

: N° d'ordre  
: N° de série

**PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA**  
**Ministry of Higher Education and Scientific Research**



**UNIVERSITY OF ECHAHID HAMMA LAKHDAR D'EL OUED**  
**FACULTY OF EXACT SCIENCES**  
**Department of Computer Science**



**End of Study Dissertation**  
**Presented for obtaining the Diploma of**

## **ACADIMIC LICENCE**

Domain : **Mathematics and Computer sciences**  
Field : **Computer sciences**  
Specialty : **Informations Systems**

Presented by :

- **HASSOUNA Athmane**
- **HAKKOUM Abdelazize**
- **GUEDIRI Abderrahman**

### **Subject**

**The simulation of a maze  
solving robot using NetLogo**

Proposed and Supervised by : **Miss. Chourouk Guettas**

Presented on xx-xx- 2017 before the jury:

M. ....	MCA	President
M. ....	MAA	Rapporteur

**Academic year: 2016-2017**

## **Thanks**

**To the one that never dies, who guided us to the righteous path, thank you “Allah” for getting us to finish this humble work. Then, we thank every one who was a reason for us to be at this point of our life, we mention in particular Miss. Chourouk Guettas who had a great patience and guided us through this work with all of the effort she had, also, we send our thanks to every teacher and professor who has a hand in forming us as computer scientists, then to all of our friends, thanks for making these years easy for us. And last to the jury for correcting us.**

*Really, thanks everybody.*

## **Dedication**

**I dedicate this work to the two ones who never doubted me, and stood there by me in health and sickness, “mom” “dad”, I worked on this report with all of my heart and now it is dedicated to you, it is nothing compared to your patience with me and your love for me.**

**Also, I dedicate this work for all of my brothers and sisters “Khaled, Sara, Abdessalem, Aicha and Zakaria” , I appreciate your support for me, and without forgetting my little niece “Nour Elhouda” and her father “Abdelmajid”.**

**To my two grandmothers, my two uncles and my grandfather whom did not have the chance to see me at this moment, I hope that you were with me right now, may the mercy of Allah be upon you all.**

**To my grandfather, my uncles and my aunts for being such models for me, this work is dedicated for you too. Just if those lines were much longer I would mention a lot more.**

**For all of my loved ones.**

***Athmane Hassouna***

## **Abstract**

The artificial intelligence (AI) and the robotics are two connected fields, the use of the AI in robotics got us a step closer in creating robots that think like humans. In this dissertation, we process one of the most famous problems for the AI and robotics which is maze solving, and try to apply some algorithms in robots to see their behavior. The best choice for solving the maze was “A\* path finding algorithm”, but considering that it costs a lot of time, we optimized it to give the minimum time possible. We used “NetLogo” to simulate the mazes and the robot in order to test and compare the different maze solving algorithms.

**Keywords:** Artificial intelligence, Robotics, Robots, Simulation, NetLogo, Path finding, Maze.

## Résumé

L'intelligence artificielle (AI) et la robotique sont deux champs connectés, l'utilisation de l'IA en robotique nous a rapproché vers la création des robots qui pensent comme des humains. Dans cette dissertation, nous traitons l'un des problèmes les plus célèbres pour l'IA et la robotique, la résolution d'un labyrinthe et essayer d'appliquer certains algorithmes sur les robots pour voir leur comportement. Le meilleur choix pour résoudre le labyrinthe était «algorithme de recherche de chemin A\*», mais parce qu'il consomme beaucoup de temps, nous l'avons optimisé pour donner le minimum de temps possible. Nous avons utilisé "NetLogo" pour simuler les labyrinthes et le robot afin de tester et comparer les différents algorithmes de résolution de labyrinthe.

**Mots-clés:** Intelligence artificielle, Robotique, Robots, Simulation, NetLogo, Recherche de chemin, labyrinthe.

## ملخص

الذكاء الاصطناعي و الروبوتيات مجالان متصلان، فاستعمال الذكاء الاصطناعي في الروبوتيات جعلنا نتقدم خطوة أقرب لصناعة روبوتات تفكر كالإنسان. في هذه المذكرة، نحن نقوم بمعالجة واحدة من أشهر المشاكل المطروحة في مجالي الذكاء الاصطناعي و الروبوتيات وهي حل المتاهة، و سنحاول تطبيق بعض الخوارزميات في الروبوتات لنرى كيفية تصرفها. أفضل خيار لحل المتاهة كان "خوارزمية A\* لإيجاد الطريق"، و لكن بإعتبار أنها تكلف الكثير من الوقت، قمنا بتحسينها لإعطاء أقل وقت ممكن. و قد إستعملنا "NetLogo" لمحاكاة المتاهة و الروبوت لنقوم بمقارنة مختلف خوارزميات حل المتاهة.

**الكلمات المفتاحية:** الذكاء الاصطناعي، الروبوتيات، الروبوتات، المحاكاة، NetLogo،

إيجاد الطريق، المتاهة.

## Index

General introduction.....	2
CHAPTER I.....	4
I.1. Introduction.....	4
I.2. Artificial Intelligence.....	4
I.3. Robotics.....	5
I.4. Mazes.....	10
I.5. Maze solving.....	11
I.6. Conclusion.....	11
CHAPTER II:.....	13
II.1. Introduction.....	13
II.2. Architecture.....	13
II.3. Conclusion.....	19
CHAPTER III:.....	21
III.1. Introduction.....	21
III.2. NetLogo.....	21
III.3. Motivation of using NetLogo.....	21
III.4. The machine used for simulation.....	22
III.5. Implementing.....	22
III.6. Simulation.....	26
III.7. Results.....	29
III.8. Conclusion.....	30
General conclusion.....	32
Bibliography.....	33

## Figures Index

Figure-I.1. The digesting duck.....	6
Figure-I.2. Unimate.....	7
Figure-I.3. Shakey the mobile robot.....	7
Figure-I.4. The RIMHO-2 demining robot.....	8
Figure-I.5. Da Vinci surgery robot.....	8
Figure-I.6. Maze solving robot.....	9
Figure-I.7. Aibo the robot dog.....	9
Figure-I.8. 15*15 maze.....	10
Figure-I.9. The labyrinth of the “Minotaur”.....	10
Figure-II.1. The system architecture.....	13
Figure-II.2. Example of an environment.....	14
Figure-II.3. Sensors in a robot.....	15
Figure-II.4. Sensors needed in the maze.....	15
Figure-II.5. Robots with GPS devices.....	16
Figure-II.6. Looping robot.....	17
Figure-II.7. The original A* cost (sky color cells).....	18
Figure-II.8. The proposed A* cost (the sky color cells).....	19
Figure-III.1. The general interface.....	22
Figure-III.2. Setup button.....	22
Figure-III.3. Initialized world.....	23
Figure-III.4. The maze generating button.....	23
Figure-III.5. A 3D view for a generated maze.....	23
Figure-III.6. A chooser for the algorithms.....	24
Figure-III.7. The button of creating the robot.....	24
Figure-III.8. The button of initializing the variables.....	24
Figure-III.9. The maze solving button.....	25
Figure-III.10. Simulation of a robot solving the maze.....	25
Figure-III.11. Time comparing plot.....	25
Figure-III.12. Simulation: step 1.....	26
Figure-III.13. Simulation: step 2.....	26
Figure-III.14. Simulation: step 3.....	27
Figure-III.15. Simulation: step 4.....	27
Figure-III.16. Simulation: step 5.....	28
Figure-III.17. Simulation: step 6.....	28
Figure-III.18. Chart for time of solving different mazes.....	29
Figure-III.19. Chart for the average time of solving mazes.....	29

Tables Index

Table-III.1. Table of time needed for finishing different mazes.....29

## ***General introduction***

## **General introduction**

Today's world offers an easy life for humans, that is achieved with the existence of robots, they may be considered a way of entertainment or a sort of luxuries by a lot of people. But, they are not just like that, they could replace humans in many fields like industry, danger military missions and nuclear reactors. The first robots were able to do only few simple jobs that they made especially for doing it, then after the appearance of computer sciences, the use of the artificial intelligence with the robots starts, so they can do more difficult and intelligent actions.

The success made by applying the artificial intelligence on the robots took the researchers to another level of using the robots, the goal now is to make the robots think like the humans, when facing a problem, when asked a question, helping people and a lot more, but taking in mind to never harm the humans.

The appearance of the robots was not very long, in the opposite of robotics, which appeared a long time B.C (Before Christ), they was just some primitive machines with simple task to do, but they gave the idea of making machines capable of thinking and making decisions of their own which are the ideal robots that researchers dream of.

Of course the researchers made a huge progress in this field, but considering the expensive prices of the materials used in robots, the simulation of the robots was the perfect way of avoiding such loss, so after analyzing and solving a problem by simulation, it could be taken to the real world.

The goal of our dissertation is to simulate a robot that solves one of the most famous problems in artificial intelligence which is "maze solving". So, this report will show the results of the simulation by going through three (03) chapters, which are:

- I. Chapter I: This chapter will show the general idea of the maze solving problem along with its relation to the robots, and that by mentioning the history of the robots and robotics and their uses, the mazes and their history and without forgetting the artificial intelligence.
- II. Chapter II: After knowing all about the problem, it is necessary to make a system for solving it by the robot, and what algorithms will be used, and this chapter will cover that.
- III. Chapter III: Now, and the theory part is done, it is time to implement all the ideas and see what results will appear.

