



LU2IN013 GROUPE 3

Projet Robotique

Enseignant:

- Nicolas BASKIOTIS
- Olivier SIGAUD

Étudiants:

- Ghilas LOUNICI
- Matteo PIETRI
- Rémi RAJARATNAM
- Rodrigo VILA

2022/2023

1.Introduction :

- Objectif du projet
- Importance et pertinence du projet dans le domaine de la robotique
- Utilisation et importance d'une simulation virtuelle
- Choix d'utiliser Python comme langage de programmation

2.Développement logiciel

- Outils Python utilisés pour le développement
- Organisation du code source (modules, packages, etc.)
- Principales fonctionnalités du logiciel de contrôle du robot
-

3.Méthodologie de développement

- Méthodologie de développement utilisée (cycle de vie du logiciel, méthodes agiles, etc.)
- Planification, suivi et gestion des étapes de développement

4.Difficultés et apprentissages

- Difficultés rencontrées lors du développement et comment elles ont été surmontées
- Apprentissages et enseignements tirés du projet
- Améliorations et développements futurs envisagés pour le robot

5.Résultats et évaluation

- Résultats obtenus grâce à l'implémentation du robot en Python
- Comparaison des résultats avec les attentes initiales

6.Conclusion

- Récapitulation des principales réalisations du projet
- Réflexions finales sur l'expérience de développement et les enseignements tirés

1.Introduction

La robotique est un domaine en constante évolution qui suscite un intérêt croissant dans de nombreux secteurs, allant de l'industrie à la médecine en passant par la recherche scientifique. Dans le cadre de ce projet, nous avons entrepris l'implémentation d'un robot en utilisant le langage de programmation Python, offrant ainsi une opportunité d'explorer les aspects pratiques de la robotique et de mettre en pratique nos connaissances en programmation.

L'objectif principal de ce projet était de concevoir et de développer un robot autonome capable d'effectuer des tâches basiques telles que : faire un carré, utiliser un capteur de distance et une caméra. Pour atteindre cet objectif, nous avons commencé par la création d'une simulation. Cette simulation nous a permis de visualiser notamment grâce à Tkinter le comportement du robot dans un environnement virtuel, de tester différentes stratégies et de valider notre implémentation avant de passer à la phase de déploiement sur un véritable robot physique.

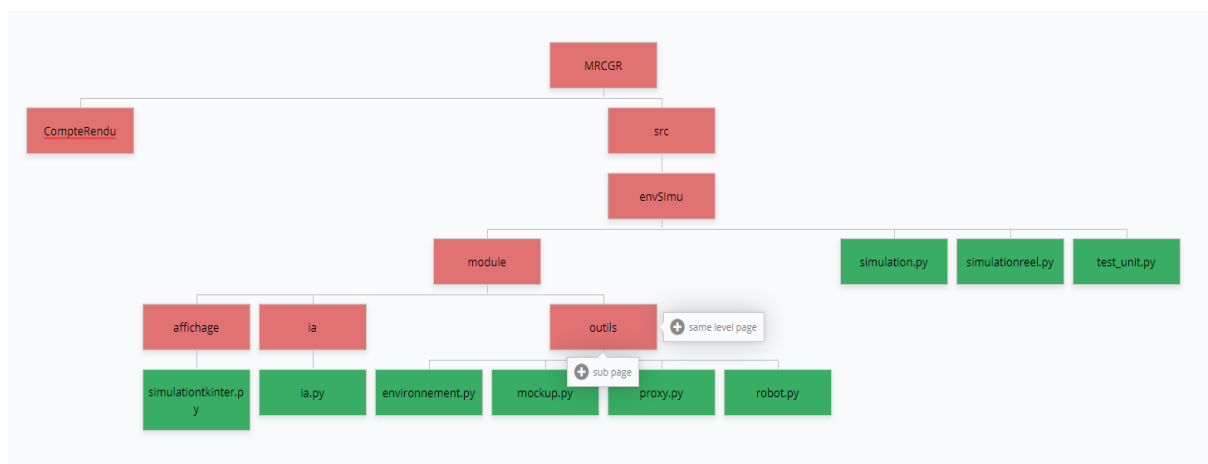
Au cours de ce rapport, nous décrirons en détail l'architecture du robot, la méthodologie de développement que nous avons adoptée, les outils Python utilisés, ainsi que les résultats obtenus grâce à notre implémentation. Nous discuterons également des difficultés rencontrées tout au long du projet et des solutions que nous avons trouvées pour les surmonter.

