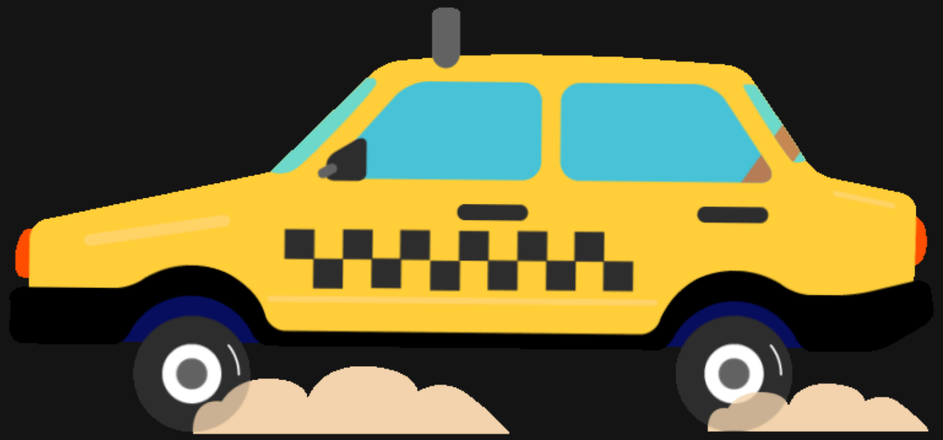
T-AIA-901

Taxi Driver



Bastien Berguer - Jatin Bhatia - Romain Piot

SOMMAIRE

PROJET

Taxi Driver

Problématique à résoudre

Tester différentes approches d'IA pour gagner au jeu Taxi-V3

Organisation

Nos outils pour gérer notre projet

OUTIL DE TRAVAIL

- TEAMS
- GITLAB REPO
- TRELLO

TECHNOS UTILISÉES

- JUPYTER NOTEBOOK
- PYTHON
- DJANGO

SERVEUR & DATA

- HEBERGEMENT LOCAL

LIBRAIRIES

- NUMPY
- PANDAS
- TENSORFLOW
- GYM

Organisation

Gestion d'équipe

FONCTIONNEMENT

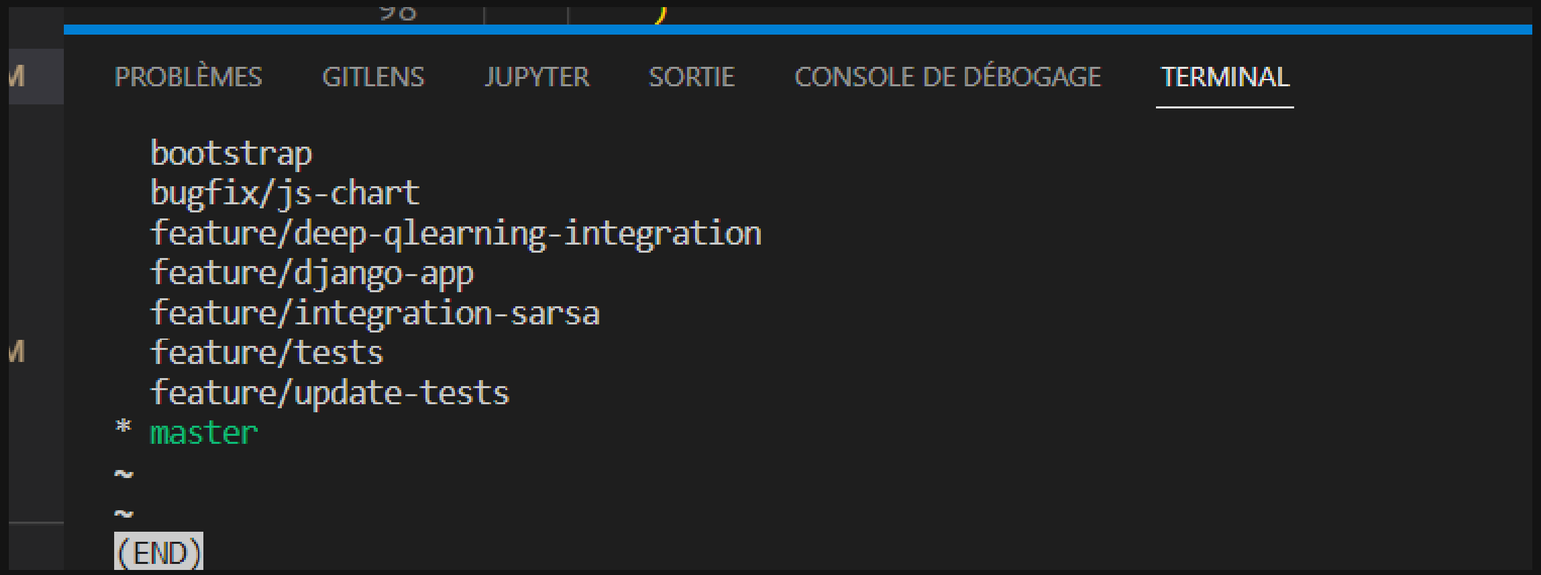
- Découpement et affectation des taches
- Réunion hebdomadaire
- Peer programming
- Versionning et Pull Requests

