

Département Génie Electrique

# Avancement du projet IA

## « IAvoid »

Le 25/12/2025

Réalisé par: **SOUADI Nouamane**

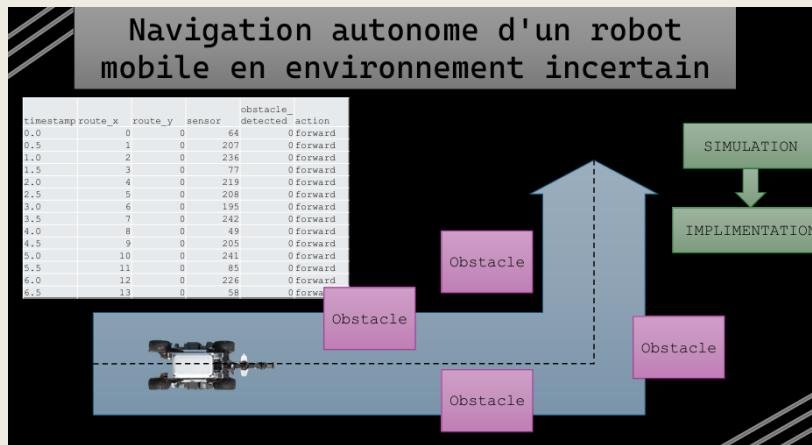
Encadré par: **AZZAM JAI Asmae**

Année universitaire : 2025/2026

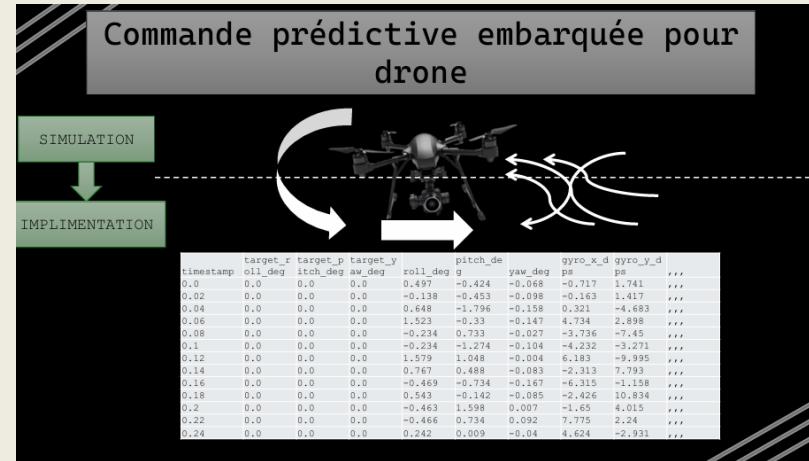
# Choix du sujet

Proposition de deux sujets concernant: **La navigation autonome d'un robot mobile en environnement incertain et la commande prédictive embarquée pour un drone**

## SUJET1:

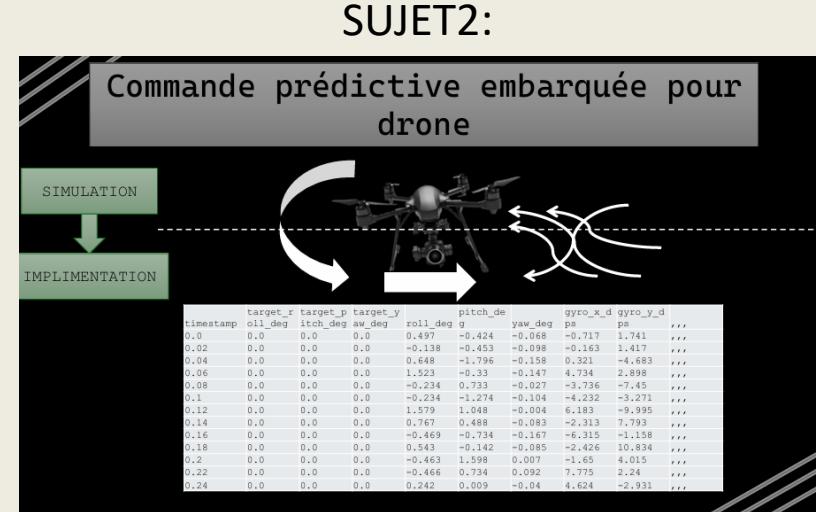
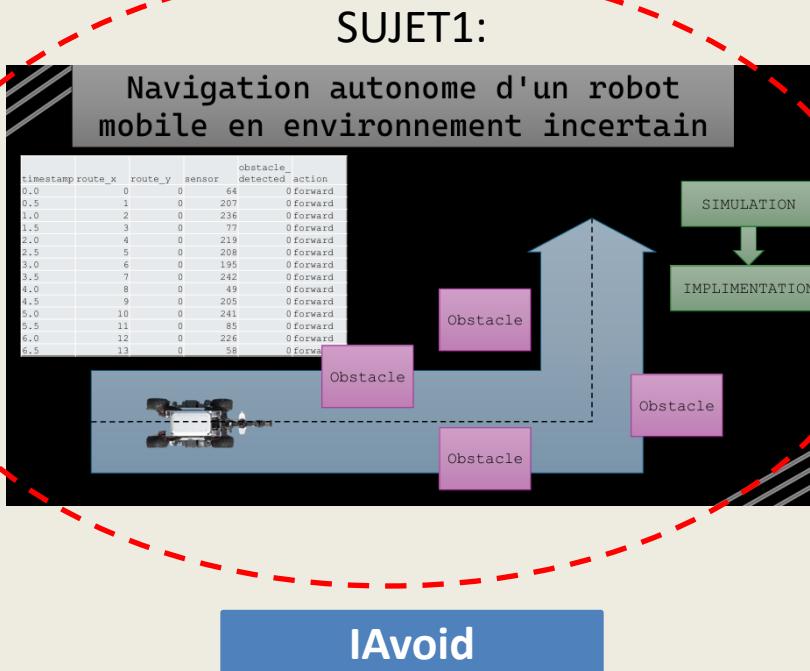


## SUJET2:



# Choix du sujet

Choix du premier sujet. Sujet avec matériel plus accessible et implémentation d'algorithme d'apprentissage renforcé



Prochain projet...

# C'est quoi IAvoid ?

## IAvoid

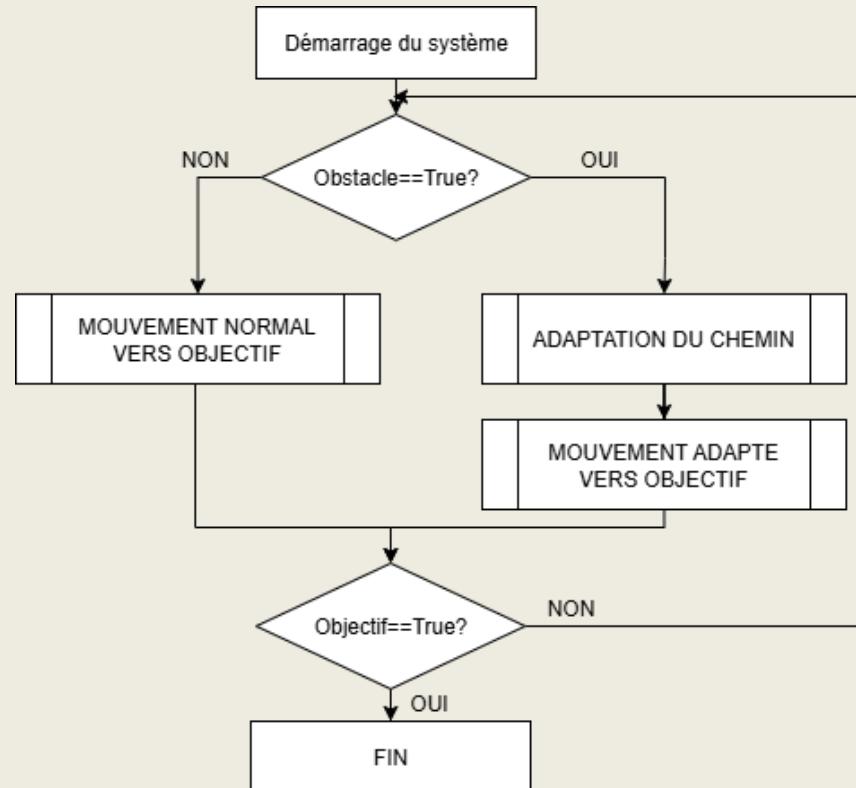
### IAvoid

C'est un robot concis pour un environnement industriel incertain (ex. déplacer des charges lourdes) capable **d'agir face à des obstacles**

### IAvoid

Le processus de traitement de l'action est 100% basé sur l'IA (Intelligence Artificiel) afin de modifier le chemin optimal à **un chemin efficace contre les obstacles**

# Principe de fonctionnement général



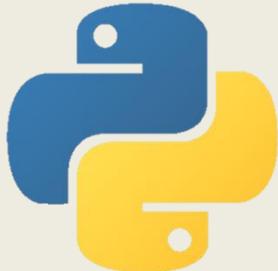
# Ressources et outils

Démarrage sans base de données  
(mais avec un chemin préféré)

Carte d'acquisition  
(pas encore choisie)

Apprentissage renforcé  
(utilisation en premier lieu de  
Q-Learning algorithm)

Stockage d'historique des actions  
(analyse des paramètres)

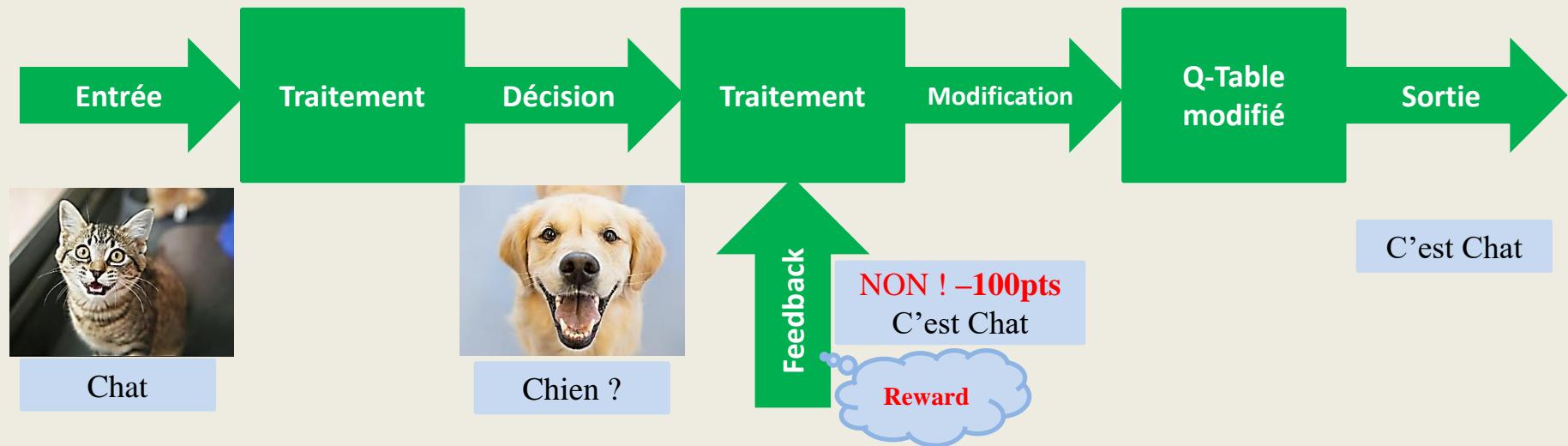


**pygame**  
**numpy**  
**random**  
**pandas**

Visualisation du processus d'apprentissage  
Calcul matriciel  
Génération aléatoire des obstacles  
Gestion des BD et stockage sur Excel

# C'est quoi le Q-Learning Algorithm ?

Le **Q-Learning algorithm** est un algorithme d'apprentissage renforcé populaire qui aide un système à prendre des décisions face à des nouvelles entrées en se basant sur son '**expérience**' des actions passées et leurs résultats.



# C'est quoi le Q-Learning Algorithm ?

## Q-TABLE

A fur et à mesure l'algorithme actualise le **Q-Table** et prend la décision avec le minimum des pertes (maximiser les récompenses)

$$Q(S,A) \leftarrow Q(S,A) + \alpha(R + \gamma Q(S', A') - Q(S,A))$$

**S:** l'état actuel.

**A:** l'action entreprise par l'agent.

**S':** l'état suivant vers lequel l'agent se déplace.

**A':** la meilleure action suivante dans l'état S'.

**R:** la récompense reçue pour avoir pris l'action A dans l'État S.

**$\gamma$  (Gamma):** le facteur d'actualisation qui équilibre les récompenses immédiates avec les récompenses futures.

**$\alpha$  (Alpha):** le taux d'apprentissage déterminant la quantité de nouvelles informations affectant les anciennes valeurs Q.

