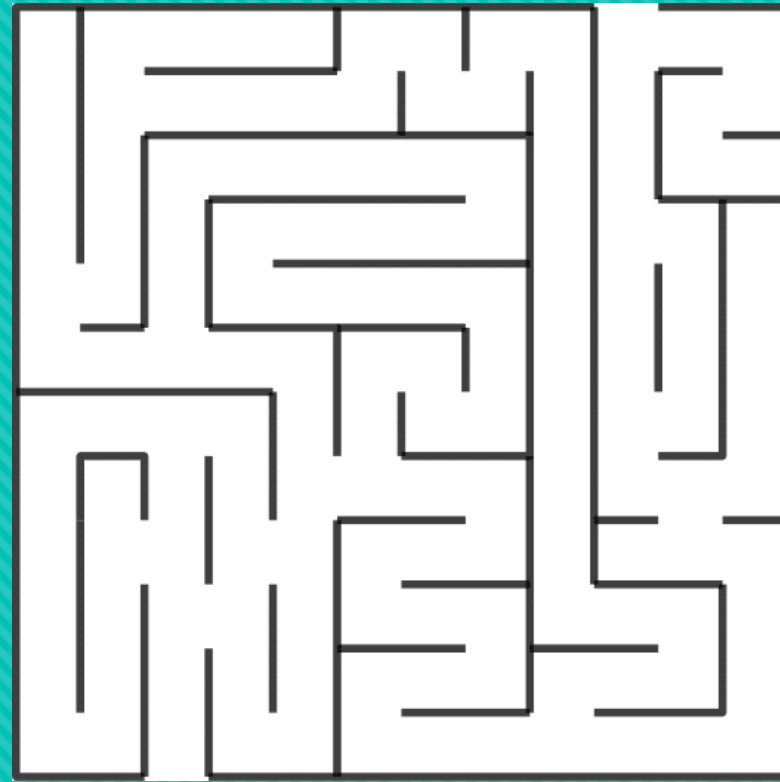


Apprentissage par renforcement



Le labyrinthe

- I. Contexte et Problématique
- II. Solution : architecture & design
- III. Vie du projet
- IV. Performances
- V. Conclusion

Sommaire

I. Contexte et Problématique

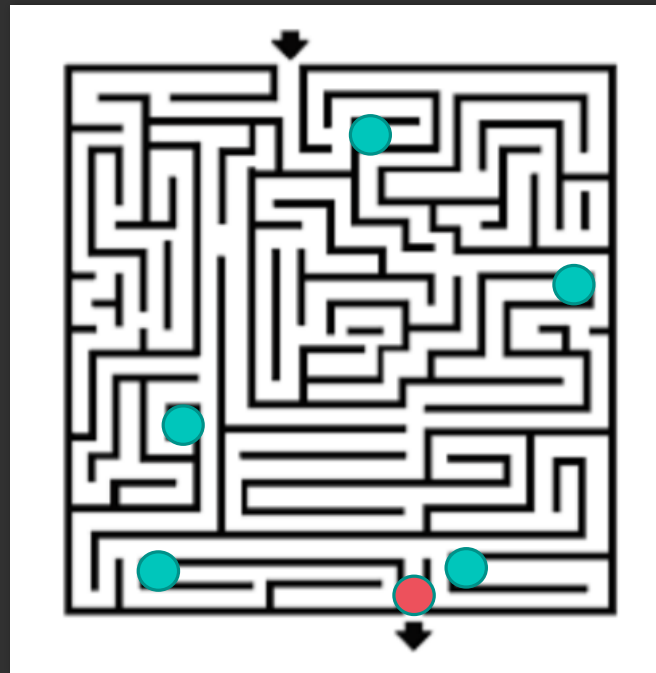
Trouver le chemin optimal pour résoudre le labyrinthe

