

Projet d’implémentation d’un jeu de plateau ou d’un jeu vidéo simple

1 Organisation et Planning

Le projet doit être réalisé en binôme (les redoublants ont la possibilité d’être seul, sinon ils doivent s’associer à un autre redoublant avec la possibilité d’enrichir le projet réalisé l’an dernier, les alternants en groupe de 4 avec des attentes supplémentaires que nous détaillerons ensemble). Vous devez implémenter en C un jeu (soumis à validation), puis le déposer sur claroline avant le lundi 4 janvier 12h00 (les soutenances auront lieu le lundi 4 janvier ou mardi 5 janvier). Le dépôt doit être rendu sous la forme d’une archive (tar.gz) et doit **UNIQUEMENT** contenir :

- l’ensemble du code source (fichiers sources, Makefile) ;
- une notice “README” expliquant comment créer l’exécutable ;
- un rapport rédigé en L^AT_EX de 4 pages (max) expliquant comment vous avez réalisé ce projet.

NB : n’oubliez pas de commenter votre code et de faire un rendu “propre” !!

2 Le projet

2.1 Choix du jeu à implémenter

N’oubliez pas de faire valider le choix de votre jeu par mail ou discord :

Un jeu non validé = 0 pour votre note pratique.

2.2 Le travail à réaliser

Vous devez implémenter un jeu **simple**¹ avec une interface graphique simple gérée avec la bibliothèque MLV² installée sur les machines des salles de TPs.

Les fonctionnalités qui doivent apparaître **obligatoirement** dans vos projets sont :

- Quelque soit le jeu choisi, vous vous limiterez à 4 joueurs au maximum.
ATTENTION : les jeux multi-joueurs doivent donc pouvoir se jouer sur un seul et même écran : vous pouvez donc implémenter un jeu coopératif.³
- On doit pouvoir interrompre une partie et la sauvegarder pour la reprendre ultérieurement (la possibilité de sauvegarder différentes parties est un plus).
- Les dix meilleurs scores sur l’ensemble des parties seront affichables (et donc sauvegardés également).
- Le menu devra donc comporter au minimum les options :
 - nouvelle partie ;
 - sauvegarder la partie (gestion de plusieurs “slots” de sauvegardes) ;
 - charger une partie ;
 - afficher les meilleurs scores.

Vous devez commenter votre code et adopter un style de codage clair et cohérent. N’oubliez de créer différents modules associés à une tâche précise. De même, préférez les fonctions courtes avec un rôle précis. Gérez les cas exceptionnels/d’erreurs (par exemple, les entrées ou les actions erronées de l’utilisateur ne doivent pas interrompre le jeu).

En outre, il vous est demandé d’aller plus loin... (voir ci-dessous).

1. Simple, mais pas trop quand même... Par exemple : le tic-tac-toe est trop simple.

2. Bibliothèque MLV : <http://www-igm.univ-mlv.fr/~boussica/mlv/api/French/html/index.html>

3. Un jeu multi-joueurs tel que la bataille navale ou un jeu de cartes type Uno ne seront pas validés.

3 Aller plus loin

Toute initiative personnelle et justifiée sera prise en compte et valorisée. Il existe de nombreux moyens d'étendre un jeu.

Vous pouvez, par exemple (liste non exhaustive) :

- permettre de jouer contre des joueurs ordinateurs en implémentant une “intelligence artificielle” simple dans le projet (Indication : commencer par un robot jouant de manière aléatoire) ;
- sauvegarder les données de jeux (parties, scores) via des fichiers binaires.

3.1 Le rapport

En même temps que votre projet, vous devez remettre un rapport (rédigé en \LaTeX). Celui-ci devra être court : 4 pages au plus. Vous devrez mettre en avant les fonctions principales de votre programme, expliquer les choix d'implémentation et votre démarche. Il comptera dans la notation.