Davide Molinelli

Wizard Chessboard

Cognitive Robotics

WARNING: THIS IS A LIVE DEMO!

You can compare this presentation to the official presentation of a new Apple product…

… there is only a 50% chance that everything goes according to plan!

BUT DON’T PANIC

If something goes wrong, there should be fire doors somewhere in this room!

FIRST QUESTION : HAVE YOU DONE A SEMINAR OR A PROJECT?

This is a Pro-Seminar: it's obvious!!!

The reasons of this choice are simple:

- I absolutely need to present public works to overcome my shyness and learn to speak English
- I want to share with you my project experience and the knowledge acquired in this months

THE IDEA

Wizard Chessboard stems from the desire to create intelligent objects capable of recognizing vocal commands and executing consequent actions.

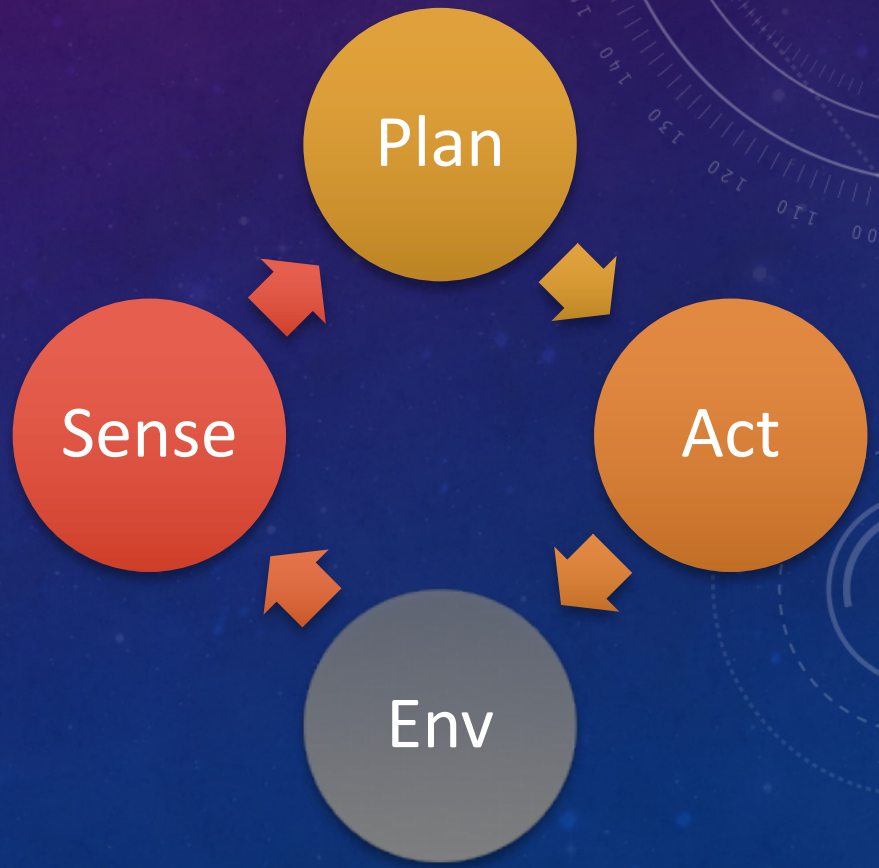
Yes, but why a chessboard?
Why not a coffeepot to which you command to make a good cappuccino?

Simply because I love Harry Potter and the evening when the inspiration was born I was looking at the first chapter of the saga, not a documentary on how to make a good cappuccino.

THE PARADIGM

The project is implemented through a deliberative approach, according to the sense-plan-act scheme:

