

TP Noté

Advanced C Programming by Aakash SONI

time : 2h00

Note:

- You must complete this test in a group of maximum 3 members.
- Working Code == No error == Good Answer
- Incomplete Code == Code with errors == Wrong Answer == 0 Point
- You must submit a short report (2-3 pages) to explain your answer.
- Bonus :
 - Comments
 - Modular code (.h, .c, .o)

Note:

- Vous devez effectuer ce test dans un groupe de 3 membres maximum.
- Code fonctionnel == Pas d'erreur == Bonne réponse
- Code incomplet == Code avec erreurs == Mauvaise réponse == 0 Point
- Bonus :
 - Commentaires
 - Code modulaire (.h, .c, .o)

Exercise (Multiplayer Game Simulation with Pthreads)

Welcome to the exciting world of concurrent programming!

Bienvenue dans le monde passionnant de la programmation concurrente !

In this exercise, you will implement a multiplayer game simulation using the C programming and pthreads. This exercise is designed to challenge your understanding of multi-threading, basic concepts in C, and coordination in a practical context.

Dans cet exercice, vous allez implémenter une simulation de jeu multijoueur en utilisant le langage de programmation C et les pthreads. Cet exercice est conçu pour mettre à l'épreuve votre compréhension du multithreading, des concepts de base en C, et de la coordination dans un contexte pratique.

Objective:

Objectif :

The primary goal of this exercise is to implement a simulated multiplayer game environment where **players navigate a 2D grid, interact with each other, and collect power-ups**. The exercise is divided into three distinct questions, each building upon the knowledge gained from the previous one.

L'objectif principal de cet exercice est de mettre en œuvre un environnement de jeu multijoueur simulé où **les joueurs naviguent dans une grille 2D, interagissent les uns avec les autres et collectent des bonus**. L'exercice est divisé en trois questions distinctes, chacune construisant sur les connaissances acquises dans la précédente.

Player Movement Simulation

Simulation du Mouvement des Joueurs

You have to simulate the movement of multiple players on a 2D grid. Each player should have a unique starting position, and their positions should be updated concurrently in separate pthreads.

Vous devez simuler le mouvement de plusieurs joueurs sur une grille 2D. Chaque joueur doit avoir une position de départ unique, et leurs positions doivent être mises à jour simultanément dans des pthreads distincts.

Health Power-ups

Bonus de Santé

Introduce health power-ups randomly scattered across the grid. The power-ups, once appeared on the grid, stay on the grid forever and can be used indefinitely by the players. This function must also run on a separate pthread.

Introduire des bonus de santé dispersés de manière aléatoire à travers la grille. Les bonus, une fois apparus sur la grille, restent sur la grille pour toujours et peuvent être utilisés indéfiniment par les joueurs. Cette fonction doit également s'exécuter dans un pthread distinct.

Player Interaction

Interaction des Joueurs

Combine the player movement and power-ups to introduce player interactions. Players that occupy the same position engage in a battle, decreasing their health. Whereas, a player that occupies the same position as a power-up, regains its health.

Combiner le mouvement des joueurs et les bonus pour introduire des interactions entre les joueurs. Les joueurs qui occupent la même position s'engagent dans un combat, diminuant leur santé. En revanche, un joueur qui occupe la même position qu'un bonus de santé regagne sa santé.

Formally, you have to implement a C program to perform the following tasks:

Formellement, vous devez implémenter un programme C pour effectuer les tâches suivantes :

- **Task 1:** Simulates the movement of players in a game using pthreads.
 - Create a thread for each player, and simulate their movement across a 2D grid.
 - Each player should have a unique starting position, and their position should be updated randomly in each 'time step'.
 - A player can move either right, left, up, down or diagonally in each step.
 - Players cannot go outside the grid.
 - If the random movement is in the direction to leave the grid, the player does not move during this time step.
 - After each movement, print the final positions of all players.

Tâche 1 : Simuler le mouvement des joueurs dans un jeu en utilisant les pthreads.

