

Rapport du projet d'informatique

L'équipe est composée de trois élèves de MEF1 : HAMELIN Simon, MARMELAT Paul, et BENDAOUED Idris.

Organisation du groupe :

Pour la réalisation du projet, nous nous sommes répartis les rôles. Dans le groupe, Paul était le bêta-testeur, il s'occupait de tester le code pour trouver ce qui n'allait pas et les points à améliorer. Bien qu'il soit moins à l'aise en informatique, Paul a voulu persévérer dans le travail, notamment dans l'initialisation des valeurs telles que les poissons, les pingouins et les joueurs.

Simon a pris le rôle d'initialisateur et de donneur de tâches. Il s'est assuré que chaque membre de l'équipe savait exactement ce qu'il devait faire, en coordonnant les efforts de chacun pour que tout soit bien en place. Simon a également contribué à la conception du système de points, rendant le jeu plus équilibré et plus compétitif.

Le programme principal, qui gère le déroulement du jeu et l'affichage, présente une difficulté plus grande que les autres fonctions du code. Nous avons dû faire face à des défis complexes, notamment l'implémentation de la logique de mouvement des pingouins sur une grille hexagonale et la gestion des interactions entre les joueurs. Malgré cela, nous avons réussi à créer une interface utilisateur plutôt intuitive avec un gameplay fluide, bien que l'affichage soit sur le terminal.

Idriss, qui ne continue pas dans cette filière, a tout de même participé en essayant de trouver les erreurs du code en nous les rapportant avec humour. Il a également aidé à tester différentes configurations de jeu, nous assurant ainsi que le code final soit robuste et sans erreurs.

Explication du code:

Pour la réalisation du projet informatique de fin d'année, nous avons choisi le projet CY-FISH. Ce jeu consiste à créer un environnement où des pingouins s'affrontent sur un iceberg pour attraper le plus de poissons possibles. Le jeu se déroule sur une grille hexagonale, et chaque joueur contrôle plusieurs pingouins, les déplaçant de case en case pour collecter des poissons. Plusieurs variantes sont proposées, offrant ainsi au joueur une plus grande liberté de jeu.

L'objectif du projet était de développer une interface utilisateur simple et intuitive permettant aux joueurs d'interagir facilement avec le jeu. Le code doit être fonctionnel, robuste et le plus clair possible, assurant le bon déroulement du jeu sans bug.

Nous avons commencé par la création de l'environnement du jeu, à savoir la mise en place de la grille hexagonale. Nous avons choisi la variante HEXA_A. Voici une explication détaillée des différentes fonctions et de leur rôle dans le déroulement du jeu, l'explication suit le déroulement de la fonction main.

La fonction main commence par demander le nombre de joueurs, compris entre 2 et 6, puis initialise les joueurs à l'aide de la fonction `initialiser_joueur`. Cette fonction demande le nom de chaque joueur et initialise son score à 0.

Ensuite, nous demandons le nombre de lignes et de colonnes souhaitées, avec un minimum de 9 lignes et 9 colonnes, tout en effectuant une vérification. Une fois les dimensions saisies, nous allouons dynamiquement de la mémoire pour le tableau des poissons.

Le joueur a alors le choix entre plusieurs variantes de jeu :

Variante `FISH_NORMAL` : Les poissons valent tous 1 point.

Variante `FISH_ROTTON` : Les poissons valent tous 1 point, mais parfois des poissons avariés, valant -1 point, se cachent parmi eux.

Variante `FISH_GOLDEN` : Chaque poisson vaut 1, 2 ou 3 points.

Les poissons et les pingouins sont initialisés grâce aux fonctions `initialiser_pingouin` et `initialiser_poissons`. Le nombre de pingouins est attribué en fonction du nombre de joueurs. Chaque pingouin possède un numéro pour faciliter le déroulement du jeu. Une deuxième fonction permet de choisir la place des pingouins automatiquement ou manuellement grâce à `initialisation_Pingouin`.

Ensuite, nous affichons la grille hexagonale. Nous avons séparé le terrain entre les cases paires et impaires pour simplifier la gestion des cases hexagonales alternant les décalages. Nous avons donc deux fonctions : `afficher_support_de_jeu_pair` et `afficher_support_de_jeu_impair`. Ces fonctions assurent la mise en place du plateau de jeu et permettent la création des poissons et des pingouins. Elles accèdent aux structures grâce à l'inclusion de "player.h".

Après cela, nous avons configuré le déroulement du jeu. Grâce à la fonction `dérouler_jeu`, tant que les pingouins peuvent bouger, le jeu continue. Chaque tour, le joueur choisit une direction. Il peut seulement bouger si le déplacement est valide ; `bool joueur_peut_bouger` renvoie true, c'est-à-dire, si personne n'est sur la case et si la case n'a pas disparu. Si le mouvement est valide, le pingouin se déplace sur la case, on compte le nombre de poissons, puis on ajoute les points grâce à la fonction `ajouter_points`. Lorsque plus aucun mouvement n'est possible, le code se termine et affiche le gagnant de la partie.

Nous avons également ajouté un mode automatique, où l'ordinateur joue seul, ce qui nous a permis de faciliter les tests, car c'était plus rapide. Pour simplifier la lecture et la compréhension du code, nous avons tout classé dans des sous fichiers. ce qui nous a permis de travailler sur les chemins d'accès en C et de complexifier le projet .

Difficultés rencontrées.

Première difficulté :

Problème : Nous avons eu du mal pour la gestion de l'affichage du bon nombre de poissons dans les cases impaires sur la grille hexagonale. A chaque fois qu' un pingouin allait sur une case, le score augmentait soit de 4 soit de 3 mais jamais du bon nombre de poissons sur la case.

Solution : Nous avons réussi à surmonter ce problème en comprenant le fonctionnement des boucles, ce qui nous a permis de nous repérer dans les grilles hexagonales du plateau.

Deuxième difficulté :

Problème : Nous ne savions pas quel pingouin était le numéro 1, 2 etc... L'affichage du numéro du pingouin et l' indication du joueur auquel il appartient, à été difficile.

Solution : Nous avons résolu ce problème en ajoutant des boucles for dans le fichier d'affichage, nous permettant de connaître à chaque fois le numéro du pingouin et le numéro du joueur associé.

Problème non résolu : Nous n'avons pas réussi à faire déplacer le pingouin de plusieurs cases dans une direction donnée. Actuellement, il ne peut se déplacer que d'une case à la fois.

Finalement ce projet nous a permis d'améliorer nos compétences en c, pour la plupart c'était le premier projet informatique, cela à été très enrichissant, nous avons découvert, le travail en groupe sur un code informatique. Nous pensons avoir plutôt bien respecté les attentes du projet malgré quelques difficultés.

HAMELIN Simon
MARMELAT Paul
BENDAOUED Idriss