

République Française
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

Université Gustave Eiffel



REPORT FOR MULTI AGENT SYSTEMS

STRIPS planner



Réalisé

Par

Dhia Aderssa

Omar M'rad

Riadh Rabti

Encadrant : M.Mahdi Zargayouna

M2 SYSTEME INTELLIGENTS ET APPLICATIONS

Année Universitaire : 2023-2024

Table des matières

1	General presentation of the project	1
1.1	Academic context :	1
1.2	Project Context :	1
2	Project objectives	2
3	Project decomposition	3
3.1	Project architecture :	3
3.1.1	Package warehouse	3
3.1.2	Package	3
3.1.3	Robot	3
3.1.4	Actions and preconditions	3
4	Use case / output	4
4.1	Use case	4
4.2	Output	4
5	Challenges encountered	5
6	Tools	5
6.1	Visual studio	5
6.2	Python	5
6.3	Django	6
6.4	Github	6
7	Axes of improvement	7
8	Conclusion and Perspectives	8
	Webographie9	

Table des figures

1 General presentation of the project

1.1 Academic context :

As part of the course entitled “Multi Agent systems“, we are divided into groups of three in order to implement what we have learned throughout this course through a project. This year’s project entitled “ STRIPS planner”.

1.2 Project Context :

“ STRIPS planner” is a simulation where the goal is to design and create a world with multiple actors"agents" . the world is characterised by an initial state ,a goal to reach ,agents and action that these actors can make and preconditions that need to be fulfilled to execute these actions.

2 Project objectives

Create an application that :

- Can take as an input a text file and deduce the different components such as initial state , actors , actions and goal
- Creates a world with an initial state and different actors
- Verifies the actor and the actions that it can take as well as the preconditions needed
- Follows a specific search strategy(depth-first,breadth-first ..) to try different simulations to achieve the goal .
- Shows the different steps followed in order to go from the initial state to the final state

3 Project decomposition

We chose as a simulation scenario for our application a Amazon packages sorting robot , with the goal of searching for the correct package and moving it to the correct shelf . in our simulation we tried both search methods the depth first and the breadth first.

3.1 Project architecture :

3.1.1 Package warehouse

The amazon warehouse is the general environment in which different actors(robot,package,shelves) are present and interact with each other .

3.1.2 Package

The amazon warehouse is the general environment in which different actors(robot,package,shelves) are present and interact with each other .

3.1.3 Robot

The robot is a main actor in our scenario its role is to search for the correct package and transport it from one shelf to the next according to the predetermined goal .

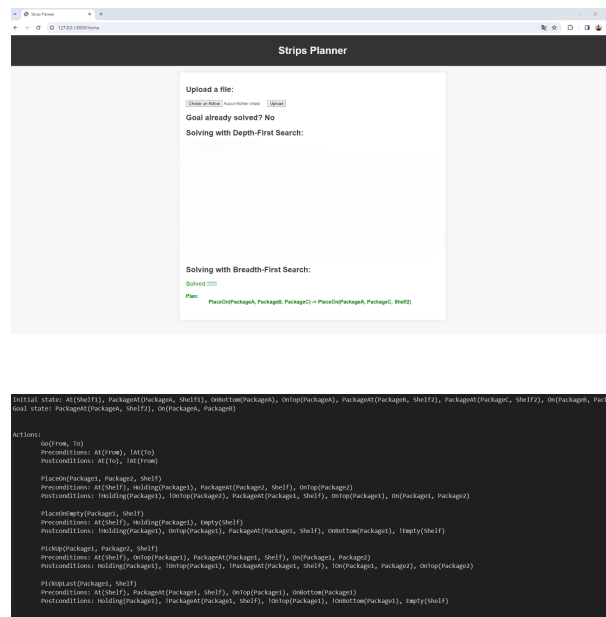
3.1.4 Actions and preconditions

The robot can choose from a variety of actions in order to achieve its goal , for instance it can move from one shelf to another and it can pickup and put packages at different locations. However, each action is preceded by a set of preconditions in order to execute that action as an example if the robot wants to pick up a package A , it need to verify that at that location there is in fact a package A .

4 Use case / output

4.1 Use case

We chose as input for our program a text file with the following instructions .However you can upload any scenario of your choice and we even added a few examples to chose from in the github link.