Ce projet nous a offert une expérience et une ouverture au monde de la robotique et nous a permis d'approfondir notre compréhension des concepts fondamentaux liés à la programmation. Nous espérons que ce rapport sera informatif et qu'il témoignera de notre engagement pour ce domaine.

2. Développement logiciel

2.1 arborescence de fichier et explication de code

Avant de plonger dans les détails de l'implémentation de notre projet de robot en Python, il est important de comprendre comment notre code est organisé et structuré. Dans cette section, nous vous présenterons l'agencement des fichiers du projet et vous expliquerons les parties principales du code. Cela vous permettra d'avoir une vision d'ensemble de l'architecture du projet, des modules et des classes utilisés, ainsi que des choix de conception que nous avons faits



- environnement.py : Contient la classe environnement qui représente le monde dans lequel se trouve le robot et les obstacles et la classe Obstacle représentant les obstacles du monde et ses méthodes.
- robot.py : Contient la classe Robot et ses méthodes
- proxy.py: Contient la classe proxy virtuel et proxy physique pour la simulation et le robot
- ia.py : Contient les différentes commandes du robot
- mockup.py: Classe RobotIN013 d'encapsulation du robot et des senseurs.
- simulationtkinter.py :Création de l'affichage 2D.
- simulationreel.py: Permet la simulation du robot.
- simulation.py: Permet la simulation virtuelle sur tkinter du robot.

- test_unit.py: Contient les tests unitaires des différentes méthodes des classes.

2.2 Tkinter : simulation graphique

Au cours des premières semaines, des recherches ont été menées sur Python et la programmation orientée objet, et les premières classes et méthodes du robot ont été développées. Une simulation sur Tkinter a été créée pour visualiser le comportement du robot dans un environnement virtuel.