Exemple

Récompenses

● 5 points

● 500 points



II. Solution : architecture et design

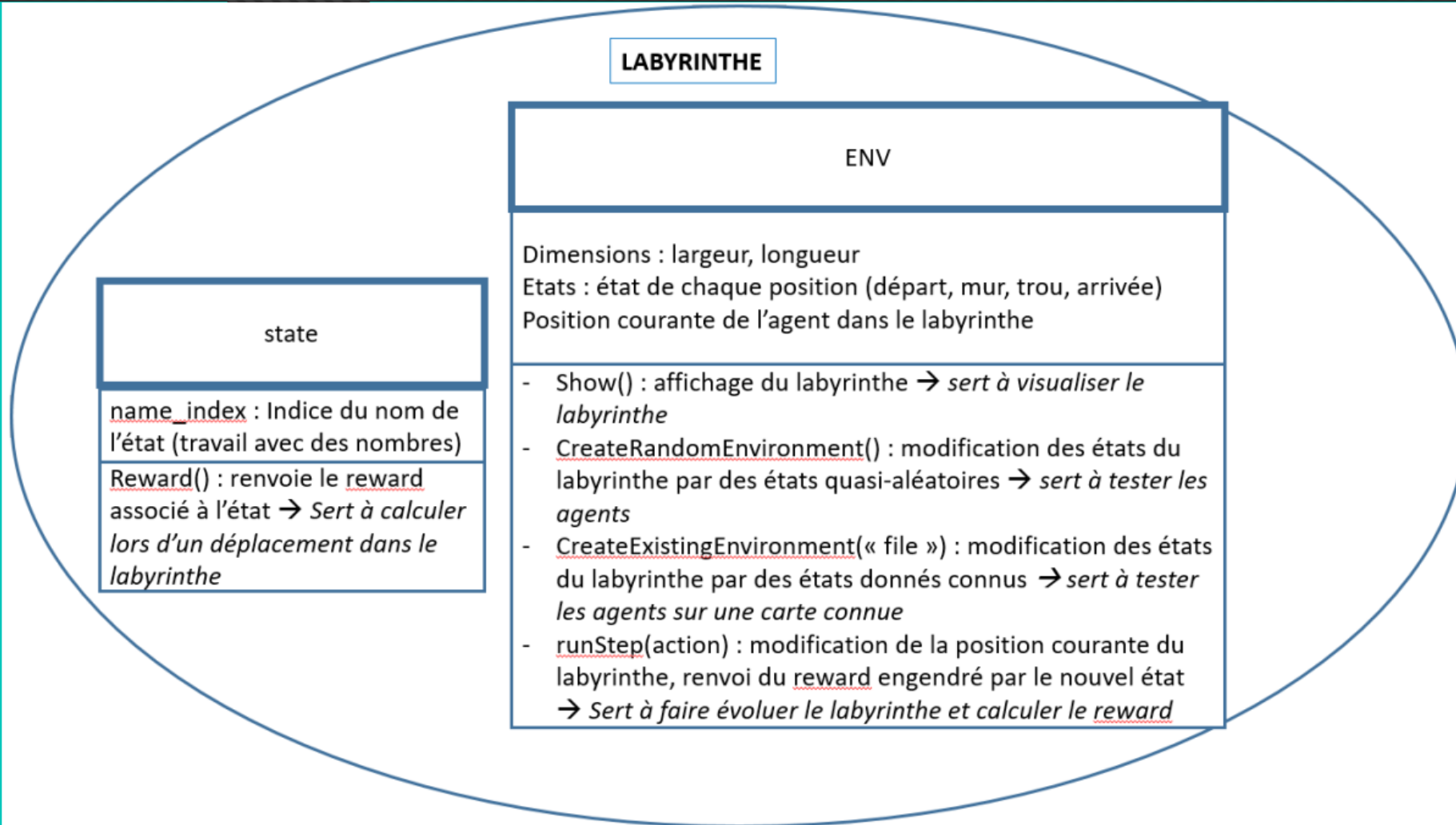