## *Chapter I*

# CHAPTER I

## I.1. Introduction

It is known that the computer science reached a lot of human technologies and it is almost everywhere, and the artificial intelligence is one of its branches it studies robotics and programs robots to do a dangerous job that human can not do, or to replace hand working in factories or other places, also to solve problems and issues as an example of these problems “mazes and maze solving problems”. So, what is the artificial intelligence? Robotics? Mazes? And maze solving? And what we know about them?.

## I.2. Artificial Intelligence

Artificial Intelligence (AI) is some specific properties and behaviors implemented in computer programs that makes it simulate human brain. Some important properties are the ability to learn and conclude and then react to an event that these programs are not programmed to it.

Artificial intelligence is an approach of computer science, and many scientists said that it is the studying and designing of the “intelligent agent”, the intelligent agent is a system that studies its environment and makes decisions to increase the chances of solving its problem[1]. That field established assuming that the intelligence can be described so precise that a machine can simulate it.

Based on the division of the problems concerned by the AI[2]. the researches targeted the AI to several goals which are:

- Concluding, logical thinking and problems solving.
- Knowledge representation.
- Planning.
- Learning.
- Natural language processing.
- Movement and the ability to change.
- Perception.
- Social Intelligence.
- Creativity.
- Motions and manipulation.
- General intelligence.

There is no specific theory or paradigm used in AI researches. Many researchers argue about what will make the machines simulate the humans ?, is it the study of the psychology ? or the neurology ?, or is the human biology completely irrelevant to the AI and only by the logic or optimizing will achieve human behavior?, so the AI divided to several approaches, which are :

- Cybernetics and brain simulation : In the 1940s and 1950s, a number of researchers discovered a relation between neurology, information theory and cybernetics. Some of them built machines that used electronic networks to exhibit rudimentary intelligence.
- Symbolic AI : In the middle of 1950s, AI researchers discovered the possibility to represent human intelligence in symbolic manipulations. During the 1960s this approach reached great success in simulating high-level thinking in small demonstration programs.
- Sub-symbolic AI : This approach rejected the symbolic AI and focused on the engineering of the robotics and the “control theory”.
- Statical AI : In the 1990s some researchers used mathematical tools to solve some problems, although this approach was successful in a specific field but it failed to get close to the ultimate goal of the AI which is general intelligence.
- Integrating the approaches : The approach is based on “Intelligent Agent” paradigm. It uses the paradigm to solve specific problems without arguing about which approach will be used, some agent use symbolic approach and some other use sub-symbol and may be using a new approach. A hybrid intelligent system was created by combining sub-symbol at its lowest levels and symbolic at highest levels (low human control and high self-control).

### **I.3. Robotics**

1. Definition : The robotic is the group of sciences that is concerned of studying, designing, developing, manufacturing the robots and its applications, it is related to computer science, electronics, mechanics and computer systems for controlling, sensory and information processing[3].

The main objective of robotics is to create machines that can substitute humans in dangerous environments to avoid human loss or in manufacturing for more accurate and faster production.

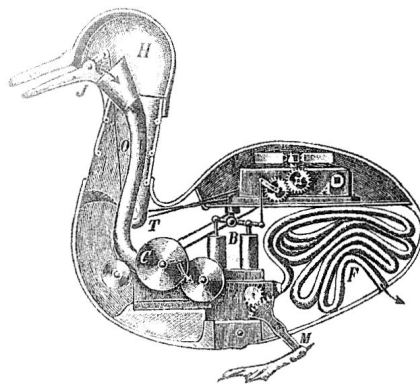
The conception of creating a self-control machine existed from the classical ages, but it really appeared in the 20<sup>th</sup> century, and there was assumption that there will

be a revolution against the humans if these robots start to think of their own mind, so Asimov put the four laws of robotics for robots thinking and behavior[4]:

- ✗ Law 0: A robot may not harm humanity, or by inaction, allow humanity to come to harm.
- ✗ Law 1: A robot may not injure a human being or through inaction, allow a human being to come to harm, unless this would violate a higher order law.
- ✗ Law 2: A robot must obey orders given to it by human beings, except where such orders would conflict with a higher order law.
- ✗ Law 3: A robot must protect its own existence as long as such protection does not conflict with a higher order law.

2. History : The robotic history is a long history that returns to years BC and keep on developing to the present time:

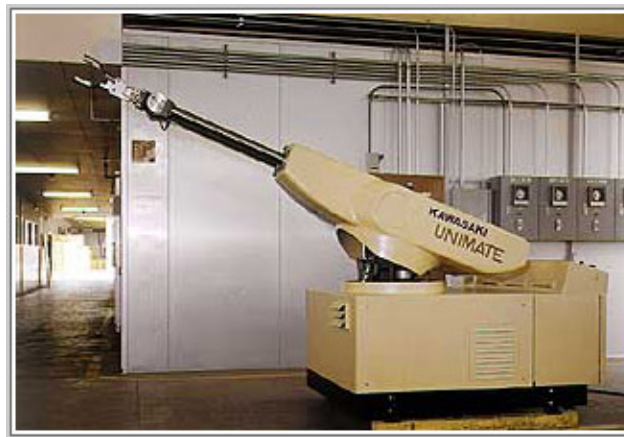
- 1400 BC: The Babylonians invented the clepsydra, a clock measures time using the flow of water, and considered as the first robotic device in history.
- 1495: Leonardo da Vinci designed a clock knight that can stand up, wave its arms and move its jaw. It is not known if the knight is ever built or not but it could be considered as the first humanoid robot design.
- ~1738: A clockwork duck was invented by the French inventor Jacques de Vaucanson, it was capable of flapping its wings, quaking and eating.



**Figure-I.1. The digesting duck.**

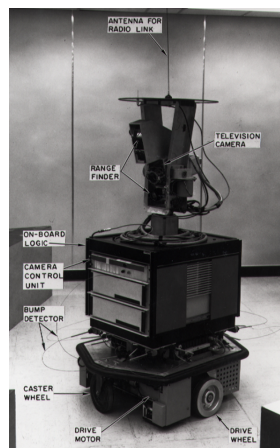
- 1770: The Hungarian author and inventor Wolfgang von Kempelen revealed “the turk” a maple-wood box with a mannequin, dressed in cloak and turban, The device gained great fame as an automaton capable of playing chess against skilled opponents until it is discovered that a human operator hides inside the box.

- 1804: The French inventor Joseph Marie Jacquard created a loom controlled by some punched cards, it simplified the process of manufacturing textile and thousands of it were produced and used across Europe within a decade.
- 1921: The first appearance of the term “robot” was in science fiction play R.U.S ( Rossums Universal Robot) by the Czech author Karel Čapek. The play tells the story of manufacturing robots by humans and their rebellion that leads to the extinction of mankind.
- 1942: Isaac Asimov introduced the three laws of robotics (the fourth was added later) in his short story “Runaround”.
- 1954: George Devol granted the patent for the first programmable robot (“Unimate”) employed in industrial production work.



**Figure-I.2. Unimate.**

- 1966-1972: The creation of the first mobile robot “Shakey” by Stanford Research Institute.



**Figure-I.3. Shakey the mobile robot.**

- 1980s: The robot industry enters a phase of rapid growth. Many institutions introduce programs and courses in robotics.

- 1995-present: The need for a faster and better quality products and safer environment for humanity pushed the robotics development to a massive growth in most of the fields.
3. Applications : The robotics can be a substitution for humans in many fields, there are certain fields where humans can not endanger them self, so what are these fields? And what are the uses and applications of the robotics?:
- ✓ Military uses: The military uses of robotics are bombs detection, demining and spying[5].



**Figure-I.4. The RIMHO-2 demining robot.**

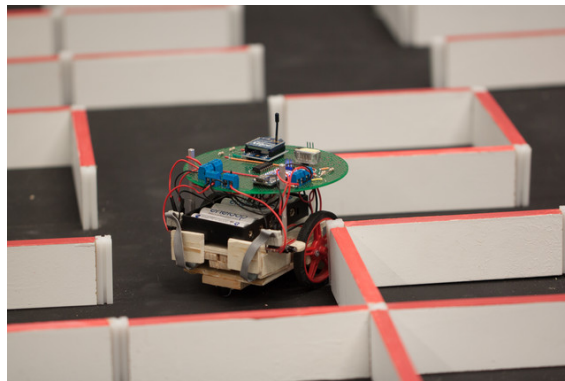
- ✓ Medical uses : To minimize the human loss, the hospitals can use robotics in many ways such as surgery, patient care and research to ensure more precise and better results[5].



**Figure-I.5. Da Vinci surgery robot.**

- ✓ Agricultural uses: Agricultural robotics is one of the promising technological solutions for the problems of the rapidly growth of human population and the future demand for more cheaper and better quality food.

- ✓ Manufacturing uses: Some of the industrial process require high-levels of precision like the automobile manufacturing, and the robotics are the perfect choice for such a mission. Or to clean-up radioactive waste.
- ✓ Exploration uses: The robots can go to places that mankind can not go, like far planets or the depth of the seas and oceans[5].
- ✓ AI uses: The best tools to use for AI researches are the robotics, especially in problems solving, as example maze solving problems.



**Figure-I.6. Maze solving robot.**

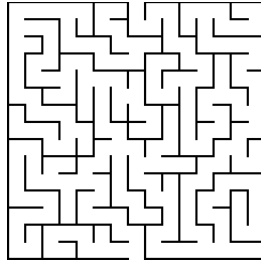
- ✓ Entertainment uses: The modern houses need more than the classical entertainment tools, the robotics came with a solution in the form of robots it has a lot of functions like playing music, preparing food, and cleaning[5].



