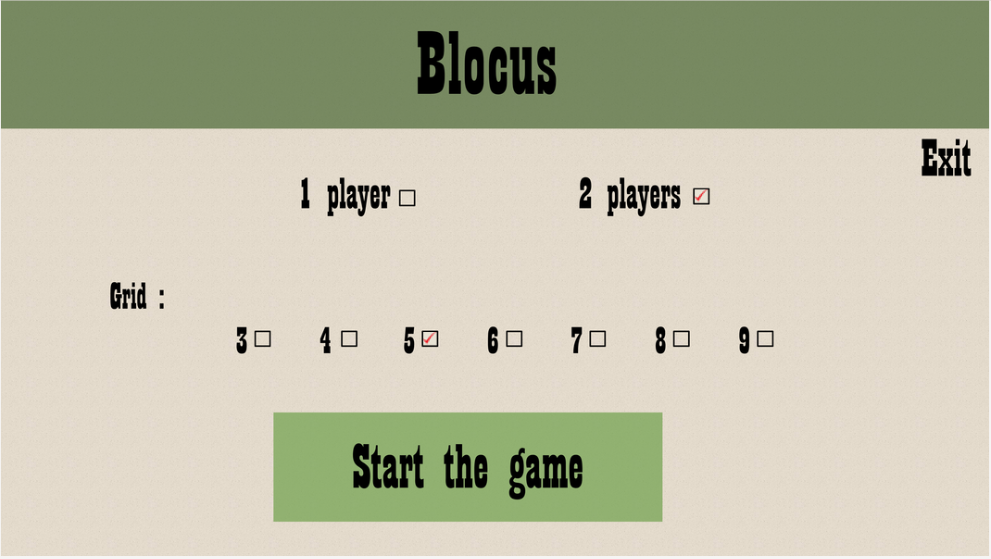
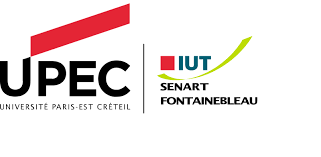
Romain SAVARY

Antoine MARCHAL DOMBRAT

Groupe 2

## **Projet Tutoré**





2020-2021

IUT de Sénart-Fontainebleau

# Sommaire

[**Projet Tutoré** 1](#_Toc61564622)

[Sommaire 2](#_Toc61564623)

[Introduction 3](#_Toc61564624)

[I les différentes fonctionnalités du programme 4](#_Toc61564625)

[II structure du programme 6](#_Toc61564626)

[III Explication des données représentant l’état d’une partie en cours 10](#_Toc61564627)

[IV Conclusions personnelles 11](#_Toc61564628)

[La conclusion personnelle de Romain ! 11](#_Toc61564629)

[La conclusion personnelle d’Antoine : 11](#_Toc61564630)

# Introduction

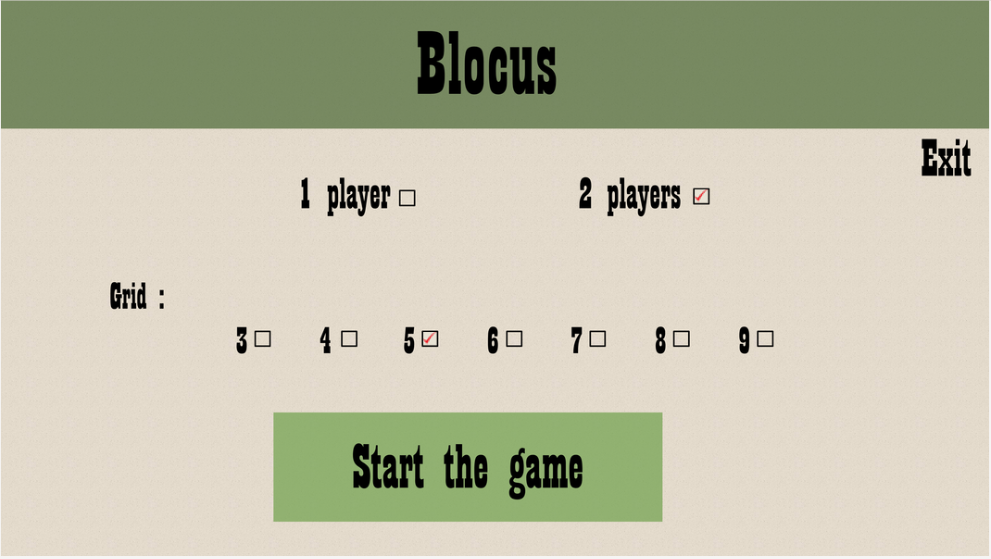
Ce document est un rapport décrivant le projet tutoré réalisé durant le premier semestre de la formation en DUT informatique. Il s’agit de la réalisation du jeu blocus en utilisant le langage C et la bibliothèque graphique de l’IUT.