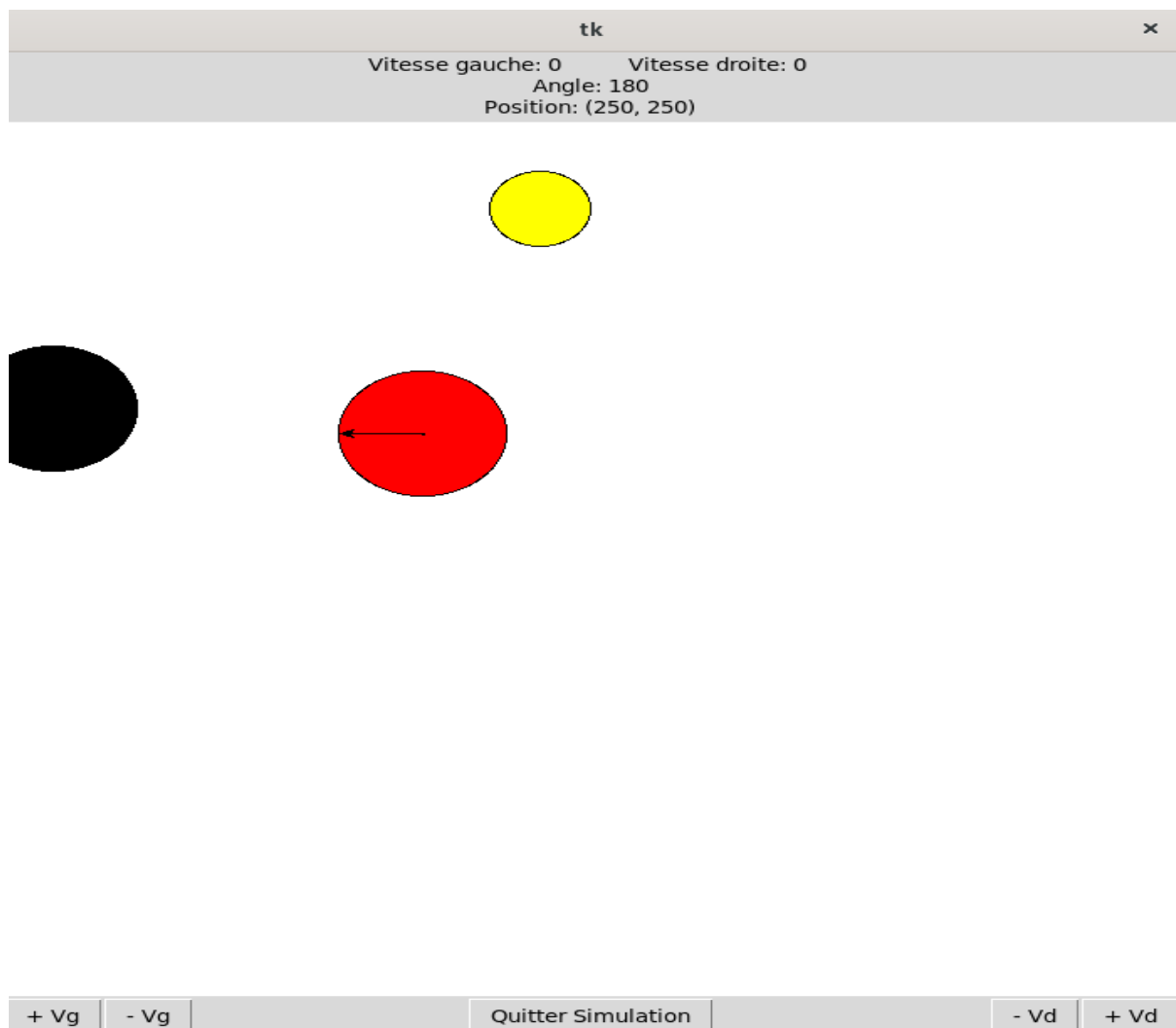


Figure 1:Fenêtre tkinter basique

- robot en rouge avec une flèche donnant la direction du robot
- en noir et jaune nous avons les obstacles, si le robot entre en collision avec ceux-ci, il s'arrête (cf figure 3)
- des données telles que la vitesse des roues, l'angle et la position du robot
- des boutons sur lesquelles on peut augmenter les vitesses des roues droite et gauche