**Figure-I.7. Aibo the robot dog.**

## I.4. Mazes

- Definition: A maze is a path or collection of paths that are very confusing and complex typically with one entrance and one exit (goal), people try to solve it by starting from the entrance to get to the exit[6].



**Figure-I.8. 15\*15 maze.**

It could be drawn on a paper or built in reality using different materials (grass, bricks, snow, or generated in video games, ...). The paths of the maze are normally fixed, but in video games they could change in order to make the maze harder and more challenging.

The maze could be confusing with the labyrinth, the difference is that the labyrinth has one path to the goal which is generally in the middle, but the maze has many paths which could be crossed or looping and the goal can be anywhere.

- History: The appearance of the mazes returns to a lot of labyrinths through the ages, like Pliny's "Lemnian labyrinth" and Pliny's "Italian labyrinth", but two great labyrinths were noticed in the ancient history:
  - ✓ The labyrinth of Hawara: It was built by Amenemhet III to protect his burial chamber which was placed in the middle.
  - ✓ The labyrinth of the minotaur: It was created by the legendary inventor Daedalus by the order of king Minos of Crete to imprison the “Minotaur”.



**Figure-I.9. The labyrinth of the “Minotaur”**

- Generating the mazes: There are various algorithms for generating mazes, there are some of them are based on the graph theory, as examples:
  - ✓ Depth-first search.
  - ✓ Recursive back-tracker.
  - ✓ Randomized Kruskal's algorithm.
  - ✓ ....

Also there is “recursive division method”, which keep dividing a chamber with random walls where each wall contains a random passage.

And other algorithms like cellular automaton algorithms and simple algorithms.

### **I.5. Maze solving**

There are many maze solving algorithms, some have no prior knowledge of the maze, and some others based on the graph theory, those can see the hole maze at once or at least have some information about it, the algorithm used to solve a maze depends on the type of the maze, some algorithms will keep looping if the maze has loops, and here are some examples of these algorithms:

- ✓ Random mouse algorithm.
- ✓ Wall follower algorithm.
- ✓ Pledge algorithm.
- ✓ A\* algorithm.
- ✓ Trémnaux's algorithm.
- ✓ Dead ends filling algorithm
- ✓ ....

### **I.6. Conclusion**

After going through the definition of the artificial intelligence and its uses and fields, as we did to the robotics and its applications, we notice that they are strongly related with each other. And by knowing about the generating of the mazes and how to use specific algorithms to solve it, in the next chapter we will try to generate a maze and simulate some of the algorithms to solve it by a robot respecting the AI principles.

***Chapter II:***  
***Conception***

## CHAPTER II: Conception

### II.1. Introduction

As it is mentioned in the previous chapter, the robotics are used for AI researches, and that is to program a robot how to analyze a situation or a problem and calculate the best solution, all by its own, and one of the famous and best problems to test that is the mazes solving problems, they allow us to see how a programmed robot will avoid the walls (obstacles) and at the same time thinks of a solution to solve the maze. To do that, first we need an algorithm to generate a maze, and then algorithms for solving the maze, but before all of that what will be used in the robot system to discover the maze and avoid obstacles?, all of that will be mentioned in this chapter.

### II.2. Architecture

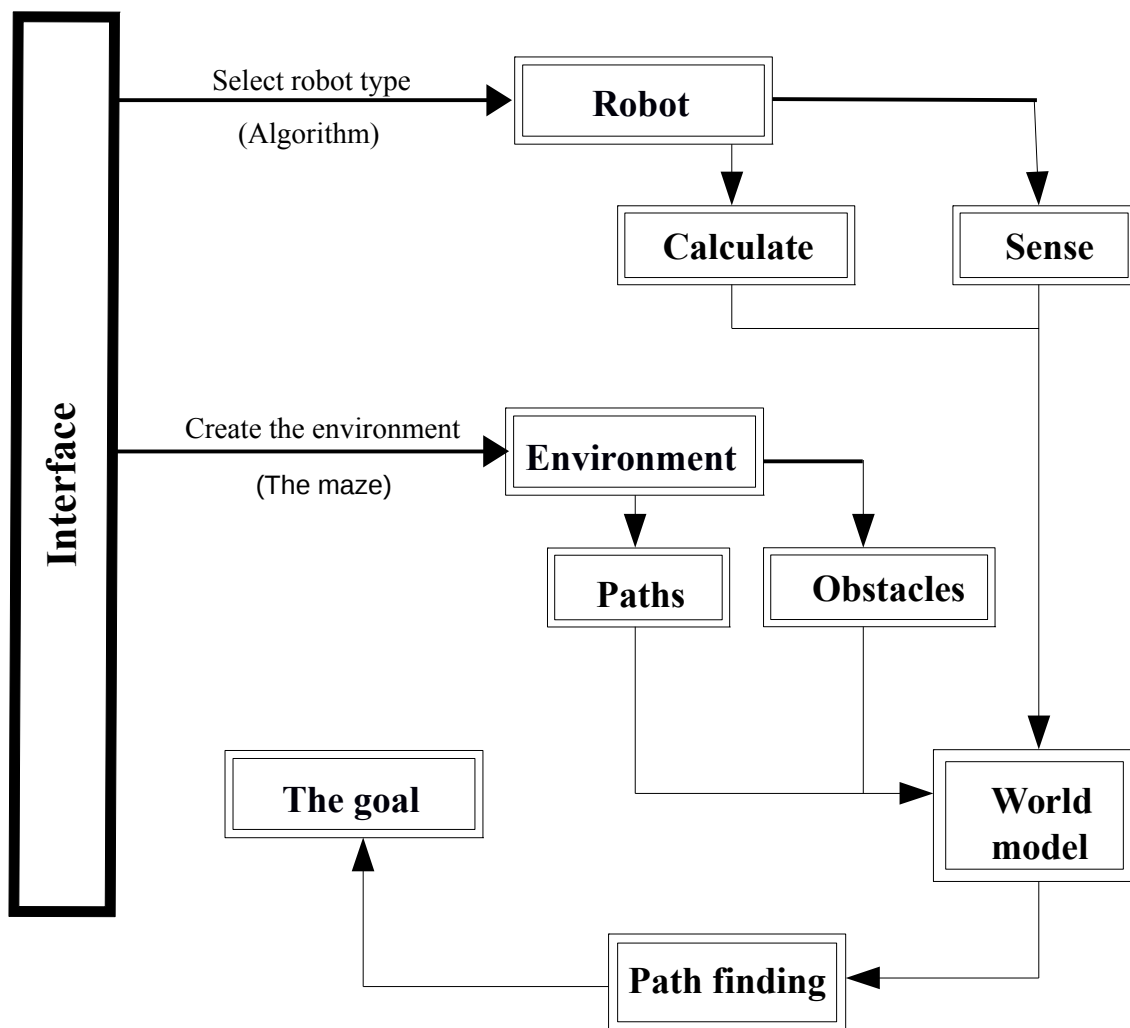
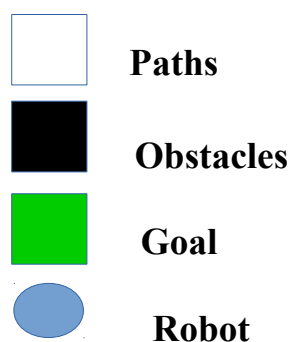
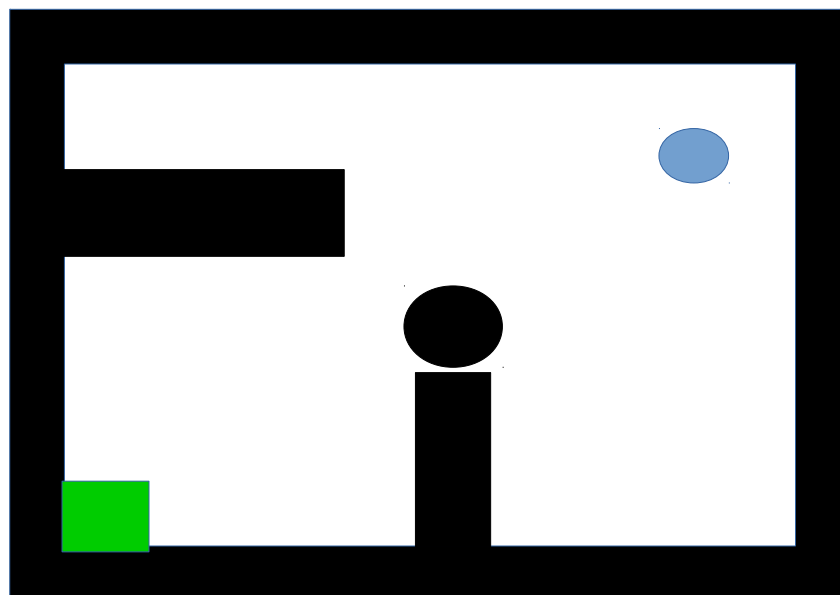


Figure-II.1. The system architecture.

- ✓ **Interface:** For this system the interface is a very important element, it give the opportunity to create the environment and and the robots and then give them orders to find its path.
- ✓ **Environment:** The environment is the space which will be explored by the robot, it contains many obstacles to be avoided by the robot and follows the paths until it gets to its goal.
- ✗ **Paths:** the paths are the available choices for the robot to move, every path is a possibility to its goal, they could be represented as nodes in a graph.
- ✗ **Obstacles:** They could be walls, rocks, or anything that prevent the robot from moving ahead, so the robot must avoid them, in order to reach the goal.



