

2025年臺灣國際科學展覽會 優勝作品專輯

作品編號	190029
參展科別	電腦科學與資訊工程
作品名稱	Project M. I. R. A. S
得獎獎項	一等獎

就讀學校	Lycée Michel Rodange
作者姓名	Maxime Buck

作者照片



FEBRUARY 4, 2024

Project M.I.R.A.S

A MODULAR SELF ASSEMBLING ROBOTIC
SYSTEM LEVERAGING ARTIFICIAL INTELLIGENCE

TABLE OF CONTENTS

1. Introduction	
1.1 Short project summary	2
1.2 Background	2
1.3 What are modular robots?	3
1.4 Motivation	4
2. Project description	
2.1 Idea & Brainstorming	5
2.2 Requirements of a worker module	5
2.3 Key concept	6
2.4 Tools and Equipment	7
3. Project realization	
3.1 Theoretical proof of the worker module	
A. Servo motors.	7
B. Battery charging	8
C. Circuit diagram	8
D. 3D model	9
3.2 Using deep reinforcement learning to learn walking	10
3.3 Base simulations	11
3.4 Voxelization Algorithm	13
3.5 Centralized mass control & communication	16
3.6 Simulation in VR with hand tracking & passthrough	17
3.7 Testing & Problems	19
3.8 Optimization	21
3.9 Results	22
4. Extras	
4.2 Conclusion	23
4.3 Future plans	23
5. Appendix	
5.1 References	24
5.2 Image sources	25

1. Introduction

1.1 Short project summary

My project involves the conceptualization and development of an innovative approach to modular self-assembling robotic systems. Through its ability to form any complex configuration, the system is highly adaptable to various scenarios and environments.

Before delving deeper into the details of my project, I will provide an overview of my background and motivations.

1.2 Background

Ever since I first watched the movie "Big Hero 6", I felt amazed by the applications of the so called "microbots". From that point on, it made me always wonder what would be possible in the real world. When I did the research, I stumbled upon this field of modular robotics. Initially, I was unsure whether to embark on a project focused on electronics and robotics due to my background in programming.



Main character Hiro showcasing the «Microbots»

On the other side, this year gave me a chance to see the incredible performances of various projects at different science expos. Besides, I took part in the program of CANSAT LU and learned a lot during it, such as microchips, the control of miniature robotics, and the sensors of it. Finally, at school, I took the option Electronics where we dig into similar topics. With this accumulated knowledge and experience I felt confident enough to start this project.



Cansat launch station



My Team's Can

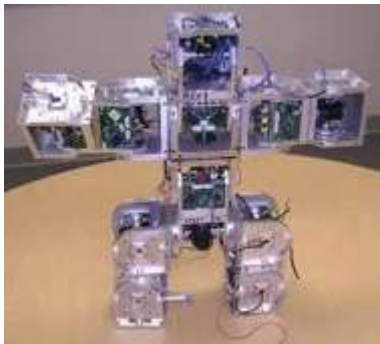


School project : Audio amplifier

1.3 What are modular robots?

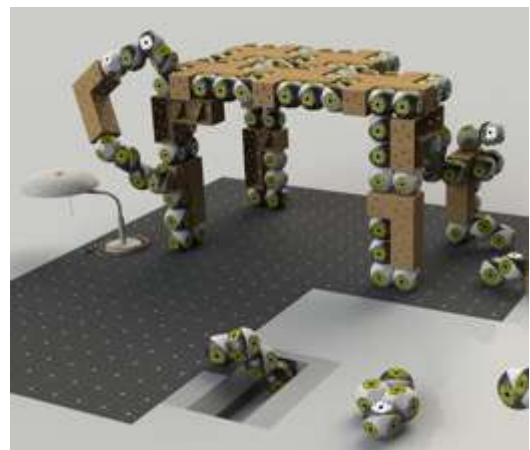
Self-Assembling Modular Robots: These are a type of swarm robotics where individual modules assemble into various configurations to accomplish tasks.

- Each module has limited capabilities (degrees of freedom, mobility), but when assembled, they create a system with extensive capabilities.
- Assembled modular robots offer adaptability in any environments, where normal robots are constructed for only a certain task.
- Replacement of failed modules is simple, with a new module taking the place of the failed one in the configuration.
- Automatically self-assembling eliminates the need for prolonged human interaction and enables the formation of complex structures.



Use cases of modular robotic systems:

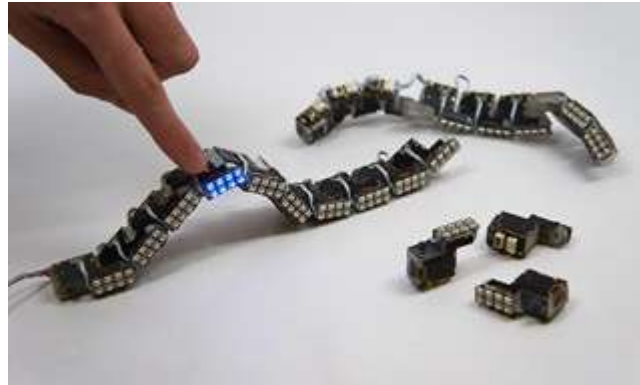
- Capable of carrying objects and equipped with special modules for performing all kinds of actions on external objects.
- Can be controlled to access dangerous environments (e.g., extreme temperatures, collapsed buildings, radioactive areas).
- Diagnose problems and reorganize themselves to provide solutions to the tasks they face.
- Addressing labor shortages:
 - In construction by augmenting or replacing traditional construction workers.
 - In manufacturing, reducing the need for manual labor.
 - Adaptable furniture design, allowing for dynamic changes in furniture configuration based on user needs.



1.4 Motivation

Current Problem:

Current robots are built for specific tasks in which they excel but lack adaptability for any other tasks. They cannot adjust to changes in their environment, rendering them unsuitable for unknown environments such as exploring a new planet.



The Solution – Modular Robots, but:

While modular robots offer a promising solution, there are three main problems with many current modular robotics solutions:

1. Modular robots are limited in tasks and remain bulky despite efforts to minimize size.
 - Because each bot is designed as an individual robot capable of computing, localizing, moving, and housing a battery, leading to bulkiness.
2. Each module has limited degrees of freedom and cannot move independently.
 - This limitation restricts assembly possibilities and the ability to adapt to diverse environments.
 - Also, if a robot disconnects or gets lost, he will be unable to get back to the colony.
3. Modular robot projects fail to realize their full potential due to lack the inability to show promising simulations. Often only trying to use a tiny number of real bots which can't show their true potential leading to a possible lack of funding.

2. Project description

2.1 Idea & Brainstorming

To address the challenges outlined earlier, I have several solutions:

1. I propose a unique approach that aims to significantly reduce the size of each module by eliminating the need for individual components. For instance, huge batteries, multiple sensors as well as big computing necessity. Drawing inspiration from the organizational structure of bee swarms or other insect colonies, I envision a system with a central coordinator and numerous worker modules. In this project, my primary focus lies on developing the foundational worker module, which forms the backbone of the entire system.



2. The core objective is to design a self-assembling modular robot system characterized by modules possessing high degrees of freedom. These modules will exhibit efficient mobility across various surfaces and possess autonomous movement capabilities using the potential of artificial intelligence.

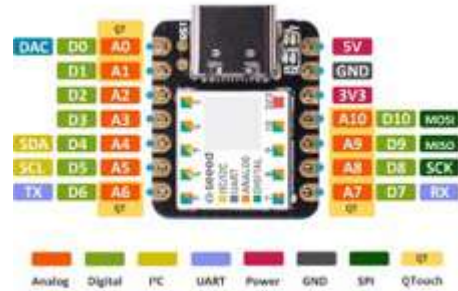
3. Given the immense complexity of this project, coupled with my primary passion and motivation being in programming, I plan to initially focus on establishing the feasibility of the worker bot's construction through theoretical frameworks. This approach allows me to concentrate on key aspects such as communication, simulation, and the organizational structure of modular robotics systems, ensuring a solid foundation before moving into physical prototyping.

