

## Introduction

Ce projet a pour but de proposer la création d'un ensemble de **deux applications** écrites en **langage C++** et **en binôme** (exclusivement<sup>1</sup>), une application de conception de plateaux de jeu et une application permettant de jouer sur un ensemble de plateaux.

Le jeu est à temps discrets pilotés par l'utilisateur. Celui-ci déplace à l'aide du clavier un personnage afin d'aller récupérer un maximum de trésors tandis que des personnages hostiles automatisés tentent de l'en empêcher.

La notation jugera d'abord l'originalité des solutions en termes d'architecture logicielle et de qualité du code produit ainsi que la «jouabilité» (ergonomie de l'interaction et de la représentation). Parmi d'autres critères on trouvera les algorithmes proposés, la gestion de projet, etc.

**Attention:** aucune interface graphique évoluée n'est demandée, autrement dit des solutions de type Qt ou autres sont explicitement interdites. Une simple interface en mode texte est suffisante, on peut donc essentiellement se contenter d'afficher à chaque tour l'ensemble de plateau de jeu. Si vous en avez le courage, pour rendre les choses plus «agréables» vous pouvez utiliser la bibliothèque de gestion du mode terminal **ncurses**, sans jamais y passer trop de temps...

**Attention:** s'inspirer du travail des autres est possible (vous savez les géants, les nains sur leurs épaules, etc) **mais** le plagiat est strictement interdit. Il vous est fortement déconseillé de tenter l'expérience et en général de ne jamais tester la sagacité de vos enseignants. Le plagiat est assimilable à une tricherie et peut donc vous amener devant une commission disciplinaire dont les réquisitions peuvent être très lourdes.

## Description

Un plateau est constitué d'un rectangle de cases, chacune pouvant contenir au plus un élément parmi (au minimum) les suivants :

- un **oeurj** symbolisé<sup>2</sup> par le caractère **J** et qui peut se déplacer, à chaque tour, dans l'une des huit cases adjacentes (si possible), rester sur place ou se téléporter,
- des **streumons** symbolisés par le caractère **s** et qui peuvent se déplacer, à chaque tour, dans l'une des huit cases adjacentes (si possible). Un streumon n'a qu'un but dans la vie : dévorer le ouerj.
- des **reumus** symbolisés par un **X**. Un reumu est inamovible et aucun élément ne peut le traverser ou monter dessus. C'est un obstacle parfait. Il y a tout autour du plateau des reumus ainsi qu'au moins une teupors (voir après).
- des **diams** symbolisés par le caractère **\$** qui peuvent être ramassés par le ouerj. Celui-ci doit d'ailleurs en ramasser au moins un avant qu'une porte de sortie ne s'ouvre et qu'il puisse tenter de s'y engouffrer. Les diams sont ignorés par les streumons.
- des **geurchars** symbolisés par le caractère **\*** et qui permettent d'obtenir, pour le ouerj, une possibilité de téléportation en plus et activable à volonté. Le ouerj peut avoir des téléportations dès le départ du jeu (c'est une possibilité).
- des **teupors** symbolisées par **+** si elles sont ouvertes et **-** si fermées. À chaque fois que le ouerj ramasse un diam un nouvelle teupor s'ouvre par laquelle le ouerj peut sortir du plateau.

<sup>1</sup>**Pas d'exception!** Si vous êtes un nombre impair alors un trinôme ou un monôme sera accepté. Vous pouvez utiliser les forums de moodle afin de trouver un compagnon d'infortune, n'attendez pas janvier pour le faire. Si vous avez des difficultés, prévenez rapidement l'enseignant responsable. Les cas très particuliers comme les auditeurs libres ou non, Erasmus, etc sont priés de prendre contact avec l'enseignant si nécessaire, afin de trouver une solution le cas échéant.

<sup>2</sup>Les symboles ne sont que des propositions, vous pouvez utiliser ce que bon vous semble mais limitez-vous à un caractère seul... Attention aux couleurs (si vous utilisez **ncurses** par exemple), tous les terminaux ne les acceptent pas.

Un plateau pourrait ressembler à :

```
XXXXXXXXXXXXXXXXXXXXX Téléports: 12
X   X$           $X Diams:    0
X  *XXX   s    X Niveau:    3
X                   X
-                   X
X   J           s X
XXXXXXXX-XXXXXXXXXXXX
```

## Tâches à réaliser

---

Trois choses sont à réaliser et livrer :

1. deux programmes sous la forme d'un code source en langage C++ compilable *via* un `Makefile`<sup>3</sup> :
  - **GAME CREATOR**  
(commande `gc fichier.board | fichier.game fichier.board [fichier.board...]`) permettant de créer des plateaux (fichiers d'extension `.board`). Un plateau sera stocké sous la forme d'un simple fichier texte (on pourra donc utiliser dans un premier temps un simple éditeur de texte pour créer des plateaux.). Cet outil permettra aussi de créer un «jeu» (fichier d'extension `.game`) : un jeu est un ensemble ordonné de noms de plateaux que le oueurj doit «passer» l'un après l'autre.
  - **GAME PLAYER**  
(commande `gp fichier.game | fichier.board`) permettant de jouer à un jeu ou s'entraîner/tester un plateau.
2. un modèle UML de l'architecture «objet» des applications.
3. un manuel d'utilisation précis des applications. Ceci n'est pas un rapport, mais un document expliquant comment jouer ainsi que toutes les particularités qui intéresseraient un simple utilisateur.

Les détails de la livraison (date, mode de rendu, etc) seront précisés plus tard, mais vous avez encore le temps. Les soutenances n'auront lieu qu'en janvier (vers la mi-janvier). Ne vous laissez toutefois pas déborder par le temps!

## Extensions

---

Bien qu'*a-priori* rien ne soit vraiment interdit (à part utiliser un framework graphique comme Qt), il est très important de respecter l'esprit du sujet. Il ne faut donc pas s'en éloigner et dans le doute en demander la faisabilité au responsable de cet enseignement. Ne pas changer les règles par ce que cela vous «simplifie la vie»...

Mais on peut imaginer :

- différents types de streumons : doués de plus ou moins d'intelligence pour attraper le oueurj et repérés par divers symboles,
- différentes interactions entre les streumons qui se marchent dessus : reproduction, élimination, création d'artefacts divers (mur, diams, etc),
- différents types de téléportations donc de geurchars : aléatoire, destination sûre (garantissant au moins un coup sûr au oueurj), destination choisie, etc
- il n'est rien précisé sur l'existence d'un score ou non, vous pouvez inventer la notion correspondante afin, par exemple, d'établir un classement en conservant les meilleurs scores, etc.
- on peut imaginer pouvoir sauvegarder une partie afin de pouvoir la reprendre plus tard.

---

<sup>3</sup>Respecter impérativement cette consigne! Vous pouvez utiliser n'importe quel outil pour développer mais la livraison doit contenir de quoi générer le projet avec la commande `make`. Une bonne façon de procéder est de simuler la livraison auprès de camarades lesquels devraient pouvoir générer les applications par simple appel à `make` (éventuellement en modifiant le nom ou l'emplacement du compilateur et des bibliothèques). Et si cela fonctionne, de livrer le tout comme il sera demandé ensuite...