Projet IFB2/IFE2 P2021 (Puissance N)

L'objectif de ce projet et de réaliser un jeu en C inspiré par le jeu « Puissance 4 » que nous appellerons « Puissance N ».



Dans le jeu Puissance N, les joueurs placent à tour de rôle des jetons dans une grille, l'objectif étant d'être le premier joueur à aligner N jetons de sa propre couleur.

Réalisation du projet

Le projet sera intégralement réalité en C par un **groupe de 2 ou 3 étudiants** du même groupe (A, B..K, mais pas forcément dans les mêmes demi-groupes présentiel/distanciel).

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)

Le projet devra être rendu au plus tard le **13/06 à 18h** dans l'espace de dépôt dédié sur Moodle « Dépôt projet P21 ». Tous les fichiers (rapport et sources) seront contenus dans une archive nommée **NOM-MEMBRE1_NOM- MEMBRE2_NOM- MEMBRE3.zip**

Règles du jeu

Une partie de puissance N se déroule de la façon suivante :

- L'ordinateur demande à l'utilisateur combien de jetons N devront-être alignés pour gagner, la grille de jeu sera un carré de côté N + 2
- Ensuite le joueur qui aura l'initiative (jaune ou rouge) est déterminé par tirage au sort
- En début de manche, le joueur qui a l'initiative doit placer un jeton de sa couleur dans l'une des N + 2 colonnes, le jeton « tombe » et prend place au-dessus du dernier jeton inséré dans cette colonne (en début de partie, le jeton tombe tout en bas)
- Le second joueur insère alors un de ses propres jetons dans l'une des N + 2 colonnes

- La partie se termine dès que l'un des joueurs a aligné N jetons horizontalement, verticalement, ou en diagonale

Une règle supplémentaire au jeu traditionnel de puissance 4 est ajouté :

- Lors de son tour, un joueur peut décider de ne pas poser de jetons, mais au contraire d'en retirer un ; lorsqu'un jeton vient d'être retiré, l'autre joueur ne peut reposer de jetons dans cette colonne pendant sa prochaine action

Consignes du projet

Il s'agit de réaliser un programme en C permettant à un ou deux joueurs de jouer au jeu « Puissance N ».

Au démarrage le programme affiche un menu demandant à l'utilisateur ce qu'il souhaite faire via un menu :

- Démarrer une nouvelle partie
- Charger une partie existante
- Quitter

Si l'utilisateur choisit « Quitter », le programme se ferme. Si l'utilisateur choisit « Charger », la dernière partie sauvegardée reprend.

Si l'utilisateur choisit « Nouvelle partie », l'ordinateur demande combien de joueurs joueront (1 ou 2) :

- Dans le cas du jeu à un joueur, la machine joue aléatoirement à la place du second joueur (optionnel : il sera possible d'intégrer une *IA* pour améliorer l'expérience de jeu)
- Dans le cas du jeu à deux joueurs, les joueurs sont interrogés à tour de rôle par la machine

Avant le début de la partie, l'ordinateur demande combien de jetons devront être alignés, puis la machine annonce quel joueur à l'initiative (joueur « jaune » ou joueur « rouge »).

Une manche se déroule de la façon suivante :

- On affiche la grille à l'écran
- L'ordinateur demande au joueur ayant l'initiative quelle action il souhaite réaliser : ajouter un jeton, retirer un jeton (on ne proposera pas l'option retirer si la grille ne contient pas de jetons), ou « sauvegarder et quitter »
- Si l'utilisateur choisit « Ajouter », l'ordinateur demande dans quelle colonne puis contrôle la validité de la réponse (la colonne doit exister et ne doit pas être pleine)
 - Après ajout, l'ordinateur vérifie si l'un des joueurs a gagné (N jetons alignés) : si oui, on annonce le gagnant et on interrompt la partie, si non, on passe au tour de l'autre joueur
- Si l'utilisateur choisit « Retirer », l'ordinateur demande dans quelle colonne puis contrôle la validité de la réponse (la colonne doit exister et contenir au moins un jeton)
- Si l'utilisateur choisit « Sauvegarder et quitter », on sauvegarde et on quitte le programme
- On recommence tant qu'un gagnant ne peut être désigné et tant qu'il reste de la place dans la grille si la grille est pleine, il y a match nul

Instructions supplémentaires

Le programme devra être réalisé en C et devra être programmé de façon modulaire (division du code en plusieurs modules – fichiers C et fichier H). Le code devra être dûment commenté et documenté.

Des fonctions devront être créées afin de structurer le programme.

Notamment vous devrez:

- Définir une structure « Grid » permettant de caractériser la grille, elle contiendra un tableau de caractères, une largeur et une hauteur (le caractère « _ » représente une case vide, le caractère « O » représente un jeton du joueur jaune et le caractère « X » représente un jeton du joueur rouge).
- Définir une fonction « show_grid » qui prend en paramètre la grille à afficher et la représente à l'écran
- Définir une fonction « add_token » qui prend en paramètre la grille, l'indice d'une colonne et un type de jeton (caractère O ou X) et insère ce jeton dans la bonne colonne ; cette fonction retournera 1 si l'insertion était possible et 0 sinon
- De même définir une fonction « remove_token » qui prend en paramètre la grille et l'indice d'une colonne et retire le jeton le plus haut de la colonne ; cette fonction retournera 1 si la suppression était possible et 0 sinon
- Définir la fonction « check_winner » qui prend en paramètre la grille et qui vérifie si un des joueurs a gagné, elle retourne 0 si le premier joueur a gagné, 1 si c'est le second, et -1 si aucun des joueurs n'a gagné pour l'instant

Par ailleurs:

- La fonction « save » prend en paramètre la grille et le joueur dont c'est le tour et écrit ces informations dans un fichier
- La fonction « load » prend en paramètre une grille vide, et l'emplacement d'un fichier et initialise la grille à partir des données du fichier sélectionné, puis retournera 0 ou 1 selon le joueur dont c'est le tour

Ces fonctions seront nécessaires pour réaliser le programme demandé mais non suffisantes, vous serez amenés à définir vos propres fonctions pour réaliser ce projet. Le programme principal mobilisera les différentes fonctions créées afin de permettre aux joueurs de jouer.