# C'est quoi le Q-Learning Algorithm ?

## Politique $\epsilon$ -cupide

**Pour l'exploitation :** L'agent choisit l'action avec la valeur Q la plus élevée avec probabilité  $1-\epsilon$ . Cela signifie que l'agent utilise ses connaissances actuelles pour maximiser les récompenses.

**Pour l'exploration :** Avec probabilité  $\epsilon$ , l'agent choisit une action au hasard, explorant de nouvelles possibilités pour savoir s'il existe de meilleures façons d'obtenir des récompenses. Cela permet à l'agent de découvrir de nouvelles stratégies et d'améliorer sa prise de décision au fil du temps.

## Equation de Bellman

$$Q(s,a) = R(s,a) + \gamma \max(Q(s',a))$$

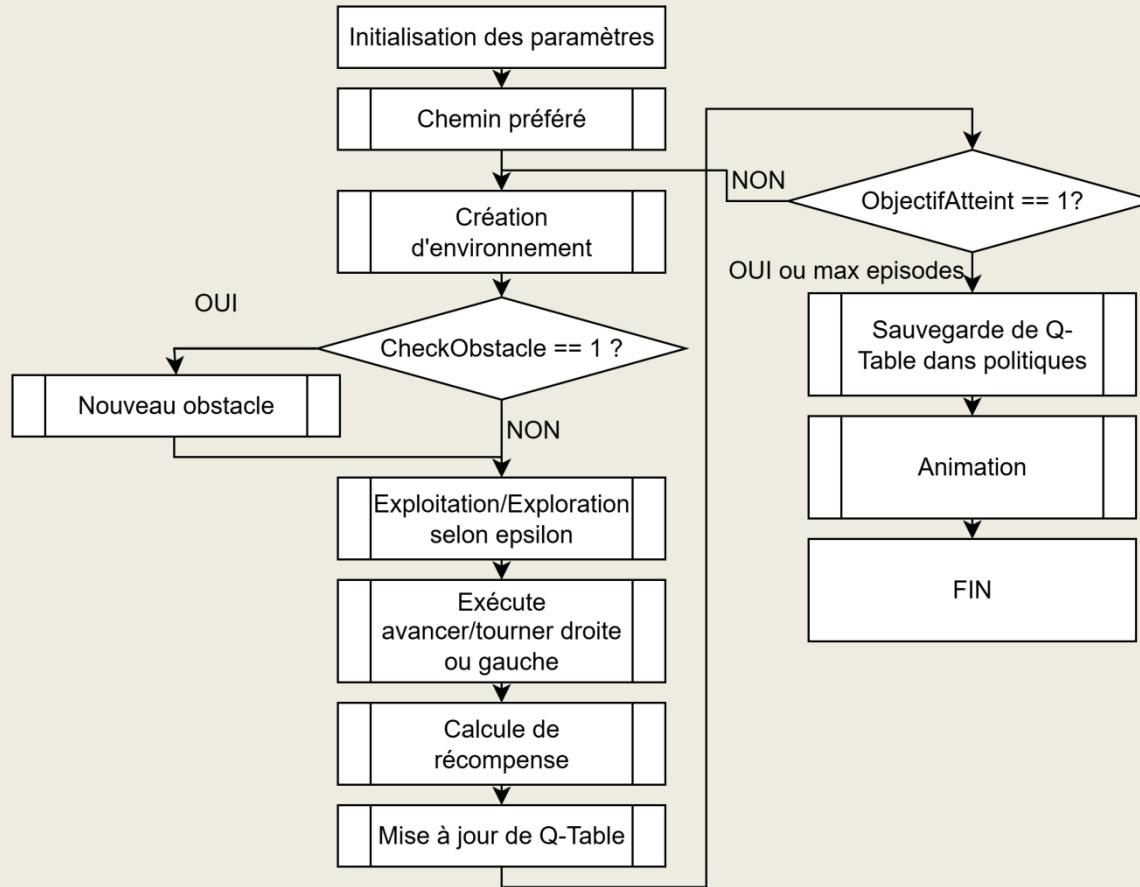
**Q(s, a)** est la valeur Q d'une paire état-action donnée.

**R(s, a)** est la récompense immédiate pour avoir pris des mesures a dans l'état s.

**$\gamma$**  est le facteur d'actualisation, représentant l'importance des récompenses futures.

**$\max(Q(s',a))$**  est la valeur Q maximale pour **l'état suivante** de toutes les actions possibles.

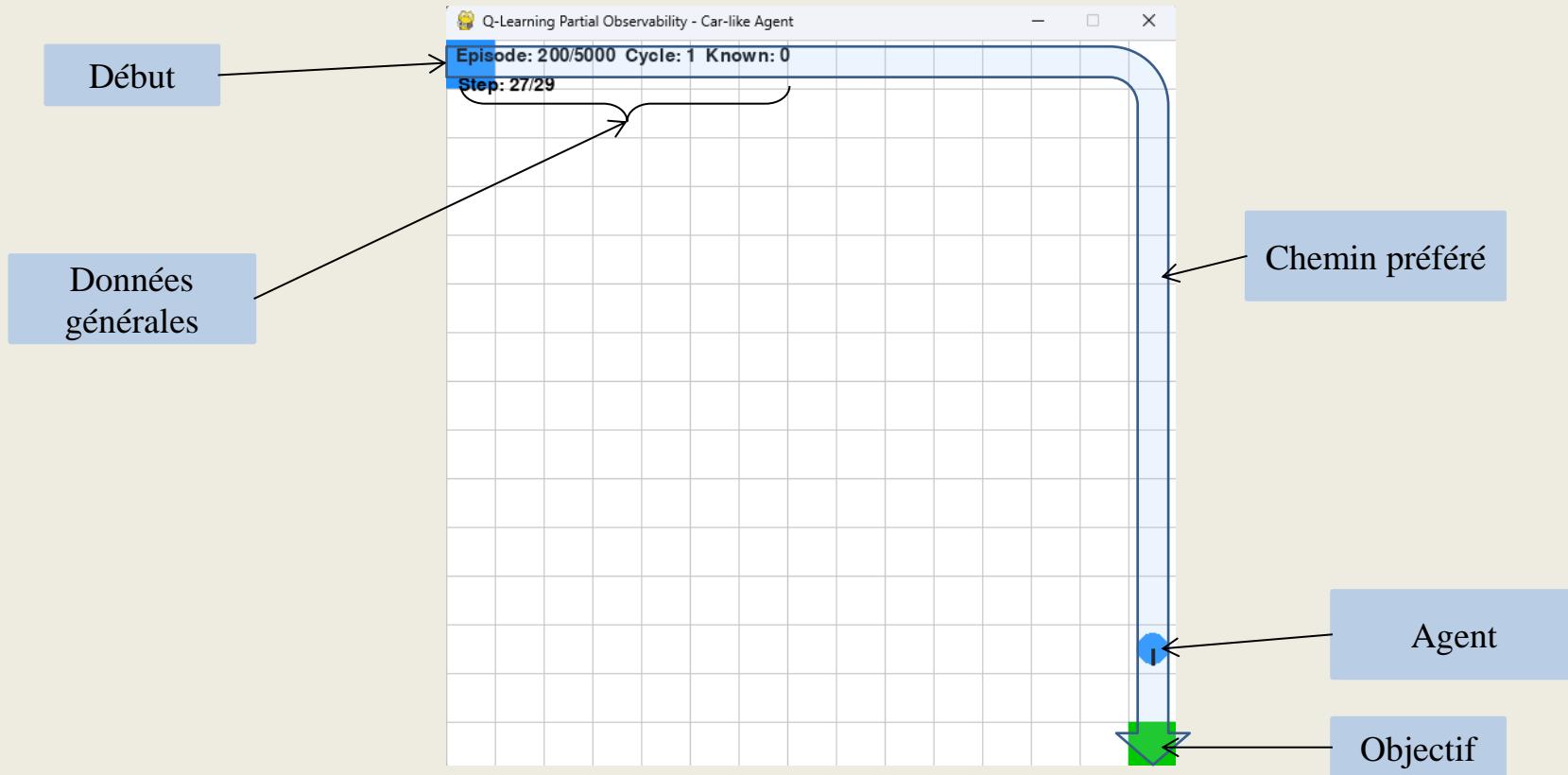
# Principe de fonctionnement du code Python



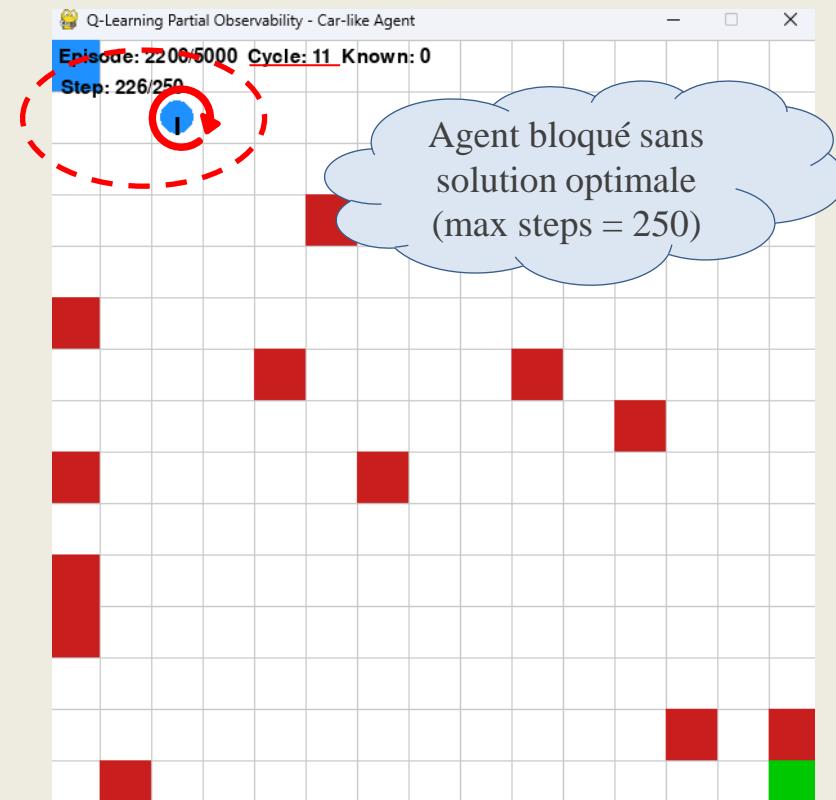
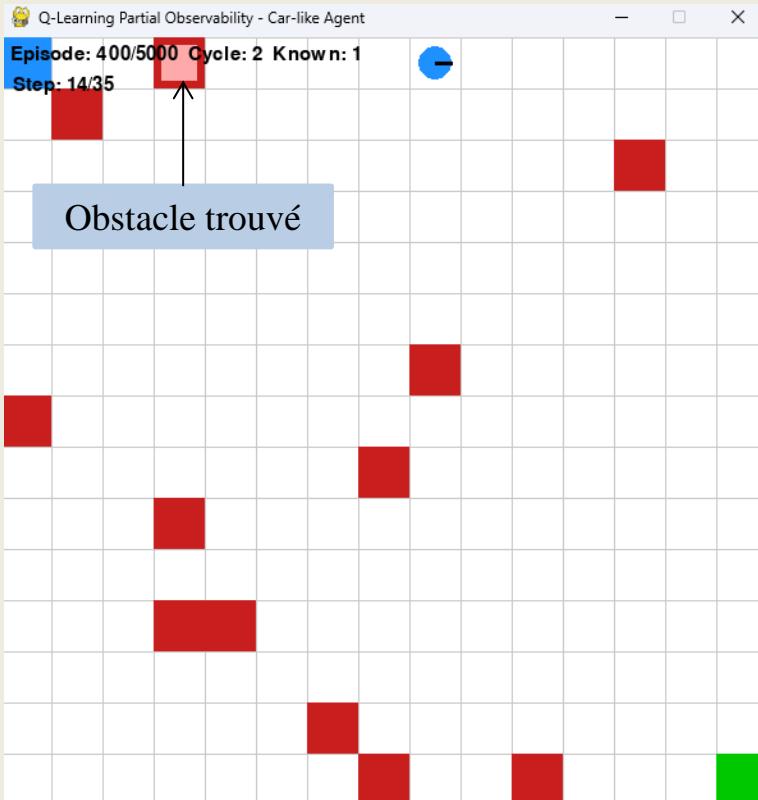
# Paramètres génériques de première simulation

```
# -----
# Hyperparameters
# -----
learning_rate = 0.3
discount_factor = 0.95
epsilon = 0.5
epsilon_min = 0.01
epsilon_decay = 0.997
num_episodes = 5000
grid_size = 15
obstacle_change_interval = 200
max_steps_per_episode = 250
cell_size = 40
screen_size = grid_size * cell_size
tick_time = 60
```