Le jeu se présente en 3 écrans. Le premier est le menu où nous choisissons les options de lancement, sélection de la taille du plateau (taille allant de 3x3 à 9x9) et du nombre de joueurs (Les utilisateurs peuvent jouer l’un contre l’autre ou contre une IA). Le deuxième est l’écran de jeu, les joueurs commencent par poser leur pion sur le plateau à l’endroit qu’ils souhaitent. Puis, chacun leur tour, ils déplacent leur pion sur une case adjacente de leur choix (diagonales comprises) et condamnent une case du plateau, case où les joueurs ne pourront plus déplacer leur pion dessus. La partie se termine quand l’un des joueurs ne peut plus déplacer son pion. Enfin, il y a l'écran de fin sur lequel est affiché le joueur gagnant. Il y a aussi deux options, fermer le jeu ou retourner sur l’écran du début.

# I les différentes fonctionnalités du programme

Notre programme est composé de 3 parties : le menu de lancement du jeu, le jeu et le menu de fin de partie.

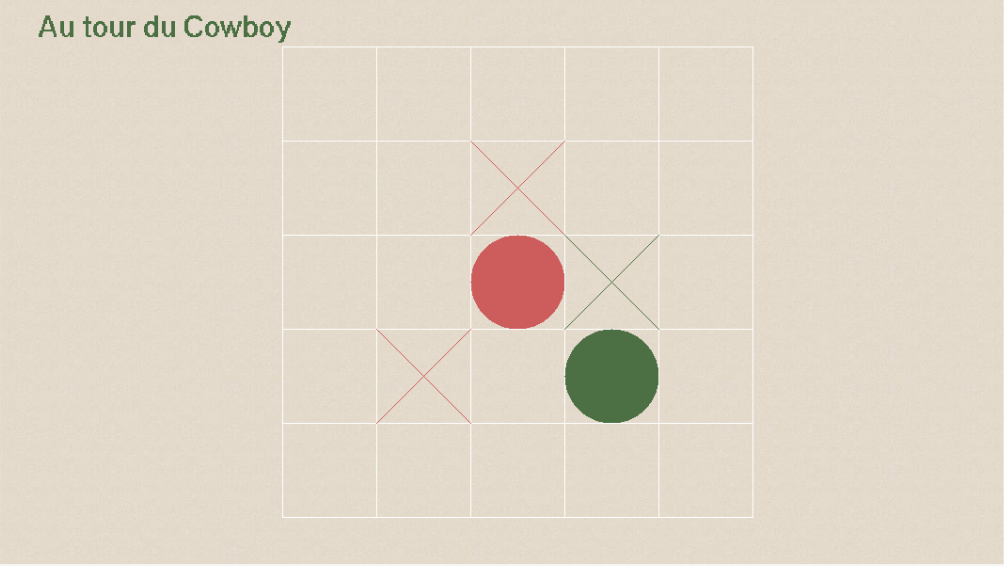
Lorsque le programme est lancé, une fenêtre s’ouvre pour afficher le premier menu.



Nous avons choisi d’utiliser une image pour rendre le menu plus esthétique et ne pas se limiter aux fonctionnalités de tracé de la bibliothèque. De plus, nous avons choisi des couleurs qui rappellent le western, qui est le thème de notre jeu. Lorsque cette interface apparaît, l’utilisateur sélectionne les paramètres de la partie : la taille de la grille et le nombre de joueurs. Par défaut, il y a 2 joueurs et la taille de la grille est de 5. Il est également possible d’arrêter le programme en cliquant sur Exit.

La partie se lance lorsque le bouton “Start the game” est sélectionné.

