



# Systeme Unix

## LemIPC

Contact [b-psu-330@epitech.eu](mailto:b-psu-330@epitech.eu)



# Table des matières

Détails administratifs	2
Sujet	3
Contraintes techniques	4
Fonctions interdites	5
Fonctions autorisées	6



## Détails administratifs

- Les sources doivent être rendues sur le dépôt SVN
- Rendu :  
`svn+ssh ://kscm@koala-rendus.epitech.net/lemipc-année_promo-login_x`
- Votre projet devra compiler avec un Makefile.
- Le binaire du projet sera `lemipc`.
- Ce projet est à réaliser en binôme



## Sujet

"Arts Electroniques" (c) (ci-après dénommée "AE"), petite entreprise spécialisée dans l'import/export de sirop d'érable, a décidé d'élargir son cercle d'activités dans le but d'accroître son pouvoir lucratif. Pour ce faire, ils ont choisi de se lancer sur le marché émergeant des applications vidéoludiques.

AE n'ayant pas l'intention de se la jouer petit joueur, ils veulent faire une entrée fracassante sur la scène mondiale en sortant un premier titre innovant, immersif, addictif et (surtout) lucratif.

Ils font aujourd'hui appel à vous pour réaliser ce chef d'oeuvre qui laissera une trace [in]dé[lé]bile dans l'histoire des jeux vidéo.

Votre mission est donc de développer un programme répondant aux spécificités suivantes :

- Le but du jeu est de faire combattre des joueurs (répartis sur plusieurs équipes) sur un plateau de jeu en 2 dimensions
- Pour qu'une équipe gagne la partie, il faudra qu'elle soit la seule restante sur le plateau
- Pour tuer un joueur, il faut qu'au moins 2 joueurs d'une même équipe (et différente du premier) soient à son contact. C'est à dire sur une case adjacente à la case où se situe le joueur que l'on veut tuer (y compris en diagonale)
- Lorsqu'un joueur se rend compte qu'il est entouré d'au moins 2 joueurs d'une même équipe adverse, il doit quitter le plateau de jeu (oui, car c'est bien connu que les NPC sont honnêtes dans les jeux, ils respectent scrupuleusement les règles ... si si, demandez au plombier moustachu sur son kart s'il lui arrive de tricher ... et bien non !)
- Une case du plateau ne peut accueillir plus d'un joueur simultanément



## Contraintes techniques

- Chaque client est un processus et il ne doit y avoir qu'un seul exécutable, ce qui implique que le premier joueur qui démarre crée les ressources partagées (shm, msgq, semaphores)
- De la même façon, lorsqu'un joueur quitte le jeu, il doit vérifier s'il est le dernier sur le plateau. En effet, le dernier processus a la charge de nettoyer tous les IPCs créés par le premier processus.
- Le plateau doit être stocké dans un segment de mémoire partagée (shm). Chaque joueur peut consulter le contenu du plateau comme il le souhaite, mais pour le modifier, il doit respecter les contraintes liées aux ressources partagées et aux accès concurrentiels (sémaphores).
- Un joueur ne peut communiquer avec les autres joueurs que par des MsgQ
- Sur le plateau, un joueur peut voir si une case est vide ou si elle contient un joueur. Dans ce cas, c'est le numéro de l'équipe du joueur qui est accessible. Il n'est pas possible de différencier 2 joueurs d'une même équipe.
- Vous devez faire un affichage de ce qu'il se passe sur le plateau :
  - Soit en mode texte, et dans ce cas, seul le premier joueur (celui qui crée le plateau) affiche le contenu (le processus doit donc continuer à fonctionner après la mort du joueur)
  - Soit en mode graphique, et dans ce cas soit le premier processus fait l'affichage (selon les mêmes contraintes que précédemment), ou soit vous pouvez faire un autre exécutable dédié à l'affichage



# Fonctions interdites

- aucune



# Fonctions autorisées

- la librairie C
- les appels systemes liés aux IPC