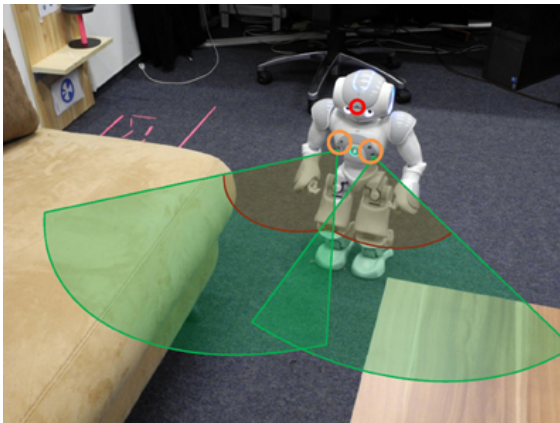
**Figure-II.2. Example of an environment.**

✓ Robot: The robot is the subject of this study, its purpose is to avoid the environment's obstacles and find the shortest possible path to reach the goal, in order to that the robot must contain some devices that helps it explore the environment around it, each device has its function:

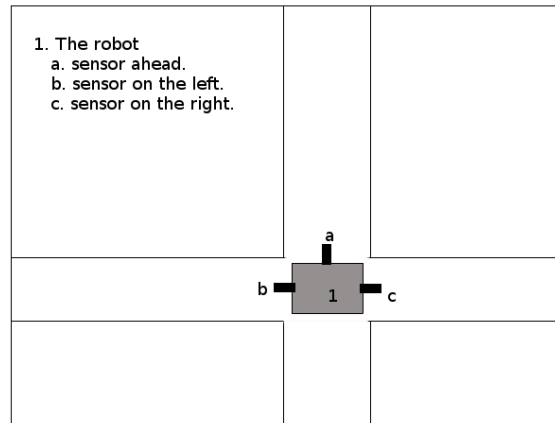
✗ Memory: it is a way for remembering the positions visited by the robot, it helps it when it gets to a dead-end so it can track back its steps until the next possible path [5].

✗ Sense: The sense is a very important function for the robot, its the only possible way to detect the obstacles ahead an avoid them, and the perfect devices to that function are “the sensors”, three (03) of them are enough for a robot [5]:

- A sensor to detect the obstacles ahead of the robot.
- A sensor to detect the obstacles at the left of the robot.
- A sensor to detect the obstacles at the right of the robot.

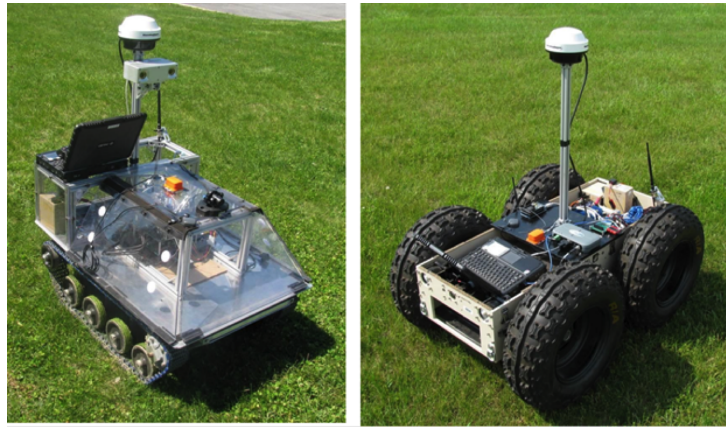


**Figure-II.3. Sensors in a robot.**



**Figure-II.4. Sensors needed in the maze.**

✗ Calculate: In order to find the shortest path to its goal, the robot must know where the goal is, and then calculate the distance between its location and the goal location, the device for such function is “Global Positioning System” or GPS.



**Figure-II.5. Robots with GPS devices.**

✓ **World model:** The world model is the representation of the environment, in this case the world model is a maze, its obstacle are the walls and the paths are the passages between these walls, it contains a lot of paths leading to the goal, and the robot must find the shortest one. But how to build such a model ?.

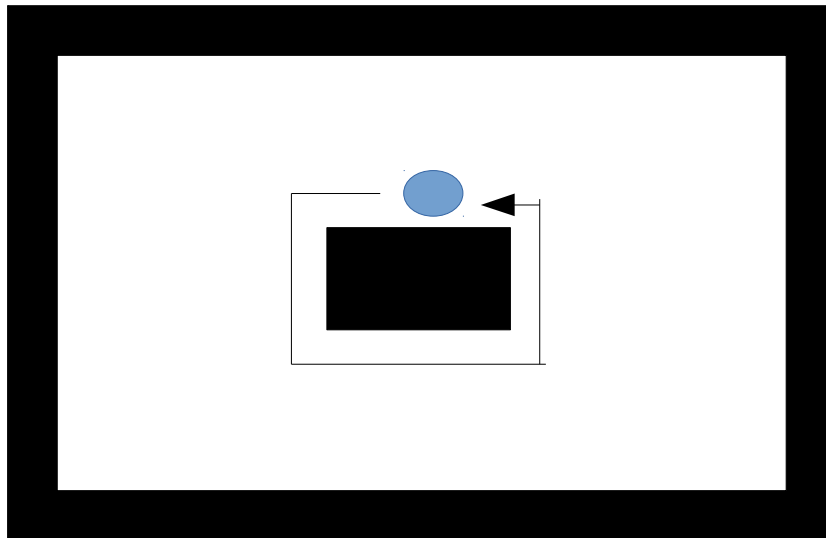
✗ **Generating:** As it was mentioned before the mazes have a lot of algorithms to generate them, and the “recursive back-tracker algorithm” was chosen, first of all we divide a rectangle into a grid of cells, then implement the algorithm:

Begin

- (1) choose an initial cell;
- (2) mark it as the current;
- (3) mark it as visited;
- (4) push it to a stack;
- (5) while (the stack is not empty) do
  - 1) if (the current has any unvisited neighbor) then
    1. choose one of them;
    2. remove the wall between the neighbor and the current;
    3. mark the neighbor as the current;
    4. mark it as visited;
    5. push it to the stack;
  - 2) else
    1. pop the current from the stack;
    2. mark the next cell in the stack as the current.

End.

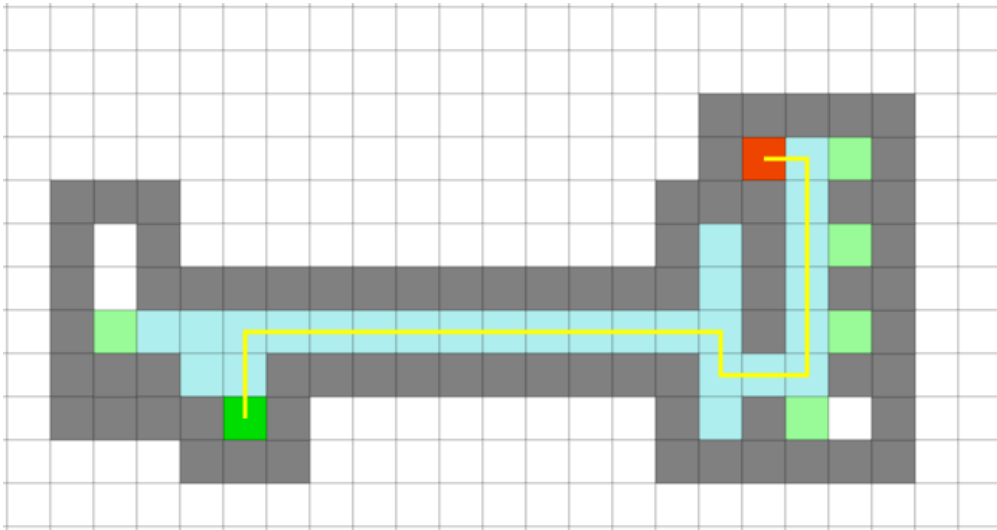
- ✓ Path finding: To contain the devices mentioned above is not enough for a robot to solve a maze, it must be programmed to use these devices properly, three (03) algorithms were chosen:
  - ✗ Random mouse: It is unintelligent algorithm that imitate the behavior of a human, or lap mouses, its principle is to walk through the paths and if it found an intersection chose a random path, and keep doing that until finding the exit of the maze.
  - ✗ Wall follower: It is a little bit more intelligent than the random mouse algorithm, it makes the robot to keep a wall on one of its sides, then to keep tracking it with the same side, until it find the exit, it is a very slow algorithm but efficient in case of a maze without loops, otherwise it may keep looping for ever.



**Figure-II.6. Looping robot.**

- ✗ A\* algorithm: The original A\* path finding algorithm gives every spot (node) three (03) variables which are G cost represents the cost from the spot to the goal, H cost represents the cost from the start to the spot, F cost represents  $G+H$ . It looks for the minimum expensive cost, and that by starting from the start spot and choosing the spot with the minimum F cost possible from its neighbors, then start from that spot, if there are two neighbors with equal F costs, it tests for the minimum G cost until reaching the goal. The problem with the algorithm is that if it reached a spot with an F cost more than one of the

previous one it returns to test the old spot with all of its neighbors, and that costs a lot of time.