GIT & Gitlab

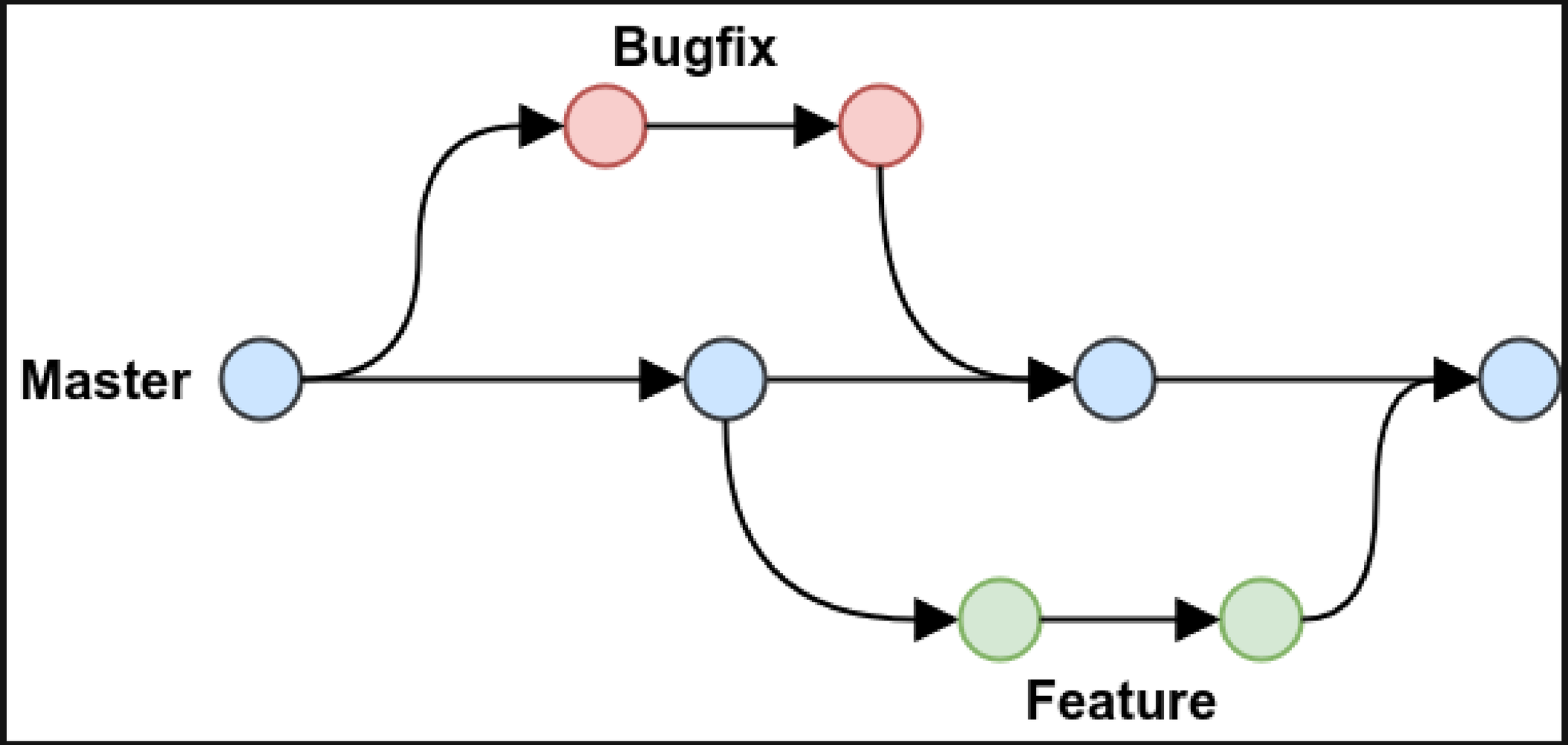
Comment avons-nous procédé

FONCTIONNEMENT

- Respect du github flow par branches (feature, bugfix, etc)