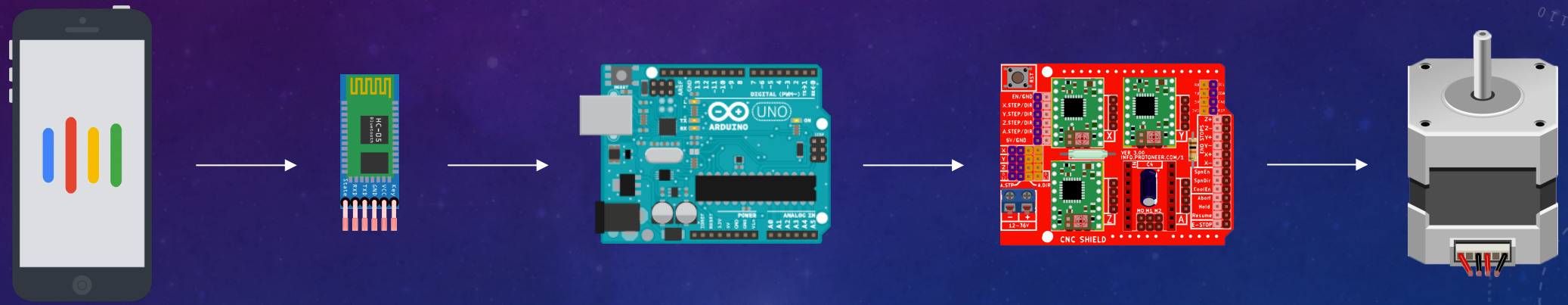
1. **Sense** - The system acquires a signal from the environment, through a wireless communication
2. **Perception** - The signal is processed in order to extract information, i.e. a voice command
3. **Model** – A representation of the chessboard is realized to have an internal model of the world
4. **Plan** – Compute the sequence of actions to be performed
5. **Act** – Execute the actions and update the model



HARDWARE COMPONENTS

- A homemade wooden chessboard
- Magnetized pieces
- Two stepper motors Nema 17
- An electromagnet controlled in voltage
- An Arduino Mega 2560 Rev3 microcontroller
- A CNC Shield V3
- An HC-05 Bluetooth module
- An Android smartphone

THE SYSTEM



THE VOICE COMMAND

The system is able to recognize two type of command:

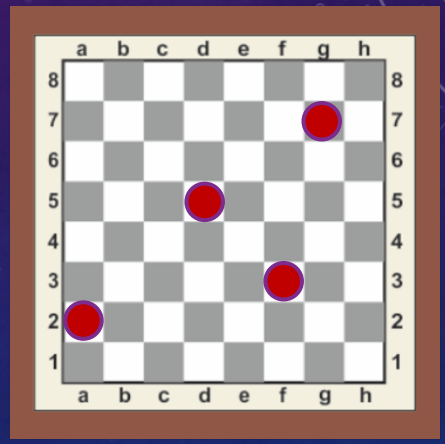
1. If there is no ambiguity and only a single piece can reach the destination cell you can simply express a generic command like the following:

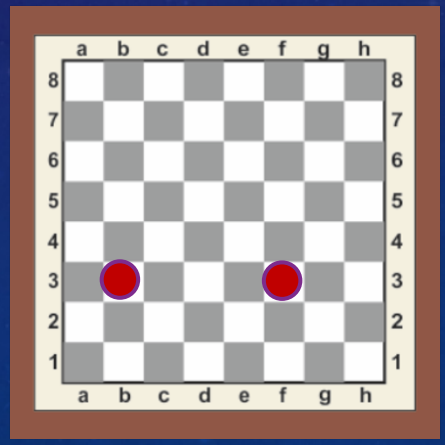
   PEDINA IN A3

   The system recognize which of the pawns is the only candidate and perform the move on it.

2. If there are more than one piece that can reach the destination cell it is necessary to express also the coordinates of the source cell to be moved. An example of this type of command is the follow:

   ALFIERE DA B3 A D5





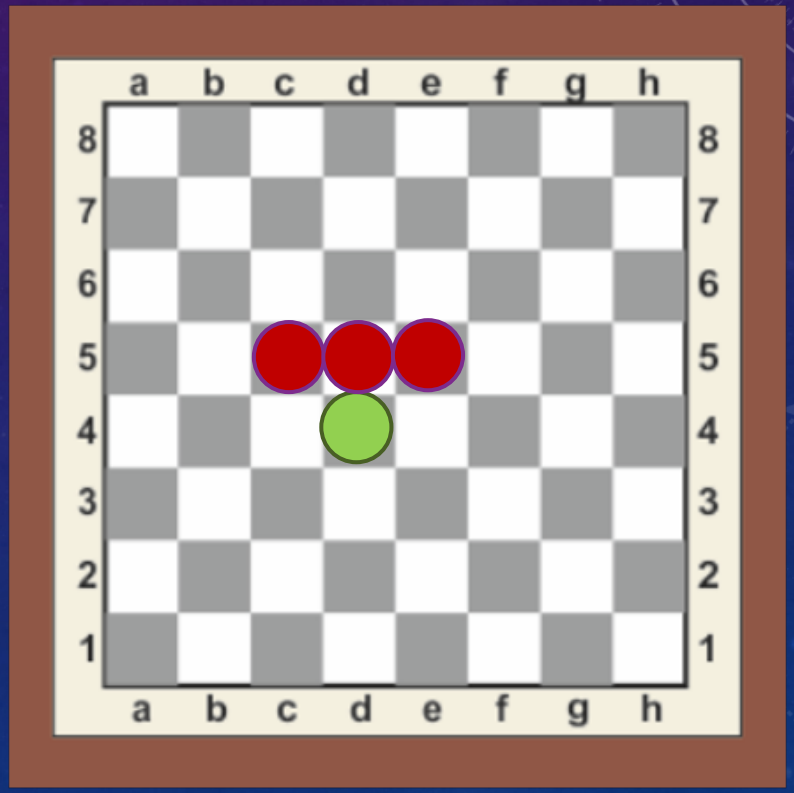
THE KNIGHT PARADOX

A not negligible problem is represented by the Knight piece:

IT CAN JUMP… BUT IT CAN'T JUMP!

When a Knight has to be moved but there are other pieces in front of it could be impossible to move it. So, move the Knight is a nightmare!

The functionality has not been implemented yet, but considering that a Knight moves two positions relative to an axis and one position relative to the other axis, there are different possibility to perform the move. The following is an example.



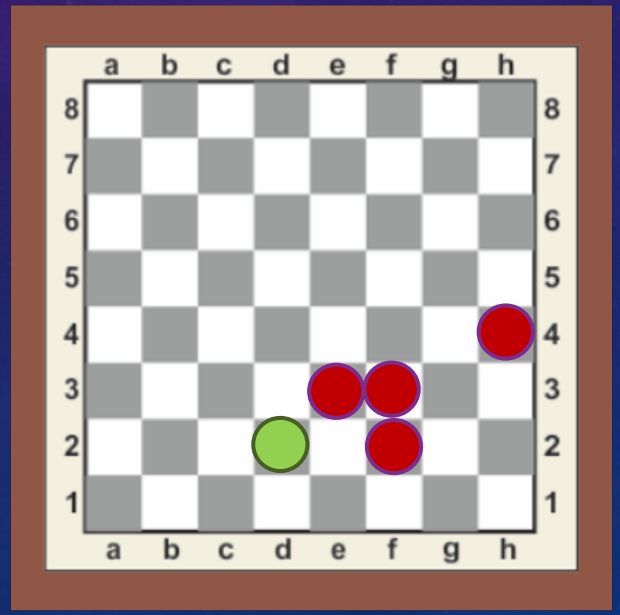
CAVALLO IN E6

ARE NOT YOU HUNGRY?

The same goes for the eaten pawns: we need to build an algorithm that can handle the movements of the pawns eaten towards the exit of the chess board.

Manage all the cases is a difficult task…

PEDINA IN E3



Manage a single case could be easy, but manage all the case with a general algorithm requires a lot more effort

I DON'T REMEMBER

Microcontrollers are very interesting devices: they opened the doors of the imagination to all the makers of the world.

But there are terrible problems to keep in mind for a programmer used to working on machines with great computing capacity: microcontrollers are exactly the opposite.

One of these is the limited dynamic memory that can make the program instable.

Program with 4GB of RAM is one thing, program with 32KB of RAM available is another thing.

YOU HAVE TO TRY!... AND CRY!

ARE YOU READY FOR A LIVE DEMO?

CONCLUSION

This project opened my eyes to the following, no longer negligible aspects:

- Made a project alone is nice, but doing it together is much better. If you make a project alone you can manage your time, you can code with your style. But when there is a problem you can only count on your brain, i.e. one calculator, not two, not three. When you work in a team, instead, you can rely on more brain, i.e. more than one calculator!
- Writing code does not mean being a programmer: writing good code means becoming one.

THANKS

A special thanks to:

- Luca Vecchi, master degree student in Data Science at Statale di Milano
- Alessandro Castiglioni, master degree student at Politecnico di Milano
- Emanuele Falzone, master degree student at Politecnico di Milano

For the thousand tips and the concrete support