La grille de la taille adéquate est alors affichée :



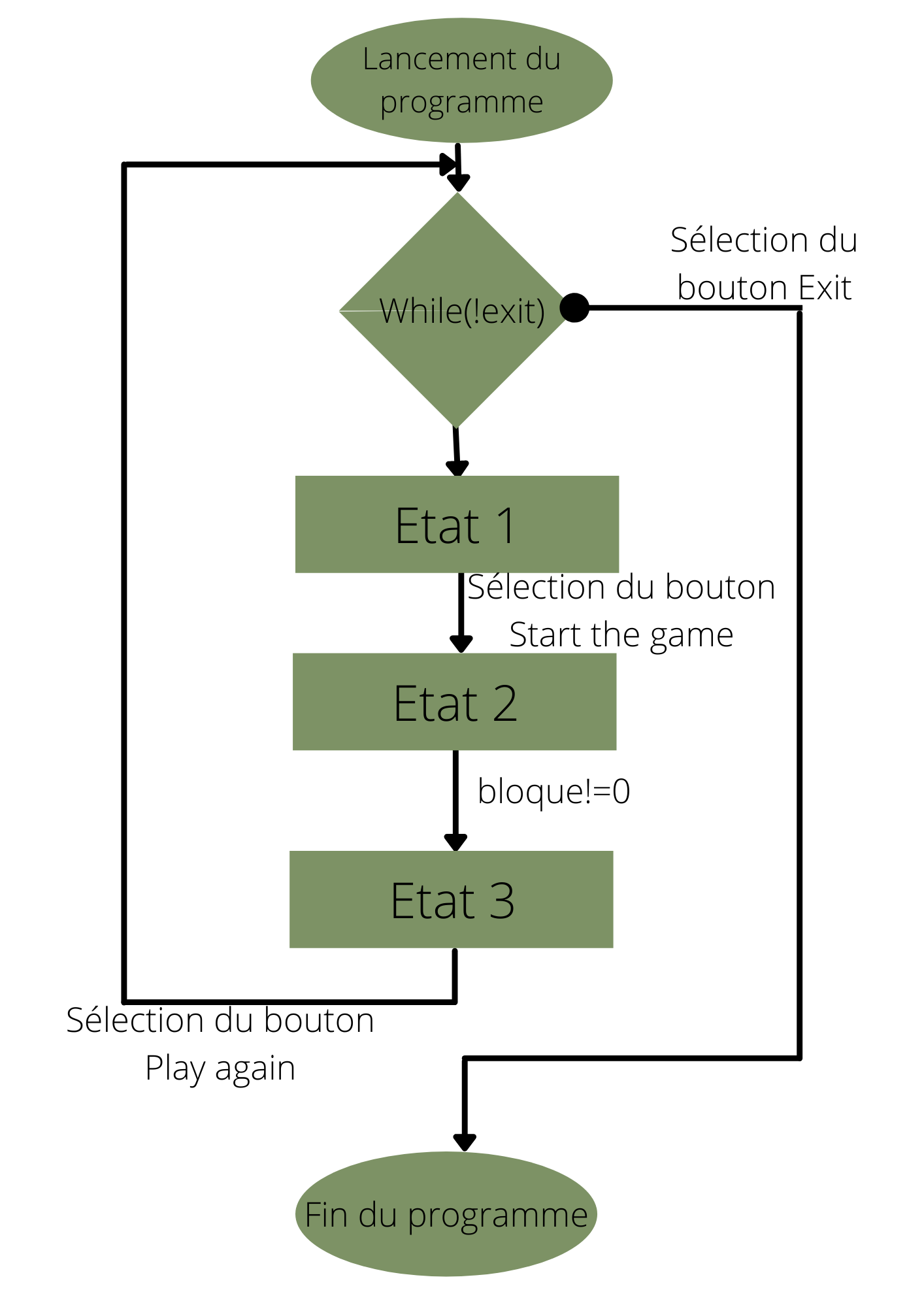
La partie débute, chaque joueur choisit où placer son pion en cliquant sur une case, puis tour à tour, il va déplacer son pion sur une case adjacente à son ancienne position, puis il va condamner une case vide. Un indicateur en haut à gauche rappelle le nom et la couleur du joueur qui doit jouer.

Lorsque le pion d’un joueur ne peut plus se déplacer, la partie est finie et le menu de fin de partie est alors affiché. Il présente le nom et la couleur du vainqueur. L’utilisateur a le choix entre relancer une autre partie en retournant au menu principal ou de quitter le programme.

