



MANHATTAN PROJECT

Degroodt Nicolas, Hernalsteens Cédric,
Lumban Tobing Samuel, Mouak Khaled

Département d' Informatique.

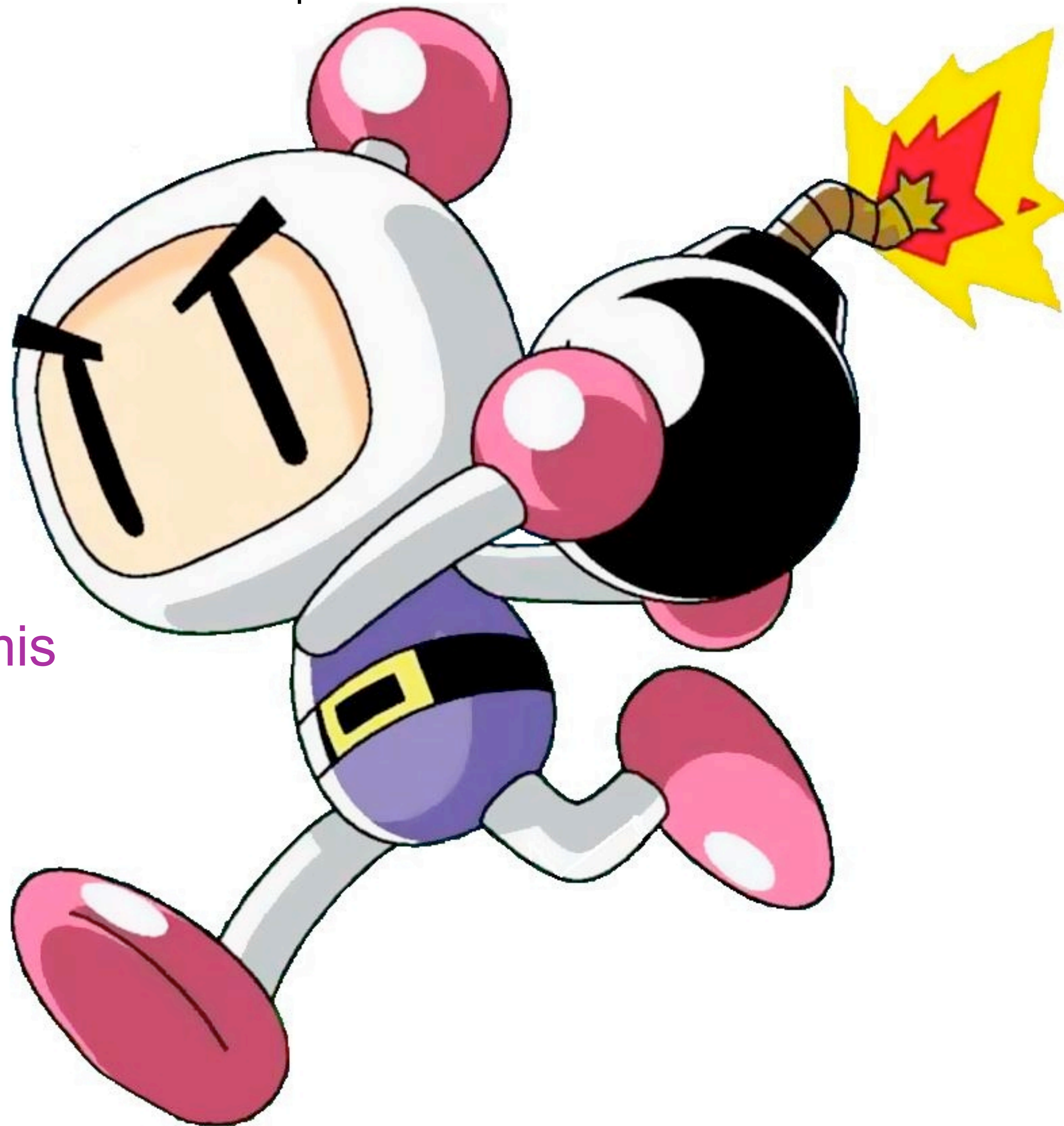
Bomberman Vs Worms



Jouez en réseau



Affrontez-vous entre amis
De 1 à 6 joueurs



Défiiez nos intelligences artificielles
