

Projet de qualité de programmation

Jeu de robots écraseurs

Réalisé par :

- Asmae YASSINE
- Salma Insaf BENAMARA
- Najwa KWIDRI
- Fatima Zahra MAARAF

Encadré par :

- M. Stéphane RIVIERE

Remerciement

Nous tenons à remercier monsieur Stéphane Rivière qui nous a appris les principes de la programmation orientée objet et les astuces pour avoir un programme de bonne qualité ainsi que les méthodes de réalisation de tests unitaires et qui nous a orienté et encadré tout au long de la réalisation du projet.

Table des matières

1. Introduction :.....	3
3. Description du projet :	4
5. Diagramme de classe :	5
6. Répartition des tâches :	6
7. Déroulement du projet :	7
8. Les tests unitaires des classes :	8
9. Les difficultés rencontrées :	8
10. Remaniements effectués :.....	8
11. Outils utilisés :	9
12. La pertinence des outils utilisée :.....	10
13. Conclusion.....	11

1. Introduction :

Dans le cadre de notre 3ème année licence Méthode Informatique Appliquée à la Gestion d'entreprise Informatique nous nous sommes intéressés à l'implémentation d'un jeu de robot.

Ce projet a été élaboré dans le cadre de l'unité d'enseignement de qualité de Programmation sous la responsabilité de Monsieur Stéphane Rivière.

Tout au long du projet nous avons essayé de respecter les consignes demandées :

- ✓ Utiliser Git.
- ✓ Utiliser le développement dirigé par les tests.
- ✓ En écrivant du code propre, accompagné de ses tests (utiliser Doctest).
- ✓ En remaniant le code pour garder ce code propre et pouvoir le faire évoluer.

Pour cela, nous avons réparti les tâches entre nous, pour que chacun puisse développer sa propre partie et bien sûr on met tout sur git, comme ça chacun peut consulter les codes des autres et discuter ce qui doit être modifier ou remanier lors des réunions.

3. Description du projet :

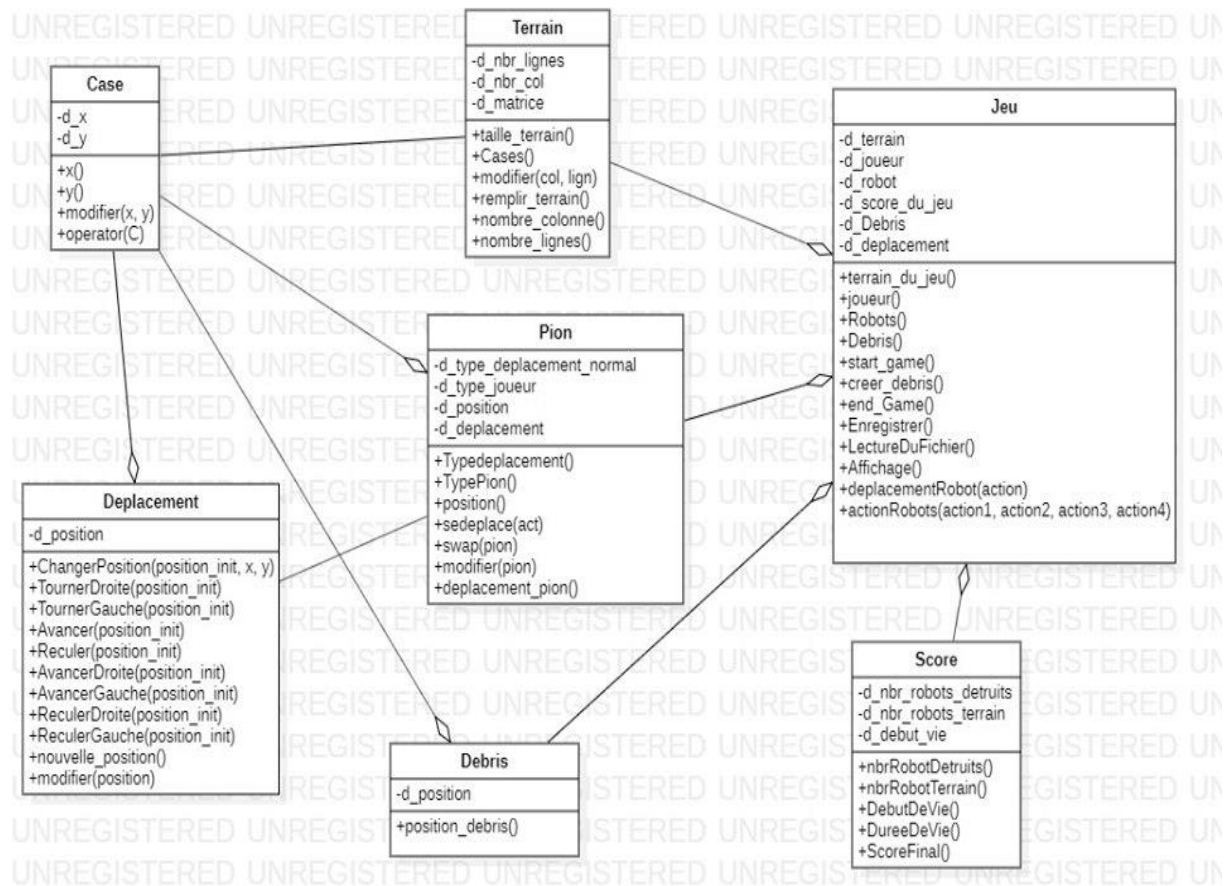
Notre projet consiste a réalisé un jeu de robot. Parmi les classes que nous avons créées, la classe jeu permet de gérer le jeu dès son début à sa fin. Pour qu'un jeu puisse exister il a besoin d'un ensemble de ressources qui sont : pion, terrain, score et débris.