Labyrinthe

System

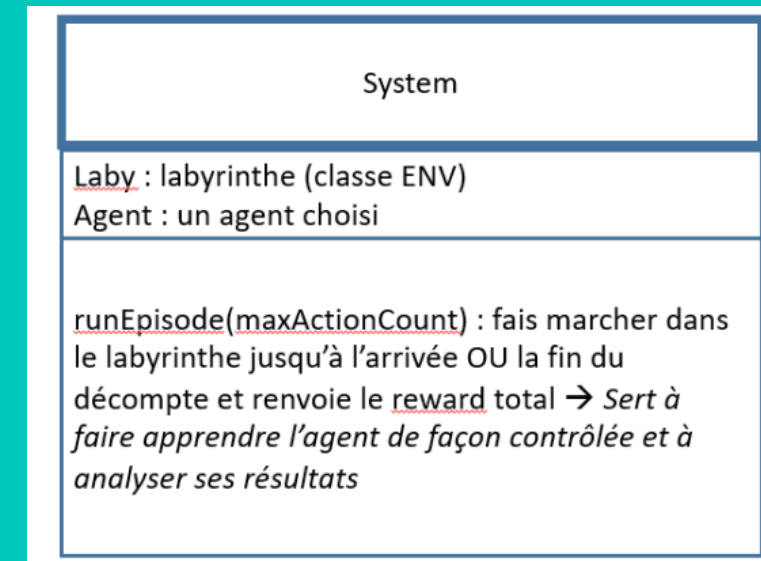
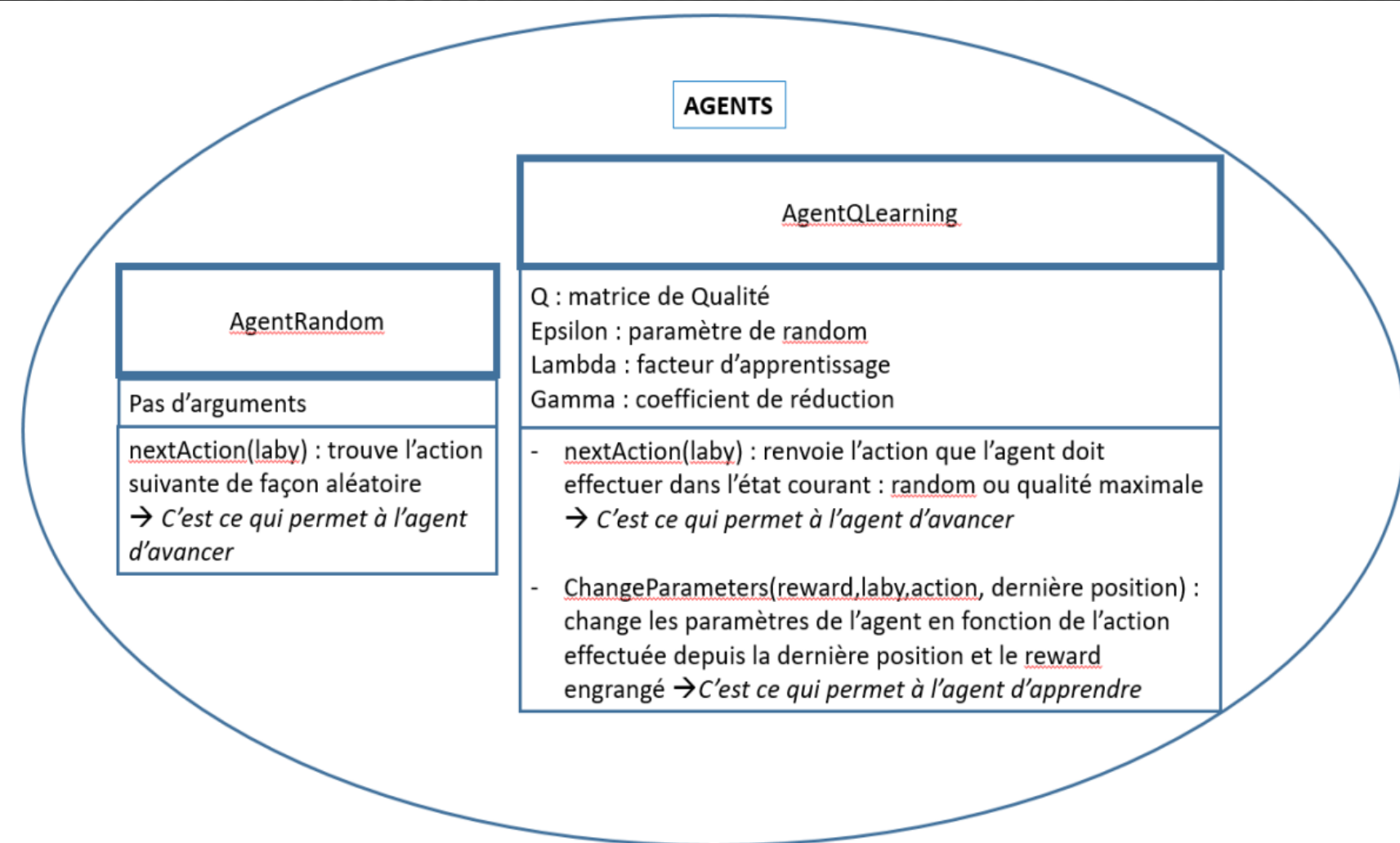
Agent



