

Projet Timebomb - Rapport n°2

Interfaces entre les différents modules



Guillaume BEQUET

Samuel DIERSTEIN

Craig HIERNARD

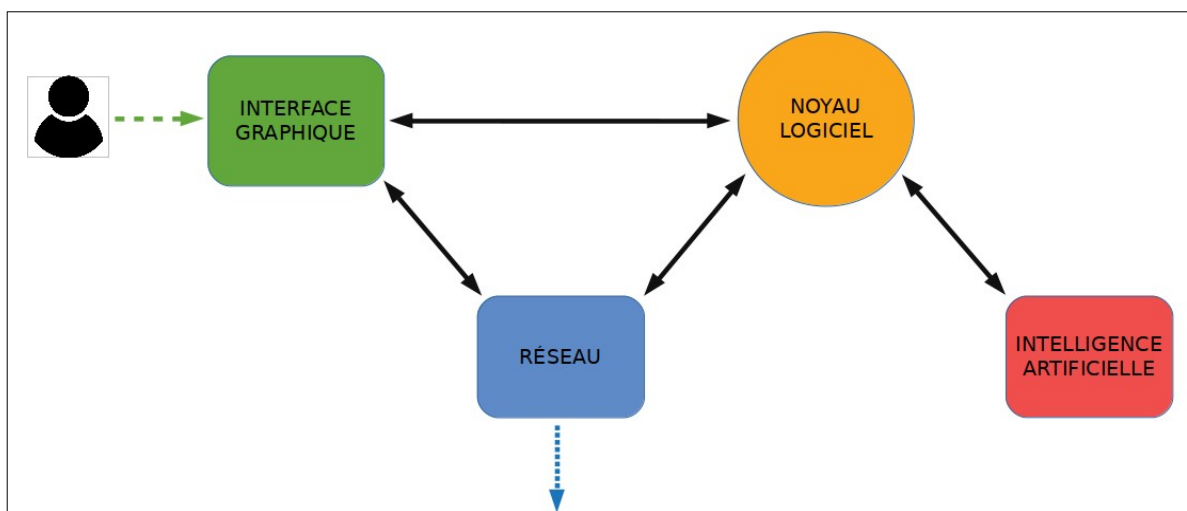
Arthur KLIPFEL

Joren SCHUSTER

Parmi les modules proposés, nous avons choisi d'en implémenter quatre : interface graphique, noyau logiciel, réseau (pour les parties en ligne) et intelligence artificielle (pour les bots).

La FIGURE 1 ci-dessous illustre les différents modules et leurs interfaces : chacune correspond à une flèche à double sens qui relie deux modules. Au total, nous décrivons donc quatre interfaces.

FIGURE 1 — *Schéma des interfaces entre les différents modules*



Remarque : Lorsqu'un joueur utilise le logiciel, il manipule en réalité directement l'interface graphique (cf flèche verte en tirets sur la FIGURE 1).

1. Interface graphique ↔ Noyau logiciel

Dans le cas d'une partie hors-ligne, l'interface graphique communique directement avec le noyau logiciel, le module réseau n'étant pas nécessaire.

Lorsqu'une action est effectuée par un joueur, l'interface graphique envoie les données correspondantes (carte cliquée, câble coupé, etc.) au noyau logiciel qui se charge du traitement de ces informations et renvoie les données nécessaires au rafraîchissement de l'interface graphique du joueur.

Par exemple, à chaque fois qu'une carte est coupée, l'interface envoie des informations relatives à la coupe d'une carte spécifique. Le noyau logiciel retire donc cette carte de la liste de cartes du joueur (et donc du tas de cartes de la partie) pour ce tour, et met à jour son nombre total de cartes. Celui-ci renvoie alors la liste de cartes actualisée à l'interface, et adapte l'affichage des cartes du joueur concerné. Ici, le joueur dont la carte a été coupée a donc une carte en moins et les informations relatives à la partie (nombre de câbles de désamorçage, nombre de câbles classiques coupés, etc.) sont mises à jour.

L'architecture du code entre ces deux parties repose sur un modèle « *Modèle - Vue - Contrôleur* », ou MVC. La partie *Vue* contient l'interface graphique et tous les éléments relatifs à l'affichage et directement visibles par l'utilisateur.

