

ARTIE

*Challenges and opportunities of the new era:
A Handbook for Students*



Co-funded by the
Erasmus+ Programme
of the European Union

ARTIE: Artificial Intelligence in Education - challenges and opportunities of the new era: development of a new curriculum, guide for educators and online course for students
Project co-funded by European Union under Erasmus+ Programme, 2020-1-HR01-KA201-077800

ARTIE

Challenges and opportunities of the new era: A Handbook for Teachers

Authors

Ivana Ružić
Jura Cmrečak

Publisher

I. osnovna škola Čakovec, Croatia
“Artificial Intelligence in Education - challenges and opportunities of the new era: development of curriculum, guides for teachers and e-courses for students”, project number 2020-1-HR01-KA201-077800 under Erasmus+ Programme.

Consultants/ Reviewers

Katarzyna Garbacik
Andrzej Garbacik
Bogusław Klimczuk
Željko Krnjajić
Janko Radigović
Ana Pina
Christina Eirini Karvouna

Graphic Design & Illustrations

Christina Eirini Karvouna

Language Translators

Jura Cmrečak (Croatian)
Bogusława Denys (English)
Ana Pina (Portuguese)
Christina Eirini Karvouna (Dutch)





8

3

What is Artificial Intelligence ?

How do we use artificial intelligence in everyday life?

11

Let's play AI Bingo!

15

How can computers learn and solve problems?

19

How do algorithms shape our world?

23

How do machine learning systems recognize what they see ?

27

Can machine learning recognize poses?

29

What about the ethical side of the data that we use to train models ?

31

Let's create an AI project with Scratch!

36

Let's get fun with AI!

38

Let's recycle together !

40

Let's chat!

43

Smart Home

45

Let's meet Codey Rocky!

51

Codey Rocky meets sequences and loops

55

Codey Rocky drives safely and responsibly

60

Codey Rocky meets functions

65

Rock, Paper, Scissors!





What is artificial intelligence?

Can a computer or other smart devices think or solve problems?
What is artificial intelligence?
Do we use AI?
How?
Where?

Today we will learn what AI is and how we use an AI in everyday life.



Co-funded by the
Erasmus+ Programme
of the European Union



<http://erasmus-artie.eu>



Topics for discussion

What is AI? Which is the definition of AI?

Since when does AI exist?

Can devices be smart/intelligent?

Can devices replace humans?

What is an ideal intelligent machine?

Advantages and disadvantages of AI

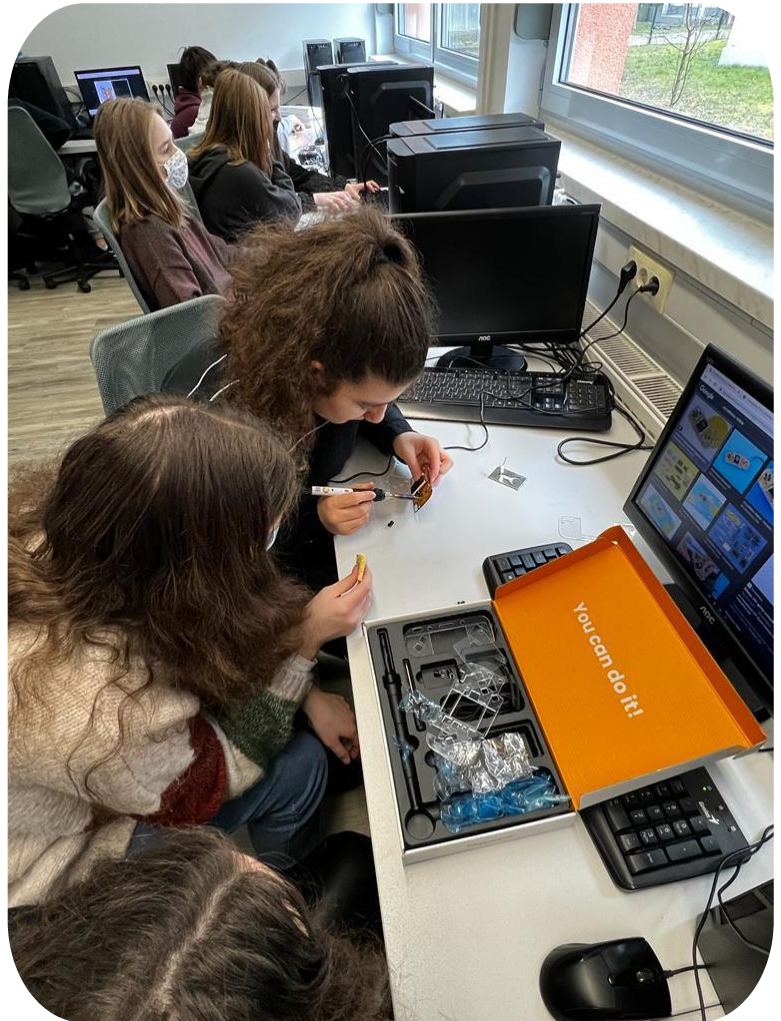
f

Artificial intelligence (AI)

Artificial intelligence (AI) is the ability of a computer program or a machine to think and learn. It is also a field of study that tries to make computers "smart".

John McCarthy came up with the name "artificial intelligence" in 1955.

In general use, the term "artificial intelligence" denotes a machine that mimics human cognition. At least some of the things we associate with other minds, such as learning and problem solving can be done by computers, though not in the same way as we do.





An ideal (perfect) intelligent machine is a flexible agent which perceives its environment and takes actions to maximize its chance of success at some goal. As machines become increasingly capable, mental facilities once thought to require intelligence are removed from the definition. For example, optical character recognition is no longer perceived as an exemplar of "artificial intelligence": it is just a routine technology.

At present we use the term AI for successfully understanding human speech, competing at ***a high level in strategic game systems (such as chess), self-driving cars, and interpreting complex data.*** Some people also consider AI a danger to humanity if it progresses undauntedly.

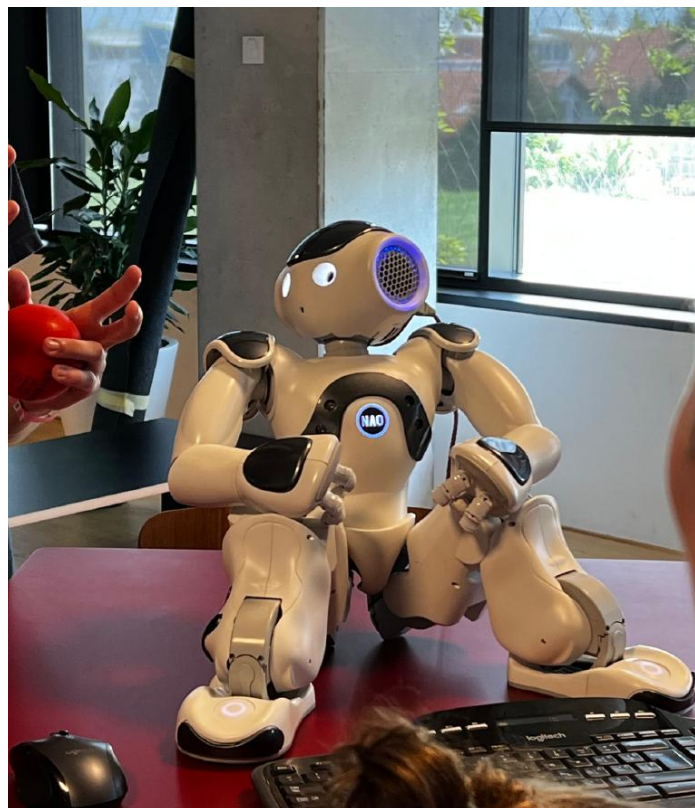
An extreme goal of AI research is to create computer programs that can learn, solve problems, and think logically. In practice; however, most applications have picked on problems that computers can do well. Searching databases and doing calculations are things computers do better than people. On the other hand, "perceiving its environment" in any real sense is way beyond present-day computing.

AI involves many different fields like computer science, mathematics, linguistics, psychology, neuroscience, and philosophy. Eventually, researchers hope to create a "general artificial intelligence" that can solve many problems instead of focusing on just one. Researchers are also trying to create creative and emotional AI which can theoretically empathize or create art. Many approaches and tools have been tried.

History of AI

The idea of a 'machine that thinks' dates back to ancient Greece. But since the advent of electronic computing (and relative 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 paper, Turing—famous for breaking the Nazi's ENIGMA code during WWII—proposes to answer the question 'can machines think?' and introduces the Turing Test to determine if a computer can demonstrate the same intelligence (or the results of the same intelligence) as a human. The value of the Turing test has been debated ever since.





1956: John McCarthy coined the term 'artificial intelligence' at the first-ever AI conference at Dartmouth College. (McCarthy would go on to invent the Lisp language.) Later that year, Allen Newell, J.C. Shaw, and Herbert Simon created the Logic Theorist, the first-ever running AI software program.

1967: Frank Rosenblatt builds the Mark 1 Perceptron, the first computer-based on a neural network that 'learned' through trial and error. Just a year later, Marvin Minsky and Seymour Papert published a book titled Perceptrons, which became both the landmark work on neural networks and, at least for a while, an argument against future neural network research projects.

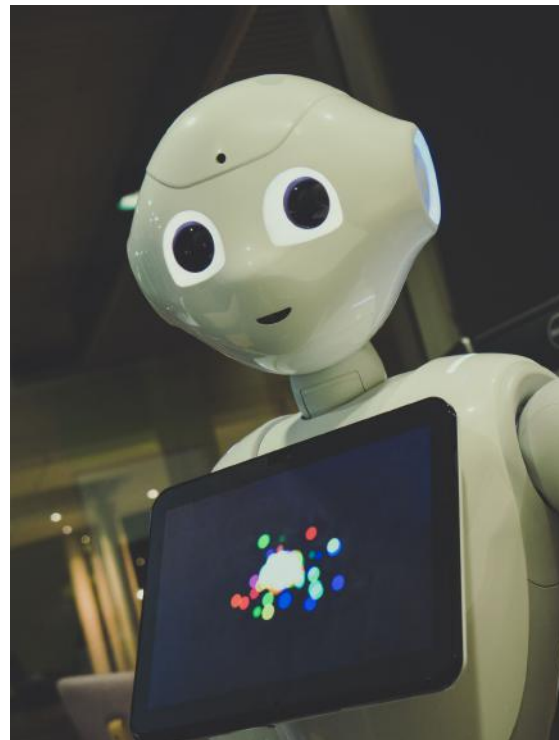
The 1980s: Neural networks which use a backpropagation algorithm to train itself become widely used in AI applications.

1997: IBM's Deep Blue beats the world chess champion, Garry Kasparov, in a chess match (and rematch).

2011: IBM Watson beats champions Ken Jennings and Brad Rutter at Jeopardy!

2015: Baidu's Minwa supercomputer uses a special kind of deep neural network called a convolutional neural network to identify and categorize images with a higher rate of accuracy than the average human.

2016: DeepMind's AlphaGo program, powered by a deep neural network, beats Lee Sedol, the world champion Go player, in a five-game match. The victory is significant given the huge number of possible moves as the game progresses (over 14.5 trillion after just four moves!). Later, Google purchases DeepMind for a reported \$400 million





Graphic works:

Let's recognize and highlight the advantages and disadvantages of AI.
 Imagine and create your own example of AI that helps them in performing daily activities (in performing tasks at school, at home, for hobbies or in their free time).
 What are the characteristics of their fictional product/service?
 How does it make tasks easier?
 Why is it special and how will it win over future users?

Organize a class competition:
<https://studio.code.org/s/oceans/lessons/1/levels/1>
https://petalica-paint.pixiv.dev/index_en.html

Think!



http://erasmus-artie.eu



What do I know now?



