

# Réseau: Morpion Aveugle

## Soutenance de Projet

Mathieu PERROY - Yoan MARTY, Licence 3 Info de l'

université  
de BORDEAUX

Plan:

- Conception
- Extensions

# Concept

---

Basé sur un échange Client-Serveur

Usage de paquets de taille 1

Communication asynchrone (les clients ne savent pas ce que l'autre fait)

# Conception - Serveur : Méthodes

— — —

`host_start()`: - Création de socket `AF_INET` (ipv4) en écoute.

`host_main()`: - Partie principale du serveur, `game-handler`.

`waiter()`: - Attente de la connexion d'un client.

`add_player(player_s)`: - Ajout du joueur *player\_s* dans la partie.

`status(data,index)`: - Envoi d'information (*data*) au client *index*

`client_getinfo(id)`: - Renvoie la socket du client *id*.

`delete_user(player_s)`: - Suppression du joueur *player\_s* des listes du serveur.

# Conception - Serveur : Variables

— — —

`host` : - le socket du serveur.

`client_select` : - liste des sockets des clients pour *select*

`client_socket` : - liste des sockets des clients pour la partie.

`data` : - informations envoyées par le client.

`grids` : - grilles du jeu : dans l'ordre, Total, J1, J2.

# Conception - Client : Méthodes

— — —

`client()`: - Partie principale du client.

`init()`: - Création de la socket client en `AF_INET` (ipv4) et initialisation de la grille.

`me_player()`: - Connexion de la socket client au serveur.

# Conception - Client : Variables

— — —

me: - le socket du client.

grille: - la grille du client.

game: - interrupteur réseau pour démarrer la partie.

fiyerd: - interrupteur pour démarrer le tour du client.

me\_attack: - indicateur du coup joué.

data: - données reçues par le serveur.

egg: - Error-handler et compteur de mauvaises utilisations lors du tour de jeu.

# Extensions

— — —

Redémarrage de partie.

Jeu contre une machine.

Gestion de la déconnexion du joueur en jeu.



# Redémarrage de partie

— — —

Nouvelles méthodes:

Client:

`relog()`: - Réinitialise la connexion client-serveur.

Serveur:

`reset()`: - Réinitialise *client\_select*, *client\_socket*.

# Jeu contre une machine

— — —

Nouvelle méthode:

`vs_ia()`: - Fonction contenant le jeu contre la machine.

Nouvelle action du client:

Choix de l'adversaire (`IA/Humain(Réseau)`)

# Gestion de la déconnexion

— — —

Serveur :

Couple try/except sur *ConnectionResetError* et *AssertionError*

Echange d'informations sur la déconnexion avec le client restant s'il existe.

Attente des joueurs manquants

Client:

Réception d'une donnée différente du début de tour pour indiquer la déconnexion de l'adversaire.