2.2 Requirements of a worker module



Stable Attachment – Integrating small neodymium permanent magnets with electromagnets enables the creation of a custom energize-to-release mechanism. This design ensures that modules maintain a strong connection without the need for continuous current. Turning the electromagnet on at any moment neutralizes the magnetic field of the magnet releasing the modules from each other.

Receiving and Executing Commands – Each bot will be equipped with an Arduino microcontroller. Currently, the smallest microcontroller with Bluetooth capability is the "Seeed Studio XIAO SAMD21" it can be programmed via USB C. Additionally, it provides a total of 11 GPIO ports, which is precisely the number required for the task.



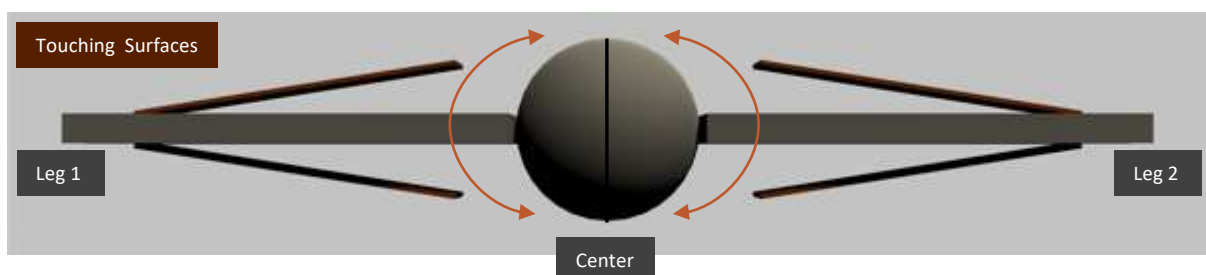
Localizing – The robots begin in a predefined structure where all positions are already known. Upon completion of their tasks, they will return to this initial structure. Positional data during operation can be derived from the known movements of each module. Additionally, to correct errors some could be equipped with a gyroscope module like the MPU6500.

Surroundings Perception – This will be handled by the mother bot that will check the surroundings before and during the performing of a task. Specific sensors like LIDAR or cameras could be used. As said before the focus will remain on the worker bot, and during simulations, it will be assumed that the bots possess awareness of their surroundings.

Power supply – The bots will rely on small batteries that will be continuously charged while attached to the system. The plan involves allowing current to flow through all bots in the system, creating a grid connected to a main power source. The small batteries should suffice so that in case the robot is disconnected it can walk back to the system.

Movement – The bot needs to be able to freely move in 3-dimensional space using a type of hinge motion comparable to walking already used by different modular projects. This will be possible through a formation of 3 tiny individual servo motors.

2.3 Key concept



The key concept shown in this picture shows a worker module characterized by its central core housing all components, along with two legs. Each leg possesses the capability to rotate within a two-dimensional plane. Additionally, the central core is divided into two halves, allowing rotation around a central axis.

2.4 Tools and Equipment

	Tools	Name
1.	Programming languages	(Circuit)Python, C#
2.	Lab	Home, school, makerspace
3.	Planning application	Jira
4.	VR headset with hand-tracking capability	Oculus Quest 2 and 3
5.	Physics accurate engine with VR support	Unity
6.	Code editor	Visual Studio
7.	Computer	Mac
8.	A 3D modelling software	Shapr3D
9.	Circuit diagram software	Flux
10.	Library for machine learning	TensorFlow
11.	3D slicing software for 3d printing	UltiMaker Cura

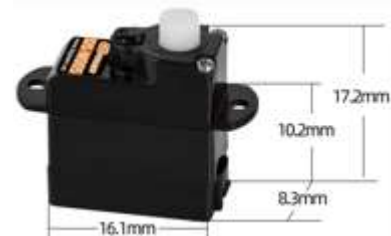
3. Project realization

3.1 Theoretical proof

The size of a single module plays a great role in defining the complexity and potential of a larger structure capabilities. Keeping this into view, the size of the module becomes a major consideration for these next steps.

A. Servo motors

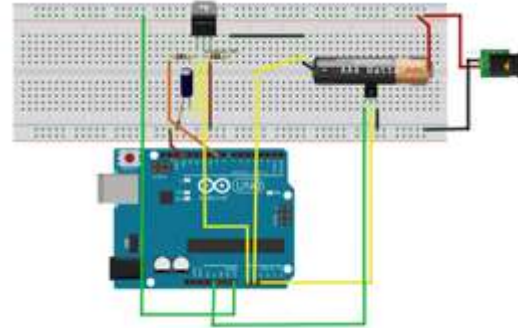
In my quest for compact electric motors, I found the DS-M005, renowned as one of the world's smallest servo motors. While this motor offers exceptional size advantages, it necessitates the design of a custom gearbox to increase torque, as my project prioritizes torque over speed. Additionally, since these servos exhibit low backward torque when powered off, I incorporated a worm gear mechanism. This addition ensures stability by preventing any unintended motion in the legs when external forces are applied, thereby enhancing the system's robustness and reliability.



Operating Voltage	3.7~4.2V DC
No Load Speed	≤0.08sec/60° at 3.7V (STD.) ≤0.06sec/60° at 4.2V (STD.)
Stall Current	≤320mA at 3.7V (STD.) ≤350mA at 4.2V (STD.)
Max. Torque	≥0.25kgf·cm at 3.7V (STD.) ≥0.30kgf·cm at 4.2V (STD.)
Pulse Width Range	500~2500μs
Operating Travel Angle	90°±10°
Max. Operating Travel Angle	180°±10°
Mechanical Limit Angle	360°
Weight	2.2±0.2g

