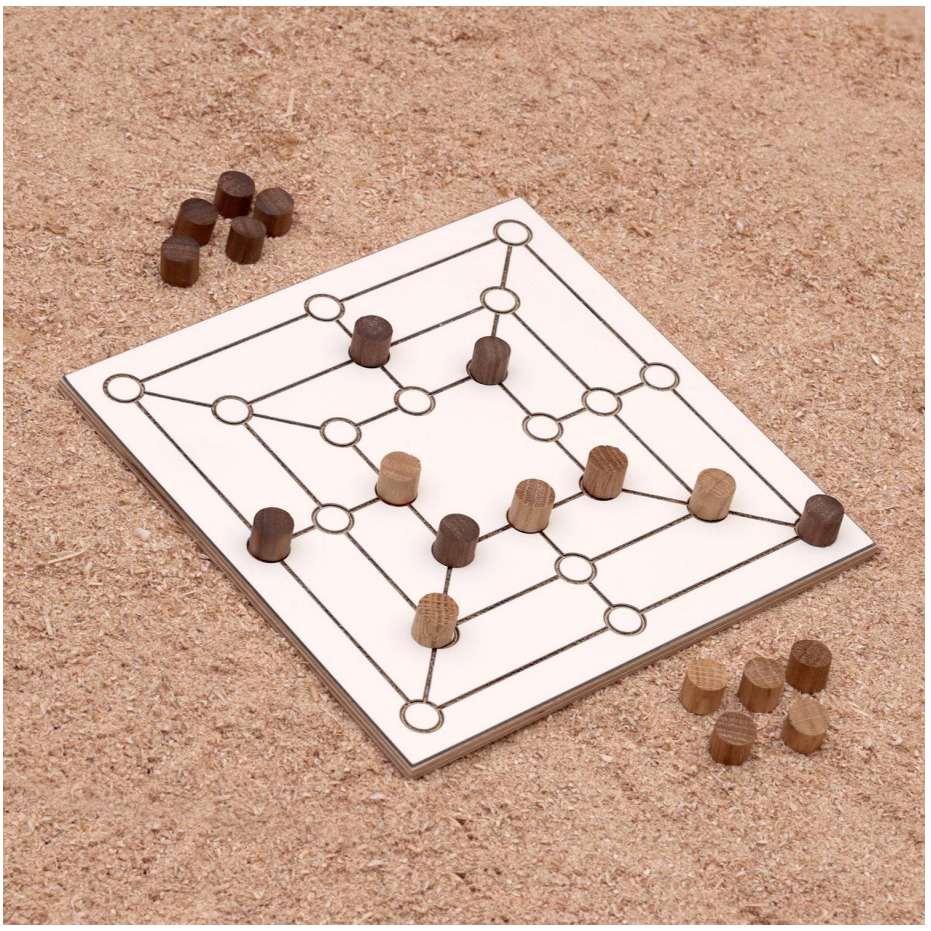
IF2B Automne 2021

Projet Informatique IF2B A2021 – Jeu du Morabaraba



HUBLART Lucas - FESSER Mathias - WITZ Robin

Responsable d'UV : LOMBARD Alexandre

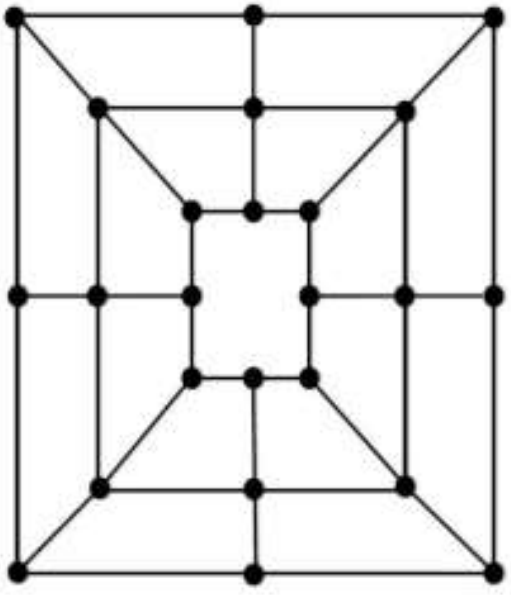
Sommaire :