II. Solution : architecture et design



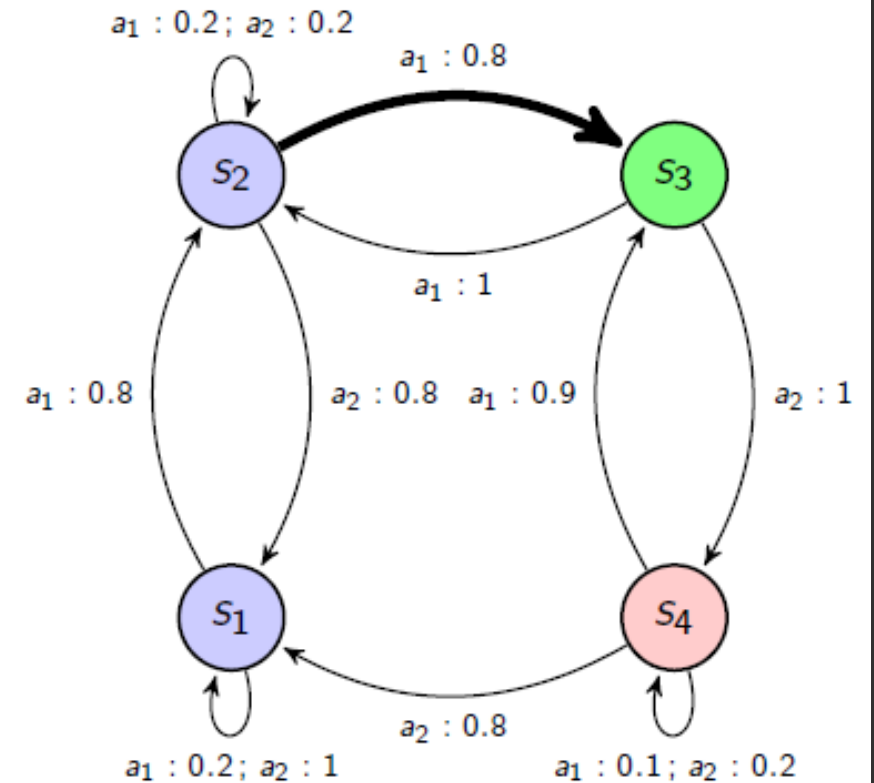
II. Solution : architecture et design



Parenthèse – Le Q-Learning

- à chaque arrivée sur un état, on note l'action ayant mené à cet état en renforçant la note existante (l'utilité/Qualité) avec une partie de l'utilité du noeud.
- Arrivé sur l'état s' à partir de l'état s et de l'action a , on note la Qualité de l'arc (s, a, s') :

$$Q(s, a) = \lambda \times (r + \gamma \times \max_{a'} (Q(s', a'))) + (1 - \lambda) \times Q(s, a)$$



III. Vie du projet

1^{ère} partie du projet

- Classe Environnement 2D
 - Modélisation d'un labyrinthe aléatoire et de ses différents états / Visualisation graphique
 - Méthode « possibleActions » essentielle pour l'agent et évolution du labyrinthe
- Tests / Méthode agile
 - Tests unitaires de l'environnement et tests globaux de modélisation du labyrinthe
 - Classe Agent aléatoire

III. Vie du projet

2^{ème} partie du projet

- Classe Agent Q-Learning
 - Matrice de Qualité et son évolution
- Tests
 - Tests unitaires de l'agent et tests globaux appliqués au labyrinthe
 - Méthode pour créer un labyrinthe lu dans un fichier