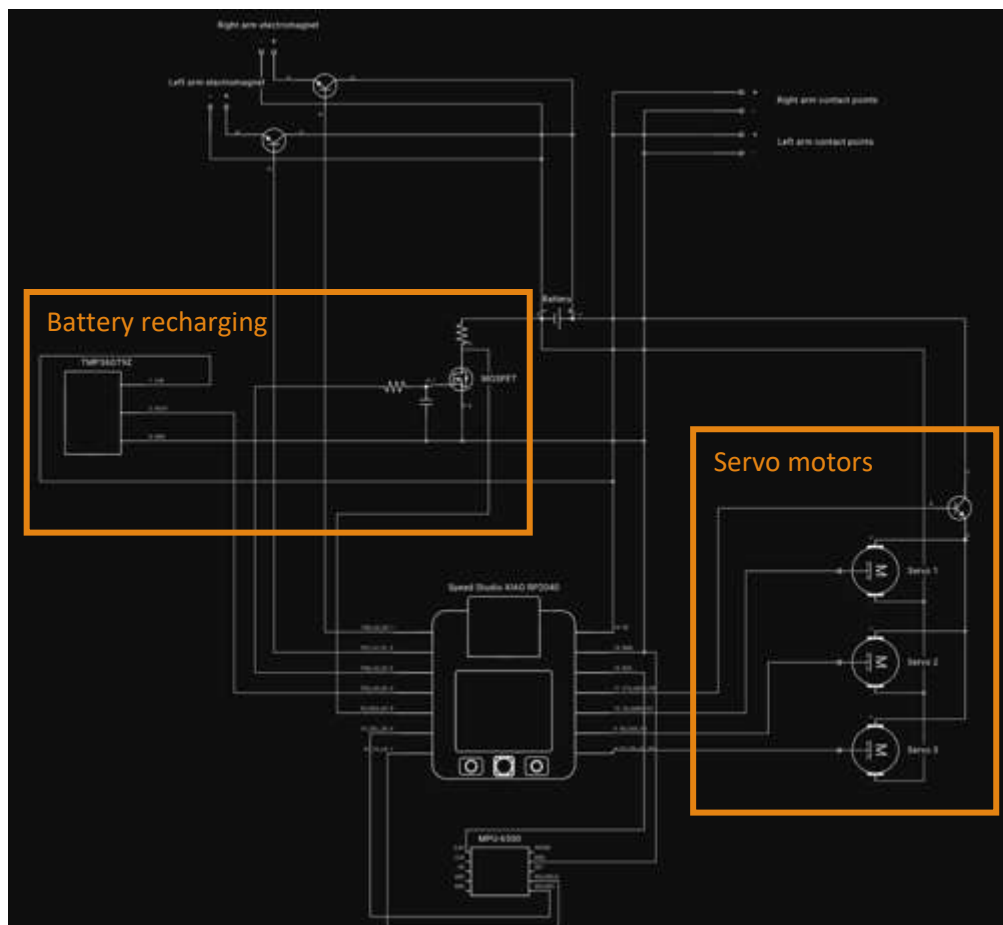
B. battery charging

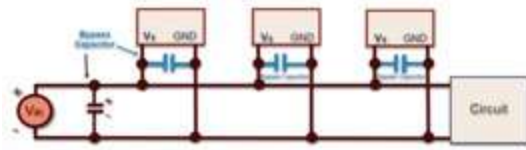
When the robot is connected to the grid, electrical current flows through contact points, which then needs to be efficiently converted and stored inside the battery. This process is made possible using a thermal sensor and a MOSFET. The thermal sensor plays a critical role in monitoring the temperature to prevent overheating, while the MOSFET is key in controlling the flow of electricity, ensuring that the charging is done safely and optimally, adjusting the charge as necessary to protect the battery's longevity and performance.



C. circuit diagram

I designed the entire circuit diagram using Flux, a free, online software equipped with an extensive library of components. I chose Flux because of its user-friendly interface and its integrated AI assistant, which helps identify potential errors—a valuable feature for someone like me, who has not had formal training in electronics. Despite this, I recognize that the circuit likely contains issues, such as missing components like transformers, that need to be addressed. For now, however, the primary focus of the project remains on establishing theoretical feasibility and developing robust control algorithms.



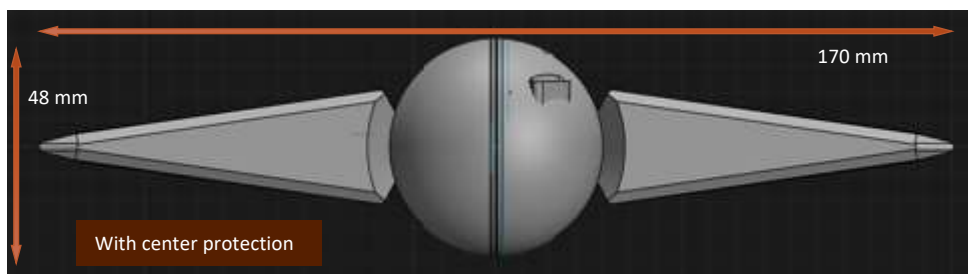


Though not illustrated in the diagram, multiple bypass capacitors are crucial for stable operation and minimizing noise interference. Utilizing various types of capacitors in parallel, such as ceramic and electrolytic, enables effective filtering of power supply fluctuations, maintaining a clean and stable voltage for sensitive components like microcontrollers. Strategically placing these capacitors close to the power pins maximizes their effectiveness in filtering noise, particularly significant as even the wire itself can introduce noise.

To control the electromagnets of the metal plates I only had 2 GPIO pins of the microcontroller in reserve as such the 2 sides of a leg will always be controlled together. This is achieved using 2 transistors which will prevent current from passing. The controller knowing if any plate is connected can regulate or turn of the charging amount thanks to the MOSFET.

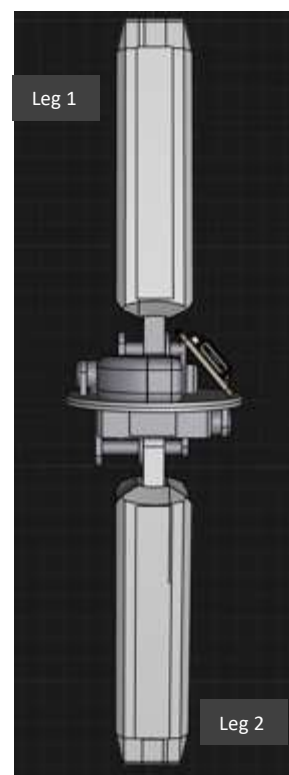
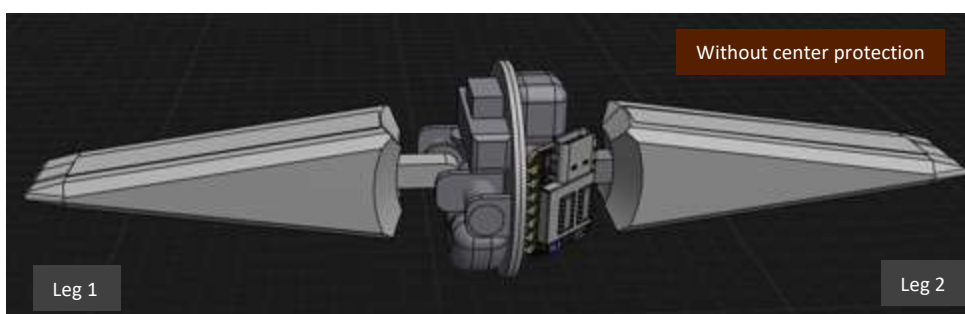
D. 3D model

First, I calculated the minimum volume which would allow for the possibility of the center sphere to have a radius of only 24 mm, and I aimed to get as close to this as possible. The limiting factors were the length of the motors, and I had to ensure enough space for cabling. Finally, I was left with the remaining space, which I used to maximize for the battery volume.



These are millimeter accurate 3D models made in Shapr3D considering the real size of all components in use.

This image depicts a partially detailed 3D model showing the two



halves of the center. Some cables need to pass through the center bearing. However, this presents no issue since the center bearing never needs to rotate more than 180 degrees in either direction.

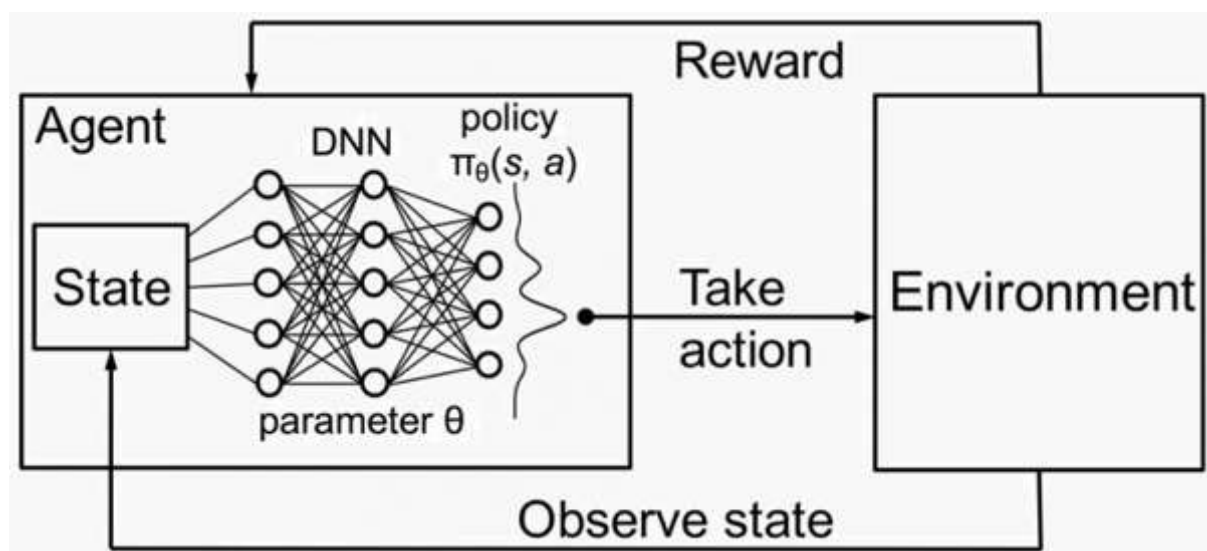
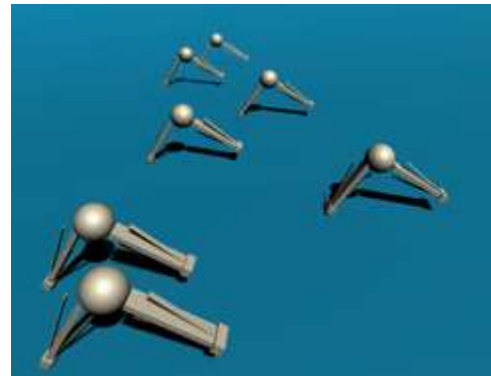
3.2 Using deep reinforcement learning to learn walking