Le projet sera intégralement réalisé en C par groupes de 2 ou 3 étudiants du même groupe.
Les livrables attendus sont :
- Le code source (fichiers .c et .h), dûment commenté et documenté, accompagné des instructions de compilation (fichier CMakeLists.txt ou fichier Makefile)
- D'un rapport d'une dizaine de pages présentant le travail réalisé, notamment la structure générale de votre code, les choix réalisés pour le développement du jeu, et le résultat final (en faisant le bilan de ce qui a été réussi et des points d'amélioration éventuels)

Introduction :

Le Morabaraba est un jeu de stratégie traditionnel à deux joueurs joué en Afrique du Sud et au Botswana avec une variante légèrement différente jouée au Lesotho . Le jeu est connu sous de nombreux noms dans de nombreuses langues, notamment mlabalaba , mmela (en setswana), muravava et umlabalaba . Le jeu est similaire à douze hommes morris, une variante du jeu de société romain neuf hommes.
Nous avons choisi le Morabaraba car le jeu paraissait plus intéressant d'après nous.

Règles du jeu:

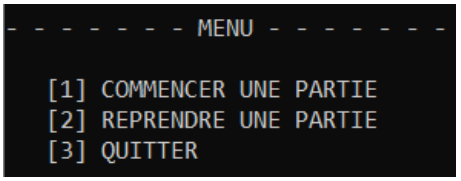


Pour commencer, les 2 joueurs vont placer leurs 12 pions sur le plateau de jeu. Lorsqu'un joueur aligne 3 pions, on dit qu'il a créé un "moulin" ce qui lui permet d'enlever un pion adverse. Si tous les joueurs ont placé tous leurs pions, ils vont maintenant déplacer leurs pions pour essayer de créer des "moulins" et éliminer les pions adverses. Si aucun pion n'est enlevé du plateau lorsqu'un des deux joueurs a moins de 3 vaches alors il y a égalité, sinon un joueur gagne quand son adversaire n'a plus que 2 vaches. Lorsqu'un joueur n'a plus que trois

vaches, il peut les déplacer n'importe où sur le plateau, on dit qu'il fait "voler" les vaches. Également, les vaches formant un moulin ne peuvent être tuées par l'adversaire, augmentant les possibilités de stratégies de jeu.

Menu:

Dans un premier temps, lorsque qu'on démarre le jeu, le menu apparaît. Il est configuré de manière à expliquer les règles du jeu et les instructions pour y jouer.
Ensuite le joueur est invité à commencer une partie, reprendre une partie ou quitter le jeu.



Si le joueur choisit de commencer une partie, il devra saisir les noms de deux joueurs et ensuite une fenêtre graphique sera créée pour qu'il puisse jouer.
Si la fenêtre ne s'affiche pas ou ne se crée pas, un message d'erreur est envoyé sur la console.

Dans le cas où le joueur souhaite reprendre une partie précédemment commencée. Il devra quitter une partie déjà commencée et celle-ci sera sauvegardée. Pour que celle-ci soit sauvegardée, on clique sur le bouton pause, on vérifie si le clic de la souris est bien sur le bouton pause, on crée un fichier texte où l'on détruit les valeurs auparavant contenues si elles existaient, et on sauvegarde les valeurs utiles pour reprendre le jeu là où il s'était arrêté. La partie est bien sauvegardée lorsque dans la console le message "la partie est sauvegardée" est affiché. Si le joueur a fait une action comme déplacer un pion il devra d'abord finir cette action pour pouvoir sauvegarder. Pour finir il devra sélectionner dans le menu "reprendre une partie" pour récupérer la partie dans laquelle il était en train de jouer.
Si le joueur décide de quitter le jeu dans le menu, alors le programme se termine.