What did I learn?



What do I want to know?

Conclusions

AI is computer systems on the basis of rationality and thinking vs. acting:

Human approach:

Systems that think like humans

Systems that act like humans

Ideal approach:

Systems that think rationally

Systems that act rationally



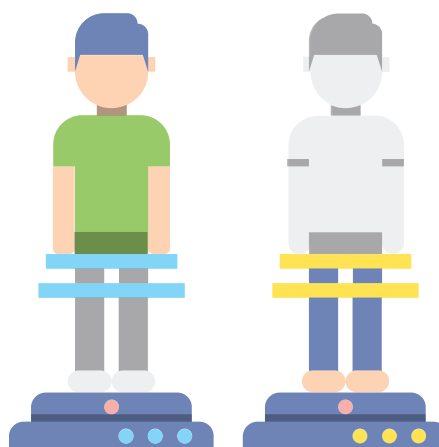


How do we use artificial intelligence in everyday life?

Let's get to know artificial intelligence and how artificial intelligence is used in everyday life.

Can a computer or other smart devices think or solve problems?
Do we use AI?
How do we use AI? Where?
Can we use AI for learning?

Today we will learn what AI is and how we use it in everyday life.



Co-funded by the
Erasmus+ Programme
of the European Union



<http://erasmus-artie.eu>



Topics for discussion:

Do we use AI? How?

Examples for discussion:

Open your phone with face ID, Google search, Social media, Send an email or message,

Digital voice assistants,

Netflix, Smart home devices,

Online shopping recommendations, etc.

Can AI help to do everyday tasks? How?

Can AI be abused? How?



Applications of AI

There are numerous, real-world applications of AI systems today. Below are some of the most common examples.



Speech recognition: It is also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, and it is a capability that uses natural language processing (NLP) to process human speech into a written format. Many mobile devices incorporate speech recognition into their systems to conduct voice search—e.g. Siri—or provide more accessibility around texting.

Customer service: Online chatbots are replacing human agents along the customer journey. They answer frequently asked questions (FAQs) around topics, like shipping, or provide personalized advice, cross-selling products, or suggesting sizes for users, changing the way we think about customer engagement across websites and social media platforms. Examples include messaging bots on e-commerce sites with virtual agents, messaging apps, such as Slack and Facebook Messenger, and tasks usually done by virtual assistants and voice assistants.

Computer vision: This AI technology enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action. This ability to provide recommendations distinguishes it from image recognition tasks. Powered by convolutional neural networks, computer vision has applications within photo tagging in social media, radiology imaging in healthcare, and self-driving cars within the automotive industry.





Recommendation engines: Using past consumption behavior data, AI algorithms can help to discover data trends that can be used to develop more effective cross-selling strategies. This is used to make relevant add-on recommendations to customers during the checkout process for online retailers.

Automated stock trading: Designed to optimize stock portfolios, AI-driven high-frequency trading platforms make thousands or even millions of trades per day without human intervention.

Imagine and create your example of AI that helps you in performing daily activities (in performing tasks at school, at home, for hobbies or in their free time).

- What are the characteristics of your fictional product/service?
- How does it make tasks easier?
- Why is it special and how will it win over future users?

Interactive Exercise

Find your way from school to your desired destination!

Use Google Maps to explore and find ways from school to your desired destination (your favorite place in the mountains, on the coast, fun park, etc.)

Analyze the obtained results. Which travel options are better and more acceptable to you, and which ones you want to avoid. Why?

Present your search and the path you have chosen to the teacher and students in the class.

Organize a class competition:

<https://www.transum.org/Maths/Investigation/CarPark/>

https://www.transum.org/software/SW/magic_square/magic_square.asp

<https://www.chesskid.com>

Think!



What do I know
now?



What did I learn?



What do I want to
know?

Conclusions

Artificial intelligence is used in the daily lives of all of us. It also makes it easier for us to perform various tasks.



AI Bingo ! What does “Artificial Intelligence” make you think of?

What do robots have in common?
What do they do? How do robots help us?

Examples:

Cozmo - visual recognition

Alexa/Google home - speech recognition

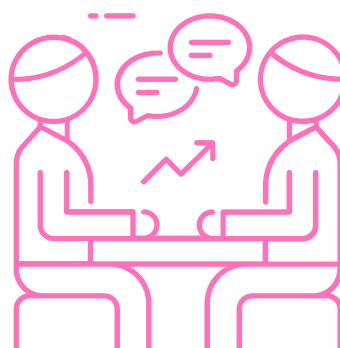
Are robots the only form of AI?

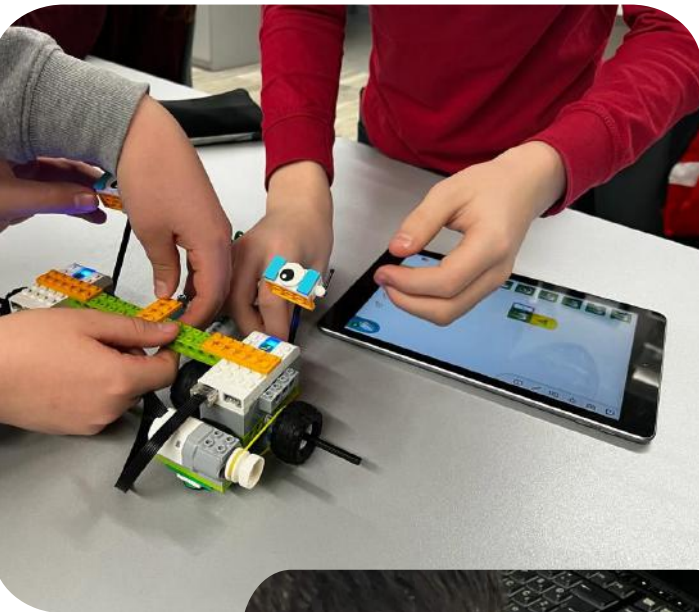
What about social networks: Instagram, YouTube, Facebook, Google,... ?

How many of you use these?

What do they have in common?

Today we will learn what the main task of artificial intelligence is: prediction. How can computers learn?

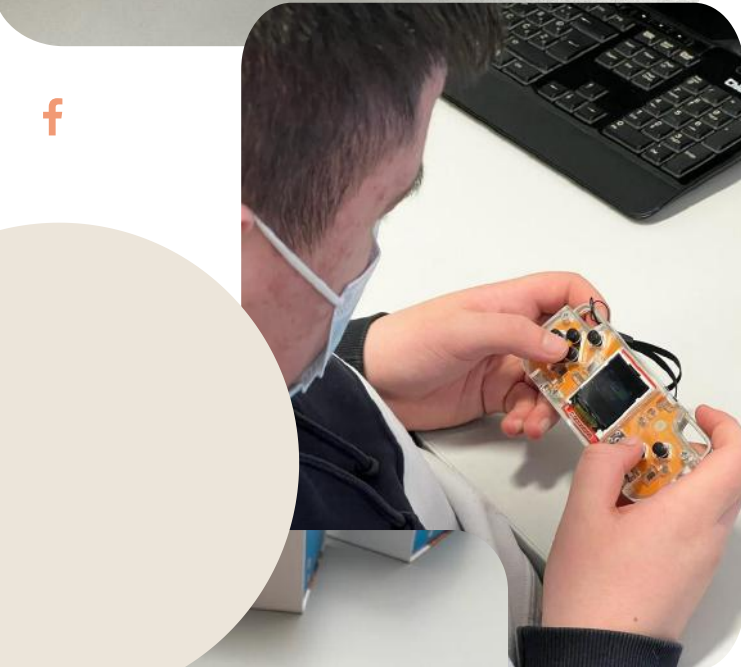




Short definition:

AI trying to predict something in the future or something that the data says.

Example of what you predict (weather, what happens in a movie).



What is the dataset?

A dataset is a collection of curated data

- Images
- Measurements (time, views, inches, etc)
- Text
- Video recordings!

The dataset contains a lot of separate pieces of data and can be used to train an algorithm with the goal of finding predictable patterns inside the whole dataset.

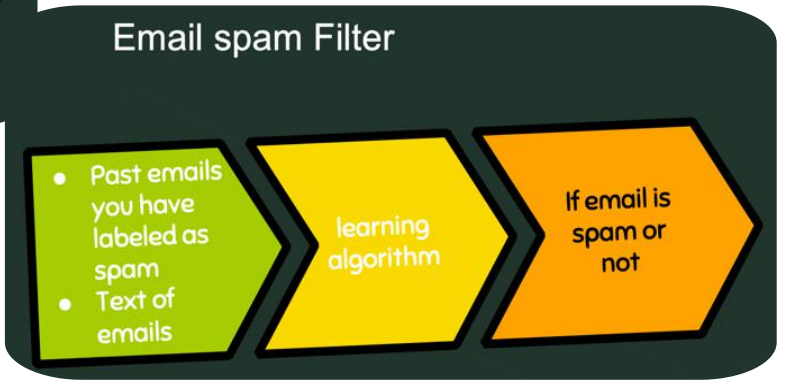
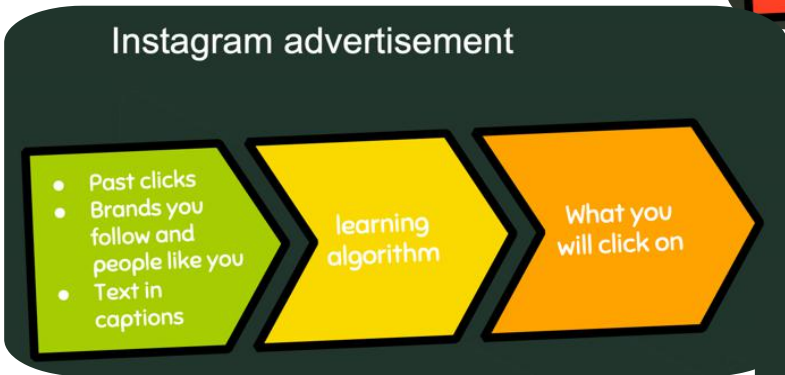
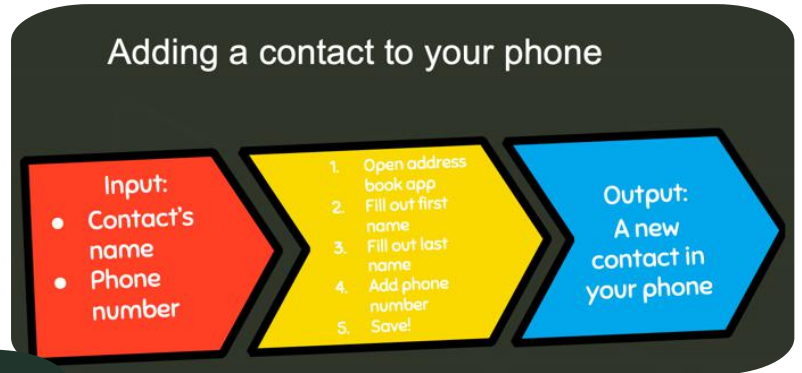
Isn't just numbers, weather data vs. online video data (gives you information).
Have you ever collected a dataset?

Artificial intelligence is a kind of algorithm...
there are other kinds as well.





What is Instagram's algorithm trying to pick?
 What kinds of ads do you get?
 What do you think they are trying to do?



<http://erasmus-artie.eu>

Interactive Exercise

AI Bingo: Learning by playing ftw. Payne, a researcher at MIT, AI bingo builds on pedagogical research that shows how exposing kids to the way technology works helps develop their interest in STEM and improve their job prospects later on in life.

