

# Critères de notation pour le projet Hex

## Cahier des charges

Le cahier des charges doit être précisé de façon concise (sans répéter inutilement les règles du jeu).

En quelques lignes sur une feuille, indiquez ce que fait votre programme. Peut-on jouer contre la machine ? Peut-on choisir la taille du terrain ? etc.

Il vaut mieux en faire moins mais le faire bien, que d'en faire plus mais mal. Néanmoins à qualité égale la quantité de travail investie dans le projet sera naturellement prise en compte.

## Interface

Vous n'avez pas encore eu de cours sur les interfaces graphiques donc seul le mode console est demandé. Il n'est même pas vraiment obligatoire, mais ce sera très apprécié, d'afficher le terrain de jeu mais dans ce cas apportez une grille le jour de la soutenance !

## Test automatiques

Idéalement chaque fonction de votre programme doit être testée avec des **assertEquals** qui seront placés bien à part, typiquement dans une fonction de test propre à chaque type de données (chaque struct).

Pour de nombreuses fonctions un seul test ne suffit pas : les cas particuliers doivent être couverts.

Le manque ou l'insuffisance de tests écrits correctement sera pénalisé de plusieurs points en moins.

Des tests non automatiques (affichages, traces ...) ne comptent pas !

N'oubliez pas que des tests bien écrits contribuent à préciser le cahier des charges. Soignez leur présentation. Ils doivent être faciles à comprendre. Ce sera pris en compte dans la note.

## Bugs

Votre programme doit fonctionner pour l'essentiel. Un petit nombre de bugs secondaires qui n'empêchent pas de finir une partie ne seront pas trop pénalisés.

La saisie des données, pour ce premier projet, ne doit pas obligatoirement être « blindée » mais c'est un plus qui sera apprécié.

Si vraiment vous avez du mal à faire fonctionner votre programme réduisez drastiquement, au moins dans un premier temps, le cahier des charges : pas d'affichage du terrain, terrain très petit etc.

## Organisation du code

Le code de votre programme doit être organisé comme vu en cours : un fichier .h et un .cpp pour chaque type de données (struct). Le main.cpp doit être aussi court que possible.

Il sera très apprécié de respecter l'encapsulation mais si vous n'y arrivez pas faites au moins un petit programme qui fonctionne.

Les fonctions doivent être aussi courtes que possible.

Les identificateurs doivent avoir des noms « parlants ».

Seuls des commentaires véritablement utiles sont attendus : pas de paraphrase du code.