

# Réseau : Projet 2024-2025

D.Moreaux

8 septembre 2024

## 1 Abstract

Pour ce projet, il faudra réaliser un système multijoueur permettant de jouer au jeu Pente.

Le serveur sera réalisé en langage C et le client en Python avec Pygame-ce (<https://pyga.me/docs/>) et éventuellement le système Pygame GUI (<https://pygame-gui.readthedocs.io/en/latest>).

Les données des joueurs (identifiant, nombre de parties jouées et gagnées, score) seront sauvées dans une base de données SQLite3 (<https://www.sqlite.org/>) sur le serveur. Les parties en cours ne seront pas sauvées dans la DB.

## 2 Pente

Le jeu de Pente se joue sur un plateau de jeu de 19x19 lignes (on joue sur les intersections). L'un des joueurs prend les pions blancs et l'autre les pions noirs<sup>1</sup>

Chaque joueur pose un pion à son tour, là où il le désire. Le but du jeu est d'arriver à aligner 5 pions de sa couleur selon une ligne, une colonne ou une diagonale.

## 3 Le serveur

Le serveur sera en écoute sur le port 55555 en TCP. La mémoire sera gérée de façon dynamique (malloc/free).

Une fois connecté, le joueur doit s'authentifier. S'il n'existe pas, il sera créé avec le mot de passe entré et il commencera avec un score de 1000.

---

1. On peut aussi utiliser les couleurs rouge et verte utilisées dans le jeu de plateau. Pente est une version simplifiée du jeu gomoku qui se joue sur un plateau de Go

Le serveur pourra envoyer la liste des parties ouvertes avec le nom, le score et les nombres de victoires et de parties jouées. Un joueur pourra décider de rejoindre une partie ouverte ou d'en créer une nouvelle.

Le serveur gèrera la partie : plateau de jeu, vérification de la validité des coups et détection de la victoire d'un joueur.

En fin de partie, chaque joueur gagnera ou perdra un certain nombre de points. Le nombre de points gagnés dépendra des scores des joueurs.

$$\text{delta} = \frac{30}{1 + 10^{(s_g - s_p)/400}}$$

$s_g$  est le score du gagnant et  $s_p$  le score du perdant. Cette formule permet de gagner plus de points quand on joue contre un adversaire plus fort que soi (et d'en perdre plus quand on joue contre un joueur plus faible) et est similaire à la formule du classement ELO<sup>2</sup>. Les scores des joueurs sont des entiers.

Toute partie commencée sera considérée comme jouée (même si elle est abandonnée suite à une perte de connexion).

## 4 Le client

Le client commencera par se connecter sur le serveur. En cas d'échec de la connexion, il affichera un message d'erreur.

Il demandera ensuite les identifiants (login/password). En cas d'erreur, il affichera un message et restera sur l'écran de connexion.

Une fois connecté, il récupèrera les informations du joueur (score, victoires, défaites, total) et les informations sur les parties disponibles.

Le joueur arrivera ensuite sur l'écran de jeu. S'il n'a pas encore d'adversaire, un message sera affiché indiquant qu'il est en attente de jouer.

Une fois la partie terminée, il affichera un message de victoire ou de défaite, le gain (ou la perte) de score. Quand le joueur cliquera sur le bouton ou la fenêtre (au choix), il reviendra sur l'écran de parties ouvertes.

Si le joueur demande de fermer la fenêtre, le client enverra un message au serveur indiquant la fermeture de connexion avant de s'arrêter.

---

2. Penser à utiliser des `double` pour les calculs