AI for Children

Artificial Intelligence Curriculum for Elementary and Secondary Schools

Card

Reinforcement Learning

04 Machine Learning Deck

https://kurikulum.aidetem.cz/cards

Created by: Eva Nečasová & Radek Špáta
Expert guarantors: Tomáš Mlynář, Pavel Kordík
Methodological consultants: Peťa Dovhunová
Language proofreading: not yet done
Last update: 01/2025
Version: 04

Form for comments

npi National Pedagogical Institute of the Czech Republic

We create methodologies in cooperation with the National Pedagogical Institute.

Teaching Material for the AI Curriculum for Elementary and Secondary Schools
Computer Science at Secondary Schools - Cards, Machine Learning Deck

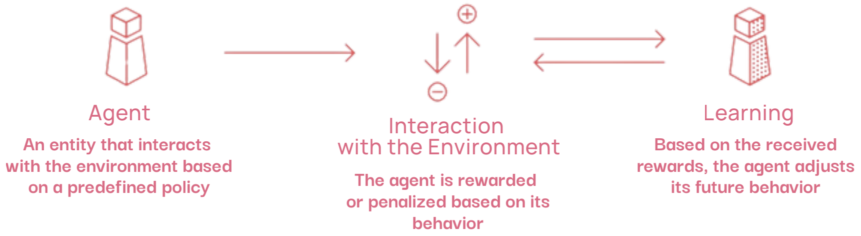
Reinforcement Learning

Basic information about reinforcement learning

Reinforcement learning is a method that helps a program learn to make the right decisions through trial and error. It tries out different strategies and figures out which ones lead to the best results. You can imagine it like a game where the program earns points for good decisions and loses points for bad ones. The goal is to collect as many points as possible. In this way, the program learns how to achieve the best outcome. Take a look at this little analogy with a chicken. The chicken represents the program and gets a reward only when it pecks at the pink dot. So naturally, it won't even think about other colors.

Advanced definition

Reinforcement learning (sometimes called feedback-based learning) is a type of machine learning in which an agent (an AI system) learns to take actions in a given environment in order to maximize a specific reward. Learning happens through interaction with the environment – this can be a real-world setting, such as a robot learning to walk, or a simulated one, like a card game, chess, cryptocurrency trading, and so on. Each action results in either positive or negative feedback. Through repeated feedback on its behavior, the agent gradually discovers optimal strategies – without being explicitly told what to do, as would be the case with a traditional algorithm.



Agent
An entity that interacts with the environment based on a predefined policy

Interaction with the Environment
The agent is rewarded or penalized based on its behavior

Learning
Based on the received rewards, the agent adjusts its future behavior

Machine Learning Deck

Dataset
Supervised Learning
Bias
Reinforcement Learning

Lesson presentation in PDF

Editable presentation in Canva

Note 1: Gender equality is a key value for AI for Children, but to keep our teaching materials concise, we use masculine grammatical forms.

Classroom Activity

How computers learn to play games

30–45 min

Activity description

Students are introduced to the concept of reinforcement learning through an interactive game called Hexapawn. Hexapawn is a simplified chess game played on a 3×3 board, with three pawns on each side. The goal of the game is to either move a pawn to the opposite side of the board or eliminate all of the opponent's pawns.

Students not only play against a computer that uses reinforcement learning to improve its strategy after each game, but also actively work with a game tree. The game tree shows all possible moves in Hexapawn and helps students visualize and understand the decision-making process of a machine learning algorithm. Each node in the tree represents a specific game state, and students add the possible next moves to see how different decisions can affect the outcome. This visual tool illustrates how AI learns from previous games and uses experience to refine its strategies and improve over time.

How reinforcement learning connects to the activity

This activity directly demonstrates the principles of reinforcement learning, where the computer (AI agent) interacts with an environment (the Hexapawn game) and learns from the outcomes of its actions. As the game progresses, the AI receives feedback in the form of wins and losses, which help it determine which moves lead to success and which do not. Through this process of continuous improvement based on experience and rewards (reinforcement learning), the AI adapts and optimizes its strategies without being given exact instructions on what to do. Students can observe how the AI uses trial and error to improve its decision-making in real time.

Lesson Overview

Recommended Age, Lesson Length
Children aged 11–15, 45 minutes.

Building Blocks
Reinforcement Learning.

What Are the Students Learning?
Reinforcement learning is a type of machine learning where machines learn through trial and error.

Why Are They Learning This?
To develop analytical thinking and understand decision-making processes in games.