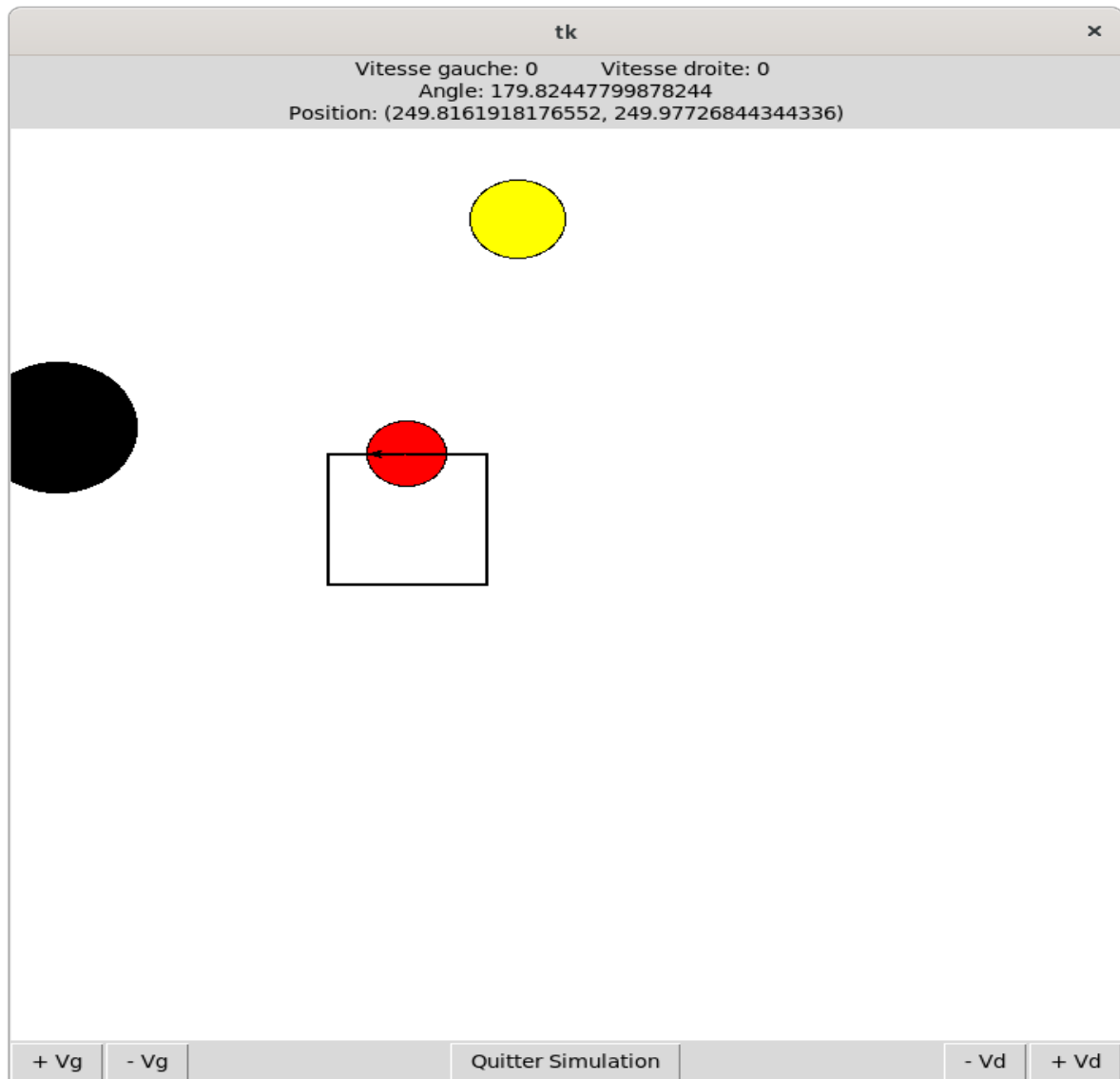


Figure 2:Fenêtre tkinter représentation carré

- robot entrain de faire un carré sur la simulation avec l'IA carré
- on peut modifier la taille du carré, le faire dans le sens qu'on veut