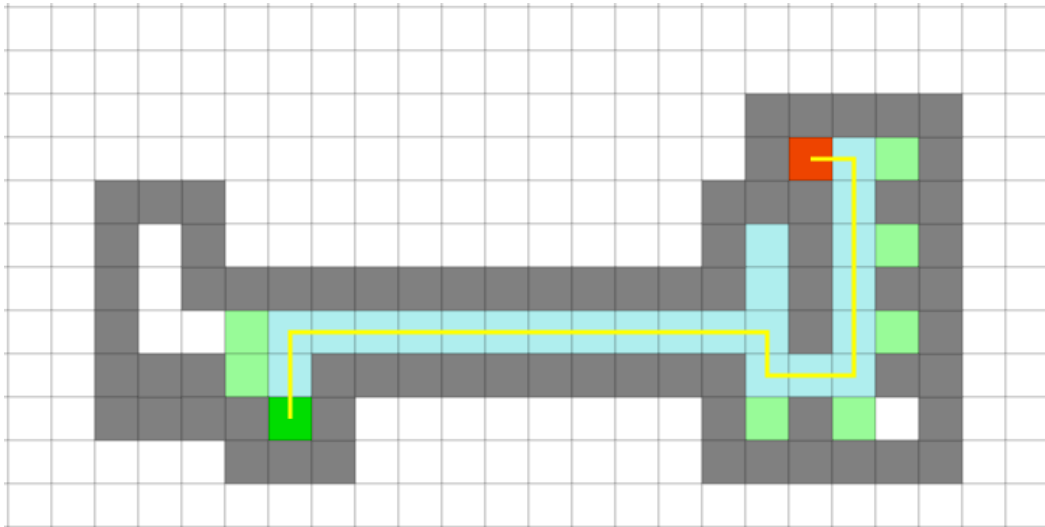
**Figure-II.7. The original A\* cost (sky color cells).**

The proposition in the case of the maze is to test the G cost only, and not to return only in the case of that the robot reached a dead-end, so it saves time of keeping exploring every possible path:

Begin

- (1) mark the start spot as the current;
- (2) push it to the stack;
- (3) mark it as visited;
- (4) while (the current is not the goal) do
  - 1) if (the current has any unvisited neighbors) then
    1. move to the neighbor with the minimum G cost;
    2. mark that neighbor as the current;
    3. mark it as visited;
    4. push it to the stack;
  - 2) else
    1. pop the current from the stack;
    2. mark the next spot in the stack as the current;
    3. move to the current;

End



**Figure-II.8. The proposed A\* cost (the sky color cells).**

### **II.3. Conclusion**

After knowing all about mazes and mazes solving problems, and the system architecture for this problem, a maze generation algorithm was proposed along with different types of maze solving ones with some improvements, and a lot of questions will be asked: is the system well build? Are the improved algorithms better than the original ones? And, which algorithm is working better?.

In the next chapter all of the questions above will be answered, by doing an implementation of the system and testing the mentioned algorithms.

*Chapter III*  
*Implementation and Results*

## **CHAPTER III: Implementation and Results**

### **III.1. Introduction**

After proposing the architecture of the system and the algorithms for generating and solving a maze, It must be tested before going to the real world, and that is by simulating it and see what results will be given.

In order to do that NetLogo will be used, so what is NetLogo? And how the simulation will work? And what results we will get? All of these questions will be answered in this chapter.

### **III.2. NetLogo**

[7]NetLogo is an agent-based programming language and integrated developing environment (IDE), it is free and open source, developed by Uri Wilensky (1999) in Northwestern University's Center for Connected Learning and Computer-Based Modeling (CCL).

It is a modeling language that helps simulating real world using the concept of the agents, which are:

- The turtles: represent the actors (humans, animals,cars, ...) of the environment.
- The patches: which are the fixed objects of the environment they can represent walls, trees, paths, ....

NetLogo helps to define the interaction between the turtles and the patches to simulate a problem or a phenomena that appears in the real world, and then study it.

It has two (02) versions, NetLogo which is 2D, working only with (x,y) coordinates, and NetLogo 3D which has (x, y, z) coordinate to represent the third dimension.

- The version used for the simulation is NetLogo 3D 6.0, which was released in December 20<sup>th</sup>, 2016.

### **III.3. Motivation of using NetLogo**

- ✓ NetLogo is a free and open source programming language.
- ✓ It is an easy language to learn and understand.
- ✓ It provides an interactive interface for users.
- ✓ It can simulate the real world's problem or a phenomenas which is perfect for the maze solving problem.

- ✓ The interaction between its agents shows the real world actions.
- ✓ NetLogo 3D provides a 3D view of the environment, so the simulation looks more realistic.

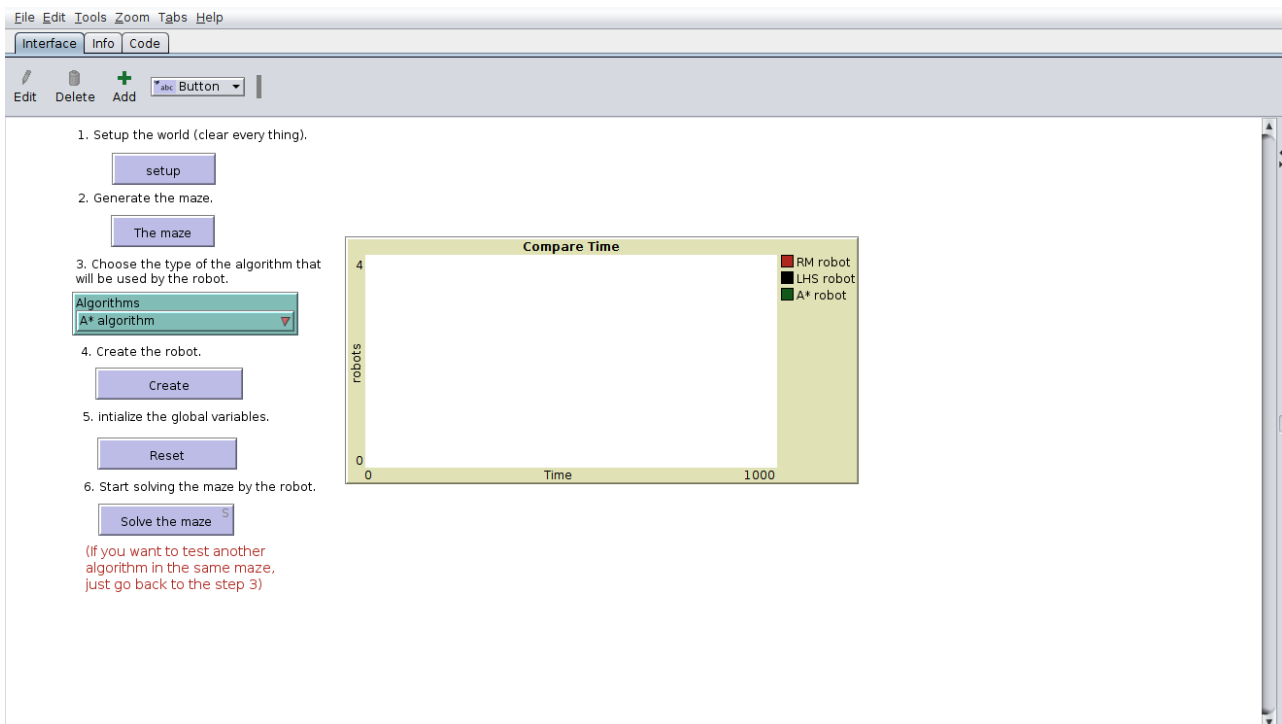
### III.4. The machine used for simulation

The characteristics of the machine used for simulating the architecture of the system are:

- Brand: Acer Aspire E 15 E5-573-38GH (Laptop).
- Processor: Intel core i3-4005u (1.70 GHz 3mb L3 cache).
- Ram: 4GB DDR3 L.
- Video card: Intel HD Graphics 4400.
- Operating system: Linux Ubuntu 16.04 (64-bit).

### III.5. Implementing

The general interface of the program contains the components needed to initialize the environment and simulate the maze solving.

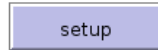


**Figure-III.1. The general interface.**

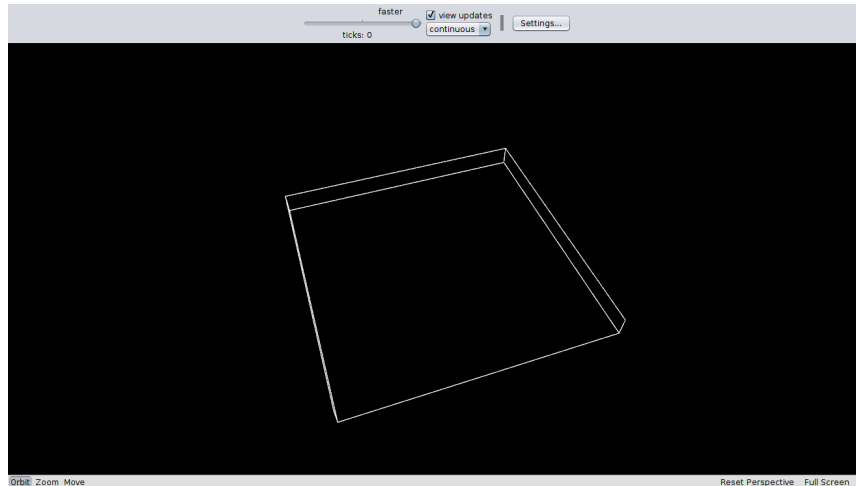
In order to walk through the implementation in a right way, these steps must be followed:

- ✓ Setup the world: Initializing the world to clear everything and to be ready for a new simulation is very important.

