

RAPPORT BLOCUS 2024-2025



Réalisé par
Ibrahima Bah
Malo Reboux

SOMMAIRE

Introduction	(Page 1)
Brainstorming	(Page 2)
Difficultés	(Page 3)
Fonctionnalités.....	(Page 4)
Fonctionnalités 2.....	(Page 5)
Fonctionnalités 3.....	(Page 6)
Découpage de Fichiers.....	(Page 7)
Découpage de Fichiers 2	(Page 8)
Explication de l'IA	(Page 9)
Conclusion	(Page 10)

INTRODUCTION

Dans le cadre de notre parcours en BUT Informatique à l'IUT de Fontainebleau, nous avons pour projet de développer le jeu Blocus. Ce jeu de stratégie à deux joueurs, à la fois simple à comprendre et riche en possibilités tactiques, est un excellent défi pour mettre en pratique nos compétences en programmation et en conception de jeux. Blocus, dont les origines remontent à 2003, se joue sur une grille carrée dont la taille peut varier de 3 à 9, offrant ainsi un terrain de jeu modulable.

Le principe du jeu est relativement simple : chaque joueur déplace à tour de rôle son pion sur la grille, puis bloque l'adversaire en posant une croix sur une case vide. Le but étant de déplacer son pion tout en entravant les mouvements de l'autre joueur, jusqu'à ce que l'un des deux ne puisse plus bouger. Derrière cette simplicité apparente se cache une grande profondeur stratégique, où l'anticipation des mouvements de l'adversaire et la gestion de l'espace jouent un rôle clé. En créant Blocus, nous visons à explorer les principes du design de jeux tout en développant des compétences techniques dans la création de mécaniques de jeu dynamiques et interactives

APPEL AUX BLOCUS

~~LE 10 NOVEMBRE~~
LE 25 NOVEMBRE

BRE
MOBILISE

LA FORCE

DANS TOUS LES LYCÉES

BRAINSTORMING

Au cours de notre période de brainstorming pour le jeu Blocus, nous avons pris soin de structurer notre travail en plusieurs étapes essentielles pour garantir un développement fluide et cohérent de notre projet. Nous avons d'abord élaboré un programme détaillé, comprenant toutes les tâches nécessaires à l'achèvement du jeu. Cela nous a permis de mieux organiser notre emploi du temps, en définissant des missions claires à accomplir chaque semaine, tout en réservant le week-end à la finalisation des tâches en cours et à la planification des nouvelles activités pour la semaine suivante.

Pour assurer une gestion optimale et un suivi constant de notre projet, nous avons organisé nos réunions à différents moments et dans divers lieux, aussi bien en présentiel dans les salles informatiques que virtuellement sur notre plateforme Discord. Cette flexibilité nous a permis de maintenir une communication fluide et d'ajuster notre travail en temps réel. Ces réunions étaient cruciales pour discuter de l'évolution du jeu, échanger des idées créatives et innovantes que nous avions envisagées, et réexaminer les fonctionnalités qui avaient été temporairement mises de côté. Ces idées ont ensuite été intégrées dans un fichier commun partagé entre tous, servant de zone de réflexion où nous pouvions regrouper nos propositions sous forme de brouillon.

Nos échanges ne se limitaient pas aux seuls jours de la semaine ; le week-end a également joué un rôle fondamental dans notre collaboration. Ces sessions étaient dédiées à des discussions plus approfondies sur la progression du projet. Elles ont permis de traiter des points de vue variés, de confronter nos opinions, et même de résoudre certains désaccords qui auraient pu émerger. Cependant, notre engagement commun envers l'objectif final a toujours été maintenu, ce qui a assuré la satisfaction générale de toute l'équipe.

Cette approche collaborative, combinée à une gestion rigoureuse et à des échanges réguliers, a non seulement renforcé notre cohésion en tant qu'équipe, mais a également enrichi la qualité et la diversité des idées intégrées dans le jeu. Cela a permis de créer une dynamique de travail productive et harmonieuse, essentielle à la réussite de notre projet Blocus.