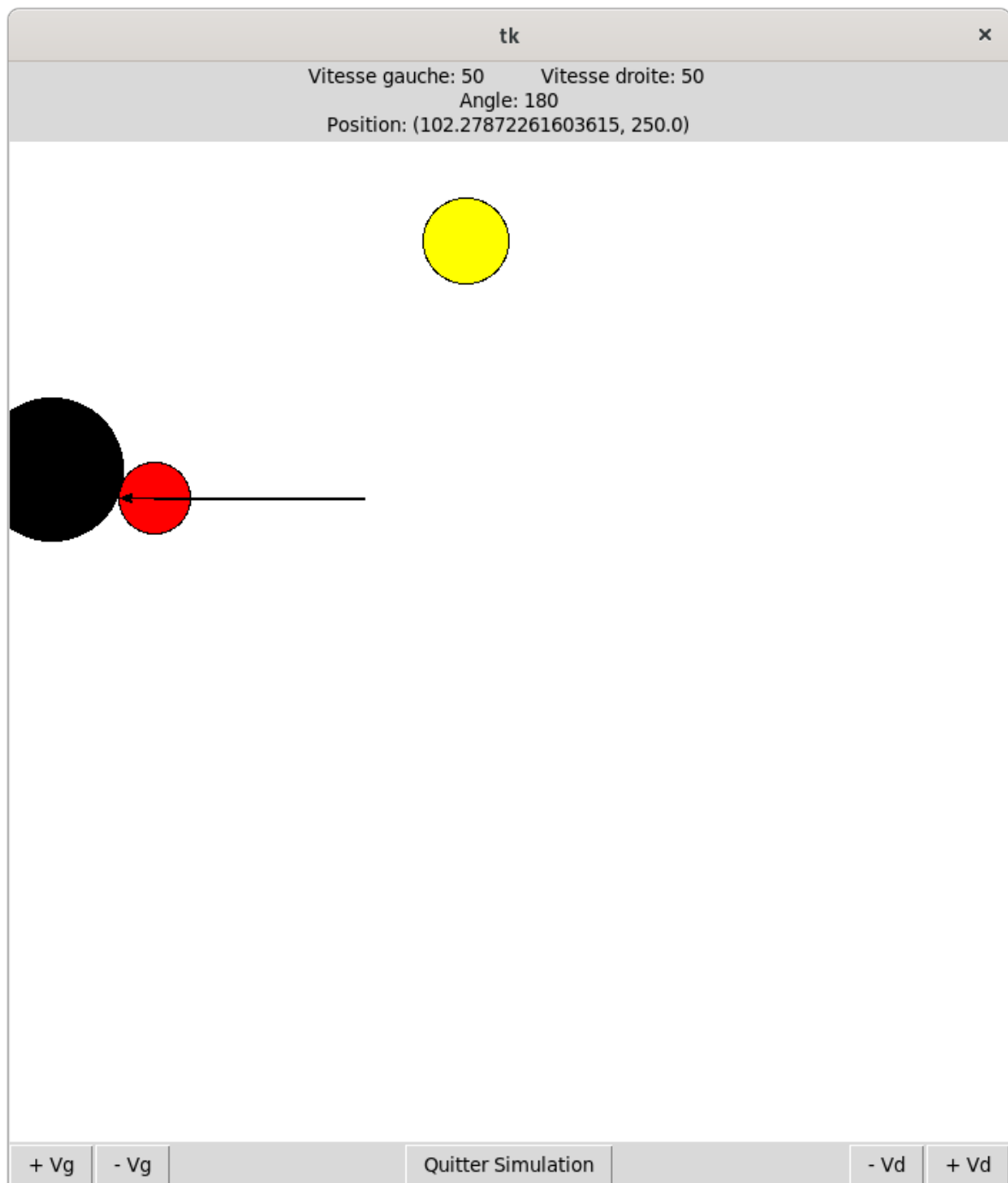


Figure 3: Fenêtre tkinter représentation collision
-lorsque le robot va entrer en collision avec un obstacle il s'arrête avant de le toucher

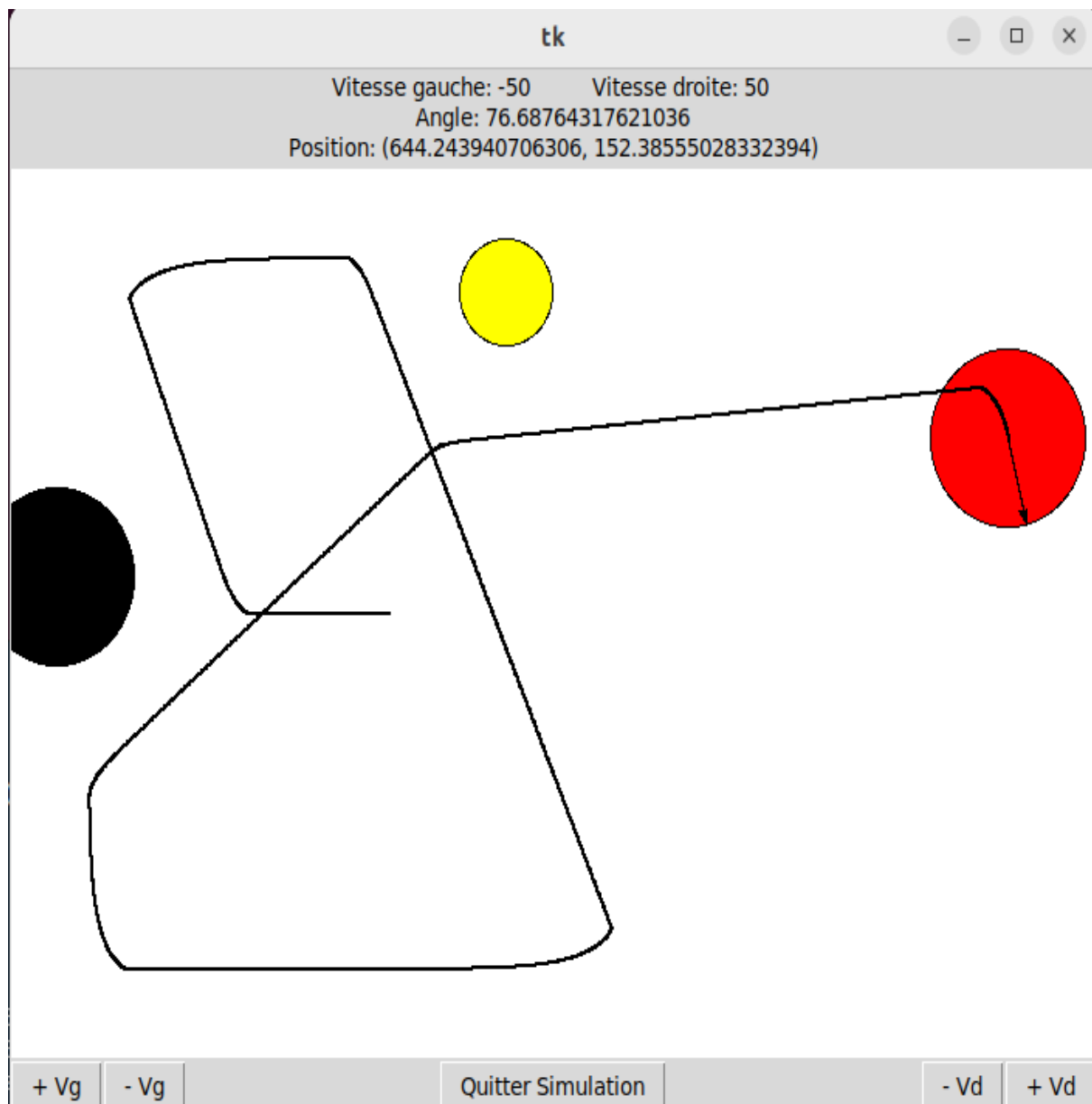


Figure 4: Fenêtre tkinter esquive collision/environnement
-lorsque le robot va entrer en collision avec un obstacle il s'arrête puis change de direction et c'est pareil avec les bordures de l'environnement