III. Vie du projet

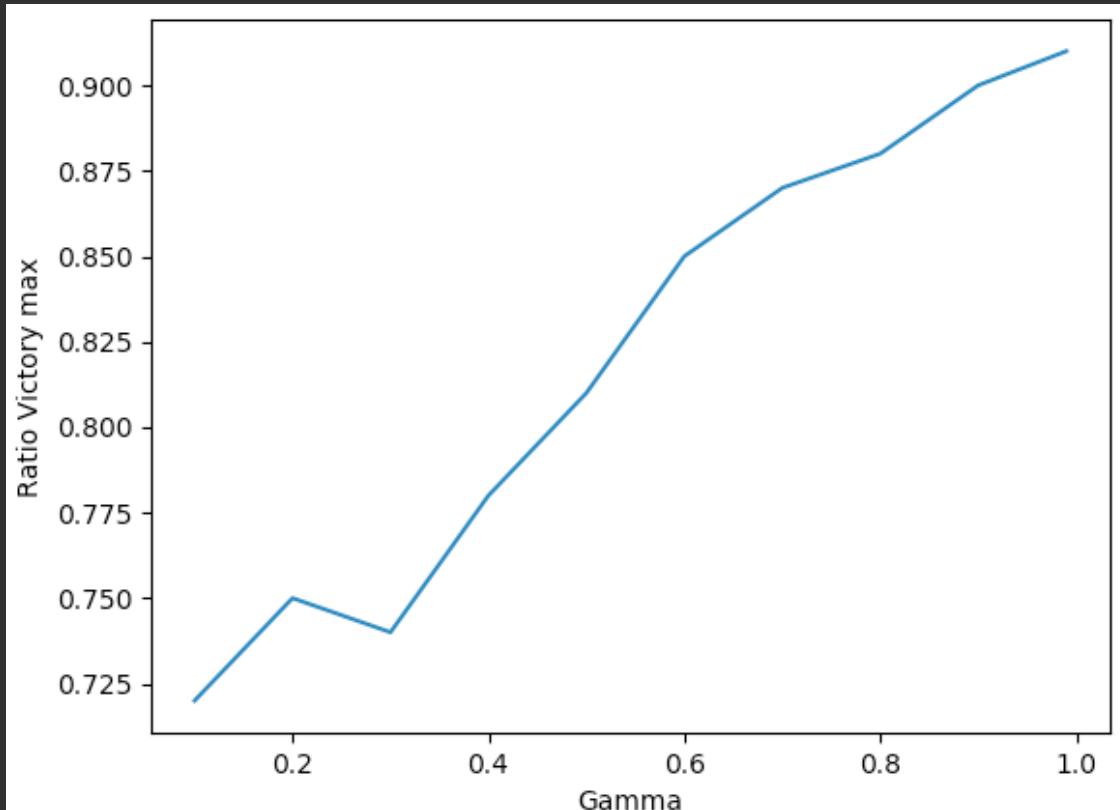


Expérience Git

- Utilisation d'Atom et du package « Git Plus »
 - Interface graphique pour interagir avec Git
 - « Git add + commit + push » simplifié
 - Conflits facilement résolus et fusion instantanée
- Avantages
 - Travail en parallèle sur le même projet (potentiellement même fichier)
 - Fonctionnalités très pratiques et productives pour un tel projet (+1000 lignes de code)

IV. Performance

○ Influence du facteur d'apprentissage Gamma



Conditions des tests

Labyrinthe de taille 10x10

Nombre d'épisodes : 1000

Nombre d'actions par épisode : 1000

Ratio Victory : Nombre de victoires sur Nombre d'épisodes

Gamma variant de 0,1 à 0,99 (pas de 0,1 ou 0,09)

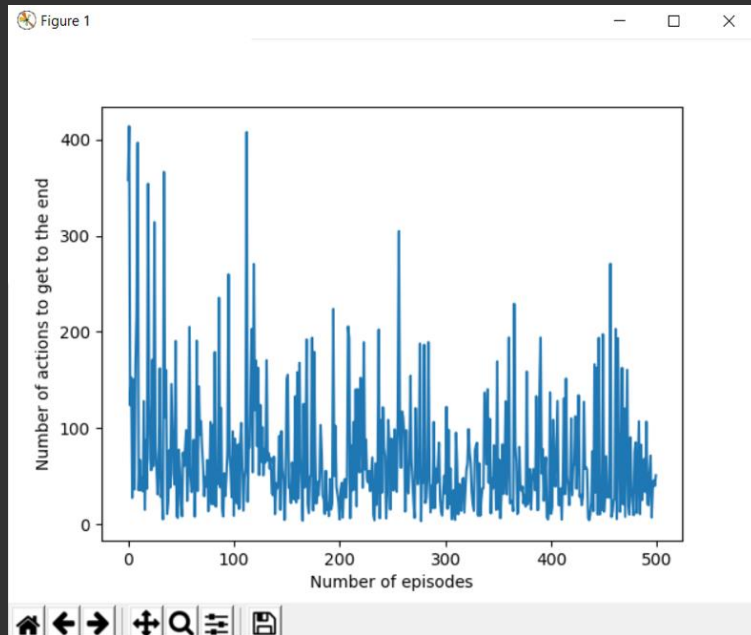
Pour un même labyrinthe:

Moyenne sur 10 valeurs de Ratio pour chaque Gamma

IV. Performance

- Nombre de déplacements pour sortir du labyrinthe en fonction du nombre d'épisodes