GIT & Gitlab

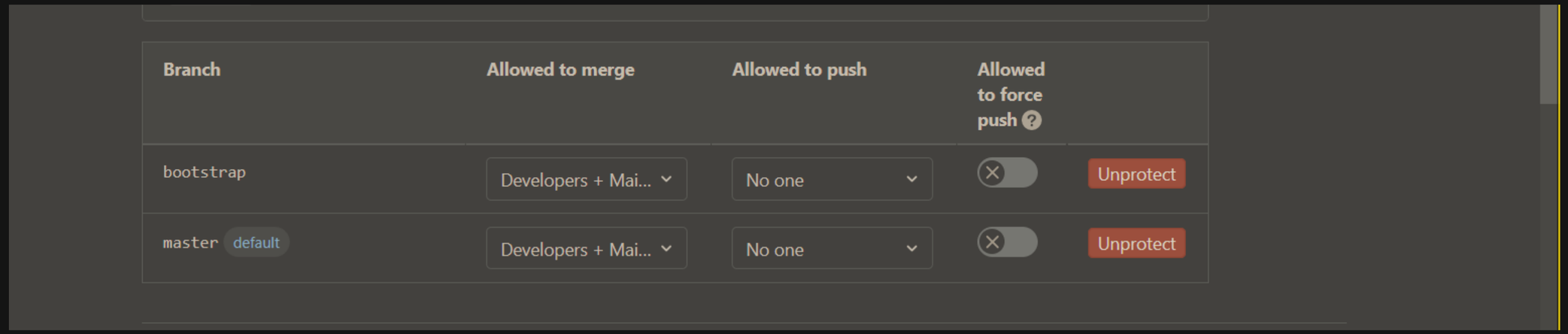


GIT & Gitlab

Comment avons-nous procédé

FONCTIONNEMENT

- Respect du github flow par branches (feature, bugfix, etc)
- Protection des branches principales

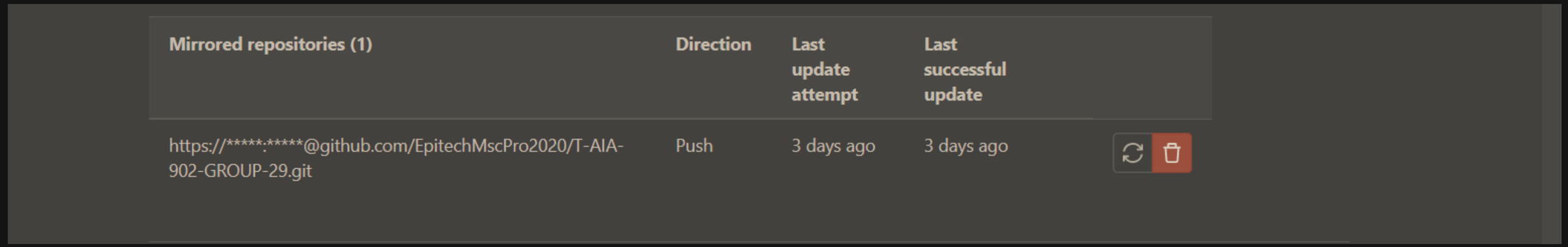


GIT & Gitlab

Comment avons-nous procédé

FONCTIONNEMENT

- Respect du github flow par branches (feature, bugfix, etc)
- Protection des branches principales
- Mirroring gitlab > Github



ARCHITECTURE DU PROJET

Comment avons-nous procédé

STRUCTURE

- Django (MVT, Services, classes)
- Respect des conventions de nommages
- Flake8 & black : detection et correction automatiques d'erreurs

Fichier Edition Sélection Affichage Atteindre Exécuter Terminal Aide q_learning.py - t-aia-902-group-29 [WSL : Ubuntu-20.04] - Visual Studio Code

EXPLORATEUR

views.py | q_learning.py 6, M

T-AIA-902-GROUP-29 [WSL : UBUNTU-20.04]

back_django > src > t_aia_902 > services > q_learning.py > QLearning > play_to_taxi

```
109
110             if      show_print:    multiple spaces after keyword
111                 print(f"Total episodes : {episodes}")
112                 print(f"Average rewards : {avg_rewards}")
113                 print(f"Average steps : {avg_steps}")
114                 print(f"Won episode : {won_episode} ({won_rate}%)")
115     blank line contains whitespace
116                 if training:
117                     print(f"Training time : {duration} seconds"          )    whitespace before ')'
118
119         avg_rewards = avg_rewards if "avg_rewards" in locals() else 0
120         avg_steps = avg_steps if "avg_steps" in locals() else 0
121         duration = duration if "duration" in locals() else  0    multiple spaces after keyword
122         won_rate = won_rate if "won_rate" in locals() else 0
123     blank line contains whitespace
124         return {
125             "q_table": q_table,
126             "avg_rewards": avg_rewards,
127             "avg_steps": avg_steps,
128                 "duration": duration,    continuation line unaligned for hanging indent
129         }
130
```