2.3 Du virtuel au réel

Après que notre simulation virtuelle fonctionne comme nous le voulions il était temps de passer au robot, nous avons donc implémenté des IA telles que:

- IAseq qui permet de faire une séquence d'actions
- IAloop qui permet de répéter une action plusieurs fois
- IAifte qui permet de faire une action jusqu'à un point puis de changer si on passe une condition

pour utiliser celle-ci nous avons forcément besoin d'implémenter des IA de commandes bien plus basiques

- ParcourirDistance qui permet de parcourir une certaine distance
- Arrête qui stop le robot
- TournerDroite et TournerGauche également

Avec tout ça nous pouvions donc commencer l'implémentation du carré qui est "simplement" [ParcourirDistance,TournerDroite/Gauche,Arrete] 4 fois, mais les choses n'étaient pas aussi simples.

L'échelle et donc le passage de distance/angle entre notre simulation virtuelle et la réelle était assez complexe car la distance que l'on mettait sur la SV ne représentait aucune distance réelle (idem pour les angles).

Le premier travail consistait donc à se calibrer sur le robot réel, sur l'environnement réel afin de faire concorder les données virtuelles au monde réel.

Mais nous parlerons plus en détails de la comparaison entre le virtuel et le réel dans un prochain chapitre.

3.Méthodologie de développement

3.1 Méthodologie utilisée

Lors du développement de ce projet, nous nous sommes organisés grâce à la méthode agile. Celle-ci est un cadre de gestion de projet qui divise les tâches entre tous les membres du groupe. Chaque semaine, nous faisons un récapitulatif du travail effectué durant la semaine passée. Puis, nous nous décidons des tâches à réaliser et à s'attribuer pour un objectif commun à atteindre pour la semaine suivante.

Cette méthode favorise le travail collectif et la collaboration. Donc, une communication claire et efficace entre les personnes était une nécessité. Ainsi, l'avancement du processus est plus facile à suivre pour le client et les fournisseurs peuvent ensuite être plus flexible et s'adapter à ses demandes. Elle permet de revoir rapidement leur stratégie sans pour autant avancer dans une mauvaise direction.

3.2 Planification, suivi et gestion des étapes de développement

Le projet a débuté par la création d'une équipe de cinq personnes, la définition du nom du projet et l'entretien avec le client pour clarifier les objectifs. Une communication efficace a été établie via un groupe Discord. De nombreuses réunions ont été faites à l'université facilitant la communication, la visualisation et la compréhension des points de vue de chacun. Des outils tels que GitHub et Trello ont été mis en place pour organiser et planifier les tâches.

Au cours des premières semaines, des recherches ont été menées sur Python et la programmation orientée objet afin que l'équipe s'habitue à cet environnement. Par la suite, les premières classes et méthodes du robot ont été développées.

Des recherches sur des simulateurs virtuels ont été menées afin de visualiser le comportement du robot dans un environnement. Après débat, il a été décidé que tkinter était le plus adapté pour cette tâche grâce à son intégration avec Python et ses différentes fonctionnalités graphiques. Des itérations successives ont permis d'améliorer la simulation graphique, de résoudre les problèmes de déplacements et de collision et d'ajouter des fonctionnalités telles que la hauteur des obstacles et la vitesse du robot.

L'implémentation de l'intelligence artificielle (IA) a été un défi majeur du projet. Différentes approches ont été explorées, notamment la création d'un contrôleur et la mise en place de fonctions IA directement dans le robot. Des ajustements et des tests ont été effectués pour améliorer les performances de l'IA. Des essais ont été réalisés sur un vrai robot, ce qui a nécessité des ajustements supplémentaires pour prendre en compte les différences entre la simulation virtuelle et la réalité. Malgré quelques retards et difficultés rencontrés en cours de route, l'équipe a réussi à progresser dans le projet et à améliorer la qualité du code. Les tests unitaires ont été ajoutés, et la documentation a été régulièrement mise à jour.

