DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2019

Human Robot Interaction - Conception and testing of modular robotics kits based on Poppy Ergo Jr

TALLULAH GILLIARD

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

KTH Royal Institute of Technology

Master Thesis

Master in Computer Science - Human Computer Interactions and Design

SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Human Robot Interaction - Conception and testing of modular robotics kits based on Poppy Ergo Jr.

Domain: Robotics, Education

Author:
Tallulah Gilliard

Supervisor:
Gabriel Skantze (KTH Royal Institute of Technology)

Examiner:
Roberto Bresin (KTH Royal Institute of Technology)

Supervisors INRIA Sud-Ouest:
Pierre-Yves Oudeyer
Thibault Desprez

September 10, 2019

Abstract

Innovation in electronics and mechanics have made it possible to construct less expensive robots and made them accessible to inexperimented and curious populations. Educational robots are used in classrooms and homes for both educational or learning purposes and entertainment. Since technology is more and more present in our lives, having programming notions is becoming a key knowledge to be taught.

Poppy is a robotic open source platform for the creation, use and sharing of interactive 3D printed robots, developed by the FLOWERS team at INRIA (French National Institute for Research in Computer Science and Automation). The Flowers team directed a branch of their work towards Poppy Education with the Poppy Ergo Jr, making robotic kits more accessible to beginners. It allows the users to discover computer science and robotics through different activities where they manipulate the tangible robot, and with different level of difficulties for the programming part.

The main goal of this work is to create new variations of the Poppy Ergo Jr to permit students and teachers to transform and modulate their robot. The second goal of this work is that we want to evaluate learning components such as the feeling of choice or the tangible manipulation during an activity and their impact on students perceptions, that through our old and new robotic kits.

After a conception phase for the creation of the new robot Dragster Mini and its corresponding programming activities, we ran two experiments on 7 group samples of high school students performing robotic activities assessed with surveys on their perceptions regarding the two activities evaluating choice (1st experiment) or tangible manipulation (2nd experiment).

After analyzing the results, we have found that having the choice of how to build their robot would put the students in a more "active" attitude, making them go faster than the group with no choice, even if they might feel that they have less choice. We also found that tangible manipulation requires more effort from the students, which may lead to them scoring better results on the quiz. This suggests that the students understand the material better when tangible manipulation techniques are applied.

This work ends by an opening on what can be done for making programming concepts applied to robots and Artificial Intelligence more accessible to children and answer teachers eager to go further. Building an autonomous robot of their own seems to be one of their expectations and could be widened in the near future of the Poppy Education Project.

Keywords : Robotics, Education, Human Robot Interaction, HCI, Learning, Pedagogy, Social Robots, Educational Robots

Sammanfattning

Mänsklig robotinteraktion - Conception och testning av modulära robotik-satser baserade på Poppy Ergo Jr

Innovation inom elektronik och mekanik har gjort det möjligt att konstruera billigare robotar och gjort dem tillgängliga för oexperimenterade och nyfikna personer. Utbildningsrobotar används i klassrum och hem för både utbildnings eller inlärningsändamål och underhållning. Eftersom teknik är mer och mer närvarande i våra liv, betyder det att programmeringsbegrepp blir en viktig kunskap att läras ut.

Poppy är en robot med öppen källkodsplattform för skapande, användning och delning av interaktiva 3D-tryckta robotar, utvecklade av FLOWERS-teamet vid INRIA (French National Institute for Research in Computer Science and Automation). Flowers-teamet ledde en gren av sitt arbete mot Poppy Education med Poppy Ergo Jr, vilket gjorde robotkit mer tillgängliga för nybörjare. Det gör att användarna kan upptäcka datavetenskap och robotik genom olika aktiviteter där de manipulerar den konkreta roboten och med olika svårighetsnivåer för programmeringsdelen.

Huvudmålet med detta arbete är att skapa nya varianter av Poppy Ergo Jr så att elever och lärare kan transformera och modulera sin robot. Det andra målet med detta arbete är att vi vill utvärdera inlärningskomponenter som känslan av val eller den konkreta manipulationen under en aktivitet och deras inverkan på elevernas uppfattningar, det genom våra gamla och nya robotkit.

Efter en befruktningsfas för skapandet av den nya roboten Dragster Mini och dess motsvarande programmeringsaktiviteter, körde vi två experiment på 7 grupp-prover av gymnasieelever som utför robotaktiviteter utvärderade med undersökningar om deras uppfattningar angående de två aktiviteterna som utvärderar val (första experiment) eller konkret manipulation (andra experiment).

Efter att ha analyserat resultaten har vi funnit att om valet av hur man bygger sin robot skulle sätta eleverna i en mer "aktiv" inställning, få dem att gå snabbare än gruppen utan val, även om de känner att de har mindre val. Vi fann också att konkret manipulation kräver mer ansträngning från eleverna, vilket kan leda till att de får bättre resultat på frågesporten. Detta antyder att eleverna förstår materialet bättre när konkreta manipulationstekniker används.

Detta arbete avslutas med en öppning om vad som kan göras för att göra programmeringskoncept tillämpade på robotar och artificiell intelligens mer tillgängliga för barn och svara lärare som vill gå vidare. Att bygga en egen autonom robot verkar vara en av deras förväntningar och skulle kunna utökas inom en nära framtid för Poppy Education Project.

Nyckeord : Robotik, Utbildning, Mänsklig robotinteraktion, HCI, Lärande, Pedagogik, Sociala robotar, Pedagogiska robotar

Acknowledgments

First of all, many thanks to my supervisor Gabriel Skantze, my examiner Roberto Bresin and to the teachers in charge of this master thesis project Jarmo Laaksolahti and Vasiliki Tsaknaki. Thanks for their time and precious teaching.
At Inria Bordeaux Sud-Ouest, I would like to first thank Pierre-Yves Oudeyer, research director in the Flowers team working on reinforcement learning and developmental robotics for supervising me along this master thesis, for his peaceful team leading bringing an healthy atmosphere, and especially for his excellent advices, open minded supervision and inspiration as a researcher.
Then my thanks goes to Thibault Desprez for supervising me during this master thesis, immerse me in the pedagogy, the cognitive sciences and the educational robotics of the Poppy Project. It was great opportunity to experiment on the field with students in schools.
Friendly greetings to Sébastien Forestier for launching me on new projects and allowing me to gravitate arround exploration algorithms and discover Explauto, a library to simulate robotic agents.
Numerous thanks for Alexandre Péré, Rémy Portelas, Cédric Colas, Alexander Ten, Chris Reinke, Adrien Laversanne-Finot, Benjamin Clément, Mayalene Etcheverry, Timothée Anne, Martin Serret, Mehdi Alami, Anouche Banikyan, Marion Schaeffer, Lucie Galland, Hélene Sauzéon, Théo Segonds, my colleagues members (and ex members) of the Flowers team, who have all contributed to daily happiness at work and outside.
Thanks to KTH university. Being a student in such a multicultural and rich environment was really inspiring. Thanks to Paris-Sud, my entry university and to EIT Digital. Many thanks to our teachers from Inria-Saclay, LRI and IRCAM who guided us through these years.
Enormous thanks to David Rene Gonzalez, my opponent for this master thesis, to his clever advises and supportive cheering. Gracias mucho amigo !
Thanks to my friends in Stockholm (teknikringen crew), Bordeaux (moussirideuses, team escalade, kheys) and all my free electrons elsewhere for bringing me joy, fun, talks and fresh new ideas.
Infinity thanks to my family for always have been super supportive, for giving me strength since day one and for bringing happiness in my life.
I dedicate this thesis to Philip Claesson.

Originality declaration

The aim of this report is to create and test variations of educational robots and activities based on Poppy Ergo Jr. To my knowledge, it is the first time of a conception and experiment of that kind to be built on theses sample groups. This report contains the work done by myself, Tallulah GILLIARD, during my internship of 2nd year of Master degree Human Computer Interactions and Design in the Flower team at Inria Sud-Ouest. All the external resources are quoted and appear in the references.

Number of words : 12 784

List of Acronyms

Acronyms

HCI Human Computer Interaction.

HRI Human Robot Interaction.

IMI Intrinsic Motivation Inventory.

SUS System Usability Scale.

TUI Tangible User Interface.

List of tables and figures

List of Figures

List of Tables

Contents

1 INTRODUCTION

Since the birth of educational coding tools and robotics for children in the 1960s, this field has grown substantially in the last years (I was myself part of the first generation of french students to have computer science as an option in high school), recently fuelled by political will to set up educational programs that include computer science more broadly in the school context through new school reforms in France, particularly through the course "Sciences Numériques et Technologie", introducing students to various themes like web programming, data and image processing, or connected objects and robots.

1.1 Context

The National Institute for Research in Computer Science and Automation (Inria) is a French national research institution focusing on computer science and applied mathematics. It is a Public Scientific and Technical Research Establishment and is under the supervision of the French Ministry of National Education, Advanced Instruction and Research.
The FLOWERS team studies mechanisms that can allow humans, robots and algorithms to acquire autonomously and cumulatively repertoires of novel skills over extended periods of time. They develop algorithms, tools and robots for various purposes, the Poppy robot is one of their creations. It was initially designed for research purposes and has appeared to be also adapted for educational purposes.

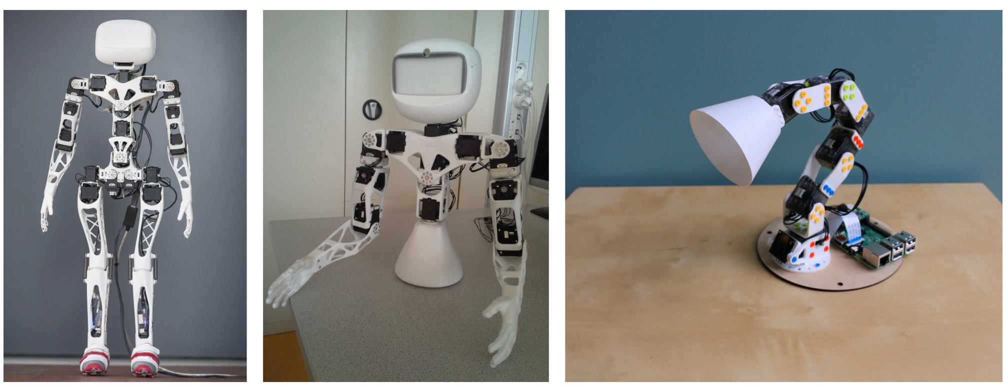


Figure 1: Poppy Humanoid, Poppy Torso and Poppy Ergo Jr

1.2 Aim of project

This master thesis project is based on the Poppy project (see section 2.5), an open source project aiming to develop easy-to-build and use robots for educational, artistic and scientific purposes. Several kits based on this project have been developed, including the Ergo Jr kit that allows you to build your own robotic arm from motors, 3D printed parts and a Raspberry Pi card.
In order to enlarge the collection of Poppy robots, the purpose of this project is to build new robots and activities to allow teachers with Ergo Jr kits to be able to vary their range of activities.

1.3 Problem Statement

In the fields of Human Computer Interaction (HCI), Human Robot Interaction (HRI), and applied developmental psychology research, there is extensive research on the creation of educational robots where we both need technical programming skills but also user behaviour understanding in order to develop coherent solutions. Previous work around the Poppy project consisted of creating a robot and activities, as well as studying the impact of these robots on different populations.

Our first question here is **how we can create new variations of these robots and integrate them into schools programs**. The new robots should be based on the Ergo Jr robot (reuse the expensive parts such as the motors and electronic components) to permit users to transform and modulate their robot and thus come across new challenges and applications. The techniques used must be accessible to the greatest number of users and the resources should be made to be shared and reused. That new robot would be an example of possible variation and could inspire and give a path to follow to users who would like to develop their own variation.

On the parallel of that a second goal of this work is that we want use our robotic kits for **evaluating learning components such as the feeling of choice or the tangible manipulation during an activity and their impact on students perceptions**. First by introducing the new robot to a group of students and teacher users to have early feedback, but also by testing the Ergo Jr robot to investigate the tangible manipulation aspect.

1.4 Methods

In order to fulfill these requirements and fully achieve its purpose, this research needs to explore first how similar problems were addressed by going through existing resources from multidisciplinary fields but also by understanding fully the needs of the users. This can be made by discussing with the teachers and students which will direct my work until the final creation of the robot and its activities. Here we use Computer-Aided Design (CAD), 3D printing, and programming (python). Then two experiments with groups of students are performed to test the validity of the developed product and at the same time allow us to explore different aspects of our educational tools.

1.5 Outline

This present chapter introduced the topic of this research, summarizing the goals to reach and the approach taken. Chapter 2 further expands on the history, background and existing materials. We will go through notions of robotics, learning processes and HCI to understand how we can answer for the best to the objectives of the Poppy Project. Chapter 3 details the conception processes (sketches, 3D printing, codes, activities) and goes through the steps taken and decisions made in order to build such a robot. Chapter 4 describes the method used to be able to fulfill the goals set for the project. The two experiments performed during these 6 months of master thesis are described, links to the online material are given and results are presented. Chapter 5 discusses the findings and limitations of this research. Chapter 6 indicates potential future directions and gives a glimpse of new work to be done. Finally, Chapter 7 gives some overall conclusions of the project based on the results derived from the data measurements and the experience of this whole project.

2 BACKGROUND

2.1 Robotics

Origin and etymology of the term "Robot". It is difficult to trace historically when and where the idea of the robot came to mankind. The term "robot" itself comes from the Czech "*robota*" which means "forced labor, chore, serfdom".
In Greek mythology, Hephaistos is a craftsman, god of metallurgy and fire. Ingenious, he makes remarkable weapons but he also makes wheelchairs that move on their own (equivalent of mobile robots), gold maids that help him in his tasks (domestic service robot), and he even shape Pandora, a clay statue to which Athena will give life (power). (Laumond, 2012)
The first robot according to ancient greek mythology was a bronze giant called Talos (Mayor, 2018). Talos was an animated statue, charged to defend the isle of Crete against pirates. He was built to patrol around the island, detect intruders and sink foreign vessels.

According to Oxford Dictionnary, a robot is "A machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer."

2.1.1 First Automatons

An automaton is a self-operating machine, such as the Duck of Vaucansson, the peacocks of Al-Jazari, the Hercules of Hero of Alexandria, the chess player Mechanical Turk or even a Cuckoo Clock, who by simple gestures pre-programmed mechanically find themselves interacting dynamically with the objects of their environment.

It must be emphasized, however, that the robot is neither an automaton nor a computer. According to Laumond (2012): "Robotics deals with the relationship that can maintain with the real world a machine that moves and whose movements are controlled by a computer. Thus the robot is distinguished from both the automaton, whose movements are mechanically determined, and the computer, which manipulates information but does not move."

2.1.2 Robotic Applications

There are many applications fields for robotics, from industry to medicine, from services to military, but also domestic, entertainment and social robotic assistants. In this report we will focus on social and educational robots (section 2.2).

2.1.3 Moving and interaction techniques

Since the goal of the project is to design and build small educational robots and activities at low cost, it seems that maneuverability and simplicity are the criteria that matter most. Nevertheless, it is important to know and understand the different moving and interaction techniques and their implications.
Each robot has a form of its own fulfilling particular needs. This form entails constraints and

possibilities on the mode of travel. There are thus several great types of locomotion. According to the established goals (eg: to be able to advance on a non flat ground) the movements of the robot can be more or less perfected and the choice of the type of locomotion the robot influences its movements.
There are several ways of locomotion used in robotics. We are focusing on land travel, to distinguish between air and marine movements that are less common and less easy to set up in class and by beginners. Nevertheless, this does not exclude their realization.
Robots can be classified according to the environment in which they operate.