# Environnement de simulation - Pygame

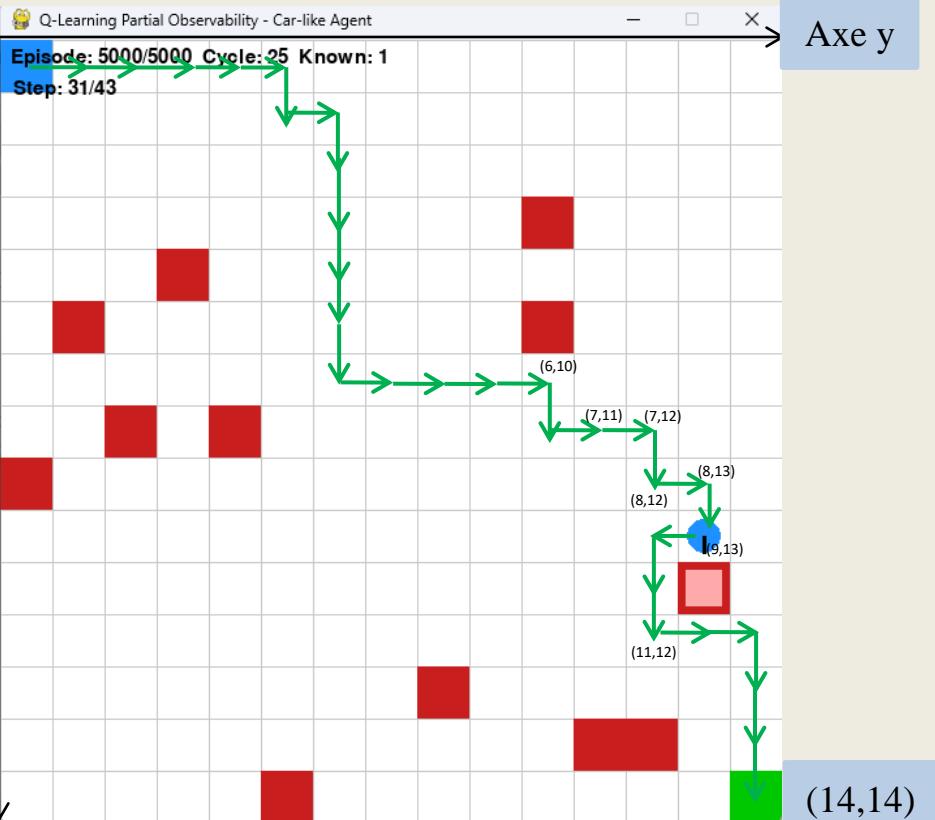


# Environnement de simulation - Simulation



# Environnement de simulation - Fin de simulation

(0,0)



(0,0)

Historique des positions enregistrées  
Cycle 25, Episode 5000  
(dernière itération)

(x,y, direction)	(6,7,East);	(10,12,South);
(0,0,East);	(6,8,East);	(11,12,South);
(0,1,East);	(6,9,East);	(11,12,East);
(0,2,East);	(6,10,East);	(11,13,East);
(0,3,East);	(6,10,South);	(11,14,East);
(0,4,East);	(7,10,South);	(11,14,South);
(0,5,East);	(7,10,East);	(12,14,South);
(0,5,South);	(7,11,East);	(13,14,South);
(1,5,South);	(7,12,East);	(14,14,South);
(1,5,East);	(7,12,South);	
(1,6,East);	(8,12,South);	
(1,6,South);	(8,12,East);	
(2,6,South);	(8,13,East);	
(3,6,South);	(8,13,South);	
(4,6,South);	(9,13,South);	
(5,6,South);	(9,13,West);	
(6,6,South);	(9,12,West);	
(6,6,East);	(9,12,South);	

43 steps

# Environnement de simulation - Fin de simulation

## Stockage pour analyse ultérieure

```

Policy facing West (S=start, G=goal, X=obst, F/L/R=actions):
S L F L F L R L R L L F F L L
L R F R L L L F F F F F L F L
R R L F L F L L R R R F L F L
L R L R F L L L F F X R F L F
R L R X F R L L R R L F F L F
L X F L L R L R F F X L R L F
R L L L L R L L R R R L L L F
L F X L X F R R F F F L L F
X F R R L R L R L R F R F L F
L F R R L R R L L R L R L F F
R R R R R R L R R R R R R X L
L R L R F R F L R R F R F R L
L L R L R L R L X R F R R R L
L L L L R R R L R R L X X L R
L F R L R X R L R R L R L F G
Training finished. Saving all snapshots and episode data.
Policies appended to 'policy_database.xlsx'.
Episode data saved to 'episode_data.xlsx'.

```

episode\_data.xlsx

episode	cycle	path	total_reward	steps	reached_goal	obstacle_config
2	1	1 (0,0,East);0,0,North	-290	250	FAUX	()
3	2	1 (0,0,East);0,1,East	-302	250	FAUX	()
4	3	1 (0,0,East);0,1,East	-168	229	VRAI	()
5	4	1 (0,0,East);0,1,East	-133	210	VRAI	()
6	5	1 (0,0,East);0,1,East	-278	250	FAUX	()
7	6	1 (0,0,East);0,1,East	-69	150	VRAI	()
8	7	1 (0,0,East);0,0,North	-81	138	VRAI	()
9	8	1 (0,0,East);0,1,East	-258	250	FAUX	()
10	9	1 (0,0,East);0,0,South	-60	145	VRAI	()
11	10	1 (0,0,East);0,0,North	-16	101	VRAI	()
12	11	1 (0,0,East);0,1,East	-1	74	VRAI	()
13	12	1 (0,0,East);0,1,East	-20	113	VRAI	()
14	13	1 (0,0,East);0,1,East	21	68	VRAI	()
15	14	1 (0,0,East);0,1,East	-72	173	VRAI	()
16	15	1 (0,0,East);0,1,East	-274	250	FAUX	()
17	16	1 (0,0,East);0,1,East	13	84	VRAI	()
18	17	1 (0,0,East);0,0,South	-88	173	VRAI	()
19	18	1 (0,0,East);0,1,East	23	74	VRAI	()
20	19	1 (0,0,East);0,1,East	-262	250	FAUX	()
21	20	1 (0,0,East);0,1,East	-61	150	VRAI	()
22	21	1 (0,0,East);0,1,East	-20	113	VRAI	()
23	22	1 (0,0,East);0,1,East	33	64	VRAI	()
24	23	1 (0,0,East);0,1,East	32	69	VRAI	()
25	24	1 (0,0,East);0,1,East	-49	122	VRAI	()
26	25	1 (0,0,East);0,1,East	-40	125	VRAI	()
27	26	1 (0,0,East);0,1,East	-274	250	FAUX	()
28	27	1 (0,0,East);0,1,East	-66	143	VRAI	()
29	28	1 (0,0,East);0,1,East	11	86	VRAI	()
30	29	1 (0,0,East);0,1,East	-74	167	VRAI	()
31	30	1 (0,0,East);0,0,South	-75	148	VRAI	()
32	31	1 (0,0,East);0,1,East	-10	95	VRAI	()

EpisodeData

Policy\_database.xlsx

1	x	y	facing	action	stacle_config
2	0	0	North	Start	((2, 2), (3, 3), (4, 10),
3	0	0	East	Start	((2, 2), (3, 3), (4, 10),
4	0	0	South	Start	((2, 2), (3, 3), (4, 10),
5	0	0	West	Start	((2, 2), (3, 3), (4, 10),
6	0	1	North	turn_left	((2, 2), (3, 3), (4, 10),
7	0	1	East	forward	((2, 2), (3, 3), (4, 10),
8	0	1	South	turn_right	((2, 2), (3, 3), (4, 10),
9	0	1	West	forward	((2, 2), (3, 3), (4, 10),
10	0	2	North	turn_right	((2, 2), (3, 3), (4, 10),
11	0	2	East	turn_right	((2, 2), (3, 3), (4, 10),
12	0	2	South	forward	((2, 2), (3, 3), (4, 10),
13	0	2	West	turn_left	((2, 2), (3, 3), (4, 10),
14	0	3	North	turn_right	((2, 2), (3, 3), (4, 10),
15	0	3	East	turn_left	((2, 2), (3, 3), (4, 10),
16	0	3	South	turn_right	((2, 2), (3, 3), (4, 10),
17	0	3	West	forward	((2, 2), (3, 3), (4, 10),
18	0	4	North	turn_left	((2, 2), (3, 3), (4, 10),
19	0	4	East	turn_right	((2, 2), (3, 3), (4, 10),
20	0	4	South	forward	((2, 2), (3, 3), (4, 10),
21	0	4	West	turn_left	((2, 2), (3, 3), (4, 10),
22	0	5	North	turn_right	((2, 2), (3, 3), (4, 10),
23	0	5	East	turn_left	((2, 2), (3, 3), (4, 10),
24	0	5	South	turn_right	((2, 2), (3, 3), (4, 10),
25	0	5	West	turn_left	((2, 2), (3, 3), (4, 10),
26	0	6	North	turn_right	((2, 2), (3, 3), (4, 10),
27	0	6	East	forward	((2, 2), (3, 3), (4, 10),