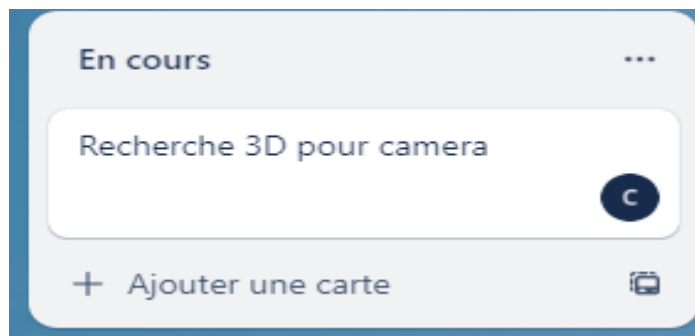
La méthode Agile Scrum a permis une approche collaborative, flexible et adaptative pour le développement de ce projet. La communication transparente entre les membres de l'équipe a facilité le suivi de l'avancement et la prise de décisions rapides. Les itérations régulières ont favorisé l'adaptabilité aux changements et l'amélioration continue. Les outils tels que GitHub et Trello ont été utilisés pour une planification et une gestion efficaces des tâches. L'implémentation de l'intelligence artificielle a été un défi majeur, mais l'équipe a su trouver des solutions et effectuer des ajustements pour améliorer les performances. Les tests unitaires et la documentation ont renforcé la qualité du code. Dans l'ensemble, la méthode Agile Scrum a permis d'atteindre nos objectifs en se concentrant sur les travaux, la collaboration et l'adaptation aux besoins changeants.

4. Difficultés et apprentissages

Concernant le capteur, nous n'avons pas réussi à le faire par manque de temps et une mauvaise compréhension de cet outil.

La perte d'un membre au cours d'un projet peut constituer une difficulté significative pour l'équipe restante. Lorsqu'un membre quitte le groupe, cela peut entraîner un déséquilibre dans la répartition des tâches et une perte de connaissances spécifiques qu'il ou elle avait apportées au projet.

C'est effectivement ce qui nous est arrivé avec notre collègue Colin qui a décidé de ne plus venir aux cours et de laisser en standby toutes ces tâches comme la plus importante: la caméra.



Cela fait depuis la semaine 8 que dans le trello cette fiche n'a pas bougée. Le "C" représentant effectivement le nom de Colin...

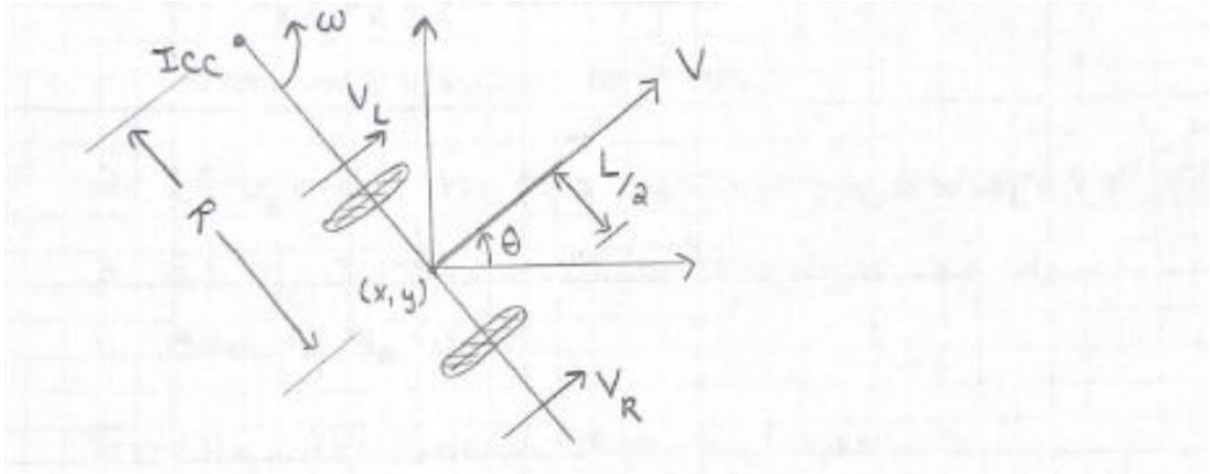
La réaffectation des responsabilités devient nécessaire pour combler le vide laissé par le membre parti. Cela peut entraîner une charge de travail supplémentaire pour les membres restants, qui doivent s'adapter et prendre en charge les tâches qui étaient initialement attribuées à la personne absente. Dans notre cas cela était très compliqué car nous l'avons appris en période de révision d'examens, et nous avons déjà des tâches au niveau du robot pour l'implémentation du carré et du capteur, le temps nous a donc manqué pour faire la caméra.