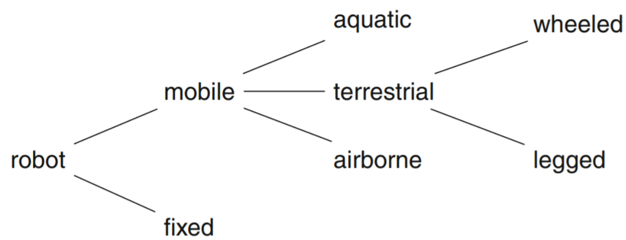


Figure 2: Classification of robots by environment and mechanism of interaction, by Ben Ari and Mondada in "Elements of Robotics" (Ben-Ari and Mondada, 2018)

2.1.3.a Fixed robots

Fixed robots are mainly manipulator robotic arms used in industry with well-defined environments and specific repetitive tasks such as painting parts in car manufacturing plants, setting bolts, etc.. The robotics arms can be used for collaborative robotic purposes. By being compliant and easy to move, the arm directly assists the operator's gesture by multiplying its capacities in terms of efforts to safely handle hot, heavy or bulky objects, or on the contrary too small to be grasped naturally with the necessary precision, while adapting to the characteristics of the user and to the user's motion.
With improved sensors and human-robot interaction devices, robotic manipulators are increasingly used in less controlled environments such as high-precision surgeries.

2.1.3.b Legged robots

There are various forms of legged robots depending on the number of legs and the degrees of freedom. One of the first bipedal robot, Wabot-1 (Kato, 1973), was created by Waseda University in Japan in 1973. In the nineties, the Aibo quadruped robot was developed and marketed by Sony (Fujita and Kitano, 1998), making an advance in the democratization walking robots by offering the consumers a dog-robot pet. The Jansen Mechanism used by Theo Jansen for his Sandbeests, wind powered moving structures, permits to create a leg of one degree of freedom (Nansai et al., 2013). Walking is a challenge because stability must be maintained. Some robots are built specifically to be able to evolve on rough terrain like Boston Dynamics Big Dog and Little Dog (Raibert et al., 2008). Educational robots are often small-sized quadruped robots like Metabots robots (Passault et al., 2016).

2.1.3.c Wheeled robots

Wheeled robots are the most common nowadays. This means of locomotion is highly effective in man-made environments, but is not suitable for use in environments with large irregularities or obstacles to overcome. It is important for a wheeled robot to move without its wheels skidding or slipping off. There are three main categories: unicycle, tricycle, and omnidirectional robots.

Figure 3: Example of unicycle (mBot), tricycle (Google car), and omnidirectional robots (Nexus Robot)

2.1.3.d Other robots

Wheel-Legged robots have the ability of traveling on irregular sloping terrain. This locomotion performance enhance their stability in critical environments (Grand et al., 2004). The Rollerwalker (Hirose and Takeuchi, 1996), Azimut (Michaud et al., 2003), Hylos developped by the CNRS or Handle by Boston Dynamics are typical examples of such robots.

Soft robotics is a field in robotics that deals with robots built out of soft and deformable materials capable to interact with humans and the environment safely. Compliance and softness are inspired by living organisms. The use of electric fields, thermal polymers or inflatable pneumatics, permits to implement the principles of embodied intelligence, or morphological computation.

Swarm robotics is an approach to collective robotics that takes inspiration from the self-organized behaviors of social animals. Through simple rules and local interactions, swarm robotics aims at designing robust, scalable, and flexible collective behaviors for the coordination of large numbers of robots. (Brambilla et al., 2013)

2.2 Robotics in learning contexts, few examples

Innovation in electronics and mechanics have made it possible to construct less expensive robots and made them accessible to inexperimented and curious populations. Educational robots are used in classrooms and homes for both educational or learning purposes and entertainment. Since technology is more and more present in our lives, having programming notions is becoming a key knowledge to be taught. In the book "The Second Self - Computers and the Human Spirit" by

Sherry Turkle (2005), the social role of "intelligent toys" is described as relational artifacts that allow children to explore "matter, life, and mind" (Turkle et al., 2006). For Edith Ackermann, a psychologist who explored the interactions between developmental psychology, play, learning and design, animated toys are "playthings that do things" (Ackermann, 2005), by playing with them, children get genuinely amused and intrigued by their incongruity and consider them as playmates. The history of learning tools dedicated to children begins in the dawn of computing in the 1960s. The possibilities of interactive computer use in schools have been made possible after the invention of computer time-sharing. Wallace "Wally" Feurzeig was one of the first to teach children arithmetic, algebra, and trigonometry problems by coding TELCOMP programs (one of the new high-level interactive programming languages at the time). This appeared to be really motivating for students.

2.2.1 Tangible educational robots

In this section, a non exhaustive list of the widely used educational robots is presented. A large part of them are rolling robots that can be programmed either by setting blocks of code on a visual programming language, or by pressing buttons on the robots to give them instructions to follow. The choice of the robot to use will depend mainly on the age of the children who will use it (for instance Beebot and Bluebot are for really young kids from 4 years old, Thymio is rather dedicated to children from 6 to 16 years old, Poppy Ergo Jr targets students from 12 years old), but also it will depend a lot on the desired activities. The idea behind all of these robots is to have a flexible tool that can adapt to the learning and to different levels of difficulty.

The Valiant turtle was one of the first tangible and programmable device for educational purposes. Turtles robots have appeared int the 1950s and are related to the work of Seymour Papert (Papert, 1980) and Harel ((Harel, 1988) Thesis supervised by Seymour Papert) using the Logo programming language applied to graphics (a pen is attached to the turtle) and mathematical calculation training presented as a game. At the time, computers were mainly reserved to research and teaching coding and computational concepts to children was quite disruptive. One of his key ideas, which started the whole movement of constructionism in education (see section 2.3), was that children learn best by doing.
Similar devices such as the Valiant turtle were also created, like the TORTIS for "Toddler's Own Recursive Turtle Interpreter System" which is a drawing and programable turtle (Perlman, 1974). The OZNAKI project (Cohen, 1977) is a polish project (OZNAKI means "symbols" in polish) based on Logo and Smalltalk languages that has similar drawing applications and can create patterns.

Legos Mindstorm is a software / hardware platform produced by Lego for the development of programmable robots based on Lego building blocks. The first prototype was created at the MIT Media Lab in 1987 based on Lego programming environment. Lego commercialized it one year later. It is named after the book "Mindstorms: Children, Computers, and Powerful Ideas" by Seymour Papert (Papert, 1980).
https://www.lego.com/fr-fr/mindstorms

Cubetto is a robot by Primo Toys created in 2013. It's a wooden robot designed to teach young children how to understand basic functions of coding (loops, conditions). It works by inserting

shapes on the wooden board to create simple algorithm sequences such as moving forward (with the green blocks), turning left (with the yellow blocks) then by pressing execute, it will follow the created sequence.
https://www.primotoys.com/fr/

Beebot Bluebot are robots for young kids with bluetooth connection and buttons on their back. Beebot's back is made of directional arrows keys that can be pressed into a specific sequence, recorded and executed to roll according the order of pressed keys. They can be programmed to do little choregraphy or be used with a pen and draw. Bluebot is controllable via bluetooth and instructions can be sent remotely from a tablet. It can also be used with a tactile board where children place cards to create a linear sequence. Numerous side resources like carpet maps are avaliable with it.
https://www.bee-bot.us/bluebot.html

Kibo is a robot for young kids to discover concepts of mathematics and algorithms by coding with wooden building blocks, creating sequences, and learning design processes.
http://kinderlabrobotics.com/kibo/

Dash and Dot (previously called Bo and Yana) were introduced by Wonder Workshop in 2014 and are two small educational robots. They are used with an app that can be loaded on phone or tablet to have access to pre programmed activities. Dot is a sphere robot that can be programmed by motions. Dash can roll, dance and record sound sequences.
https://www.makewonder.com/robots/dash/

Ozobot is a very small robot that can be coded either by code blocks or by color pens and stickers. It can recognize different colors and follow drawn paths. Color sequences (for instance blue-red-blue dots sequence on a line will make it turn back). A tablet app permits to design paths that the robot will follow.
https://ozobot.com/

Cozmo is a continuous track programmable robot launched by Anki in 2016. You can edit its code Cozmo Code Lab, a block based interface that has different levels. Cozmo can carry and move the blocks and play integrated games. You can also take control of its motions with a tablet and thanks to its camera, see what your robot sees and recognise objects.
https://www.anki.com/fr-fr/cozmo

Cubelets are small coded cubes with different properties stickable together magnetically to form a variety of modular robots by combining their components : some have wheels or rotate, others have distance, heat or brightness captors. It is possible to combine them to create a structure that can interact with the environment.
https://www.modrobotics.com/cubelets/

Metabot is a platform based on basic components (motors, electronics, battery) and 3D printable pieces. It is developed for students, teachers, researchers and robotics enthusiasts.
http://www.metabot.fr/

Tymio is a small wireless robot which will allow kids to discover the universe of robotics and learn a robot's language. Thymio has behaviours like following your hand, avoiding obstacles, following a line, etc. It is possible to program it with Blockly or Scratch.
https://www.thymio.org/home-en:home

Poppy Robot see section 2.5.

Non-robotic educational devices are also available on the market like the Arbalet Project (http://www.arbalet-project.org/), an open-source hardware and software LEDs table, or the Makey Makey (https://makeymakey.com/) an electronic tool and toy that allows users to connect conductive objects (like fruits, flowers, fingers, water,...) to computer programs using a circuit board, alligator clips, and a USB cable to create keyboard and arrows signal. These devices are most of the time compatible with small robots permitting the users to explore different modalities.

Learning assistant robots and Intelligent Tutorials Systems An intelligent tutoring system (ITS) is a computer systems that aims to provide immediate and customized instructions or feedback to learners (Psotka et al., 1988) (Nkambou et al., 2010). They can be used in education to learn new concepts but also in industry to learn to use new tools. Robots are a way to personalize the assistant. For instance the robot P.E.A.N.U.T is a learning companion used to enhance brain-computer interfaces (BCI) usability (Pillette et al., 2017). Embodied conversational agents (ECAs) also play an important role because the communication with the agent can be made with natural language like Furhat, a social robot that communicates through multimodal communication ways (gaze, breath, speech, head motions, visual facial expressions) (Al Moubayed et al., 2012).

2.2.2 Softwares and tools for educational robots

Block coding or a visual programming language (VPL) is any programming language that lets users create programs by manipulating program elements graphically rather than by specifying them textually. They are based on "boxes and arrows", where boxes are treated as entities connected by arrows which represent relations.
Scratch (https://scratch.mit.edu/) has been developed by the MIT Media Lab, and translated into many different languages. It has a large community and their motto is "Imagine, Program, Share" to encourage users to inspire other members.
Snap! (https://snap.berkeley.edu/) have been developed by Berkley University and inspired by Scratch but targets both beginner and advanced coders with extended features. Then a lot of educational robot platform with their own block coding language.

MakeCode (by Minecraft) permit to code with blocks or text and teaches the basics of programming languages, variables, control flow, if statements, loops and functions. More advanced users can explore more complex concepts such as recursion, fractals and object oriented or distributed programming.
https://minecraft.makecode.com/

Block Coding Plateforms for AI Cognimates (http://cognimates.me) is and AI education platform for building games, programming robots training AI models
Machine Learning For Kids (https://machinelearningforkids.co.uk) is free tool that introduces machine learning by providing hands-on experiences for training machine learning systems and building things with them. This web-based platform is being built by Dale Lane using APIs from IBM Watson.
Both of these platforms have been built to help kids to be aware of how our world works, understand the capabilities and implications of it.

3D modeling softwares and CAD softwares are used to do modeling, animation, motion graphic and rendering application. They can be highly specific dedicated to professional levels in engineering, industry or architecture but also more accessible to beginners like Tinkercad, Blender and cinema 4D.

Arduino is an open source hardware and software company that designs and creates board and microcontroller kits for interactive objects. The IDE is a mix between Java and Processing. There is a large online community so it is relatively easy to find resources.
https://www.arduino.cc/

Raspberry Pi is a low-cost Linux-based single-board computer designed to run GNU/Linux. It was originally developed to promote teaching of basic computer science in schools and in developing countries. However it has been beyond its initial target market by reaching robotics, makers and DIY communities. It does not include peripherals (such as keyboards and mice) and cases.
https://www.raspberrypi.org/

2.3 Pedagogy, learning and motivation theories

In order to create relevant activities, it is important to first understand how does the brain processes information and how our learners will integrate that information. Pedagogy is the study of how knowledge and skills are exchanged in a learning context. Theories of pedagogy are based on understanding needs, backgrounds and interests of individual students to adapt teaching strategies and ease the learning processes.

The FLOW from the Flow Theory (Nakamura and Csikszentmihalyi, 2014; Csikszentmihalyi and Nakamura, 1979) is the mental state of a person performing an activity while being fully focused and engage on the task. Whether the skills and challenges are high or low, the person can feel bored

(not enough challenge for the skills) which can be demotivating and eventually ending by quitting the task, or feel anxiety (challenge too intense for skills too low) which can lead to lose confidence and quitting the task.
Montessori Method of education seemed to purposefully set up continuous flow opportunities and experiences for students so Csikszentmihalyi and his colleagues investigated that question by doing multi-year study of student experiences in Montessori. Their research supported observations that students achieved flow experiences more frequently in Montessori settings. (Rathunde and Csikszentmihalyi, 2005)

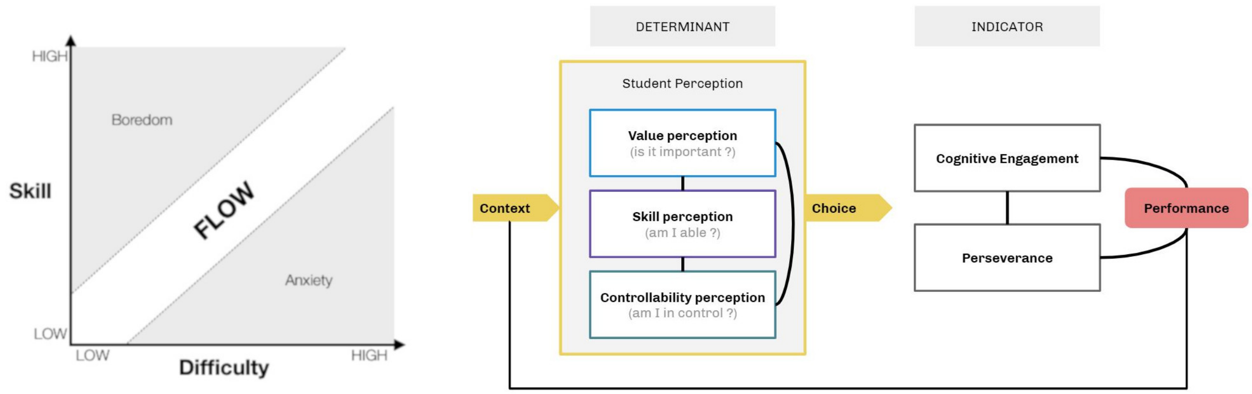


Figure 4: LEFT: Flow theory (Nakamura and Csikszentmihalyi, 2014), RIGHT: Viau model (Viau and Louis, 1997)

Viau model The Viau model of motivation (Viau and Louis, 1997) is about what components have an impact of students motivation. In his book, "La motivation en contexte scolaire" (Viau, 1994), Rolland Viau proposes a model. According to him, the motivation of students depends not only on the learning context, but also of students themselves. The context, independent of them, can influence their motivation. They understand the discipline taught, tasks related to their learning, but also the relationships between students, teachers, administration, etc. Viau defines seven components related to the student that he divides into two groups: determinants and indicators.
Determinants are related to the context of how the student apprehends the tasks.

- "The perception of the value of an activity": It corresponds to the extent to which a student considers an activity useful to achieve a goal. Is it relevant from the student's point of view.
- The perception of one's own skills: It corresponds to a student's self-assessment of their ability to perform a requested activity.
- The perception of controllability: It corresponds to the level of control that the students thinks they have over the realization of the activity and its results. Can students influence the progress of the activity by their choices. This can be created by empowering the students by letting them make choices. The authority and the monopoly of the professor can make fall this feeling of choice.

The indicators :

- "The choice to realize an activity": A student may choose whether or not to perform the activities tasks requested by the teacher, depending on the degree of motivation.
- "Perseverance": perseverance is not defined here as being stubborn, but as being involved and can be related to the time the student invests in their learning
- "The cognitive commitment": Student are cognitively engaged if they apply learning strategies and self-regulation when carrying out their activities. They will be attentive and focused.
- "Student performance": This is influenced by the three previous indicators. Here it refers to the students' use of various strategies, knowledge and know-how to achieve their objectives.

In this model, the different components are interconnected to each other, implying that the motivational dynamic of the student is gradually changing.

