

Mini-Projet de simulateur de robot mobile autonome

Objectifs pédagogiques :	2
Modalités du rendu :	2
Critères d'évaluation :	2
Contexte	3
Descriptif du Mini-projet L1 SPL.....	4
Tâches à effectuer	5
Etape 1 (Séance 4) :	5
Etape 2 (Séance 5) et Etape 3 (Séance 6) :	6
Etape 4 (Séance 7) :	9
Etape 5 (Séance 8) :	9

Objectifs p dagogiques :

- Interpr ter l'algorithme d'un projet Python   partir du code source
- D finir un algorithme avanc    partir d'un probl me r el
- Utiliser un environnement de d veloppement Python
- Structurer les fichiers sources et leurs contenus afin qu'ils soient r utilisables
- V rifier par des tests pertinents, identifier et corriger les erreurs d'un algorithme

Modalit s du rendu :

- Individuellement
- A la fin de chaque s ance du projet
- D poser vos algorithmes ainsi que vos tests.
 - Algorithme
 - Fichier(s) source(s) .py
 - Tests
 - Capture(s) d' cran
 - L'ensemble du projet peut  tre d pos 
 - S par ment ou regroup s au format compress  (.rar .zip .tar .7zip) ou dans un m me document (.doc .odt .pdf)
 - Sur l'espace de d p t eCampus de votre groupe

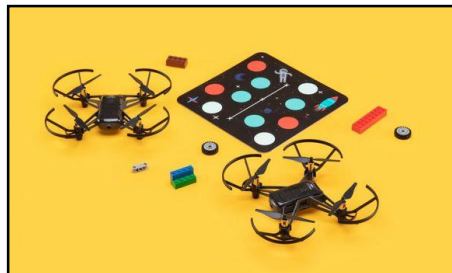
Crit res d' valuation :

- Assiduit  (Pr sence & Rendu)
- Lisibilit  :
 - Commentaires dans vos programmes
 - Pertinence des noms de variables
 - D coupage   l'aide de fonctions
- Tests r alis s
- Nombre d'objectifs fonctionnels remplis et en coh rence avec le sujet
- Niveau de complexit 
- Rapport de projet (4 pages maximum) :
 - Description du sujet et des objectifs sp cifiques
 - M thode utilis e, justification des choix de conception (avantages et inconv nients)
 - R sultats obtenus et perspectives pour une potentielle suite du projet

Contexte

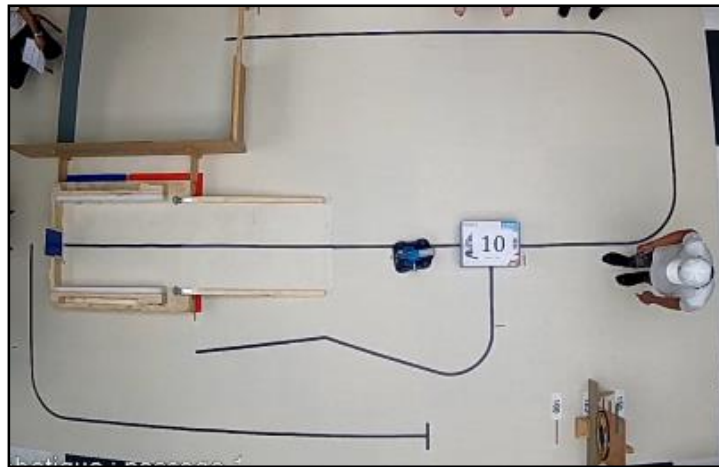
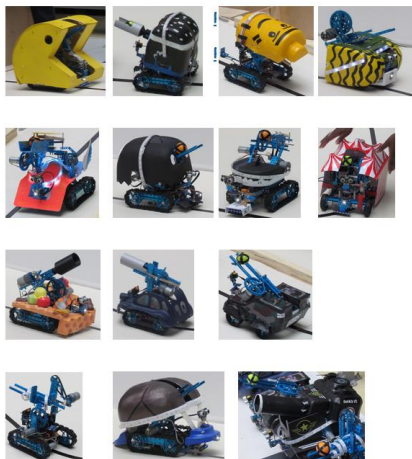
Du primaire au Master en passant par la Licence

Il existe de nombreuses méthodes pédagogiques pour découvrir la programmation. Le déplacement des robots sur un quadrillage équipé de cibles et d'obstacle est une approche appliquée dès le plus jeune âge. Apprendre la rédaction et les tests des algorithmes est également possible à l'aide d'un mini-drone tel que le Tello Ryze EDU.