How Do We Know They Have Learned It?
They'll be able to explain how computers use machine learning algorithms to adjust their strategies.

Tools
Teacher: projector, presentation.
Students: computer, laptop, or tablet (except iOS or Android tablets) – one per group, plus worksheets.

Digital Competence
Communication and Collaboration.

Bloom's Taxonomy
Understanding: Students understand the decision-making processes of a machine learning algorithm in a simple game.
Applying: They apply their understanding of the algorithm while playing the game.
Analyzing: They analyze game strategies.

Five Big Ideas
3-A-VI Nature of Learning (Learning from experience).

Engage

10 min

Describe

Presentation slide 02

Discuss with your students – you can use some of these questions:
People learn in different ways. What methods have you used? Can you give an example?
What happens when you make a mistake and get immediate feedback? How does that help you learn?
Do you think you learn better when someone shows you what to do, or when you try it yourself? Why?

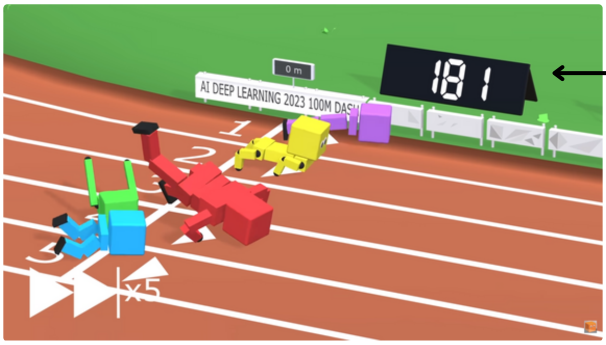
Guide students toward this generalization through discussion:
1) People can learn by example – if we see an orange once, we'll recognize it next time.
2) People also learn through trial and error – if we burn ourselves on a hot stove, we'll avoid touching it again when it's on.

In previous lessons, we saw that machines (like computers or robots) can learn from examples. We used the Teachable Machine program to show it examples of different alien families, and it learned to recognize them. Do you think machines can also learn from experience?
Correct answer: Yes, they can.

YouTube

Presentation slide 03

Play a part of the video for your students.
Link: youtube.com/watch?v=pJPdW8WWAso
Video explanation: The video shows five AI agents (3D robot-like figures) whose task is to learn how to run a 100-meter race. The first one to finish wins a cupcake. The agents learn through trial and error, and their movement decisions are controlled by a neural network (like the one we used in Teachable Machine). The program rewards agents the closer they get to the finish line and penalizes them if they fall or fail to stay upright. Over many training runs, the agents learn how to reach the goal. In doing so, they gradually develop better running strategies – they learn how to "run smarter."



Number of attempts

The video is 11 minutes long – feel free to show just a few short clips. For example:
Beginning: Explain what the video is about, what the goal is, and show the number of attempts.
Attempt 204: The green robot takes its first steps. You can have students guess which agent will finish first.
Attempt 390: The red one is moving but unsure what to do, and the purple one decides to cheat.
Attempt 738: The robots are improving but still struggling with balance.
Attempt 813: The purple one is starting to get good!
Attempt 954: The red one almost manages to stay upright… but then…
Attempt 1638: The red one runs and we watch with suspense as it approaches the finish line… And it makes it!

During the evocation phase, do not explain that the agents are learning through reinforcement learning. We'll return to the video during the reflection.