Config\_176\_12obs Config\_177\_12obs Config\_178\_12obs

# Comparaison entre début et fin de la simulation

Trié par total récompenses descendant

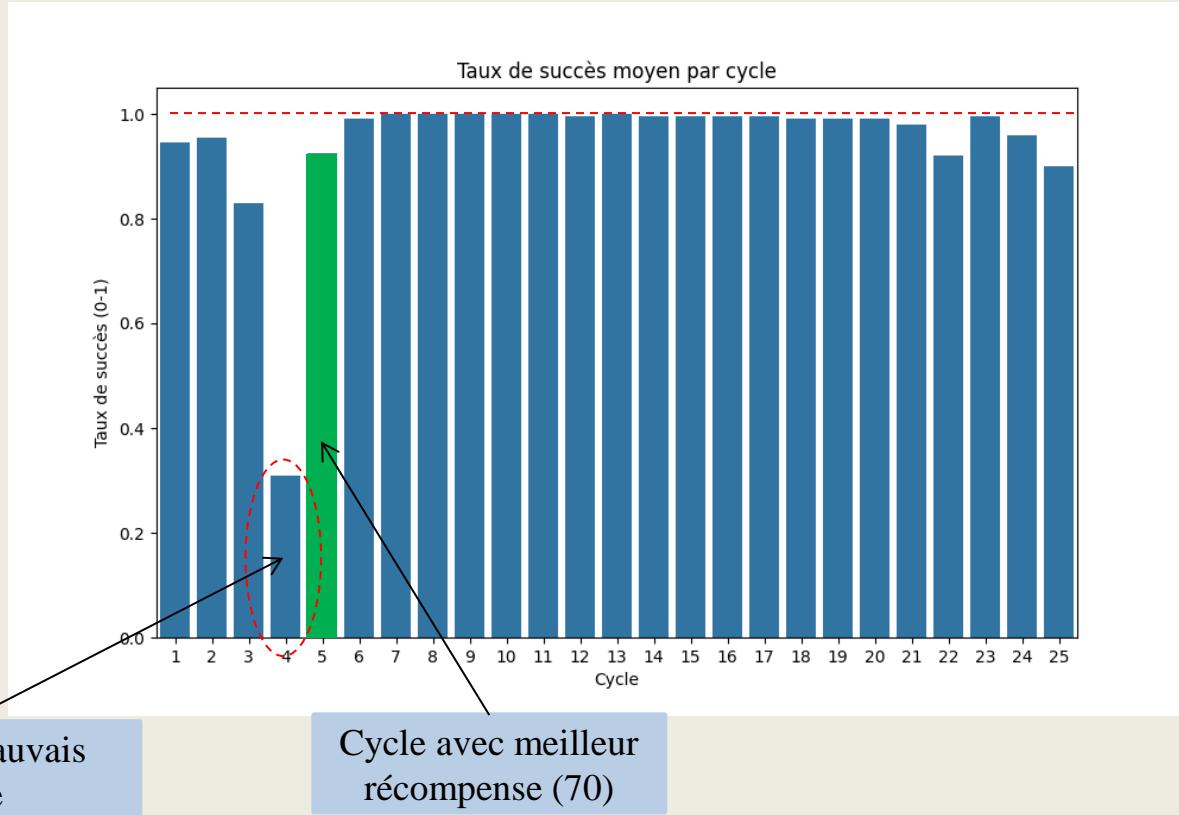
episode	cycle	path	total_reward	steps	reached_goal	obstacle_config
790	4	(0,0,East);(0,1,East)	48	53	VRAI	((1, 8), (1, 11), (4, 14))
727	4	(0,0,East);(0,1,East)	-40	61	VRAI	((1, 8), (1, 11), (4, 14))
783	4	(0,0,East);(0,1,East)	36	65	VRAI	((1, 8), (1, 11), (4, 14))
735	4	(0,0,East);(0,1,East)	32	69	VRAI	((1, 8), (1, 11), (4, 14))
744	4	(0,0,East);(0,1,East)	24	77	VRAI	((1, 8), (1, 11), (4, 14))
771	4	(0,0,East);(0,0,North)	24	77	VRAI	((1, 8), (1, 11), (4, 14))
781	4	(0,0,East);(0,1,East)	24	77	VRAI	((1, 8), (1, 11), (4, 14))
766	4	(0,0,East);(0,1,East)	16	85	VRAI	((1, 8), (1, 11), (4, 14))
755	4	(0,0,East);(0,0,South)	14	87	VRAI	((1, 8), (1, 11), (4, 14))
770	4	(0,0,East);(0,0,North)	12	89	VRAI	((1, 8), (1, 11), (4, 14))
732	4	(0,0,East);(0,1,East)	8	93	VRAI	((1, 8), (1, 11), (4, 14))
778	4	(0,0,East);(0,1,East)	8	93	VRAI	((1, 8), (1, 11), (4, 14))
791	4	(0,0,East);(0,0,North)	8	93	VRAI	((1, 8), (1, 11), (4, 14))
772	4	(0,0,East);(0,0,South)	6	95	VRAI	((1, 8), (1, 11), (4, 14))
760	4	(0,0,East);(0,0,South)	2	99	VRAI	((1, 8), (1, 11), (4, 14))
779	4	(0,0,East);(0,0,South)	2	90	VRAI	((1, 8), (1, 11), (4, 14))
730	4	(0,0,East);(0,1,East)	-5	93	VRAI	((1, 8), (1, 11), (4, 14))
756	4	(0,0,East);(0,1,East)	-6	107	VRAI	((1, 8), (1, 11), (4, 14))
653	4	(0,0,East);(0,0,North)	-12	105	VRAI	((1, 8), (1, 11), (4, 14))
736	4	(0,0,East);(0,1,East)	-16	117	VRAI	((1, 8), (1, 11), (4, 14))
793	4	(0,0,East);(0,1,East)	-16	117	VRAI	((1, 8), (1, 11), (4, 14))
725	4	(0,0,East);(0,1,East)	-18	119	VRAI	((1, 8), (1, 11), (4, 14))
729	4	(0,0,East);(0,1,East)	-20	121	VRAI	((1, 8), (1, 11), (4, 14))
795	4	(0,0,East);(0,1,East)	-22	123	VRAI	((1, 8), (1, 11), (4, 14))
799	4	(0,0,East);(0,0,North)	-22	123	VRAI	((1, 8), (1, 11), (4, 14))
759	4	(0,0,East);(0,0,North)	-24	125	VRAI	((1, 8), (1, 11), (4, 14))
764	4	(0,0,East);(0,0,North)	-32	124	VRAI	((1, 8), (1, 11), (4, 14))
731	4	(0,0,East);(0,1,East)	-34	135	VRAI	((1, 8), (1, 11), (4, 14))
782	4	(0,0,East);(0,0,South)	-37	134	VRAI	((1, 8), (1, 11), (4, 14))
774	4	(0,0,East);(0,1,East)	-41	138	VRAI	((1, 8), (1, 11), (4, 14))
665	4	(0,0,East);(0,0,North)	-44	145	VRAI	((1, 8), (1, 11), (4, 14))

Cycle 4

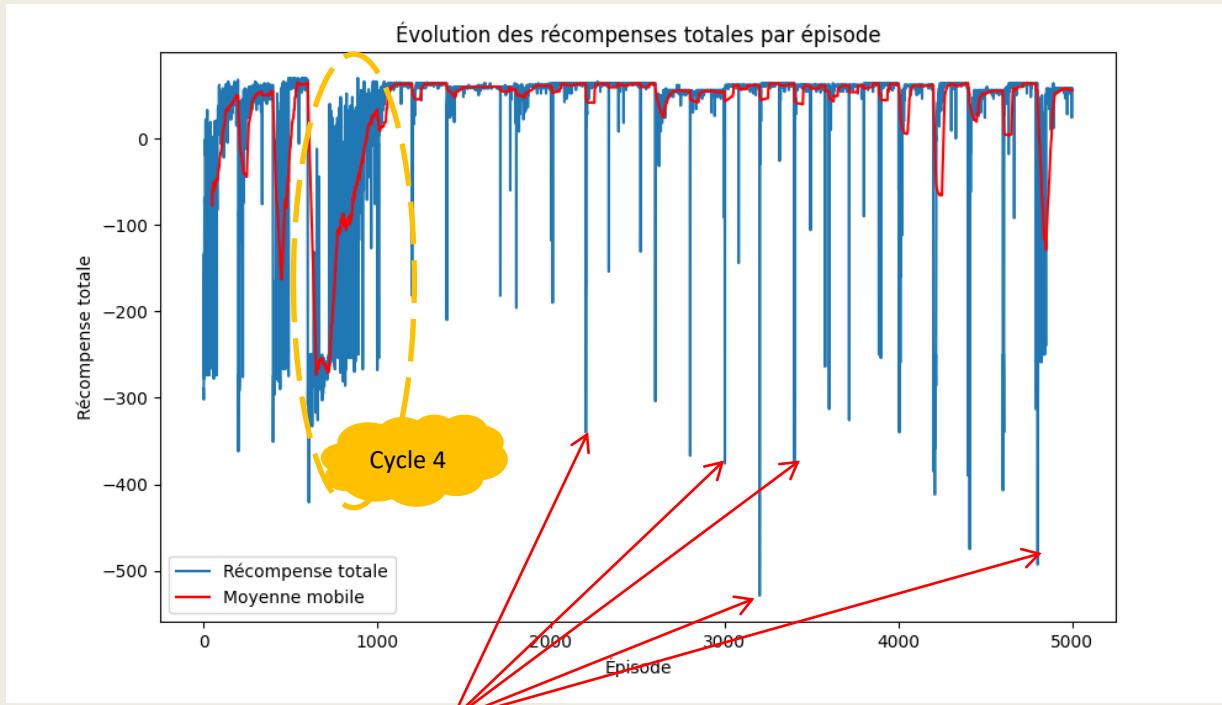
episode	cycle	path	total_reward	steps	reached_goal	obstacle_config
4891	25	(0,0,East);(0,1,East)	64	37	VRAI	((3, 10), (4, 3), (5, 5))
4852	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4861	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4862	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4863	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4864	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4865	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4866	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4883	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4887	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4888	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4890	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4897	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4899	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4900	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4901	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4902	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4904	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4906	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4907	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4908	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4909	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4912	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4913	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4915	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4916	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4917	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4918	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4919	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4920	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))
4921	25	(0,0,East);(0,1,East)	58	43	VRAI	((3, 10), (4, 3), (5, 5))

Dernier cycle

# Analyse des données de la simulation – Taux de succès par cycle

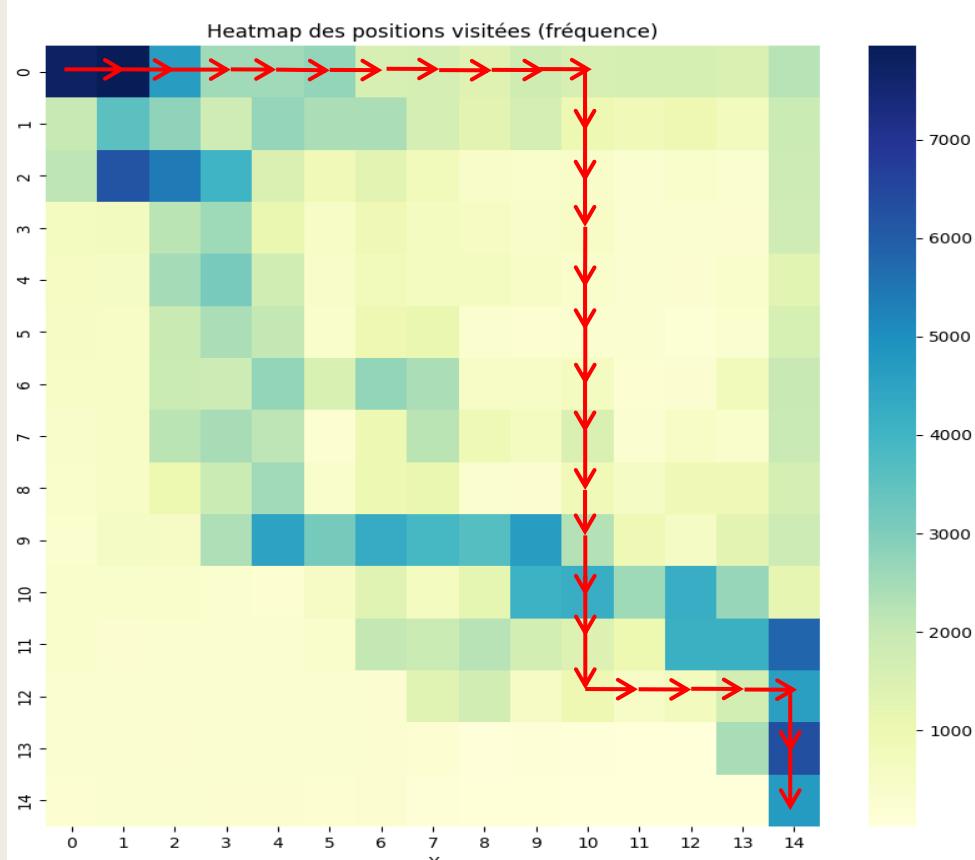


# Analyse des données de la simulation – Récompense totale par épisode



Pics représentants exactement les moments où l'agent est bloqué sans solution

# Analyse des données de la simulation – Heatmap et le chemin optimal

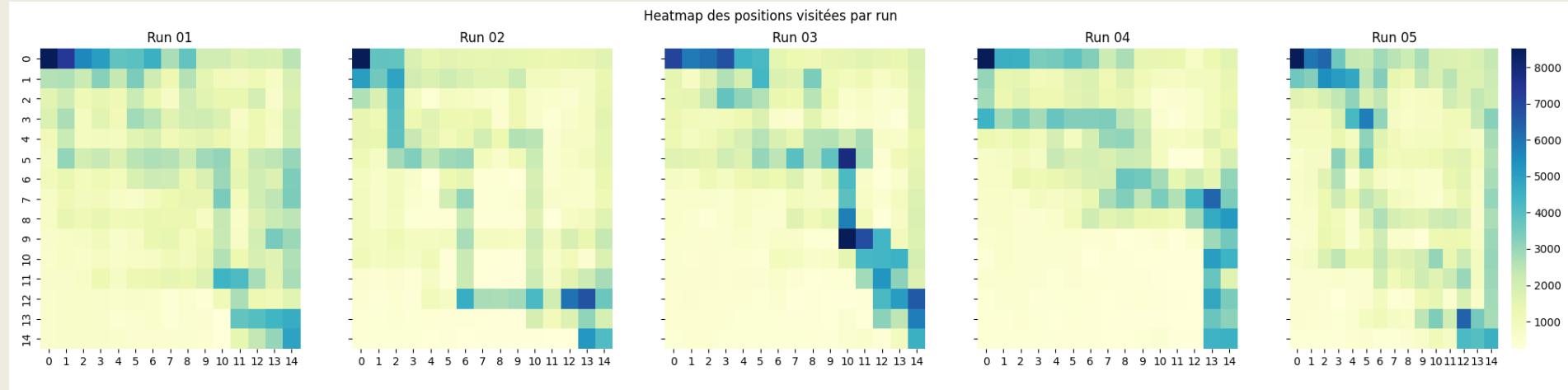


Historique des positions enregistrées  
Cycle 5, Episode 889/5000  
max reward dans la simulation (70)