Commandes du joueur :

Pour pouvoir jouer, les joueurs disposent de plusieurs possibilités pour placer et déplacer leurs pions et certaines particularités du jeu s'appliquent pour progresser dans celui-ci. Nous avons créé des fonctions pour pouvoir réaliser ces mouvements et les

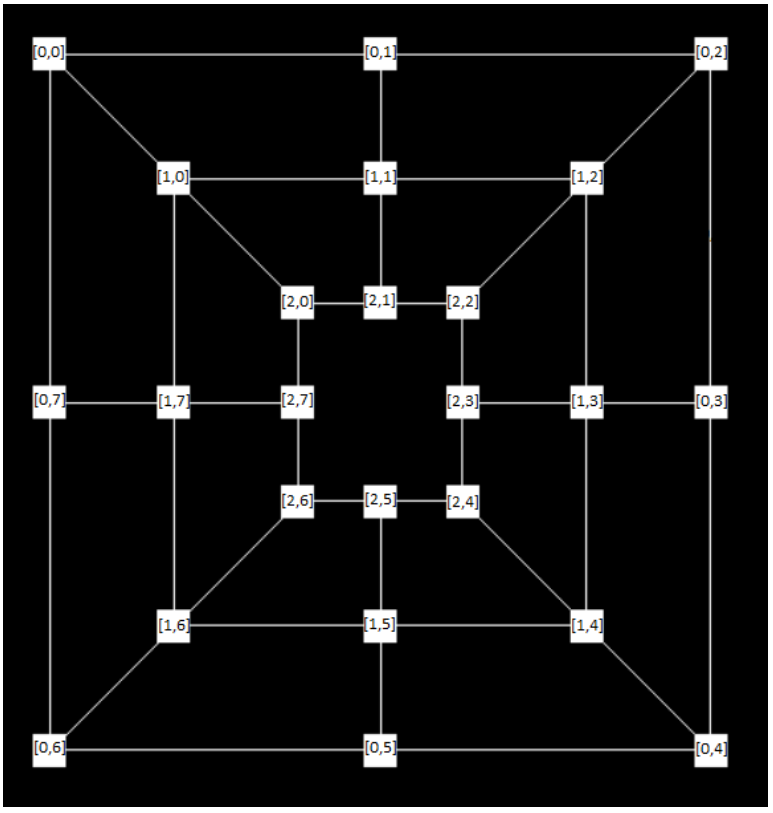
particularités du jeu, toujours de manière à ce qu'elles fonctionnent pour chaque joueur à tour de rôle.

Pour pouvoir placer les pions des joueurs, on vérifie :

- s'ils peuvent poser leurs pions dans la case sélectionnée
- qu'elle ne soit pas déjà occupée par un autre pion
- s'il reste des pions à placer ou non

Dans le cas où tous les pions ont été placés, les joueurs vont devoir déplacer leurs pions pour gagner la partie. Ils peuvent les déplacer grâce à un tableau où l'on place toutes les cases l'une à côté de l'autre, on regarde si ces cases sont déjà occupées par un pion adverse ou le sien. Si le pion est déplacé, la case où le pion était situé est redessinnée en blanc pour signifier que la case est à nouveau libre, ce qui correpond dans notre programme à "déplacement=1".

Le jeu du Morabaraba comporte des règles spécifiques comme la règle du moulin, qui correspond à trois pions alignés. Une fois cela effectué, le joueur peut tuer une vache (pion) adverse. Cette règle a été retranscrite par des fonctions. Dans l'un des fonctions on vérifie la présence d'un moulin. Un moulin peut être créé verticalement, horizontalement et diagonalement, du moment que les pions sont reliés par un des axes du plateau. La fonction vérifie la présence d'un moulin aux sommets des rectangles créés ou au arêtes des rectangles.



Ensuite nous identifions si la vache ciblée par l'adversaire qui possède un moulin n'est pas dans un moulin, ce qui est la seule condition pour pouvoir la tuer. On fait donc appel à la fonction moulin, si la vache sélectionnée est en dehors d'un moulin on lui renvoie 1, si toutes les vaches sont dans un moulin on lui renvoie 0.

Nous avons ensuite créé une fonction pour éliminer la vache

sélectionnée par le joueur qui possède un moulin, L'emplacement de cette vache est donc remis à 0 et on retrace la case en blanc, puis après cette action, on change de joueur.