PROBLÈMES 6 GITLENS JUPYTER SORTIE CONSOLE DE DÉBOGAGE TERMINAL

zsh - back_django

```
> source /home/romain/projects/epitech/t-aia-902-group-29/back_django/.venv/bin/activate
~/projects/epitech/t-aia-902-group-29/back_django master !1                    back_django
>
```

WSL : Ubuntu-20.04 master* 4 2 Git Graph You, il y a 3 semaines L 125, col 32 Espaces : 4 UTF-8 LF MagicPython 3.8.10 ('.venv': venv) Go Live Prettier

15:46 23/06/2022

Approche et reflexion

Comment avons-nous pris en main le projet

APPROCHE

- Recherche types d'algo
- Selection et implementation
- Test de performance
- Comparaison des performances
- Front pour selection et configuration d'algo
- Vue comparative des différents résultats

Algorithmes implémentés

3 algorithmes implémentés

- SARSA
- Q-LEARNING
- DEEP Q-LEARNING

Q-Learning

Explication & réalisation

ALGORITHME

- Model off policy
- Mise à jour de la Q table en prenant la valeur max
- Prise de décision en fonction d'epsilon greedy
- Paramètres modifiables par l'utilisateur
- Sauvegarde des résultats d'entrainements avec différents paramètres

```
nb_episodes_list = [i for i in range(1, 50001) if (i)/10000 % 1 == 0]
learning_rate_list = [0.5, 0.6, 0.7, 0.8, 0.9]
reward_discount_rate_list = [0.9,  0.786, 0.618, 0.5, 0.382, 0.2, 0.1]
decay_rate_list = [0.01, 0.03, 0.05, 0.07, 0.1]

best_avg_rewards = 0
best_nb_episodes = 0
best_learning_rate = 0
best_reward_discount_rate = 0
best_decay_rate = 0

for nb_episodes in nb_episodes_list:
    for learning_rate in learning_rate_list:
        for reward_discount_rate in reward_discount_rate_list:
            for decay_rate in decay_rate_list:
                training_result = play_to_taxi(environment=env,
                                q_table=q_table,
                                epsilon=epsilon_rate,
                                epsilon_min=epsilon_min,
                                epsilon_max=epsilon_max,
                                learning_rate=learning_rate,
                                episodes=nb_episodes,
                                reward_discount_rate=reward_discount_rate,
                                decay_rate=decay_rate,
                                training=True,
                                show_print=False,
                                reset=True
                                )
```

	A	B	C	D	E	F	G	H	I	J	K
1	nb_episodes	epsilon_rate	epsilon_min	epsilon_max	learning_rate	reward_disco	decay_rate	avg_rewards	avg_steps	duration	
2	10000	1.0	0.01	1.0	0.5	0.9	0.01	6.8	14.2	8,1	
3	10000	1.0	0.01	1.0	0.5	0.9	0.03	6.84	14.16	8,9	
4	10000	1.0	0.01	1.0	0.5	0.9	0.05	6.85	14.15	7,6	
5	10000	1.0	0.01	1.0	0.5	0.9	0.07	6.87	14.13	8,7	
6	10000	1.0	0.01	1.0	0.5	0.9	0.1	6.83	14.17	9	
7	10000	1.0	0.01	1.0	0.5	0.786	0.01	6.84	14.16	9	
8	10000	1.0	0.01	1.0	0.5	0.786	0.03	6.83	14.17	8,9	
9	10000	1.0	0.01	1.0	0.5	0.786	0.05	6.79	14.21	7,4	
10	10000	1.0	0.01	1.0	0.5	0.786	0.07	6.82	14.18	7,3	
11	10000	1.0	0.01	1.0	0.5	0.786	0.1	6.85	14.15	7,3	
12	10000	1.0	0.01	1.0	0.5	0.618	0.01	6.81	14.19	7,8	
13	10000	1.0	0.01	1.0	0.5	0.618	0.03	6.87	14.13	8	
14	10000	1.0	0.01	1.0	0.5	0.618	0.05	6.81	14.19	7,4	
15	10000	1.0	0.01	1.0	0.5	0.618	0.07	6.86	14.14	7,6	
16	10000	1.0	0.01	1.0	0.5	0.618	0.1	6.84	14.16	7,6	
17	10000	1.0	0.01	1.0	0.5	0.5	0.01	6.86	14.14	8,4	
18	10000	1.0	0.01	1.0	0.5	0.5	0.03	6.85	14.15	8,4	
19	10000	1.0	0.01	1.0	0.5	0.5	0.05	6.8	14.2	7,4	
20	10000	1.0	0.01	1.0	0.5	0.5	0.07	6.82	14.18	8,4	
21	10000	1.0	0.01	1.0	0.5	0.5	0.1	6.85	14.15	8,8	
22	10000	1.0	0.01	1.0	0.5	0.382	0.01	6.81	14.19	7,6	
23	10000	1.0	0.01	1.0	0.5	0.382	0.03	6.81	14.19	8,3	
24	10000	1.0	0.01	1.0	0.5	0.382	0.05	6.84	14.16	7	
25	10000	1.0	0.01	1.0	0.5	0.382	0.07	6.82	14.18	8,8	
26	10000	1.0	0.01	1.0	0.5	0.382	0.1	6.84	14.16	7,8	
27	10000	1.0	0.01	1.0	0.5	0.2	0.01	6.81	14.19	7	
28	10000	1.0	0.01	1.0	0.5	0.2	0.03	6.82	14.18	7,1	
29	10000	1.0	0.01	1.0	0.5	0.2	0.05	6.82	14.18	9	
30	10000	1.0	0.01	1.0	0.5	0.2	0.07	6.82	14.18	7	