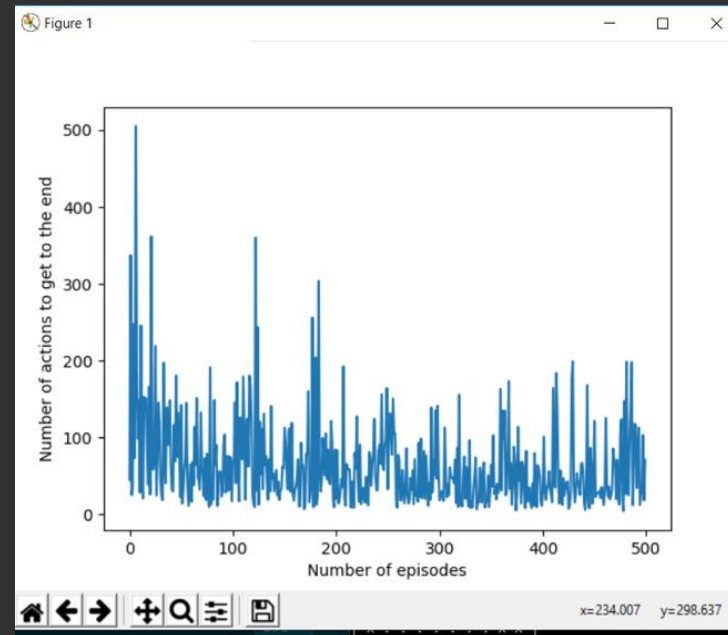
Labyrinthe 10x10 500 épisodes 500 actions



Gamma : 0.1

Moyenne d'actions nécessaires :

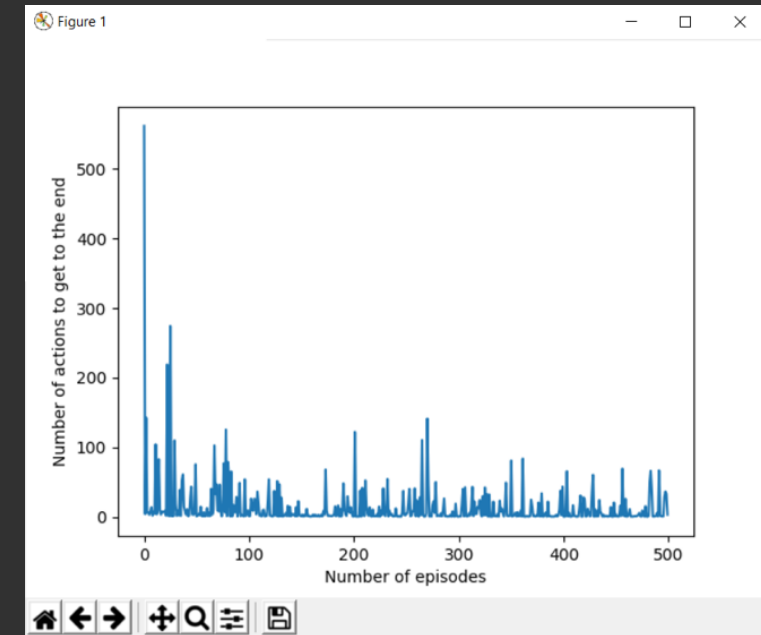
80



Gamma : 0.4

Moyenne d'actions nécessaires :

50



Gamma : 0.8

Moyenne d'actions nécessaires :

10

IV. Performance

- Défauts et améliorations possibles
 - Manque de puissance de calculs : Complexité spatiale et temporelle en $O(n^3)$ en le nombre de cases du labyrinthe
 - Structure et architecture de l'environnement et de l'agent créés par nous-mêmes
 - Non nécessairement optimal
 - Défauts d'un labyrinthe aléatoire
 - Chemin de l'entrée vers la sortie potentiellement bloquée par des murs
 - Entrée juste à côté de la sortie
 - Mise à jour de la matrice de Qualité uniquement dans un rayon de deux cases de la position courante

V. Conclusion

- Modélisation et implémentation d'un projet conséquent
 - Recherche d'informations et état de l'art à effectuer avant de pouvoir commencer
- Expérimentation du logiciel Atom et de ses fonctionnalités Git
 - Outils optimisés permettant de gagner du temps et de l'efficacité
- Expérience du travail de groupe pour des projets futurs
 - Organisation du projet et division des tâches