1. Setup the world (clear every thing).



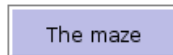
**Figure-III.2. Setup button.**



**Figure-III.3. Initialized world.**

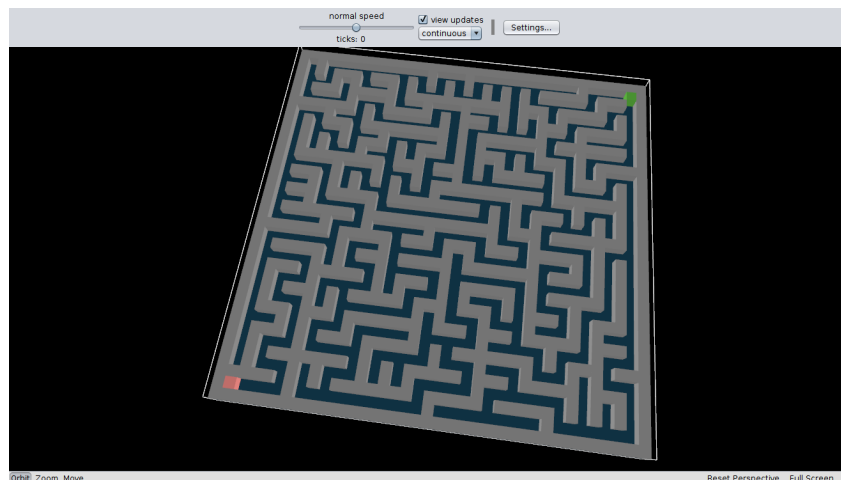
- ✓ The maze: The maze is the environment that is going to be solved, so it had to be generated at first.

2. Generate the maze.



**Figure-III.4. The maze generating button.**

It generates a totally random maze every time it is pressed, the red color patch is the start spot, and the green one is the exit, as example:



**Figure-III.5. A 3D view for a generated maze.**

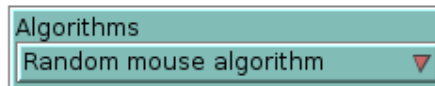
✓ Algorithms: As it was mentioned before, there is three (03) algorithms implemented in this system, and it is necessary to choose one of them for the robot to be working with.

\* Wall follower Algorithm (Left hand side algorithm).

\* Random mouse algorithm.

\* A\* algorithm.

3. Choose the type of the algorithm that will be used by the robot.



**Figure-III.6. A chooser for the algorithms.**

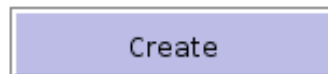
✓ Create: After choosing the algorithm, the robot to work with it must be created, depending on the type of the algorithm the robot will take a particular color:

\* Wall follower Algorithm (Left hand side algorithm): Yellow color.

\* Random mouse algorithm: Sky color.

\* A\* algorithm: Blue color.

4. Create the robot.



**Figure-III.7. The button of creating the robot.**

The robots are represented as breeds from turtles, it means that they are special turtles whom act only if they called by their names, and the breeds of this program are:

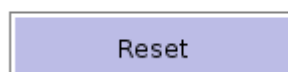
➤ robotLHSs: the breed representing the robot that works with the wall follower algorithm.

➤ robotRMs: the breed representing the robot that works with the random mouse algorithm.

➤ robotASs: the breed representing the robot that works with the A\* algorithm.

✓ Reset: The global variables used in the system may confuse a robot while finding the path, so they must be initialized first.

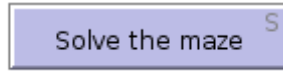
5. initialize the global variables.



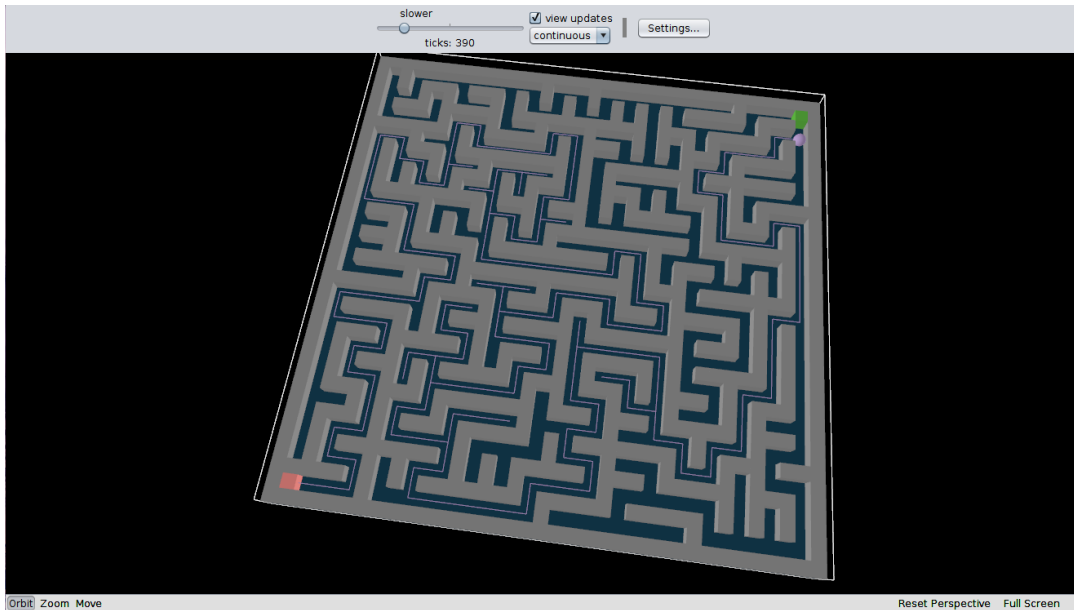
**Figure-III.8. The button of initializing the variables.**

- ✓ Solve the maze: After initializing everything and setting up the maze and the robot, it is time to start the simulation of the maze solving.

6. Start solving the maze by the robot.



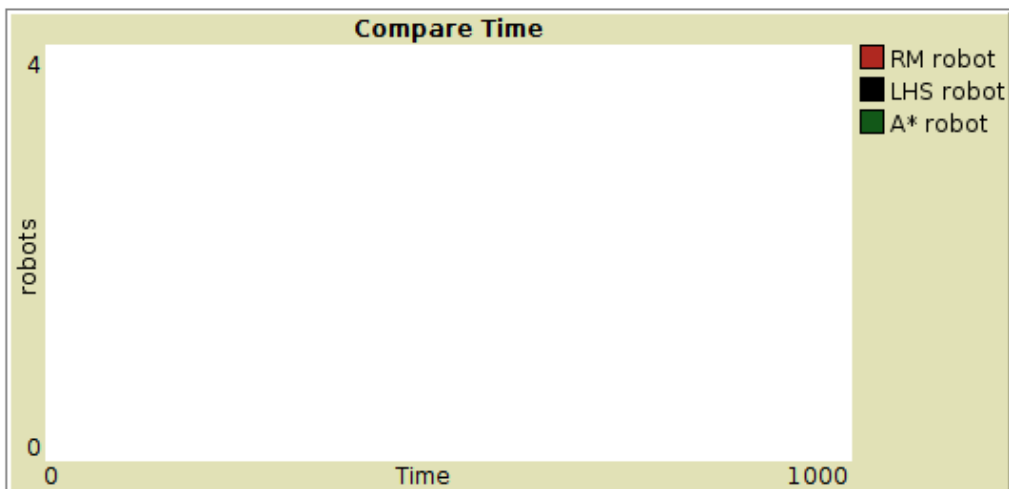
**Figure-III.9. The maze solving button.**



**Figure-III.10. Simulation of a robot solving the maze.**

- ✓ Comparing: In case of wishing to compare algorithms in the same maze, return to the step 3 (choosing the algorithm).

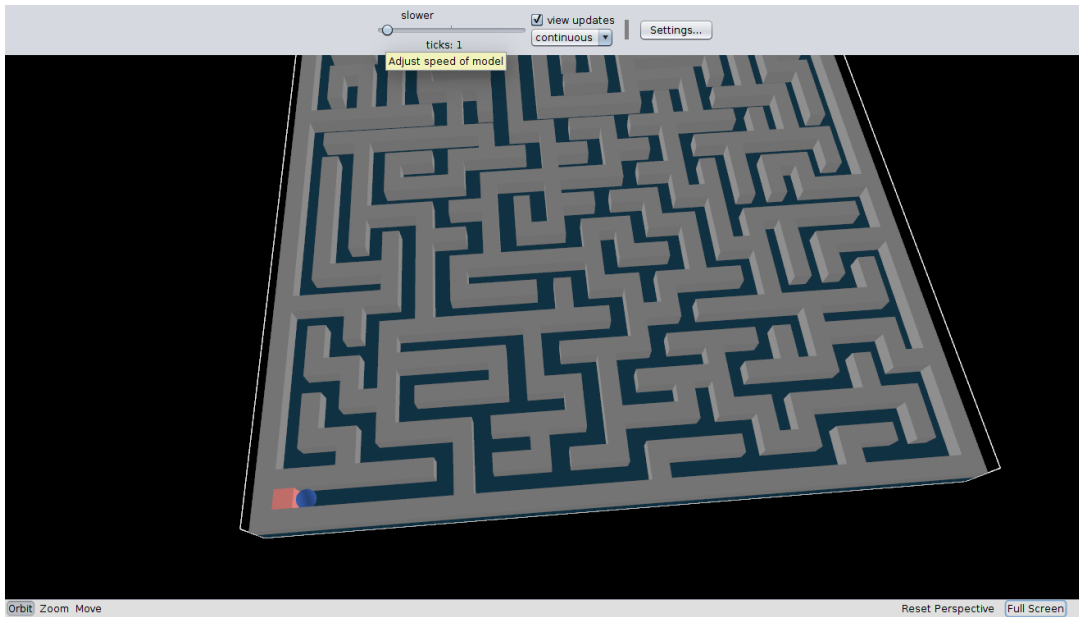
And the difference of time should appear in the plot while solving the maze.



