Students will further develop their understanding of artificial intelligence (AI) and its capabilities. They will explore how AI learns and how it can be used in various contexts.

2. **Exploring Scratch**: Students will try their hand at Scratch, a visual programming language, to learn more about AI. They will work in pairs or small groups to explore AI-related projects and gain hands-on experience with AI concepts.

3. Reflection **on Scratch** activities: Students will reflect on their experiences with Scratch activities focused on learning and play. They will articulate what they have learnt and consider how AI can be used in educational games and interactive experiences.

4. Labelling **and features**: Students will understand the concept of labelling in AI, particularly in the context of image recognition. They will learn to identify features that distinguish different objects and help to label them correctly.

5. **Introduction to feature extraction**: Students will be introduced to the concept of *feature extraction*, which involves the conversion of data (such as images) into quantitative or qualitative features that can be used to distinguish and compare different objects. They will understand the importance of features in the training of artificial intelligence models.

6. **Human image recognition**: Students will explore how humans recognise images and make connections with the image recognition capabilities of artificial intelligence. They will engage in a group activity in which one student draws and the others guess the object drawn, simulating the human image recognition process.

7. **Understanding neural numbers**: Students will reflect on the previous activity on 'neural numbers' and understand how computers process and interpret images using neural networks. They will gain an insight into the computational aspect of image analysis and pattern recognition.

8. **Computer vision**: Students will watch a video showing *computer vision* in action, focusing on image analysis and object tracking. They will observe how computers use edge detection and image segmentation to understand and interpret visual information.

9. **Collaboration and discussion**: Students will work in pairs or small groups during activities and discussions. They will share their observations, explanations and insights on AI and its applications, fostering a collaborative learning environment.

10. **Critical thinking**: Students will develop critical thinking skills by analysing how AI processes and interprets images. They will evaluate the strengths and limitations of AI in image recognition and identify potential areas for improvement.

11. **Technology literacy**: students will continue to improve their technological literacy by using computers, iPads or mobile phones to interact with Scratch and explore AI concepts. They will develop comfort and familiarity with digital tools and platforms.

12. Reflection **and self-assessment**: Students will reflect on their learning experiences and assess their understanding of AI concepts such as labelling, feature extraction and image recognition. They will identify connections between AI and their everyday life.

Note: adjust learning objectives according to the specific level, students' prior knowledge and available class time. Ensure that activities are engaging, hands-on and age-appropriate so that students can actively participate and deepen their understanding of AI.

Funded by
the European Union

Evaluation grids for IA Lesson 2 in primary school:

1. Understanding AI:

- Level 1: demonstrates a basic understanding of what AI is and its general capabilities.

- Level 2: Shows a satisfactory understanding of AI, including its learning process and its potential applications in various contexts.

- Level 3: demonstrates a thorough understanding of AI, explaining its learning mechanisms and demonstrating knowledge of different applications of AI.

2. Exploration of Scratch:

- Level 1: participates in Scratch activities with guidance, demonstrating basic engagement and exploration of AI concepts.

- Level 2: participates independently or collaboratively in Scratch activities, actively exploring AI-related projects and demonstrating an understanding of key AI concepts.

- Level 3: demonstrates advanced engagement with Scratch, creating and modifying AI projects effectively and demonstrating a deep understanding of AI concepts.

3. Reflection on Scratch activities:

- Level 1: provides basic reflections on Scratch activities, briefly discussing one's own experiences and mentioning some key insights.

- Level 2: Reflects on Scratch activities, articulating their learning experiences, identifying specific AI concepts encountered and considering potential educational applications of AI.

- Level 3: offers comprehensive reflections on Scratch activities, demonstrating a deep understanding of AI concepts, providing detailed insights into their learning experiences and proposing creative ideas for integrating AI into educational games and interactive experiences.

4. Labelling and characteristics:

- Level 1: Recognises the concept of labelling and identifies basic characteristics of objects in images.

- Level 2: Demonstrates a satisfactory understanding of labelling and identifies relevant features that contribute to the accurate recognition of objects.

- Level 3: demonstrates an advanced understanding of labelling, accurately identifies distinguishing features and explains their importance in the training of artificial intelligence models.

Funded by the European Union

5. Introduction to feature extraction:

- Level 1: understands the concept of feature extraction and its role in training artificial intelligence models at a basic level.