Understand

20 min

5 min

Activity 1

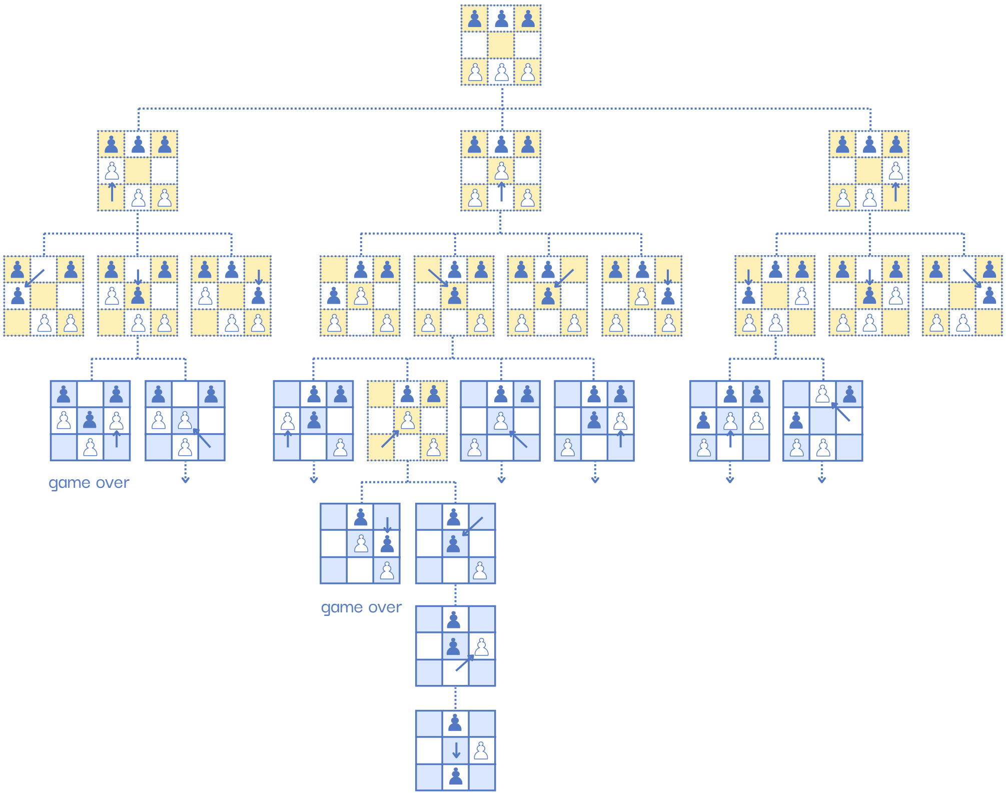
Worksheet 1

Presentation slide 04

Students work independently or in pairs using worksheets. Show them the game rules for Hexapawn and the visual from slide 04 of the presentation.

Hexapawn is a simplified version of chess. It's played on a 3×3 board, and each player starts with three pawns. The goal is to get one of your pawns to the opposite side of the board. Pawns move like in chess: they can move one space forward if it's empty, or capture an opponent's pawn by moving diagonally forward. The player who either reaches the opposite side first or captures all opponent pawns wins the game.

Students' task is to complete the game states shown in blue below. Correct solutions look like this:



15 min

Activity 2

Presentation slide 05

Students will teach the program to play Hexapawn. Share the QR code from slide 05 of the presentation. Link: mrozilla.cz/lab/hexapawn

Developer Mrozilla offers a program on their website that demonstrates the principle of reinforcement learning using the game Hexapawn.
At first, the program (AI) makes random moves in response to the player. These moves are completely random at the beginning, but whenever a move leads to a loss, the program stops using it (unless the player refreshes the browser page). Thanks to reinforcement learning, the program quickly becomes unbeatable.

Let students play Hexapawn (it works on smartphones too). They can also record their moves on pre-printed game tree sheets. These sheets are included in Worksheet 2 on page 07 of the teaching material.

Hexapawn

Reflect

10 min

Recall & analyze

Presentation slide 06

Today we worked with a game tree. It helped us visualize the course of the Hexapawn game, which is much simpler than chess. Now try to imagine how many possible game states there are in chess. It's estimated to be somewhe^{50}re between 10^{43} and 10^{50}. The Go game is even more complex.

What did you learn today about how machines can learn from experiences?
Machines can learn in a similar way to us – through trial and error. When they make decisions in an environment, they receive feedback – either a reward for a good decision or a penalty for a mistake. That helps them "remember" what works and improve over time. We saw this in Hexapawn, where after just a few games, the program stopped making mistakes and learned to play optimally. This process is called reinforcement learning, and it shows how machines can improve their strategies through experience.

What strategies did you use during the game? How were they similar to or different from the computer's strategies?

Where do you think reinforcement learning is used in real life?
Possible answers: In other games like chess, Go, or video games. In real life, reinforcement learning helps robots or self-driving cars learn how to behave effectively.