(x,y, direction)
(0,0,East);
(0,1,East);
(0,2,East);
(0,3,East);
(0,4,East);
(0,5,East);
(0,6,East);
(0,7,East);
(0,8,East);
(0,9,East);
(0,10,East);
(0,10,South);
(1,10,South);
(2,10,South);
(3,10,South);
(4,10,South);
(5,10,South);

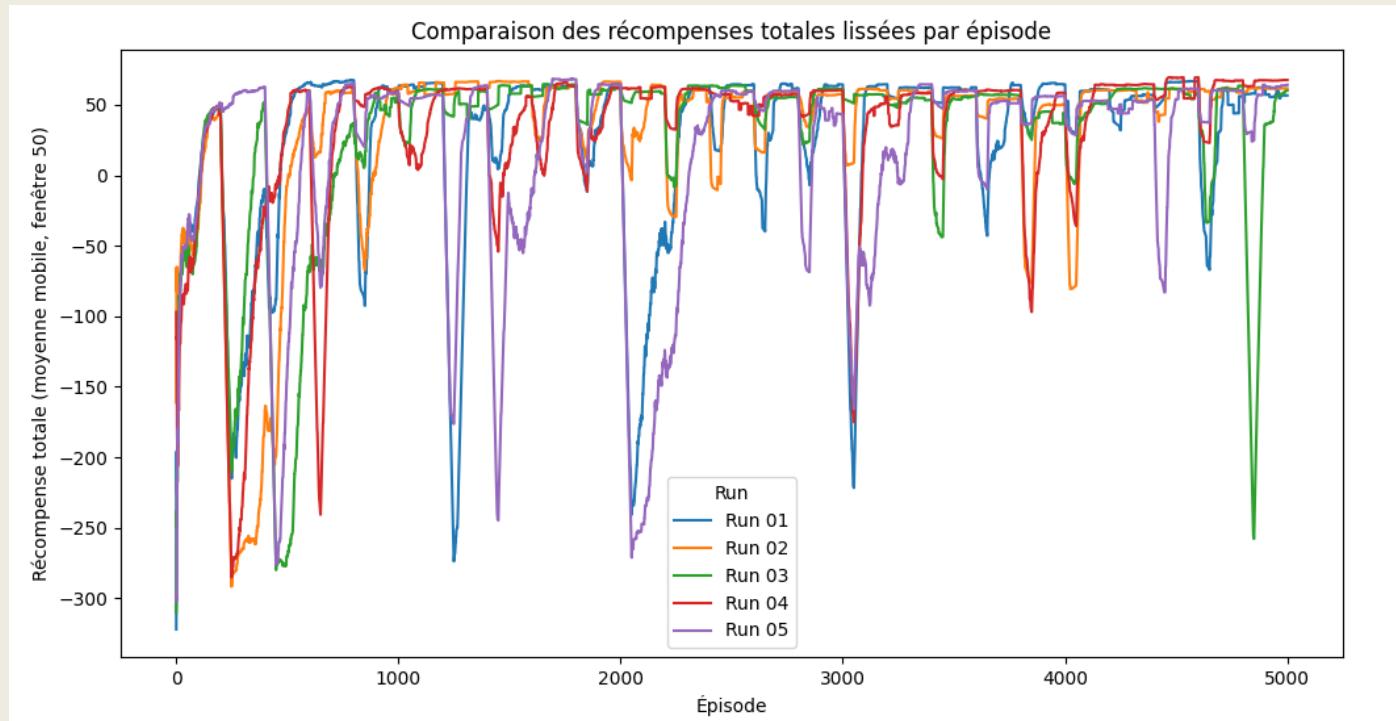
31 steps!

# Analyse entre 5 simulations avec même paramètres - Heatmap



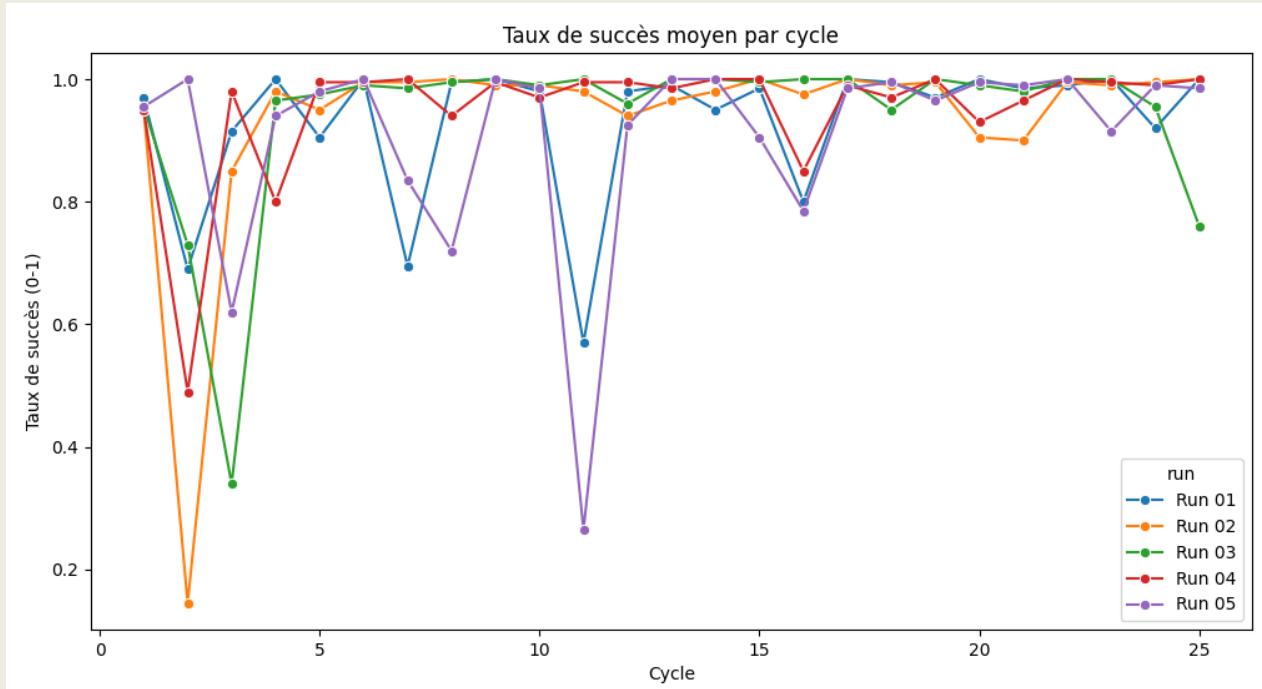
Même tendance des zones les plus fréquentées

# Analyse entre 5 simulations avec même paramètres – Récompenses totales



Sensible au moment du ‘**failure**’ du système (où l’agent est bloqué) selon l’emplacement des obstacles

# Analyse entre 5 simulations avec même paramètres – Taux de réussite



Sensible au moment du ‘failure’ du système (où l’agent est bloqué) selon l’emplacement des obstacles

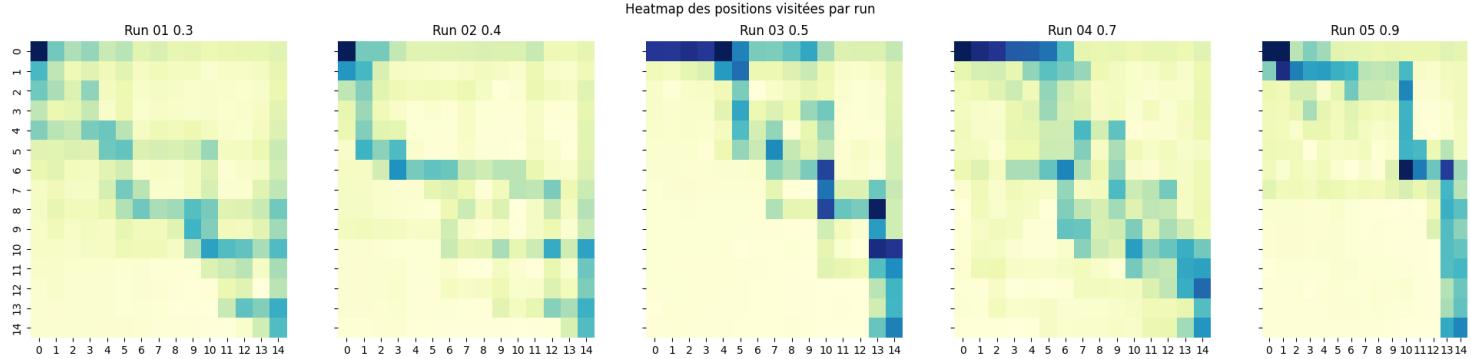
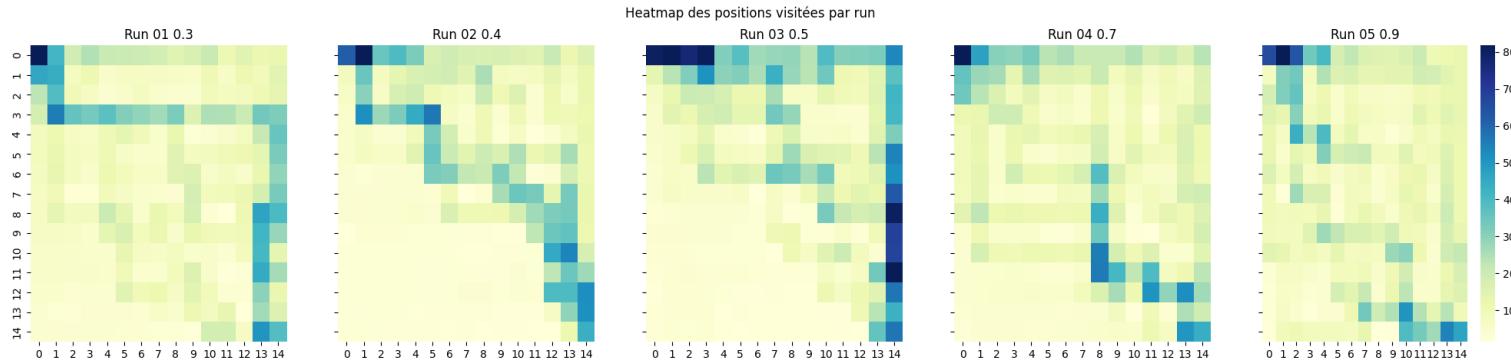
## Analyse entre 5 simulations avec même paramètres – Globalement

La réponse du système est sensible aux obstacles qui est une variable aléatoire incontrôlable (en nombre et emplacement) d'où le décalage entre les simulation mais on constate qu'il y a la même tendance d'apprentissage à travers le **Heatmap** et la même corrélation **récompenses/steps** et pente d'apprentissage similaire