DIFFICULTÉS

Après avoir exposé nos objectifs lors du brainstorming initial du projet, nous avons rapidement été confrontés à des difficultés. Bien que notre première session de brainstorming ait permis de définir des objectifs clairs, il était encore difficile de passer de manière concrète à la réalisation d'un véritable projet. Afin de surmonter ces obstacles, nous avons organisé une réunion pour examiner de manière approfondie nos forces et faiblesses spécifiques par rapport au projet.

Lors de cette réunion, nous avons eu l'occasion d'analyser en détail les défis rencontrés, qui étaient en grande partie cohérents avec les difficultés identifiées lors du brainstorming initial. Malo Reboux a pris en charge l'aspect design et la charte graphique du jeu, ainsi que la gestion des fichiers, la recherche d'images, le développement de certaines fonctionnalités comme la gestion des différents boutons ainsi que l'écriture du rapport.

Ibrahima Bah quant à lui a assumé la gestion du dépôt GIT, la responsabilité du système de déplacements des joueurs mais aussi de la gestion de la grille ainsi que la gestion de l'IA. Notre collaboration a été extrêmement bénéfique, nous permettant de nous conseiller mutuellement et d'apporter des solutions lorsque l'un d'entre nous faisait face à des difficultés.

Cette approche collaborative a renforcé notre compréhension mutuelle des domaines de compétence de chacun et a facilité le partage d'idées et de compétences. Les échanges réguliers ont également contribué à améliorer notre efficacité, en nous permettant de résoudre rapidement les problèmes et d'avancer de manière plus fluide dans le développement du projet. Grâce à cette collaboration étroite, nous sommes parvenus à surmonter les défis initiaux et à progresser de manière significative dans la réalisation de notre projet commun.

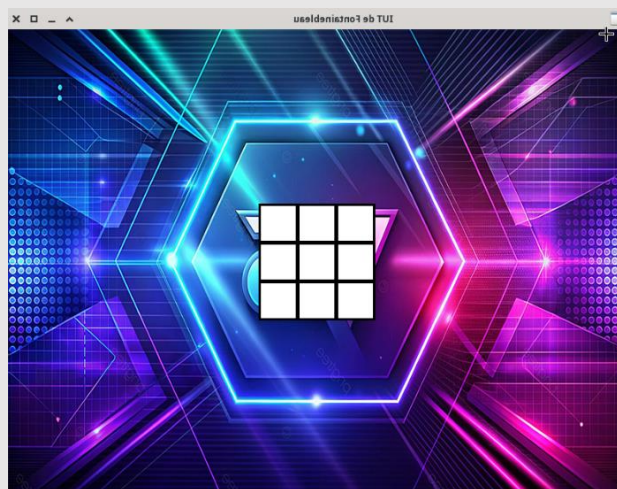
FONCTIONNALITÉS

Image du menu :



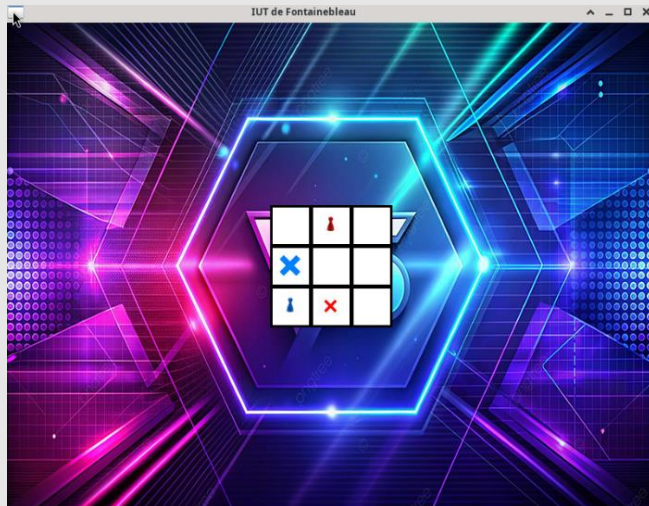
Lorsque le programme est lancé, la fenêtre du menu s'ouvre nous permettant de choisir la taille du plateau et le nombre de joueurs. Il est impossible de lancer la partie si le nombre de joueur et la taille du plateau ne sont pas sélectionnés.