Initial Approach and Transition to AI

Initially, I attempted to implement walking through hardcoded movements, establishing a baseline for the robot's performance. However, this method resulted in slow movement speeds, prompting me to explore more dynamic solutions. From September to October, I immersed myself in machine learning, drawing upon my experience with creating an AI module for the Chrome Dino game. Leveraging



Jupyter notebooks facilitated training on Microsoft's Azure cloud computing service, while tools like TensorFlow, OpenCV, Keras, and OpenAI Gym enabled rapid development thanks to their extensive documentation. The success of the Chrome Dino AI project, which achieved near-endless running, served as a foundational step for developing the walking AI.



Neural Network Design

Inputs: The neural network was designed to process inputs such as the robot's center position and the current angles of its left and right legs. This data is crucial for determining the robot's stance and orientation, enabling precise control over its movements.

Outputs: The network generates outputs that dictate the desired angles for the left and right leg motors, thus controlling the robot's gait and balance during walking.

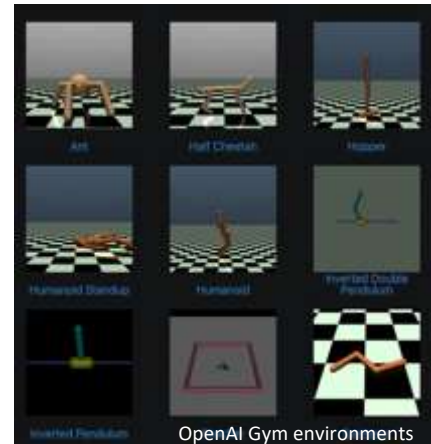
Reward System and Performance Metrics

Reward System: A reward-based system was implemented to encourage forward progression. The robot receives higher rewards for increasing its x-position, incentivizing efficient forward movement.

Performance Enhancement: To further motivate speed and efficiency, a 30-second time limit was imposed on walking tasks. This constraint forces the AI to optimize both the speed and effectiveness of the robot's walking pattern.

Training the Model

I created a custom environment within OpenAI Gym, which approximates real-world conditions with a simple 3D model while allowing for extensive simulation and training. The agent's neural network model, constructed using Keras, processes inputs and predicts outputs to guide the robot's movements. This setup enables iterative learning and adjustment based on the robot's performance and the rewards it accumulates. Once I was happy with the module, I executed it with the real 3D model in unity.



Future plans

While the current model enables two-dimensional walking, it does not account for center rotation, which would allow for three-dimensional movement. Integrating this capability requires developing new environments and training goals, as well as adapting the AI to handle more complex terrains beyond flat surfaces. Future enhancements will also explore efficiency improvements for navigating uneven terrains, expanding the robot's operational versatility.

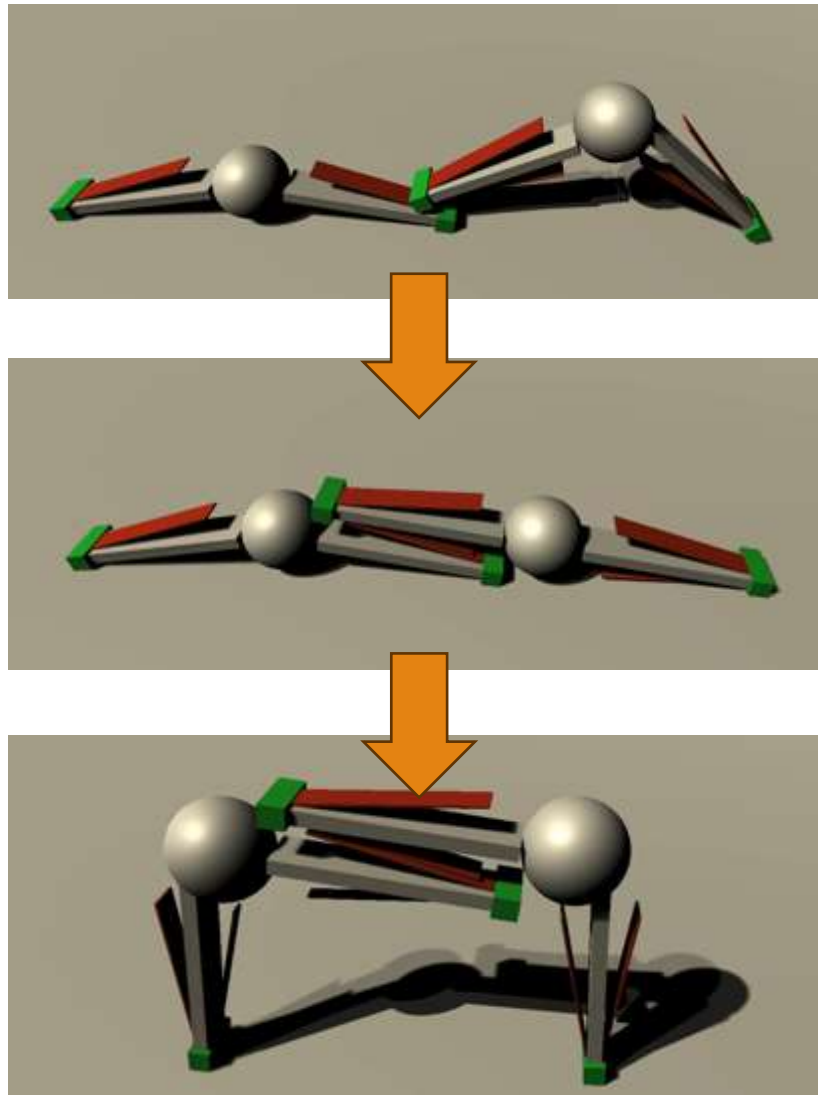
Encountered Challenges

One of the initial challenges was the robot's tendency to fall sideways under excessive force. This issue was resolved by enlarging the legs, a modification that doesn't significantly impact the future system's complexity.

3.3 Base simulations

The foundation of the entire system relies on the attaching operation between two modules. This fundamental process consists of the following steps:

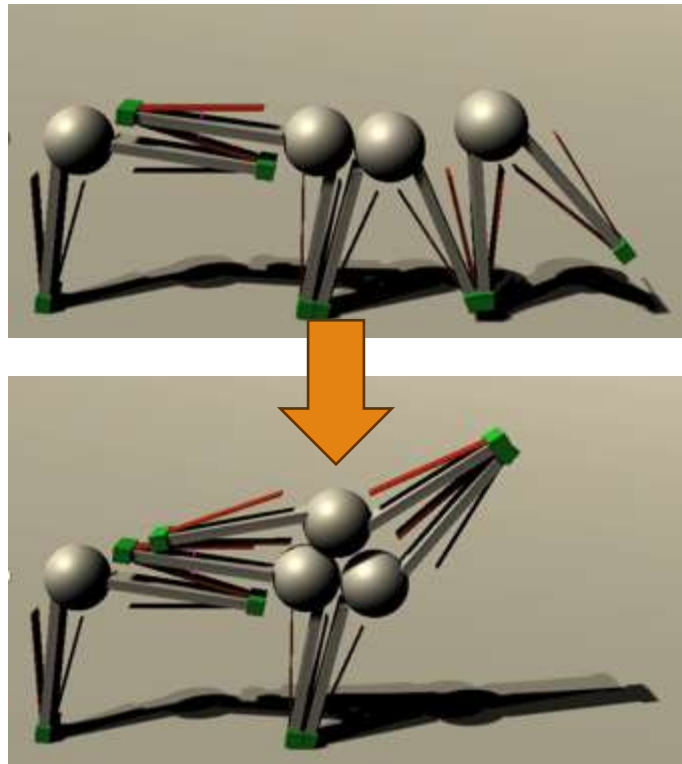
1. **Positioning:** One module moves and aligns itself over the target module.
2. **Attachment Preparation:** The module lowers itself onto the target module, ensuring precise alignment.
3. **Activation:** Magnets are activated, securely connecting the two modules and enabling them to hold firmly onto each other.