Instructions:

Students must find a partner who has used an AI system listed on the card and together students must identify the prediction the system is trying to make and the dataset it might use to make that prediction. The first student to get five squares filled out in a row, diagonal, or column wins (or, for longer play, the first student to get two rows/diagonals/columns).



A.I. BINGO

<p>Gotten a weather forecast from a website or used a weather app</p> <p>Dataset: Prediction:</p>	<p>Sent a voice-to-text message</p> <p>Dataset: Prediction:</p>	<p>Used an online search engine like Google or Bing</p> <p>Dataset: Prediction:</p>	<p>Seen a Google autofill search result</p> <p>Dataset: Prediction:</p>	<p>Had a writing assignment graded by a computer</p> <p>Dataset: Prediction:</p>
<p>Used "safe search" on Google</p> <p>Dataset: Prediction:</p>	<p>Seen a suggested response on Gmail to an email</p> <p>Dataset: Prediction:</p>	<p>Used a Snapchat filter (what's your favorite?)</p> <p>Dataset: Prediction:</p>	<p>Played a motion-sensitive video game e.g. Mario Party, Nintendo, Wii U, etc.</p> <p>Dataset: Prediction:</p>	<p>Had an Emoji suggested instead of a word e.g. "lol" is replaced for an Emoji smiley face</p> <p>Dataset: Prediction:</p>
<p>Seen a sponsored product on Google or Amazon e.g. "since you bought ___, we thought you might like..."</p> <p>Dataset: Prediction:</p>	<p>Had an email go to your spam folder (was it actually spam?)</p> <p>Dataset: Prediction:</p>	<p>FREE</p>	<p>Clicked on an Instagram ad (what kinds of ads do you normally see on the app compared to your partner?)</p> <p>Dataset: Prediction:</p>	<p>Seen news articles suggested in a news app (what kinds of articles do you normally see compared to your partner?)</p> <p>Dataset: Prediction:</p>
<p>Had an email labeled as "important"</p> <p>Dataset: Prediction:</p>	<p>Seen a suggested ad on Snapchat (if so, what for? How does this compare to what ads your partner sees?)</p> <p>Dataset: Prediction:</p>	<p>Had a text auto-completed or used autocorrect</p> <p>Dataset: Prediction:</p>	<p>Listened to a recommended song on Spotify (what kind of music do you usually get recommended compared to your partner?)</p> <p>Dataset: Prediction:</p>	<p>Seen a recommended product on Facebook (if so, what for?)</p> <p>Dataset: Prediction:</p>
<p>Seen a "nudge" reminder on Gmail to respond to an email</p> <p>Dataset: Prediction:</p>	<p>Used a fingerprint to unlock a device or opened a device with your face</p> <p>Dataset: Prediction:</p>	<p>Used a map app to find a path to a destination</p> <p>Dataset: Prediction:</p>	<p>Used an app to recognize a song playing</p> <p>Dataset: Prediction:</p>	<p>Communicated with a customer service bot</p> <p>Dataset: Prediction:</p>



Conclusions

AI trying to predict something in the future or something that the data says. The dataset contains a lot of separate pieces of data and can be used to train an algorithm with the goal of finding predictable patterns inside the whole dataset.

Think!



What do I know now?



What did I learn?



What do I want to know?





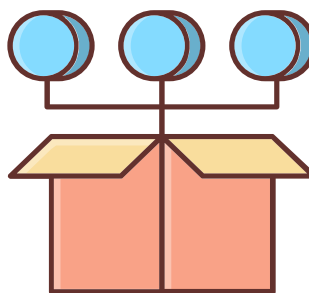
How can computers learn and solve problems?

Let's be aware of the changing ways of living under the influence of the newest technology.

Let's get to know artificial intelligence and how artificial intelligence is used in everyday life.

Can a computer or other smart devices think or solve problems?
Do we use AI?
How do we use AI? Where?
Can we use AI for learning?

Today we will learn what AI is and how we use it in everyday life.





Machine learning as a subfield of AI

Machine learning gives computers the ability to learn without being explicitly programmed. It is a subfield of computer science.

The idea came from work in artificial intelligence. Machine learning explores the study and construction of algorithms that can learn and make predictions on data. Such algorithms follow programmed instructions, but can also make predictions or decisions based on data. They build a model from sample inputs.

Machine learning is done where designing and programming explicit algorithms cannot be done.

Examples include spam filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), search engines and computer vision.

Neural networks

A neural network (also called an ANN or an Artificial Neural Network) is an artificial system made up of virtual abstractions of neuron cells. Based on the human brain, neural networks are used to solve computational problems by imitating the way neurons are fired or activated in the brain. During a computation, many computing cells work in parallel to produce a result. This is usually seen as one of the possible ways artificial intelligence can work. Most neural networks can still operate if one or more of the processing cells fail.

Neural networks can learn by themselves, an ability that sets them apart from normal computers. Today's computers cannot do anything they are not programmed to do.

There are three ways a neural network can learn: Supervised learning, Unsupervised learning and Reinforcement learning. These methods all work by either minimizing or maximizing a cost function, but each one is better at certain tasks.

Supervised learning

In Supervised learning, the neural network is trained by using example inputs and the correct output. The network can then work out the relationship between the input and output. For example, a network could be trained by showing it details about houses and the sale price. Once it has finished training it could estimate the sale price of another house by analyzing information like the number of bedrooms and local crime rate.



Another example is the ALV (Autonomous Land Vehicle). DARPA funded this project in the 1980s. In a demonstration in 1987, it traveled 600 meters at 3 km/h over difficult land, with sharp rocks, vegetation and steep ravines. This vehicle could drive itself as fast as 30 km/h. This network watched a 'teacher' drive and saw the road using laser radar. The learning process was repeated for different road types. ALV used a kind of neural network called a multi-layer perceptron in which multiple layers of neurons are connected in series.

Unsupervised learning

Unsupervised learning only trains using inputs, and the network has to figure out how they relate to each other. This method is used to solve Clustering problems, estimation problems, and self-organizing maps. For example, a self-organizing map can be used to categorize iris flowers by stem size and colour.

Reinforcement learning

A reinforcement learning neural network learns by watching a teacher's actions. It works out the smallest cost and tries to use this to work out how to make the smallest cost in the future. It can be thought of as a Markov decision process. Another simple way to think of this is as "carrot and stick" learning (learning that rewards good behaviour and punishes bad behaviour).

Recently, a research team from the University of Hertfordshire, UK used reinforcement learning to make an iCub humanoid robot learn to say simple words by babbling.

Class competition

- <https://www.autodraw.com>
- <https://quickdraw.withgoogle.com>
- <http://misfire.io>





Answer the following prompts:

1. Write a brief description of your tool.
2. Which stakeholders might be interested in this technology? Who might be affected by this technology most? Brainstorm at least 10 stakeholders.
3. If this technology was used for evil, how might that be done?
4. If this technology was used to help other people, who might it help?
5. In 50 years this technology could do the most good by...
6. In 50 years this technology could do the most harm by...

Present the results to the students in the class. Discuss. Save your work to the class e-portfolio.

Conclusions

*Machine learning gives computers the ability to learn.
Neural networks are used to solve computational problems by imitating the way neurons are fired
or activated in the brain.*



Think!



**What do I know
now?**



What did I learn?



**What do I want to
know?**



How do algorithms shape our world?

The purpose of this course is to understand machine learning algorithms.

An algorithm is a detailed, step-by-step process followed in order to accomplish a specific task or to solve a specific problem. We can define an algorithm by writing out the step-by-step instructions, thinking about things in terms of discrete steps. For example, our algorithm for a child's morning routine could be the following:

Wake up and turn off the alarm
Get dressed
Brush teeth
Eat breakfast
Go to school

Let's get acquainted with machine learning algorithms.



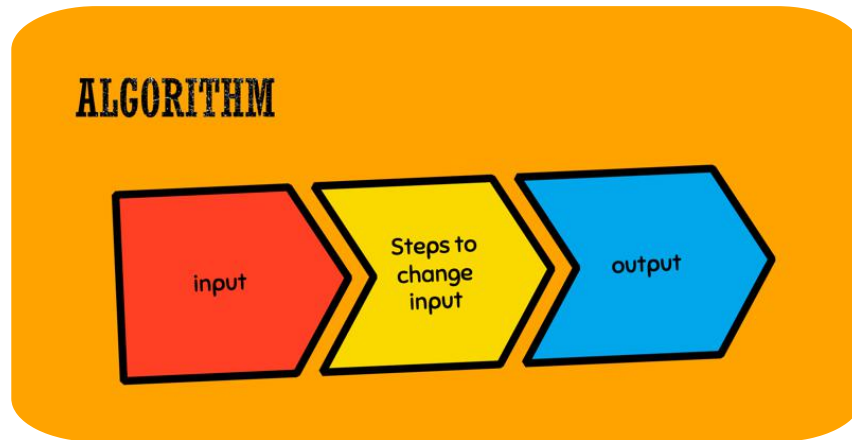


How do algorithms shape our world?

What an algorithm is?

What are the three parts of an algorithm?

An algorithm needs some input data and follows specific steps or instructions to give us the desired output



f Computers use algorithms, but so do humans.

Algorithms are a lot like a recipe.

For example, if we were baking a cake, the algorithm would take in the following ingredients, such as flour, sugar, salt, eggs, etc.

We would mix the dry ingredients and then mix in the wet ingredients like eggs or milk.

We would pour into a cake pan, set the oven to 180, and put the cake pan in the oven.

Our output would be a cake!



**Task:**

Write your algorithms.

Write an “algorithm” (or recipe) for the best sandwich. Be sure to specify what your inputs are.

What input data (or ingredients!) do you need?

Write out the steps in your algorithm.

What is the output of your algorithm?

Discussion:

Did any of you include instructions to put away your ingredients after you used them?

Then you were optimizing for tidiness in your algorithm!

Did any of you cut your sandwich into fun shapes? Cut off the crust?

Then you were optimizing for playfulness or aesthetics!

Computer algorithms also optimize for various goals, but sometimes this can be hard to spot.

What do you think the goal of Google’s search algorithm is?

Students might say “best” results. If so, we can ask them what word they would replace with best as they did earlier.

Students might also say “best results for me,” so we can prompt students to ask what they mean by that, or how Google might confirm that they’ve shown “the best results for me.”

We’re looking for answers that will likely get us to click on links, to get us to click on advertisers' links - things that show students understand the search results benefit Google first.

**Interactive Exercise**

Open up Google search under two different accounts (or one under an account that is logged in, and one in an incognito browser).

Search for some of the following items: pizza place, best movie, news.

What do you think, why are the results different?

Topics for discussion

Google's algorithm determines how pages on the internet are displayed and ranked based on their relevance to your search. In less than a second, interrelated search algorithms process information extremely quickly, interpreting your query and returning personalized results. Sites such as Amazon and Netflix base recommendations on collaborative filtering algorithms that look at other users with similar interests and tastes and subsequently deliver predictions for purchases and shows.

Mapping applications such as Google Maps need to calculate routes through cities, taking into account distance, traffic, and accidents. Tools such as Google Flights also consider routes through many airports while considering layovers, prices, and time.



Topics for discussion

In order for the player to win the game, what conditions have to be met?

How can we keep track of the score in our game?

How can we count the number of times each letter appears in a word?

What are the steps we have to take to swap the smallest and largest numbers in a list of numbers?

Interactive Exercise

Let's play the ArtBot game. In ArtBot players of all ages learn the basics of AI.

Players' quest is to find and retrieve stolen art objects. Players train their AI helper to recognize and locate the objects hidden in a maze of dungeons, and see how supervised and reinforcement learning work.