Image de début de partie :



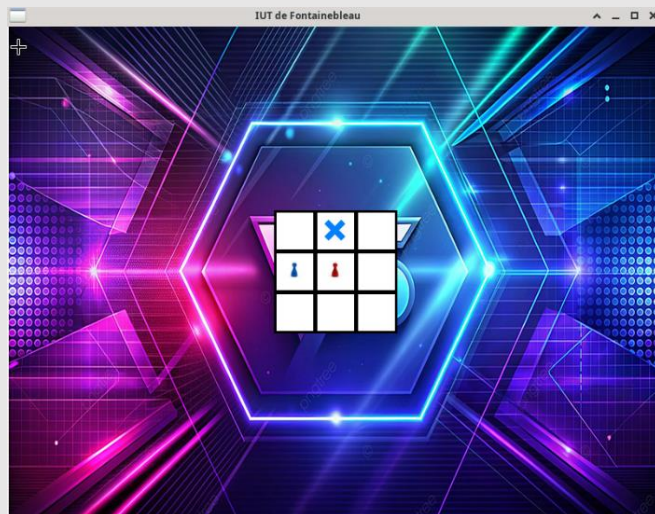
Lors du début de la partie, le plateau peu importe le mode de jeu est vide et sa taille varie de 3x3 à 9x9 en fonction de la taille choisie sur le menu. Les joueurs devront placer leurs pions puis leur croix

Image de jeu à 1 joueur :



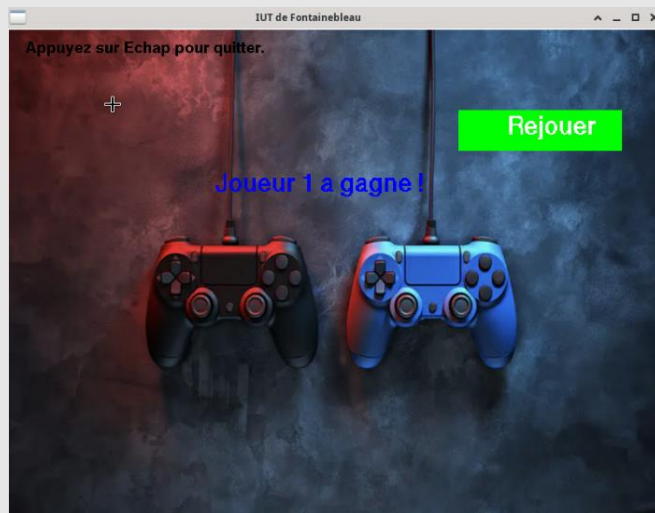
Lorsque le mode 1 joueur est sélectionné, le joueur commencera par placer son pion puis sa croix ensuite une IA posera son pion et sa croix instantanément.

Image de jeu à 2 joueurs :