Nous avons été assez naïfs à ce niveau et s'il y avait quelque chose à refaire dans ce projet, c'est la mise en place de deux personnes sur la caméra au lieu de donner les pleins pouvoirs à une seule personne, cela fonctionne pour chaque grosse tâche.

Cependant c'est un très bon apprentissage pour la suite et nous saurons que malgré la mise en place d'un trello pour suivre les tâches, il faut demander bien plus souvent aux personnes concernées par celles-ci car les mauvaises surprises peuvent vite arriver.

Si nous étions arrivés à implémenter ces 2 outils, nous avions pensés à faire des parcours d'obstacle, des créneaux, des courses entre robots et même tenté de faire un mini jeu du style rocket league ou ce serait des tirs avec des ballons et un robot gardien qui serait chargé de suivre la balise émise par la balle.

5. Résultats et évaluation



Ce que nous avons utilisé pour le déplacement du robot est le “differential driving kinematic”. C’est un modèle cinématique couramment utilisé dans les systèmes robotiques, en particulier dans les robots mobiles. Ce modèle décrit le mouvement et le contrôle d’un robot doté de deux roues ou chenilles séparées, chaque roue étant entraînée indépendamment.

Ce qui nous a valu beaucoup de calculs et sûrement des choses bien plus compliquées que nos camarades et cela s’est ressenti sur le robot réel dans lequel nous avons de nombreux problèmes d’échelle et en même temps lors de la réalisation d’un carré nous avons un accumulateur de distance qui faussait le carré.

Pos initial (250,250)			
	Mesures voulues	Mesures simulation	Ecart relatif
Avancer	50	50,04	0,08%
Position: (199.96137475967444, 250.0)			
Angle initial (180°)			
	Mesures voulues	Mesures simulation	Ecart relatif
Tourner	90	90,04	0,05%
Angle: 89.96464894430613			

Nous avons pas mal d'écart entre la réalité et la simulation et c'est ce qui nous a valu le plus de problèmes, par manque de temps nous ne les avons pas tous résolus mais dans l'ensemble.

Cette méthode de calcul était certes complexe, mais nous apportait des possibilités bien plus grandes. Telles que le tracés de forme circulaire qui nous était facilités.

Il y avait peut-être moyen de faire plus simple mais au final cette méthode de calcul aurait pu nous offrir plus d'opportunités avec plus de temps, nous ne regrettons donc pas cette prise de risque.

6.Conclusion

En conclusion, notre projet d'implémentation d'un robot en Python a été une expérience enrichissante qui nous a permis de plonger dans le monde de la robotique et d'approfondir nos compétences en programmation. Tout au long de ce projet, nous avons fait face à divers défis et avons dû faire preuve d'adaptabilité pour les surmonter.

Nous avons commencé par la création d'une simulation sur Tkinter, ce qui nous a permis de visualiser le comportement du robot dans un environnement virtuel et de tester différentes stratégies. Ensuite, nous avons progressivement amélioré notre code en ajoutant des fonctionnalités telles que la détection d'obstacles, le contrôle de la vitesse et l'implémentation de l'IA. Nous avons également effectué des tests sur un véritable robot physique, ce qui nous a confrontés à des défis supplémentaires.

Tout au long de ce parcours, nous avons rencontré des difficultés, telles que la séparation de la simulation graphique et non graphique, la maîtrise de GitHub, l'organisation du code et la perte d'un membre de l'équipe. Cependant, nous avons fait preuve de résilience, de collaboration et de persévérance pour continuer à avancer malgré ces obstacles.

Ce projet nous a également offert de précieuses opportunités d'apprentissage. Nous avons acquis des compétences en programmation orientée objet, en utilisation de bibliothèques Python telles que Tkinter, ainsi qu'en résolution de problèmes liés à la robotique. Nous avons également appris à travailler en équipe, à communiquer efficacement et à gérer notre temps et nos ressources de manière optimale.

En résumé, ce projet de robot en Python nous a permis de mettre en pratique nos connaissances en programmation et de nous immerger dans le domaine passionnant de la robotique. Nous sommes fiers du travail accompli malgré les difficultés rencontrées, et nous espérons que notre rapport témoigne de notre engagement et de notre détermination à relever les défis. Ce projet nous a véritablement ouvert de nouvelles perspectives et nous a préparés pour de futurs projets et opportunités dans le domaine de la robotique et de la programmation.