Another critical aspect of the system is the ability for modules to climb and stack on one another. This operation requires the assistance of an intermediate module to facilitate the elevation process. The steps are as follows:

1. **Connection:** The climbing module attaches itself to the intermediate module.
2. **Elevation:** The intermediate module lifts the climbing module above the target module, positioning it for attachment.
3. **Final Connection:** Once elevated, the climbing module connects to the target module.
4. **Disconnection:** The intermediate module detaches, completing the climbing process.

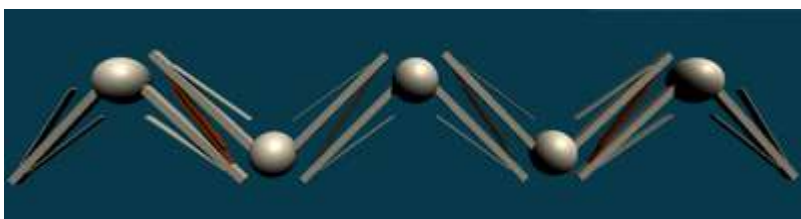
This simulation is essential for constructing more complex, multi-layered structures and demonstrates the system's versatility and scalability.



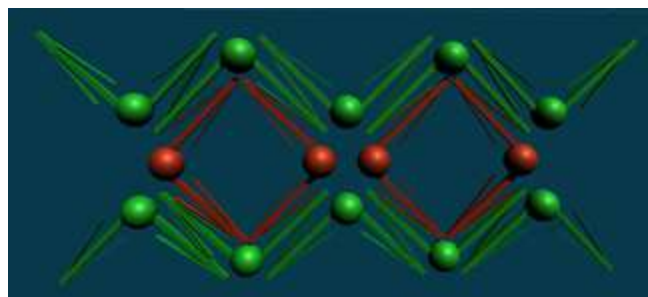
3.3 Voxelization Algorithms

Basic Structures

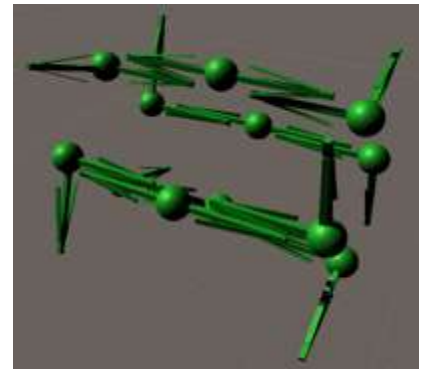
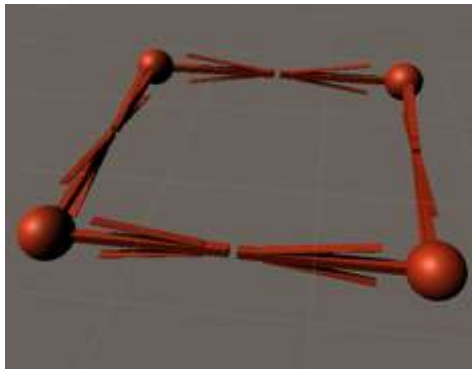
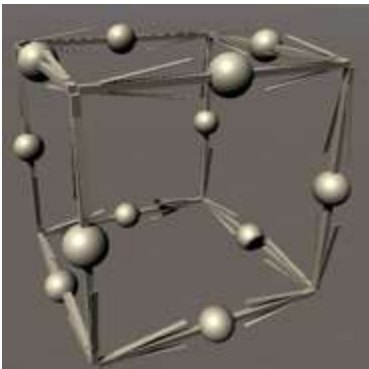
At first, I started with finding basic structures which could be stacked to make a bigger as show the following pictures. I am sure that with time I will find many others probably more stable and more efficient.



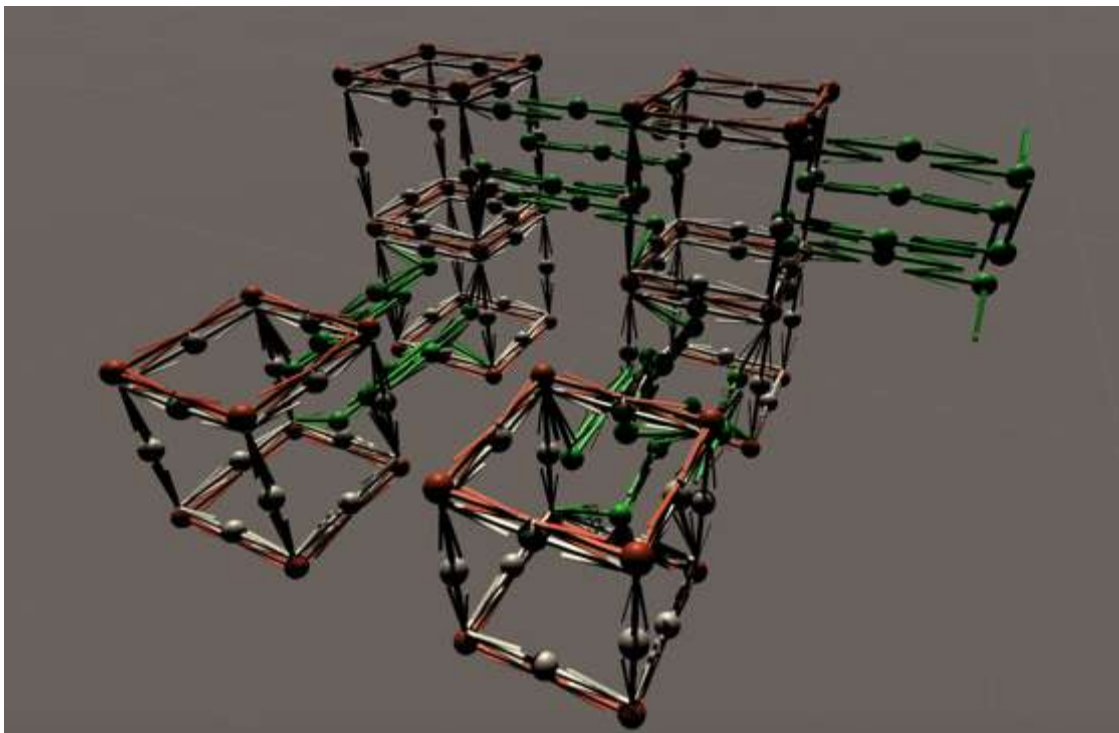
1D strip that can go on forever consisting of a sequence of flipped bots with both legs at a 45-degree angle.



2D structure that can go on forever consisting of (green) the strips from before and (red) a part holding 2 strips together.