Une case est construite à partir des coordonnées x et y et a une méthode modifier qui facilite la modification de ces coordonnées. Plusieurs cases composent un terrain.

La classe pion est construit à partir d'une référence d'une case, d'un type de pion qui peut être soit un joueur soit un robot et d'un type de déplacement qui démontre s'il peut se déplacer 8 déplacements ou quatre. Un pion peut se déplacer dans le terrain selon la direction choisi grâce à la classe déplacement qui renvoie une nouvelle position à chaque action choisie. La classe déplacement permet de définir le changement de position d'un pion avec les huit déplacements possibles. La classe score permet de calculer la durée du jeu et le nombres de robots détruits. Sur un terrain de jeu on trouve un seul pion joueur, plusieurs pions robots et des débris qui résultent de la destruction de deux robots ou plus.

5. Diagramme de classe :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il montre la structure interne et permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.



6. Répartition des tâches :

❖ Les classes :

Taches	Participants
Classe jeu	Insaf
Classe Score	Najwa
Classe Debris	Batoul
Classe Pion	Asmae
Classe Deplacement	Najwa
Classe Terrain	Batoul
Classe Case	Batoul
Main	Asmae

❖ Les tests :

Taches	Participants
TestPion	Asmae
TestDéplacement	Najwa
TestCase	Batoul
TestTerrain	Batoul
TestJeu	Insaf
TestScore	Najwa

En ce qui concerne la compilation, le débuge, la rédaction du rapport et la présentation, nous nous sommes organisés comme suit :

Tâches	Participants
Compilation et débuge	Insaf & Najwa
Rapport et présentation	Asmae & Batoul

7. Déroulement du projet :

Dans un premier lieu on a créé les classes qu'on a constaté nécessaire pour le fonctionnement du projet. Pour chaque classe créée on a ajouté les méthodes et données appropriés pour son fonctionnement.

Au début, on a commencé par créer une classe case qui représente une position à l'aide des coordonnées x et y, puis la classe terrain qui est construite d'un certain nombre de case donnée par l'utilisateur en forme de ligne et colonne puis on a créé deux classes robot et joueurs qui contenait une position et un type de déplacement ainsi qu'une méthode qui leurs permet de se déplacer. Éventuellement on a constaté que on a deux classes identiques alors on a décidé de créer une seule classe qu'on a appelé pion en rajoutant une donnée type de pion pour différencier entre un pion robot et un pion joueur.

Ensuite, on a créé la classe déplacement qui contient une position de type case, une méthode changer position et des méthodes pour les huit déplacements possibles. On a aussi créé la classe score qui stocke le nombre de robots sur le terrain du jeu, le nombre des robots détruits et le début du jeu en seconds.

Finalement, on a créé la classe jeu qui contient une liste de pions robots ainsi qu'une liste de débris, un pion joueur, un score du jeu et un terrain. Cette classe contient une méthode configurer terrain qui modifie le terrain du jeu en fonction des informations données par l'utilisateur, une méthode créer débris qui permet de créer un débris et lui affecter la position où deux ou plusieurs robots ont été détruits, une méthode start_Game qui affiche l'état initial des composants du jeu ainsi que la position des pions et débris après chaque déplacement et une méthode end_Game qui affiche que la partie est finie si il n'y a plus de robots sur le terrain ou si le joueur est mort.