The Cognitive load is introduced by the psychologist John Sweller in the 80s (Sweller, 1988), it relates to the amount of information that working memory can hold at one time. Since working memory is limited (Miller, 1956), learning experiences should be designed to reduce working memory load in order to promote schema acquisition. Sweller differentiates cognitive load into three types:

- "Intrinsic" cognitive load is the effort directly associated to the tasks.
- "Extraneous" cognitive load is how the information is presented (if there is irrelevant information presented or if the information is summed up for instance)
- "Germane" cognitive load is the cognitive process that permits to create a permanent store of knowledge and makes the learner going from beginner to expert.

Constructivism theory of learning was developed, among others, by Piaget (Piaget, 1923) in reaction to behaviorism which was, according to Piaget, too limited to stimulus-response association. The constructivist approach emphasizes the activity of the subject to build a representation of the reality around him. The learning is done by students constructing knowledge out of their experiences. (Piaget, 1976)
In parallel to Piaget's cognitive constructivism, Lev Vygotsky developed social constructivism (Vygotsky, 1980) where learning is driven by external forces including culture, language and social interactions. Sociocultural and peer-learning beeing key factors to form mental constructs. The central concept of this theory is mediation, by the expert (the teacher) and the tools (the knowledge). He highlighted several inadequacies of Piagetian constructivism, particularly in terms of school learning (Vygotsky, 1964).
Constructivism is often confused with **constructionism** (Papert, 1980), an educational theory developed by Seymour Papert (who was a student of Piaget), inspired by constructivist and experiential learning ideas of Piaget. Piaget was talking about how mental constructions get formed while Papert is saying that constructing is a good way to get mental constructions built. Levels here are shifting from the physical (constructionism) to the mental (constructivism), from theory to philosophy to method, from science to approach to practice.

2.4 Human Computer Interactions and Tangible Interfaces

A definition of tangible is "real and not imaginary; able to be shown, touched, or experienced" (from Cambridge Dictionary). The tangible is something, it is matter, it is something palpable that gives sensations. This may be an object. So there is a physical effect because the object is manipulated and physical sensory information is felt, but also a cognitive dimension because it will affect the perception and emotions of the user facing the object and its environment. Depending the nature of these emotions and perceptions, the attitude of the user towards the object and situation will be modified.

A Tangible User Interface (TUI) is an interface that allow the user to interact with digital information through the physical environment. These actions of the user can be done thanks to physical objects also called "tokens". They are the physical representation of a virtual information. Eva Hornecker, professor at the University of Bauhaus in Weimar, Germany, defines tangible interfaces as a way to create immersion in the experience. It can support users' spatial cognition, reduce cognitive load, and enable more creative immersion in the problem. Actions that happen in the real space have a virtual parallel (Hornecker, 2004).

Tangible materials are a key factor for the learner. The works by Freinet (1969), Montessori (2013) and Alvarez (2016) show that having a specific object to manipulate focus on a defined task gives the learner autonomy, permits the learner to self correct errors and understand the abstract concepts behind.

We know that haptics are complex phenomenons, during object manipulation tasks the brain selects events to monitor and update task performance. The signal received, afferent type and response properties will rely on the tactile sensory innervation of the hand (Johansson and Flanagan, 2009). Lederman and Klatzky (2009) have depicted six manual "exploratory procedures" for tangible manipulation permitting to examine the following object properties : Texture, Weight, Hardness, Volume, Temperature and Global Shape. Abraira and Ginty (2013) describes anatomical and physiological properties of low-threshold mechanoreceptors, particulary comparison of cutaneous mechanoreceptor subtypes and their response proprieties and locations but also their optimal stimulus (Stretch, vibration, hair follicle deflection, ...).

Since there is different ways to interact through touch, grasp, haptics sensations and manipulation, it can be interesting to explore generated emotions. McGlone et al. (2014) investigate affective touch. They show that a soft brush stroking on hairy skin activates somatosensory areas S1 and S2 as well as the posterior contralateral insular cortex. Insular cortex is a region of great interest in relation to affective mechanisms and is considered as a gateway from sensory systems to the emotional systems of the frontal lobe (Augustine, 1996). It is also believed to play a role in diverse functions linked to emotions like empathy, perception or self-awareness and to the regulation of the body's homeostasis (Damasio et al., 2012).

When it comes to integrate theory applied to objects to manipulate, the previous references are good to keep in mind especially to remember that the tangible manipulation of objects could have an impact on the perception. In their paper "Tangible Bits", Ishii and Ullmer (1997) describe an intelligent way to integrate tangible interfaces in the daily life of the users with affordances which are the possibilities of actions on the objects (Gibson, 1977). Grasping and manipulating objects allow users to center their attention on these physical objects. A parallel can be made between Ishii and Ullmer's quote "Tangible Bits is to bridge the gaps between both cyberspace and the physical environment, as well as the foreground and background of human activities" and Eva Hornecker's quote "Actions that happen in the real space have a virtual parallel" : Eva's "real space" and "virtual" are Ishii's "atoms" and "bits".

2.5 Poppy project

The team FLOWERS (for FLOWing Epigenetic Robots and Systems) at Inria Bordeaux Sud-Ouest and Ensta ParisTech, studies mechanisms that can allow robots and humans to acquire autonomously and cumulatively repertoires of novel skills over extended periods of time. This includes mechanisms for learning by self-exploration, as well as learning through interaction with peers, for the acquisition of both sensorimotor and social skills. Sensorimotor skills include locomotion, affordance learning, active manipulation. Interactive skills include grounded language use and understanding, adaptive interaction protocols, and human-robot collaboration.

Their approach is organized along two strands of research:
Artificial intelligence: constructing machines and robots, inspired by animal cognitive development, and capable of lifelong development, adaptation and interaction with the physical and social world.
Cognitive Science: Elaborating computer and robotic models as tools for understanding developmental processes in humans.

Poppy is a robotic open source platform (both hardware and software) for the creation, use and sharing of interactive 3D printed robots, developed by the FLOWERS team at INRIA (Lapeyre et al., 2014). It has been designed to allow users to easily remove and replace and modify some parts of the robot. It is also an interdisciplinary community of beginners and experts, scientists, educators, developers and artists, who all share a vision: robots are powerful tools to learn and be creative !

There are different Poppy robots : Poppy Humanoid, Poppy Torso and Poppy Ergo Jr.
Poppy Humanoid is the flagship creature. It is modular, parametric and aesthetic. From a single arm to the complete humanoid, this platform is actively used in labs, engineering schools, FabLabs (digital fabrication labs), and artistic projects. As all Poppy robots, it uses 3D printed parts and Dynamixel servomotors, known for their reliability.
Poppy Torso is a simplified version of Poppy Humanoid. It is more affordable and can be installed easily on tabletops, it is particularly suitable for classroom use, in fablabs or in all kinds of public events. The robot is attached to a suction pad with lever arm to ensure the utmost stability.
With its six engines, **Ergo Jr** is the most accessible Poppy robot. It is cheap, portable and easily customizable. Perfectly suited to beginners and young people, it is a good introduction to robotics through a series of turnkey activities.

Originally designed in the "Mathematics: a beautiful elsewhere" (https://flowers.inria.fr/robots/ergo-robots/) artistic experiment created by the Fondation Cartier, the Ergo Jr has been adapted to fit with education needs.
It is a robotic arm, consisting of 6 motors allowing life-like movements and 3D printed elements. The use of rivets make the assembly, modification and reassembly easy. Ergo Jr comes with three tools for different interactions with its environment: a lampshade, a gripper and a pen holder. The robot is controlled with a Raspberry Pi board, and a camera helps it interact with the world. Everything

in the robot is let transparent and accessible to promote an educative approach and an interactive technology.
The Flowers team directed a branch of their work towards Poppy Education with the Poppy Ergo Jr, making robotic kits more accessible to beginners. It allows the users to discover computer science and robotics through different activities where they manipulate the tangible robot, and with different level of difficulties for the programming part. Ergo Jr relies on Snap! (developed by Berkeley University), a variant of Scratch, the visual programming language. Its accessibility and efficiency make it a great tool to discover robotics and understand the digital world. It can be used directly on the robot with no additional installation. It comes with a set of pedagogical activities, developed with high-school teachers, ranging from the discovery of the robot and how to control it to more complex and integrated project. These activities are accessible for free on the project's forums, maintained by Poppy Project members (researchers, robotic companies like Pollen Robotics, teachers). https://forum.poppy-project.org/

Poppy was initially designed for research purposes and has appeared to be also adapted for higher education. Collaborating with schools has been a win-win situation for both researchers and teachers.
For teachers, the main goals were to gain experience of using such tools in a project context, evaluate the potential and limitations for educational purposes and enhance key points of learning concepts for their students.
For us, researchers, we are interested in the reaction of young students to Poppy Education and evaluate the possibility of integrating these technologies in schools' computer science programs to help students acquire programming concepts more easily.

To do so, the Poppy robots have been tested with robotic kits (Noirpoudre et al., 2016) and evaluated (Desprez et al., 2018*b*) to fit the schools demand and by also taking in account the motivation and satisfaction of the learners. These activities aim to promote the students' scientific approaches, cooperation and the creation of a learning world through the robot (Noirpoudre et al., 2017)
They are now distributed in more than 35 schools in Nouvelle-Aquitaine with some very involved professors who contribute to the community by creating various activities for the Poppy Education project https://www.poppy-education.org

3 CONCEPTION OF A ROBOT VARIATION

As defined in the problem statement (section 1.3), the main goal of this work is to create new variations of the Poppy Ergo Jr to enable students and teachers to transform and modulate their robot.

3.1 Ideation process

The two first weeks of my internship were dedicated to understanding the Poppy platform and environment in order to generate alternative versions of the already existing Poppy Ergo Jr. I started in few brainstorming sessions where I tried to think broad about variations without restrictions. I generated several ideated models and existing prototypes and at some point gathering them into a classification was appearing to be relevant.

As common in other design disciplines, sketches are widely used at early stages of development of TUI (Buxton, 2010). They permit to generate, explore and experiment ideas, form factors, user experience and possible relationships between physical interaction objects.

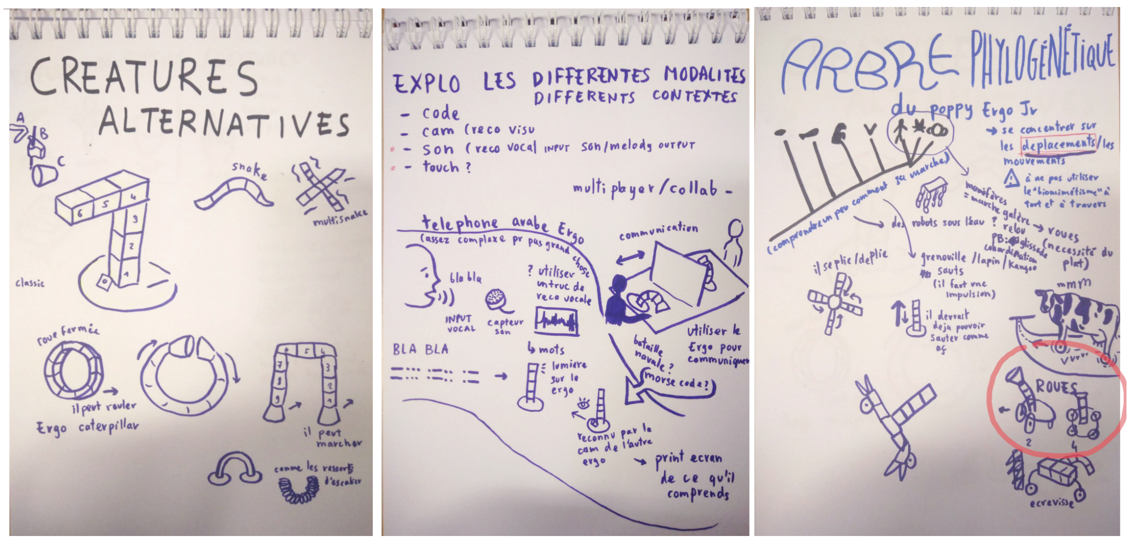


Figure 5: Brainstorming sketches and notes

I created a classification of Ergo robots variations based on the existing ones, the potential new ones and also based on already existing classifications (see section 2.1.3) :

- **Fixed** robots
- Robots with **legs**
- Robots with **wheels**

FIX	LEGS	WHEELS	OTHER
• Ergo Jr starter (**4**) + End Effectors (see below) • Ergo Jr (**6**) + End Effectors - Lamp - Grip - Pen holder - Ball launcher - U end - Labyrinth - Vader head - Captain head - Lynch head - Classic base - Simple base • Torso (**different motors**)	— Crawl — • Soft Starfish (**6**) • Snake mini (**±8**) — Walk — • Transform X (**12**) • Horse Robotica quattro (**12**) • Gipsy (**12**) • Multipod (**9**, **12**, **18**) • Diplo (**12+6**) • Humanoid (**different motors**)	• Poppy 4 wheels (**4**) • Dragster mini (**2+4**)	• Continuous track ? • Flying ? • Aqua ? • ???
Modify hardware structure (.stl files)	Modify hardware + software (motions functions)	Modify hardware + software (motors, motions functions)	Innovate

a	FIX + LEGS / ex : Diplo
b	LEGS + WHEELS / ex : ?
c	FIX + WHEELS / ex : Dragster mini
X	FIX + LEGS + WHEELS / ex : ? (combine everything)

(**n**) number of motors XL-320

Figure 6: Classification of Poppy Variations

For the fixed robots, it is possible to only modify the structure (3D pieces) without modifying the code. The robot keeps the same behaviour and motion functions but the design of its appearance can be modify with computer aided design softwares. For the robots with legs and wheels, modifying the code is necessary because a different structure will affect their motion functions. Combinations of combinations like "fix + legs", "fix + wheels" or even "fix + wheels + legs" are also possible. This is by combining two robot structures that I came up with the Dragster mini : it is a combination of a fixed structure to a wheel structure.

3.1.1 The Dragster mini

The Dragster mini is a Ergo Jr Starter arm of 4 motors fixed on a 2 wheels platform. The goal is to make it accessible to people that already own a Ergo Jr so they can transform it into a Dragster just reusing the motors and 3D printing some parts. Depending the tool used, it is able to grab, hold or throw objects, but it is also able to roll forward, backward and turn. Various activities can then be made : obstacle detection, bringing objects, drawing patterns, etc ...

