

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

Projet de session - TP3

Jacques Berger

**DANS LE CADRE DU COURS
Génie Logiciel: conception
INF5153
Groupe: 40**

**PAR
Jean-Michel Poirier (POIJ26089200)
Audrey Eugene (EUGA21589707)**

21 décembre 2017

Diagrammes de séquences	3
DS1 – Sauvegarder une partie	3
DS2 – Charger une partie	4
DS3 – Jouer un coup	5
DS4 - Enregistrer un nouveau temps	6
DS5 - Consulter les meilleurs temps	7
DS6 - Visualiser le mouvement suivant d'une partie	8
DS7 - Visualiser le mouvement précédent d'une partie	9
DS8 - Quitter le jeu	10
DS9 - Retourner au menu principal	11
DS10 - Choisir de jouer contre un humain	12
DS11 - Choisir de jouer contre une IA de niveau débutant	13
DS12 - Choisir de jouer contre une IA de niveau avancé	14
Diagramme de packages	15
Diagramme de classes	17
Patron Singleton	20
Patron Façade	21
Différence entre l'implémentation et la conception	23
Glossaire	25

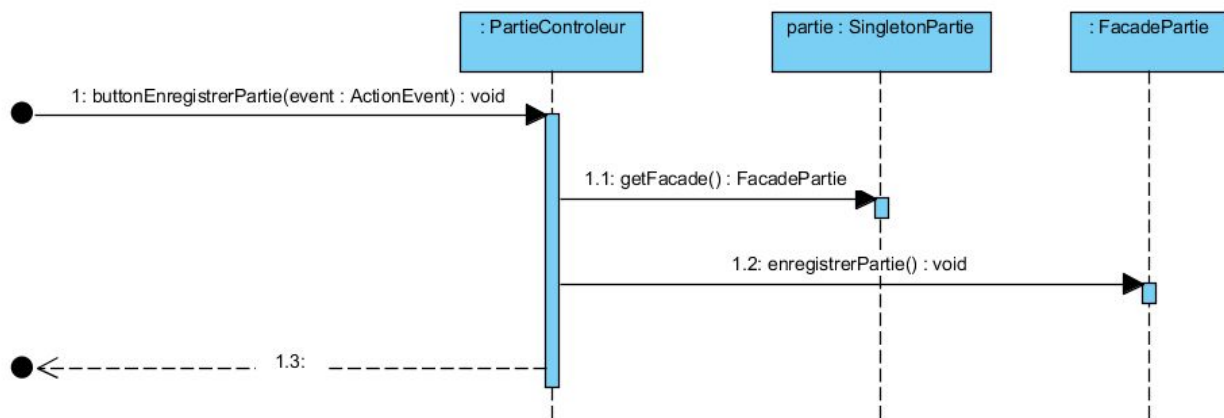
Diagrammes de séquences

Les diagrammes suivants modélisent les différentes fonctionnalités du système au travers des échanges des objets dans le temps.

Pour chacune des situations qui vont suivre, le diagramme associé présentera les messages échangés par l'utilisateur, le joueur artificiel et le système de jeu.

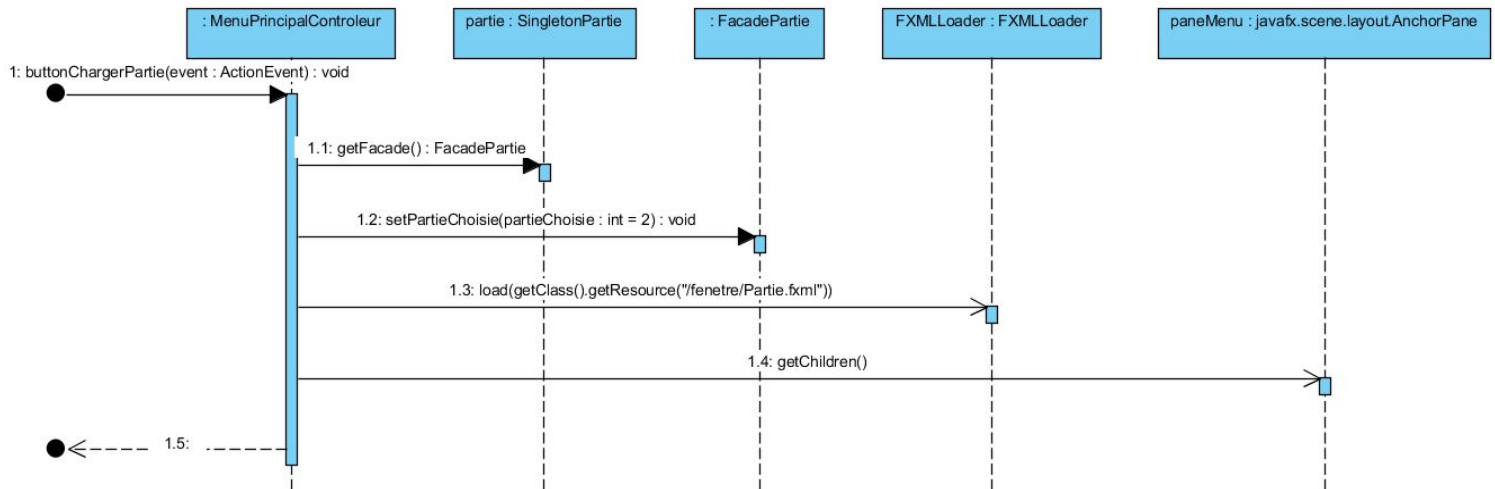
DS1 – Sauvegarder une partie

L'utilisateur souhaite arrêter la partie et voudrait pouvoir la reprendre par la suite. Lorsque l'utilisateur choisit un joueur artificiel comme adversaire et la partie est commencé, celui-ci à l'option de sauvegarder ça partie en cours. La partie se sauvegarde dans un document XML sur la machine de l'utilisateur.



