EURECOM
Sophia Antipolis

S6 semester project

Cti
Commission des Titres d'Ingénieur

Agenda

- Intro, teaching goals, organization (M. Todisco)
- The technical project (P. Lisena and M. Todisco)
- Introduction to Machine learning (H. Chaptoukaev)
- Introduction to Artificial intelligence for gameplay (P. Lisena)
- The target platform, development tools and methods (P. Lisena and M. Todisco)
- Expectations and evaluation (M. Todisco)
- Groups forming (All)

Intro

- **First of all**
  - a great recognition goes to all of you for your work on the S5 project
- **You have learnt that the semester project is important**
  - it is a key element of your curriculum
  - it is something that you will probably use in your resume
  - you have learnt a lot (technical and non-technical)
- **..and now S6 project has begun!**
  - it accounts for 5 ECTS
  - equal… indeed, greater… results are expected

Teaching goals

- **Technical**
  - put several theoretical knowledge and skills in practice
  - programming, development tools
  - image and audio processing
  - embedded systems
  - machine learning and artificial intelligence
  - communication protocols
- **Less or non-technical**
  - teamwork
  - searching and finding documentation
  - documenting and presenting

No obligation to use specific programming languages or development tools

Less dependence, more autonomy!

Organisation

- Full semester, 37 three hours sessions, 2 or 3 / week
- Dedicated working rooms
- Eight 5-students groups (2 types: groups A and B)
- Same goal for all groups
- Group tutors
- Technical experts
- Project management experts
- One full day at the end for presentations and evaluation

Group tutors

- One professor per group
- Regular meetings (see with your tutor)
- Check progress, detect issues, give recommendations…
- Participate the final evaluation and grading

Technical experts

- **Do not hesitate to ask them for help or advices**
  - but of course, request appointment first
- **Experts**
  - raspberry Pi board
  - coding and software tools
  - image and video processing
  - audio processing
  - machine and deep learning

Schedule

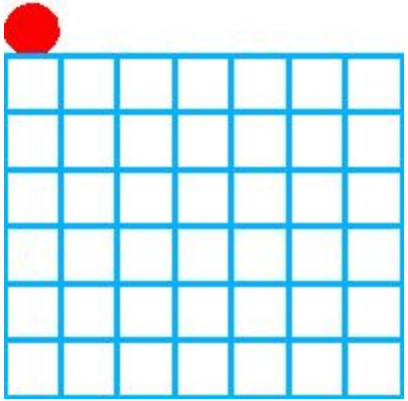
- **Technical helper sessions**
  - today, 30m: H. Chaptoukaev, introduction to machine learning
  - today, 30m: P. Lisena, introduction to artificial intelligence for gameplay
- **Project management helper sessions**
  - 01/03 (A groups) and 02/03 (B groups), 55mn/group: coaching
  - 29-30/03, 11-12/05, 06-08/06: coaching
- **All other slots: work, work, work, try, fix, work…**

The technical project

P. Lisena and M. Todisco

The technical project

Remote gaming Connect 4 on Raspberry Pi using gesture recognition and AI gameplay

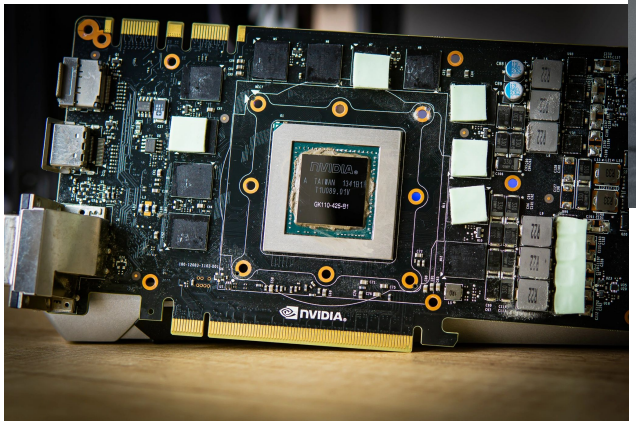


Connect 4

- Connect 4 is a two-player strategy game in which players take turns dropping colored discs from the top into a vertical board of 7 columns across and 6 rows.
- The objective of the game is to connect 4 of one's own discs of the same color next to each other vertically, horizontally, or diagonally before your opponent does.
- The game ends when one player successfully gets four in a row or when the game board is full and no more moves can be made (draw).

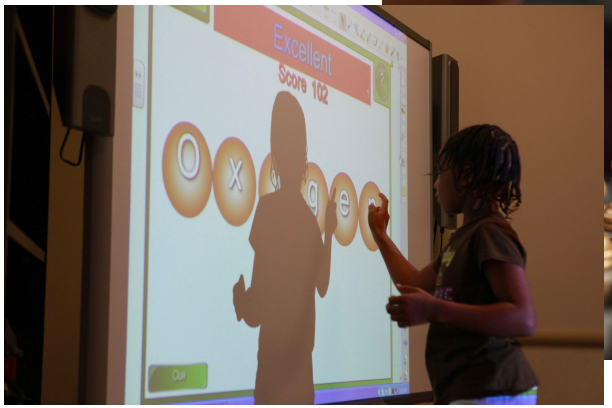
Why a game?