Remarque : Des améliorations pourront être apportées afin d'améliorer l'esthétique ou l'ergonomie du jeu, ces modifications impacteront positivement votre note.

Project IFB2/IFE2 P2021 (Connect N)

The objective of this project is to realize a game in C inspired by the game "Connect 4" that we will call "Connect N".



In Connect N, players take turns placing chips in a grid, with the objective of being the first player to line up N chips of their own colour.

Implementation of the project

The project will be carried out entirely in C in **groups of 2 or 3 students** from the same group (but not necessarily in the same face-to-face/remote groups).

The expected deliverables are:

- The source code (.c and .h files), duly commented and documented, accompanied by the compilation instructions (CMakeLists.txt file or Makefile)
- A report of about ten pages presenting the work done, including the general structure of your code, the choices made for the development of the game, and the final result (making an assessment of what has been achieved and possible points of improvement)

The project must be handed in no later than **13/06 at 6pm** in the dedicated submission space on Moodle "P21 project submission". All files (report and sources) will be contained in an archive named **NAME-BINOME1_NAME-BINOME2_NAME-BINOME3.zip**

Rules of the game

A game of Connect N proceeds as follows:

- The computer asks the user how many chips N must be lined up to win, the game grid will be a square of side N + 2
- Then the player who will have the initiative (yellow or red) is determined by drawing lots
- At the beginning of the round, the player who has the initiative must place a chip of his colour in one of the N + 2 columns, the chip "falls" and takes place above the last chip inserted in this column (at the beginning of the game, the chip falls at the bottom)

- The second player then inserts one of his own chips in one of the N + 2 columns
- The game ends as soon as one of the players has aligned N chips horizontally, vertically, or diagonally

An additional rule to the traditional Connect Four game is added:

 On his turn, a player may decide not to place any chips, but instead to remove one; once a chip has been removed, the other player may not place any chips in that column during his next action

Project guidelines

The aim is to create a program in C allowing one or two players to play the game "N Power".

At startup the program displays a menu asking the user what he wants to do via a menu:

- Start a new game
- Load an existing game
- Quit

If the user chooses "Quit", the program closes. If the user chooses "Load", the last saved game is resumed.

If the user chooses "New game", the computer asks how many players will play (1 or 2):

- In the case of the one-player game, the machine plays randomly instead of the second player (optional: it will be possible to integrate an AI to improve the game experience)
- In the case of the two-player game, the players are questioned in turn by the machine

Before the start of the game, the computer asks how many chips should be lined up, and then the machine announces which player has the initiative ("yellow" or "red" player).

A round proceeds as follows:

- The grid is displayed on the screen
- The computer asks the player with the initiative what action he/she wants to perform: add a chip, remove a chip (the remove option will not be offered if the grid does not contain any chips), or "save and exit".
- If the user chooses "Add", the computer asks in which column and then checks the validity of the answer (the column must exist and must not be full)
 - After adding, the computer checks if one of the players has won (N tokens aligned):
 if yes, the winner is announced and the game is interrupted, if no, the other player's
 turn is taken
- If the user chooses "Withdraw", the computer asks in which column and then checks the validity of the answer (the column must exist and contain at least one chip)
- If the user chooses "Save and Quit", the program is saved and quit
- The game is repeated as long as there is no winner and as long as there is space left in the grid if the grid is full, there is a draw

Additional instructions

The program must be written in C and must be programmed in a modular way (division of the code into several modules - C files and H files). The code must be properly commented and documented.

Functions must be created to structure the program.

In particular, you must

- Define a "Grid" structure to characterise the grid, it will contain an array of characters, a width and a height (the character "_" represents an empty square, the character "O" represents a token of the yellow player and the character "X" represents a token of the red player).
- Define a "show_grid" function that takes the grid to be displayed as a parameter and displays it on the screen
- Define an "add_token" function which takes as parameters the grid, the index of a column and a type of token (character O or X) and inserts this token in the right column; this function will return 1 if the insertion was possible and 0 otherwise
- Similarly, define a "remove_token" function that takes the grid and the index of a column as parameters and removes the highest token from the column; this function will return 1 if removal was possible and 0 otherwise
- Define the function "check_winner" which takes as parameter the grid and checks if one of the players has won, it returns 0 if the first player has won, 1 if it is the second, and -1 if none of the players has won yet

In addition:

- The "save" function takes as parameters the grid and the player whose turn it is and writes this information to a file
- The "load" function takes as parameters an empty grid, and the location of a file and initializes the grid from the data of the selected file, then returns 0 or 1 according to the player whose turn it is

These functions will be necessary to carry out the requested program but not sufficient, you will have to define your own functions to carry out this project. The main program will use the different functions created to allow the players to play.

Note: Improvements can be made to improve the aesthetics or the ergonomics of the game, these modifications will have a positive impact on your grade.