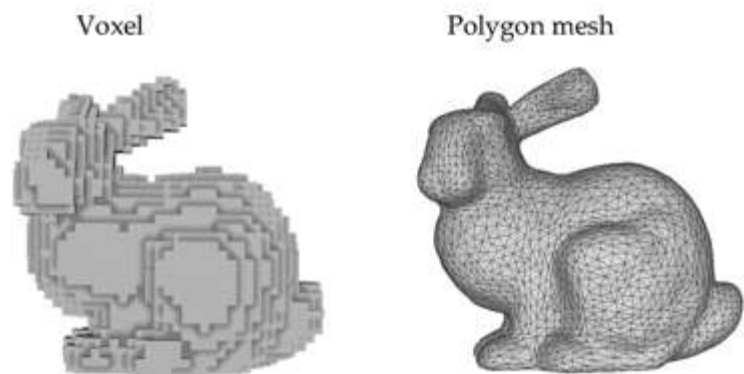
Going for 3 dimensional structures I tried to recreate basic geometry. Starting with a cube here in gray (only the outline) which wouldn't hold as no plates are touching. The red construct fixes this by holding 2 panels while the green connects the rest and booth acts as a bridge to the next Cube.



Combining those 3 elements it is possible to create a rigid structure that can stretch in all dimensions. This would for now be the ground structure of the system on which could be attached more specific structures.

Voxelization algorithms

Voxelization algorithms convert geometric objects into a discrete grid of voxels (3D pixels), capturing the 3D structure in a regular, grid-based format. It is like if you forced a mesh to conform to a 3D grid. Unlike polygons that define surfaces through vertices and edges, voxels represent volume, making them ideal for applications requiring volumetric analysis or rendering.

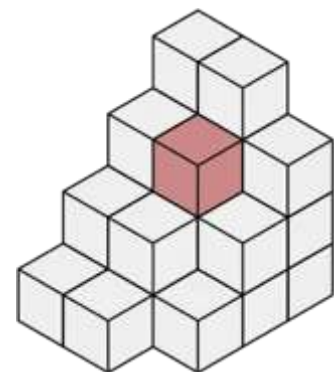


The key aspect of those algorithm is Occupancy Evaluation which is determining whether a voxel is inside, outside, or on the boundary of the model. The detailing of the voxelated object can easily be changed by adapting the size of a single voxel of course this will make the program more resource intensive.

The most common methods include:

Surface Voxelization: Focuses on converting the model's surface into voxels. Techniques such as ray casting or rasterization are used to determine voxel occupancy based on the intersection with the model's surfaces.

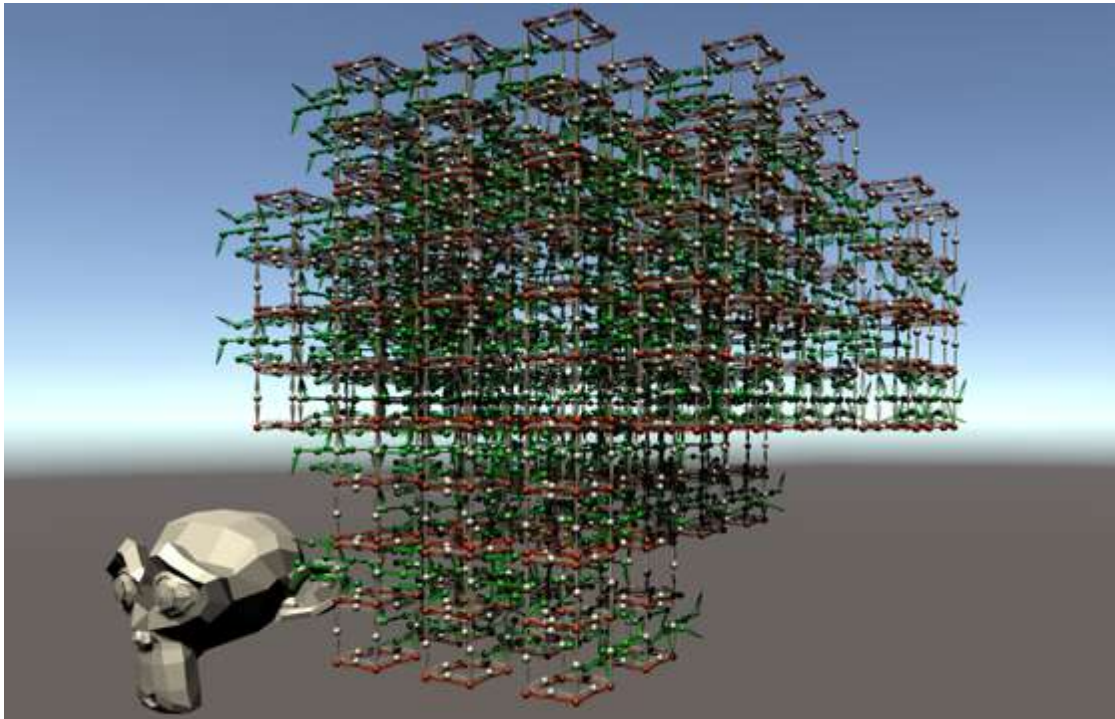
Solid Voxelization: Aims to fill the interior of the model with voxels, often using algorithms like flood fill to ensure all internal spaces are voxelated.



Feature-Based Voxelization: Targets the preservation of model features, such as edges and corners, by adjusting voxel sizes or using adaptive sampling to maintain detail.



Here I am using Suzanne the blender monkey for my test. The algorithm creates a list which then spawns a block of modules for each voxel. This voxelization tactic allows me to replicate any 3D model I feed it. As such it becomes easier to mass control the worker bots. For example, act as a support pillar to protect a collapsing bridge or ancient ruin. It is also useful to have a 3D representation of an object your enterprise is currently working on or. The stability isn't always assured, and I propose to add support like the ones used when 3D printing.



Voxelization with different shapes

To achieve a more realistic look an approach from paper 3 explain the possibility to use a palette of objects and then the algorithm picks which object will fit best.

In my case this would only be important for the outer layer. To check if a certain voxel is part of the outer layer it's fairly simple by checking if it has 6 neighbors or not. But for now, this doesn't play a significant role other than aesthetics.



3.4 Centralized mass control & communication

To enable coordinated movement of the bots, mass control algorithms are essential. The headmaster, a nearby computer, undertakes all the necessary calculations and issues commands via Bluetooth, providing each module with instructions for the next rotation of its motors.

This process unfolds in three key steps:

- **Ground Structure Construction:** Similar to the methods employed by the ancient Egyptians when constructing the pyramids, the bots ascend from the sides and take their positions. Movement within the system is possible by sequential engagement and disengagement of the metal plates. If certain areas are too confined or require impossible rotations, additional worker modules can assist in advancing further.
- **Special Modules:** Specialized modules or tools are transported to their designated locations either by autonomously navigating through the structure or by being carried by multiple worker modules.
- **Constant Outer Layer Reorganization:** Throughout the task execution, there is ongoing reorganization of the outer layer to accommodate evolving requirements.



To ensure robustness and fault tolerance, the mass control system incorporates redundancy at various levels. This includes redundant Bluetooth transfer and in the event of a bot failure or communication loss, the system can quickly reconfigure and redistribute tasks to maintain overall functionality.

Communication

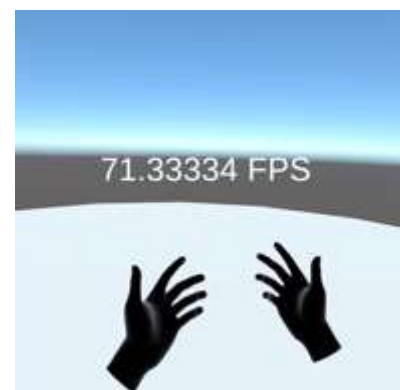
In developing a modular robotic system with potentially hundreds or thousands of worker modules, selecting an efficient communication method is crucial. Zigbee was selected for its superior scalability, low power consumption, and excellent mesh networking capabilities. Furthermore, the seeduino is directly compatible with this system without the need of external hardware. Zigbee supports up to 65,000 nodes, making it ideal for swarm-based applications where reliable communication across distributed devices is crucial. Compared to alternatives like Bluetooth LE, Wi-Fi, Sub-GHz RF, and LoRa, Zigbee offers the best balance of range, data rate, and power efficiency. Its ability to form self-healing, decentralized networks ensures robust performance, making it the optimal choice for this project.