3.2 Modifying robot structures (hardware)

3.2.1 Creation .stl files

STL is an abbreviation of "stereolithography" and is a file format native to the stereolithography CAD software created by 3D Systems (a company that engineers, manufactures and sells 3D printers). They are widely used for rapid prototyping, 3D printing and computer-aided manufacturing. They are made to describe the surface geometry of a three-dimensional object.
For the Dragster mini, the 3D .stl files were created with TinkerCAD (https://www.tinkercad.com/) by reusing already existing shapes of the Ergo Jr or creating new shapes with simple forms. Since the Dragster is a wheeled robot, I needed rims to fix the wheels so the shape fits our wheels dimensions which are Lego Mindstorms wheels. The base needed to support the arm of the robot (fixed in front), the raspberry, have space for motors and it is possible to add optional batteries to the robot so there is no need to plug it to the power, making it roll more freely in its environment.
The .stl files for the Dragster mini are available here : https://github.com/tgll/poppy-dragster-mini/tree/master/doc/stl

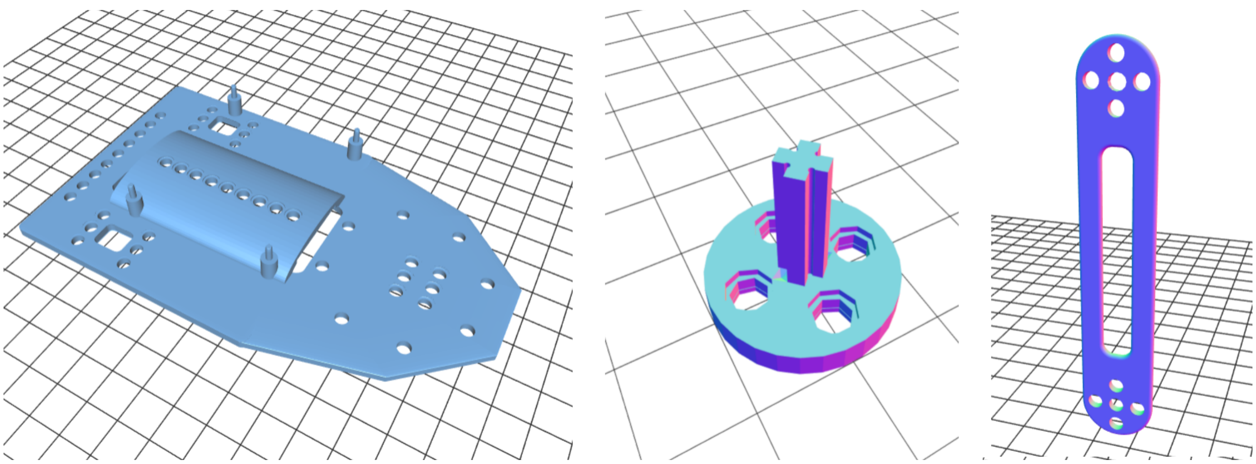


Figure 7: Example of stl files created for the Dragster mini ("dragster-base.stl", "dragster-wheel.stl" and "4dofs-horn2horn-5holes.stl")

3.2.2 3D printing

3D printing (also called additive manufacturing) permits to create three-dimensional object of any shape modelled with a computer and then printed layers by layers of materials (plastics, resins, metals, ceramic, etc) by a machine. Here, I used an Ultimaker 2 Extended + 3D printer and PLA plastic filament to print the pieces of the Dragster mini. To do so, I exported my .stl files to Ultimaker Cura 4.0 software (https://ultimaker.com/en/products/ultimaker-cura-software) made to prepare the 3D models for printing and select the parameters (layer height, infill, support of the pieces, adhesion to the support, material, speed, etc) which will have an impact on the time of printing. To get an idea, printing one "dragster-wheel.stl" of 0.1 layer height (since it is a small

piece, it needs more details so smaller layer) was 17 minutes using 0.13m (1g) of PLA filament. While printing one "dragster-base.stl," of 0.15 layer height was 3 hours and 24 minutes using 2.97m (23g) of PLA filament.
10 Dragster Mini were fully printed for the Dragster Mini experiment. More pieces were printed for prototyping before printing the final pieces.

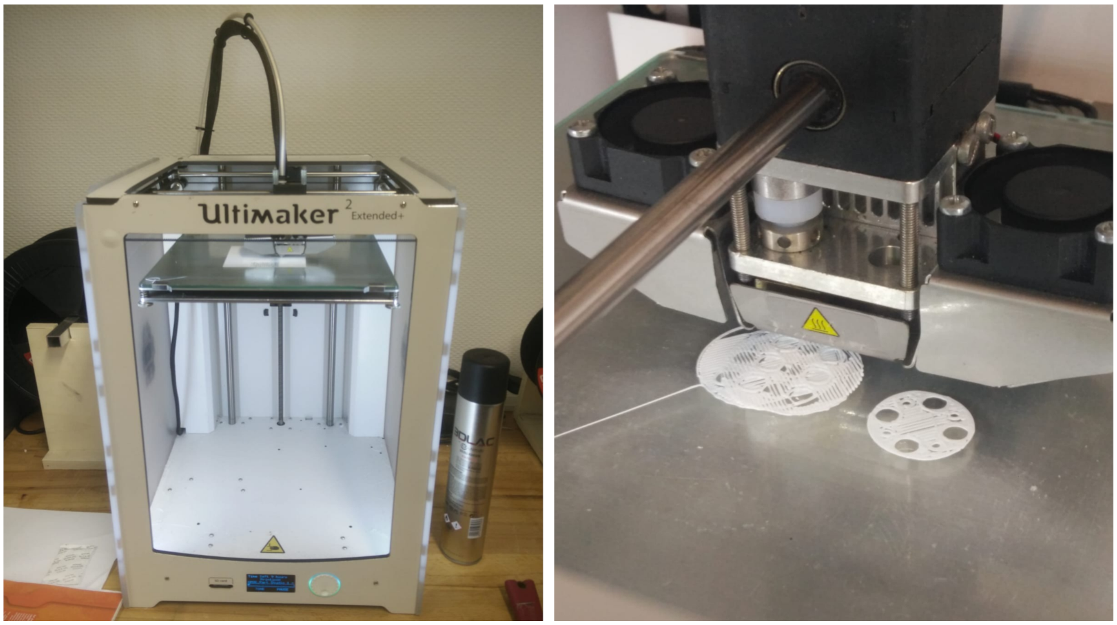


Figure 8: Ultimaker 2 Extended + 3D printer

3.2.3 Assembling

For assembling the pieces together and to the motors we use small rivets, easy to clip in the dedicated holes.
I designed assembling tutorial sheets to give help and guidelines to someone trying to build a Dragster Mini. They describe several steps : electronic assembling, mechanical assembling, motor configuration and testing your robot. Based on the tests and trials run with students and during the pilot study, it seems that it takes about one hour to assemble the robot to a person who never assembled a poppy robot before.
This assembling tutorial sheet can be found here : https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/ActiviteContinue.pdf
A reflection was made about how to present the information to the users, should they have choices of the assembling order (detailed in section 4.1.3.b).

3.3 Adapting moving techniques (software)

3.3.1 Dragster code : json and python

Initially the Ergo Jr robot was not designed to have wheels, so all the motors have limit angles so they don't force the robotic arm in an extreme position which would break it.
Several parameters in the code needed to be adapted to that situation.
(full code files are available in appendix section 9.2)

poppy_dragster_mini.json
This is the example of the wheel settings for "w1" in the .json file. Motors are set differently depending on wether they are joints of the arm (m1, m2, m3, m4) or if they are wheels (w1, w2). Joints depend on a limit angle (between -90° and 90°) while wheels don't have limits angles since they can rotate in one or another direction forever. This code can be easily modify through the jupyter notebook poppy interface (see appendix section 9.1)

```
"w1": {
      "offset": 0.0,
      "type": "XL-320",
      "id": 1,
      "wheel_mode" : true,
      "orientation": "direct"
    },
```

dxlconfig.py
To set the motors properly when the robot is used for the first time, some command lines are needed. A terminal is accessible through the jupyter notebook poppy interface (see appendix section 9.1) by clicking on "new" > "terminal".
Then, typing the following command line will permit to flash the motor "m1"
`poppy-configure ergo-jr m1`
All the motors have to be flashed to run and start the robot. The code below is the code called when a motor is flashed. The code is not directly accessible through the jupyter notebook poppy interface because it is based on the imported framework "Pypot".

```
parser.add_argument('--wheel-mode', type=bool, default=False,
                        help='Set wheel mode.')
```

```
    # Set wheel Mode
    if args.wheel_mode == True:
        print('Set wheel mode')
        with DxlIOPort(args.port) as io:
            io.set_control_mode({args.id :'wheel'})

            time.sleep(.5)
            check(io.get_control_mode([args.id])[0] == 'wheel',
                  'Could not set wheel Mode')
            for speed in range(0,500,10):
                io.set_moving_speed({args.id:speed})
                time.sleep(.1)
            for speed in range(0,525,25):
                io.set_moving_speed({args.id:500-speed})
                time.sleep(.1)
        print('Done!')
```

configure_utility.py

This code defines the possible arguments used in the json file for the motor configuration. It is also part of the "Pypot" framework.

```
    motor_config = c['motors'][args.motor]

    args = [
        '--id', motor_config['id'],
        '--type', motor_config['type'],
        '--port', find_port_for_motor(c, args.motor),
        '--return-delay-time', 0
    ]

    if 'wheel_mode' in motor_config.keys():
        args.extend(('--wheel-mode', motor_config['wheel_mode']))
    else:
        args.extend(('--angle-limit',motor_config['angle_limit'][0],motor_config['
   angle_limit'][1],
                     '--goto-zero'))
```

Pypot (https://poppy-project.github.io/pypot/) is a framework developed in the Inria FLOWERS team to make it easy and fast to control custom robots based on dynamixel motors. It is part of the Poppy Project and is mainly used to control Poppy Creatures. The poppy libraries like Poppy Ergo Jr are built on top of pypot, abstract most of its operating and already come with convenient method for creating and starting the robot.

3.3.2 Activity code : Snap!

Once the robot was ready, the pieces were printed, the motors were set and functional, I needed to have an activity to test it with. The users (students, children and beginners) needed to first understand the basics of the robot : how make it move, take certain positions, make it roll forward/backward, understand motors angles. And then more complex exercises could be done like obstacle detection. The goal was to have an increasing difficulty as the skills improve (see Flow theory in section 2.3). The full activity can be found here :
https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/ActiviteSnap.pdf

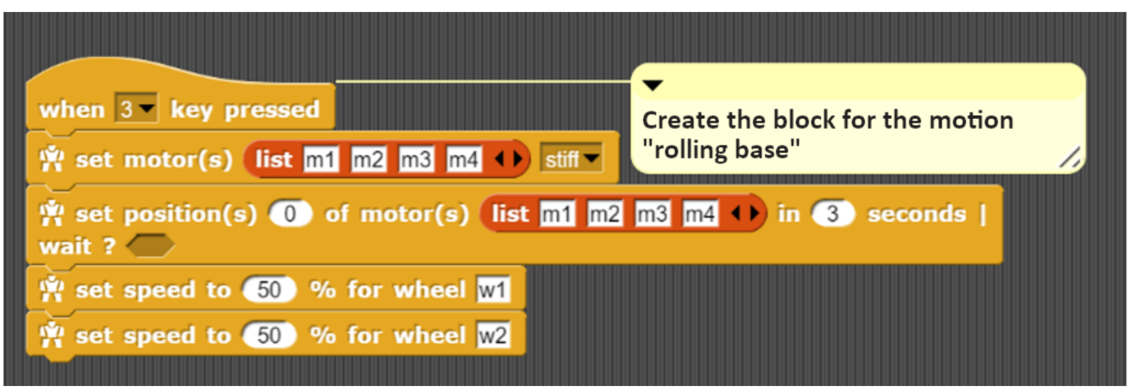


Figure 9: Solution of the exercise 3.a: make the robot roll with its arm straight

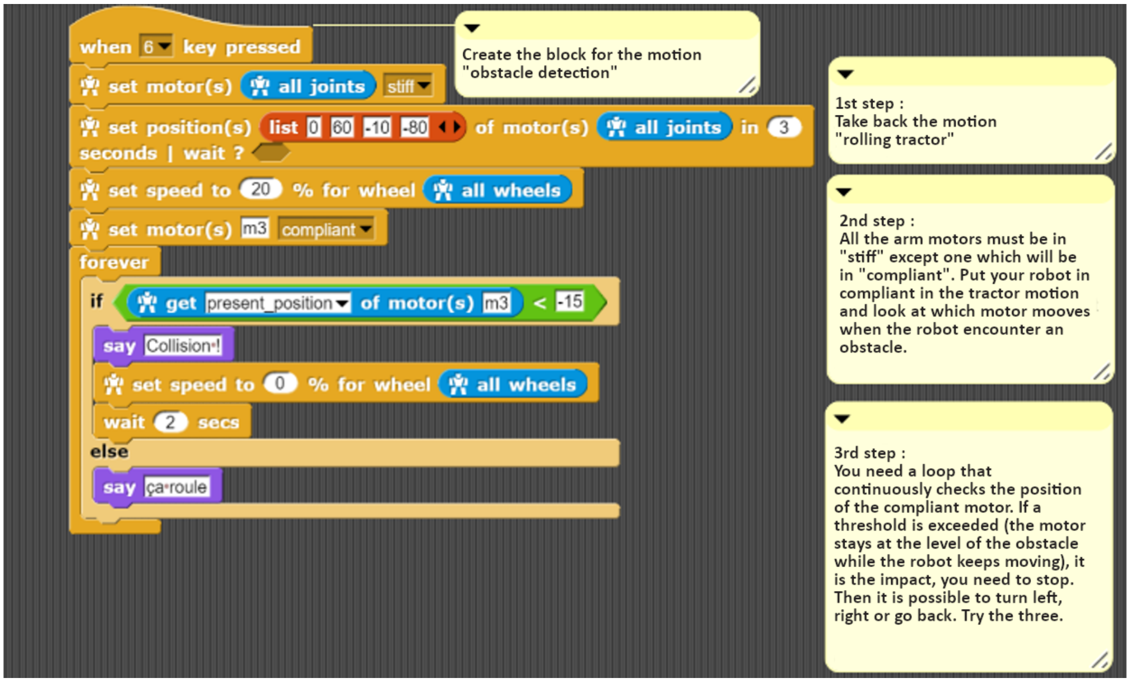


Figure 10: Solution of the exercise 6: Obstacle detection (translated from french)

On the side of the figure 10, there is a description of the actions ("pseudo-code") to do to help the student to come with the presented solution. Further in this report, we evaluate the impact of a pseudo-code description or a tangible demonstration on the student perception (section 4.2).

3.4 Online resources

It is important for the users to be able to find the resources they want online. Several platforms are available and updated by people working on the Poppy Project : The website **Poppy Project** (https://www.poppy-project.org/en/) gather information about poppy robots, the website **Poppy Education** (https://www.poppy-education.org/) propose ideas of robotic activity, the **Poppy Forum** (https://forum.poppy-project.org/) where users can ask their question to the community, and the **Poppy Project Github** (https://github.com/poppy-project/) to get the codes.

3.5 Conception of experiments

Since the Ergo Jr robot is made to be used in classrooms and be accessible for novice users, doing some user tests in schools is a way to evaluate its impact on students. We decided to run two experiments. The first experiment is done with 30 high school students split in 2 groups, working with the Dragster mini robot and focusing on assembling the robot and see if it's better to give them more or less freedom. The second experiment is done with 66 high school students from different classes, working with the Ergo Jr robot, focusing on the Snap! activities and see the impact of tangible manipulation.

4 EVALUATION

4.1 FIRST EXPERIMENT: Dragster Mini & Controllability

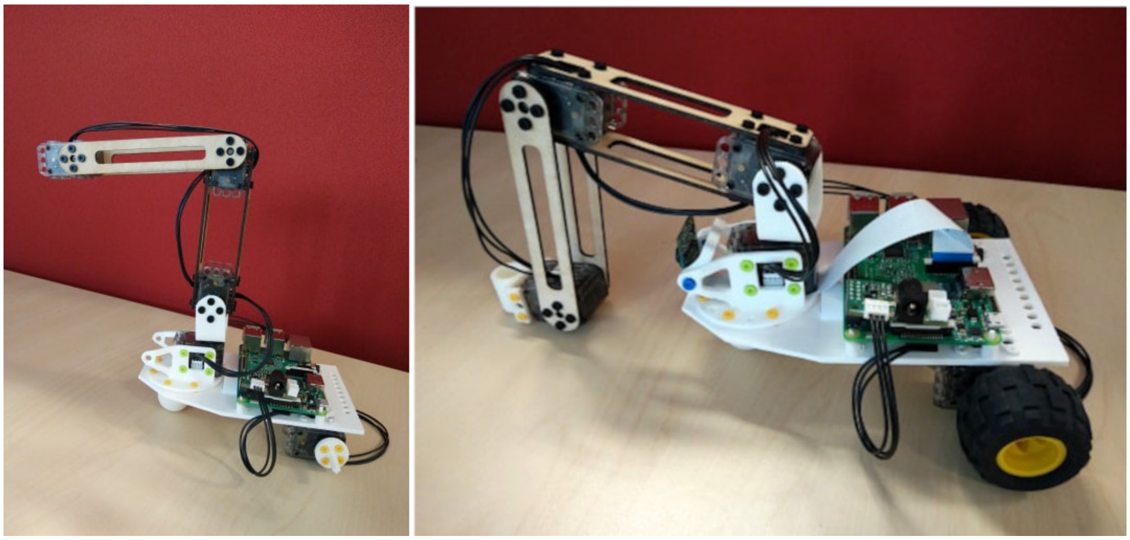


Figure 11: Dragster Mini robot

In this first experiment we want to assess the controllability perception (see Viau Model in section 2.3) of the students and see the impact of different activities practices on their learning, satisfaction and task perception.
The perception of controllability corresponds to the level of control that the students thinks they have over the realization of the activity and its results, this can be created by empowering the students by letting them make choices. So we want to have one group who will make choices between several proposed options while the other group will have to follow instructions step by step.
The authority and monopoly of the professor can make fall this feeling of choice that's why we should interfere as less as possible with the group unless necessary, don't give them authoritarian instructions and let them take their decisions until they call us for validation. Our aim is to design assembling activities that can be presented in a linear format (instructions to follow step by step) or in a modular format (different sheets to choose).