Interface graphique :

Pour pouvoir observer le plateau de jeu, l'avancement de la partie, et interagir directement avec celui-ci, nous avons choisi d'utiliser la bibliothèque SDL.
Pour commencer, nous avons tracé le plateau de jeu à l'aide de rectangles, et les cases ont été positionnées sur les sommets des rectangles et sur les arêtes des rectangles. Les coordonnées des cases ont été placées dans un tableau, on exclut le bouton central de notre tableau, puis on définit les cases comme des carrés avec x, y les coordonnées et w et h les largeurs et hauteurs. Les cases sont placées dans l'ordre pour que l'on place la case de coordonnées(0,0) en haut à gauche et que l'on tourne dans le sens des aiguilles d'une montre. Pour tracer les trois rectangles, on utilise une boucle dans laquelle on trace les 3 rectangles du plateau. Ensuite nous traçons les cases grâce au tableau prévu à cet usage, puis on trace les traits qui relient les rectangles entre eux. On trace ensuite le bouton pause à l'aide d'un rectangle pour le fond, et deux autres pour dessiner le logo pause, cela permettant d'éviter de devoir insérer une image.

Une fois notre plateau de jeu terminé nous avons créé la fonction "clic souris" pour pouvoir interagir avec le plateau et donc pouvoir jouer. Celle-ci, comme son nom l'indique, permet de reconnaître l'action d'un clic gauche de la souris.
Pour commencer cette fonction, on crée une variable qui servira à arrêter la boucle si le nombre de vaches posées maximum est atteint. On initialise ensuite un tableau à 2 dimensions qui nous servira à ne pas réformer le même moulin lorsque l'on vient de le détruire. On initialise les événements et les systèmes d'événements pour pouvoir commencer à jouer.
Pour pouvoir déplacer la souris, on définit les positions de la souris avec les axes x et y.
On initialise une structure qui contiendra les coordonnées des cases où le joueur pourra déplacer ses pions.
Pour pouvoir interagir avec le plateau, nous avons mis en place une fonction qui attend un clic de souris sur un des boutons pour pouvoir continuer la boucle. Si le bouton de la souris est relâché à un certain endroit du plateau, on récupère les coordonnées de la souris pour

pouvoir identifier la case sélectionnée, on vérifie si le clic souris est bien placé sur les coordonnées d'une case sinon on renvoie un message d'erreur.
Si le joueur veut mettre en pause le jeu, il devra donc cliquer sur le bouton correspondant à cet effet. Une fois qu'il aura cliqué dessus, un fichier sera créé où l'on détruit les valeurs qu'il contenait, puis on sauvegarde les valeurs essentielles à la reprise du jeu, qui sont les positions de tous les pions, si des moulins sont formés et quel est le prochain joueur à devoir jouer.
Dans le cas où le clic n'est effectué sur aucune case existante alors le programme renvoie un message d'erreur.

Choix réalisés pour le développement du jeu :

Dans un premier temps nous avons installé la SDL 2.0 pour pouvoir avoir une interface graphique et ainsi interagir avec celle-ci. Après cela nous avons commencé à tracer le plateau en initialisant notre fenêtre, puis créé les trois rectangles qui définissent le plateau, et ensuite nous avons tracé les cases et les lignes diagonales, perpendiculaires et horizontales. Nous avons ensuite créé la fonction pour reconnaître les clics de souris, pour laquelle nous avons initialisé la fonction "SDL_event" qui nous est utile pour récupérer les coordonnées du clic de la souris. Après avoir fait cela nous avons remarqué que nous devions regarder la position du clic de la souris et regarder si cet endroit est une case ou non. Nous avons donc modifié notre fonction pour tracer les cases, nous avons entré dans un tableau à 2 dimensions ("Tab_Carre[i][j]" avec i vaut 3 pour les 3 rectangles et j vaut 8 pour les 8 case pour chaque rectangle) les coordonnés de chacune de nos cases et ainsi ensuite nous avons fait appel au tableau pour tracer nos case, ainsi nous avons juste à regarder si la position du clic de la souris appartient à notre tableau des cases.
Nous avons ensuite créé une fonction qui permet de placer les vaches sur le plateau chacun son tour en affectant une couleur à chacun de nos joueurs. A chaque fois que l'on pose une vache sur une case on regarde si elle n'appartient à personne, si c'est le cas alors on trace la case en fonction de la couleur du joueur et on remplit la case dans le tableau du joueur. Après avoir créé cette fonction qui nous permet de placer nos 2 pions sur le plateau, nous avons commencé à créer notre fonction moulin qui nous permet de regarder si un joueur a 3 pions alignés. On commence par regarder dans le tableau du joueur si le joueur possède la diagonale et colonne (horizontale et verticale) c'est-à-dire s'il possède le pions avec le même