Seul ou à plusieurs



Personnalisez votre bomberman



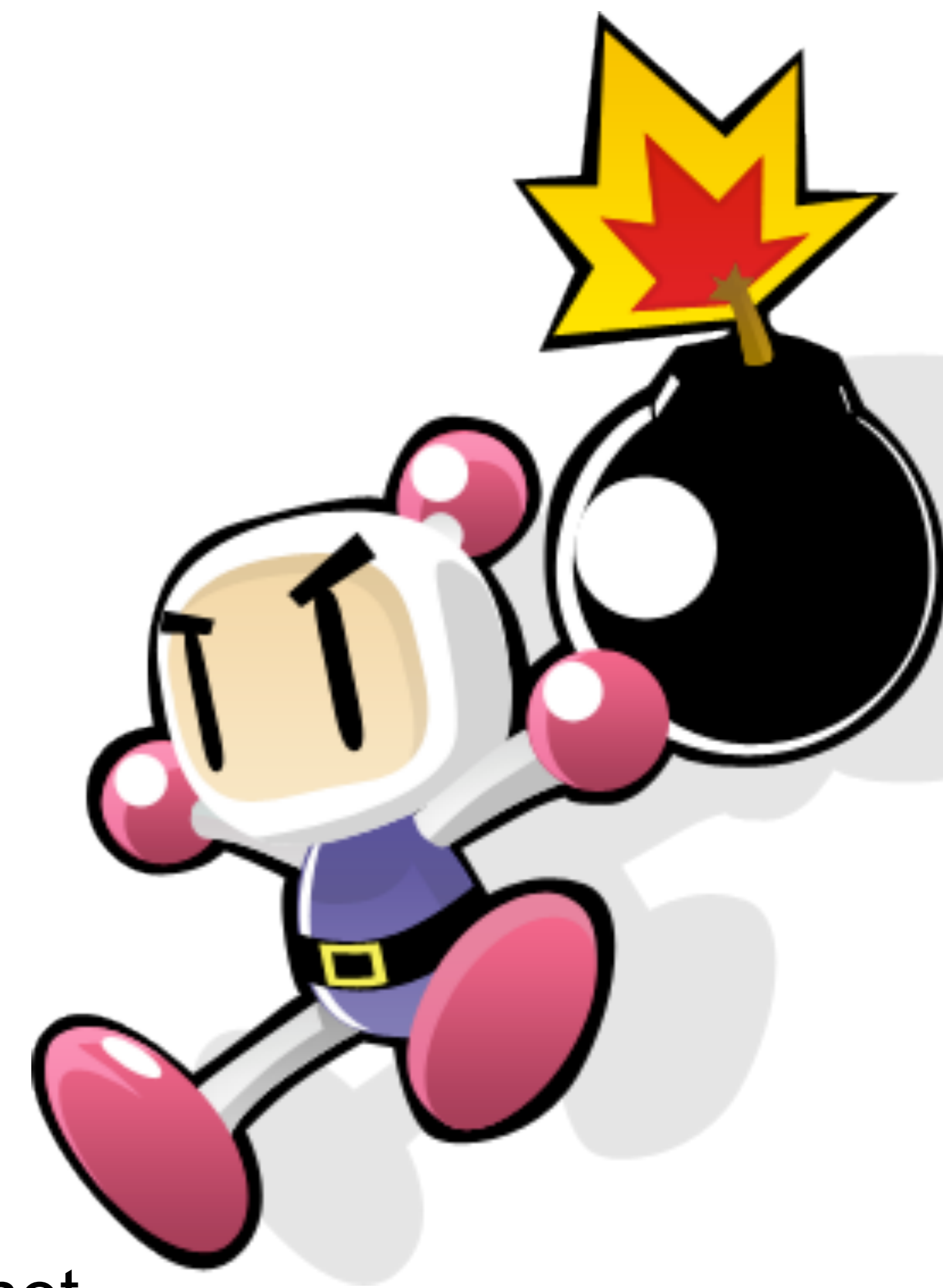
Communiquez entre vous grâce à un chat



MANHATTAN PROJECT

Degroodt Nicolas, Hernalsteens Cédric,
Lumban Tobing Samuel, Mouak Khaled

Département d'Informatique.