- Gaming industry leads innovation



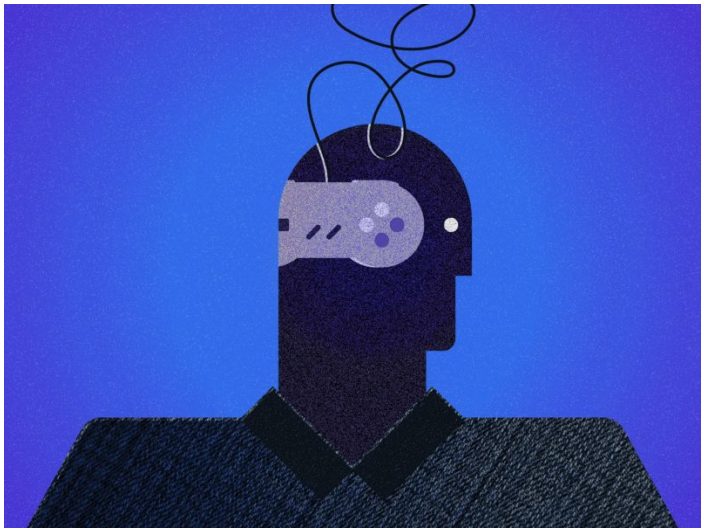
Why a game?

- Gaming industry leads innovation
- Games can be used for education and training



Why a game?

- Gaming industry leads innovation
- Games can be used for education and training
- Games stimulate creativity



Pillars of the project

- **Pillar 1 - Game interface development (provided and developed in Python)**
  - improvement made to the interface (design or sound) is a plus
- **Pillar 2 - Artificial intelligence for the gameplay**
  - decision tree: e.g. minmax
  - others of own choice
- **Pillar 3 - Hands gesture recognition to control the game**
  - DNN-based algorithms: e.g. CNNs
  - others of own choice
- **Pillar 4 - Communication protocol between Raspberry Pi for one-to-one play**
  - transportation layer protocol: e.g. TCP
  - others of own choice

Computer Programming (C. Galdi)
Sound and Music Processing (M. Todisco)

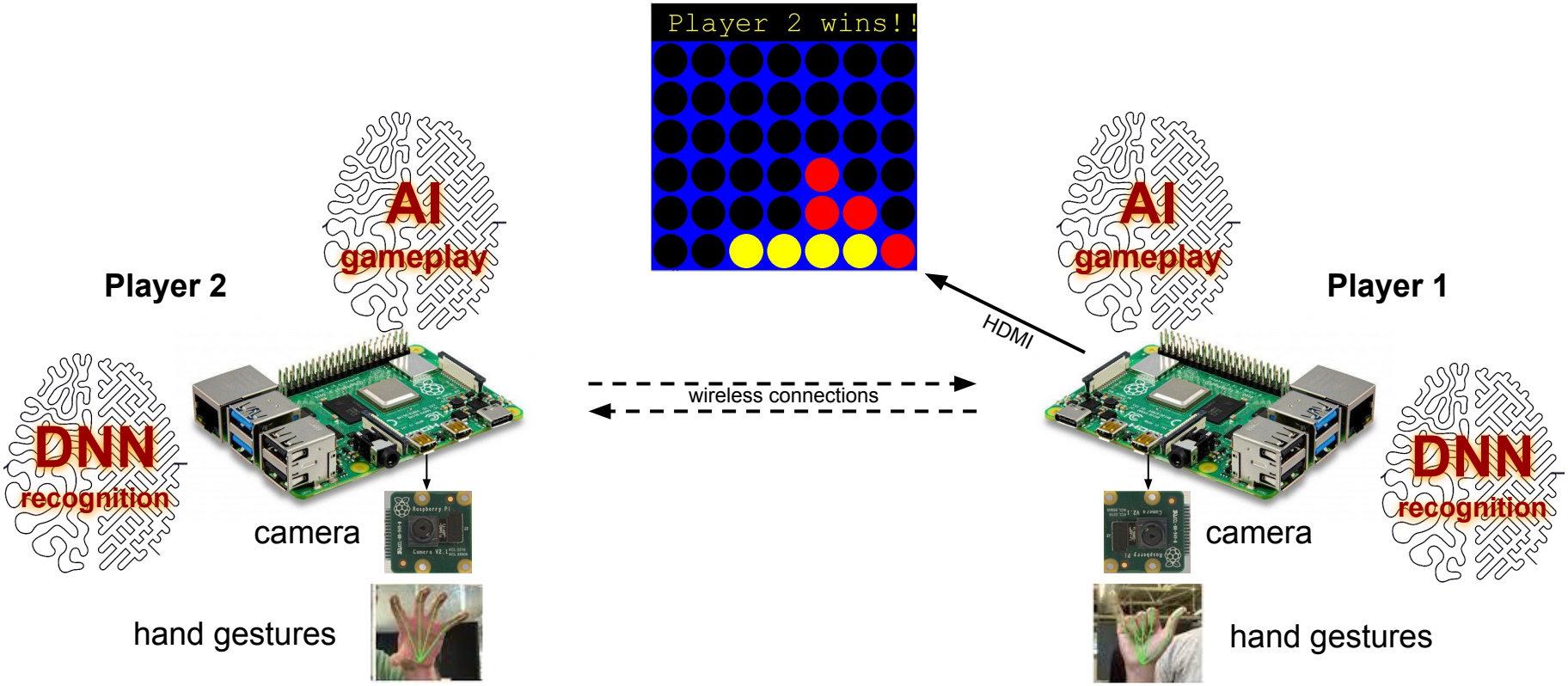
Machine Learning and Intelligent Systems (M. Zuluaga)