- Créer un thread pour chaque joueur et simuler leur déplacement à travers une grille 2D.
- Chaque joueur doit avoir une position de départ unique, et leur position doit être mise à jour de manière aléatoire à chaque "étape de temps".
- Un joueur peut se déplacer vers la droite, la gauche, vers le haut, vers le bas ou en diagonale à chaque étape.
- Les joueurs ne peuvent pas sortir de la grille.

- Si le mouvement aléatoire est dans la direction de quitter la grille, le joueur ne se déplace pas pendant cette étape de temps.
 - Après chaque mouvement, imprimer les positions finales de tous les joueurs.

 - **Taks 2:** Extend the program to include health power-ups.
 - Create a separate thread that generates health power-ups.
 - These power-ups appear at random positions on the grid.
 - They must appear at a regular time interval (example: each 3 seconds).
 - Players can pick up these power-ups to increase their health.
- Tâche 2 : Étendre le programme pour inclure des bonus de santé.**
- Créer un thread distinct qui génère des bonus de santé.
 - Ces bonus apparaissent à des positions aléatoires sur la grille.
 - Ils doivent apparaître à des intervalles de temps réguliers (par exemple, toutes les 3 secondes).
 - Les joueurs peuvent ramasser des bonus pour augmenter leur santé.
-
- **Task 3:** Enhance the program to introduce player interactions.
 - Create a function that checks for collisions between players and power-ups.
 - If two players occupy the same position, they engage in a brief interaction, and both the players lose a portion of their health.
 - If a player occupies the same position as a power-up, it regains a portion of its health.
 - Print the final health of each player after each interaction.
 - No need to create a separate thread for this function. This function must be called by the player thread (Task 1).
 - You can choose any condition to end the simulation (Example: one of the player health becomes 0)

- Tâche 3 : Améliorer le programme pour introduire des interactions entre les joueurs.**
- Créer une fonction qui vérifie les collisions entre les joueurs et les bonus.
 - Si deux joueurs occupent la même position, ils s'engagent dans une brève interaction, et les deux joueurs perdent une partie de leur santé.
 - Si un joueur occupe la même position qu'un bonus, il gagne une partie de sa santé.
 - Montrer la santé finale de chaque joueur après chaque interaction.
 - Pas besoin de créer un thread distinct pour cette fonction. Cette fonction doit être appelée par le thread du joueur (Tâche 1).
 - Vous pouvez choisir n'importe quelle condition pour mettre fin à la simulation (par exemple : la santé d'un des joueurs devient 0).

You are free to write your own code from scratch !

Vous êtes libre d'écrire votre propre code à partir de zéro !

Or, if you need some help to quickly start coding, you can use the code template provided on the Boostcamp.

Ou, si vous avez besoin d'aide pour commencer rapidement à coder, vous pouvez utiliser le modèle de code fourni sur le Boostcamp.

The code template includes three files:

Le modèle de code comprend trois fichiers :

- *game.h* : This file has basic definitions and structures to create the game environment.

game.h : Ce fichier contient des définitions et des structures de base pour créer l'environnement de jeu.

- *game.c* : This file has some predefined variables to create players, powerups, and threads. It also contains some function declarations to help you implement the required tasks.

game.c : Ce fichier contient quelques variables prédéfinies pour créer des joueurs, des bonus et des threads. Il contient également des déclarations de fonctions pour vous aider à implémenter les tâches requises.

- *displaygamegrid.c* : This file implements a function to display the game grid along with the players and power-ups positions.

displaygamegrid.c : Ce fichier implémente une fonction pour afficher la grille de jeu avec les positions des joueurs et des bonus.