Manhattan Project

Manhattan Project est un jeu basé sur le jeu Bomberman auquel sont rajoutées de nombreuses options notamment le multijoueur online.



Objectifs

- **Multijoueurs Online** : jusqu'à 6 joueurs via internet
- **Facilité d'utilisation** : Installation facile et prise en main intuitive
- **Immersion graphique qui décoiffe**
- **Jouable sous Windows, Linux et MacOS X**

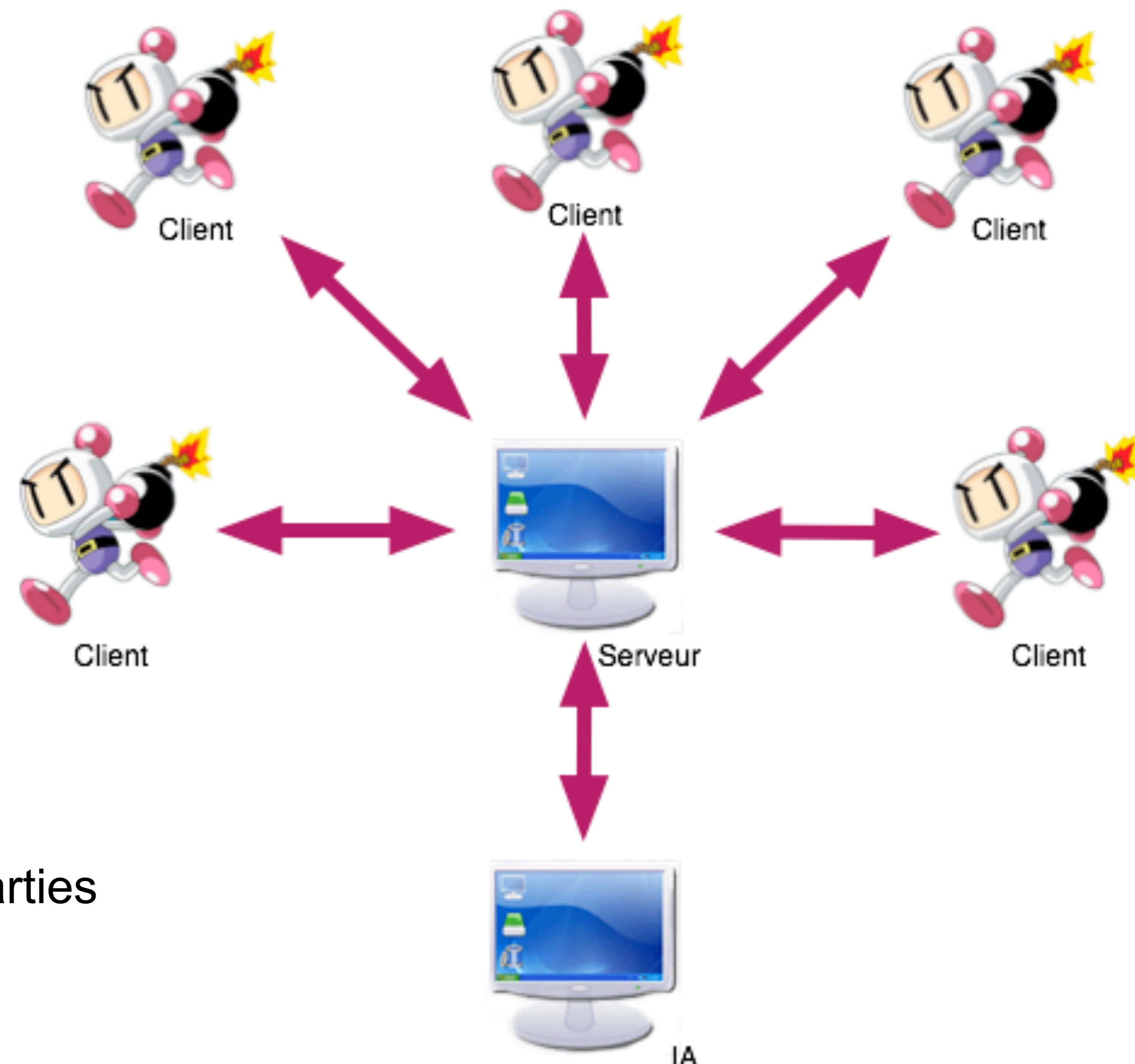
Modes de jeu

- **Par équipe** : Tentez de battre notre IA avec vos amis
- **Survivor** : Restez le dernier joueur en vie sur la map

Un jeu multijoueur Online

Manhattan Project se base sur un modèle client serveur et fournit trois programmes :

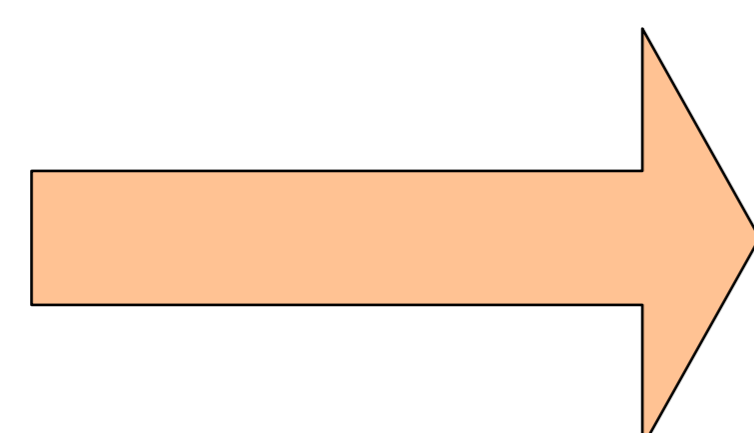
- le **serveur** : gère le déroulement du jeu
- la **GUI** (Graphical User Interface) : l'interface graphique du jeu pour l'utilisateur
- l' **IA** (Intelligence artificielle) : gère des ennemis



Organisation du serveur

- Gère les connexions de nouveaux clients
- Centralise les données concernant l'état des parties
- Traite les requêtes des clients
- Gère les transferts vers et depuis le réseau

Besoin de traitements parallèles ...



- Découpe modulaire en processus indépendants
- Communication entre ces processus
- Rapidité de traitement pour chaque client... pas de temps d'attente



MANHATTAN PROJECT

Degroodt Nicolas, Hernalsteens Cédric,
Lumban Tobing Samuel, Mouak Khaled

Département d'Informatique.