Introduction to Computer Networking and the Internet 1 & 2 (A. Ksentini)

Game modalities

- **Playing Connect 4 through two Raspberry Pi boards that are wirelessly connected, allowing:**
  - two players to compete against each other on two different devices
  - one player competing against AI
  - a scenario of *the battle of artificial intelligences*, AI against AI
- **Human players must use hand gestures instead of physical gaming devices**

Design workflow diagram



Group work plan

- **Each group must nominate a pillar responsible**
- **Pillar 4 leaders of each group must agree to have a common communication protocol**
  - have regular meetings where ideas can be proposed and discussed
  - the final protocol must be defined at the end of the first third of the total number of working sessions, i.e. within about 12 sessions
  - enhancing inter-group communication skills
- **Pillar 2 and 3 developments belong to each group and are private**
  - keep the secret!
- **Pillar 1 must keep its core untouched**
  - permitted improvements concern the interface design or sound

Introduction to machine learning

H. Chaptoukaev

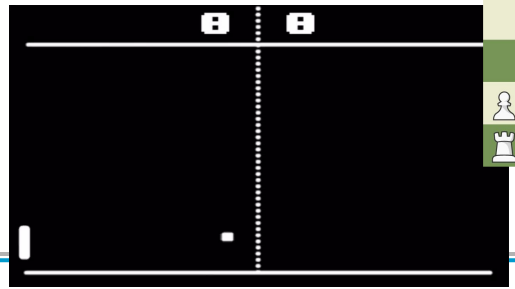
Introduction to artificial intelligence for gameplay

P. Lisena

AI in gameplay

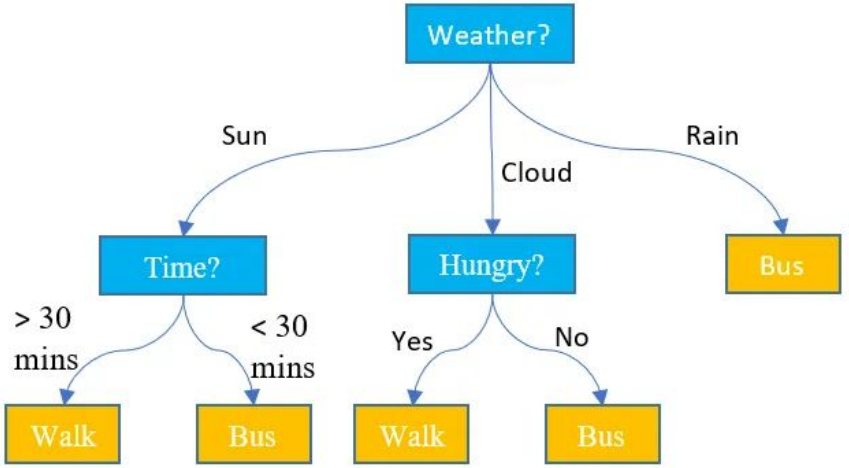
- Control of NPCs
- Control of objects around
- Oppose the player