3.5 Virtual reality simulation with hand tracking & passthrough

As in the film “big heroes 6” the microbots are controlled with their brain I wanted to try and achieve something similar.

Drawing from my experience developing a hand-tracking VR game last year, I intend to use this remarkable technology to interact with a simulated version of the system. This approach allows for prototyping and visualization without the need to physically construct any robots.

Unity simulations can also be run on VR headsets, albeit with some limitations. Therefore, my goal is to create an interactive environment for users to explore and become familiar with the system's capabilities. This approach emphasizes



aesthetics, enabling players to control the system using hand gestures and allowing it to take shape as desired.



Hand gestures or poses are defined by the position and rotation of various parts of the hand, such as fingertips and knuckles. Additionally, with the pass-through feature available in the new Meta Quest 3, the simulation can appear to take place in the user's actual room, enhancing the sense of realness.

Think of it as 3D modeling, but with countless tiny robots. Once enough robots are built, the same could potentially be realized in the real world.

Passthrough

Compared to VR games, where the environment often serves as a decorative backdrop to enhance immersion, the surroundings in this project play a critical functional role. Using a plain white room as the backdrop would not only fail to provide necessary context but could also evoke feelings of claustrophobia or anxiety due to the stark emptiness. Additionally, for a system that needs to interact with the real world, it's essential to visualize it within its actual environment to achieve a better understanding of its scale, functionality, and practicality.

To address these challenges, I implemented passthrough, enabling a seamless blend of the real and virtual worlds. This approach allows users to see their actual surroundings through the headset, with the virtual system overlaid on top. The integration provides a more intuitive and grounded experience, ensuring the system is viewed and evaluated in context.

The process begins with the VR headset generating a 3D mesh of the surrounding environment. This mesh acts as a rough approximation of real-world geometry, providing the system with essential spatial data. By combining this information with colored passthrough, users can view the system superimposed on their physical environment, making it easier to assess how the system interacts with its surroundings and how it would function in real-world scenarios.

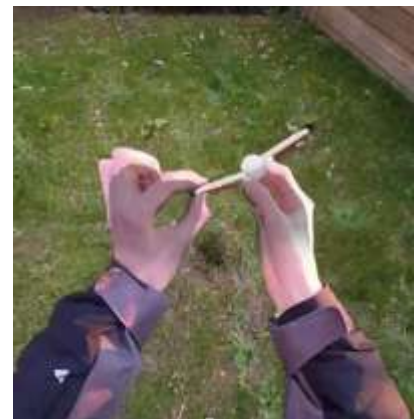


This implementation not only enhances usability but also contributes to more informed decision-making during development, as the system's operation can be better visualized and optimized in a realistic context.



The initial implementation focused on enabling users to inspect the 3D model in detail, with the ability to scale it up or down for close examination. Using intuitive hand gestures, such as pinching, the system detects when a user attempts to grab an object. When two hands are used to grab, the object dynamically scales based on the distance between the two grab points, offering precise and natural control over size adjustments.

Building on this foundation, I added functionality to move individual modules and attach them together. This feature significantly enhances the workflow, allowing users to quickly iterate through various structural designs. It also provides a deeper understanding of the assembly and mechanics of the structure within a fully interactive 3D space. The combination of scaling, manipulation, and modular interaction ensures a highly efficient and immersive design process, bridging the gap between virtual prototypes and practical implementations.



These images are screen captures from my application during testing in my garden. The gray hands depicted represent the headset's approximation of hand positions.

3.6 Testing & Problems

During the realization of the project, I encountered numerous challenges and encountered a multitude of programming bugs.

Bot's feet

One significant issue revolved around the design of the bots' feet. The question arose: should they be round, spiked, or cubic in shape? The primary requirement was for the bot to maintain stability when stationary. Hence, I opted for a design that incorporated various angles commonly encountered by the robot during both walking and standing, serving as the foundation of the structure.

Energy distribution

Another critical challenge centered on the distribution of current within the system. Modules further from the initial power source would risk getting not enough voltage. To address this, I propose three potential solutions that can be integrated:

Multiple Power Sources: Introducing multiple power sources throughout the system could mitigate voltage drop issues. For instance, additional power sources could be installed on the ceiling to supplement charging. However, this approach may increase complexity, which is not ideal.

Selective Charging: Implementing a system to deactivate charging for bots that are stationary once their batteries reach a certain threshold could help optimize power distribution. This ensures that energy is prioritized for active modules, enhancing overall efficiency.

Distributed Battery Modules: Introducing specific modules equipped with batteries distributed throughout the system could provide localized power sources. These battery modules can be strategically placed to supplement power where needed.

Center bearing

While testing the simulations, Unity's physics-accurate engine allowed me to assign breaking forces to the robot's joints for stress testing. Early in the process, I identified a significant issue: the center bearing could not adequately support any load. Even before reaching the breaking point, its rotation became compromised under strain, severely impacting functionality.

A possible solution could come from the advancements in 3D printing: in-place printing.

What is In-Place Printing?

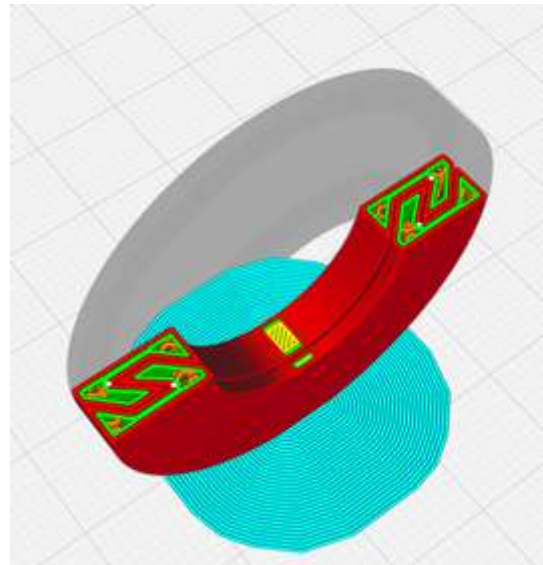
In-place printing refers to the process of designing and manufacturing moving parts or assemblies directly within a single print, without requiring post-assembly. The components, such as bearings or joints, are printed in their final operational configuration, with appropriate clearances designed into the model. This method eliminates the need for assembly, improves part integrity, and allows for the creation of complex mechanisms that would otherwise be difficult or impossible to produce.

Implementation

Using in-place printing, I designed a bearing that is fully integrated within itself, creating a highly resistant structure. This design ensures that the bearing can withstand significant loads without sacrificing rotational functionality.

The images provided show a cross-section of the bearing as visualized in Ultimaker Cura, the 3D printing software I am using.

Additionally, for future iterations where reduced friction is necessary, ball bearings could be incorporated into the design to further enhance performance and efficiency. This evolution would make the system even more robust and adaptable to a wider range of applications.



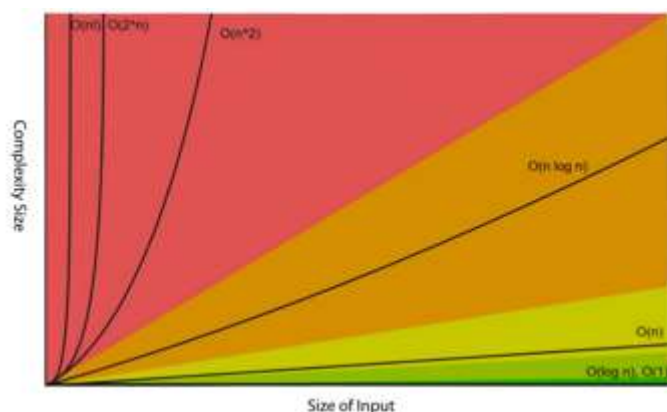
Center of gravity

During the initial design phase, I intentionally aimed to position the center of gravity (CoG) perfectly at the geometric center of the sphere. This was crucial because a balanced CoG ensures stability during motion and reduces the likelihood of the robot tipping or experiencing uneven stress on its joints. A well-centered CoG is particularly important for modular robots like this, as it allows for smoother and more predictable movement, enabling the robot to handle complex tasks with greater precision.