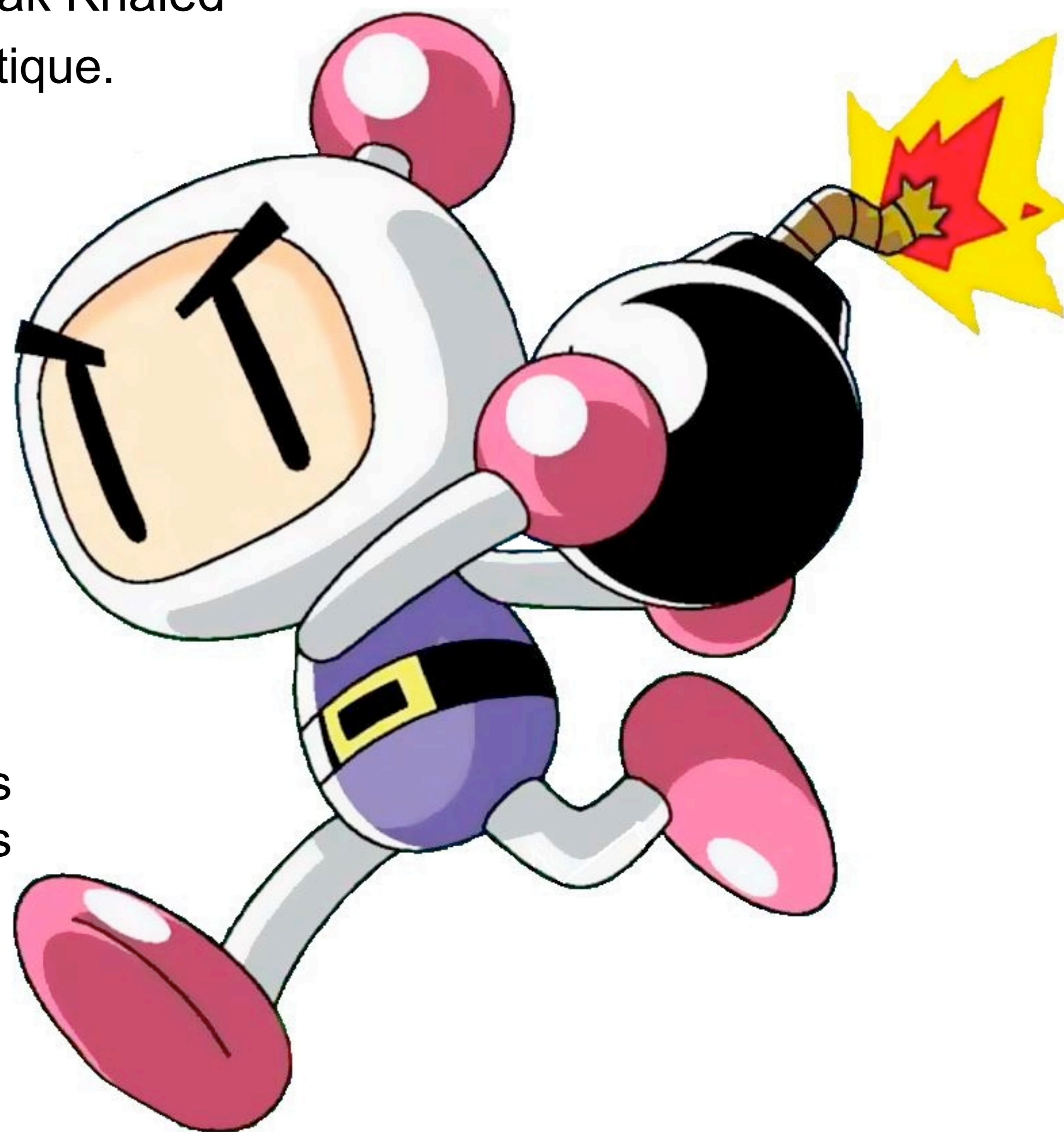
Interface graphique utilisateur

L'interface graphique se connecte au serveur

→ Interprète les actions du joueur et transmet des requêtes au serveur

→ Reçoit des messages du serveur et met à jour l'affichage de tous les joueurs

→ Fonctionnement asynchrone de modules indépendants utilisant le mécanisme des signaux de Qt