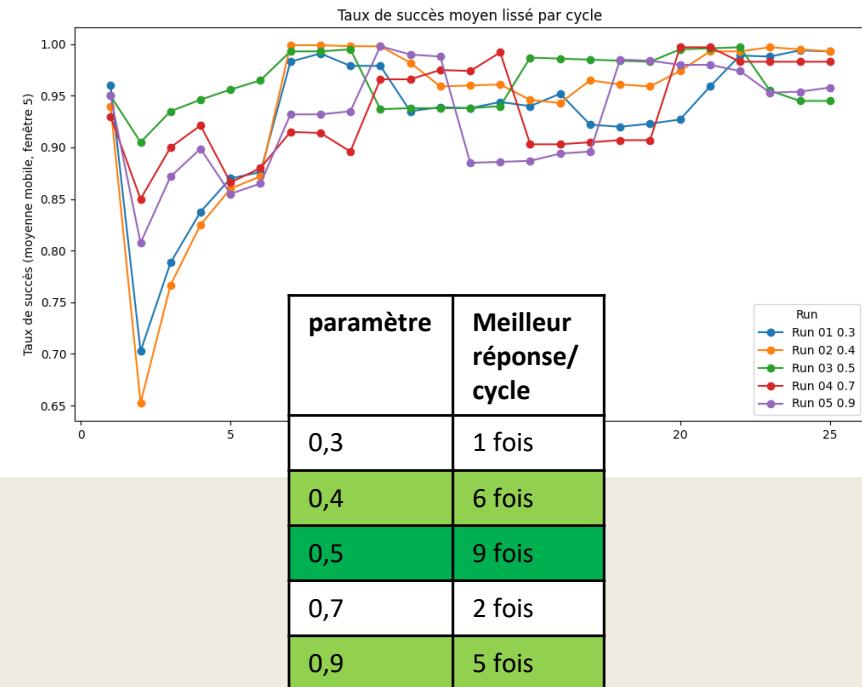
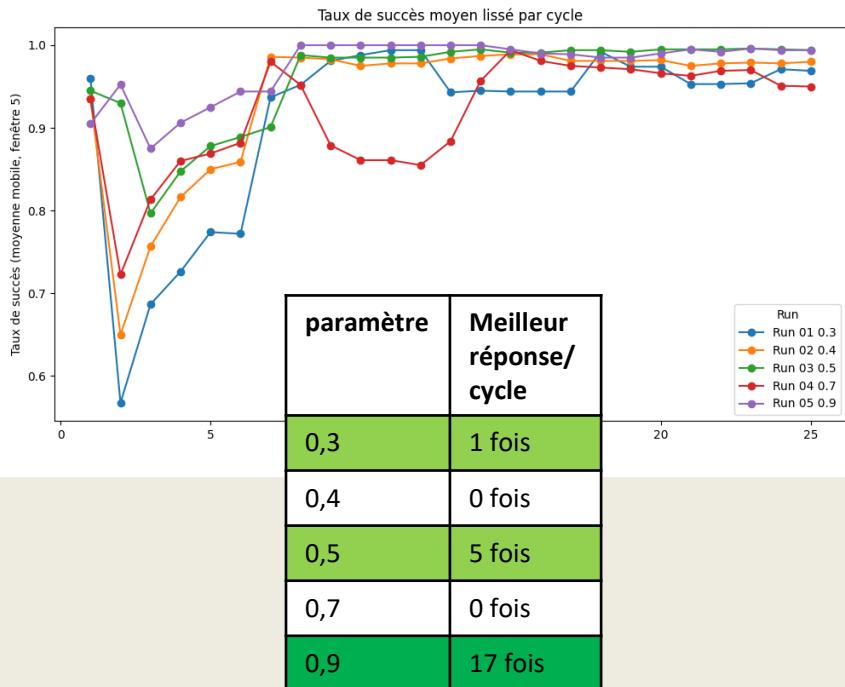
### Next step

Simulation d'effet de chaque paramètre fixé au début de la simulation sur l'apprentissage du système pour trouver un compromis pour l'efficacité du système en terme de rapidité et résolution du problème posé

## Comparaison des paramètres – Learning rate

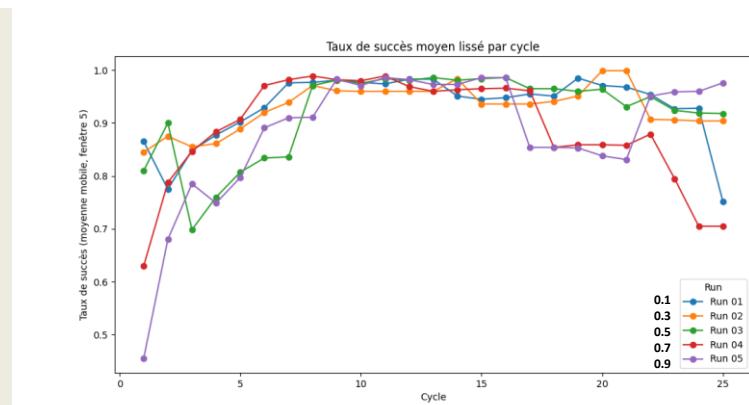
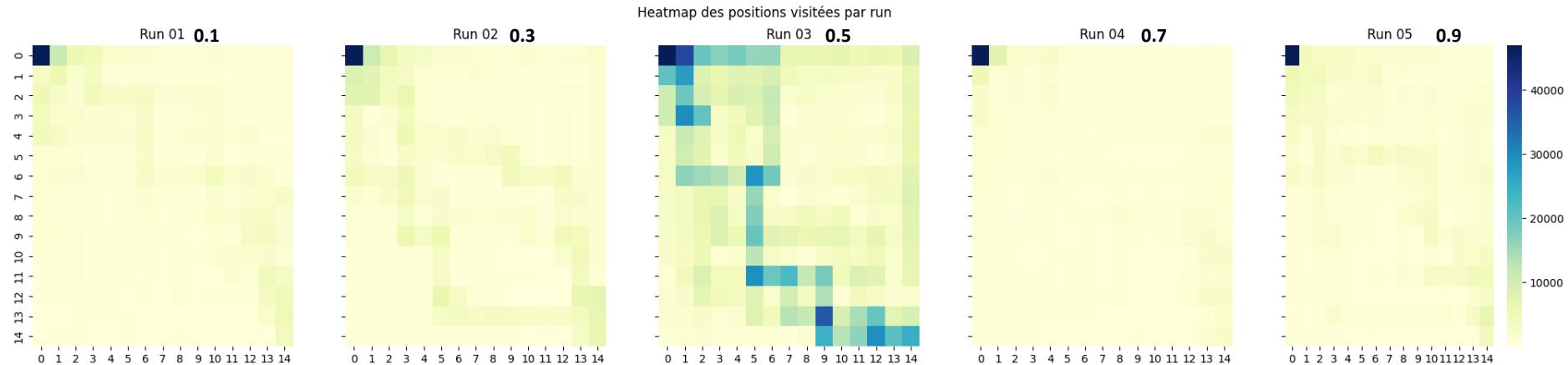


# Comparaison des paramètres – Learning rate

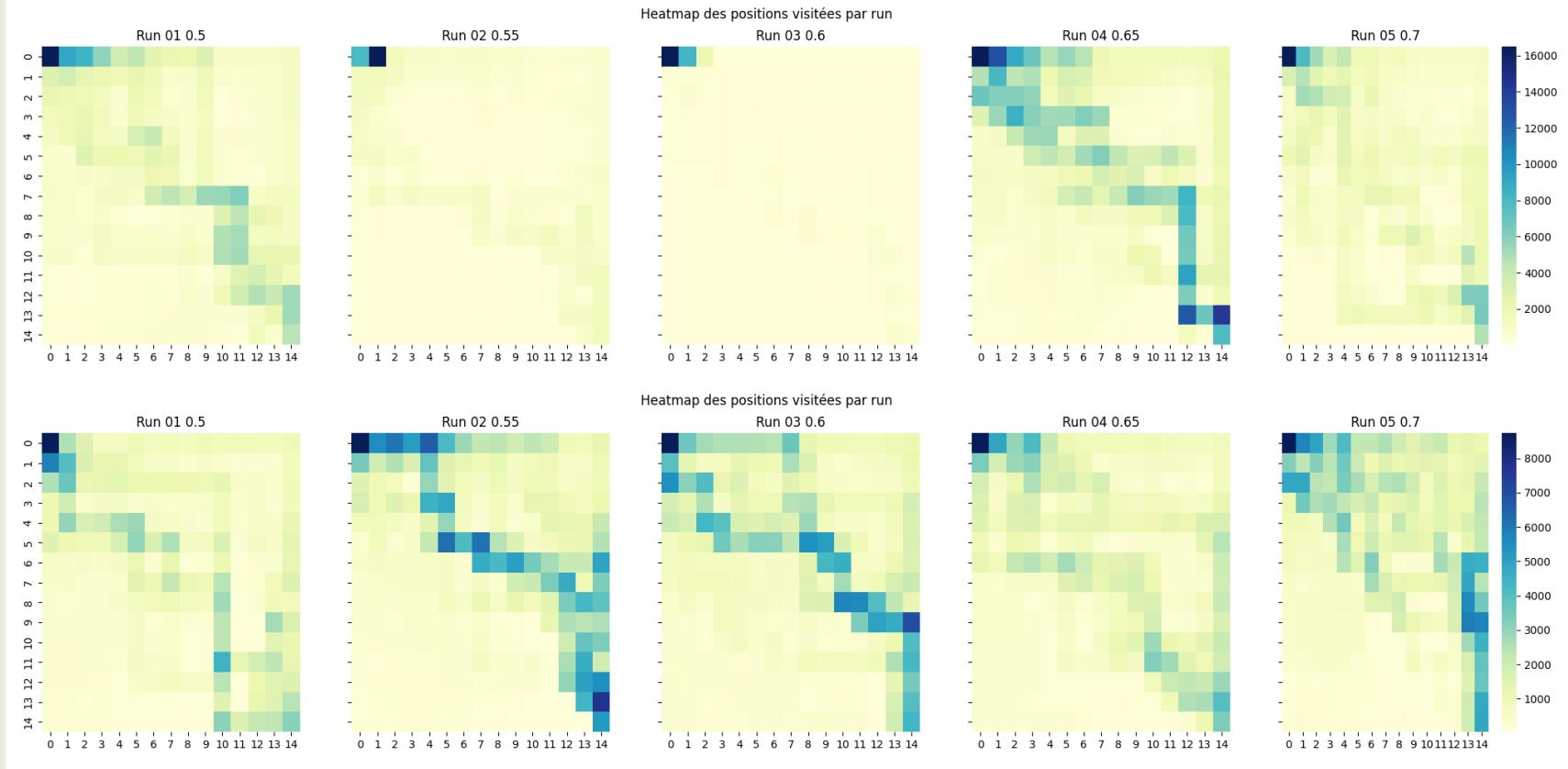


Face à des obstacles aléatoires et complexes parfois, learning rate ↗ donne un taux de succès ↗

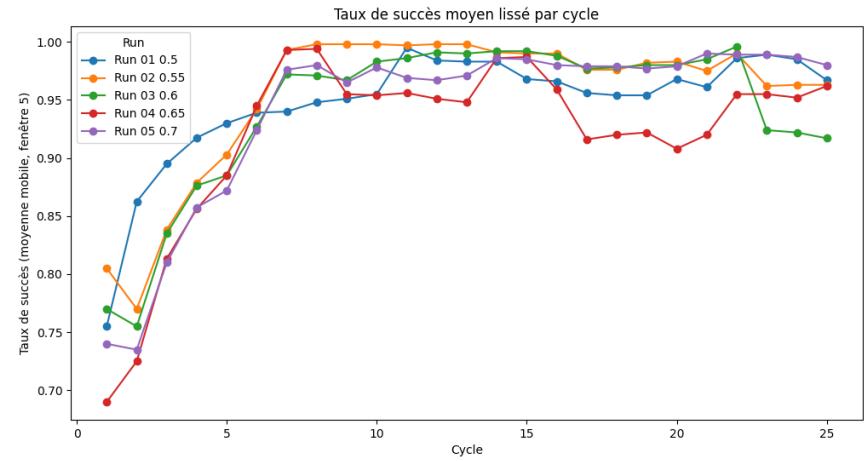
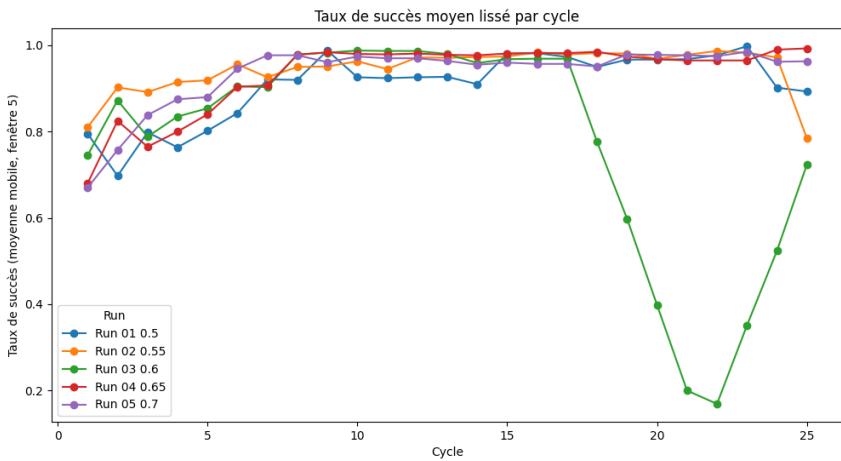
## Comparaison des paramètres – Epsilon