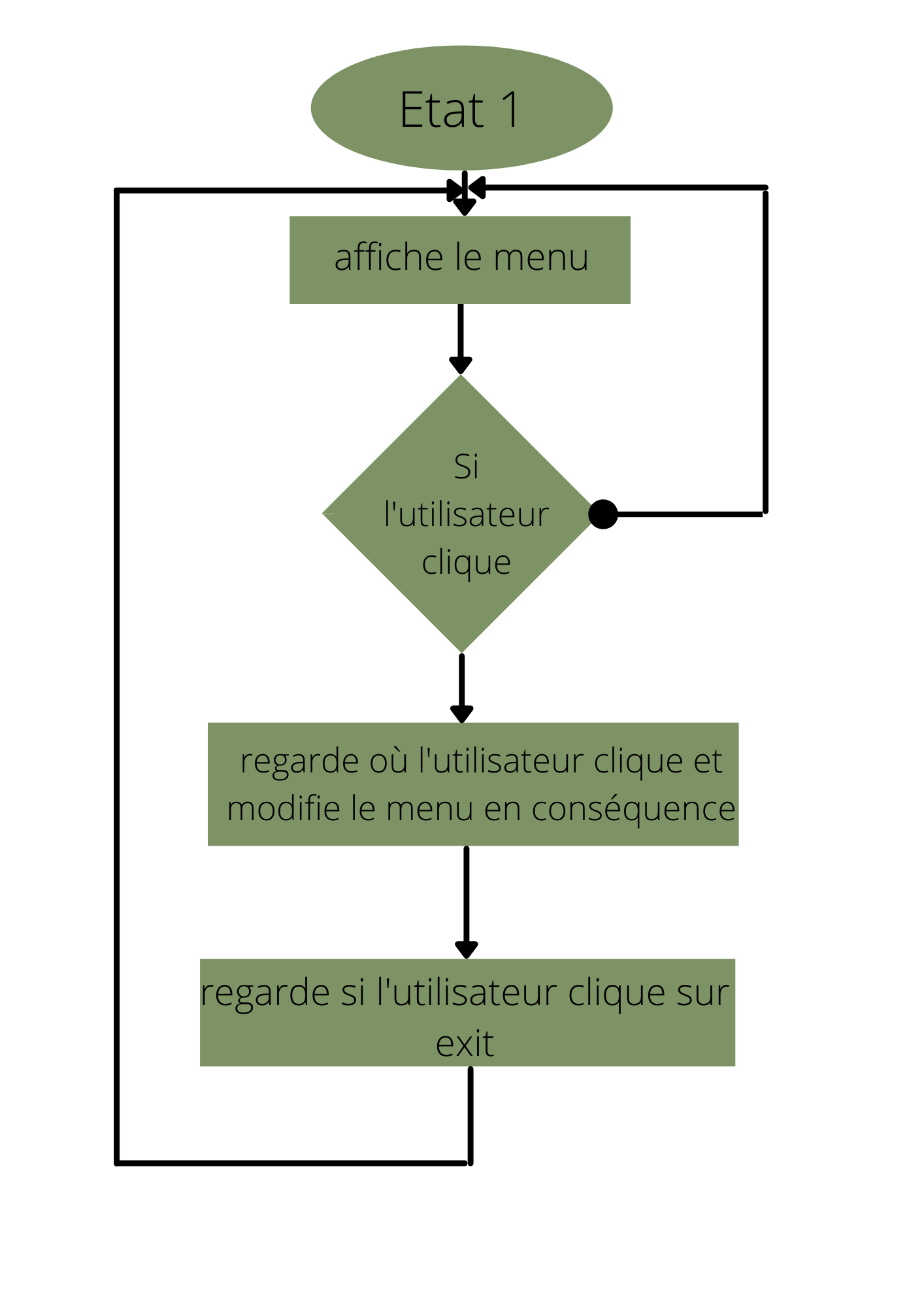


# II structure du programme

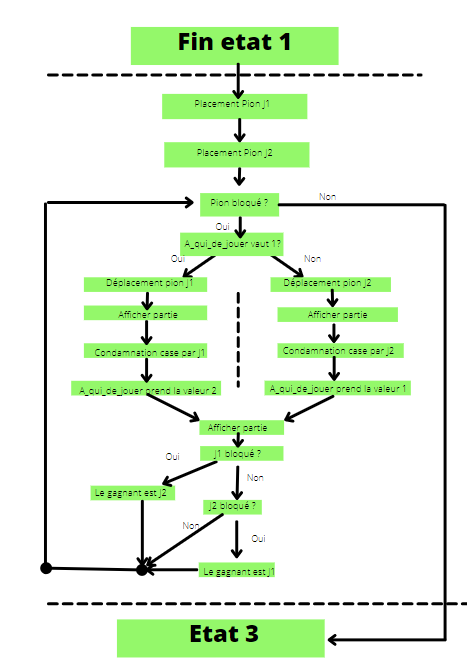
La structure principale du programme est assez simple, il suit le principe des écrans. Il y a une boucle (état) pour chaque écran, le tout dans une boucle while pour permettre de rejouer sans avoir à relancer le programme. L’état 1 est le menu principal où l’utilisateur sélectionne les paramètres. L’état 2 est le jeu et l’état 3 est le menu de fin de partie.