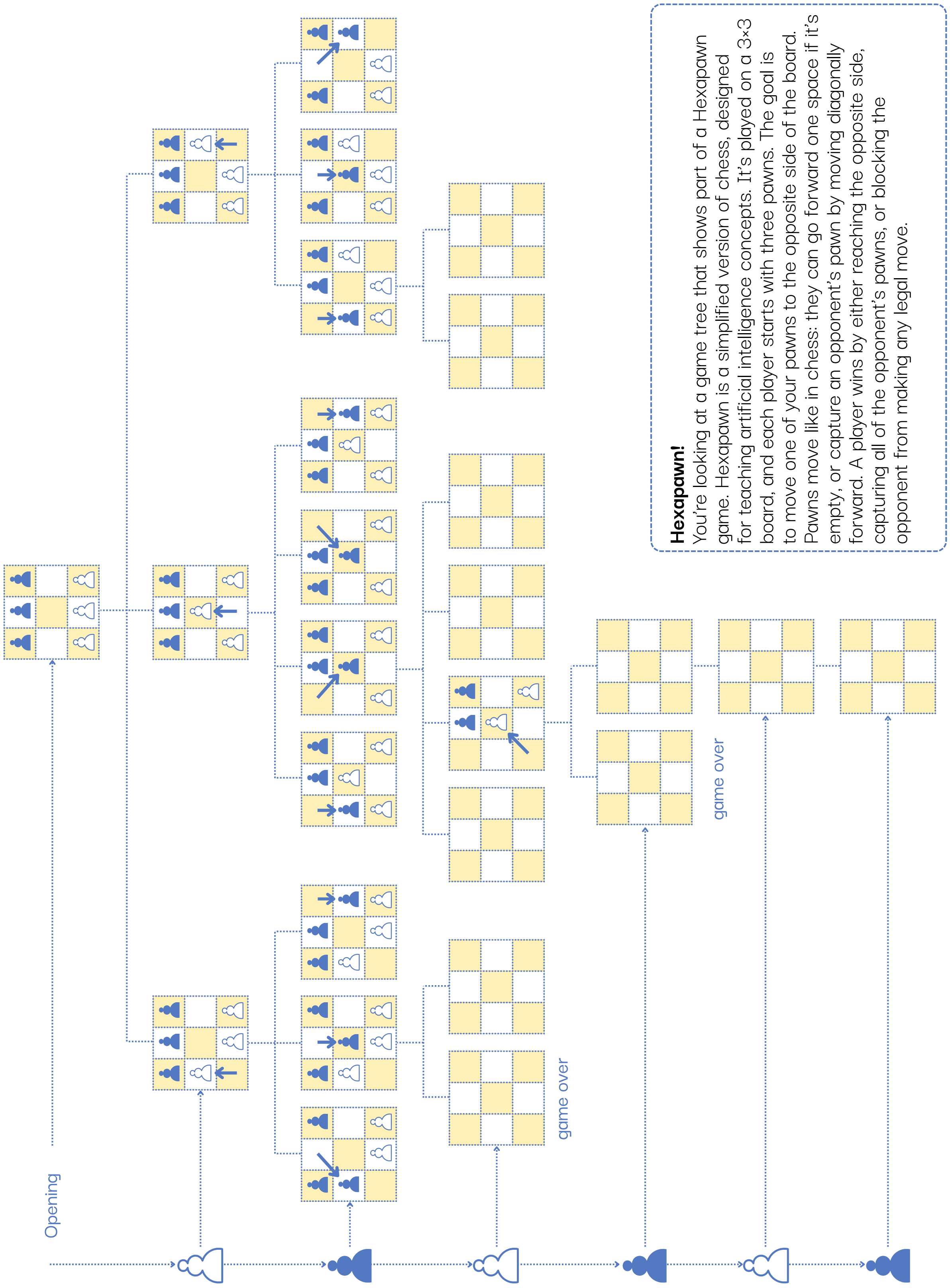
You can show students a few examples:
Agent57 learned to play all Atari console games using reinforcement learning: youtu.be/TmPfTpjtdgg
A four-legged robot learns to stand up: youtube.com/watch?v=xAXvfVTgqr0
A simulation of a self-driving car: youtube.com/watch?v=SX08NT55YhA

All programs that are able to learn use machine learning – the method that allows artificial intelligence to learn. But reinforcement learning is not the only type of machine learning. In earlier lessons, we used a different kind with Teachable Machine. Do you remember what it was called?
Correct answer: Supervised learning.

We humans learn from both examples and experience. So, can machines do the same?
Correct answer: Yes. When machines learn from examples, it's called supervised learning. When they learn from experience, it's called reinforcement learning.

Some decisions lead to more possible moves, while others do not.

Fill in the empty spaces of the game tree with the next possible Hexapawn game states.



Hexapawn!
You're looking at a game tree that shows part of a Hexapawn game. Hexapawn is a simplified version of chess, designed for teaching artificial intelligence concepts. It's played on a 3×3 board, and each player starts with three pawns. The goal is to move one of your pawns to the opposite side of the board. Pawns move like in chess: they can go forward one space if it's empty, or capture an opponent's pawn by moving diagonally forward. A player wins by either reaching the opposite side, capturing all of the opponent's pawns, or blocking the opponent from making any legal move.

Record all the moves and watch whether the program repeats its mistakes.