j. Nous regardons ensuite si le pion qui vient d'être posée est sur une case paire ou impaire, si il est sur une case paire on regarde si j+1 et j+2 sont occupés par le joueur ou si j-1 et j-2 sont occupés par le joueur (tout en faisant attentions au cas particulier du 0 et 6 où l'on regarde respectivement 1,2 et 6,7 et 7,0 et 4,5). Si le pion est sur une case impaire on regarde donc j-1 et j+1 en faisant attention au cas particulier 7. Si nous avons donc 3 pions alignés nous retournons 1. Si la fonction moulin retourne 1, alors on fait appel à la fonction tuer vache que nous devons créer. Pour celle-ci nous regardons si le pion que le joueur veut enlever appartient au joueur adverse et s'il est dans un moulin ou non. S'il n'est pas dans un moulin, on enlève le pion adverse de son tableau et on redessine en blanc la case.

Nous avons ensuite créé une fonction détection qui permet de tuer une vache adversaire si elle dans moulin uniquement si le joueur ne possède que des vaches dans un moulin. Dans la fonction, on regarde pour tous les pions du joueur s'ils sont dans un moulin ou non. Si toutes ses vaches sont dans un moulin on retourne 0 et donc dans la fonction tuer vache si le joueur clique sur une case adverse et que la fonction détection nous renvoie 0 alors le joueur peut tuer n'importe quelle vache de l'adversaire.

Après avoir terminé cette partie du programme nous avons commencé à créer celle qui permet de déplacer ses pions. On a donc créé la fonction déplacement vaches qui regarde si le pion que le joueur vient de sélectionner peut être déplacé ou non sur les cases adjacentes. Pour cela nous enregistrons dans un tableau les cases possibles où le joueur peut se déplacer (3 en général, sauf pour le rectangle "1"). Ensuite nous regardons dans dans le tableau des joueurs si ces cases sont prises ou non, si au moins une case est vide alors on retourne 1 et on remet à 0 cette case dans le tableau du joueur et on redessine en blanc la case, si le joueur à moins de 3 pions alors on lui dit juste qu'il peut se déplacer et on redessine sa case en blanc et son tableau à 0. Ensuite pour reposer notre pions nous regardons si la case que le joueur vient de sélectionner est une case du tableau des cases possibles et si elle n'est pas occupée, si toutes les conditions sont réunies, alors on fait appel à la fonction créée au début pour poser les vaches. Après avoir fini cela nous avons créé la fonction pour empêcher le joueur qui vient de casser un moulin de ne pas réformer le même au tour suivant, en adéquation avec les règles du jeu. Pour cela, dans la fonction déplacement vaches, nous avons enregistré les coordonnées du pion s'il était dans un moulin, et au moment de poser les pions on regarde si la case où le joueur veut poser son pion est celle-ci ou non. Nous avons ensuite ajouté la règle des 10 tours sans tuer un pion lorsqu'un des deux joueurs à moins de 3 pions, on compte donc les tours de boucle lorsqu'il faut déplacer les vaches. Si le nombre de tours arrive à 10 ou si

un des deux joueurs n'a plus que deux vaches, alors on quitte la boucle et on donne la position de chaque joueur (classement), ou égalité. Nous avons ensuite simplifié certaines fonctions comme "Tab_j1" qui contenait les coordonnées des vaches du joueur 1, nous avons donc fait de ce tableau à 2 dimensions un tableau a 3 dimensions avec comme indice : [joueur][i][j] , avec i et j les coordonnés. Cela aurait également été faisable en utilisant une structure, mais pour des raisons d'optimisation ainsi que de coordination entre les membres du groupe, nous avons préféré utiliser un tel tableau.