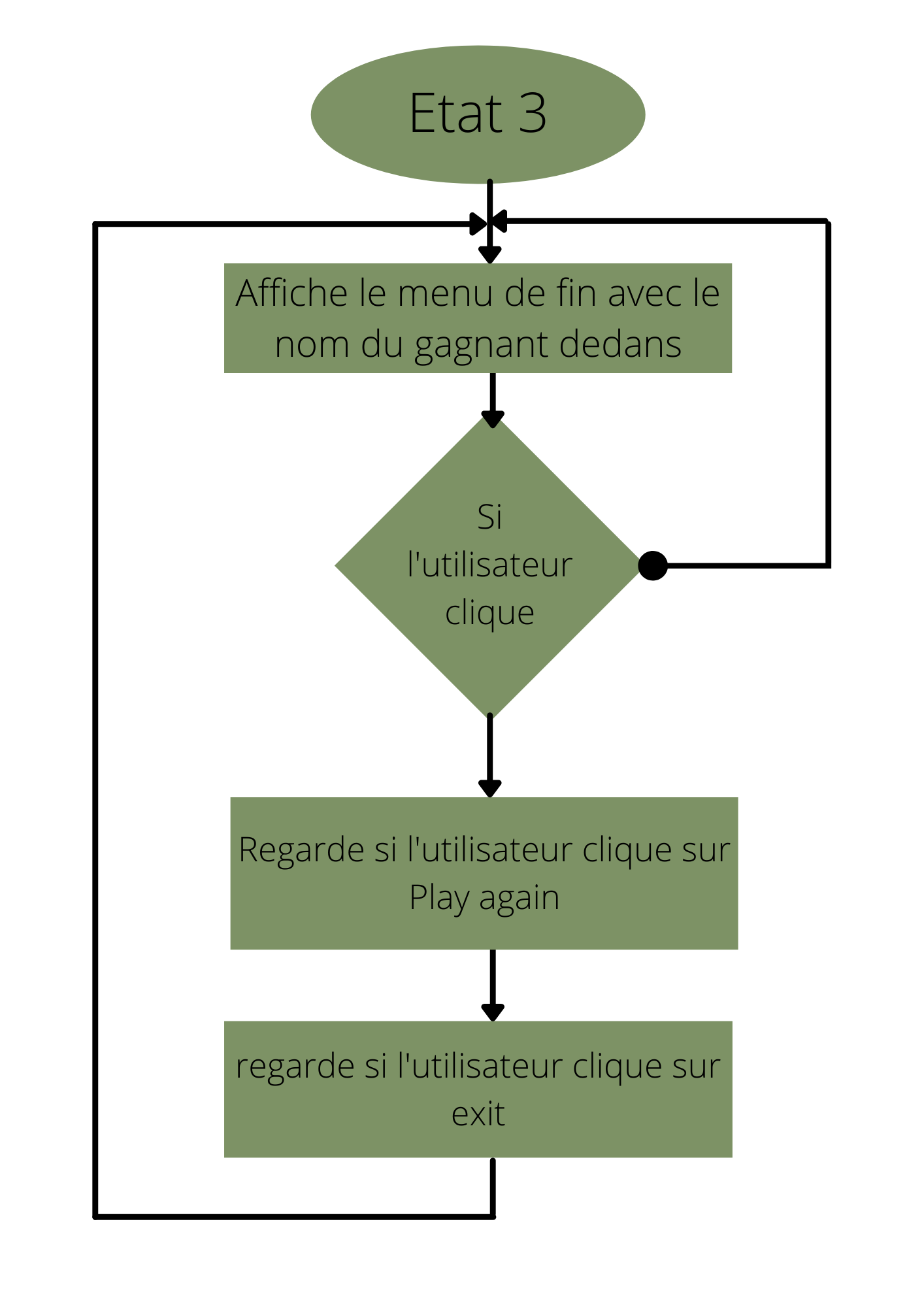
La structure de l’état 1 : Affichage du menu, et regarde si l’utilisateur a cliqué quelque part. En fonction de la réponse, modifie le menu de manière à correspondre avec ce que l’utilisateur a coché.