Since this is the first time the Dragster robot will be presented and used by students, we want to focus rather on the assembling than on the programming activities. The programming activities will be the same for both groups, the assembling will differ. That's why students will be spread in two groups, each group being in a different room.

4.1.1 Population

The subjects were coming to visit the Inria Sud-Ouest center so they were available during the morning for the experiment. Since they were all under the age of 18, a consent form signed by the parents was mandatory.

There were 35 students from 1ere S3 lycée La Sauque. (5 were in a group that did not participate to the study).

Number of subjects	Age		Sex	
30	Mean	16	Women	12
	std	1	Men	18

Table 1: Population of the first experiment

They were split in 2 groups :

- Modular group (MG): 15 students, 7 women, 8 men (5 groups of 3) with Dragster Mini, modular activity
- Linear group (LG): 15 students, 6 women, 9 men (5 groups of 3) with Dragster Mini, linear activity

4.1.2 Hypotheses

According Viau model of student's motivation, students can influence positively their progress on the activity by their choices, by feeling responsible and in control. So the group with the assembling presented in a modular format (MG) should feel more in control with a higher feeling of choice, and have more progress on the assembling activity.
Thus, our hypotheses are that **the group which is assembling with the modular activity will be more satisfied and efficient regarding the task realization due to the freedom of choice induced by the different modules (1a)**.
Another hypothesis is that regarding the differences of the activity sheets, they will not be completed at the same time and **we will observe a time difference between the groups (1b)**.

4.1.3 Material

4.1.3.a Dragster robotic kits

- 12 Dragster Mini robotics kits, containing:
  - 3D plastic parts (pre 3D printed) (available on github (https://github.com/tgll/poppy-dragster-mini/tree/master/doc/stl)
  - wooden parts (lazer cut but can be 3D printed)
  - 6 motors XL-320
  - SD cards
  - Raspberry Pi 3
  - Ethernet and power cable
- 12 computers connected to the network of the rooms
- Snap! activity

4.1.3.b Activity Sheets

Linear activity Guide of assembling and configuring the motors of the Dragster mini, all in a continuous sheet. The activity sheet is given to the students with pages attached together, thus they follow the tasks one by one (electronic assembly, motor configuration, mechanical assembling of the base then the arm, testing the robot) by the imposed order.
(https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/ActiviteContinue.pdf)

Modular activity Guide of assembling and configuring the motors of the Dragster mini separate in 8 different sheets. The guidelines are written in the passive form to give freedom in the actions instead of instruction orders (ex: "the motors can be set by entering that command" instead of "set the motors with that command"). The students can choose the sheet to start with even if there is some actions to do before others (the testing cannot be done if the motors are not configured). If that case arrive, the students have to notice themselves that they have to do preliminary tasks.
(https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/ActiviteDiscontinue.pdf)

Snap! activity Snap! (https://snap.berkeley.edu/) has been developed by Berkley University and inspired by Scratch but targets both beginner and advanced coders with extended features.
A Snap! activity was proposed after the assembling of the robot. It permits to the students to check and see in real time the actions of their robots. It also permits to notice and correct potential mistakes in the assembling. The activity was exploring basic functions as moving a motor, then going a bit further with rolling motions and then object detection activity based on what have been learned.
(https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/ActiviteSnap.pdf)

4.1.3.c Surveys

For a more user-centered design approach, collecting qualitative data give us the opportunity to analyze the impact of several components. Regarding our objectives, a first measure of the overall usability of the device was essential. To carry out this study, we have selected two standardized surveys : the SUS ("The Systeme Usability Scales") (Brooke, 1996) and the IMI ("The Intrinsic Motivation Inventory") (McAuley et al., 1989). These two questionnaires allow, on the one hand, to identify the efficiency of the system and on the other hand to account for the perception and motivation of the user during activities. Answers related to students' interest levels are standardized to be able to compare the results.

IMI The Intrinsic Motivation Inventory (IMI) (McAuley et al., 1989) is a multidimensional measurement method intended to assess participants subjective experience related to a target activity in laboratory experiments. It is possible to compose a personalized IMI survey based on what theoretical questions are addressed. The items from those factors are randomly ordered.
The following scales are used:

- n1 Interest/Enjoyment (Did this activity was fun to do ?)

- n2 Perceived Competence (Am I satisfied with my performance at this activity?)
- n3 Effort/Importance (Did I put a lot of effort into this activity?)
- n5 Perceived Choice (Do I believe I had the choice about doing this activity?)

SUS The System Usability Scale (SUS) (Brooke, 1996) is a ten-item attitude Likert scale giving a global view of subjective assessments of usability. As defined by the ISO standard ISO 9241-11, a system usability can be measured by taking into account the context of use (who is using it, what they are using it for, what's the environment). It measures effectiveness, efficiency and satisfaction. It has been reviewed in 2013 (Brooke, 2013).

4.1.4 Experimental Design

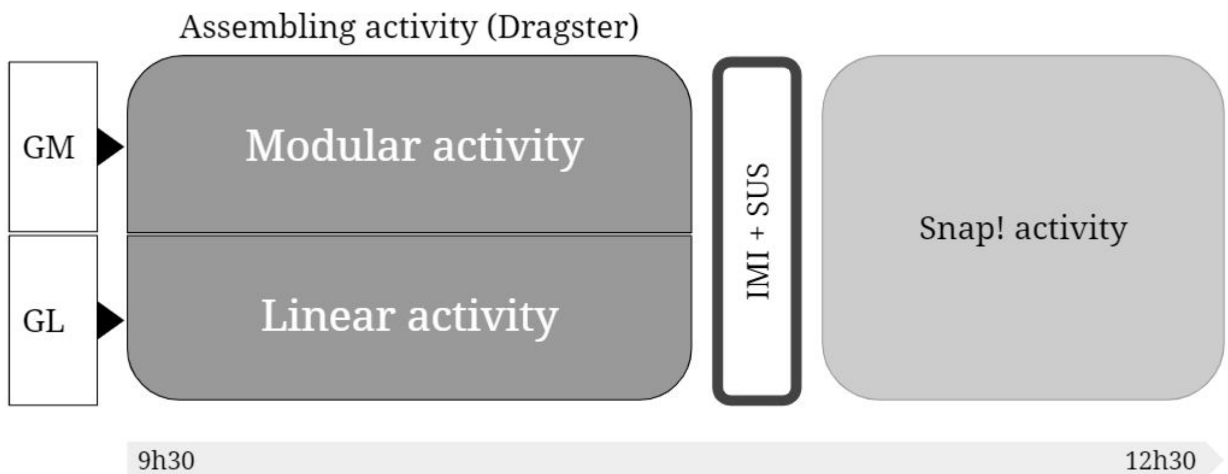


Figure 12: Experimental Design for the first experiment (Dragster Mini)

All the students have arrived at the same time, the groups are pre formed by the teacher. In each room there is tables with one computer and one robotic kit per groups of 3 students.
For the MG (modular activity group) the 8 sheets are disposed on the table separately in the same order for each group. For the group LG (linear activity group) the sheets are stapled and put next to the computer. Both groups MG and LG have a survey of IMI and SUS to fill at the end of the assembling to assess their enjoyment, perceived competence, effort, perceived choice and the usability of the system.
After the assembling activity and answering the surveys, they were programming their robots with Snap! through another activity which was the same for both groups MG and LG. The evaluated part was the assembling.
I had to follow the constraints of time (start at 9:30 and finish at 12:30) imposed by the schedule of the school who was permitting this experiment during its visit in the Inria Sud-Ouest center. And also follow the constraints of the rooms that could fit groups of 15 students.

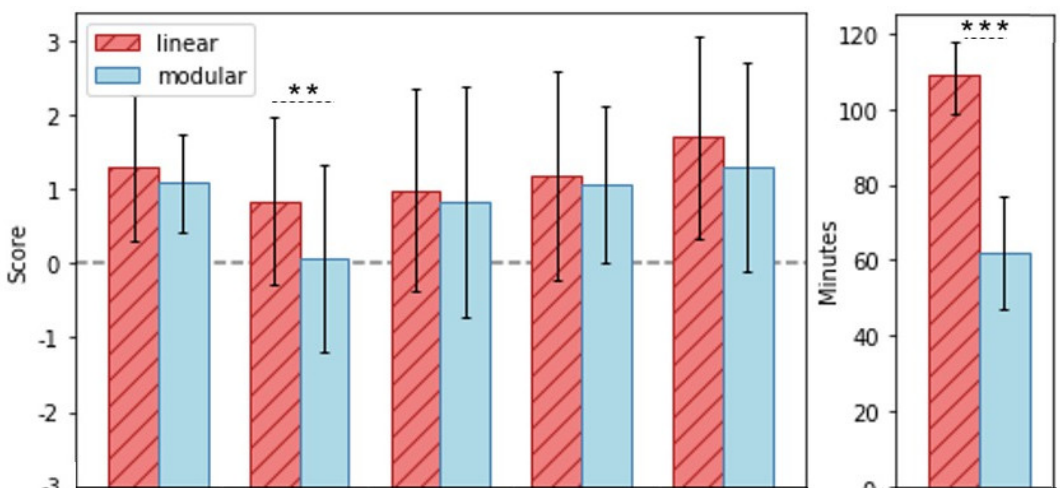
4.1.5 Pilot Study

Before beginning our real experiment, I did a pilot study on 3 students (22 years old students who never programmed before) in order to spot potential mistakes in the activity sheets, missing pictures or information, and also to check how long would the construction of a robot or the realization of the Snap! activity last. The results and feedback of the pilot study helped me to clarify important parts and confirmed the validity of the activities.

4.1.6 Data analysis

The answers collected on paper have been extracted with AMC QCM (https://www.auto-multiple-choice.net/index.fr) and converted into csv files to analyze them. A one-tailed distribution test for two-sample group with unequal variance (heteroscedastic) is used to calculate the means, standard deviation and p-value to the significance level $\alpha = 0.05$ of the Student's t-test.

4.1.7 Survey Results



	SUS		Choice		Competence		Enjoyment		Effort		Time	
mean	1.293	1.080	0.844	0.067	0.989	0.822	1.173	1.053	1.700	1.289	109	62
std	0.979	0.648	1.133	1.255	1.371	1.545	1.408	1.051	1.363	1.406	10	15
t-test p-val	0.24412		0.04287 (**)		0.37849		0.39675		0.21155		$1.88 * 10^{-10}$	

Figure 13: Test results (means) for the first experiment (Dragster Mini)

Each subject from MG and LG (N=30), filled in the IMI and SUS at the end of the assembling. In Fig 13, we observe that for the time of assembling, there is a significant difference (Student test, p=1.88e-10) between the groups modular (mean=62 minutes) and linear (mean=109 minutes). For the IMI test, the results show that there is a significant difference between MG and LG for "Choice" scale (Student test, p=0.04287) which is lower for the modular group MG. However, there are neither any differences between groups for the other IMI scales "Competence" (Student test, p=0.37849), "Enjoyment" (Student test, p=0.39675) and "Effort" (Student test, p=0.21155) that we evaluate, nor between the two groups for the SUS (Student test, p=0.24412).

4.1.8 Observations Results

We tried to interact as little as possible (except when needed, not more than two times per group, and by always redirecting them towards the activity sheets if it's not a technical issue) with the students to not influence the groups and create supplementary differences between them. However, all groups struggled with the rivets so we specified to each sub-groups in both groups how to use the Ollo tool to assemble rivets. Though, some students from the MG didn't used it at all and used their fingers and some other group in MG didn't build fully the rivets. In MG, only one group started with assembling the wheels, the 4 others groups of MG started to assemble first the base of the Ergo Jr starter. Several groups in MG organized their sheets at the beginning of the activity to have an overview. Almost all the groups in MG split their tasks for assembling elements at the same time while almost non in LG split tasks.

4.2 SECOND EXPERIMENT: Ergo Jr & Tangible Manipulation

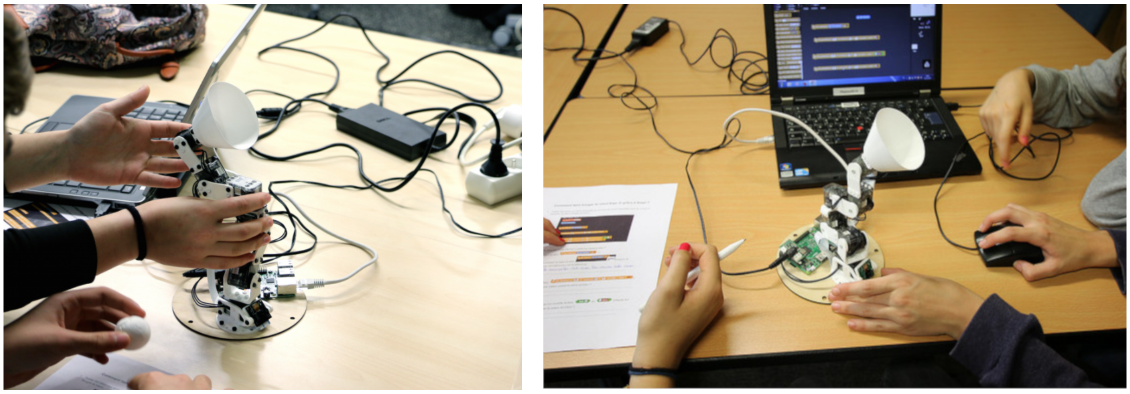


Figure 14: Ergo Jr robot

In this second experiment we want to assess the tangible manipulation impact (see Tangible Interfaces section 2.4) on students. Tangible manipulation can create immersion in the experience. It can support users' spatial cognition, reduce cognitive load, and enable more creative immersion in the problem. It also can gives more autonomy to the learner, permitting to self correct errors and understand the abstract concepts behind. It already has appeared as an efficient support for programming by its use in the classroom. In addition to have access in class to their robot, we want students take advantage of the manipulation opportunities : the robot is not fragile, not dangerous and simple to manipulate. We want to bring them to be more active in the manipulation and want to test if it affects their activity perception and quiz results.
We decided to create two learning situations: one is focusing on demonstration (the student has to reproduce the presented action), the other on description (the student has to reproduce the described action). We wanted a simple action involving sensory-motor loops so the students could manipulate their robot with their hands and have direct feedback. An "animal behaviour" seemed adapted and interesting to recreate. Since the Dragster Mini had wheels we preferred to use the Ergo Jr.
To do so we present to each group a Snap! activity of four pages. Only the last page differs from the others according to if it "Description" or "Demonstration". They have to reproduce a motion of a "chicken" (the head of the robot stays stable when its body is moved), either by observing and manipulating their robot, or by following a list of advises for building the wanted behaviour. Before they start the last page, a video of the chicken behaviour is shown.

4.2.1 Population

The subjects were high school students from high schools in partnership with Inria. There were 66 students from lycée St Genes (Bordeaux) and lycée Condorcet (Bordeaux). 14 students were taken out because they didn't complete or finished the evaluated activity.

Each time, they were split in 2 groups :

Number of subjects	Age		Sex	
52	Mean	15	Women	24
	std	1	Men	28

Table 2: Population of the second experiment

- Demonstration group (Demonstration): 27 students, 11 women, 16 men
- Description group (Description): 25 students, 14 women, 11 men

4.2.2 Hypotheses

Tangible manipulation can lead to more immersion in the problem and can give more autonomy to the learner, permitting to self correct errors and understand concepts more precisely. For our hypotheses **we expect the tangible manipulation to have a stronger impact on the learning results assessed by the quiz for the demonstration group (2a)**. The other scales of the IMI could help us to understand our results.