<https://learnml.eu/artbot.php>

visit: <http://learnml.eu/games.php>

Minecraft Learns ML is a short game showcasing how videogames can be used to teach core principles of the architecture of neural networks using imitation learning. In Minecraft Learns ML, players have the chance to familiarize themselves with creating a machine learning dataset and decide on the basic architecture of a neural network to solve a simple navigation problem on the fly.

Super Meat Bot is a video game for teaching the principles of reinforcement. In Super Meat Bot players have the chance to design levels with rewards and deterrents for an AI and train it to overcome more and more difficult navigation challenges.

Conclusions

From Google search to morning routines, algorithms are ubiquitous in our everyday life.

Think!



What do I know now?



What did I learn?



What do I want to know?



How do machine learning systems recognize what they see?

The purpose of this course is to understand how computers and other smart devices perceive and identify events from their surroundings.

How do smart devices see the world around them?

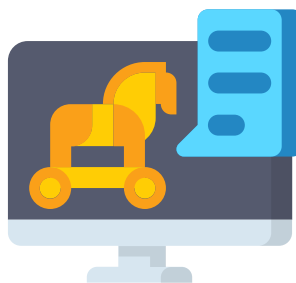
How do smart devices decide what to do?

How do machine learning systems recognize what they see?

Think about smart cars, web search, music and video recommendations.

Can devices learn?

Today we'll be learning how to train our own AI models!

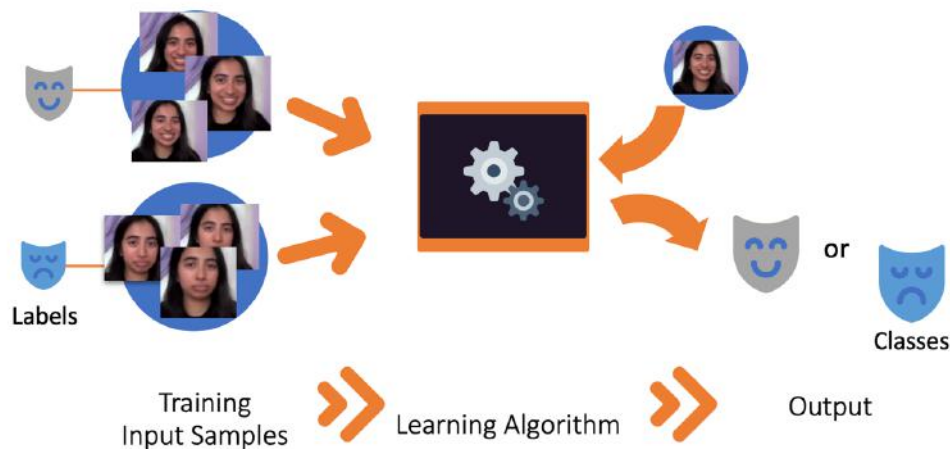




Machine learning is a technique to train computer systems to recognize images and motion. To train a machine learning model, we feed it a bunch of examples to learn from. This is like when you learn a new word, you have to see a bunch of examples of it to understand what it means.

Imagine we want to create an app that cheers you up when you are sad. To detect whether you are happy or sad, we could train a machine learning model to look at photos of you and recognize your smiles vs. frowns. We would feed in many photos of you smiling, and many photos of you frowning, and let the learning algorithm learn from that. In the end, we can have a machine learning system for our app that can tell us whether you are smiling or frowning!

How do machine learning systems recognize what they see?



In this example, the input examples (also known as input data) are photos of you smiling or frowning, each with a label of whether the photo is a smile or frown.

The learning algorithm takes these input examples and learns how to predict whether a given photo is a smile or frown. This is like recognizing patterns in a set of examples.

When the learning algorithm is trained, we can use it to classify or predict what label any photo has. So we can take a new photo, put it into the system, and it will tell us whether it thinks it's a smile or frown.



Teachable Machine

(<https://teachablemachine.withgoogle.com>) is a web-based machine learning tool that you can use to train a computer to recognize different types of data. We'll learn how to use Teachable Machine and discuss the importance of training data and representation in our very own machine learning models. This is like recognizing patterns in a set of examples.

» Topics for Discussion

What might the training input data be for a machine learning model?

What would the labels be that the training data is tied to?

How does a computer know what it sees?

Let's think about the learning step where the machine learning algorithm finds patterns between the labeled input data. Ultimately it comes up with a way to differentiate between the two sets of pictures.

What are some features (or characteristics) of these two groups of photos that help a computer recognize the difference?

Hint: All of these are going to be visual characteristics since the computer can't feel or taste the cereal (since we've only given it photos of the bits). More: size, color, texture, shape, background

What are the sorter's output classes?

What can the fully-trained machine learning model tell us about new images of cereal bits?

When the model has been trained, we can ask the model what categories it thinks new images are.

What might these categories (or output classes) be?

Hint: Output classes tend to correspond with the input labels.





Interactive exercise:

Teach a model to classify images using files or your webcam.

Go to: <https://teachablemachine.withgoogle.com>

Topics for discussion:

We might be able to confuse the model.
Did anybody notice that all the pictures I took have the same background?
What if we try changing the background?
How do we know that the model is getting confused?

How do I use it?

- 1 Gather**
Gather and group your examples into classes, or categories, that you want the computer to learn.
[Video: Gather samples](#)
- 2 Train**
Train your model, then instantly test it out to see whether it can correctly classify new examples.
[Video: Train your model](#)
- 3 Export**
Export your model for your projects: sites, apps, and more. You can download your model or host it online for free.
[Video: Export your model](#)

Get Started – New Project– Image Project



<http://erasmus-artie.eu>

Conclusions

Artificial Intelligence gives us a world of possibilities: we can train models to learn on numerous types of data and apply those models to help solve real human problems.

Think!



What do I know now?



What did I learn?



What do I want to know?



Can machine learning recognize poses ?

We trained models that recognize images and sounds. Can we make models recognize poses?

Today we'll be learning how to train our own AI models to recognize poses.

Machine learning models can be trained using different data representations. Images are one form of data representation, similar to pictures. Poses, which you may have noticed during your model training are represented as blue dots and lines, are a simplified form of data representation known as points (dots) and edges (lines). These points are calculated from your camera image using another machine learning model known as posenet.

Interactive Exercise

Teach a model to classify body positions using files or striking poses on your webcam.

Go to: <https://teachablemachine.withgoogle.com>

Present your model to the students in the class. Discuss. Save your work to the class e-portfolio.



>> Topics for Discussion

What do you think is different about the pose model?

Do you think that it would get confused if we used a different background?

Artificial Intelligence gives us a world of possibilities: we can train models to learn by processing numerous types of data and apply those models to help solve real human problems?

But what are some of the downsides of machine learning models?

What might go wrong?

Conclusions

Image models were trained on the entire images from the camera, while pose models looked only at the dots and lines that are recognized as a body pose.



Think!



What do I know now?



What did I learn?



What do I want to know?



What about the ethical side of the data that we use to train models ?

The purpose of this course is to understand and be able to answer to the following questions:

What is privacy ?
How do we protect our privacy ?

We'll be talking about the ethics of the data that we use to train these models





AI: privacy and ethical issues

AI technology is developed by humans, and created with human-centric goals in mind. Siri, Alexa and Google Home are not just robots, a person with an arsenal of computer engineering knowledge is responsible for the intelligent voices coming out of these sleek devices. The technology that runs on AI is programmed by people, which means that these machines are prone to the mistakes we make and the biases we have as human beings.

Just the fact that this intelligence is called "artificial" should be a tip, but the answer here is data. And the more, the merrier. When AI designers are developing a given technology, they need to collect an enormous amount of data to ensure maximum accuracy. The source of that data? You and I, our personal information, our daily routines, our interests, and how it all manifests itself online. With that, come issues of privacy, and how ethical it is to use information that was not necessarily intended for the development of a given technology. Because children are now getting cell phones before hitting their teens and living in a tech-filled world, it is important to know the ins and outs of the technology around them.

Data can be a pretty powerful tool for computational systems, but it must be used responsibly and ethically. Privacy is an important element in data usage because we might not want our personal information to be shared on a large scale.

Topics for discussion:

- How many of you would be okay with hackers having access to your email username? Your phone number? Your TikTok username?
- A picture of your face? What if people not only had access to them but could also trace what you post and what do you do by using those identifiers?

Conclusions

AI devices are not almighty robots. They are devices created with the minds and hands of people and just like people, they can make mistakes. More importantly, mistakes can be fixed and they can be the ones to grow up and fix them.

Think!



What do I know now?



What did I learn?



What do I want to know?



Let's create an AI project with Scratch!

The interaction between humans and AI is a two-way street: we respond to the outputs and behaviors of AI systems in robots and computers and AI responds to various forms of human behavior.

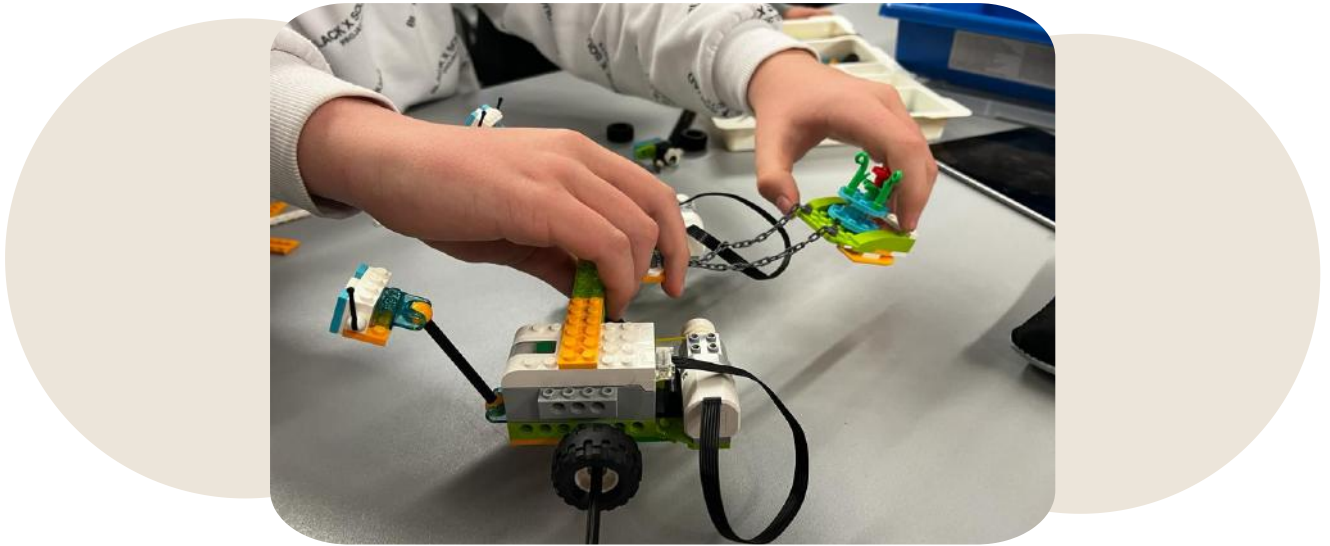
What do we as humans use motion for on a day-to-day basis?

We move to get places! Whether it's by walking, driving, riding a bike, swimming, wheeling, or other, we put in the effort to get ourselves from one place to another.

We also move a whole lot when we're playing sports. Let's not forget the role of motion and movement in the creative arts! Let's think about the ways we move when we're making art, making music, or performing theater.

Creative processes take many forms: some involve movement and others don't. Pieces of art that do use movement are quite powerful: think of a moving art piece like a hanging mobile swinging in the wind, or the synchrony of an orchestra or ballet troupe moving flawlessly to the melody of an opera.

Sometimes we express ourselves visually without even thinking about it. We also constantly use our facial expressions as markers for how we're feeling and others can read those expressions and understand our emotions or intentions.