- Real-time
- Turn games
  - The case of Connect4

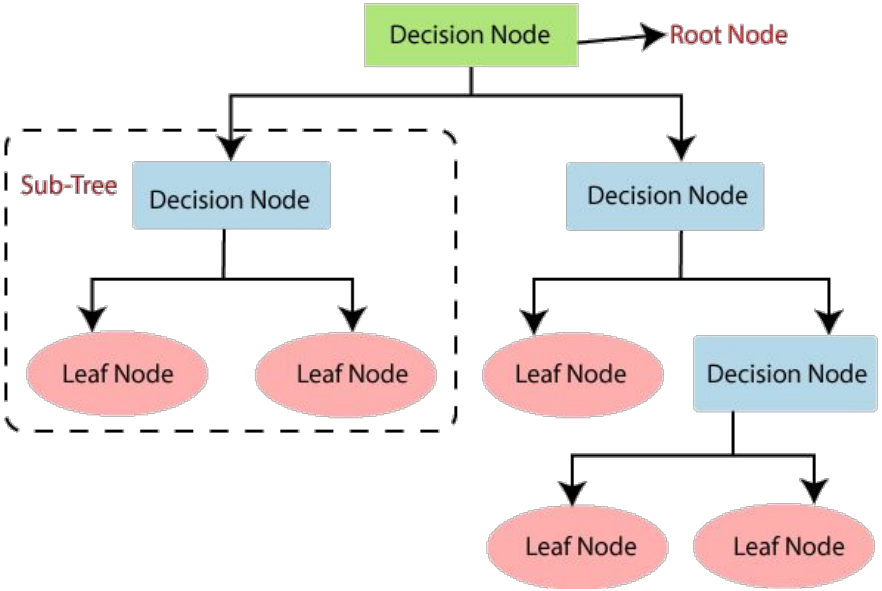


Decision Tree

- Tree structure for taking a decision
- At every step, I go down in the structure
  - Until I arrive to a terminal node
- Used in business, maths, …
- Decision trees are at the base of some Machine Learning algorithms (e.g. Random Forests)

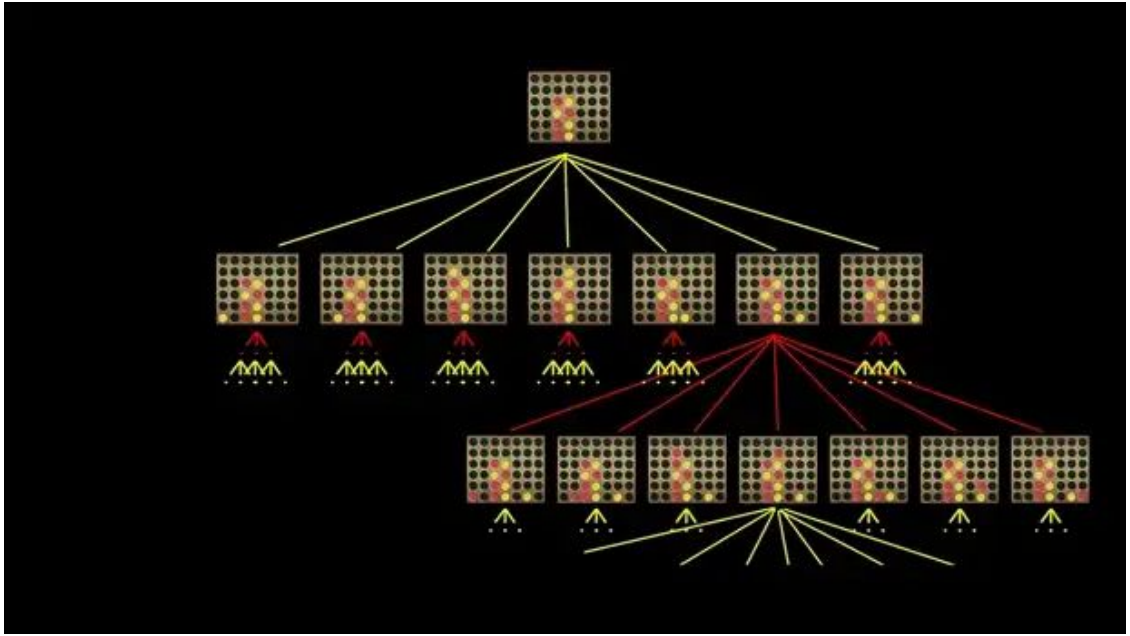


Decision Tree



- I start at the **root**
- At every **decision node**
  - A sub-tree is selected
- When a **leaf node** is reached
  - The algorithm ends and give me the solution
- Open points:
  - Which decision to make before?
  - Can I stop earlier? (costs)