La structure de l’état 2 : On commence d’abord par placer le pion de chaque joueur. On se place ensuite dans une boucle où l’on ne pourra en sortir, pour passer à l’état 3, que quand l’un des pions sera bloqué. Dans cette boucle, on fait jouer à tour de rôle les joueurs (grâce à la variable a\_qui\_de\_jouer) et on vérifie si les pions ne sont pas bloqués.



La structure de l’Etat 3 : Après avoir déterminé dans l’état 2 qui est le vainqueur, on affiche le menu de fin avec le nom du gagnant. Enfin on attend que l’utilisateur clique sur un bouton, Play again ou Exit. Play again fera revenir à l’état 1 et Exit fermera le programme.



# III Explication des données représentant l’état d’une partie en cours

Lors du lancement du programme l’entier *etat* est initialisé à 1, l’entier exit à 0. Le menu du jeu est affiché. L’utilisateur peut modifier certains paramètres à l’aide de la souris. Ainsi, les valeurs des entiers *nb\_joueurs* et *taille\_grille* vont être modifiées. Si Exit est sélectionné, alors la variable *exit* prend la valeur 1 et le programme s’arrête. Ensuite, lors de la sélection du bouton “Start the game”, *etat* prend la valeur 2.

Le jeu commence alors. *virtual\_board* est une table bidimensionnelle contenant des entiers correspondant à la grille affichée à l’écran. Elle contient plusieurs informations : la position du pion du joueur 1 (représentée par la valeur 1), du joueur 2 (2), des cases condamnées par le joueur 1 (3) et par le joueur 2 (4).

Au début de la partie, ce tableau est rempli de 0. Chaque joueur va placer son pion, de ce fait les valeurs contenues dans *virtual\_board* vont changer pour s’adapter à la situation de la partie.

L’entier *a\_qui\_de\_jouer* est initialisé à 1 : c’est le premier joueur qui commence à jouer. L’entier *bloque* vaut 1 si un des pions est entouré de cases non vides, ainsi, au début de la partie il est initialisé à 0. Tant que bloque vaut 0 la partie continue : alternativement *a\_qui\_de\_jouer* va prendre les valeurs 1 et 2 pour que les joueurs jouent chacun leur tour.

Le tour d’un joueur se déroule de la manière suivante :  
Il déplace son pion sur une case à proximité grâce à la fonction *selection\_case\_pion* qui retourne à *ligne* et *colonne* les coordonnées de la case sélectionnée. Puis le joueur va condamner une case à l’aide de la fonction *selection\_case\_condamnation* qui va retourner à *ligne* et *colonne* les coordonnées de la case sélectionnée. Ces changements sont transcrits dans le tableau *virtual\_board* grâce aux variables *ligne* et *colonne.* A la fin d’un tour d’un joueur, on regarde si les pions sont bloqués. Si le pion du J1 est bloqué, alors l’entier *winner* prend la valeur 2 pour signifier que le vainqueur est le joueur 2. Réciproquement si J2 est bloqué. Lorsqu’un joueur est bloqué, *etat* prend la valeur 3. L’écran de fin de partie est alors affiché, il est également indiqué le nom du gagnant.

En cliquant, l’utilisateur peut sélectionner le bouton “Play again”, *etat* va alors prendre la valeur 1 et le menu principal va alors être affiché. Comme dans l’état 1, il peut quitter la partie.

# IV Conclusions personnelles

## La conclusion personnelle de Romain !

Avoir un jeu qui marche c’est cool.

En réalisant ce projet, j’ai trouvé que la première difficulté que l’on ait rencontrée c’était de connaître ce qu’on pouvait faire et ce qu’on ne pouvait pas faire avec la bibliothèque graphique. Mais une fois qu’on a compris comment fonctionnent les différents écrans et comment passer de l’un à l'autre, ça a été plus simple.

Je suis content qu’on ait réussi à faire que le jeu fonctionne, je peux enfin le présenter à mon petit frère et à ma petite sœur.

## La conclusion personnelle d’Antoine :

Ce projet m’a permis de grandement m'améliorer en programmation. J’ai pu programmer pour la première fois une interface graphique grâce à la bibliothèque de l’IUT. Au départ, je trouvais cela complexe, mais en passant du temps et en cherchant à comprendre les fonctions de la bibliothèque graphique, cela s’est avéré plus simple. Enfin je suis fier d’avoir réalisé ce projet car nous avons fait face à plusieurs difficultés mais nous sommes toujours parvenus à trouver des solutions.