La partie *Modèle* est constituée des classes "métier" du jeu. La logique contenue dans cette partie ne permet de manipuler qu'une instance à la fois (obtenir le nombre ou la liste de cartes d'un joueur en particulier, la nature d'une carte en particulier, modifier le rôle d'un joueur, etc.).

La partie *Contrôleur* manipule plus largement les données, en réalisant des opérations sur un ensemble d'instances comme le mélange des cartes, l'attribution des rôles, etc.

Cette partie constitue la majorité de la logique contenue dans le noyau.

Ce modèle nous permettra de bien organiser notre code, d'avoir une architecture claire avec des méthodes réparties dans chacune des parties leur étant propre. De plus, cela permettra d'éviter les conflits en nous répartissant le travail entre ces différentes parties.

2. Réseau ↔ Interface graphique

Lors d'une partie multijoueur, le joueur peut tout d'abord choisir un salon dans lequel entrer.

C'est ici qu'intervient en premier le module réseau, qui se charge de communiquer à l'interface graphique la liste des salons existant ayant une ou plusieurs places disponibles.

Le joueur a également la possibilité de créer son propre salon. L'interface graphique communique alors toutes les informations entrées au travers des formulaires de création de salon (nombre de joueurs, présence de *tokens*...) au module réseau, qui se charge de créer le salon conformément aux informations transmises par l'interface graphique.

Lors d'une partie, l'interface graphique du joueur actuel communique au réseau la carte sélectionnée ainsi que son propriétaire, afin que le module réseau puisse transmettre l'information au module réseau de chacun des autres joueurs de la partie^[1]. Chacun de ces modules réseaux transmet à son tour cette information à l'interface graphique respective de chaque joueur.

Le module réseau gère également la bonne transmission des messages écrits dans le chat aux autres joueurs. Chaque joueur pouvant écrire à tout moment dans le chat, l'interface graphique transmet chaque message au module réseau qui se charge ensuite de l'envoyer à tous les joueurs de la partie^[1], sauf son expéditeur. Chaque message est réceptionné par le module réseau des autres joueurs et est ensuite transmis à leur interface graphique.

[1] *Les échanges entre le module réseau d'un joueur et le module réseau de chacun des autres joueurs sont représentés par la flèche bleue en pointillés sur la FIGURE 1.*

3. Réseau ↔ Noyau logiciel

Le module réseau communique la carte sélectionnée par le joueur actuel au noyau logiciel, qui se charge de déterminer l'action adéquate à effectuer, puis la communique au module réseau qui se charge de diffuser l'action aux modules réseaux des autres joueurs qui à leur tour communiquent cette information à l'interface graphique de chacun d'entre eux.

Imaginons par exemple que le module réseau reçoive une carte à couper. Le réseau ignore lui-même le type de la carte, mais connaît son propriétaire et sa place dans la liste de cartes.

Il communique ces deux informations au noyau logiciel, qui détermine le type de la carte (câble, désamorçage, ou Big Ben) et demande alors au module réseau, en fonction de l'état actuel de la partie, soit de transmettre les informations actualisées de l'état de la partie (*tokens*, nombre de cartes de désamorçage restantes, etc) au noyau logiciel de chaque joueur, qui communique ensuite ces changements à leur interface graphique respective ; soit de communiquer directement la fin de la partie à l'interface graphique de chaque joueur (par exemple si la carte Big Ben a été coupée), car il n'y a plus d'opérations à effectuer dans le noyau logiciel de chaque joueur.

4. Noyau logiciel ↔ Intelligence artificielle

Le noyau logiciel transmet au module d'intelligence artificielle l'état actuel de la partie ainsi que les paramètres qui lui sont nécessaires pour prendre une décision. Le module d'intelligence artificielle s'occupe uniquement du calcul du tour actuel, lorsque c'est à un bot de jouer, et transmet le résultat au noyau logiciel. Si plusieurs bots sont présents, le noyau logiciel fait appel au module d'intelligence artificielle à chaque fois que c'est à un bot de jouer.