

Projet-Informatique

Clabaut Ewan
Marmelat Antoine
Pereira Alaoui Lina

CY-BER PATH

27 mai 2024

Equipe

L'équipe est constituée de trois membres: Lina, Antoine et Ewan. C'est une équipe mixte qui a été choisie par affinité.

Description

Le jeu consiste à déplacer des robots glisseurs sur une grille en 2D pour atteindre des cibles définies aléatoirement en un minimum de mouvements. Il peut se jouer à 2 joueurs ou plus sans limite définie. Les robots se déplacent sur une grille créée aléatoirement, cette grille contient des murs placés aléatoirement qui les empêchent d'avancer librement. Au début de la partie les joueurs choisissent un niveau de difficulté parmi trois possibilités et le nombre de manches pour que la partie se termine. C'est les joueurs qui saisissent leurs estimations du nombre de mouvements. Le joueur avec le moins de mouvements débute, si il arrive à atteindre la cible avec le nombre de mouvements qu'il a choisi auparavant, il gagne 2 points. Si il atteint la cible avec moins de mouvement il perd un point et s' il n'y parvient pas les adversaires gagnent tous un point. Le gagnant est le joueur avec le plus de points à la fin du nombre de manches.

Objectifs

Notre objectif commun est de réussir le projet en mettant tous la main à la patte et surtout de le faire en y prenant du plaisir ! La réalisation de ce jeu est comme une concrétisation de notre année d'informatique, nous voulons être fière de nous.

Construction du projet

- Organisation du groupe

Pour l'organisation, nous avons décidé de travailler chacun de notre côté et de faire des points à chaque cours d'informatique. Nous n'avons donc pas attribué de rôle spécifique à chacun. Nous avons tous les trois un niveau d'informatique assez similaire, de ce fait tout le monde avait la capacité de construire le code. Antoine a commencé avec la base du code qui implique l'affichage de la grille, des cibles et des robots. Lina a ensuite continué en implémentant le début de la partie et Ewan s'est occupé de la fin. De cette façon les tâches étaient assez bien réparties. Bien sûr la communication est la clé, c'est pourquoi nous n'hésitons pas à échanger à l'école ou sur Whatsapp. Le code était stocké sur un dépôt Git ce qui nous a permis de pouvoir modifier le code à notre guise tout en voyant les modifications apportées par les autres membres du groupe et cela de façon rapide et organisée. Nous avons travaillé régulièrement sur le code, chez nous et pendant les td d'informatique. Le plus gros flux de travail a été durant le dernier week-end avant le rendu car il a fallu mettre en ordre, apporter les détails et surtout corriger les bugs et les problèmes d'affichage ce qui nous a pris le plus de temps.

- Réalisation du projet

Pour la réalisation du projet informatique de fin de séquence, nous avons choisi le projet CY-BER PATH. Nous avons commencé par la création de l'environnement du jeu, à savoir la mise en place de la grille rectangulaire/carré. Voici une explication détaillée des différentes fonctions et leur rôle dans le déroulement du jeu. Les premières fonctions sont 'initialiserGrille', 'placerCibles', 'murExterieurs' et 'placerRobots' elles servent à initialiser la grille, les murs, les cibles et les robots. Ensuite on affiche la grille avec 'afficherGrille'.

Toutes les fonctions contiennent des instructions tel que des if, while et for pour être sûr que le programme soit correct.

Vient ensuite le placement des derniers éléments et la mise en place du déroulement du jeu avec les fonctions suivantes : 'NombreJoueurs', 'choisirRobotCible', 'chronometrer', 'choix_player', 'choix_direction' et 'deplacement'.

Pour chaque manche, le programme choisit un robot et une cible, affiche la grille, lance un chronomètre pour les joueurs, puis déplace le robot en fonction des choix des joueurs.

Après chaque déplacement la grille est affichée. Elle est parcourue en entier et imprime chaque cellule avec les éléments correspondants.

Nous avons créé une fonction 'estPositionValide' pour s'assurer lors du placement d'une cible ou d'un robot que la position respecte les limites de la grille et est non occupée par un mur ou un autre robot.

Plus tard nous avons ajouté la couleur et les emojis dans le terminal grâce aux ressources données dans le cahier des charges. La fonction 'void_deplacement' elle vérifie à l'aide d'un switch break que le robot peut se déplacer correctement sans heurter un mur ou un autre robot.

- Problèmes rencontrés

Certaines fonctionnalités du jeu ont été difficiles à créer, nous avons notamment eu beaucoup de mal avec l'affichage des murs de la grille de jeu . Nous n'arrivions pas à aligner les murs au départ en utilisant ' | ' pour les murs verticaux et ' - ' pour les murs horizontaux. Nous avons donc dû changer notre affichage pour les murs verticaux en utilisant '+---+', avec des plus pour remplir les coins des cases.

Ensuite il a fallu colorer seulement certains murs en rouge que nous appellerons 'murs interdits' car dès qu'un robot rencontre un de ces murs il doit être bloqué. Ce n'a pas été facile car il a fallu stocker les coordonnées de ces murs interdits pour pouvoir les utiliser en tant voulu et pouvoir les colorier à notre souhait. La grille s'affiche donc correctement avec un fond blanc, les murs interdits en rouge et les murs normaux en bleu.

Enfin nous avons rencontré un autre problème qui était de s'assurer que les robots se déplacent correctement sur la grille en tenant compte des murs aléatoires et des cibles. Car certaines fois les robots ne s'arrêtaient pas au bon endroit c'est-à-dire qu'ils allaient parfois après les murs interdits ou parfois ils s'arrêtaient avant d'atteindre les murs. Il a donc fallu implémenter des fonctions spécifiques pour vérifier les positions valides et gérer les déplacements avec des boucles et des conditions appropriées. Les robots bougent maintenant correctement et s'arrêtent à chaque fois qu'ils croisent un mur.