SARSA

Explication & réalisation

ALGORITHME

- Modèle on Policy
- Mise à jour de la Q table en fonction d'upsilon Greedy
- Prise de décision en respectant la policy & Epsilon Greedy
- Paramètres modifiables par l'utilisateur
- Sauvegarde des résultats d'entrainements avec différents paramètres

	column 1	column 2	column 3	column 4	column 5	column 6	column 7	column 8	column 9	column 10
1	nb_episodes	epsilon_rate	epsilon_min	epsilon_	learning_rate	reward_discou	decay_rate	avg_rewards	avg_steps	duration
2	10000	1.0	0.01	1.0	0.5	0.9	0.01	5.76	22.4	7.016526222229004
3	10000	1.0	0.01	1.0	0.5	0.9	0.03	5.47	16.15	12.667381048202515
4	10000	1.0	0.01	1.0	0.5	0.9	0.05	6.54	16.64	18.305476665496826
5	10000	1.0	0.01	1.0	0.5	0.9	0.07	6.17	22.3	24.37621283531189
6	10000	1.0	0.01	1.0	0.5	0.9	0.1	6.47	15.9	29.806821584701538
7	10000	1.0	0.01	1.0	0.5	0.786	0.01	5.4	46.91	45.63572859764099
8	10000	1.0	0.01	1.0	0.5	0.786	0.03	4.93	59.41	59.37526273727417
9	10000	1.0	0.01	1.0	0.5	0.786	0.05	6.53	42.95	70.98104333877563
10	10000	1.0	0.01	1.0	0.5	0.786	0.07	7.95	57.37	83.6065731048584
11	10000	1.0	0.01	1.0	0.5	0.786	0.1	6.42	42.01	93.87324500083923
12	10000	1.0	0.01	1.0	0.5	0.618	0.01	7.34	69.7	114.36566472053528
13	10000	1.0	0.01	1.0	0.5	0.618	0.03	6.68	70.7	133.4906759262085
14	10000	1.0	0.01	1.0	0.5	0.618	0.05	6.72	63.4	151.27170062065125
15	10000	1.0	0.01	1.0	0.5	0.618	0.07	6.38	68.21	169.46469640731812
16	10000	1.0	0.01	1.0	0.5	0.618	0.1	7.81	69.42	188.0488133430481
17	10000	1.0	0.01	1.0	0.5	0.5	0.01	6.45	82.83	209.52252626419067
18	10000	1.0	0.01	1.0	0.5	0.5	0.03	7.68	89.92	232.066570520401
19	10000	1.0	0.01	1.0	0.5	0.5	0.05	9.2	89.57	254.52487683296204
20	10000	1.0	0.01	1.0	0.5	0.5	0.07	5.19	80.48	275.1728301048279
21	10000	1.0	0.01	1.0	0.5	0.5	0.1	6.71	75.67	296.8964207172394
22	10000	1.0	0.01	1.0	0.5	0.382	0.01	4.74	83.11	319.75680208206177
23	10000	1.0	0.01	1.0	0.5	0.382	0.03	4.16	69.3	341.0043978691101
24	10000	1.0	0.01	1.0	0.5	0.382	0.05	6.27	75.98	361.5018730163574
25	10000	1.0	0.01	1.0	0.5	0.382	0.07	7.69	71.93	381.98092794418335
26	10000	1.0	0.01	1.0	0.5	0.382	0.1	6.11	82.09	403.9304850101471

Deep Q-Learning

Modèle

ALGORITHME

- Création du modèle

```
def build_model(actions):
    model = Sequential()
    model.add(Embedding(500, 10, input_length=1))
    model.add(Reshape((10,)))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(actions, activation='linear'))
    return model
```