4.2.3 Material

4.2.3.a Ergo Jr robotic kits

- 6 Ergo Jr robotics kits, containing:
  - 3D plastic parts
  - 6 motors XL-320
  - one SD card
  - one Raspberry Pi 3
  - Ethernet and power cable
- 6 computers connected to the local network
- Snap! activity

4.2.3.b Activity Sheets

Description activity When the students get to the fourth page and start the "chicken challenge", they first watch a video (https://frama.link/ergo-poule) that describes them the task, then read the following instructions : "the motors m1 m2 m3 are in compliant mode and green. m4 m5 m6 are in stiff mode and red. Continuously get the value of the bottom motors, calculate their opposite, send these new values to the top motors" The students have to understand what is meant by that and infere with what they have just learned and recreate that behaviour.
(https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/XP-chicken/ChickenA-description.pdf)

Demonstration activity When the students get to the fourth page and start the "chicken challenge", they first watch a video (https://frama.link/ergo-poule) that describes them the task, then they can click on a button that reproduce the behaviour they have seen in the video. They can manipulate their robot, bend it to the right and see that it compensate to the left and try to understand its actions and reactions. Then few questions are asked to put the students on the right track : "in what mode are the green/red motors ?", "what happen when you only move the motor m1/m2/m3 ?". The students have to reproduce the behaviour.
(https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/XP-chicken/ChickenB-demonstration.pdf)

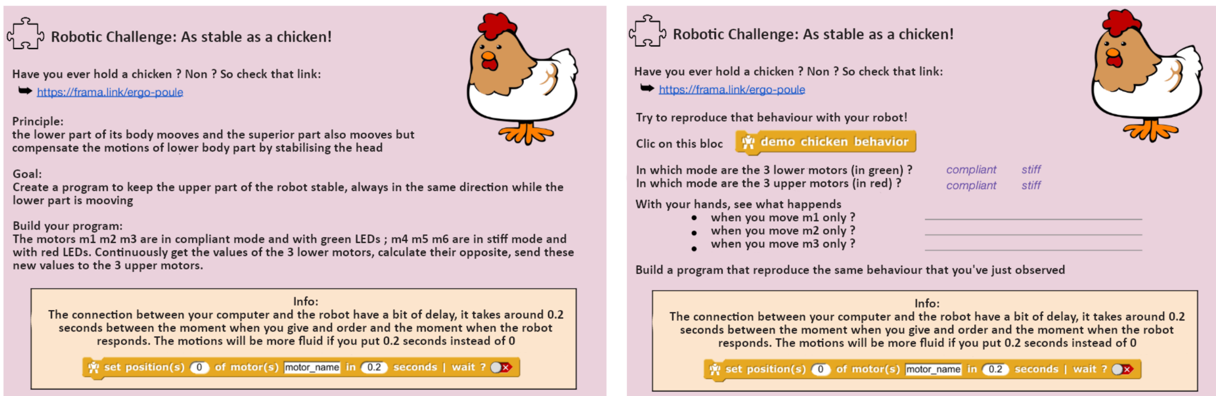


Figure 15: Description and demonstration activities (translated in english, original are in french)

4.2.3.c Surveys

IMI The IMI (McAuley et al., 1989) is the same test than the one used in the previous experiment but with different scales.
The following scales are used:

- n1 Interest/Enjoyment (Did this activity was fun to do ?)
- n2 Perceived Competence (Am I satisfied with my performance at this activity?)
- n3 Effort/Importance (Did I put a lot of effort into this activity?)
- n4 Pressure/Tension (Did I felt pressured while doing this activity?)

quiz The quiz is a two pages quiz with 6 questions. In the last question, 10 code examples are presented, the students have to decide if they are sensory motor loops or not. The goal of is to check the students' knowledge acquisition about what is a variable, how to use the robot, what is a sensory motor loop. Robots moving through an environment need to take the physical laws into account, thus getting inputs from sensory signals and generating motor signals can be done, it's sensory motor loops.
(https://github.com/tgll/poppy-dragster-mini/blob/master/activitysheets/XP-chicken/quiz.pdf)

4.2.4 Experimental Design

All the students arrive in class at the same time and sit by groups of two students per table and with one computer and one robotic kit.
Whether they are in group "Description" or "Demonstration", they all do the same first three exercise sheets at the speed they want. We give them the sheets one by one so they have to call us when they are done with one sheet to obtain the next one. Before giving them the following sheet, we quickly check they are done with the current one and can continue. Since the first sheets are more easy than the fourth one, they are quite autonomous in their work. They are allowed to talk with their partner since they are groups of two but they shouldn't communicate with other groups to ask them for help or give answers. To avoid communications between groups, we simply separate each groups since the classrooms were large enough.
Finally when they are done with the third sheet, they start the last one with the "chicken challenge" which is the part we are interested in. The step is a bit high between the previous exercises that are more guided and this one when they have to come up with their own solution. They watch the video and start to build their functions with Snap!. 1h50 after entering the class, they have to stop where they are, even if they didn't start, if they are close or on the right track, and fill the survey and quiz. Once they are done, they can leave the classroom.

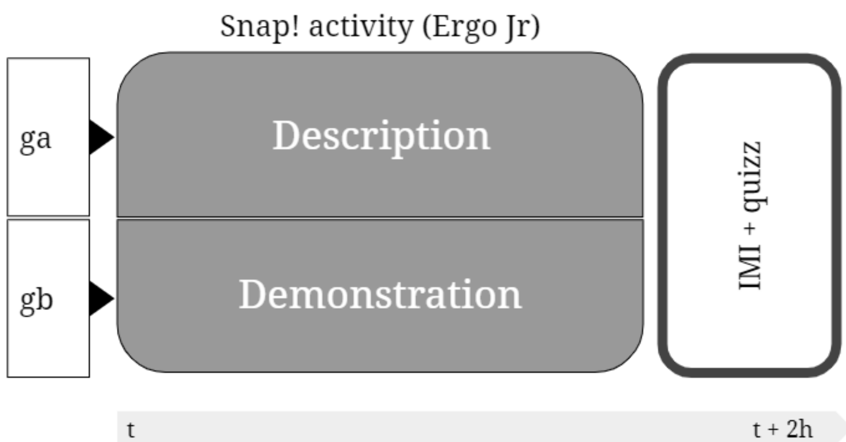


Figure 16: Experimental Design for the second experiment (Ergo Jr)

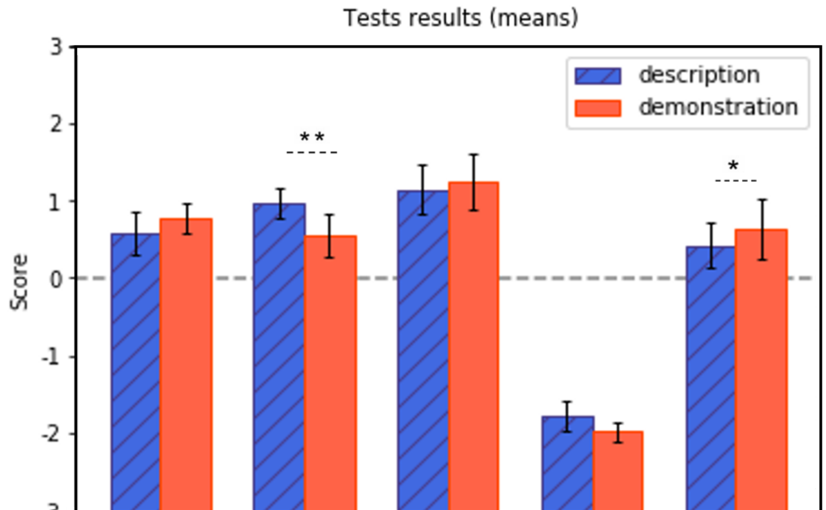
4.2.5 Pilot Study

We performed a prior pilot study on 5 high school students (1 group of 3 and 1 group of 2) with Ergo Jr at the Inria center before doing the classroom experiments. This permitted us to adjust and clarify some instructions. We also met the teachers and visited the two classrooms (from lycée St Genes (Bordeaux) and lycée Condorcet (Bordeaux)) one week before to check computer, connectivity and technical details to make sure everything work directly the day of the experiments. This prior checking was crucial given that the activity depends entirely on the proper functioning of the electronic equipment and its correct connectivity.

4.2.6 Data analysis

As in the first experiment, the collected answers are extracted with AMC QCM and converted into csv files. A one-tailed distribution test for two-sample with unequal variance is used to calculate the means, standard deviation and p-value of the Student's t-test.

4.2.7 Survey Results



	Competence		Effort		Interest		Pressure		Quizz	
mean	0.593	0.772	0.976	0.57	1.149	1.259	-1.76	-1.97	0.433	0.636
std	0.271	0.196	0.2	0.279	0.317	0.363	0.188	0.118	0.294	0.39
t-test p-val	0.2017		0.0417661 (**)		0.266		0.166		0.067 (*)	

Figure 17: Test results (means) for the second experiment (Ergo Jr)

Each participants from demonstration group and description group (N=52), filled in the IMI and quiz at the end of the activity. In Fig 17, it seems that the description would ask slightly more efforts to the students than the demonstration (Student test, $p=0.0417661$ with $p < 0.05$). The students from the demonstration group, even though the significativity is not really high, seem to obtain higher score to the quiz (Student test, $p=0.067$ with $p < 0.1$). We also observe that the scales for the Competence, Effort, Interest and quiz are positives for both groups, while the Pressure scale is negative. However, the scales Competence, Interest and Pressure are not significatively different between the two groups.

4.2.8 Observations Results

Even if visible differences between the groups didn't popped up during the experimentation, some more general observations, especially about the context, could have been made. We run 6 sessions of our experiment, they were all in May and June, three of them were on the last day of the school year so the groups were at the same time happy to do a different activity from usual work but also

looking forward to end. One of the sessions was before the lunch break so, even if the students were doing the activity seriously the motivation was more about going for lunch than the activity itself. Then about the population, some groups were students from "scientific" path and other from "literature" path. Even if the activity and robotics were new for all of them, students from "scientific" path are more used to and comfortable with these logical reasoning of loops and conditions.
About the activity itself, the first and second sheets were usually quite fast to complete for everyone. But then, the step between the chicken challenge and the rest was a bit high, so the students spent more time on it.

5 DISCUSSION

5.1 FIRST EXPERIMENT: Dragster Mini & Controllability

In the previous section, the results showed that controllability and choice in a modular activity (MG) would put the students in a more "active" attitude, even if they might feel that they have less choice.
In this section, we attempt to interpret and discuss the results described in the previous part.
First of all the context is very specific, since it was run in one time with determined conditions, the experiment could be different under other conditions (place, day, number of students per groups). As the experiment was integrated in their "school trip day", it induces a very unusual context.

The time of assembling the robot is the value with the more important gap between the two groups. The modular group was almost twice quicker (mean of 62 minutes for modular and 109 for linear) than the linear. This could be explained by the fact that the assembling task was more easy to split between the three members of each groups with the modular version due to the separated sheets (for instance, one can assemble wheels while one assemble arm). In the observations, almost all the groups split tasks in the modular group (MG) and almost non have done it in the linear group (LG) where there were more one student acting while two others were watching or giving advises. Our hypothesis (**1b**) about the time of assembling that differs between the groups is accepted (hypothesis in section 4.1.2)

We were expecting a more important feeling of choice freedom in the modular group (MG) due to the possibility of choosing between the sheets and the vocabulary used (suggestions and not orders). We were also expecting the impact of modular or linear to be more important and create differences between the groups visible on the scales like maybe having more effort to order the sheets and figure out the order or more enjoyment doing that order investigation task that could have been seen as a game for the modular group. However, only the "Choice" scale is significant but is the opposite of what we predicted : the feeling of choice is higher for the group with the linear activity (LG). This result could be due to groups organizations making appear some form of leadership that could induce some form of constrains inside the groups and reduce the feeling of choice. Our hypothesis (**1a**) about the feeling of choice higher for the modular group (MG) is rejected (hypothesis in section 4.1.2).

5.2 SECOND EXPERIMENT: Ergo Jr & Tangible Manipulation

In the previous section, the results showed that tangible manipulation (Demonstration) requires more effort from the students who might end up getting better results at the quiz so potentially, better understanding of the task.
In this section, the results of the Tangible Manipulation experiment are analyzed. This time we had 66 participants in total and 52 in the presented results. The study was replicated in 6 different contexts, each time the group was split in the two conditions so even if the groups are heterogeneous, it permits to have a much more realistic and ecological study fitting the population "high school students" since our group samples vary between two high schools with students from different paths,

backgrounds and motivations.

First and foremost, our hypothesis (**2a**) about tangible manipulation having a stronger impact on the learning results assessed by the quiz for the demonstration group is accepted (hypothesis in section 4.2.2). But even if the results are in line with our expectations of manipulation that would give better results for the quiz answers and therefore the understanding of the activity, the results are not so significant or really distinct from one group to another. This could be due to the difference between the activities that would be too faded. Maybe seeing the video at the beginning of the activity, even if it was displayed for both groups, has softened the results by putting everyone on the same level and reducing the impact of the demonstration or description instructions since they were more relying on the video than on the fact they were manipulating the robot or reading instructions. However displaying the video is essential to make the students understand the challenge because it displays a real chicken and links it to the robot motions.

Another point I would like to emphasize is that we assume in that experiment that the quiz results are representative of the global understanding of the activity and of the sensory motor loops. Doing quizzes was one of the most efficient ways to test their knowledge (for instance, oral interviews with each students to go more in details, would have been too time taking). We know that putting a student into a quiz situation can be stressful unequally according to the individuals and and alter the results. Yet, if we observe the pressure scale being quite low for both conditions it does not seems that the students felt they were in a stressful situation. Same remark regarding the quiz perception but for a non graded quiz, situation to which the students (French at least) are not used to and that could and which could lead to negligence and taking questions lightly when completing the quiz. This lead to another remark concerning more the observations than the results: The experimentation sessions took place after the end of exams of the year so the students were already a little relaxed in the approach of the holidays, which probably explains the low pressure felt.

We must keep in mind that we are not only interested in seeing the impact of concrete choice or manipulation on students' perceptions, otherwise there would certainly be better tools as well as experimental designs that could be more focused on these specific aspects, but we also want to test, and this in the first place, our robots in a class situation. So they are on a level a tool imposed in this experience that can be seen as a constraint if we focus on the evaluation of choice and manipulation but on another level, they are also and in particular, the center of our interests for our investigations with which we compose and we then seek to evaluate their possible side impacts.

6 LATE BREAKING WORK AND PERSPECTIVES

6.1 Motivation and objectives

In July the schools were closed and no more user test could be run, it was the opportunity to dedicate myself toward new objectives : adapt the Dragster Mini to Exploration Algorithms. This work has been inspired by the APEIA days that we organized in June at Inria which was a gathering about ways to teach AI in schools. During that gathering, pedagogical strategies and objectives were discussed and various activities organized around making AI more accessible to students and teachers to understand how it works but also the socials impacts behind it, biases, impacts of decisions, etc (Program: http://tiny.cc/apeia).

While the previous activities presented in Chapter 4 were made for total beginners, this one is targeting more advanced users. This could permit to schools that want to go further with their robot to do more elaborate activities on one semester instead that in only 2 hours. The objectives are to do a first step into exploration algorithms and apply a learning context to a real robot in a simulated and then physical environment, but also understand the basics of Machine Learning when it's applied to robots, what algorithms could be used, the constrains of a real environment and the advantages bring by doing simulations and therefore use both simulated and real environment to compare their results.

6.2 Background

To go toward this goals, I started to develop an activity and tutorial made to be a study of Exploration Algorithms in developmental robotics with Explauto applied to the Dragster Mini robot. The goal is to implements some exploration algorithms used in developmental robotics and explicit them in a tutorial permitting to get used to the concepts of Goal Babbling, a coordination skill by repetitively trying to accomplish goals related to that skill (Oudeyer and Kaplan, 2009) (Moulin-Frier and Oudeyer, 2013), Active Learning where the learner is actively or experimentally involved in the learning process (Baranes and Oudeyer, 2013), and Model Babbling that drives sensorimotor data collection by considering a modular representation of the sensorimotor space (Forestier and Oudeyer, 2016*a*) (Forestier and Oudeyer, 2016*b*), by experimenting with those strategies in simple environments composed of a robotic arm interacting with objects.
I based myself on Sebastien Forestier's tutorial on Exploration Algorithms (https://github.com/sebastien-forestier/ExplorationAlgorithms) which is applied to a simulated robotic arm grasping objects. I had to change the environment to adapt it to the Dragster Mini.