Decision Tree in Connect 4

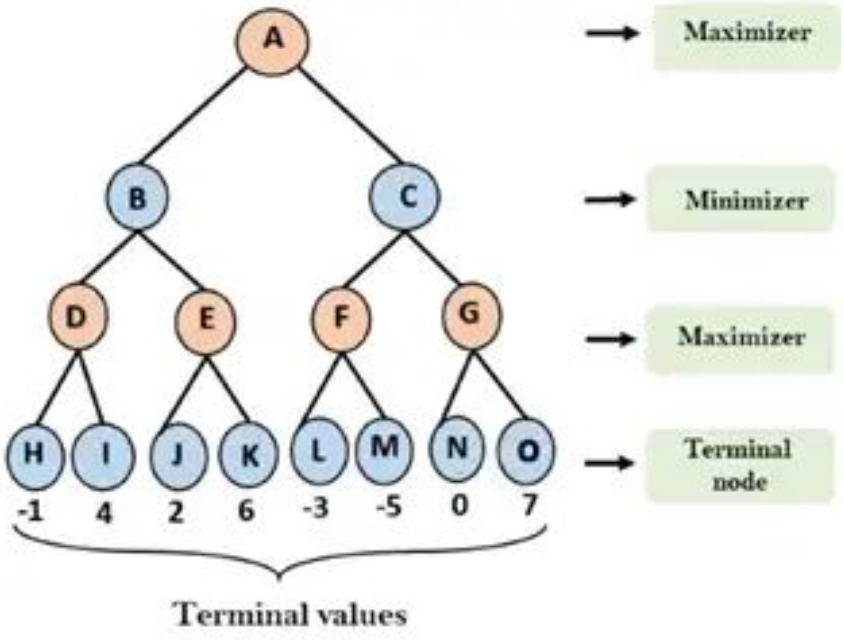


7 columns = 7 branches of a tree for each iteration

Minimax algorithm

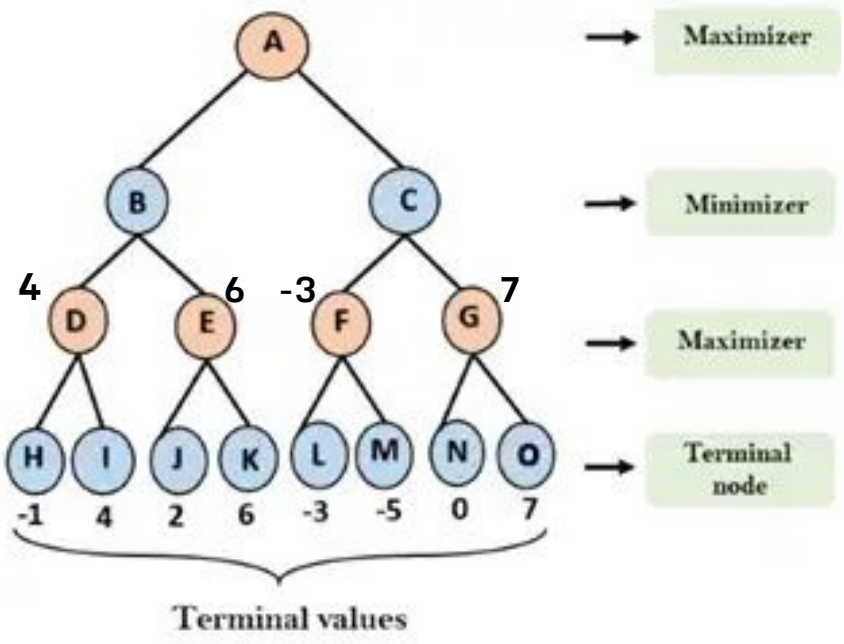
- AI algorithm exploiting Decision Trees
  - Used especially in gaming
- It simulates the **presence of 2 players**:
  - Player 1 (**Min**) has to reach -∞ (minimize the score)
  - Player 2 (**Max**) has to reach +∞ (maximise the score)
  - 2 goals in contrast = they are playing one against each other
- The algorithm decide which move to make considering that the two players in each successive turn will make the **best move for their own goal**

Minimax algorithm



- Initialisation: the full tree is generated and the terminal values are set
- The red nodes are the turns of Max, the blue are the turns of Min
- Max and Min start at their worst score
  - -∞ for Max and +∞ for Min
- Starting from the end, we compute the best possible scores for each node (next slide)