However, I did not initially account for the dynamic nature of the robot's complex movements. During operation, certain motions or configurations can cause the CoG to shift outside the intended center, potentially compromising stability and efficiency. This unforeseen challenge highlights the need for additional refinements in future iterations. Possible solutions may include redistributing internal components, adding counterweights, or implementing active stabilization systems to dynamically adjust the CoG during operation.

3.7 Optimization

Currently, parts of the voxelization and control algorithms show exponential complexity. This means that with each additional bot added, the entire script must undergo significantly more computation. This limitation could potentially restrict the scalability due to the intensive computational resources required.



However, through my involvement with the Luxembourg Tech School, I secured a student job during the summer vacation where I delved into the intricacies of quantum computing and various quantum algorithms. Among these, Shor's algorithm the most important one offers the theoretical capability to solve exponential problems much faster.



While quantum computing technology is still in the developmental stages, it holds immense promise for the future. Soon, using quantum computing may offer a viable solution for efficiently controlling thousands of bots.

3.8 Results

In the point 1.4 motivation I explain 3 problems of current modular systems.

1. Thanks to the motors the module has complete freedom and can walk on a surface in all directions. It can also form complex 3D structures where specific angles are required. More details on point 3.1.A and 3.2
2. Going for the theoretical proof allowed me to go faster towards the mass control algorithms. Additionally, the 3D voxelization allows for easier control with the ability to give a 3D model as base input. More details on point 3.3. and 3.4
3. During the entire process I tried to minimize size which worked partially. The main constraint being the length of the motor. I believe this can still be reduced if designing a custom motor. Comparing it to other modular system is difficult as no size data is always provided. A size of (48 * 140 mm) is decent as I use cheap components everyone can buy online.

4. Extras

4.1 Conclusion

I embarked on a quest to bring to life a concept from a fictional film, delving into the realms of modular robotics and self-assembling systems. Beginning with an exploration of these concepts, I eventually achieved a significant milestone: the theoretical demonstration of constructing a worker module. Moreover, I successfully tackled the challenges of controlling larger systems through voxelization algorithms. Lastly, using deep reinforcement learning, I enabled the module to walk faster than I could ever be achieved through traditional hardcoded methods.

Throughout this journey, I harnessed knowledge of important technologies, including 3D modeling, electronics, programming, machine learning, and virtual reality. These tools were instrumental in realizing the vision and pushing the boundaries of what is possible with modular robotics.

As of now, the mission has been a successful, and the potential use cases for such modular robotics are limitless. From everyday tasks like cleaning and tidying one's house to more critical applications such as supporting a collapsing bridge, the MIRAS project stands ready to tackle any challenge thrown its way.

4.2 Future Plans

My next significant milestone is the construction of a worker module.

However, before there are numerous tasks that remain. These include verifying my electrical diagram with professionals, adapting the AI for turning, and navigating rough terrains, and identifying more efficient and stable structural designs. Additionally, running more simulations to ensure that the system can effectively handle more real-world use cases.

In summary, while my adventure with this project is still ongoing, the progress made thus far has been promising, and I have successfully demonstrated the feasibility of the main concept.

5. Appendix

5.1 References

Official Documentations & forums:

- Oculus SDK
- Unity 3D
- OpenAI Gym
- TensorFlow
- Wiki.Seedstudio.Com

Articles & papers:

1. Ronan Hinchet, Velko Vechev , Herbert Shea & Otmar Hilliges, DextrES: Wearable Haptic Feedback for Grasping in VR via a Thin Form-Factor Electrostatic Brake, aitlab
2. Mariella Moon, Echolocation could give small robots the ability to find lost people, engadget
3. Davide Cavagnino & Marco Gribaudo, Discretization of 3D models using voxel elements of different shapes, researchgate
4. Park Chan-Il & Cho Do-Hyun, Comparison of Dynamic Characteristics of Spur Gears and Helical Gears, KoreaScience
5. Jianglong Guo, Jinsong Leng & Jonathan Rossiter, Electroadhesion Technologies for Robotics: A Comprehensive Review, ieeexplore
6. Jason Poel Smith, Create an Arduino Controlled Battery Charger, allaboutcircuits
7. Celera Motion, A Guide to Robot Joint Design, azorobotics
8. Mengran Gao, Ningjun Ruan, Junpeng Shi & Wanli Zhou, Deep Neural Network for 3D Shape Classification Based on Mesh Feature, MDPI
9. Nicholas Renotte, Complete Machine Learning and Data Science Courses, youtube
10. A. Utsumi, J. Ohya, Multiple-hand-gesture tracking using multiple cameras, ieeexplore
11. Sudharsan Ravichandiran, master reinforcement and deep reinforcement learning using OpenAI gym and tensorflow, google scholar book
12. Zushi Tian, Ye Tian, Hailong Ye, Xianyu Jin & Nang.o Jin, VOX model: application of voxel-based packing algorithm on cementitious composites with 3D irregular-shape particles, springer link
13. Bronson Zgeb, Simple Mesh Voxelization in Unity, bronsonzgeb.com
14. Bram Lambrecht, Voxelization of boundary representations using oriented LEGO®plates, semanticscholar
15. Yuxi Li, Deep Reinforcement Learning: An Overview, arxiv
16. Sayon Dutta, A beginner's guide to designing self-learning systems with TensorFlow and OpenAI Gym, google scholar book

ChatGPT:

1. Occasionally, after writing my own text, it was used to correct grammatical errors and enhance the fluidity of the text, allowing for quicker revisions.
2. To find synonyms and enrich the vocabulary or to modify certain expressions.

At no point did ChatGPT generate new text with information that I hadn't provided or restructure my texts. It solely rephrased existing content. The output was then reviewed and refined. Thus, ChatGPT did not directly influence the content of my project but facilitated the creation of a better report than I could have achieved alone in such time.

5.2 Image sources

If not indicated the images in this report were self-made.

Page 2:	Big Heroes 6 microbots: sidefx.com
Page 3:	Modular robot left: newscientist.com Modular robot middle: staff.aist.go.jp/e.yoshida/test Modular robot right: news.mit.edu/2019 Furniture modular robot: actu.epfl.ch/news
Page 4:	Modular robot left: robotics247.com Modular robot right: spectrum.ieee.org
Page 5:	Beehive: apis-donau.com Electromagnet: aliexpress.com
Page 6:	Seeduino with pins: wiki.seeedstudio.com/Seeeduino-XIAO
Page 7:	Servo Motor: aliexpress.com Motor driver: aiema.com/part/application Motor data sheet: ineedmotors.com
Page 8:	Battery recharger: allaboutcircuits.com/projects
Page 9:	Bypass capacitors: microcontrollertips.com
Page 11:	Chrome Dino: play.google.com Reinforcement Learning Graph: medium.com/@vishnuvijayanpv
Page 12:	OpenAI GYM environments: gymlibrary.dev
Page 14:	Rabbits: mdpi.com/1424-8220 Cubes: en.wikipedia.org/wiki/Voxel
Page 15:	Europa: diglib.eg.org/bitstream/handle
Page 16:	Egyptian pyramid: pinterest.com Passthrough room scan: developers.meta.com
Page 17:	Hands: Godotengine.com
Page 18:	Quantum computer: fintechmagazine.com/articles Identify Algorithms: tomdobb1.medium.com

【評語】 190029

This work conceptualizes the designs and implementation of a modular self-assembly robot system. Specifically it implements and shows how a two legs robot can use reinforcement learning to reconfigure themselves to learn how to walk together. The work can be extended to more sophisticated configurations that serves as the target goal structure for the multiple modular robots to reconfigure themselves under a centralized coordinator who assigns the prioritized sequence of configuration actions. The ideas are interesting and innovative. Future work can be extended to remove the centralized control of reconfiguration process.