**Figure-III.11. Time comparing plot.**

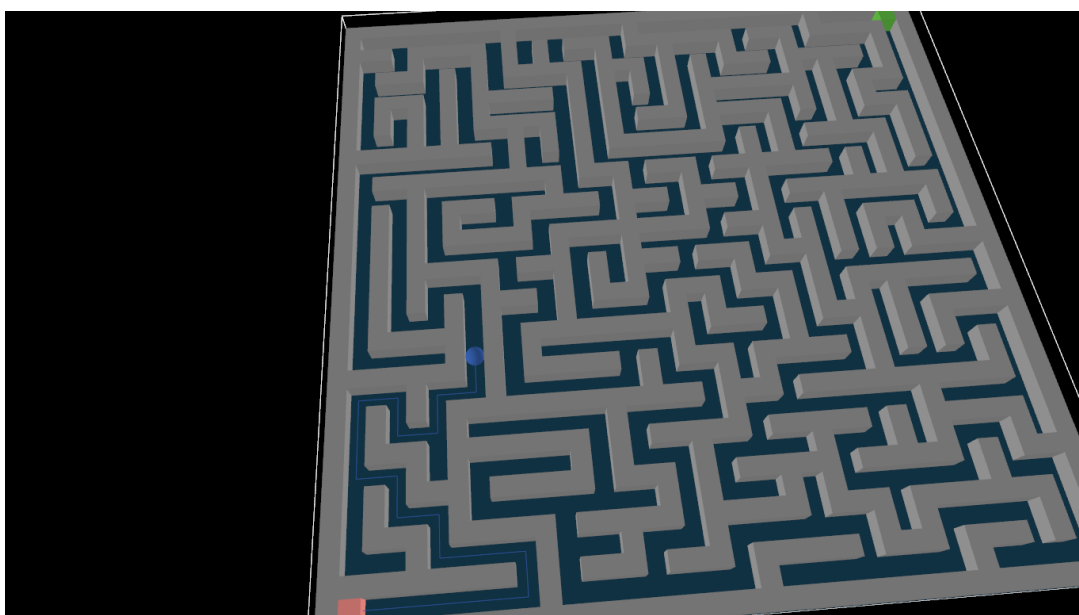
### III.6. Simulation

It is easy to understand the theory part, but how does it really work ?, the robot finds the path by testing the distance to the exit of the spots around it and takes the less one, it moves to it and memorize that distance in its memory so it knows if it returns to that same spot. Also, by using its sensors to find the walls it does not even consider them as a possible path. But to understand more, here are some steps from the simulation of a robot using A\* algorithm in a random maze.



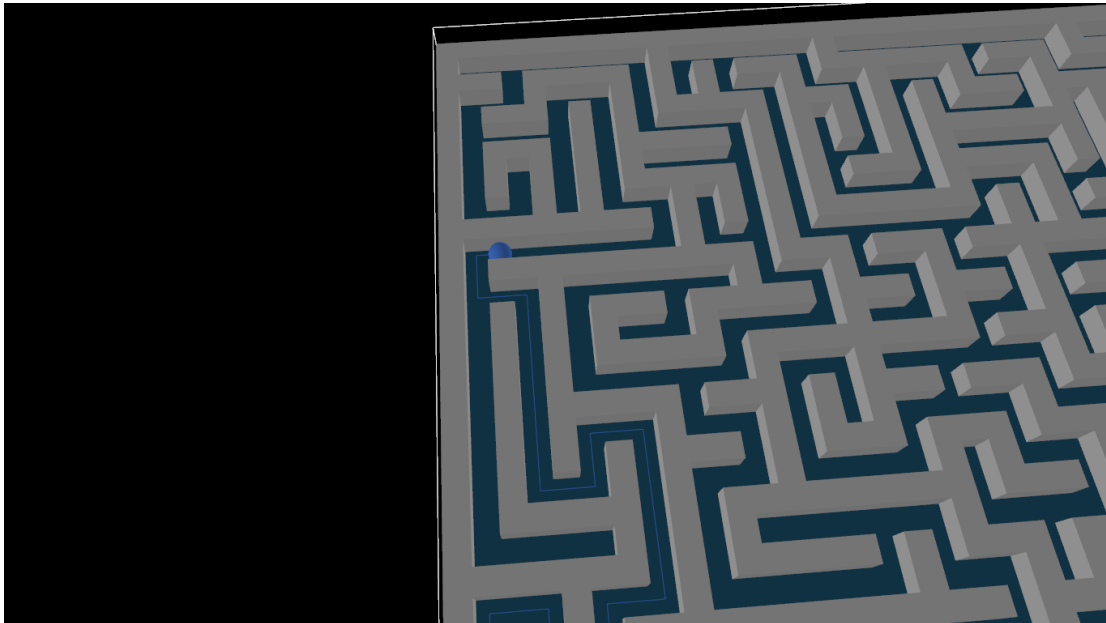
**Figure-III.12. Simulation: step 1.**

- Step 1: The start of the simulation.



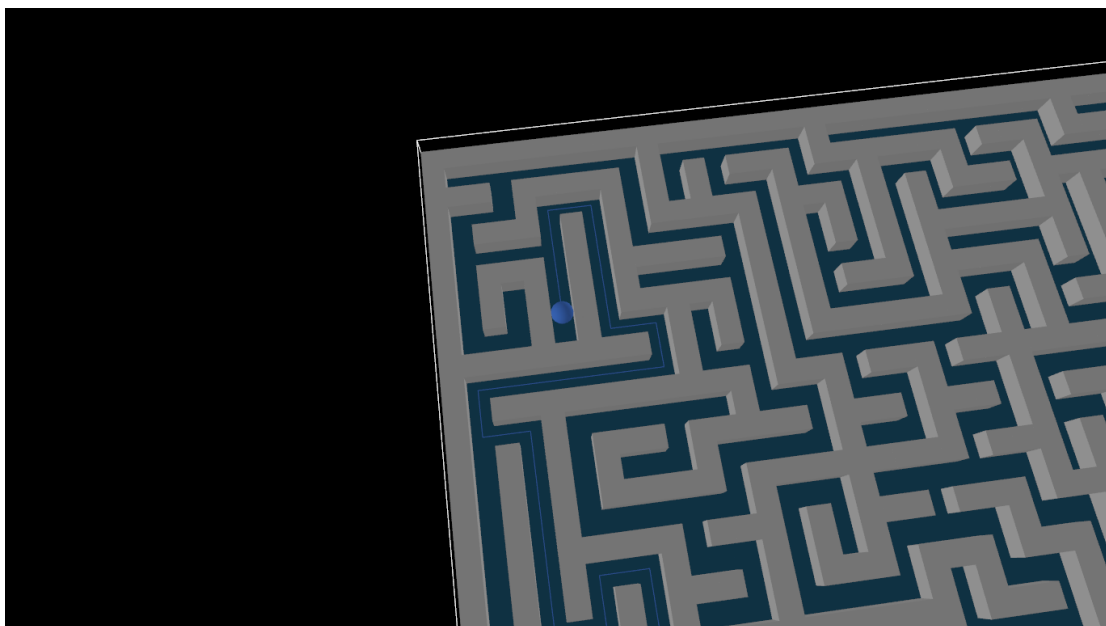
**Figure-III.13. Simulation: step 2.**

- Step 2: Avoiding the walls and walk along the paths.



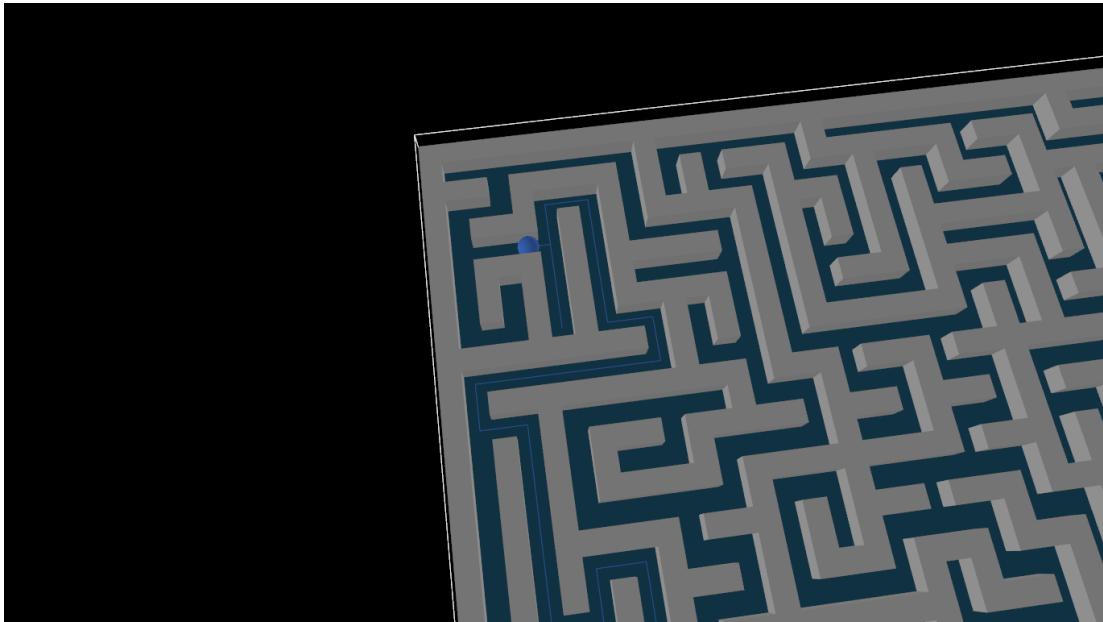
**Figure-III.14. Simulation: step 3.**

- Step 3: Avoiding to go to the farthest way and choosing the shortest one.