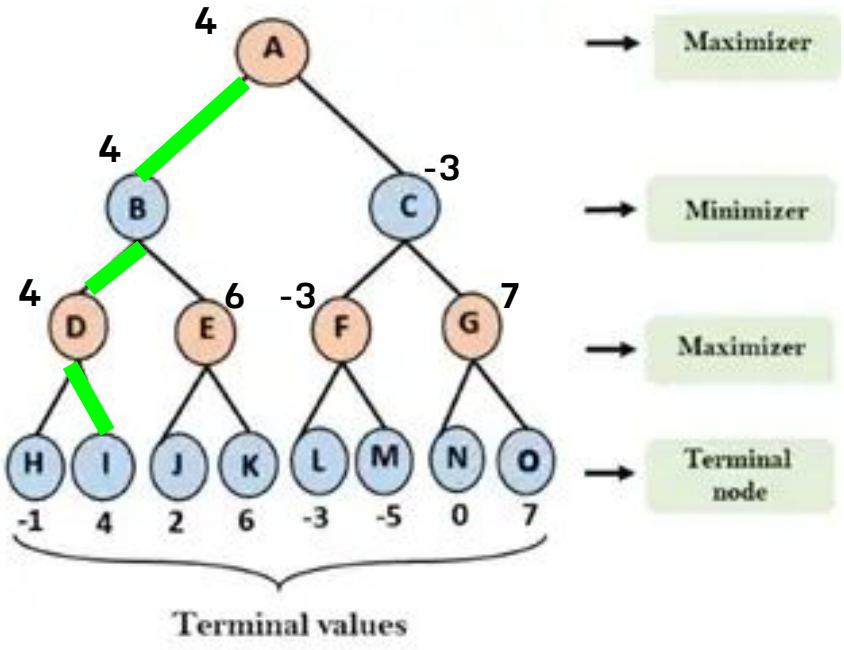
Minimax algorithm



Red nodes: Maximizer

- For node D
  - max(-1, -∞) → max(-1,4) = 4
- For node E
  - max(2, -∞) → max(2, 6) = 6
- For node F
  - max(-3, -∞) → max(-3,-5) = -3
- For node G
  - max(0, -∞) → max(0, 7) = 7

Minimax algorithm



Blue nodes: Minimizer

- For node B
  - min(4, +∞) → min(4,6) = 4
- For node C
  - min(-3, +∞) → min(-3, 7) = -3

Root red node: Maximizer

- For node A
  - max(4, -3) = 4

The path to choose is the 4!

(A → B → D → I)

Minimax algorithm - alpha-beta pruning

- Strategy to improve computational speed and memory
- Instead of re-computing the whole tree, I set 3 values
  - *depth*: at which level of the tree I should arrive
  - α: initialised at -∞ and updated by the maximiser with the higher score
  - β: initialised at +∞ and updated by the minimiser with the lower score
- The computation stops
  - if *depth* is reached, or
  - if $\alpha >= \beta$ (it is already better than what the other player can do!)

Minimax algorithm - set the terminal values

How to choose the terminal values?

- Winning move for AI (Max) → +100000000
- Winning move for Player (Min) → -100000000
- Draw move (no other possible moves) → 0
- Other cases → compute a score
  - In *Connect 4*, I count how many connection are on the board
  - E.g. connect by 3 = 5 points, connect by 2 = 2 points, …
  - The opponents' connections are counted negatively

Minimax algorithm in Connect 4

- The decision tree has as 7 possibilities at each level
  - Reduced while playing if the column is complete
- Take note of the board status
  - For each cell, it is occupied? By which player?
- Check every iteration if the game is finished
  - Player 1 wins, Player 2 wins or Draw
- An implementation in Python is available at https://github.com/KeithGalli/Connect4-Python/

What else?

- Minimax is just a suggestion, but you can
  - Modify, improve, experiment with it
  - Replace it with another algorithm
- There is plenty of possibilities
  - In 1988, Victor Allis realised a 90-pages master thesis on Connect Four AI
  - http://www.informatik.uni-trier.de/~fernau/DSL0607/Masterthesis-Viergewinnt.pdf

A Knowledge-based Approach of Connect-Four

The Game is Solved: White Wins

Victor Allis

Department of Mathematics and Computer Science
Vrije Universiteit
Amsterdam, The Netherlands
Masters Thesis, October 1988 †

ABSTRACT

A Shannon C-type strategy program, VICTOR, is written for Connect-Four, based on nine strategic rules. Each of these rules is proven to be correct, implying that conclusions made by VICTOR are correct.

Using VICTOR, strategic rules where found which can be used by Black to at least draw the game, on each $7 \times (2n)$ board, provided that White does not start at the middle column, as well as on any $6 \times (2n)$ board.

In combination with conspiracy-number search, search tables and depth-first search, VICTOR was able to show that White can win on the standard 7×6 board. Using a database of approximately half a million positions, VICTOR can play real time against opponents on the 7×6 board, always winning with White.

Credits

Jonathan C.T. Kuo, *Artificial Intelligence at Play — Connect Four (Mini-max algorithm explained)*
https://medium.com/analytics-vidhya/artificial-intelligence-at-play-connect-four-minimax-algorithm-explained-3b5fc32e4a4f

Nikita Sharma, *Understanding the Mathematics Behind Decision Trees*

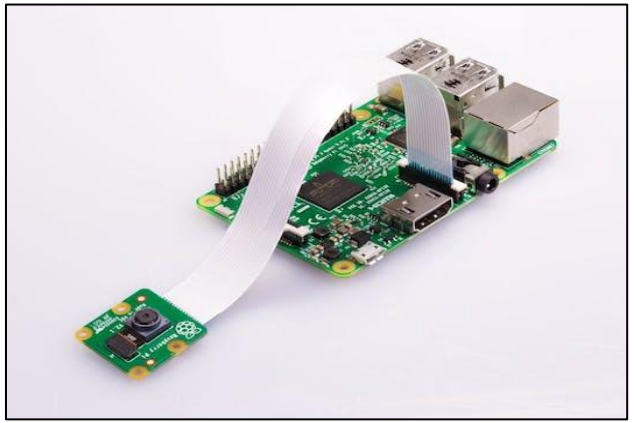
https://heartbeat.comet.ml/understanding-the-mathematics-behind-decision-trees-22d86d55906