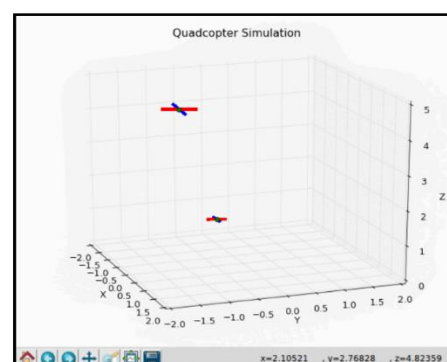
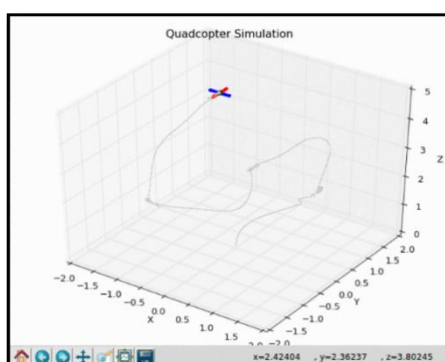


A titre indicatif, à l'UFR ST de Evry, les étudiants L3 SPI relèvent tout au long de l'année le défi de concevoir une solution pluridisciplinaire devant être validée par un rapport, une soutenance ainsi qu'une compétition :

<https://intranet.ufrst.univ-evry.fr/index.php/component/content/category/71-uncategorised>



Un projet de master en informatique pourrait être similaire à celui créé au MIT par Abhijit Majumdar et disponible sur GitHub : https://github.com/abhijitmajumdar/Quadcopter_simulator



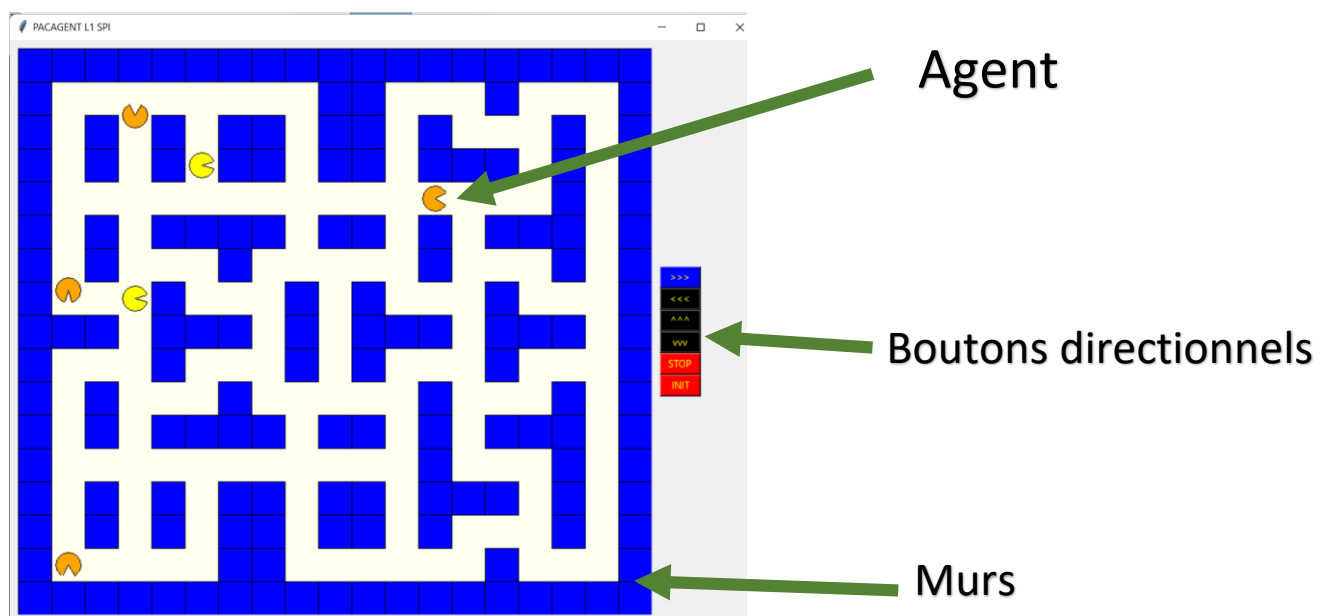
Descriptif du Mini-projet L1 SPI

Votre mini-projet en Algorithmique sera dédié à un simulateur de robot mobile autonome (agent) ou en Python.

Pour cela, nous vous proposons de concevoir en se basant sur un jeu 2D Pacman, disponible dans la rubrique « **Ressources Mini-Projet** » de votre espace eCampus.

Ce jeu est basé sur un algorithme simple ne gérant actuellement aucun score, aucune victoire et aucune défaite. Les déplacements des agents oranges dits « autonomes » sont effectués de façon automatique (aléatoire) alors que les déplacements des agents jaunes dits « non autonomes » sont effectués à l'aide des flèches du clavier (ou en cliquant sur les boutons directionnels). Les agents changent de couleur (rouge) lorsqu'ils sont en collision.