6.3 Dragster GARDEN : Autonomous Goal Dragster Robot Nano Environment

The Dragster robot is inside a closed environement more or less complex, from a simple corridor to a labyrinth with different rooms. Here I started implementing the corridor. The robot can move forward or backward on its wheels. The corridor is made of a button (B1) on one side, and a room closed by a door with a button inside (B2) on the other side. When pressed, B1 open the door and give access to the room.

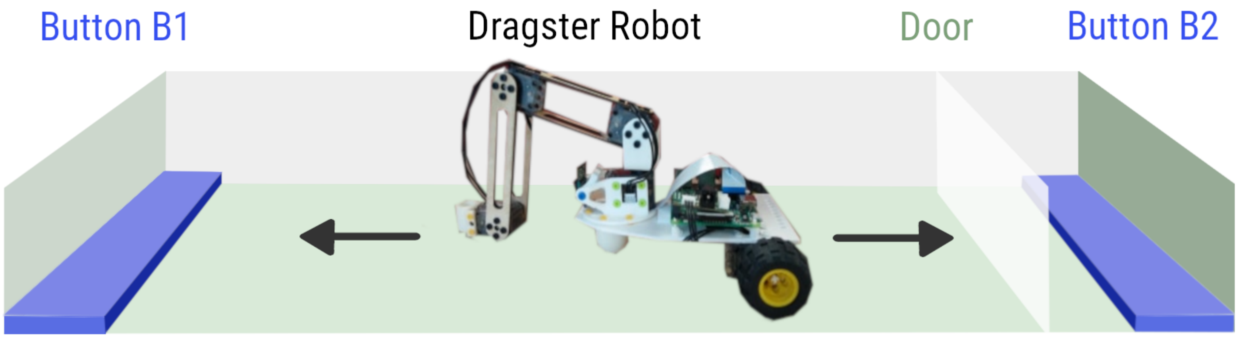


Figure 18: Dragster GARDEN Environment

6.4 Explauto

Explauto (http://flowersteam.github.io/explauto/) is a framework developed by the FLOWERS research team which provide an interface for the implementation for sensorimotor learning algorithms. As explained in the Explauto introduction, an important challenge in Developmental Robotics is how robots can efficiently learn sensorimotor mappings by experience, i.e. the mappings between the motor actions they make and the sensory effects they produce. This can be a robot learning how arm movements make physical objects move, how movements of a virtual vocal tract modulates vocalization sounds, or how motor wheel movements can lead the robot to explore the environment. Learning sensorimotor mappings involves machine learning algorithms, for which Explauto provides a unified interface through the SensorimotorModel abstract class.

I implemented the simulation environment and the Random Motor Babbling. The next steps are Random Goal Babbling and the Active Goal Babbling to compare them together. Considering that the environment will not be too simple, we should expect the results of the exploration to be better for the Active Goal Babbling than the Random Goal Babbling, which is better than the Random Motor Babbling.

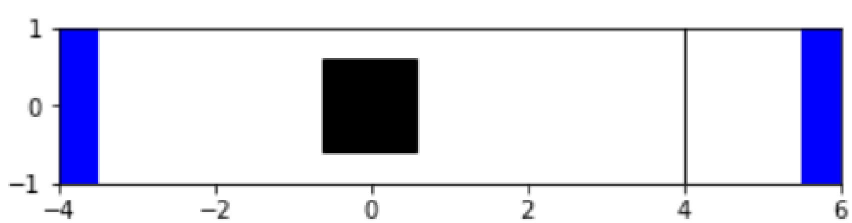


Figure 19: Dragster GARDEN Simulation environment

7 CONCLUSIONS

7.1 Overview

In this master thesis, our goals were to create new variations of the Poppy robot and integrate them into schools programs, and we also wanted to use our robotic kits for evaluating learning components such as the feeling of choice or the tangible manipulation during an activity and their impact on students perceptions, that in a class context.
This work has been made possible thanks to several steps. First an advanced analysis of the state of the art and Background (Chapter 2) in various and multidisciplinary fields. Studying technical implementations of different robotic structures was a mandatory task before embarking on the manufacture of a new robot. But also to study the cognitive processes involved in learning, memorizing, and motivation in human users and especially in the concrete case of students, and then compare educative tools and techniques was also essential before starting to work with users. This multidisciplinary approach mixing technical and human aspects fitted what I learned throughout my master degree and permits me to apply my knowledge to a concrete case.
My experience in prototyping and design helped me to structure my thoughts and generate ideas during the Conception phase (Chapter 3). Having access to the 3D printer was a great opportunity to prototype and manufacture the 3D pieces for the Dragster robot. The first goals of this master thesis work were achieved when the code of the Dragster robot was compiling and the pieces were printed leading to the birth of a new robot.
The conception of experiments for the Evaluation phase (Chapter 4) was first of all a way to introduce and test our robots with students in a real context to observe their behaviour and reactions toward our robots. It was also an opportunity to explore some psychological aspects such as motivation and what influences it, the presence of freedom and choice in an activity or the impact of tangible manipulation.
Given the results presented in the sections 4.1 Controllability and 4.2 Tangible Manipulation, this study can provide the following answers which are to be taken sparingly considering the results' significance:

- More controllability and choice in a modular activity (GM) would put the students in a more "active" attitude, even if they might feel that they have less choice.
- Tangible Manipulation (Demonstration) requires more effort from the students who might end up getting better results at the quiz so potentially, better understanding of the task.

In the Discussion (Chapter 5) we make an attempt to give possible explanations for the previous results by focusing on the shortcomings and margins of improvements in the experiments.

7.2 Benefits, Ethics and Sustainability

The work of this master thesis was the opportunity to create a new robot that could be reused by Ergo Jr users or use directly. Assembling sheets and Snap! activities going with it are also included in that creation. This foster the cross breeding of ideas in the online open source Poppy community, give more tools and challenge to explore to students and material to elaborate their programs to teachers. In addition to that, the city of Bordeaux has a strong robotic and maker community with various actors in research, companies and schools, several fabrication labs and regular meet-ups, conferences and hackathons. The city of Bordeaux has also officially been chosen to host the RoboCup 2020, the annual international robotics competition promoting robotics and AI research.

7.3 Improvements

One of the limitations of this research project is that it heavily relies on users data which is not generated by some anthropometric or physical measures but only by self-determined "feelings" mapped by questions (except for the time and quiz data). We can only base our analysis on these data and rely on our interpretations. For instance hypothesize why would the modular activity tends to decrease the feeling of choice. In addition to these interpretations, the results described in the discussion section raised a series of new questions, opening new directions for future explorations. One direction that could be explored is obviously try these questions on different group samples but also increase the differences between the activity or investigate what other scales could be used to evaluate different components.
Despite this, one of the main concern that remains is to be able to provide a robotic activity that could be simple to replicate in another context or by another teacher, so the complexity should not be accentuated at the expense of the scalability.

7.4 Perspectives

The last part of this investigation developed in Late Breaking Work (Chapter 6) is an opening on what can be done for making programming concepts applied to robots and Artificial Intelligence more accessible to children. With today's ubiquity of connected devices and algorithms place in daily life, it is natural that children want to understand how does Siri or Alexa can talk and know so much, how can autonomous vehicle drive without having accidents, how does Youtube recommend videos, etc... Providing tools that teachers, children, beginners and enthusiasts could use and especially modify is a key element of this educational poppy platform. We have seen that the Ergo Jr already has a community of users and a varied range of activities, mostly starter activities that can be implemented in a two hours session. Many teachers who were using the Ergo Jr since more than a year were eager to go further either by developing themselves new activities, or by searching for ideas in the community. The Dragster Mini is an example of variation with a new structure that permits to address new challenges. It is also an example of what can be made in terms of variations and how to process to create yours according to your imagination, needs and wishes. Several teachers, students, as well as current trends direct their interests towards Artificial Intelligence, accordingly, develop an activity allowing the discovery of exploration algorithms to build one's own autonomous robot seems to be part of these expectations and could be widened in the near future.

8 REFERENCES

Abraira, V. E. and Ginty, D. D. (2013), 'The sensory neurons of touch', *Neuron* **79**(4), 618–639.

Ackermann, E. K. (2005), Playthings that do things: a young kid's incredibles!, *in* 'Proceedings of the 2005 conference on Interaction design and children', ACM, pp. 1–8.

Al Moubayed, S., Beskow, J., Skantze, G. and Granström, B. (2012), Furhat: a back-projected human-like robot head for multiparty human-machine interaction, *in* 'Cognitive behavioural systems', Springer, pp. 114–130.

Alvarez, C. (2016), *Les Lois naturelles de l'enfant: La Révolution de l'éducation*, Les Arènes.

Augustine, J. R. (1996), 'Circuitry and functional aspects of the insular lobe in primates including humans', *Brain research reviews* **22**(3), 229–244.

Baranes, A. and Oudeyer, P.-Y. (2013), 'Active learning of inverse models with intrinsically motivated goal exploration in robots', *Robotics and Autonomous Systems* **61**(1), 49–73.

Bayle, B. (2008), 'Robotique mobile', *Ecole Nationale Superieure de Physique de StrasbourgUniversite Louis Pasteur* **2007**.

Ben-Ari, M. and Mondada, F. (2018), *Elements of robotics*, Springer.

Brambilla, M., Ferrante, E., Birattari, M. and Dorigo, M. (2013), 'Swarm robotics: a review from the swarm engineering perspective', *Swarm Intelligence* **7**(1), 1–41.

Brooke, J. (1996), 'Sus-a quick and dirty usability scale', *Usability evaluation in industry* **189**(194), 4–7.

Brooke, J. (2013), 'Sus: a retrospective', *Journal of usability studies* **8**(2), 29–40.

Buxton, B. (2010), *Sketching user experiences: getting the design right and the right design*, Morgan kaufmann.

Cohen, H. A. (1977), Oznaki and beyond", *in* 'Proceedings of the 1977 National Educational Computing Conference", The University of Iowa', pp. 170–178.

Csikszentmihalyi, M. and Nakamura, J. (1979), 'The concept of flow', *Play and learning* pp. 257–274.

Damasio, A., Damasio, H. and Tranel, D. (2012), 'Persistence of feelings and sentience after bilateral damage of the insula', *Cerebral Cortex* **23**(4), 833–846.

Desprez, T., Noirpoudre, S., Segonds, T., Caselli, D., Roy, D. and Oudeyer, P.-Y. (2018*a*), Conception et évaluation de kits robotiques pédagogiques: Analyse écologique et expérimentale des utilisations de la robotique à l'école en termes de connaissances et de représentations, *in* 'Journée de l'EDMI'.

Desprez, T., Noirpoudre, S., Segonds, T., Caselli, D., Roy, D. and Oudeyer, P.-Y. (2018*b*), Poppy ergo jr: un kit robotique au cœur du dispositif poppy éducation, *in* 'Didapro 7–DidaSTIC. De 0 à 1 ou l'heure de l'informatique à l'école'.

Forestier, S. and Oudeyer, P.-Y. (2016*a*), Curiosity-driven development of tool use precursors: a computational model.

Forestier, S. and Oudeyer, P.-Y. (2016*b*), Modular active curiosity-driven discovery of tool use, *in* '2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, pp. 3965–3972.

Freinet, C. (1969), *Pour l'école du peuple*, Maspero Paris.

Fujita, M. and Kitano, H. (1998), 'Development of an autonomous quadruped robot for robot entertainment', *Autonomous Robots* **5**(1), 7–18.

Gibson, J. J. (1977), 'The theory of affordances', *Hilldale, USA* **1**(2).

Grand, C., Benamar, F., Plumet, F. and Bidaud, P. (2004), 'Stability and traction optimization of a reconfigurable wheel-legged robot', *The International Journal of Robotics Research* **23**(10-11), 1041–1058.

Harel, I. R. (1988), Software design for learning: Children's construction of meaning for fractions and Logo programming, PhD thesis, Massachusetts Institute of Technology.

Hirose, S. and Takeuchi, H. (1996), Study on roller-walk (basic characteristics and its control), *in* 'Proceedings of IEEE International Conference on Robotics and Automation', Vol. 4, IEEE, pp. 3265–3270.

Hornecker, E. (2004), Tangible user interfaces als kooperationsunterstützendes medium, PhD thesis, University of Bremen, Germany.

Ishii, H. and Ullmer, B. (1997), Tangible bits: towards seamless interfaces between people, bits and atoms, *in* 'Proceedings of the ACM SIGCHI Conference on Human factors in computing systems', ACM, pp. 234–241.

Johansson, R. S. and Flanagan, J. R. (2009), 'Coding and use of tactile signals from the fingertips in object manipulation tasks', *Nature Reviews Neuroscience* **10**(5), 345.

Kato, I. (1973), 'Development of wabot 1', *Biomechanism* **2**, 173–214.

Lapeyre, M., Rouanet, P., Grizou, J., Nguyen, S., Depraetre, F., Le Falher, A. and Oudeyer, P.-Y. (2014), Poppy project: open-source fabrication of 3d printed humanoid robot for science, education and art, *in* 'Digital Intelligence 2014', p. 6.

Laumond, J.-P. (2012), *La robotique: une récidive dâĂŹHéphaïstos: Leçon inaugurale prononcée le jeudi 19 janvier 2012*, number 224, Fayard.

Lederman, S. J. and Klatzky, R. L. (2009), 'Haptic perception: A tutorial', *Attention, Perception, & Psychophysics* **71**(7), 1439–1459.

Mayor, A. (2018), *Gods and Robots: Myths, Machines, and Ancient Dreams of Technology*, Princeton University Press.

McAuley, E., Duncan, T. and Tammen, V. V. (1989), 'Psychometric properties of the intrinsic motivation inventory in a competitive sport setting: A confirmatory factor analysis', *Research quarterly for exercise and sport* **60**(1), 48–58.

McGlone, F., Wessberg, J. and Olausson, H. (2014), 'Discriminative and affective touch: sensing and feeling', *Neuron* **82**(4), 737–755.

Michaud, F., Létourneau, D., Arsenault, M., Bergeron, Y., Cadrin, R., Gagnon, F., Legault, M.-A., Millette, M., Pare, J.-F., Tremblay, M.-C. et al. (2003), Azimut, a leg-track-wheel robot, *in* 'Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)', Vol. 3, IEEE, pp. 2553–2558.

Miller, G. A. (1956), 'The magical number seven, plus or minus two: Some limits on our capacity for processing information.', *Psychological review* **63**(2), 81.

Montessori, M. (2013), *The montessori method*, Transaction publishers.

Moulin-Frier, C. and Oudeyer, P.-Y. (2013), Exploration strategies in developmental robotics: a unified probabilistic framework, *in* '2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)', IEEE, pp. 1–6.

Nakamura, J. and Csikszentmihalyi, M. (2014), The concept of flow, *in* 'Flow and the foundations of positive psychology', Springer, pp. 239–263.

Nansai, S., Elara, M. R. and Iwase, M. (2013), 'Dynamic analysis and modeling of jansen mechanism', *Procedia Engineering* **64**, 1562–1571.

Nkambou, R., Mizoguchi, R. and Bourdeau, J. (2010), *Advances in intelligent tutoring systems*, Vol. 308, Springer Science & Business Media.

Noirpoudre, S., Roy, D., Demangeat, M., Desprez, T., Segonds, T., Rouanet, P., Caselli, D., Rabault, N., Lapeyre, M. and Oudeyer, P.-Y. (2016), 'Livret pédagogique: Apprendre à programmer poppy ergo jr en snap!'.

Noirpoudre, S., Roy, D., Desprez, T., Segonds, T., Caselli, D. and Oudeyer, P.-Y. (2017), Poppy education: un dispositif robotique open source pour l'enseignement de l'informatique et de la robotique, *in* 'EIAH 2017-Environnements Informatiques pour l'Apprentissage Humain', p. 8.

Oudeyer, P.-Y. and Kaplan, F. (2009), 'What is intrinsic motivation? a typology of computational approaches', *Frontiers in neurorobotics* **1**, 6.

Papert, S. (1980), *Mindstorms: Children, computers, and powerful ideas*, Basic Books, Inc.

Passault, G., Rouxel, Q., Petit, F. and Ly, O. (2016), Metabot: a low-cost legged robotics platform for education, *in* '2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)', IEEE, pp. 283–287.

Perlman, R. (1974), 'Tortis (toddler's own recursive turtle interpreter system).'.