# Comparaison des paramètres – Epsilon

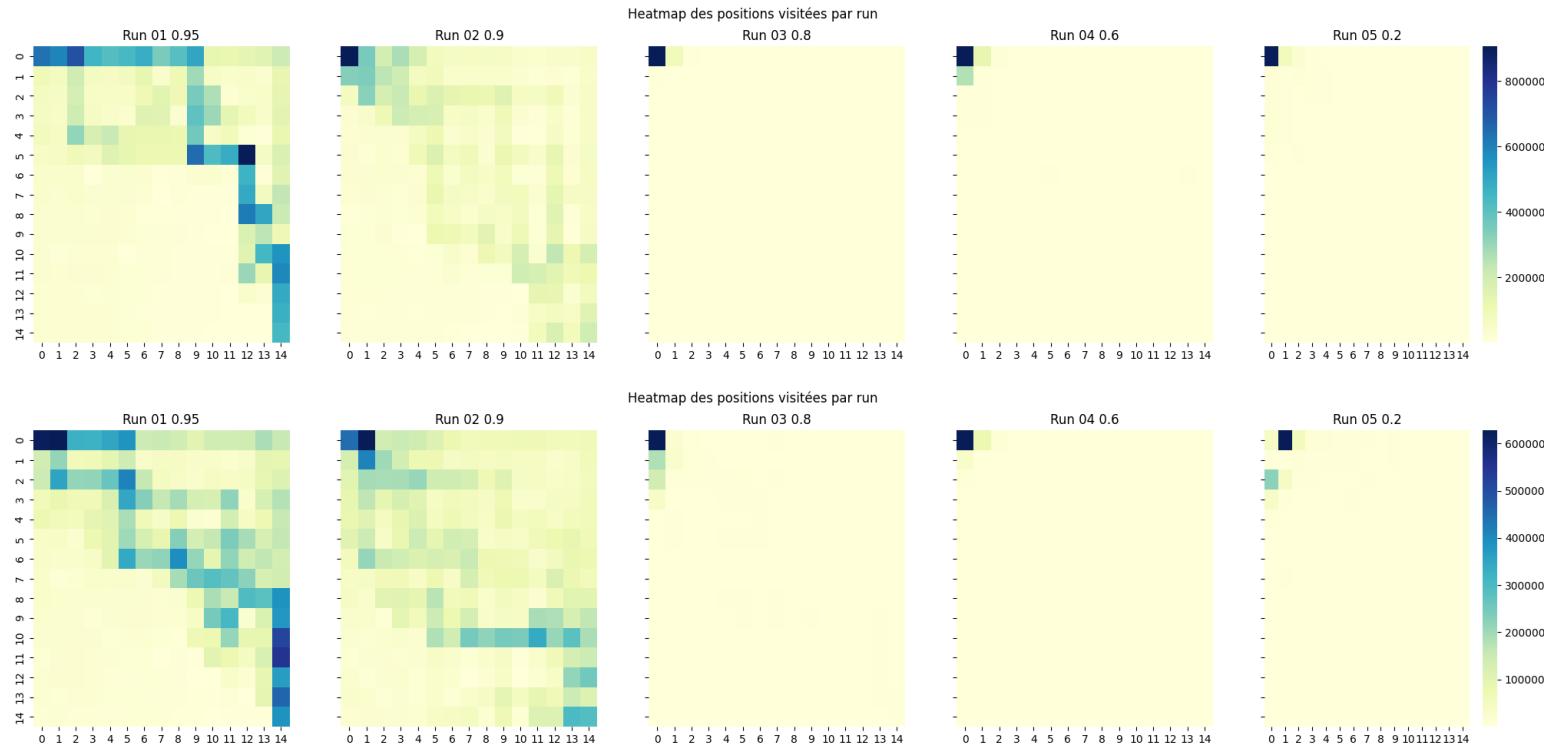


# Comparaison des paramètres – Epsilon

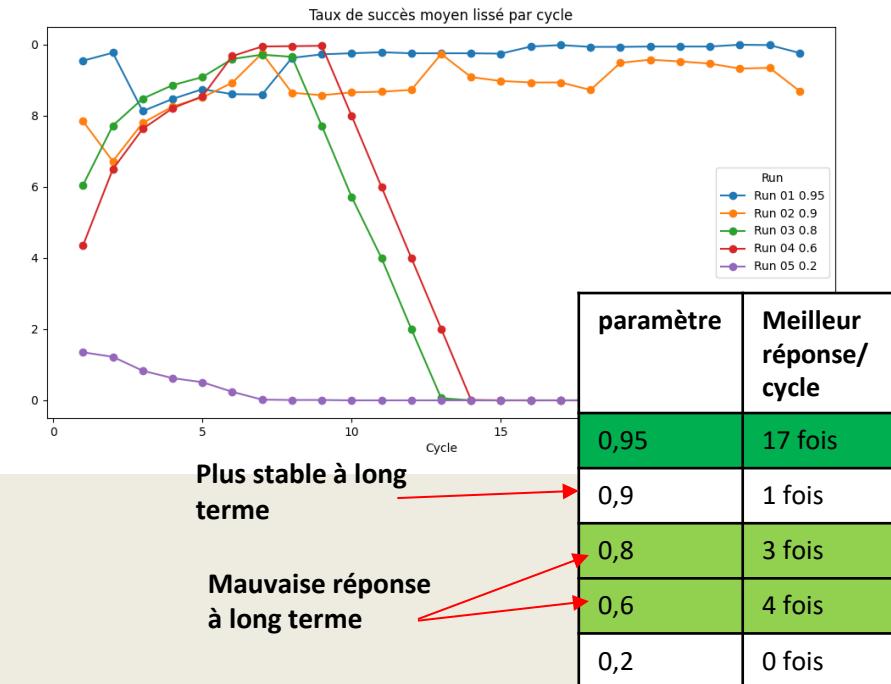
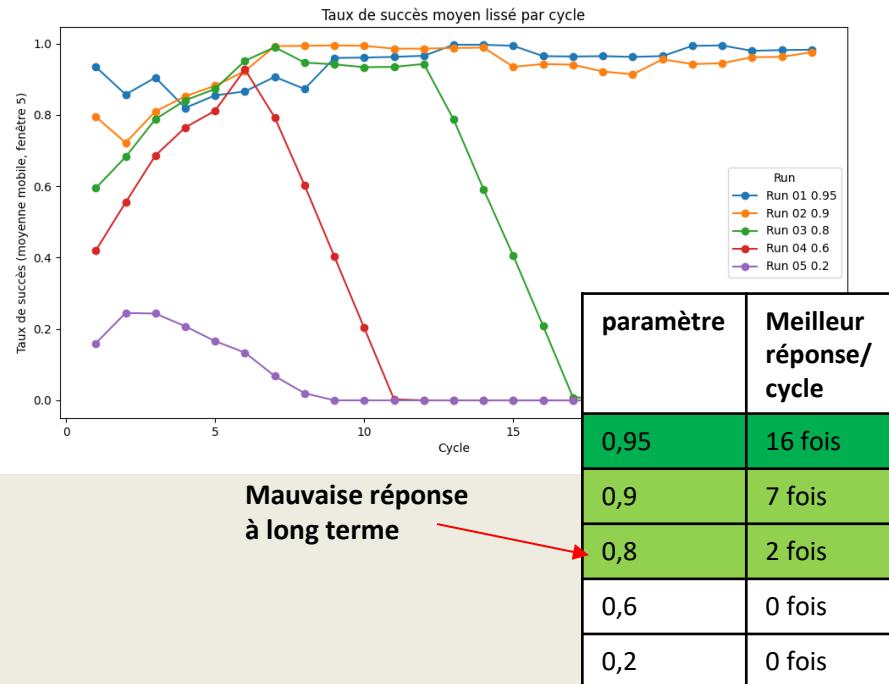


Le choix d'Epsilon autour de 0.5 n'affecte pas beaucoup les performances du système (réponse aléatoire)

# Comparaison des paramètres – Discount factor



# Comparaison des paramètres – Discount factor

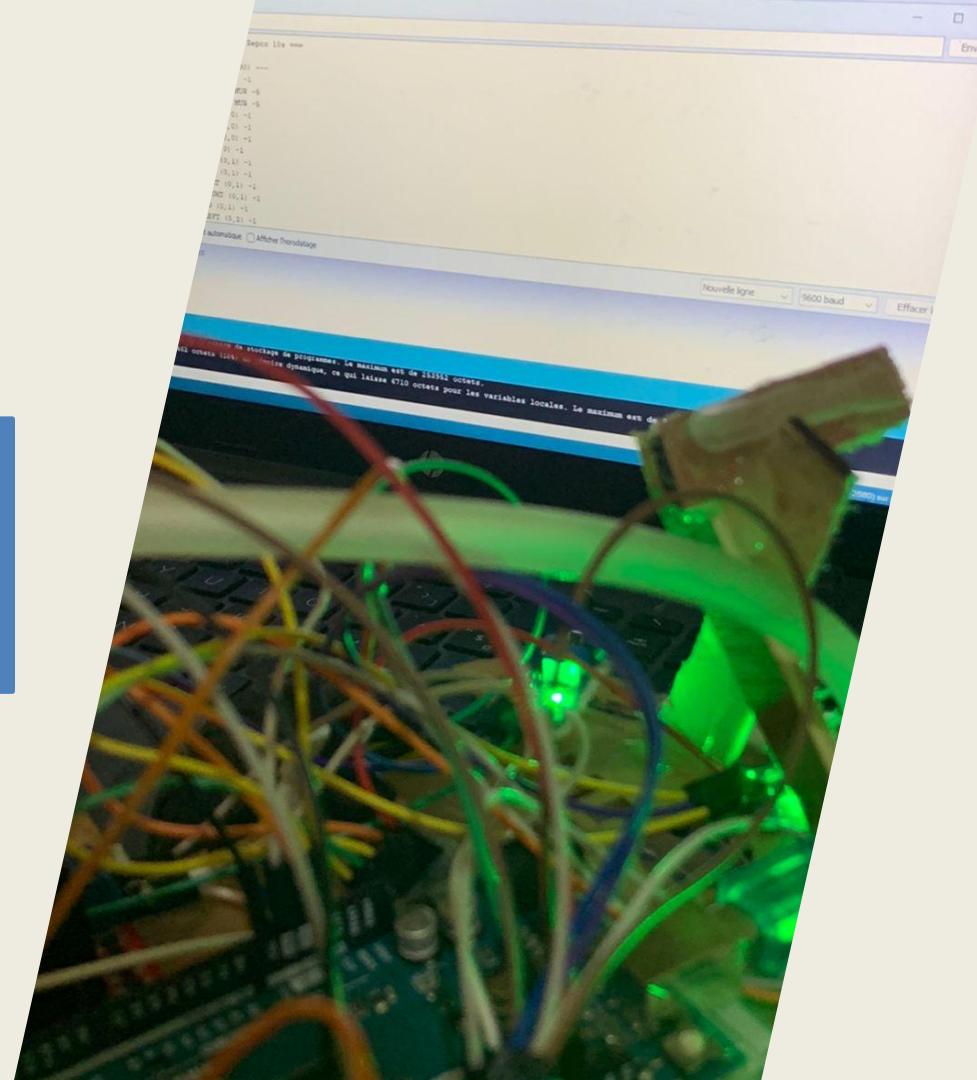


Plus le Discount Factor est grand plus le système est forcé à trouver des solutions plus optimisées et courtes