Motion and movement are really powerful means of communication and a way in which people signal one to another. They help deliver messages and carry an idea across from one person to another or even hundreds of people. In this way, movement is one key medium through which people interact with one another and understand one another.



It looks like AI entities can move if they are programmed to do so. But can AI understand human movements? What do you think?

AI doesn't "understand" motion as you or I do, but it can be trained to extract specific patterns of movement and respond to them. The images and poses that we used on Teachable Machines are cues that a computer uses to extract meaning from movement, they represent the visual components of a movement or the position of your body during a movement, but they don't actually encode motion itself.

Announcement of the goal of the lesson:

We will look at AI systems that understand human movement, facial expressions, and other non-verbal based interactions.

Scratch is a tool used for creating. We can use Scratch to make stories, games, animations, art, music, and interactive AI systems. We'll be using a special version of Scratch that adds special AI features as extensions.

We learned about the difference between Teachable Machine models trained with image and pose data. Image models were trained on the entire images from the camera, while pose models looked only at the dots and lines that they recognized as a body pose.

To translate images to poses, Teachable Machine uses a pre-trained model known as "PoseNet", which is a machine learning model that has been trained on many many examples of images and their corresponding pose skeletons.

PoseNet takes the camera image and converts it to blue dots and lines. These dots are known as "key points".

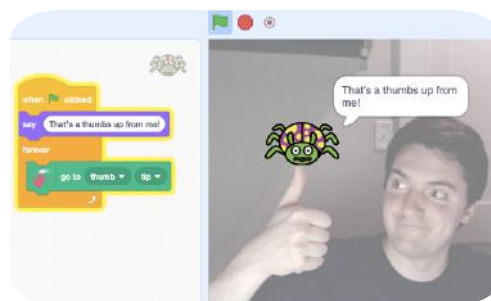
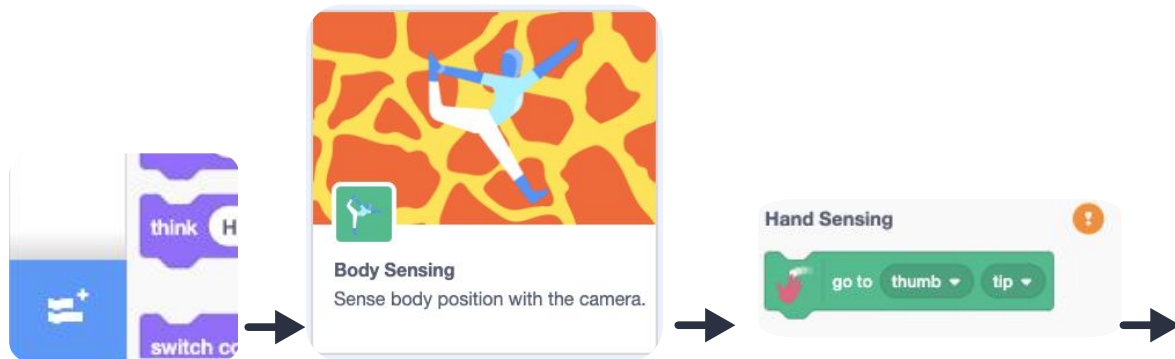
While we used these key points as input to our Teachable Machine models, we can use these same key points as parts of our Scratch projects!

The teacher shows, explains and guides the students in the first practical task:

To use PoseNet in our projects, we just need to bring in some extra blocks! We click on the "Add Blocks" blue button on the bottom of the screen. And then we select the "Body Sensing" extension.

f

<http://erasmus-artie.eu>





Once we add this, notice that our camera has turned on on the right side of the screen. The green checkmark next to "Body Pose Sensing" means the PoseNet model has a prediction of where the key points are in the camera image! If it has an orange question mark, it may mean that there is a problem with the camera or web app, or the model does not have a confident prediction. If you try covering the camera with your hand, for example, it will blink between the checkmark and yellow indicator.

If we stop the project and we click that block, our sprite moves to where the model thinks it sees a left shoulder pose keypoint in the camera scene.

If we put that block within a forever loop as we did before, we can see that it follows our shoulder.

The hand blocks usage a different pose model that is similar to PoseNet, from Google's MediaPipe team.

This model, instead of being trained on body images, has been trained on images of hands. The red dots in these examples are the key points, just like key points we get out of the PoseNet model.

Since each model you add to your project means your computer has to run every model many times per second (known as frames per second, or framerate of the model), you might want to save any work and refresh the browser page to unload the body model first.

After importing the hand sensing model, you'll notice we have a new block:

Note that unless your hands are visible from your camera, the indicator is orange, since it only shows green when it predicts that there is a hand in the view of the camera.

Interactive exercise 1:

- Open the Scratch.
- Create a New project.
- Add Blocks "Body Sensing".
- Create your project.
- Test and save your project.
- Present your project to the students in the class. Discuss. Peer evaluation.
- Save your work to the class e-portfolio.

The Face model is from an emotional AI company known as Affectiva. Similar to PoseNet and MediaPipe's Hand Model, Affectiva's model gives you key points on the face (like eyebrow, mouth, nose, etc.), but it also goes a step further and gives us "expressions", like "mouth open", "eyebrows raised", "blinking", etc.

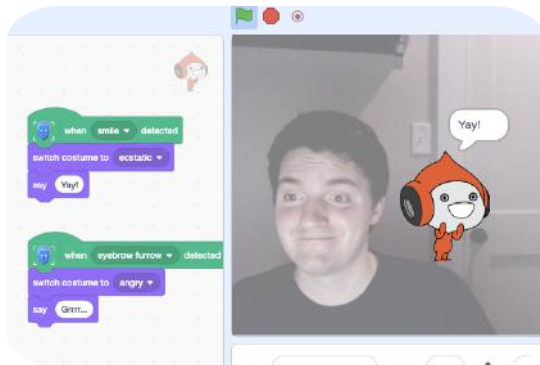
We can use the facial key points just like hand and body pose data. We can use facial expressions like "mouth open" as Events in our Scratch projects, just like how the "when Green Flag" block makes our code run.

Let's try changing costumes when we smile and furrow our eyebrows.

Since this model is more complex, and some of the output classes are more subjective, we may not expect the higher level insights about facial expressions, and especially emotions, to be as high confidence predictions as the various key points in the different models.



Examples to present and discuss:



and



Interactive exercise 2:

- Open the Scratch.
- Create a New project.
- Add Blocks “Face Sensing”.
- Create your project.
- Test and save your project.
- Present your project to the students in the class. Discuss. Peer evaluation.
- Save your work to the class e-portfolio.



Conclusion

We can build tools that will look for particular features, like the position of your eyes or the direction in which your finger is moving. These tools are designed and used by people - people like us decide how a computer behaves in response to human motion, meaning that people are in control of what an AI system does.

Think!



What do I know now?



What did I learn?



What do I want to know?



Let's get fun with an AI!

Revise the commands and skills of working in Scratch: adding blocks (Body Sensing, Face Sensing, Hand Sensing), opening a new project, adding backdrops and sprites, sharing a project, downloading a project to a computer.

Today we'll be learning how to train our own AI models to recognize poses.





Practical task

- Open the Scratch.
- Create a New project.
- Add a Sprite and Backdrop.
- Create your project by using blocks: Motion, Looks, Sound, Event, Control, Body / Hand / Face Sensing).
- Test and save your project.
- Present your project to the students in the class. Discuss. Peer evaluation.
- Save your work to the class e-portfolio.

In this version of the bird chirping project, we put our booleans inside a forever loop.



<http://erasmus-artie.eu>

Conclusions

We can build tools that will look for particular features, like the position of your eyes or the direction in which your finger is moving. These tools are designed and used by people - people like us decide how a computer behaves in response to human motion, meaning that people are in control of what an AI system does.

Think!



What do I know now?



What did I learn?



What do I want to know?



Let's recycle together!

Let's revise the commands and skills of working in Scratch and skills to train a model by Teachable Machine.

Careful waste management is one of the necessary habits needed by every inhabitant of the Earth.
Today we are going to create a recycling assistant using Scratch and Teachable Machine.





Go to:

<https://teachablemachine.withgoogle.com>

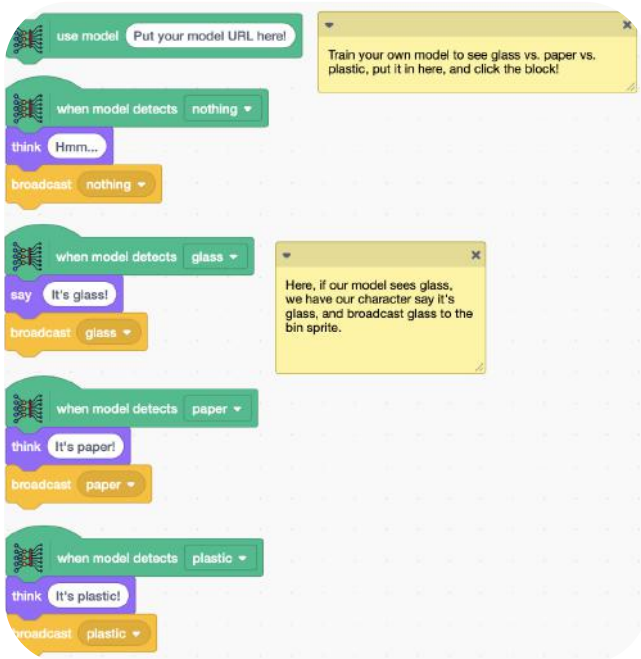
Teach a model to recognize glass vs. paper vs. plastic in your webcam.

Create a New project in Scratch.

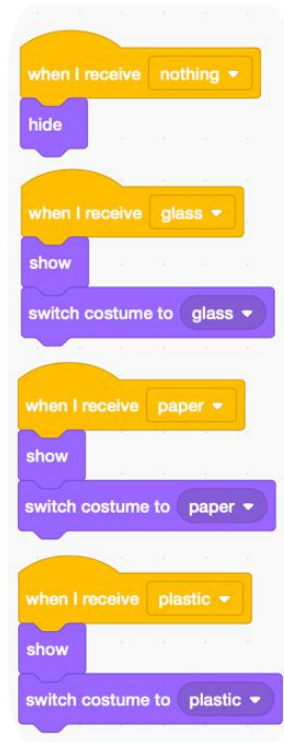
Add Video Sensing Extension.

Add a sprite (assistant) and associated blocks:

Copy and paste the URL of your model.



Add a new sprite and associated blocks:



Based on the previous example, students design their practical work individually and/or in pairs

Conclusions

We can build and use a recycling assistant using Scratch and Teachable Machine.

Think!



What do I know now?



What did I learn?



What do I want to know?



Let's chat!
Create a chatbot that can answer questions about a topic of your choice.

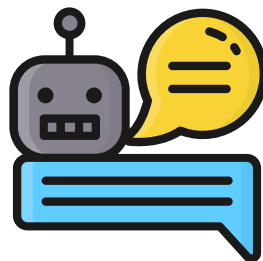
Let's repeat the commands and skills of working in Scratch and skills to train a model by Teachable Machine.

Can we talk with our computers? How?
Do you use chatbots? How and where?

Try chatbot:
<http://talktothetrex.com> is a good example of the bot. Try to get ideas of how to create your bot.

A chatbot is software that simulates human-like conversations with users via text messages on chat. Its key task is to help users by providing answers to their questions.

Today we are going to create a chatbot using Scratch and Teachable Machine





Create a chatbot that can answer questions about a topic of your choice.

The topic for discussion:

Decide on a topic for your chatbot. Choose something that you know well enough to be able to answer questions about. It could be a place, animal, history, program language, etc.