Programmation de l'intelligence artificielle

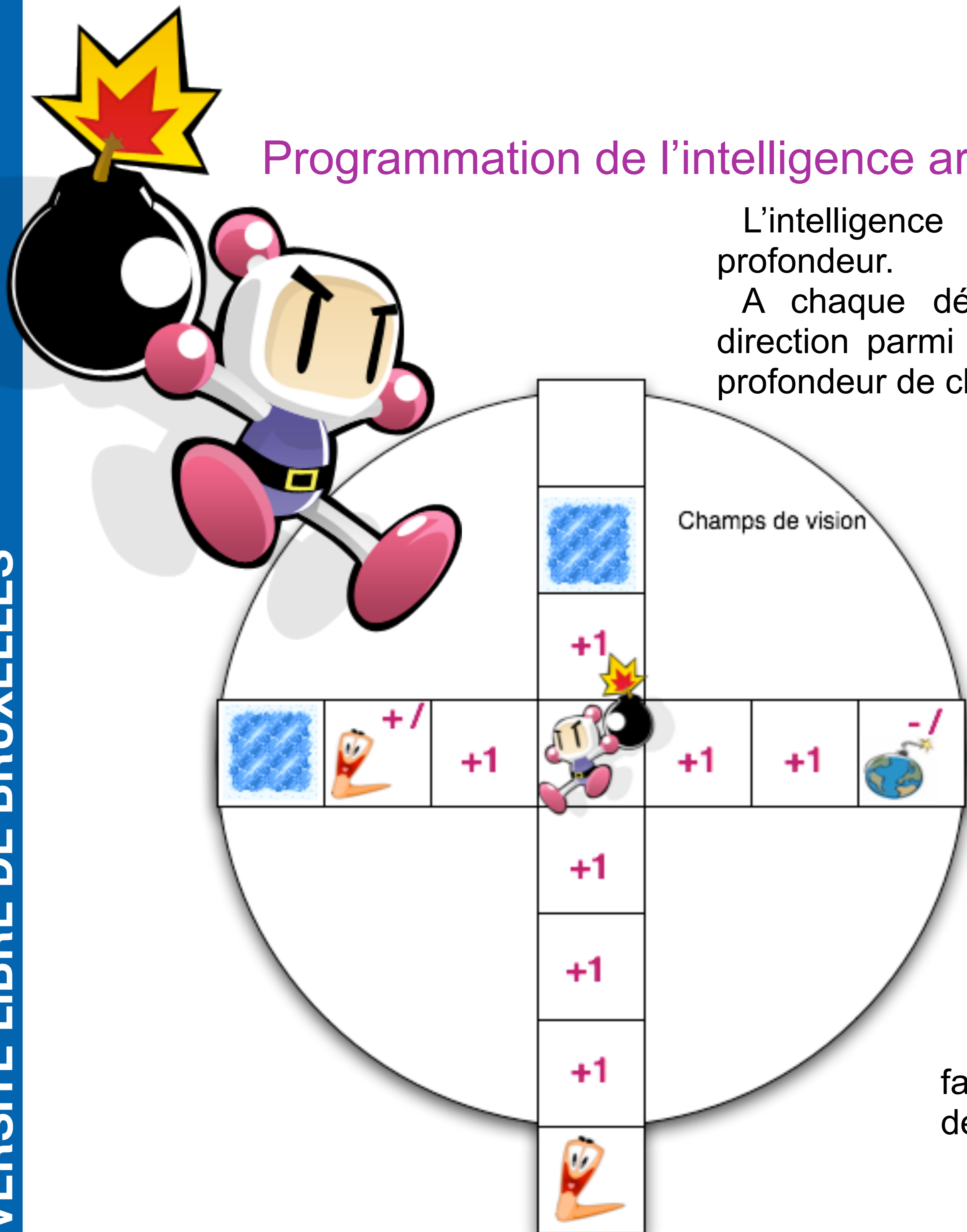
L'intelligence artificielle se base sur un simple concept de profondeur.

A chaque déplacement, notre bomberman doit choisir une direction parmi 4 (haut, bas, gauche et droite). L' IA calcule la profondeur de chaque direction et choisit la plus grande.

Calcul des profondeurs

Le bomberman est attiré par la profondeur la plus grande.

Les bombes et les adversaires augmentent ou diminuent la profondeur s'ils se trouvent dans le champs de vision; ces variations sont pondérées par la distance séparant les objets du joueur de l' IA.