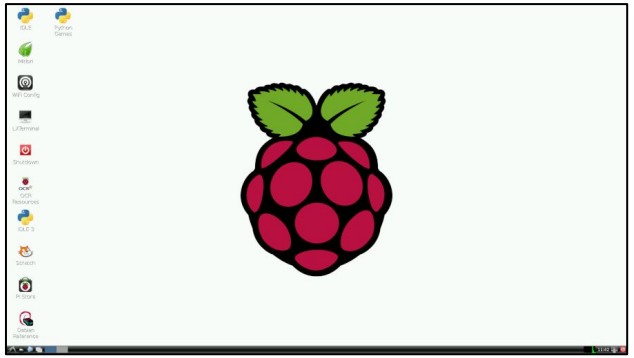
The target platform, development tools and methods

P. Lisena and M. Todisco

The target platform

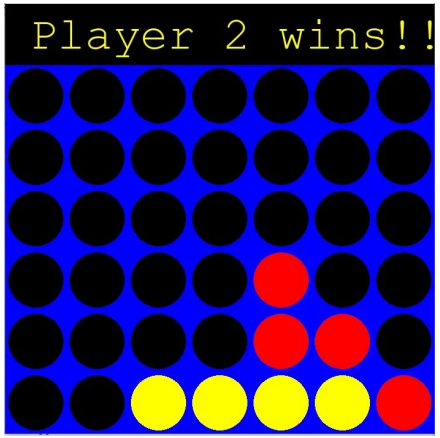
- **Raspberry Pi 4 model B 2GB RAM**
- **Raspberry Pi Camera Module 2**
- **PiJuice HAT (optional)**
- **Headphones or loudspeakers**
- **Monitor with HDMI port**
- **Raspbian OS**
  https://www.raspberrypi.com/software/
  - a Debian-based OS for Raspberry perfect general-purpose OS for Raspberry users
  - backup your Raspberry Pi SD Card with S5 project before installing it





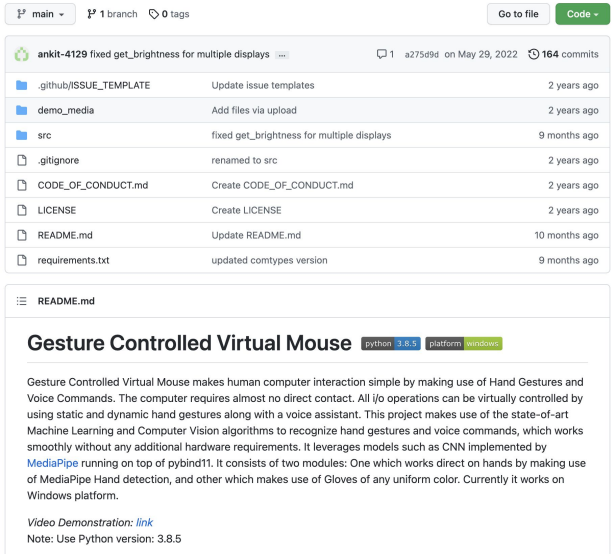
Connect 4 game interface

- **Connect 4 game interface**
  - https://www.askpython.com/python/examples/connect-four-game
- **Pygame tutorial (creating buttons, etc.)**
  - https://www.askpython.com/python/examples/pygame-graphical-hi-lo-game



Hands gesture recognition

- **Controls with gesture recognition**
  - https://github.com/xenon-19/Gesture-Controlled-Virtual-Mouse



main · 1 branch · 0 tags · Go to file · Code

ankit-4129 fixed get_brightness for multiple displays · 1 · a275d9d on May 29, 2022 · 164 commits

.github/ISSUE_TEMPLATE	Update issue templates	2 years ago
demo_media	Add files via upload	2 years ago
src	fixed get_brightness for multiple displays	9 months ago
.gitignore	renamed to src	2 years ago
CODE_OF_CONDUCT.md	Create CODE_OF_CONDUCT.md	2 years ago
LICENSE	Create LICENSE	2 years ago
README.md	Update README.md	10 months ago
requirements.txt	updated comtypes version	9 months ago

README.md

Gesture Controlled Virtual Mouse

python 3.8.5 | platform windows

Gesture Controlled Virtual Mouse makes human computer interaction simple by making use of Hand Gestures and Voice Commands. The computer requires almost no direct contact. All i/o operations can be virtually controlled by using static and dynamic hand gestures along with a voice assistant. This project makes use of the state-of-art Machine Learning and Computer Vision algorithms to recognize hand gestures and voice commands, which works smoothly without any additional hardware requirements. It leverages models such as CNN implemented by MediaPipe running on top of pybind11. It consists of two modules: One which works direct on hands by making use of MediaPipe Hand detection, and other which makes use of Gloves of any uniform color. Currently it works on Windows platform.

