

RAPPORT DE PROJET

SIMULATION ROBOTS ET INCENDIES

Introduction

L'objectif de ce projet était de développer une simulation informatique de gestion d'incendies et de robots pompiers. Ce compte-rendu présente notre démarche de travail, les choix effectués lors du développement et une synthèse des tests réalisés. Nous espérons ainsi démontrer notre capacité à mener à bien un projet de programmation en groupe, tout en mettant en évidence les compétences techniques et organisationnelles acquises au cours de ce projet.

Répartition du Travail et Gestion de Projet

Afin d'assurer une gestion efficace du projet, nous avons réparti les tâches comme suit :

- Tristan JOLY : Conception des cartes de simulation, gestion des algorithmes de déplacement des robots et intégration des fonctionnalités d'extinction d'incendies.
- Tharushan UTHAYAKUMAR : Lecture et affichage des cartes sur l'interface graphique, développement de l'interface utilisateur et implémentation des interactions utilisateur.

Nous avons utilisé un gestionnaire de version (GitLab) pour suivre les modifications apportées au code et faciliter la collaboration. Les branches de développement ont été utilisées pour chaque fonctionnalité majeure, et des revues de code régulières ont permis de maintenir la qualité et la cohérence du code. Le graphe du dépôt est disponible sur notre dépôt du projet sur GitLab. Le travail a été réparti de manière équitable, le fait de pouvoir être à deux sur ce projet a grandement aidé pour sa réalisation finale. La répartition des tâches étant définie clairement à l'aide de To-do liste, lorsque nous rencontrions un problème provenant d'un endroit, nous savions à qui demander pour le régler.

Choix d'Extensions

Pour enrichir notre simulation, nous avons ajouté l'extension suivante :

- **Propagation du Feu par le Vent** : Implémentation d'une fonctionnalité où le vent permet une propagation du feu sur la carte avec un choix de difficulté compris entre 1 et 3

Ainsi, nous avons 3 modes de jeu différents : 'Suivre un scénario prédéfini', 'Jeu avec le vent' et 'Jeu Solo'. Dans les modes de jeu 'Jeu avec le vent' et 'Jeu solo', il est possible de mettre de la musique pour un meilleur confort de jeu.

P.S : Les musiques ont été pitchées, si nécessaire, pour qu'elles soient libres de droits.

Choix d'Ergonomie et de Design

Pour rendre notre application intuitive et agréable à utiliser, nous avons pris les décisions suivantes :

- **Interface Utilisateur** : Conception d'une interface claire et épurée, avec des images distinctives pour chaque type de robot et de terrain.
- **Implémentation d'un Menu** : Implémentation d'un menu permettant à l'utilisateur de choisir le mode de jeu dans lequel il souhaite évoluer ainsi qu'un sous-menu permettant de charger la carte choisie en fonction du mode de jeu souhaité.

Choix d'Implémentation

Nous avons utilisé les technologies suivantes pour développer notre simulation :

- **Langage de Programmation** : Python, pour sa flexibilité et ses puissantes bibliothèques de gestion d'interface graphique et de traitement algorithmique.
- **Programmation Orientée Objet (POO)** : Utilisation des concepts de POO pour structurer le code de manière claire et modulable. Nous avons également utilisé les propriétés de polymorphisme propre à la POO.
- **Bibliothèques** : Utilisation de bibliothèques comme Tkinter et Pygame pour l'interface utilisateur et la conception du jeu vidéo.

Pour gérer la vitesse des robots, nous avons implémenté une variable nommée `temps_par_actualisation` qui découle d'une sérendipité. `temps_par_actualisation` est égale à $(\text{taille_case} * \text{nb_cases}) / 10000$. Il s'agit donc de notre unité de temps pour ce projet.

Synthèse des Tests

Des tests unitaires à l'aide d'assert ont été réalisés pour garantir la fiabilité et la robustesse de la simulation. Les preuves de ces tests peuvent être trouvées dans le code source du projet. De plus, pour éviter de casser le code en cas d'erreur, nous avons mis en place des tests qui renvoient des messages d'avertissement précis dans le shell lorsqu'une action impossible est tentée (par exemple, lorsque le robot essaie de sortir des limites de la carte), sans interrompre l'exécution du programme.

Difficultés Rencontrées

Nous avons rencontré plusieurs obstacles au cours de ce projet :

- **Utilisation de GitLab** : L'utilisation du gestionnaire de versions GitLab a nécessité une période d'adaptation pour gérer efficacement les branches et les fusions de code.
- **Débuggage de la fonction tour** : Une erreur due à l'inversion de deux variables dans la fonction `tour`, qui gère le déplacement en fonction de la vitesse de chaque robot, nous a pris un temps considérable (une semaine) à identifier et à corriger.
- **Dictionnaires python** : Nous avons utilisé plusieurs dictionnaires python avec comme clés des objets de nos class créés. Cependant, dû au fait que ce type d'objet (dictionnaire) est très permissif, nous avons rencontré de nombreux bugs, pour les résoudre, nous avons dû utiliser la fonction `.keys()` retournant les clés.

Conclusion

Ce projet nous a permis de renforcer nos compétences en programmation, en gestion de projet et en travail d'équipe. Nous avons appris à utiliser des outils de versionnage, à gérer des tâches complexes de manière collaborative, et à créer une application fonctionnelle et ergonomique. Nous sommes fiers du travail accompli et des compétences acquises, et nous sommes prêts à appliquer ces connaissances à de futurs projets.