Deep Q-Learning

Environnement et configuaration

ALGORITHME

- Import de l'environnement Taxi V3
- Police : epslylon greed
- Création de l'agent (dqn)
- Metrique : mae

Deep Q-Learning

Entrainement et utilisation

ALGORITHME

- Entrainement en utilisant le gpu
- Savegarde des poids (100k -> 1000k)
- Chargement des poids pour utiliser les models

TESTS UNITAIRE DE SERVICES

TESTS DES ALGO

- Tests sur les 3 classes de services
- plusieurs methodes par tests
- generation d'un rapport de coverage

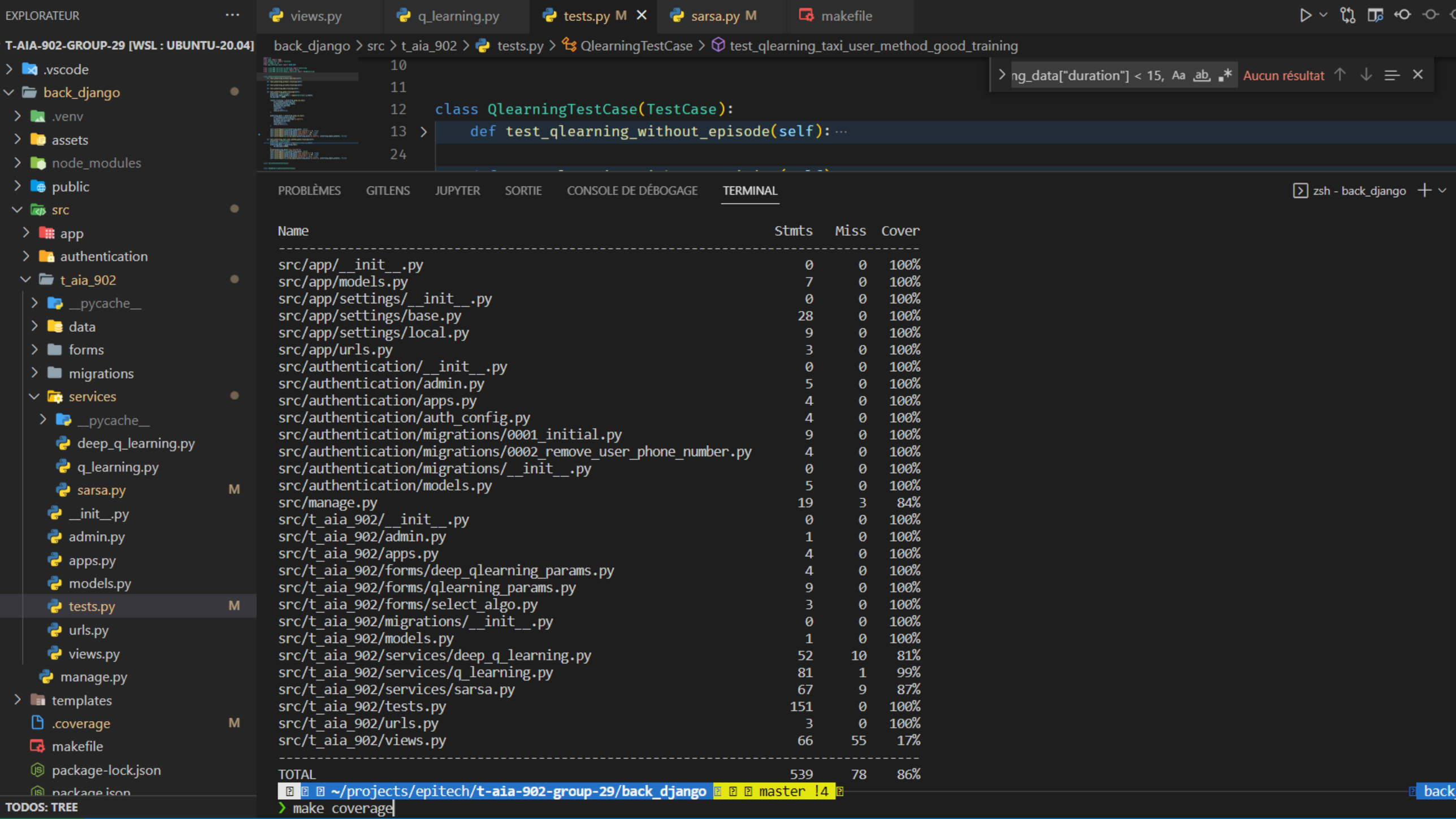
T-AIA-902-GROUP-29 [WSL : UBUNTU-20.04]
.vscode
back_django
.venv
assets
node_modules
public
src
app
authentication
t_aia_902
__pycache__
data
forms
migrations
services
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
manage.py
templates
.coverage M
makefile
package-lock.json
package.json
requirements.txt
webpack.config.js
yarn.lock
Documents
TODOS: TREE

```python
class QlearningTestCase(TestCase):
    def test_qlearning_without_episode(self):...

    def test_qlearning_without_training(self):...

    def test_qlearning_private_training(self):...

    def test_qlearning_bad_training(self):...

    def test_qlearning_good_training(self):...

    def test_qlearning_taxi_user_method_good_training(self):
        qlearning = QLearning()
        qlearning_empty_qtable = copy(QLearning().q_table)
        qlearning_data = qlearning.taxi(
            nb_episodes=10000,
        )
        print(qlearning_data["avg_rewards"])
        self.assertEqual(type(qlearning_data), dict)
        self.assertEqual(qlearning_data["avg_rewards"] > 0, True)
        self.assertEqual(qlearning_data["avg_steps"] != 99, True)
        self.assertEqual(qlearning_data["duration"] > 0.2, True)
        self.assertEqual(np.array_equal(qlearning_data["q_table"], qlearning_empty_qtable), False)
```

