

CARD YARD

Introduction :

Dans le cadre de notre projet informatique de Préing1, nous avons choisi de développer un jeu vidéo nommé *Car Yard*. L'objectif principal du projet était de créer un jeu fonctionnel en utilisant le langage C et de nous familiariser avec des outils tels que VSCode et GitHub pour la gestion du code et la collaboration. Nous avons opté pour ce projet car il nous semblait à la fois stimulant et réalisable. Les obstacles rencontrés nous ont permis de progresser et de développer notre capacité à compter sur les autres membres de l'équipe, et ainsi, à savoir travailler en groupe.

Dans cette présentation, nous commencerons par détailler notre jeu et notre dynamique d'équipe, puis aborderons la répartition du travail, les choix majeurs effectués, les difficultés et les aspects qui n'ont pas fonctionné, ainsi que les bonus ajoutés à la conception.

Description

Jeu :

Le *Card Yard* est un jeu de cartes qui combine stratégie, réflexion et un peu de chance aussi. Le principe est simple : avoir le moins de points possible à la fin de la partie. Pour cela, il faut éviter les cartes avec de fortes valeurs et acquérir celles qui en ont une faible. Nous disposons donc d'une pioche centrale et chaque joueur a son deck devant lui, face cachée. Quand la partie commence, les joueurs ont deux choix : prendre la carte de la pioche ou prendre la carte de la défausse pour l'échanger avec une de ses cartes, soit prendre la carte de la pioche et choisir entre l'échanger avec une de ses cartes qui ira à la défausse, ou placer la carte de la pioche sur la défausse mais devra obligatoirement retourner une de ses cartes.

Une fois toutes les cartes retournées, le jeu se finit et on compte les points.

Notre jeu permet à l'utilisateur de choisir la plage de valeur pour les cartes entre $[-5, 15]$, et le nombre de cartes par joueurs est aléatoire pendant une partie.

On peut y jouer jusqu'à 9 joueurs, avec la possibilité d'ajouter des IA si on n'a pas d'amis, et même faire des combats d'IA.

Équipe :

Le choix de notre projet a énormément impacté notre dynamique d'équipe. Effectivement, quand on est impliqué dans le sujet, tout de suite l'entente du groupe se fait très bien. Nous nous sommes vraiment amusés en tant que groupe, à faire des sessions de programmation tard dans la nuit et perdre des

neurones au fil des heures, ou encore à ressentir la satisfaction de groupe quand, à trois, on réussit à faire le premier affichage. Nous pouvons donc affirmer avec certitude que le groupe nous a poussés vers le haut.

Répartition du travail

Malgré des niveaux de travail relativement différents, tout le monde a mis la main à la pâte. Nous nous fixions des objectifs chaque semaine et, par semaine, nous nous retrouvions au moins deux fois : une fois en cours d'informatique et une autre pour mettre tout en commun. Cela nous a permis d'avancer efficacement, sans laisser personne derrière, mais aussi d'expliquer aux autres la structure de nos codes respectifs et d'apporter de la cohérence entre nos parties.

Au fur et à mesure de l'avancée du projet, nous avons intensifié notre collaboration en présentiel pour résoudre les problèmes en temps réel et affiner le code ensemble, surtout pour le débogage et la solidification de notre code.

Date	Camille	Marc-Olivier	Sixtine
06/04/2025	Création des structures Card et Player (card.c, player.c).	Simulation d'un tour de jeu (game.c)	Code de sauvegarde(backup.c)
14/04/2025	Création des fichiers.h	construction player.c	construction du menu.c
Vacances	Création de l'AI (ia.c)	Fonction des cartes (card.c)	Affichage des cartes (display.c)
05/05/2025	Fonctionnement de la pile (discard.c)	Fonction du player.c	Affichage des decks (display.c)
07/05/2025	Correction et fin des fonctions pour un tour de jeu (game.c)	Affichage du classement des joueurs (player.c)	Débogage de l'ia et création du makefile
10/05/2025	Débogage et fortification du code	Débogage	Harmonization de l'affichage, couleurs, débogage

Ce tableau ne représente pas à la perfection notre planning, mais il illustre les grands axes. De plus, une grande partie du travail s'est croisée, et donc il n'était pas surprenant d'aller sur les tâches des autres pour apporter un coup de pouce.

Les choix importants / difficultés

Notre première inquiétude était de ne pas être sur la même architecture de code. C'est donc pour cela que nous avons directement posé les cartes sur la table, en détaillant les

structures qu'il nous fallait, leur type, les éléments qu'il fallait gérer et les fonctions vitales au bon fonctionnement de notre code.

Ensuite, nous avons décidé de manier énormément de pointeurs et de doubles pointeurs, ce qui nous a provoqué de nombreux problèmes d'allocation de mémoire une fois le code fini. Nous l'avons rapidement réglé grâce à des vérifications récurrentes permettant de ne pas faire crasher le jeu.

Les choses pas réussies

Notre plus grand regret est d'avoir mis le système de sauvegarde de côté. Effectivement, nous avons produit un code pour cette pratique, mais la contrainte de temps ne nous a pas permis de l'implémenter dans notre code. Avec du recul, nous avons peut-être les yeux plus gros que le ventre avec le projet de l'intelligence artificielle...

C'est une fonctionnalité que nous aurions bien aimé avoir finie, mais nous pensons que les autres fonctionnalités en valent la peine.

On avait aussi eu l'idée d'implanter le système des colonnes : quand toutes les cartes d'une même colonne ont la même valeur, nous pouvons supprimer la colonne entière.

Malheureusement, l'ajout de la variante *card-rand* ne nous a pas simplifié la tâche.

Effectivement, c'est compliqué de gérer les colonnes avec un nombre incertain de cartes.

Finalement, nous avons à peine pu commencer la conception de cartes spéciales, qui à notre avis auraient rendu le jeu plus intéressant, avec la possibilité par exemple d'échanger les jeux des joueurs entre eux. Le but était bien sûr de mettre le chaos. Nous n'avons donc pas développé notre idée pour éviter de briser des amitiés.

Les bonus

Pour rendre le jeu plus intéressant, nous avons ajouté une fonctionnalité bonus : la possibilité de jouer contre une ou plusieurs IA. L'intelligence artificielle que nous avons développée est assez simple, mais elle permet de jouer seul, ce qui apporte une dimension supplémentaire au jeu.

De plus, nous avons ajouté des couleurs dans l'interface pour rendre le jeu plus dynamique et attrayant visuellement. Mais en toute honnêteté, c'était aussi pour notre satisfaction personnelle, de voir un peu de beauté dans le sombre espace du terminal.

Conclusion

En conclusion, le jeu *Card Yard* nous a permis d'utiliser toutes les ressources apprises en cours d'informatique depuis le début de l'année. Travailler en groupe a réellement été un avantage pour notre groupe et nous a permis d'apprendre plus de choses plus rapidement. Je pense que grâce à ce projet, les pointeurs et les structures ne sont plus un mystère pour nous. En résumé, ce projet nous a permis de mettre en lumière nos apprentissages et aussi nos capacités d'innover. Nous sommes très contents de notre projet et nous espérons que vous le serez aussi !! :)