

# Projet UE D 2024-2025: Gobblets

IMT Atlantique, campus Nantes

October 17, 2025

## 1 Présentation générale du projet

L'objectif de ce projet est d'implémenter une variante du célèbre morpion, appelée *Gobblets Gobblers* par la marque Blue Orange. Il s'agit d'un jeu à deux joueurs de pure stratégie. Il s'apparente beaucoup au morpion, avec les différences suivantes:

- On peut déplacer les pièces en jeu au lieu d'en ajouter une nouvelle.
- Les pièces ont des tailles différentes, et on peut couvrir une pièce plus petite avec une pièce plus grande. Seules les pièces visibles comptent pour la condition de victoire.

Des règles du jeu sont détaillées dans la Section 2. Le projet se déroulera en plusieurs phases principales. On vous fournit du code pour que vous puissiez facilement réaliser et tester votre projet partie par partie. Dans la Section 3, nous vous décrivons les différentes étapes de votre travail d'implémentation du jeu, et les différents rendus attendus. Enfin, la Section 4 les extensions possibles. Il est obligatoire d'implémenter au moins l'une de ces extensions pour être noté sur 20.

## 2 Règles du jeu détaillées

Ce jeu se joue à deux joueurs sur un plateau constitué d'un damier de trois cases par trois cases, comme le morpion. Chaque joueur dispose au début de six pièces de jeu dans sa *maison*. Les pièces peuvent être de trois tailles: petite, moyenne ou grosse, et chaque joueur possède deux pièces de chaque taille. Le propriétaire de chaque pièce du jeu est facilement identifiable (par la couleur ou la forme de la pièce).

Chaque joueur à son tour a le choix entre deux possibilités :

- Placer une nouvelle pièce de sa maison sur une case vide de son choix.
- Prendre une pièce déjà posée et le reposer sur une case de son choix sur le plateau.

Pour placer (ou déplacer) une pièce sur le plateau, il est nécessaire que la case d'arrivée soit libre ou ne contienne que des pièces plus petites que la pièce placée. Dans le cas où une pièce est posée sur une case occupée, celle-ci couvre alors les pièces déjà

présentes, les rendant invisibles. On peut donc couvrir une petite pièce avec une pièce moyenne ou une grande, ou couvrir une pièce moyenne avec une pièce grande. Il est possible de couvrir indifféremment une pièce de l'adversaire ou l'une de ses propres pièces. Une case peut donc comporter au plus trois pièces en même temps, une petite, une moyenne et une grande. Seule la plus grande pièce présente sur la case est visible. Si d'aventure votre adversaire réalise un alignement de 3, il est donc possible d'amener en jeu une nouvelle pièce même sur une case occupée. C'est la seule manière de pouvoir gober en mettant une pièce en jeu.

Lorsque l'on déplace une pièce qui en couvrait une autre, la pièce précédemment cachée redevient alors visible. Si on déplace une grande pièce qui couvrait une pièce petite et une moyenne, seule la pièce moyenne devient visible.

Un joueur remporte la partie lorsqu'il parvient à ce que toutes les pièces visibles d'une ligne, d'une colonne ou d'une diagonale lui appartiennent (quelle que soit leur taille). Dans le cas peu probable où les deux joueurs satisfont la condition de victoire lors d'un même mouvement, la partie continue jusqu'à ce qu'un seul joueur soit en position gagnante. Ce cas peut arriver lors d'un déplacement, la pièce déplacée révèle une ligne gagnante dans la position quittée en même temps qu'elle en crée une avec la case atteinte.

### 3 Déroulement du projet

Pour ce projet, nous vous fournissons des implémentations de modules du jeu pré-compilées en langage C sous la forme de fichiers `.o` accompagnés de leur fichier d'en-tête `.h`.

Une documentation détaillée des fichiers d'en-tête fournis vous permet de générer automatiquement une documentation du projet au format `html` ou `LATEX`. La documentation sera rendu accessible en ligne depuis la plate-forme du module.

#### 3.1 Rendu d'une interface console

- Dans un premier temps, vous chercherez à faire une interface dans la console pour jouer au jeu, en utilisant la bibliothèque `board.o` fournie. Vous pouvez choisir la présentation du jeu selon votre préférence, sans nécessairement suivre l'exemple fourni.
- Votre programme sera rendu sous la forme d'une archive contenant l'ensemble des sources nécessaires à la compilation du projet et un fichier `Makefile` fonctionnel pour compiler le projet. Vous n'incluez pas les fichiers précompilés dont vous fournissez le code source. Vous ne devez pas avoir modifié les fichiers `.h` et `.o` fournis.
- Ce rendu est attendu pour le **18 Novembre 2025** et sera évalué dans la note finale. La facilité de lecture du code, avec notamment un découpage pertinent en fonctions, seront pris en compte pour l'évaluation.

### 3.2 Implémentation de votre propre moteur de jeu en C++

- Dans un deuxième temps, vous aurez à rendre votre propre implémentation C++ du moteur de jeu, en remplacement du fichier `board.o` fourni. Vous devez, pour cela, organiser votre projet en classes. Le fichier fourni `.h` vous aidera à identifier les différentes classes nécessaires pour implémenter votre propre moteur de jeu. Le rendu final devra compiler simplement et sans erreur.
- Ce rendu est attendu pour le **12 Janvier 2026** et sera évalué dans la note finale.

## 4 Extensions

Chacune des extensions suivantes vous permettra d'améliorer votre note. Ce sont des tâches a priori plus délicates ou plus longues que le reste du projet décrit jusque là. Nous vous encourageons à bien finaliser votre projet avant de vous y lancer, celui-ci correspondant à la majeure partie de la note. Chaque extension sera implémentée dans un projet spécifique, accompagné des différents fichiers nécessaires à la compilation.

### 4.1 Bot

Nous vous invitons à faire ce qu'on appelle un bot, c'est à dire un programme qui se substitue au joueur. Idéalement, vous pourriez proposer une version où il est possible de régler la difficulté du bot, un bot facile jouant presque au hasard tandis qu'un bot difficile jouera très astucieusement.

### 4.2 Interface graphique

Cette extension consiste à implémenter une interface graphique pour le jeu. Nous vous encourageons à utiliser la bibliothèque QT.

Bonne programmation!