Il est à noter que celui-ci requiert l'installation du module tkinter. Ce dernier est déjà installé pour Spyder et Idle dans les salles TP.



Tâches à effectuer

- Choix du sujet
 - Dans chaque groupe TP, chaque étudiant devra choisir un sujet différent (dès la séance 4)
 - Nous essayerons de respecter au mieux, les préférences de chaque étudiant. L'ordre des demandes sera priorisé : les premiers demandés seront les premiers servis. Cependant, si deux étudiants demandent en même temps le même sujet, un tirage au sort pourra mener à les départager.

Etape 1 (Séance 4) :

- Effectuer un tutoriel tkinter
 - Lire <https://python.doctor/page-tkinter-interface-graphique-python-tutoriel>
 - Créer une première application tkinter
 - a) Fenêtre faisant apparaître une zone de dessin et un bouton
 - b) En cliquant sur le bouton, un cercle apparaîtra dans la zone de dessin
- Suivre la présentation du fichier source du projet
- Changement de contexte
 - Compléter son entête (nom, prénom, sujet)
 - Changer les couleurs des agents et des murs
 - Personnaliser le terrain : Modifier les positions des murs en laissant toute zone accessible
 - Différencier la vitesse des agents autonomes de celle des agents non autonomes
- Indépendance des agents non autonomes
 - La direction du premier agent non autonome est choisie à l'aide des flèches et le second agent non autonome à l'aide des touches ZQSD
 - Affecter une couleur différente à chaque agent autonome (une même nuance)
- Créer une nouvelle liste gérant l'état de fonctionnement de chaque agent
 - Valeur comprise entre 0 et 100
 - Lors d'une collision, l'état de fonctionnement de chacun diminue
 - Lorsqu'un agent possède un état de fonctionnement nul, il ne doit plus pouvoir se déplacer
 - Si tous les agents non-autonomes sont immobilisés, la partie se termine par une défaite.
- Des pièces doivent être placées sur le sol aléatoirement

- Apparence : Petites cercles remplies
- Disposition : Des cases de valeur 2 doivent  tre ajout es au tableau « niveau1 »
- Disparition : Lorsqu'un agent non autonome entre en collision
- Objectif : Au-del  d'une fin en victoire ou d faite, le nombre de pi ces ramass es sera li  au score   afficher (affichage r alis  dans l' tape 4).

Bonus 1.1 : Chaque agent fait demi-tour lors d'une collision s'il le peut

Bonus 1.2 : Utilisation des images   la place de l'animation pacman

Bonus 1.3 : Regrouper les donn es des tableaux vitX vitY agents et sontAutonomes dans un seul et m me tableau   l'aide d'une seule classe

Etape 2 (S ance 5) et Etape 3 (S ance 6) :

Il est   noter que certains sujets peuvent s'av rer plus complexes que d'autres. Ainsi l' valuation ne porte pas exclusivement sur le nombre d'objectifs atteints mais sur la qualit  et la quantit  du rapport et des fichiers sources rendus.

- Choix et impl mentation de l'un des sujets suivants :
 - Sujet 1 : Murs mobiles
 - a) Au cours de la partie, certains murs devront se d placer (modification du parcours en cours de partie)
 - b) Il doit  tre  viter toute situation ne laissant aucune possibilit    un agent de sortir de zone
 - Sujet 2 : T l portation
 - a) Des zones fonctionnant par paire, permettront   tout agent de changer directement de position.
 - b) Ces zones appara tront temporairement et changeront elle-m me d'emplacement r guli rement.
 - Sujet 3 : Checkpoints
 - En  vitant qu' tre trop regrouper voire tous   proximit , les zones de checkpoints appara ssent en d but de partie sous forme pseudo-al atoirement
 - a) Ils doivent  tre franchis dans un ordre pr cis. Si l'agent non autonome y parvient, la partie se termine sur une victoire.
 - b) Possibilit  de tous les afficher pr alablement ou afficher les uns apr s les autres
 - Sujet 4 : Propagation du virus
 - a) Il sera g rer 2 types d'agents autonomes : les contamin s et les sains.

Un ou plusieurs agents autonomes commenceront par une couleur rouge et leur état de fonctionnement diminuera régulièrement. Si l'un d'entre eux entre en collision avec un agent sain (autonome ou non autonome), ce dernier changera de couleur et pourra à son tour propager le virus.

- b) Tous les agents non contaminés qui passeront au moins une fois par la zone de vaccination, éviteront toute contamination lors des prochaines collisions

- Sujet 5 : Extensions du parcours
 - a) Les bordures du niveau1 sont modifiées et laisse un accès à d'autres niveaux.
 - b) Chaque agent circule librement dans les différents niveaux (visible ou non visible).
 - c) Un seul agent non autonome sera pris en compte et seul le niveau qu'il utilise sera affiché.