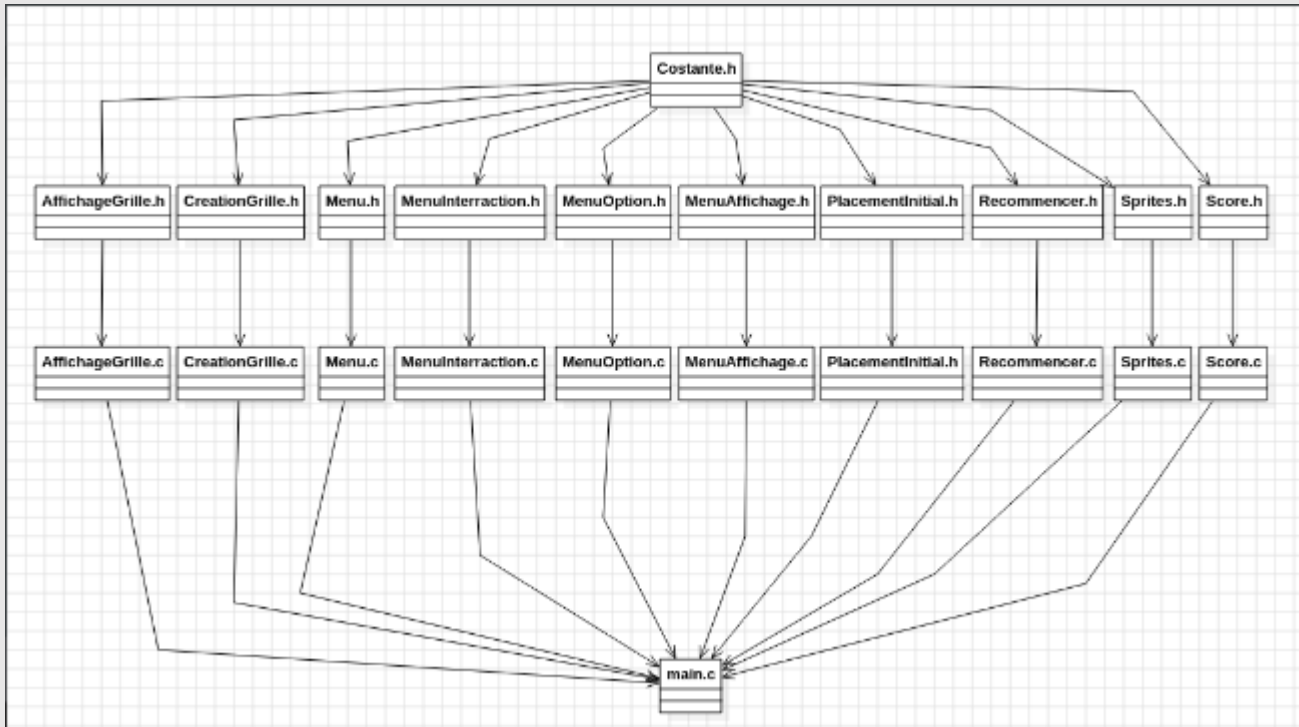
Lorsque le mode 2 joueurs est sélectionné, le joueur 1 commencera par placer son pion puis sa croix ensuite le joueur 2 placera lui aussi également puis une croix.

Image de fin de partie :



Lorsque l'un des deux joueurs à gagné la partie, il est possible de cliquer sur la touche « Echap » afin de fermer le jeu, il est également possible de cliquer sur le bouton « Rejouer » qui redirige alors sur le menu pour recommencer une partie.*

DECOUPAGE DE FICHIERS



Le découpage en différents fichiers source dans notre projet "Blocus" a été une décision délibérée visant à améliorer la lisibilité, la maintenance et la modularité du code. Chaque fichier source est dédié à une tâche spécifique, contribuant ainsi à une organisation claire du projet. Voici les justifications pour chaque fichier :

AffichageGrille.c :

Le fichier `AffichageGrille.c` est dédié à la gestion et l'affichage visuel de la grille du jeu. Il comprend les fonctions nécessaires pour dessiner la grille à l'écran avec des couleurs associées à chaque joueur ou à l'état de la cellule.

CreationGrille.c :

Le fichier `CreationGrille.c` est responsable de la gestion et de l'interaction avec la grille de jeu, incluant son initialisation, les déplacements des pions, la mise à jour de l'état des cases, et la vérification des conditions de fin de partie.

Menu.c :

Le fichier `Menu.c` centralise la gestion des interactions liées au menu principal du jeu, notamment l'affichage des options, la validation des paramètres sélectionnés par le joueur, et la transition vers l'écran de jeu.

MenuAffichage.c :

Le fichier MenuAffichage.c gère l'affichage des éléments visuels du menu principal du jeu. Il inclut le chargement et l'affichage des images de fond, la gestion de l'interface utilisateur pour la sélection de la taille de la grille, ainsi que l'interaction avec les options du menu.

MenuInteraction.c :

Le fichier MenuInteractions.c gère les interactions du joueur avec l'interface graphique du jeu, y compris la gestion des clics sur le menu et la grille de jeu, le déplacement des pions, ainsi que la logique de gestion des étapes du jeu.