- Level 2: describes feature extraction and its importance in the training of artificial intelligence models, giving examples of quantitative or qualitative features in image data.

- Level 3: Demonstrates a comprehensive understanding of feature extraction, explains its importance for distinguishing and comparing objects and provides detailed examples of features used in image recognition.

6. Human image recognition:

- Level 1: Participates in the group activity of drawing and riddling, demonstrating a basic understanding of how humans recognise images.

- Level 2: Actively participates in the group activity, making connections between human image recognition and the image recognition capabilities of artificial intelligence and offering relevant insights.

- Level 3: Leads and facilitates group activities effectively, demonstrating a deep understanding of human image recognition and providing meaningful comparisons with image recognition by artificial intelligence.

7. Understanding neural numbers:

- Level 1: Recall the 'neural numbers' activity and demonstrate a basic understanding of how computers process images using neural networks.

- Level 2: describes the process of image processing using neural networks, citing specific examples from the 'neural numbers' activity and the video provided.

- Level 3: Analyses and explains the computational aspects of image analysis and pattern recognition, making connections between neural networks and artificial intelligence capabilities.

8. Artificial vision:

- Level 1: Watch the video on computer vision, observing and identifying the components and basic concepts presented.

- Level 2: demonstrates a satisfactory understanding of computer vision, explaining how edge detection and image segmentation contribute to computer interpretation of visual information.

- Level 3: offers an in-depth look at computer vision, providing a comprehensive explanation of edge detection, image segmentation and computer detection and interpretation of objects.

9. Collaboration and discussion:

- Level 1: participates in collaborative activities and discussions, sharing observations and explanations related to artificial intelligence concepts.

- Level 2: actively participates in collaborative activities, contributing to discussions by sharing observations, explanations and insights and demonstrating effective teamwork.

- Level 3: Collaborates and discusses AI concepts with a high level of competence, fostering a collaborative learning environment and making meaningful contributions to group discussions.

10. Critical thinking:

- Level 1: demonstrates basic critical thinking skills by analysing how AI processes and interprets images, identifying the strengths and limitations of image recognition.

- Level 2: demonstrates satisfactory critical thinking skills, assessing the strengths and limitations of AI in image recognition and suggesting potential areas for improvement or enhancement.

- Level 3: demonstrates advanced critical thinking skills, critically evaluating AI image recognition capabilities, proposing innovative ideas for improving AI algorithms and anticipating future developments in AI technology.

11. Technological literacy:

- Level 1: uses digital tools, such as computers, iPads or mobile phones, to interact with Scratch and explore AI concepts with guidance.

- Level 2: demonstrates satisfactory technological literacy, independently and confidently using digital tools to interact with Scratch and explore AI concepts.

- Level 3: demonstrates advanced technological literacy, skilfully using digital tools, showing fluency in the use of Scratch and effectively applying AI concepts.

12. Reflection and self-assessment:

- Level 1: reflects on learning experiences, assessing one's understanding of AI concepts, such as labelling, feature extraction and image recognition, at a basic level.

- Level 2: Reflects thoughtfully on learning experiences, assessing understanding of AI concepts and making meaningful connections between AI and everyday life situations.

- Level 3: offers comprehensive reflections on learning experiences, providing a detailed self-assessment of one's understanding of AI concepts and demonstrating a deep awareness of the impact and implications of AI in society.

Funded by
the European Union

Note: Adapt the grids according to school level, specific learning objectives and criteria that align with lesson content and expected outcomes. Provide clear indicators and descriptors for each level to ensure consistent and fair assessment.

Funded by
the European Union

6. Teachers’ feedback

6.1 Discussion groups in your school

Collegial learning is a positive way of learning from each other. Go to class from time to time and listen to your colleagues' lesson, give positive feedback and discuss possible opportunities for improvement.

6.2 Feedback to experts

During the implementation, some experts will visit you during some lessons.

If you want to ask or give feedback, you can contact the experts both before and after each lesson.

We would be grateful if, when you fill in the *log-book* on each lesson, you would specify:

- What went well?
- Was there enough time?
- Did the students enjoy the activities?
- What improvements would you suggest?

An interview with the pedagogical expert on the lessons will be conducted by the end of the implementation.

6.3 Feedback on the teachers' course:

- Was the 14-hour self-study time sufficient?
- Were you prepared for the lessons after the teachers' course?
- Feel free to suggest improvements. and features in the training of artificial intelligence models.