Video Demonstration: link
Note: Use Python version: 3.8.5

Communication protocol

- **Communication protocol**
  - https://wiki.python.org/moin/TcpCommunication

client

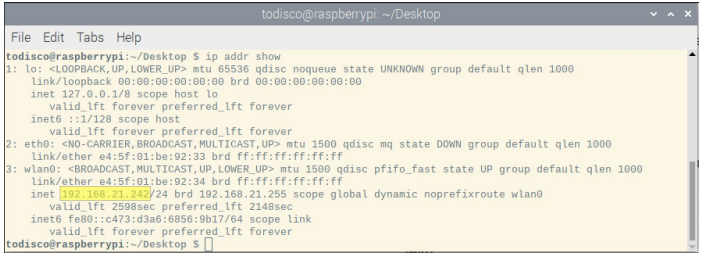
```
#!/usr/bin/env python

import socket


TCP_IP = '127.0.0.1'
TCP_PORT = 5005
BUFFER_SIZE = 1024
MESSAGE = "Hello, World!"

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((TCP_IP, TCP_PORT))
s.send(MESSAGE)
data = s.recv(BUFFER_SIZE)
s.close()

print "received data:", data
```



server

```
#!/usr/bin/env python

import socket


TCP_IP = '127.0.0.1'
TCP_PORT = 5005
BUFFER_SIZE = 20  # Normally 1024, but we want fast response

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((TCP_IP, TCP_PORT))
s.listen(1)

conn, addr = s.accept()
print 'Connection address:', addr
while 1:
    data = conn.recv(BUFFER_SIZE)
    if not data: break
    print "received data:", data
    conn.send(data)  # echo
conn.close()
```

← you can use eduroam/uranus network

A bad example not to follow

Expectations

- **You are expected to prove that you are able to:**
  - design a working prototype
  - design and code for an embedded system
  - design and develop software code (clean, efficient, tested, commented, documented…)
  - use professional development tools
  - efficiently work in a group and discuss with other groups
  - manage your project
  - acquire missing knowledge and skills
  - document and present your work in a professional way (poster, presentation, demonstration, report)

Evaluation

- **Same grade for all group members, same jury (3 professors/researchers) for all groups**
- **14/06: project day**
  - morning, 3h: poster session and presentations (40%)
  - afternoon, 3h: final competition and demonstrations (20%)
- **Exam week (end of June, TBC): group tutors evaluate technical work, source code, report and send evaluation form to jury (40%)**
- **Final competition**
  - inter-group mode: groups will play against each other in AI vs. AI: *the battle of artificial intelligences*
  - intra-group mode: groups will play in human vs. AI mode using hands gesture

Groups and tutors

- **Each group is mentored by a tutor**
- **Every 30 days, each group will send a check report (max 1 page) to the tutor on the work done to date and what they plan to do in the next period**
- **The tutor will validate the check report or ask for clarification**

Group	Tutor
1	Antonio Faonio
2	Simone Aonzo
3	Raphael Troncy
4	Nicholas Evans
5	Melek Onen
6	Maurizio Filippone
7	Paolo Papotti
8	Derya Malak

Experts

- 8 domains of expertise have been established
- One or more experts are assigned to each domain
- The role of the technical experts is to answer students' questions about the project. Questions can be asked by the students at any time.

Domain	Expert(s)
Raspberry Pi board	Aurélien Francillon and Romain Cayre
Coding and software tools (git, GitLab, make...)	Thibault Ehrhat
Image and video processing	Jean-Luc Dugelay
Audio processing	Massimiliano Todisco
Machine and deep learning	Maria Zuluaga and Francesco Galati

Constitution of groups

- **Based on responses to the survey on areas of interest**
  - will be given today
  - mercato: you can negotiate with other groups and change up to 2 members per group
- **Groups will be frozen tomorrow morning at 11am.**
- **Send us (todisco@eurecom.fr and lisena@eurecom.fr) your change requests no later than 10am tomorrow.**
- **At tomorrow's lab at 13:30**
  - you will be given the cameras
  - communicate who the pillar leaders are

Survey on areas of interest

- Enter your name and surname (real please)
- The choice must be orthogonal to the 4 pillars
- 10 minutes from now to complete it

https://forms.gle/Psk4vHsQAzJzYR1A7

Please order the following activities in order of preference *

	1 (more interested)	2	3	4 (less interested)
Pillar 1 - Connect 4 game interface development	◉	○	○	○
Pillar 2 - Artificial intelligence for the gameplay	○	◉	○	○
Pillar 3 - DNN-based hands gesture recognition to play the game	○	○	◉	○
Pillar 4 - Development of wirelessly communication protocols	○	○	○	◉