You can compile the template using the following command :

Vous pouvez compiler le code en utilisant la commande suivante :

```
gcc game.c displaygamegrid.c -o abc
```

Obligatory : You must submit your answer in a compressed (.zip) file containing :

- **Source code** (.c, .o, .h, ...)
- **A report** that explains your solution. **Maximum 3 pages**.
 - It must contain **an algorithm or flowchart along with its explanation**.
 - It must include **an analysis of the memory usage** of your code.
 - You may use GCC, SIZE and/or manual calculation.
 - How many bytes are used on Data, Heap and Text segments ?
 - How many bytes are used on stack by each function separately ?
 - You can use extra pages (annex) to show screenshots of the compiler output (gcc) and the output of your code (executable). Show the entire screen, do not crop the images.

If you use other sources from the internet to answer this exercise, for example stackoverflow.com or AI tools, please systematically **cite them** in your code and in your report. Plagiarism will not be tolerated.

Obligatoire: Vous devez soumettre votre réponse dans un fichier compressé (.zip) contenant :

- **fichiers codes** (.c, .o, .h, ...)
- **Un rapport** expliquant votre solution. **Maximum 3 pages**.
 - Il doit contenir **un algorithme ou un organigramme avec son explication**.
 - Il doit inclure **une analyse de l'utilisation de la mémoire** de votre code.
 - Vous pouvez utiliser GCC, l'outil size ou réaliser des calculs manuellement.
 - Combien d'octets sont utilisés sur les segments : Data, Tas et Texte ?
 - Combien d'octets sont utilisés sur la pile par chaque fonction séparément ?
 - Vous pouvez utiliser des pages supplémentaires (annexes) pour montrer des captures d'écran de la sortie du compilateur (gcc) et de la sortie de votre code (exécutable). **Montrez l'écran entier, ne rognez pas les images**.

Si vous utilisez d'autres sources provenant d'Internet pour répondre à cet exercice, telles que stackoverflow.com ou ChatGPT, **veuillez les citer** systématiquement dans votre code et dans votre rapport. Le plagiat ne sera pas toléré.

Grades:

Working Code == No error == Good Answer		20%
Report	Algorithm / flowchart along with its explanation	40%
	Analysis of the memory usage	40%
Bonus	Code Comments. Modular code (.h, .c, .o)	

Simulation example:

Exemple de simulation :

Game	Observations
[1] [.] [.] [.] [.] [.] [.] [.] [.] [2] [.] [.] [0] [.] [.] [.] [+][.] [.] [.] [.] [.] [.] [.] [.] Player 0 Health: 10 Player 1 Health: 10 Player 2 Health: 10	5x5 game grid showing the position of the three players (0, 1, and 2) and the power-up position (+), as well as the health of the players. Grille de jeu 5x5 indiquant la position des trois joueurs (0, 1 et 2) et la position du power-up (+), ainsi que la santé des joueurs.
[1] [.] [.] [.] [.] [.] [.] [.] [2] [.] [.] [.] [0] [.] [.] [.] [+][.] [.] [.] [.] [.] [.] [.] [.] Player 0 Health: 10 Player 1 Health: 10 Player 2 Health: 10	Player 2 moved left. Le joueur 2 s'est déplacé à gauche.
[1] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [*] [.] [.] [.] [+][.] [.] [.] [.] [.] [.] [.] [.] Player 0 Health: 1 Player 1 Health: 10 Player 2 Health: 7 Player 2 & 0 interacted. Health: 7 & 1	Player 2 moved diagonally (down-left). Interaction between Player 2 and Player 0 (represented by *). Le joueur 2 s'est déplacé en diagonale (en bas à gauche). Interaction entre le joueur 2 et le joueur 0 (représenté par *).
[1] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [.] [2] [.] [.] [.] [0] [.] [.] [.] [.] [.] [.] [.] [.]	Player 0 moved diagonally (down-left). Interaction between Player 0 and Power-Up. Le joueur 0 s'est déplacé en diagonale (en bas à gauche). Interaction entre le joueur 0 et le Power-Up.

```
Player 0 Health: 11  
Player 1 Health: 10  
Player 2 Health: 7  
Player 0 powerUp. Health: 11
```

Happy Coding