Think of five things someone might ask about your topic. e.g. for your town, this could be:

What do owls eat?

Where in the world do owls live?

How long do owls live?

What types of owls are there?

How big do owls grow?

Instructions (steps):

Go to <https://machinelearningforkids.co.uk/> in a web browser.

Click on “Get started”.

Click on “Try it now”.

Click on “Projects” on the top menu bar.

Click the “+ Add a new project” button.

Name your project and set it to learn how to recognize “text”.

Click the “Create” button.

Click on your new project in the projects list.

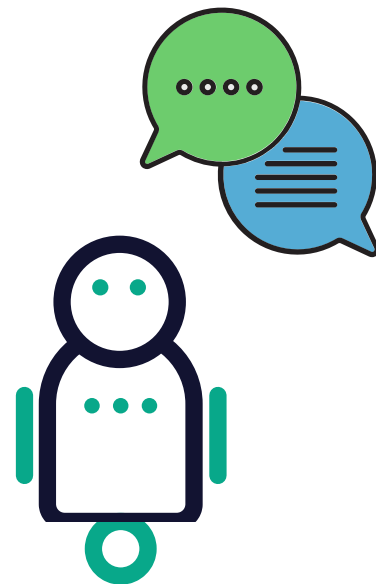
Click the Train button.

Click the “+ Add new label” button.

Type in one word that sums up the first of your things from Step 2, then click Add. We used “food” to sum up questions like “What do owls eat?”

Do that again for all of the things in your list from Step 2. The words you choose don't matter, as long as you understand what they mean.

Click the “+ Add example” button in one of the buckets.





Type in an example of how someone might ask that question.

Click "Add".

Repeat until you've got five examples of how to ask that question.

Repeat until you've got at least five examples in every bucket.

Click on the "< Back to project" link

Click the "Learn & Test" button

Click the "Train new machine learning model" button. As long as you've collected enough examples, the computer should start to learn how to recognize questions from the examples you've given to it.

Wait for the training to complete. This might take a couple of minutes. It's finished once you see the "status" change to "Available".

Click the "< Back to project" link.

Click the "Make" button.

Click "Scratch 3".

Click the "Open in Scratch" button.

Open the Owls project template. Click Project templates and find Owls in the list of templates.

Create this little snippet of a script but don't attach it to anything yet. Make sure you choose "owl says" for the orange block.

Duplicate it four times and join them all together. Right-click on it, and click "Duplicate".

Fill in each copy of the block. Drag the label for one of your questions into the top space, and type the answer to the question into the bottom space.

Drag this new block into the Green Flag block prepared for you. Remove the "Sorry. I haven't been taught anything yet." block and replace it with your new chunk of the script.

Draw your chatbot. You need to draw your character. If you provide different costumes, you can animate your character while it talks.

Test your chatbot! Click the green flag and try asking a question.

Present your project to the students in the class. Discuss. Peer evaluation.

Save your work to the class e-portfolio.



f

http://erasmus-artie.eu

Conclusions

A chatbot is software that simulates human-like conversations with users via text messages on chat. Its key task is to help users by providing answers to their questions. We can create a chatbot using Scratch and Teachable Machine.

Think!



What do I know now?



What did I learn?



What do I want to know?



Let's create a smart device in a smart home

Do we use smart devices every day?
What are smart devices?
Why do we call these devices smart devices?
Can houses be smart?
Which houses are "smart"?

A smart home refers to a convenient home setup where appliances and devices can be automatically controlled remotely from anywhere with an internet connection using a mobile or other networked device.

Today we are going to create smart devices as a part of the smart home using Scratch.





Instructions (steps):

Stage:

```

when clicked
  forever
    ask Enter your command and wait
    if answer = Turn on the fan then
      broadcast turn-fan-on
    if answer = Turn off the fan then
      broadcast turn-fan-off
    if answer = Turn on the lamp then
      broadcast turn-lamp-on
    if answer = Turn off the lamp then
      broadcast turn-lamp-off
  
```

Lamp:

```

when clicked
  switch costume to lamp-off
  when I receive turn-lamp-off
  switch costume to lamp-off
  when I receive turn-lamp-on
  switch costume to lamp-on
  
```

Fan:

```

when I receive turn-fan-on
  set on to yes
  repeat until on = no
  next costume
when I receive turn-fan-off
  set on to no
when clicked
  set on to no
  
```

Present your project to the students in the class. Discuss. Peer evaluation. Save your work to the class e-portfolio.

Conclusions

A smart home refers to a convenient home setup where appliances and devices can be automatically controlled remotely from anywhere with an internet connection using a mobile or other networked device.

We can create smart devices as a part of the smart home using Scratch.

Think!



What do I know now?



What did I learn?



What do I want to know?



What programs can do? Let's meet the basics of Codey Rocky and mBlock 5.

Codey Rocky is a tiny yet versatile robot.

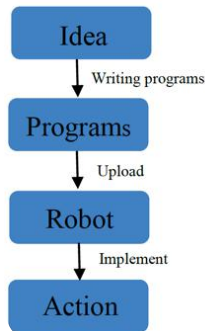
Apart from Codey Rocky, can you think of any other robots?
What are those robots used for?
How can those robots understand our instructions?

The goal of this lesson is to understand the concept of a program and what programs can do, as well as the basics of Codey Rocky and mBlock 5.





Program is an artificial language that we use to tell robots what to do. We translate our instructions into a piece of the program. Then we upload the program to the robot, making it do a variety of things as programmed.



Questions for discussion:

Do you know what the answer is now?

What's the secret of Codey Rocky?

Do you want to write code and upload the code to Codey Rocky?



<http://erasmus-artie.eu>

Codey Rocky is an educational programmable robot. You can use software to code the robot, manipulating it to do a variety of things you can imagine. It's also a good companion that can help children learn to code. With mBlock 5, children can master the basics of coding and develop logical thinking as well as computational thinking. Also, Codey Rocky supports technologies like AI and IoT, which expose children to the latest cutting-edge technologies.



Codey: As the brain of the robot, Codey is equipped with a variety of sensors and programmable blocks. It can work individually and can also work with Rocky to perform more tasks. Now pick up your Codey. Let's take a look at what sensors it has.



Feature	Block Name	Applications and Core Functions
Input	IR transmitter and IR receiver	Facilitates communication between Codeys and remote controls
	Gear potentiometer	Adjusts the input signals
Input	Gyroscope	Detects how Codey moves and the angles.
	Buttons	Buttons can be programmed to control the facial expressions, motions, and sounds of Codey Rocky.
	Light Sensor	The sensor is used to measure the volume of sounds in the surrounding environments.
	Sound Sensor	The sensor is used to measure the light intensity of surrounding environments.
Output	Speaker	The sensor can be programmed to play music.
	RGB indicator	The indicator can glow in different colors.
	LED matrix screen	The images, texts and time displayed on the screen are all customizable.

Rocky serves as the chassis of Codey. It adds more abilities to Codey, like avoiding obstacles, identifying colors, following lines and more.



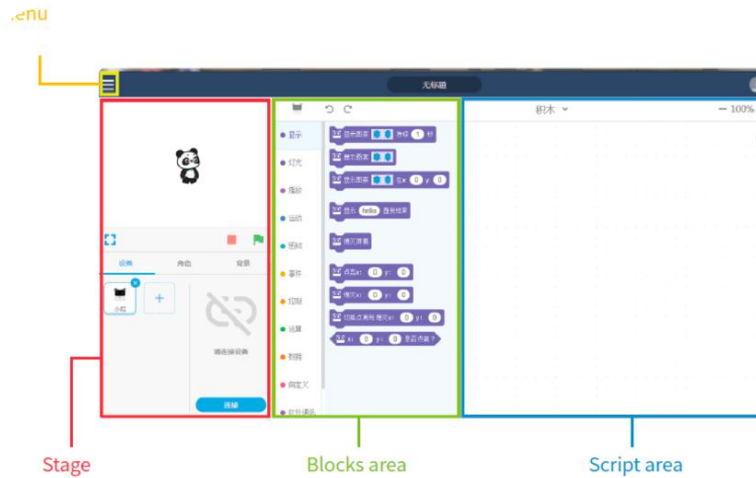
Features	Name	Applications
Input	IR Color Sensor	The IR Color Sensor integrates a color sensor, a grayscale sensor, an IR proximity sensor. By toggling the IR Color Sensor, you can make Codey Rocky perform a variety of fun tasks, like avoiding obstacles, following lines and more.
Input	Motor	The motor is used to control the motion of Rocky.

About mBlock 5

mBlock 5 is a programming tool that supports block-based and Python programming languages. It's developed based on Scratch 3.0, an open-source software tool that's developed jointly by MIT and Google. Using mBlock 5, you can write programs that tell Codey Rocky or other robots to do whatever you would like. You can even take advantage of the software to create stories, games, and animations that are engaging and unique. Moreover, mBlock 5 exposes children to technologies, like AI, deep learning and model training. In a word, mBlock 5 can be one of the best options for first-time coders. Have students open mBlock 5 PC and walk them through the interface.



Startup Interface:



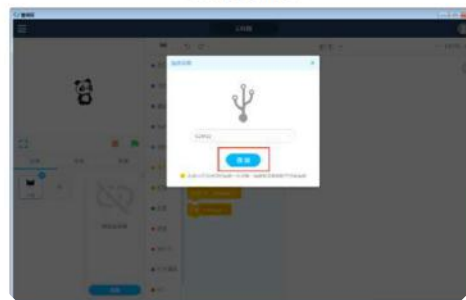
1. Stage: In this area, you can show your projects, connect devices and upload programs, add sprites and backgrounds.
2. Blocks area: You can find the blocks you need by color or category.
3. Script area: You drag blocks to this area to form programs.
4. Device/Sprites/Backgrounds Setting area: From here, you can find the devices, sprites, and backgrounds you need.

Task

1. Connect to a computer: Connect Codey to the computer via the USB cable. Then power on Codey.
2. Select the serial port: Open mBlock 5, click Connect, and select the correct serial port.

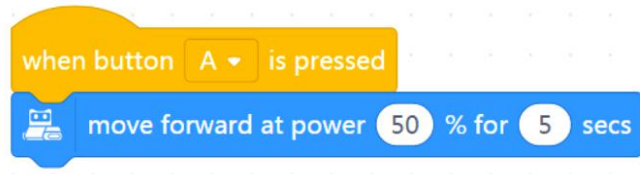


Click Connect

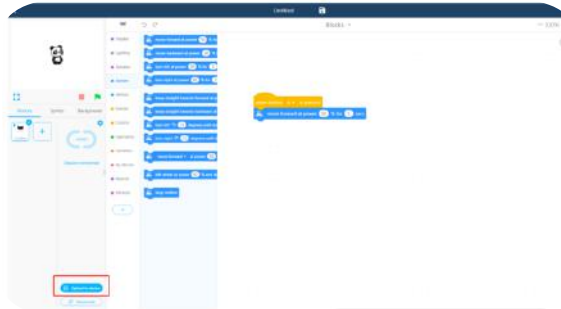




3. Use mBlock 5 to create a piece of the program as shown below:



4. Upload the program to Codey.



5. Unplug the USB cable and put Codey Rocky on the table. Press button A and observe how Codey Rocky reacts. Have students work in pairs to complete the tasks above by writing programs.

About Events

Expose students to the concept of Event. Tell students what an Event refers to. For example: When it gets dark and we enter a room, we need to turn on the light. To turn on the light, we need to press the light button. In this case, pressing the button is an event and that the light is turned on is the result.

Invite students to play a game, helping them have a better understanding of the concept.

Game rules:

Divide the students into 3 or 4 groups.