Piaget, J. (1923), 'La pensée symbolique et le pensée de l'enfant.', *Archives de psychologie* .

Piaget, J. (1976), Piaget's theory, *in* 'Piaget and his school', Springer, pp. 11–23.

Pillette, L., Jeunet, C., Mansencal, B., N'Kambou, R., N'Kaoua, B. and Lotte, F. (2017), Peanut: Personalised emotional agent for neurotechnology user-training, *in* '7th International BCI Conference'.

Psotka, J., Massey, L. D. and Mutter, S. A. (1988), *Intelligent tutoring systems: Lessons learned*, Psychology Press.

Raibert, M., Blankespoor, K., Nelson, G. and Playter, R. (2008), 'Bigdog, the rough-terrain quadruped robot', *IFAC Proceedings Volumes* **41**(2), 10822–10825.

Rathunde, K. and Csikszentmihalyi, M. (2005), 'Middle school students' motivation and quality of experience: A comparison of montessori and traditional school environments', *American journal of education* **111**(3), 341–371.

Sweller, J. (1988), 'Cognitive load during problem solving: Effects on learning', *Cognitive science* **12**(2), 257–285.

Turkle, S. (2005), *The second self: Computers and the human spirit*, MIT Press.

Turkle, S., Breazeal, C., Dasté, O. and Scassellati, B. (2006), 'Encounters with kismet and cog: Children respond to relational artifacts', *Digital media: Transformations in human communication* **120**.

Viau, R. (1994), *La motivation en contexte scolaire*, Éditions du Renouveau pédagogique,.

Viau, R. (2007), *La motivation dans la création scientifique*, PUQ.

Viau, R. and Louis, R. (1997), 'Vers une meilleure compréhension de la dynamique motivationnelle des étudiants en contexte scolaire', *Canadian Journal of Education/Revue canadienne de l'éducation* pp. 144–157.

Vygotsky, L. S. (1964), 'Thought and language', *Annals of Dyslexia* **14**(1), 97–98.

Vygotsky, L. S. (1980), *Mind in society: The development of higher psychological processes*, Harvard university press.

9 APPENDIX

9.1 Interface

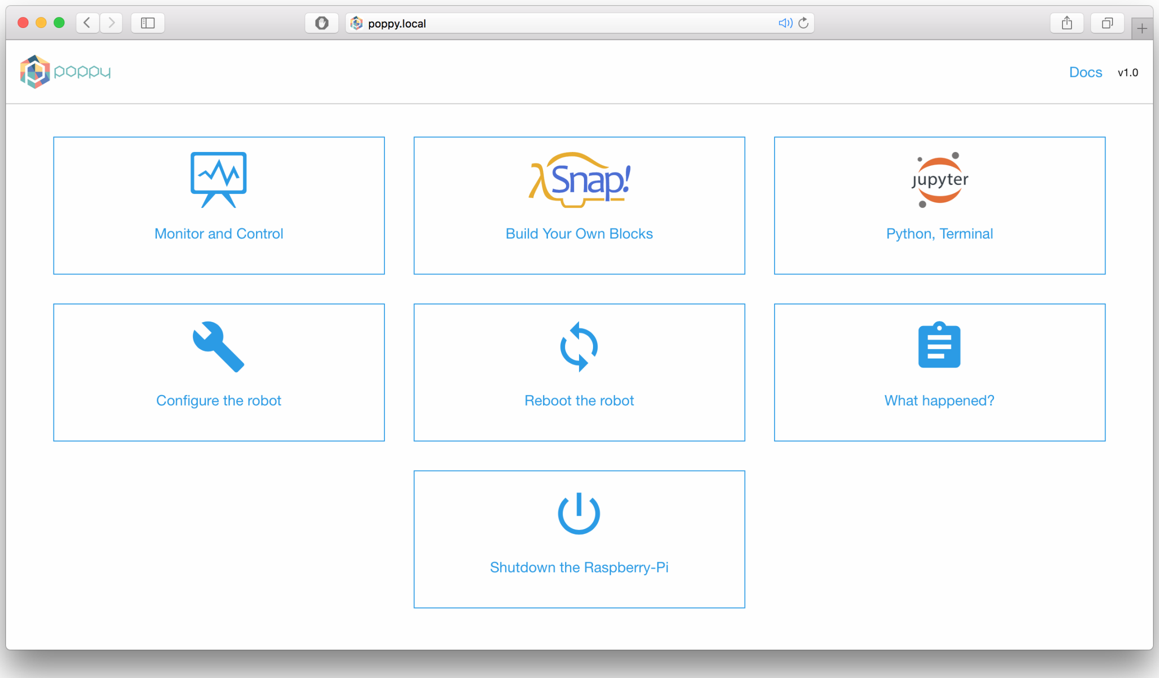


Figure 20: Poppy Interface

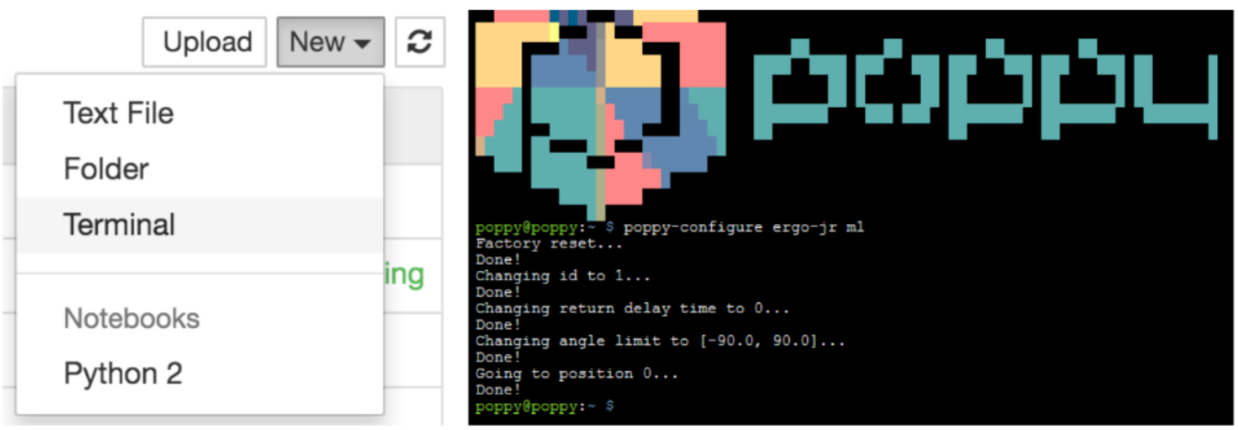


Figure 21: Terminal poppy

9.2 Codes

poppy_dragster_mini.json :
https://github.com/tgll/poppy-dragster-mini/blob/master/software/poppy_dragster_mini/configuration/poppy_dragster_mini.json

```
{
  "controllers": {
    "my_dxl_controller": {
      "sync_read": true,
      "protocol": 2,
      "attached_motors": [
        "joint",
        "wheel"
      ],
      "port": "/dev/ttyAMA0",
      "syncloop": "LightDxlController"
    }
  },
  "motorgroups": {
    "joint": [
      "m1",
      "m2",
      "m3",
      "m4"
    ],
    "wheel": [
      "w1",
      "w2"
    ]
  },
  "motors": {
    "m1": {
      "offset": 0.0,
      "type": "XL-320",
      "id": 1,
      "angle_limit": [
        -90.0,
        90.0
      ],
      "orientation": "direct"
    },
    "m2": {
      "offset": 0.0,
      "type": "XL-320",
      "id": 2,
      "angle_limit": [
        -90.0,
        90.0
      ],
      "orientation": "indirect"
    },
    "m3": {
      "offset": 0.0,
      "type": "XL-320",
      "id": 3,
```

```
      "angle_limit": [
        -90.0,
        90.0
      ],
      "orientation": "indirect"
    },
    "m4": {
      "offset": 0.0,
      "type": "XL-320",
      "id": 4,
      "angle_limit": [
        -90.0,
        90.0
      ],
      "orientation": "direct"
    },
    "w1": {
      "offset": 0.0,
      "type": "XL-320",
      "id": 5,
      "wheel_mode" : true,
      "orientation": "direct"
    },
    "w2": {
      "offset": 0.0,
      "type": "XL-320",
      "id": 6,
      "wheel_mode" : true,
      "orientation": "indirect"
    }
  },
  "sensors": {
    "camera": {
      "type": "OpenCVCamera",
      "index": -1,
      "fps": 20.0,
      "resolution": [640, 480]
    },
    "marker_detector": {
      "type": "MarkerDetector",
      "cameras": ["camera"],
      "freq": 1.0,
      "need_robot": true
    }
  }
}
```

dxlconfig.py :
https://github.com/flowersteam/pypot/blob/master/pypot/tools/dxlconfig.py

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

from __future__ import print_function

"""
Dynamixel Motor configuration.
```

```
First, it runs a factory reset. And then apply the configuration given as parameter.
*** WARNING: Make sure only motor is connected to the bus! Otherwise they will ALL be reset to their factory settings. ***
Examples:
dxlconfig --type=MX-28 --id=23 --angle-limit=(-100, 100) --zeroposition --port=/dev/ttyAMA0
dxlconfig --help
"""

import sys
import time

from argparse import ArgumentParser, ArgumentDefaultsHelpFormatter

from pypot.dynamixel.conversion import dynamixelModels
from pypot.dynamixel import DxlIO, Dxl320IO, get_available_ports
from pypot.utils import flushed_print as print


def check(pred, msg):
    if not pred:
        print(msg)
        print('Exiting now...')
        sys.exit(1)


def main():
    available_ports = get_available_ports()
    default_port = available_ports[0] if available_ports else None

    parser = ArgumentParser(description='Configuration tool for dynamixel motors '
                                        'WARNING: ONLY ONE MOTOR SHOULD BE '
                                        'CONNECTED TO THE BUS WHEN CONFIGURING!',
                            formatter_class=ArgumentDefaultsHelpFormatter)

    parser.add_argument('--id', type=int, required=True,
                        help='Chosen motor id.')
    parser.add_argument('--type', type=str, required=True,
                        choices=dynamixelModels.values(),
                        help='Type of the motor to configure.')
    parser.add_argument('--port', type=str,
                        choices=available_ports, default=default_port,
                        help='Serial port connected to the motor.')
    parser.add_argument('--return-delay-time', type=int,
                        help='Set new return delay time.')
    parser.add_argument('--angle-limit', type=float, nargs=2,
                        help='Set new angle limit.')
    parser.add_argument('--goto-zero', action='store_true',
                        help='Go to zero position after configuring the motor')
    parser.add_argument('--wheel-mode', type=bool, default=False,
                        help='Set wheel mode.')

    args = parser.parse_args()

    check(1 <= args.id <= 253,
```

```
            'Motor id must be in range [1:253]')

    check(available_ports,
            'Could not find an available serial port!')

    protocol = 2 if args.type in 'XL-320' else 1
    DxlIOPort = DxlIO if protocol == 1 else Dxl320IO

    # Factory Reset
    print('Factory reset...')
    if protocol == 1:
        for br in [57600, 1000000]:
            with DxlIO(args.port, baudrate=br) as io:
                io.factory_reset()
    else:
        with Dxl320IO(args.port, baudrate=1000000, timeout=0.01) as io:
            io.factory_reset(ids=range(253))
    print('Done!')

    factory_baudrate = 57600 if args.type.startswith('MX') else 1000000

    # Wait for the motor to "reboot..."
    for _ in range(10):
        with DxlIOPort(args.port, baudrate=factory_baudrate) as io:
            if io.ping(1):
                break

            time.sleep(.5)
    else:
        print('Could not communicate with the motor...')
        print('Make sure one (and only one) is connected and try again')
        sys.exit(1)

    # Switch to 1M bauds
    if args.type.startswith('MX') or args.type.startswith('SR'):
        print('Changing to 1M bauds...')
        with DxlIO(args.port, baudrate=factory_baudrate) as io:
            io.change_baudrate({1: 1000000})

        time.sleep(.5)
        print('Done!')

    # Change id
    print('Changing id to {}...'.format(args.id))
    if args.id != 1:
        with DxlIOPort(args.port) as io:
            io.change_id({1: args.id})

            time.sleep(.5)
            check(io.ping(args.id),
                  'Could not change id to {}'.format(args.id))
    print('Done!')

    # Set return delay time
    if args.return_delay_time is not None:
        print('Changing return delay time to {}...'.format(args.return_delay_time)
```

```python
)
        with DxlIOPort(args.port) as io:
            io.set_return_delay_time({args.id: args.return_delay_time})

            time.sleep(.5)
            check(io.get_return_delay_time([args.id])[0] == args.return_delay_time,
                  'Could not set return delay time to {}'.format(args.return_delay_time))
        print('Done!')

    # Set Angle Limit
    if args.angle_limit is not None:
        print('Changing angle limit to {}...'.format(args.angle_limit))
        with DxlIOPort(args.port) as io:
            io.set_angle_limit({args.id: args.angle_limit})

            time.sleep(.5)
            check(all(map(lambda p1, p2: abs(p1 - p2) < 1.,
                          io.get_angle_limit([args.id])[0],
                          args.angle_limit)),
                  'Could not change angle limit to {}'.format(args.angle_limit))
        print('Done!')

    # Set wheel Mode
    if args.wheel_mode == True:
        print('Set wheel mode')
        with DxlIOPort(args.port) as io:
            io.set_control_mode({args.id :'wheel'})

            time.sleep(.5)
            check(io.get_control_mode([args.id])[0] == 'wheel',
                  'Could not set wheel Mode')
            for speed in range(0,500,10):
                io.set_moving_speed({args.id:speed})
                time.sleep(.1)
            for speed in range(0,525,25):
                io.set_moving_speed({args.id:500-speed})
                time.sleep(.1)
        print('Done!')


    # GOTO ZERO
    if args.goto_zero:
        print('Going to position 0...')
        with DxlIOPort(args.port) as io:
            io.set_moving_speed({args.id: 100.0})
            io.set_goal_position({args.id: 0.0})

            time.sleep(2.0)
            check(abs(io.get_present_position([args.id])[0]) < 5,
                  'Could not go to 0 position')

        print('Done!')
```

```python
if __name__ == '__main__':
    main()
```

configure_utility.py :
https://github.com/poppy-project/pypot/blob/master/pypot/creatures/configure_utility.py

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

from __future__ import print_function

"""
Poppy configuration tools
Examples:
* poppy-configure ergo-jr m2
"""
import sys

from argparse import ArgumentParser, ArgumentDefaultsHelpFormatter
from subprocess import call

from pypot.creatures import installed_poppy_creatures
from pypot.utils import flushed_print as print


def find_port_for_motor(config, motor):
    for bus in config['controllers'].values():
        motor_on_bus = sum([config['motorgroups'][n]
                            for n in bus['attached_motors']], [])
        if motor in motor_on_bus:
            return bus['port']

    raise ValueError('Something must be wrong in your configuration file. '
                     'Could not find bus for motor {}'.format(motor))


def main():
    robots = [c.replace('poppy-', '') for c in installed_poppy_creatures]

    parser = ArgumentParser(description='Configuration tool for Poppy robots ',
                            formatter_class=ArgumentDefaultsHelpFormatter)

    parser.add_argument('robot', type=str, choices=robots,
                        help='Robot used.')

    parser.add_argument('motor', type=str,
                        help='Name of the motor to configure.')

    args = parser.parse_args()

    RobotCls = installed_poppy_creatures['poppy-{}'.format(args.robot)]
    c = RobotCls.default_config

    if args.motor not in c['motors']:
        print('"{}" is not a motor of "{}"! '
              'possibilities={}'.format(args.motor, args.robot,
                                        sorted(c['motors'].keys())))
```

```
        print('Exiting now...')
        sys.exit(1)

    motor_config = c['motors'][args.motor]

    args = [
        '--id', motor_config['id'],
        '--type', motor_config['type'],
        '--port', find_port_for_motor(c, args.motor),
        '--return-delay-time', 0
    ]

    if 'wheel_mode' in motor_config.keys():
        args.extend(('--wheel-mode', motor_config['wheel_mode']))
    else:
        args.extend(('--angle-limit',motor_config['angle_limit'][0],motor_config['angle_limit'][1],
                     '--goto-zero'))

    call(['dxl-config'] + map(str, args))


if __name__ == '__main__':
    main()
```