MenuOption.c :

Le fichier MenuOption.c gère les choix de configuration des joueurs dans le jeu, notamment le mode de jeu (1 joueur ou 2 joueurs). Il inclut également la logique pour gérer l'IA dans le jeu, notamment les déplacements de l'IA et le choix de la meilleure case pour ses actions.

PlacementInitial.c :

Le fichier PlacementInitial.c gère le processus d'initialisation du placement des pions sur la grille au début du jeu. Selon le mode de jeu (1 joueur ou 2 joueurs), le placement des pions peut être effectué manuellement par le joueur humain ou automatiquement pour l'IA.

Recommence.c :

Le fichier Recommencer.c permet de réinitialiser l'état du jeu et de revenir au menu principal, avec toutes les variables du jeu remises à zéro, prêtes pour une nouvelle partie. Cette fonctionnalité est essentielle pour offrir une option permettant de recommencer une nouvelle partie sans quitter l'application ou le programme.

Score.c :

Le fichier Score.c contient une fonction clé qui gère l'affichage de l'écran de fin de jeu, le score et les options de redémarrage ou de quitter. Il interagit également avec l'interface graphique pour permettre à l'utilisateur de choisir de rejouer ou de quitter l'application.

Sprites.c :

Le fichier Sprites.c contient des fonctions essentielles pour la gestion des sprites (images) utilisées dans votre jeu. Plus précisément, il charge et libère les images des pions et des croix de différentes couleurs utilisées pendant le jeu. Ces images sont ensuite affichées à l'écran pour représenter les éléments du jeu de manière visuelle.

EXPLICATION DE L'IA

Pour notre jeu du blocus, il y a la possibilité de jouer seul. Pour se faire il faut quand même un adversaire, c'est là que l'IA entre en jeu, celle-ci est défini par la fonction :

`void mouvementIA(int *x, int *y, int taille) ;`

`int *x` : Ce paramètre est un pointeur vers l'entier x, qui représente la coordonnée horizontale actuelle de l'IA sur la grille

`int *y` : Ce paramètre est un pointeur vers l'entier y, qui représente la coordonnée verticale actuelle de l'IA sur la grille

`int taille` : Ce paramètre indique la taille de la grille de jeu. Par exemple, si la grille est de taille 9x9, alors la taille vaudra 9. La taille est utilisée pour vérifier que les coordonnées calculées ne dépassent pas les limites de la grille lors des déplacements. Pour que x et y restent compris dans la grille, leur valeur devra être comprise entre 0 et taille-1.

CONCLUSION

Ce projet a été pour nous une expérience riche en enseignements et en défis, marquant une étape importante dans nos cursus en programmation. Face à ce projet nous avons ressenti de l'enthousiasme comme de l'agacement face à certains problèmes, mais cela nous a permis de surmonter certaines difficultés dont nous n'aurions peut-être jamais eu connaissance. Au cours de ce projet nous avons rencontré plusieurs difficultés, comme, au départ le fait d'utiliser les images comme fond pour notre fenêtre, cela nous a posé bien des problèmes et des erreurs lors de l'exécution de notre programme. Également la gestion des déplacements des joueurs a aussi été un problème, en effet lors de nos premiers essais les pions et les croix ne se déplaçaient pas comme souhaité, ce n'est que par la suite de nous sommes arrivés à corriger ce problème. En fin de projet nous avons également beaucoup de problèmes avec l'IA, au départ elle ne posait pas de croix, elle pouvait également se déplacer sur n'importe quelle case peu importe sa position. Lorsque le problème de la croix a été réglé, elle pouvait maintenant la poser sur des cases déjà prises par d'autres croix et pour finir sur le problème majeur de notre IA, elle ne pouvait se déplacer que vers le bas du plateau, ce qui la rendait extrêmement prévisible. Pour conclure sur ce projet, tous les défauts que nous avons trouvés ont été corrigés, il nous a également permis d'apprendre à travailler en équipe, d'améliorer nos compétences en communication et à approfondir nos connaissances en développement.