Pour finir nous avons créé le bouton pause qui comme son nom l'indique met en pause la partie. Nous avons donc été dans la fonction clic de la souris et nous avons ajouté la condition suivante : si un joueur clique sur le bouton pause, alors on ouvre un fichier qui enregistre quel joueur doit jouer ensuite, le nombre de pion par joueur, le nombre de pion qu'il reste a placer, si il y a eu un moulin au tour précédent et au tour actuel. On met aussi le tableau "Tab_joueur" qui contient les coordonnées des pions des joueurs. Et donc lorsque nous lançons le programme et que nous choisissons de reprendre la partie, le programme ouvre notre fichier et on reprend les données que nous avons mis dedans lorsque nous avons mis en pause la partie, ensuite on retrace les cases en fonction du "Tab_joueurs". Ensuite on refait appelle aux différentes fonctions de création du plateau mais avec les paramètres qui étaient dans notre fichier de sauvegarde.

Evénements réussi pour la constitution du jeu:

- Création d'un plateau de jeu sous forme graphique grâce à la SDL
- Mise en place d'un menu de présentation des règles et des options pour jouer
- Prise en compte de la réalisation d'un moulin (alignement de 3 pions) et des conséquences sur le jeu de celui-ci
- Limiter le nombre de pions à placer pour chacun des joueurs
- Prise en considération de la position du clic de la souris pour pouvoir interagir avec les cases et les pions
- Mettre en place un roulement pour pouvoir déterminer quel joueur doit intervenir sur le jeu
- Affichage de messages pour indiquer quel joueur doit intervenir sur le jeu.
- Interaction avec les joueurs pour connaître leurs noms.

- Création d’un bouton pause fonctionnel
- Création d’un fichier sauvegarde lorsque le bouton pause est actionné
- Affichage du résultat de la partie à la fin du jeu (égalité, joueur 1 gagnant, joueur 2 gagnant)
- Prise en compte des erreurs pouvant occurrer lors de l’initialisation de la fenêtre ainsi que pendant la partie

Points d'amélioration éventuels :

Il y a certains points que nous pourrions améliorer pour avoir une meilleur expérience de jeu, ceux-ci sont :

- Créer un bouton retour pour la phase de déplacement, lorsque nous sélectionnons le pion que nous voulons déplacer, actuellement on redéssine en blanc la case instantanémént s’il y a une case où le joueur peut déplacer son pion, nous pourrions faire en sorte que le joueur clique sur son pion à deplacer et ensuite clique sur un bouton retour pour déplacer un autre pion.
- Mettre en surbrillance les cases où le joueur peut se déplacer lorsqu’il a cliqué sur le pion qu’il veut déplacer. De la même manière, nous pourrions également mettre en surbrillance (avec une autre couleur que pour les déplacements) les pions formant un moulin actif.
- Écrire directement sur la fenêtre de jeu les règles, les commandes, le menu ainsi que les instructions de jeu (Exemple : “Au joueur 1 de joueur (couleur Rouge)”).
  Nous avons essayé de faire cela en utilisant la bibliothèque SDL_ttf, mais malgré plusieurs essais nous n’avons pas réussi à la faire fonctionner.

Conclusion :

La réalisation de ce projet, nous a permis d’être confrontés à plusieurs difficultés de code et nous a permis d’en apprendre plus sur le langage C.

Sitographie :

Image

LOZI DESIGNS. Morabaraba+SD+2+WEB In : *Squarespace* [en ligne]. Disponible sur : https://images.squarespace-cdn.com/content/58bd4c3b3e00be6887e4985c/1543489032098-R3T2RRYDPGPPBY0A4UQC (consulté le 2 janvier 2021)

Pages sur site internet

WIKIPEDIA. *Morabaraba* [en ligne]. Disponible sur : https://en.wikipedia.org/wiki/Morabaraba (consulté le 2 janvier 2022)