4.2 Output

- We used two different search strategies the first is depth-first and we have successfully reached the predetermined goal with these following steps shown in the interface

Solving with Depth-First Search:

Solved !!!!!

[illegible]

- We used breadth first search strategie and we have successfully reached the predetermined goal However the steps are hidden in this case only the output is shown

```
Solving with Breadth-First Search:  
Solved !!!!!  
Plan: PlaceOn(PackageA, PackageB, PackageC) -> PlaceOn(PackageA, PackageC, Shelf2)
```

5 Challenges encountered

- The choice of Tools and IDE best suitable for the job
- The choice of a Graphic interface and modifying to suit the specific use cases of the project
- Integrating the components (Robot , packages , shelves) in the same environment and ensuring their communication and reaching the desired goal.

6 Tools

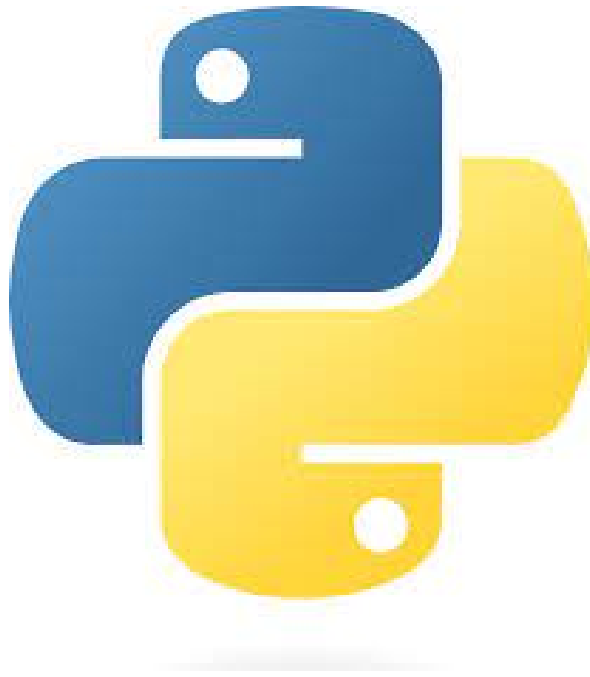
6.1 Visual studio

Visual studio is an integrated development environment for Python programming. It is included with the standard installation of Python and provides a convenient and interactive environment for writing, testing, and debugging Python code. IDLE is designed to be simple and easy to use, making it suitable for beginners as well as experienced developers.



6.2 Python

We used the language STRIPS which is an extension of python as instructed in the project description .



6.3 Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. We chose Django to create a web interface that will show the steps taken from initial state to final state.

django

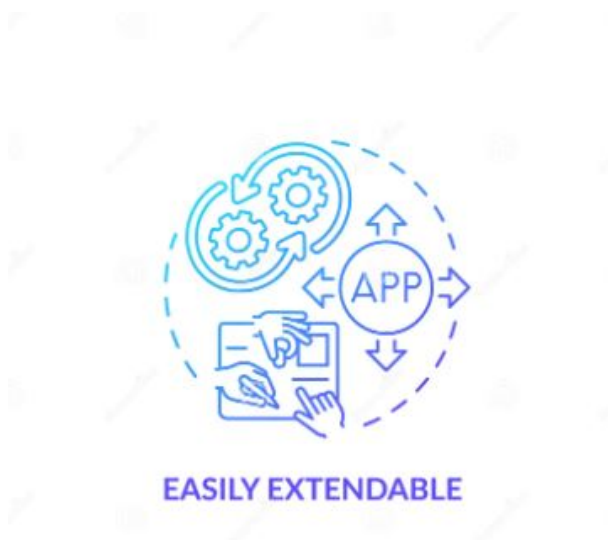
6.4 Github

is a platform and cloud-based service for software development and version control, allowing developers to store and manage their code. We used GitHub as a version control that allowed us to parallel and distribute tasks and facilitated merging the different versions and features.



7 Axes of improvement

- Introducing a reinforcement learning component to help it learn and make better choices.
- We can introduce IOT in this ecosystem in a way to give the robot more information leading to more efficient decision making.
- This program is capable of handling different scenario's therefore implementation potential is substantial.



8 Conclusion and Perspectives

Thanks to different breakthroughs in technologies, humanity was able create innovative solutions that automate and facilitate everyday tasks such as our project . However, these applications are always evolving and innovating and impacting positively humanity in all aspects and fields. .