



*Institut des
Techniques d'Ingénieur
de l'Industrie*

le cnam

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CENTRE REGIONAL ASSOCIE DE STRASBOURG**

SPECIALITE : INFORMATIQUE

Module : Développement C++

**Séance 5 : programmer un jeu permettant de jouer aux jeux de
société Morpion, Puissance 4 et Othello**

Intervenant : Edouard MANGEL

A l'issue de ce cours théorique, il est temps de passer à la pratique. Vous réaliserez ainsi les programmes suivants dans le logiciel QtCreator (ou dans n'importe quel logiciel avec lequel vous êtes à l'aise, mais en totale autonomie).

Vous écrirez **un** programme composé de plusieurs classes qui seront en rapport avec les différents paragraphes suivants. Vous serez évalués sur la qualité du code, l'application des bonnes pratiques évoquées lors des cours théoriques et l'absence d'erreurs dans vos programmes.

Les sources sont à pusher dans votre repo git **sur la branche master** au plus tard la veille de la prochaine séance à 23h59. Vous serez évalués sur la qualité du code, l'application des bonnes pratiques évoquées lors des cours théoriques et l'absence d'erreurs dans vos programmes.

Barème : 10 points pour l'absence de bugs, 10 points pour la qualité du code.

I Le jeu

En utilisant vos sources du TP précédent, vous ajouterez une fonctionnalité qui permette de choisir le jeu auquel on souhaite jouer, parmi le Morpion, Puissance 4 ou Othello.

Vous trouverez les règles du jeu sur le site de la fédération française d'Othello (parce que oui ce jeu déclenche apparemment suffisamment d'intérêt pour qu'il y ait des gens qui aient décidé d'en faire des compétitions et de déterminer qui sera qualifié pour les championnats du monde) <https://www.ffothello.org/othello/regles-du-jeu/>

II Consignes techniques

Utilisez les notions vues pendant la partie théorique du cours, sur la programmation orientée objet. En particulier mais pas seulement :

- Les principes SOLID, (+ loi de Déméter & Tell don't ask)
- Les références ou pointeurs intelligents, (pointeurs C sanctionnés)
- La const-correctness,
- L'héritage, la composition et/ou l'agrégation,
- L'early/late binding (virtual)
- La bonne utilisation des constructeurs, destructeurs...