PROBLÈMES GITLENS JUPYTER SORTIE CONSOLE DE DÉBOGAGE TERMINAL

zsh - back_django

```
Average steps : 14.21
Won episode : 10000 (100.0%)
6.79
........
----------------------------------------------------------------------
Ran 15 tests in 139.838s

OK
Destroying test database for alias 'default'...
```


EXPLORATEUR
T-AIA-902-GROUP-29 [WSL : UBUNTU-20.04]
.vscode
back_django
.venv
assets
node_modules
public
src
app
authentication
t_aia_902
__pycache__
data
forms
migrations
services
__pycache__
deep_q_learning.py
q_learning.py
sarsa.py M
__init__.py
admin.py
apps.py
models.py
tests.py M
urls.py
views.py
manage.py
templates
.coverage M
makefile
package-lock.json
package.json
TODOS: TREE
views.py
q_learning.py
tests.py M
sarsa.py M
makefile
back_django > src > t_aia_902 > tests.py > QlearningTestCase > test_qlearning_taxi_user_method_good_training
10
11
12 class QlearningTestCase(TestCase):
13 def test_qlearning_without_episode(self):
24
ng_data["duration"] < 15, Aa ab .* Aucun résultat
PROBLÈMES GITLENS JUPYTER SORTIE CONSOLE DE DÉBOGAGE TERMINAL
zsh - back_django
Name Stmts Miss Cover
src/app/__init__.py 0 0 100%
src/app/models.py 7 0 100%
src/app/settings/__init__.py 0 0 100%
src/app/settings/base.py 28 0 100%
src/app/settings/local.py 9 0 100%
src/app/urls.py 3 0 100%
src/authentication/__init__.py 0 0 100%
src/authentication/admin.py 5 0 100%
src/authentication/apps.py 4 0 100%
src/authentication/auth_config.py 4 0 100%
src/authentication/migrations/0001_initial.py 9 0 100%
src/authentication/migrations/0002_remove_user_phone_number.py 4 0 100%
src/authentication/migrations/__init__.py 0 0 100%
src/authentication/models.py 5 0 100%
src/manage.py 19 3 84%
src/t_aia_902/__init__.py 0 0 100%
src/t_aia_902/admin.py 1 0 100%
src/t_aia_902/apps.py 4 0 100%
src/t_aia_902/forms/deep_qlearning_params.py 4 0 100%
src/t_aia_902/forms/qlearning_params.py 9 0 100%
src/t_aia_902/forms/select_algo.py 3 0 100%
src/t_aia_902/migrations/__init__.py 0 0 100%
src/t_aia_902/models.py 1 0 100%
src/t_aia_902/services/deep_q_learning.py 52 10 81%
src/t_aia_902/services/q_learning.py 81 1 99%
src/t_aia_902/services/sarsa.py 67 9 87%
src/t_aia_902/tests.py 151 0 100%
src/t_aia_902/urls.py 3 0 100%
src/t_aia_902/views.py 66 55 17%
TOTAL 539 78 86%
~/projects/epitech/t-aia-902-group-29/back_django master !4
back
make coverage

Web App

World Wide Web

DJANGO

- Réception des informations de formulaire
- Lancement de l'algorithme selectionné avec les paramètres utilisateur
- Envoi du résultat
- Comparatif des résultats