8. Les tests unitaires des classes :

Les tests unitaires ont été faits à l'aide de l'infrastructure de TEST_CASE, l'objectif principal des tests unitaires est de garantir la qualité du code de production.

Nous avons choisi TEST_CASE car c'est l'infrastructure de test qu'on a vu plus en cours, et nous l'avons trouvé facile à utiliser et plus adaptée à C++. On a fait les tests sur la majorité des méthodes des classes qu'on a créé.

- TestPion
- TestDeplacement
- TestTerrain
- TestScore
- TestCase
- TestJeu

9. Les difficultés rencontrées :

- La première difficulté été de se décider sur les classes à utiliser et leurs fonctionnements.
- Le respect de la charte « code propre ». En effet on s'apercevait du code commun mais on avait des difficultés à effectuer les remaniements « proprement ».
- Le projet contient des parties plus ou moins faciles et d'autres plus dures et il faut rédiger le rapport et la présentation PowerPoint en parallèle du code, c'est pour ces raisons qu'on a trouvé des difficultés à répartir les tâches entre nous sachant qu'il faut prendre en considération les capacités de chaque membre.

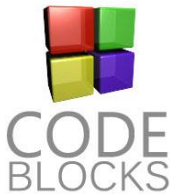
10. Remaniements effectués :

- ✓ Après avoir coder les classes initiales nous avons constater que y en a des classes qui représente un même objet ; remaniement de classe Joueur et Robots dans une classe Pion.
- ✓ La méthode deplacementRobot est une en méthode qui déplace chaque robot après le déplacement du joueur, dans ce cas cette méthode fait deux opérations au même temps, du coup nous avons crée une autre méthode actionRobots qui vérifie dans quel direction le robot doit se déplacer.

- ✓ Dans les 8 méthodes de la classe déplacement, nous avons constaté qu'il y a une ligne qui se répète dans chaque méthode, c'est une ligne qui change la position d'un Pion, du coup nous avons créé la méthode ChangerPosition.

11. Outils utilisés :

StarUML : est un outil de modélisation UML (Unified Modeling Language) qui vient se substituer aux outils payants tels que « IBM Rational Rose » ou « Borland Together », L'outil propose les diagrammes UML nécessaires à une bonne modélisation.



CodeBlocks : est un environnement de développement intégré libre et multiplateforme. Il est écrit en C++ et utilise la bibliothèque wxwidgets. Code::Blocks est orienté C et C++, mais il supporte d'autres langages comme FORTRAN ou le D.

GitHub : est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.



Doctest est un nouveau framework de test C++ mais est de loin le plus rapide en temps de compilation (par ordre de grandeur) et en temps d'exécution par rapport à d'autres alternatives riches en fonctionnalités. Il apporte la capacité des langages compilés tels que D / Rust / Nim à avoir des tests écrits directement dans le code de production grâce à un lanceur de test rapide, transparent et flexible avec une interface propre.

12. La pertinence des outils utilisée :

CodeBlocks c'est l'environnement de développement C++, nous a permis de codée et de développer nos compétences dans la programmation, exécuté nos codes et d'ouvrir plusieurs projets en même temps pour effectuer les différents tests

Le GitHub été une découverte pour nous il nous a permis de partager en ligne l'avancement de chaque une d'entre nous pour qui soit mise à jour, soit remanier ou juste consulté par les autres membres du groupe.

DocTest est un framework de test C++, qui nous a permis d'exécuter nos tests de manière rapide.

StarUML est un outil de modélisation UML, qui nous a permis de créer le diagramme de classe proprement et de manière lisible.

13. Conclusion

Ce projet nous appris de mettre en pratique ce que nous avons étudié dans les cours de qualité de projets en écrivant un code propre et le remanier grâce aux tests effectuée au bout de chaque code et s'intéresser à sa qualité et sa compréhensibilité de la part de l'utilisateur.

Le travail en groupe est aussi une expérience très importante, qui nous a permis d'échanger nos connaissances et acquérir de nouvelles compétences, et bien sur l'outils GitHub que nous avons vu lors du cours à faciliter le partage du code entre nous.

La bonne entente entre les membres du groupe a facilité le travail sur ce projet, il y avait une coordination et une entraide entre nous ce qui a permet de réaliser le projet loin du stress et de la manière la plus saine possible.