Draw some figures on the blackboard, like triangle, circle, square and star.

Define 3 or 4 events:

When you put your hand on the triangle;

When you put your hand on the circle;

When you put your hand on the square;

When you put your hand on the star.

The four events above trigger the actions below:

When you put your hand on the triangle—the 2nd group of students stands up;

When you put your hand on the circle—the 4th group of students stands up;

When you put your hand on the square—the 1st group of students stands up;

When you put your hand on the star—the 3rd group of students stands up.



Game procedure and teaching preparation:

Draw figures on the blackboard.

Divide students into groups and have them get ready for the game.

Put your hands on a shape randomly and check whether students react as required.

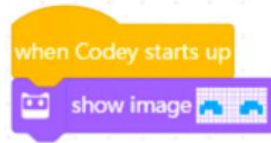
If students react as you expect, then you put your hand on another shape. If students fail to react as required, you need to repeat the game rules to students.

Repeat the game several times and speed up the process of switching between shapes.

You need to give a summary: In this case, the hand serves as an event. When the hand points to one shape, one specific group of students is expected to stand up as required.

Task 1:

Learn how to set events. Write programs to make Codey Rocky change its facial expressions based on the events. (when button A/B/ C is pressed).



Task 2:

Write programs to make Codey Rocky react in response to the events (when button A/B/C is pressed), like changing its facial expression or making different sounds. Have students share their projects

Examples:



Conclusions

Program is an artificial language that we use to tell robots what to do. We translate our instructions into a piece of a program. Then we upload the program to the robot, making it do a variety of things as programmed.

The interface of mBlock 5 consists of: Stage area, Blocks area, Scripts area, Device/Sprites/Backgrounds Setting area.

An event is the beginning of a piece of a program. When you write programs, the first thing you need to do is to select an event.

Think!



What do I know now?



What did I learn?



What do I want to know?



Codey Rocky meets sequences and loops

What is an Event?

Can you think of any events in daily life?

What events were used in the last lesson?

An event is an action that can cause things to happen.

Pressing the button leads to the light bulb lighting up.

In this case, pressing the button is an event and that the light bulb is turned on is the result.

Events that are used in the last lesson include: when the program starts up, when button A/B/C is pressed.

What are the steps to put milk into a refrigerator?

To put the milk into the refrigerator, you need to take these steps: open the refrigerator, put the milk into the fridge, close the fridge door. If you are not taking the steps, you will be not able to put the milk into the fridge.

Can you think of any cases in which you need to follow a set of steps to achieve something?

The goal of this lesson is to understand the concept of sequence and loop, as well as to develop a program to use it.





What are sequences and loops?

When we are programming, we arrange the blocks in order from top to bottom to form a set of steps. In this way, robots can follow the steps to perform a task. We refer to the set of steps as a Sequence.

Traditionally, animation designers would make animations by following these steps: Put a sheet of static drawing on the table first and unfold a new drawing paper on top of the first paper.

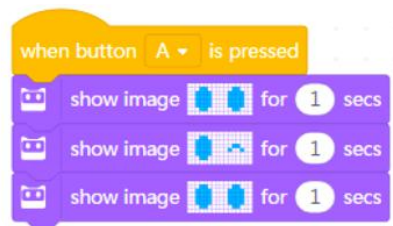
Designers would outline the frame and then change the drawing bit by bit at a time. Then another piece of paper, outline the frame and change the drawing slightly again. Designers repeat the steps over and over again until they complete a series of pictures that are slightly different from each other. Then, they flip the drawings quickly to animate the pictures. Based on the same principle, we use the block "show image () for () secs" in our program to create animations.

Task 1: Winking Eyes

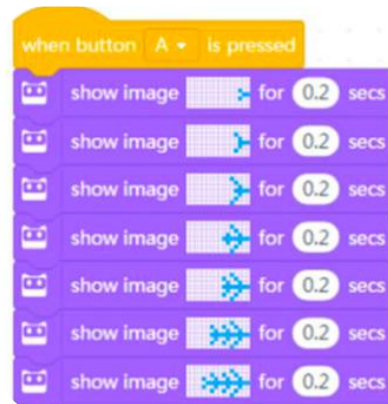
Create animations using the block "show image () for () secs". The method is simple: Use the image of the block as the base, duplicate the block, and change the image slightly. Repeat the steps and then arrange those blocks in sequence.



<http://erasmus-artie.eu>



Task 2: Growing Tree





Task 3: Walking

```

when button B is pressed
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  show image [robot] for 0.2 secs
  
```



About Loops

In programming languages, a Loop refers to a set of instructions that need to be executed repeatedly. When writing programs, we can use the Repeat block to replace those instructions that are executed repeatedly, making the code clean and concise. With the Repeat block, we can also create fun projects.



<http://erasmus-artie.eu>

We can use a loop to make the beats look neat and the game easier to play. In programming, the Repeat block is frequently used to make the code neat. With the Repeat block, Codey Rocky can carry out the same sequence of actions repeatedly. When the Repeat block is executed once, we count it as one cycle or one iteration.

Task 4: The Steamed Bread Can't Jump

```

when button A is pressed
  repeat 10
    show image [bread] for 1 secs
    play sound jump
    show image [bread] for 1 secs
  
```





Infinite Loop refers to a piece of code that is repeated endlessly. Take a close look at the Forever block. It doesn't have a bump at the bottom, which means you can't add another block to the end. This is because the code inside the Forever block will be executed endlessly. Using the Forever block, you will be able to create projects that are engaging and fun.

Task 5: The Jumping Steamed Bread

```

when button A is pressed
repeat 10
  show image [steamed bread] for 1 secs
  play sound jump
  show image [steamed bread] for 1 secs

```

Task 6: Animation

Design two animations. Give each animation a specific event (button pressed). But use the Repeat block to create one animation and use the Forever block to program the other animation. You can give the animations storylines, like one being The Steamed Bread Can't Jump and the other being The Jumping Steamed Bread; or the two animations can be mutually independent.

Conclusions

Sequence refers to a series of steps that are carried out to complete a task.

In computer programming, a Loop means the computer repeats a piece of code over and over again.

We use the Forever block to repeat a piece of code endlessly.

Think!



What do I know now?



What did I learn?



What do I want to know?



Codey Rocky drives safely and responsibly

What is a condition?

Do we make decisions in everyday life?

What kind?

Sample answers:

Every day we make decisions about what we wear: lighter or warmer clothes. We make decisions depending on the weather conditions.

Can robots make decisions?

Can robots see and feel the environment around them?

The goal of this lesson is to understand the conditions, meet and use Codey Rocky sensors to complete tasks.





Conditional refers to the instructions that depend on whether something is true or false. The instruction will be executed only when the if statement is true; otherwise, the program will skip the instruction. In the conditional block, there is a hexagonal hole, you need to put the condition code inside the hole.

Hexagonal blocks return Boolean values (with only two possible values: true or false). A Boolean variable has only two values, 1(true) and 0 (false). If the if statement is true, the Boolean variable will return the value "true". Otherwise, it will return the value "false". Therefore, we also refer to the hexagonal blocks as Boolean blocks.

Game: Conditional Box

The teacher will hand out some paper strips. Write down some conditional statements on the paper strips. Please make your statements easy to identify. Here are some examples for you: If you have long hair; if you wear glasses; if you are dressed in black; if your name includes the letter "A"; if you were born in June; if someone puts his or her hands up. Don't make your statements hard to identify (like "if it rains tomorrow" or "if the amount of your hair is an odd number"). These statements will be considered invalid.

Fold the paper in half and put it into the box.

Follow the teacher's instructions and pick a paper strip out of the box. Open it, read out loud what it says and make a decision. If the conditional statement is true, carry out the action. If it's false, then skip it.

Fold the paper in half and put it back into the box. Go back to your seat or hand the box to the next student.

Before writing programs, you need to identify where the color sensor is located.

There is a row of sensors at the front bottom of Rocky, including Color Sensor, Grayscale Sensor, IR Proximity Sensor and other sensors.



Infrared Color Sensor

The Color Sensor is in the middle. It can identify multiple colors, like red, green and blue.



Task 1: Racing Game

Create and test a program:

If Codey Rocky sees the green flag waving, it will start at its top speed.

When button A is pressed, Codey Rocky is getting ready at the starting line (play the sound ready).

If the color detected is green, Codey Rocky will move forward at its top speed.

The RGB LED will turn red once the program makes a decision based on the situation.

If Codey Rocky identifies a red item, it will move backward.

The IR Proximity Sensor is inside the Color Sensor (the left arrow indicates the IR receiver and the right arrow indicates the IR transmitter). With the IR Proximity Sensor, Codey Rocky can avoid obstacles. But you need to make sure the Color Sensor always faces forward when it is detecting obstacles.



Task 2: Avoid the Obstacle

Create and test program:

f When meeting an obstacle, Codey Rocky will avoid it and keep moving forward.

Place an item in front of Codey Rocky.

When button A is pressed, if Codey Rocky detects an obstacle, it will turn right by 90 degrees, move forward, turn left by 90 degrees and keep moving forward at a rapid speed.

If Codey Rocky fails to detect any items, it will move forward at its top speed.

Let Codey Rocky display facial expressions and make a sound when it meets an obstacle.

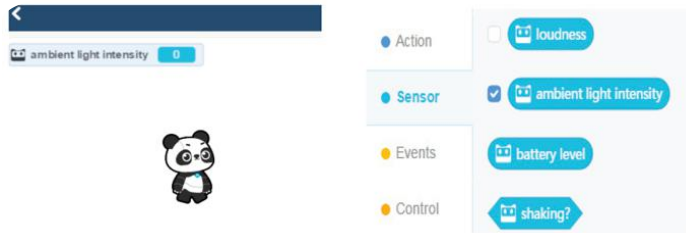
At the bottom right corner of Codey, there is a black dot. The black dot is the Light Sensor. It is used to measure the light intensity of surrounding environments. In the middle of Codey, you will find the RGB Indicator. It can shine in colors.





There are two ways to access the value of the Light Sensor:

When the device is connected, tick the „ambient light intensity" block to display the value on the stage.



Or you can use the scripts below to display the value on Codey's LED screen. Upload the programs above to Codey. Observe how the value changes on the stage and the LED screen.



In daily life, we need to compare values. Based on the results, we make decisions. Different decisions often lead to different actions. For instance, "If my temperature is beyond 38, I need to go to the doctor." Based on the thermometer value, we'll decide whether it's necessary to go to the doctor.

Similarly, we can use Comparison Operators in mBlock 5 to compare two values. The Comparison Operators include: < operator, = operator and > operator.

Comparison Operators	Implication	Example
	Less than	The ambient light intensity value is less than 20.
	Equality	The ambient light intensity value equals 20.
	Greater than	The ambient light intensity value is greater than 20.



Task 3: The Tunnel

Create and test program:

When entering a dark tunnel, Codey Rocky will turn on its light and slow its speed.

Stick a black paper on the Light Sensor of Codey Rocky.

When button A is pressed, Codey Rocky will move forward at its top speed.

If the light intensity is below 20, Codey Rocky will turn on its white RGB indicator and drive at a slower speed.



Conclusions

In programming languages, Conditional refers to the instructions that depend on whether something is true or false.



Think!



What do I know now?



What did I learn?



What do I want to know?



Codey Rocky meets Functions

Washing hair requires three steps: shampoo your hair, massage hair to form foam and rinse the foam off. But if we don't use the phrase "wash hair" to describe the set of steps, what will be the situation then?