- Sujet 6 : Pollution
 - a) Des zones apparaissent régulièrement polluées : elles changent de couleur et un agent qui circulera sur cette zone subira une diminution de son état de fonctionnement.
 - b) Seul un agent autonome (couleur spécifique) est en capacité de dépolluer par un simple passage dans la zone.

- Sujet 7 : Batteries
 - a) Chaque agent possède un niveau de batterie spécifique à 100% et en début de partie et 0% mène à un arrêt définitif. Cette batterie peut être lié à la durée ou la distance parcourue.
 - b) Pour recharger cette batterie, les agents doivent passer et s'arrêter régulièrement à une zone de rechargement (de couleur spécifique).

- Sujet 8 : Variation des vitesses
 - a) Les vitesses de chaque agent dépendent de :
 - Certaines zones de couleurs différentes réduisent la vitesse des agents
 - Leur état de fonctionnement : plus il est faible, plus la vitesse diminue
 - A proximité avec d'autres agents, la vitesse augmente
 - b) Lors d'une collision, plus la vitesse est importante plus la perte de l'état de fonctionnement sera importante

- Sujet 9 : Gestion de l'état de fonctionnement (trop simple) :
 - a) Un nouvel agent autonome à une couleur différente, sera réparateur. Sa collision avec un autre agent autonome augmentera le niveau de fonctionnement de l'autre
 - b) Cet agent se dirigera en priorité, vers l'agent autonome le plus faible.

- Sujet 10 : Inversion des r gles
 - a) Chaque agent devra en contact au moins une fois avec tous les autres robots pour la partie puisse bien termin e.
 - b) Chaque agent sera repr sent    l'aide d'un num ro. Un affichage sugg rera au joueur les num ros des agents qui restent   cibler.
- Sujet 11 : Gestion de Portes
 - a) Initialement le passage est possible pour tous les agents sur les cases blanches et impossible sur les cases bleus (les murs). A pr sent, de nouvelles cases jaunes seront dispos  es al atoirement et seuls les agents non autonomes pourront circuler dessus. De m me que de nouvelles cases oranges seront dispos  es al atoirement et seuls les agents autonomes pourront circuler dessus.
 - b) Sens unique : Une nouvelle contrainte de circulation rendra certaines cases accessibles uniquement dans un sens. Cette contrainte peut  tre cumulable avec la pr c dente.
- Sujet 12 : Pi ces sp ciales
 - a) Les pi ces n'ont plus toute la m me valeur. Leurs tailles varient al atoirement. Cette taille sera associ  e   la valeur obtenue lors de la collision.
 - b) Lors des collisions avec des agents autonomes les pi ces seront   nouveau replac  es al atoirement sur une case non occup  e par un agent ou une autre pi ce
- Sujet 13 : Optimisation des agents autonomes
 - a) Assurer que qu'un agent autonome anticipe et  vite les collisions avec les autres autonomes
 - b) Les agents autonomes sont ralentis mais priorisent leur choix de direction en fonction de la position du non-autonome le plus proche.
- Sujet 14 : G n rateur de murs
 - a) A chaque partie, les emplacements des murs du niveau sont initialement d finis al atoirement. Les deux contraintes doivent  tre respect  es : Le contour du niveau doit toujours  tre rempli de murs et aucune zone libre ne doit  tre totalement inaccessible.
 - b) Le premier niveau sera compos   d'un nombre faible de murs. A chaque fin de partie victorieuse, le joueur passe au niveau suivant et un niveau contenant un plus grand nombre de murs sera propos  .

Etape 4 (S ance 7) :

- Affichage du Score
 - Nombre de pi ces ramass es actualis  en temps r el
 - Afficher  galement la dur e de la partie
 - Gestion du record
 - Sauvegarde et chargement du record   l'aide d'un fichier texte
 - Affichage du record   c t  du score (durant la partie ainsi qu'  la fin de la partie)
 - Affichage du Classement
 - En fin de partie, un classement pr sentera l'historique des parties pr c dentes tri  par score   l'aide d'un algorithme de tri
-
- Option 4.1 : Le record sera associ    un nom de joueur qui sera saisie   chaque d but de partie.
 - Option 4.2 : Repr sentation de l' volution de la partie   l'aide d'un Graphe Matplotlib
 - Option 4.3 : Enregistrement de toutes les positions durant toute la partie afin de permettre de le revoir en replay

Etape 5 (S ance 8) :

- Enrichir l'interface
 - Afficher un menu principal adapt  au sujet
 - Param trage de la partie
- Boutons permettant de red marrer le programme
- Tests et corrections
- R daction du Rapport du projet
- Option 5.1 : Page d'Aide