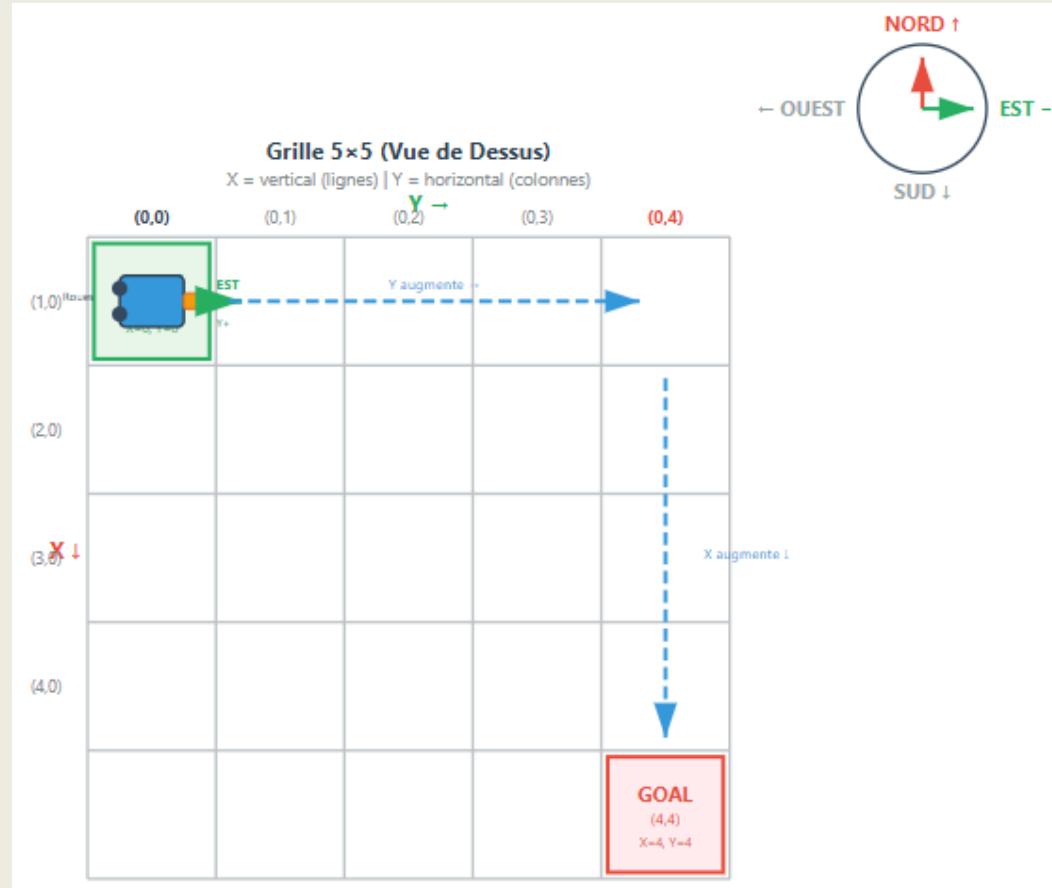
## Comparaison des paramètres – Conclusion

Paramètre	Learning rate	Epsilon	Discount factor
Impact sur le système	Construit des chemins plus fiables et structurés plus est grand	Aléatoire (valeur donnée à la première comparaison/cycle et perdue après)	Recherche de solutions optimale plus est grand
Action à faire	Augmenter	Conserver une valeur proche de 0,5	Augmenter
Valeur générique choisie	0,9	0,55	0,95

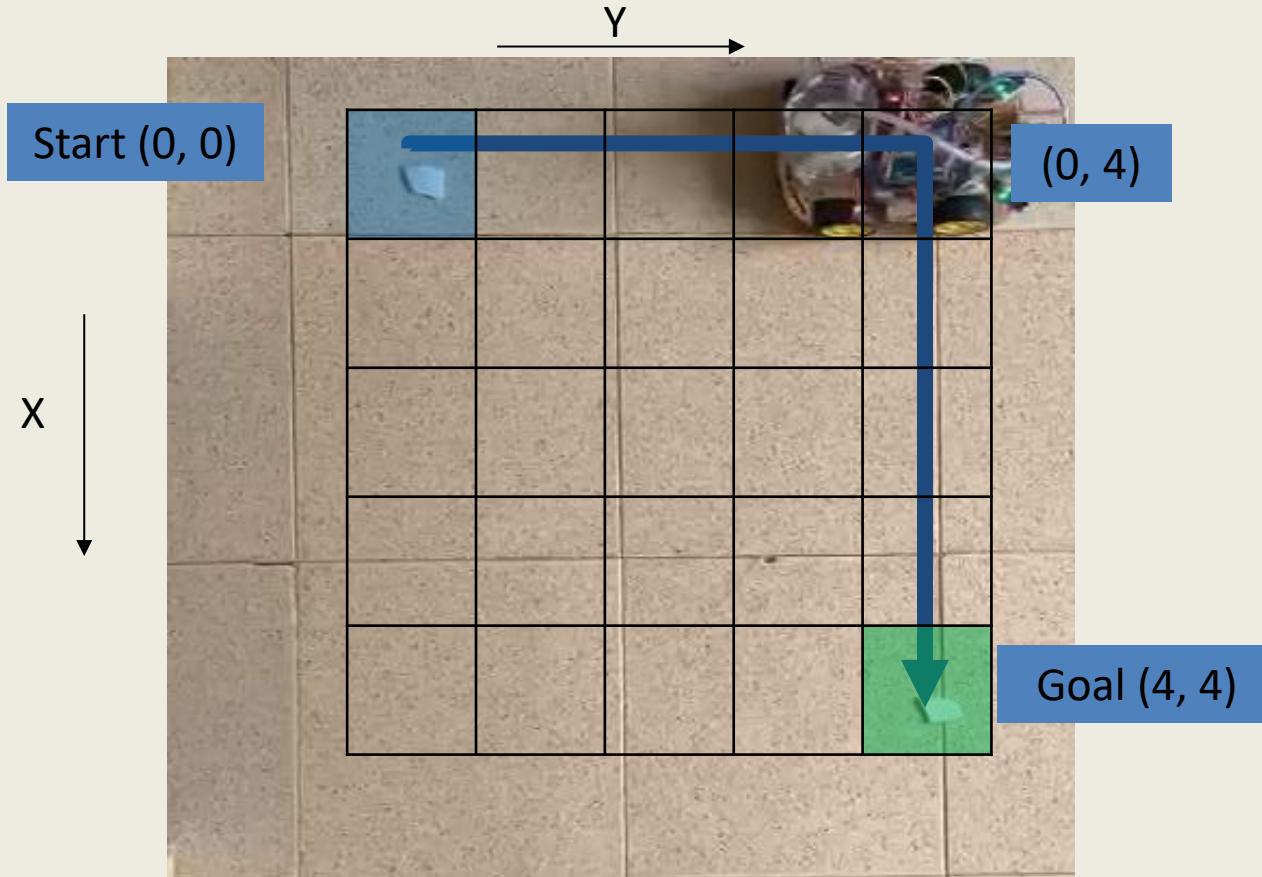
# Simulation réelle avec ARDUINO



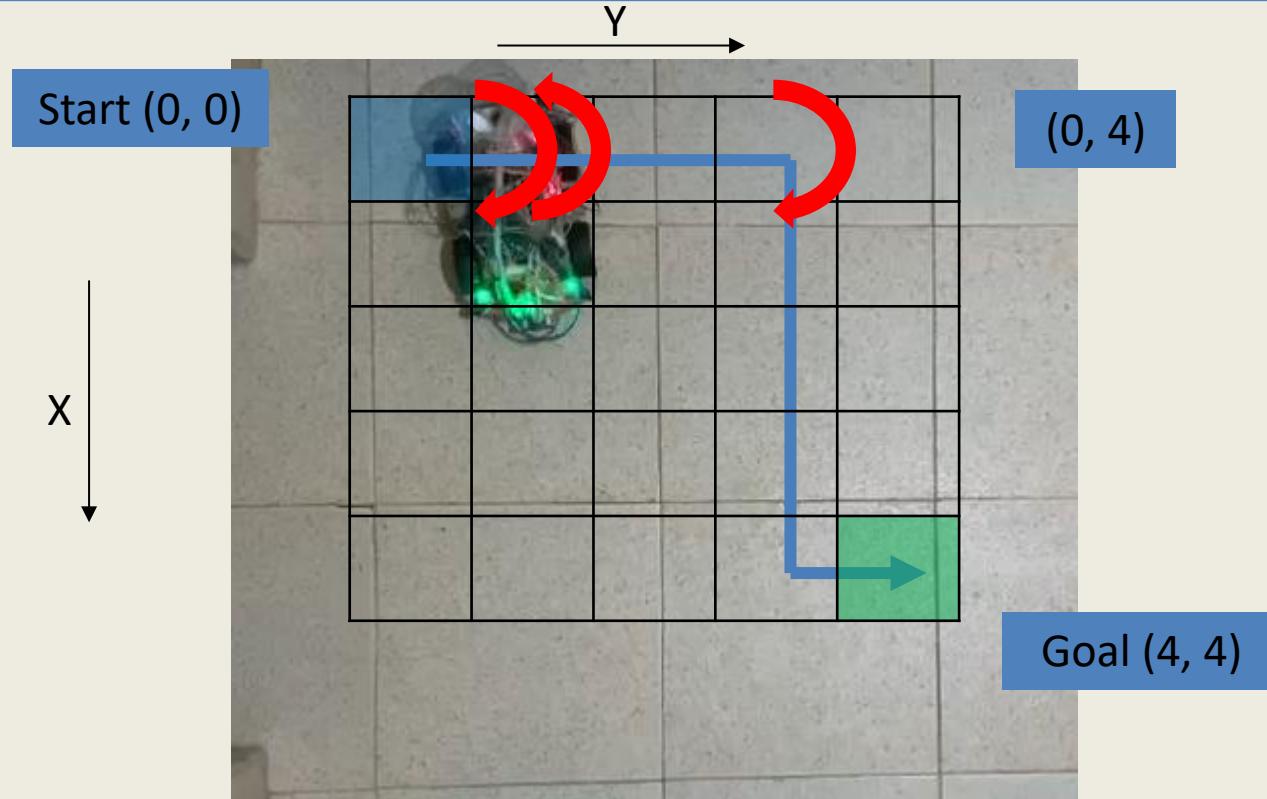
# Configuration initiale



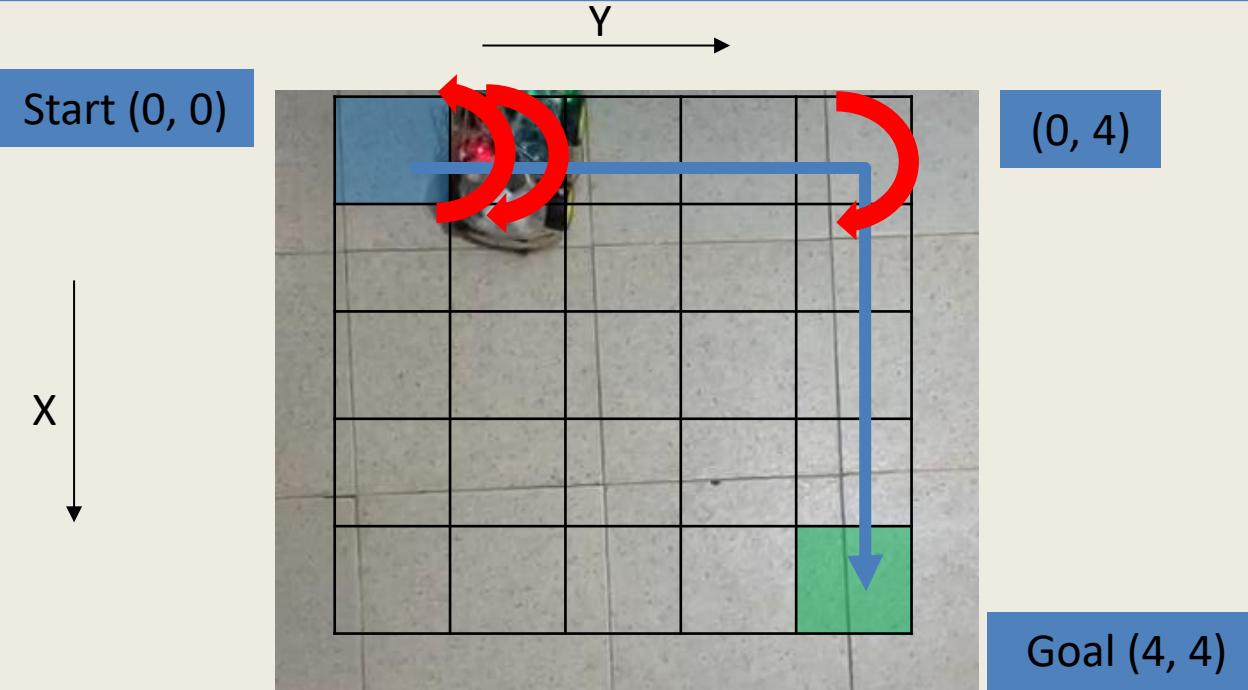
# Chemin préféré



# Test 1



## Test 2



## Test 3 – Max des épisodes