T-AIA-902 Accueil

Algo*

Q-Learning

Q-Learning
Deep Q-Learning
Sarsa

Nb episodes*

3000

default = 10 000

Epsilon rate*

1,0

default = 1.0

Epsilon min*

0,01

default = 0.01

Epsilon max*

1,0

default = 1.0

Learning rate*

0,5

default = 0.8

Reward discount rate*

0,786

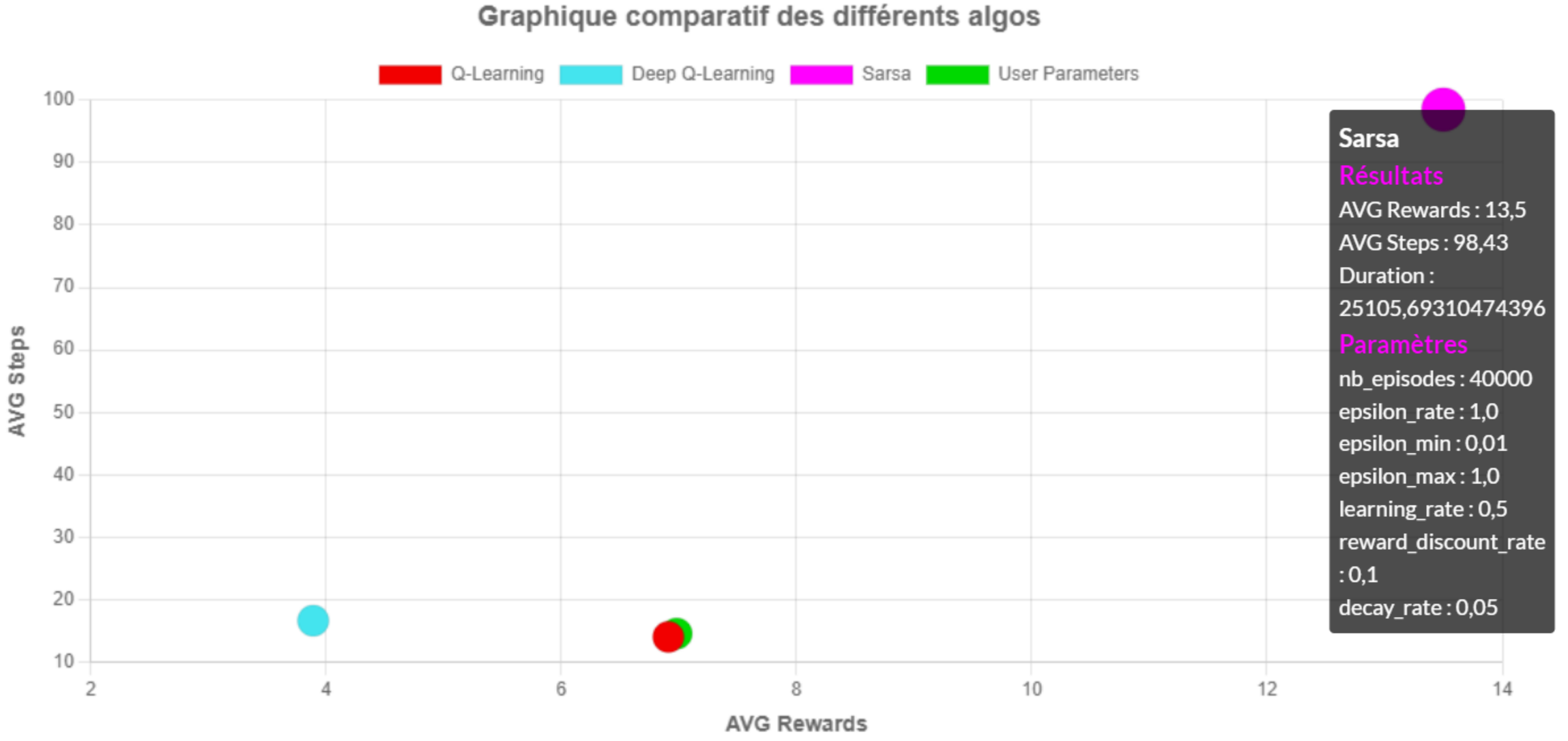
default = 0.786

Decay rate*

0,1

default = 0.07

Valider


Graphique comparatif des différents algos
Q-Learning
Deep Q-Learning
Sarsa
User Parameters
AVG Steps
AVG Rewards
100
90
80
70
60
50
40
30
20
10
2
4
6
8
10
12
14
Sarsa
Résultats
AVG Rewards : 13,5
AVG Steps : 98,43
Duration : 25105,69310474396
Paramètres
nb_episodes : 40000
epsilon_rate : 1,0
epsilon_min : 0,01
epsilon_max : 1,0
learning_rate : 0,5
reward_discount_rate : 0,1
decay_rate : 0,05

Conclusion

Pour conclure !

- Découverte du RL/deep RL
- Différents algorithmes plus ou moins efficaces
- Projet complet
- Jeu simple

DÉMO

https://travel-resolver-100.netlify.app/


THANK YOU
FOR WATCHING & LISTNING
memegenerator.net