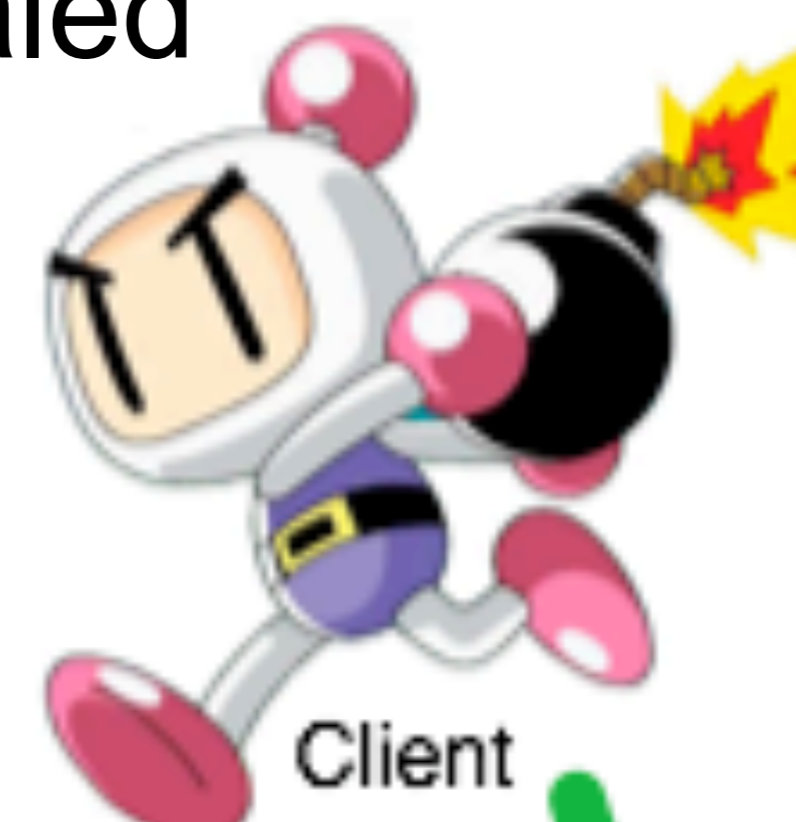
La notion de profondeur peut facilement s'étendre et rendre l'IA de plus en plus compétitive.





MANHATTAN PROJECT

Degroodt Nicolas, Hernalsteens Cédric,
Lumban Tobing Samuel, Mouak Khaled
Département d'Informatique.