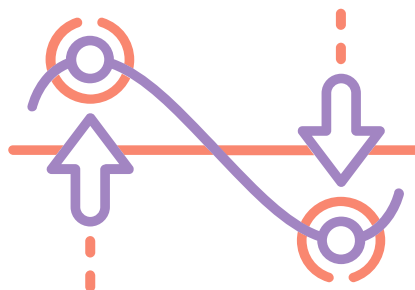
In everyday life, we often give a set of actions a single name and use the name to refer to the whole set of actions when necessary. After we name the set of actions as "wash hair", the situation will be like this:

When your friends are asking you out, you will say, "I'm going to wash hair. Wait."

Use a simple phrase to name the set of actions. The name is what we call a function.

In programming, we use a function to name a set of instructions and call the function in the code if necessary. The first thing to create a function is to give the function a name. Next, you need to define the function by adding instructions.

The goal of this lesson is to understand the functions, to define the function by adding instructions.

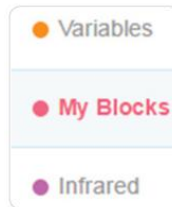




Create a startup function for Codey Rocky, making sure the function will automatically run when Codey Rocky starts up.

Open the mBlock 5 and connect Codey to the software. Follow the instructions to complete the challenge.

Click My Blocks at the category bar and select Make a Block.



Then, the “define” starting up block will appear in the Scripts area.



What instructions should be run when Codey Rocky starts up? Design programs under the defined starting up block.



After defining the function, you can directly call the function by adding the starting up block to the bottom of the event block when Codey Rocky starts up.



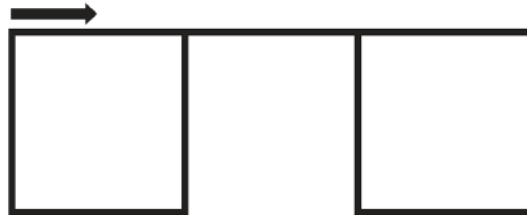
Improve on the sample project. You can change the animation or the sound. Select an image from the built-in images and refine the image as you would like. Then create your animation using the image.





Task 2:

Imagine Codey Rocky is a security guard. It is patrolling the passages in the building to make sure that all the properties are safe. Now, it's patrolling on the 1st floor. Design programs to make Codey Rocky follow the black lines as shown below.



Create a function and name it Square.



You might need to use the following blocks.

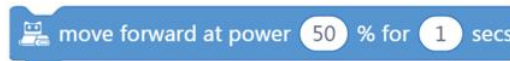


Figure out how much time it takes Codey Rocky to drive along the square-shaped line and the linking line.

Measure the length of one side of the square.

Measure the length of the linking line.

Calculate how much time it takes Codey Rocky to drive along a square.

Calculate how much time it takes Codey Rocky to drive along the linking line.

Calculate the speed of Codey Rocky (per second). (For instance, if Codey Rocky is programmed to keep moving forward at a power of 50% for 1 second, what is the driving distance then?)

Call the function Square two times under the event block when button "A" is pressed.

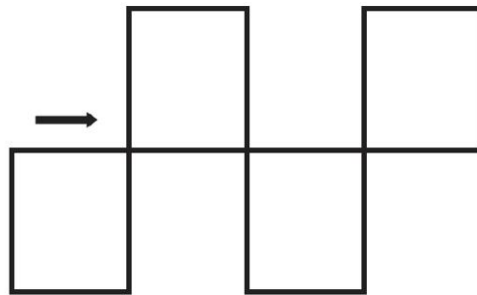


Notes

- The first step is to measure the length and width of the roadmap. Based on the measurements, you need to figure out how much time it takes Codey Rocky to complete the route.
- The motor of Codey Rocky is a DC motor so Codey Rocky is not able to make turns or follow lines precisely. In this case, it's fine that Codey Rocky roughly follows the line.
- Codey Rocky is programmed to move forward and turn right. After Codey Rocky repeats the behaviors four times, it will go back to the starting position.
- Since there are two squares in the roadmap, you are supposed to call the function Square at least twice in the code.

Task 3:

Codey Rocky comes to the 2nd floor. There are more rooms and the route is more complex. Design programs to make Codey Rocky drive along the black line as shown below.



You need to create two functions, Upper square and Bottom square.

Figure out how much time it takes Codey Rocky to drive along the square-shaped line and the linking line.

- Measure the length of one side of the square.
- Measure the length of the linking line.
- Calculate how much time it takes Codey Rocky to drive along a square.
- Calculate how much time it takes Codey Rocky to drive along the linking line.
- Calculate the speed of Codey Rocky (per second). (For instance, if Codey Rocky is programmed to keep moving forward at a power of 50% for 1 second, what is the driving distance then?)

Call the function Square two times under the Events block when button "A" is pressed. Using the repeat block will make your code concise.

Notes

- Remember to create two functions, Upper square and Bottom square, in your code.
- Measure the length and width of the roadmap. Based on the measurements, you need to figure out how much time it takes Codey Rocky to complete the route.



- The motor of Codey Rocky is a DC motor so Codey Rocky is not able to make turns or follow lines precisely. In this case, all you need to do is to make sure your Codey Rocky roughly follows the line.
- There are multiple ways to make Codey Rocky take the route as shown above. You can work on your own to figure out the solution first, or you can complete the challenge based on the following pseudocode:

Bottom square	Upper square	When button A pressed
move forward 1 second turn right by 90° Repeat four times	move forward 1 second turn left by 90° Repeat four times	bottom square move forward 1 second upper square move forward 1 second Repeat two times

Conclusions

In programming, Function is a custom coding block. Function refers to a set of instructions that can be called repeatedly in the code.



Think!



What do I know now?



What did I learn?



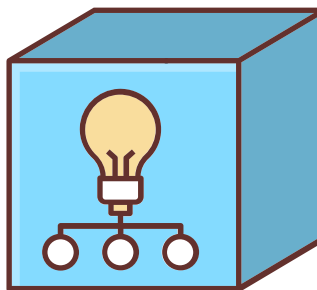
What do I want to know?



Rock Paper Scissors

Imagine that a variable is a box. You can put a value into the box and replace it with another value anytime. It works like a scoreboard which is used to record the scores during a contest. As the contest proceeds, the scores keep changing. So you can change the scores on the board accordingly. In computer programming, a variable is a storage location paired with an associated name. When a variable is created, the computer will leave a storage location to store the variable and give the location a symbolic name. By calling the variable name, programs can read and change the value in the storage location.

The goal of this lesson is to understand the functions, to define the function by adding instructions.





Let's play the game The Squirrel's Nuts Box

Take these steps:

There is a little squirrel. He has a nuts box. Every day, the squirrel puts nuts into the box and takes nuts out of the box, so the number of the nuts keeps changing all the time. One day morning, the little squirrel gets up, opens the box and finds out that there are 10 nuts left. The number of nuts will change based on different situations.

For instance, the squirrel is hungry and eats two nuts, so there are 8 nuts left. Then the squirrel picks another 5 nuts so there are 13 nuts in total. Now let's play a game, counting how many nuts are left.

The teacher writes down the number 10 on the blackboard and tells students there are 10 nuts left in the box.

Students take turns picking a paper strip out of the box.

Students read out what the paper strips say and try to figure out the number of nuts left based on the information. Then write down the result on the blackboard.

In the morning, the little squirrel eats 2 nuts for breakfast.

If the maximum temperature exceeds 5°C , the squirrel will pick 10 nuts in the wild.

At noon, the squirrel eats 3 nuts for lunch.

In the afternoon, a little bird pays a visit and gives the squirrel 5 nuts.

Today is the monkey's birthday. The squirrel gives him 3 nuts as a gift.

In the evening, a rat steals 4 nuts.

The squirrel plays the rock-paper-scissors game with a friend three times. If he loses the game, he gives away a nut; if he wins the game, he gets a nut; if it's a draw, then no gain or loss for both sides.

In the evening, the squirrel eats only 1 nut to keep fit.

It's hot today. The squirrel buys a straw hat from Aunt Bear. It costs him 7 nuts.

If the amount of the nuts in the box is less than 5, the squirrel will pick 10 nuts outdoors.

If the amount of the nuts in the box exceeds 12, the little squirrel will turn in a circle happily.





What's the variable in the game? How did you name it?
 What's the initial value for the variable?
 What operations were done with the value?

Task 1:

Using the variable to assign a value, you can make Codey Rocky move forward at any speed you would like. When button A is pressed, set the speed to 30, and Codey Rocky will move forward at a speed set for 1 second.

```

when button A is pressed
  set speed to 30
  move forward at power speed % for 1 secs
  
```

Codey Rocky, standing on the stage, is saying hello to the audience. Codey Rocky turns left 70 degrees and then turns right 140 degrees. Finally, it returns to the starting point. To make Codey Rocky turn specific degrees, you need to create a variable named "angle". Set the angle to the value of 70.

```

set angle to 70
  
```

Make Codey Rocky turn left by specific degrees as programmed.

```

turn left angle degrees until done
  
```

Assign a new value to the variable "angle". As programmed, Codey Rocky will turn right at 140 degrees and then turn left at 70 degrees. Add facial expressions, sounds and light effects to Codey Rocky.

Task 2: Rock-Paper-Scissors

Using Codey Rocky, you can play the game rock-paper-scissors with friends. When Codey Rocky starts up, the initial values are all set to be 0. Codey Rocky changes its hand gestures randomly. The numbers 0, 1 and 2 are used to represent Rock, Scissors and Paper respectively. When Codey is shaken, the variable "gesture" will be one of the three numbers randomly. If the value is 0, the LED screen will show an image of a fist. If the value is 1, the LED screen will show an image of scissors. If the value is 2, the LED screen will show an image of the paper.

```

if gesture = 0 then
  show image [fist]
  
```

If Codey wins the game, you need to press button A and the variable "win" will change its value by 1. Meanwhile, Codey shows a smiley face and plays the sound "laugh". If Codey loses the game, you need to press button B and the variable "lose" will change its value by 1. Meanwhile, Codey shows a sad face and plays the sound "sad". If it is a draw, you need to press button C and the variable "draw" will change its value by 1. Meanwhile, Codey looks calm and gives you a humming sound.

In some cases, Codey might steal a glance at its chance of winning. When the light intensity is lower than 2, the chance of winning will appear on the LED screen in the form of decimals.

```

when light intensity < 2
  show win / win + lose + draw until done
  
```





Notes:

When coding Codey Rocky, you are supposed to use no more than six Events blocks.

To get the chance of winning, you'll need to add multiple Operators blocks to the code.

Formula: $\text{Winning times} / (\text{winning times} + \text{losing times} + \text{draw times}) = \text{Chance of winning}$

When button A is pressed, Codey transmits an infrared signal; if another Codey receives the signal, its health points go down by 1; the game ends when the health points fall to 0.

The ears of the Codey are preset with IR receivers and IR transmitters, which facilitates the wireless communication between two Codeys.

IR Transmitter

Facilitate the communications between Codeys



IR Receiver

Receive the signals from the IR transmitter

When Codey starts up, the initial value of the health point is 10. The next step is to use the forever block to keep detecting whether Codey gets "bullets" from its "enemy". If Codey gets shot, the health point change by "-1" and plays the sound "surprised". The game will be over when the health points fall to 0 and Codey will look sad.

If Codey still has health points left (the value is greater than 0) when button A is pressed, it will then send an IR message and play the sound "laser".

Conclusions

In computer programming, a variable is a container that can store information. You can duplicate the variable, change the information stored in the variable and make comparisons between variables. With the Operators blocks, you can use variables to store the calculation results.

Think!



What do I know now?



What did I learn?



What do I want to know?

Once Codey A receives the message from Codey B, Codey A gets shot and its health points reduce by 1. You can use the if statement block to assess whether Codey receives an IR message (gets shot). Once Codey gets shot, it will make a surprised sound and the health points will fall by 1.