**Figure-III.15. Simulation: step 4.**

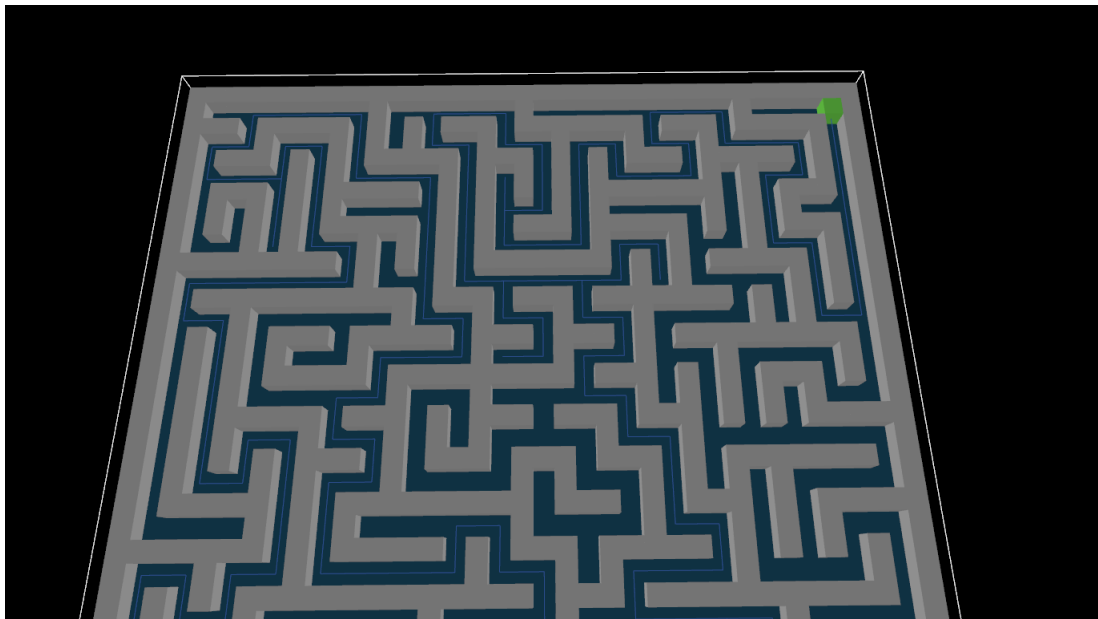
- Step 4: The distance in this dead-end is less than turning so the robot chose it.



**Figure-III.16. Simulation: step 5.**

- Step 5: Returning to the next closest path.

And these steps keep repeating until reaching the exit.



**Figure-III.17. Simulation: step 6.**

- Step 6: reaching the exit of the maze.

Like that, it is sure to get the shortest path possible, with some delay in time because of the dead-ends.

### III.7. Results

To see the results of the implementation, a test of the algorithms on five (05) different mazes was simulated, and the results are shown in the table and the charts below:

	Left hand side algorithm	Random mouse algorithm	A* algorithm
Maze 1	706	317	269
Maze 2	508	405	317
Maze 3	465	373	213
Maze 4	628	685	517
Maze 5	613	497	417
Average	584	455.4	346.6

Table-III.1. Table of time needed for finishing different mazes.

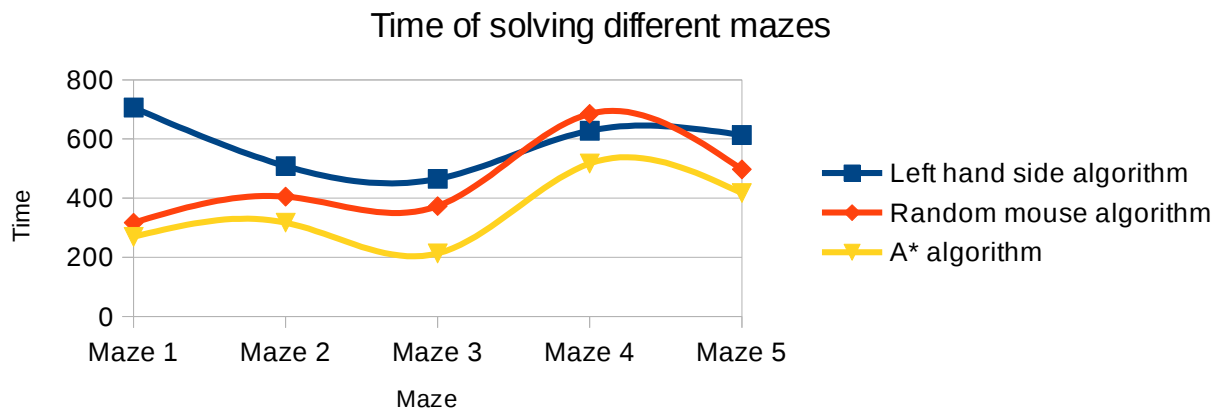


Figure-III.18. Chart for time of solving different mazes.

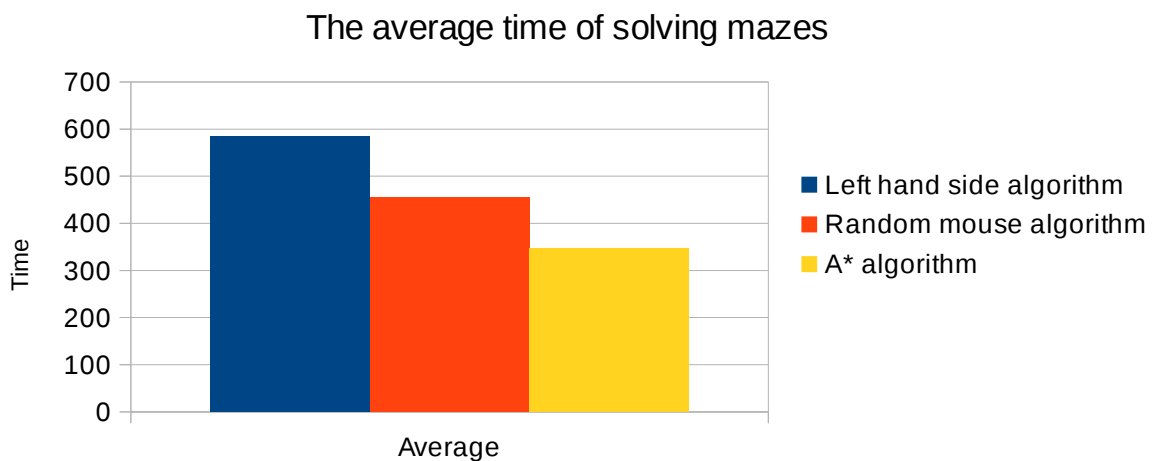


Figure-III.19. Chart for the average time of solving mazes.

As it is shown in the charts above the A\* algorithm is way better than the other ones, it always solve the maze in a short time, the wall follower algorithm otherwise solves the maze but takes a lot of time, the reason for that returns to its logic, as example even if the exit is ahead of the robot but there is a left turn before the exit, it will turn left. The random mouse algorithm takes random choices for paths, so sometimes it solve the maze in short time and other times it will take a long time, and that makes its average time always between the times of the two other algorithms.

### **III.8. Conclusion**

In this chapter the system was detailed by explaining the program, and showing all the components of the interface, it worked great because the environment is always random and the simulation worked fine.

The tests have showed exactly the result hoped for, and revealed the best algorithm of the three implemented ones.

## ***General conclusion***

## **General conclusion**

In conclusion, this report walked through the various world of the artificial intelligence and its uses in different fields, also the robotics and their evolving through time, then the appearance of the robots and the idea of using them as a method for AI researches. And then introduced the mazes and the problem of solving it and the idea of using the AI with robotics in that problem.

To simulate a maze solving robot it was necessary to propose a system architecture in order to implement it in NetLogo 3D, which is the program and the programming language used to simulate the robot.

To solve a maze, a maze need to be generated and like it was mentioned in chapter I and chapter II there was many algorithms for generating mazes and the choice was “recursive back-tracker algorithm”.

Solving the maze by a robot needed an algorithm, and the best choice was “A\* path finding algorithm” it was improved to fit the proposed architecture, but a single algorithm is not enough for comparing performance, so an algorithm that imitates the human behavior was proposed which is “random mouse algorithm”, and another more intelligent one which is “wall follower algorithm”.

After implementing the system, the simulation worked exactly as it was hoped for, and the results were promising, where the improved A\* algorithm gave always the minimum time in different mazes compared to the other algorithms.

At the end, may be the algorithm need more improvement and we are willing to do that, but it fulfilled its purpose for now. And because it solved the maze in short time, we are trying to implement the optimized A\* algorithm in a real robot, to see its performance in a real maze, hoping that it gives similar results to that of the simulation.

## **Bibliography**

- [1]: Stuart J. Russell, Peter Norvig, Artificial Intelligence. A Modern Approach, 2009.
- [2]: mccorduck pamela, machines who think, 2004.
- [3]: Oxford Dictionaries, 2011.
- [4]: Isaac Asimov, I robot, 1950.
- [5]: Vikram Kapila, Introduction to robotics. 2011,  
<http://engineering.nyu.edu/mechatronics/smart/pdf/Intro2Robotics.pdf>, 16/04/2017.
- [6]: Oxford Dictionary.
- [7]: NetLogo user manual, <https://ccl.northwestern.edu/netlogo/docs/>, 30/03/2017.