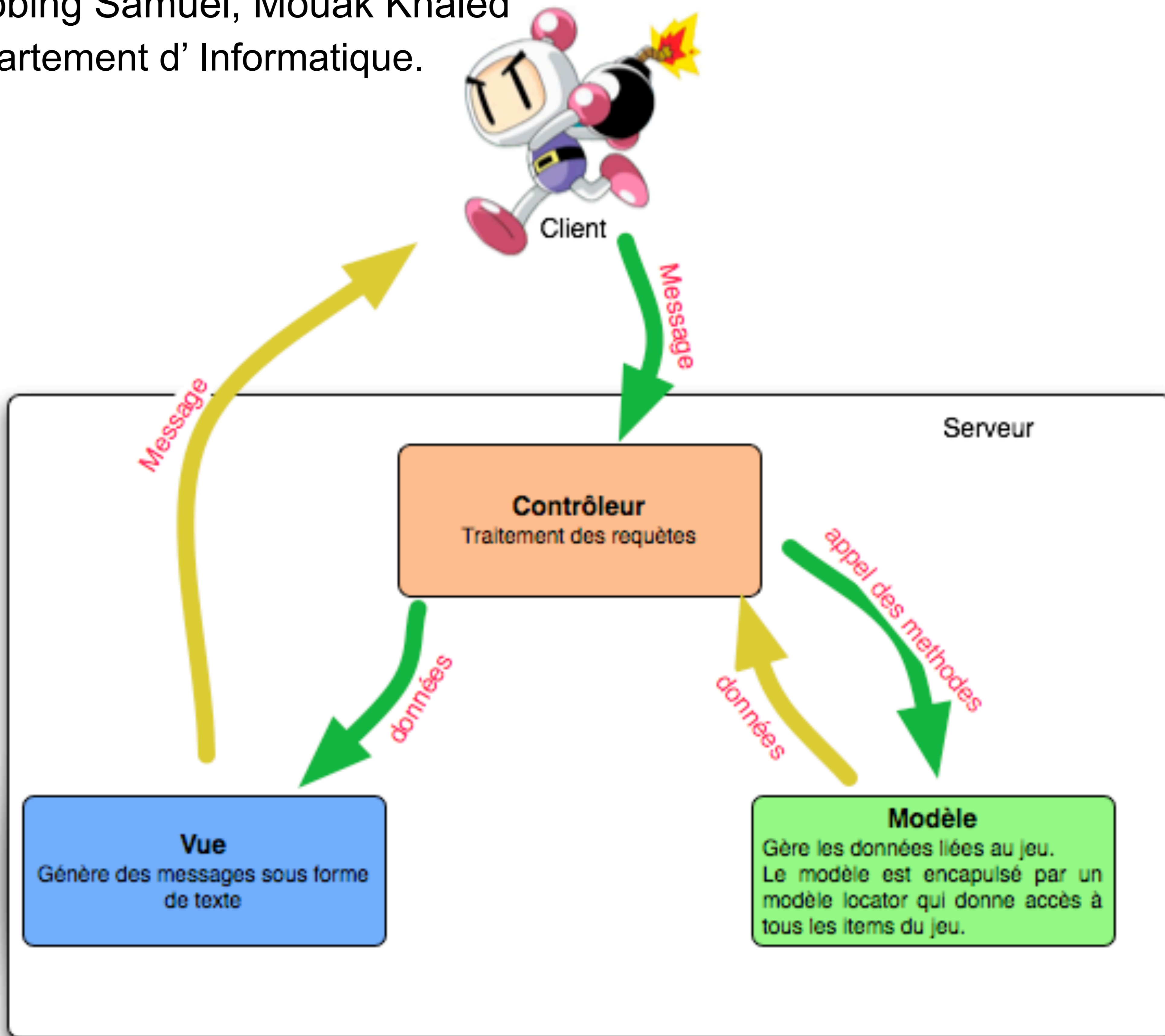
Le Pattern MVC

Pattern ?

Un pattern est un modèle de solution réutilisable correspondant à un problème informatique courant - indépendant de l'implémentation.

MVC ? Modèle Vue Contrôleur

- **Modèle** : prend en charge les données
- **Vue** : Sortie du système vers le réseau
- **Contrôleur** : gère le système, implémente la logique du jeu



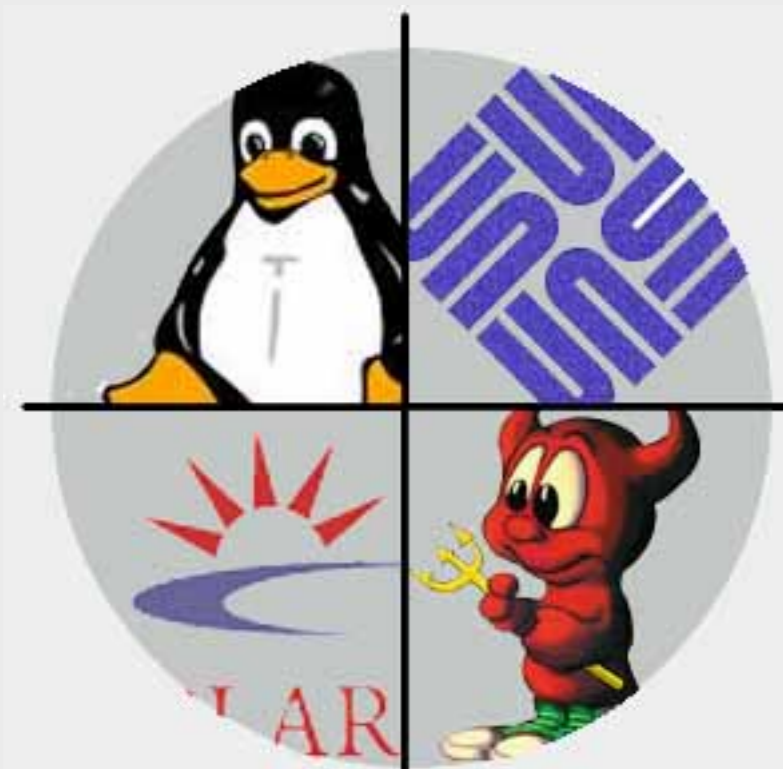
Processus et réseau

Les processus sont des modules indépendants travaillant en parallèle.

- Traitement des requêtes
- Acceptation de connexions
- Transferts réseaux entrants
- Transferts réseaux sortants

doivent **communiquer** entre eux et avec les clients

- Message Queue IPC System V
- Mémoire partagée
- Communication avec le client via le biais de sockets



Implémentation

Serveur & IA

- **C / C++**
- **libC** : librairie standard C permettant des appels système Unix
- **STL** : librairie standard C++

GUI

- **C++**
- **Qt** : librairie graphique multiplateforme