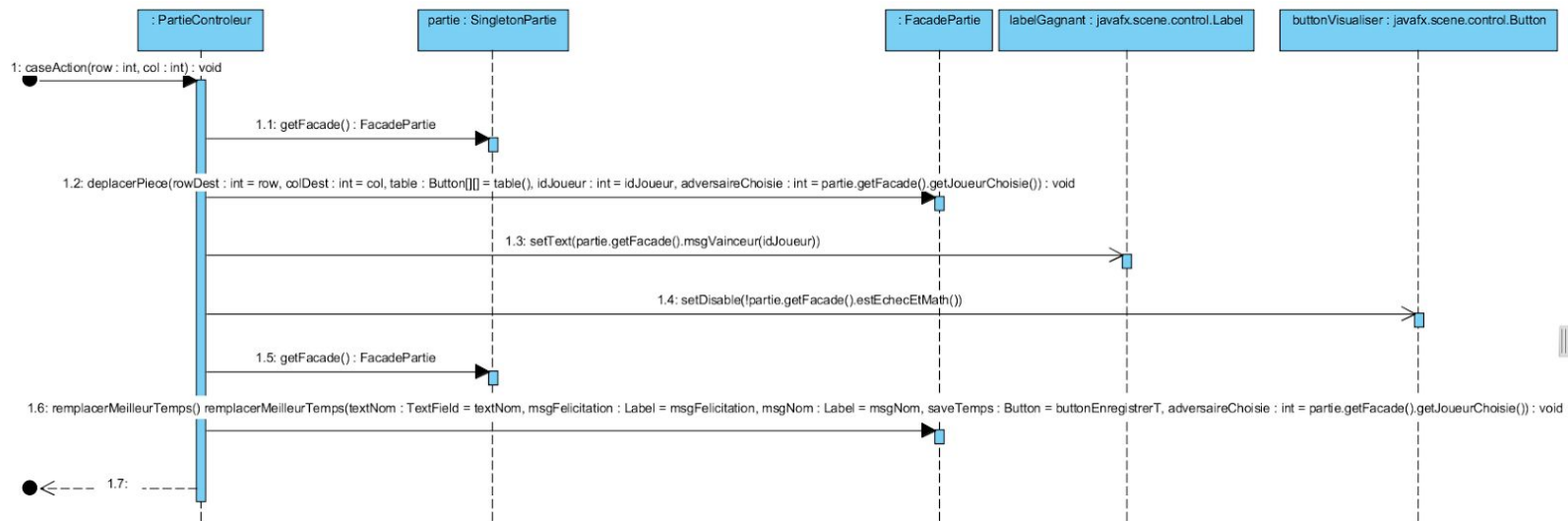
DS2 – Charger une partie

L'utilisateur souhaite reprendre une partie sauvegardé. Dans le menu principal, l'utilisateur à l'option de charger une partie qui a été précédemment sauvegardé dans un fichier XML de sa machine local.



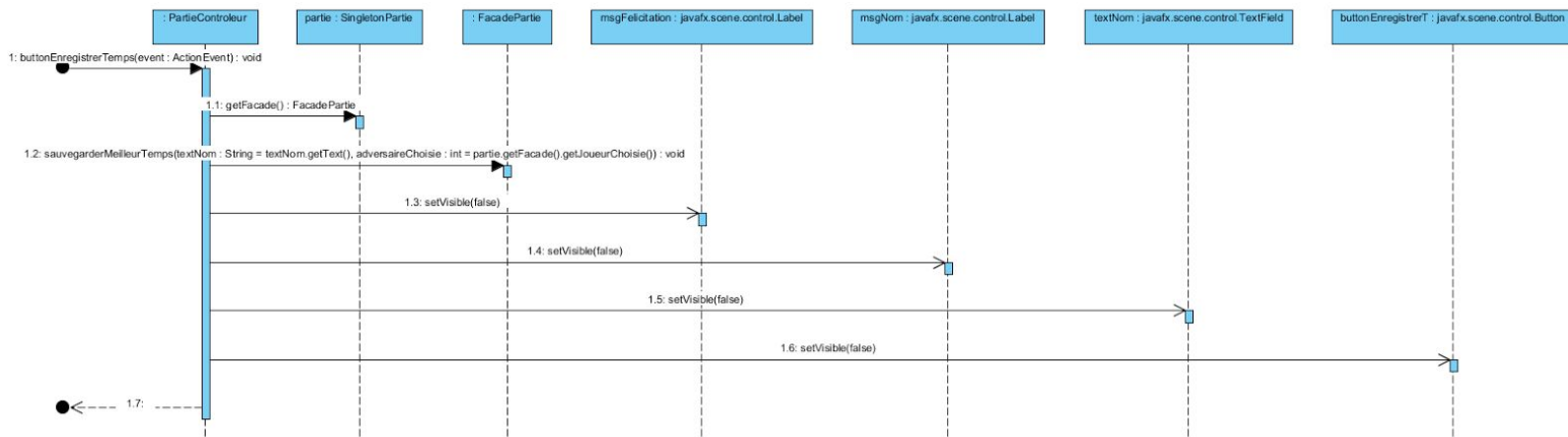
DS3 – Jouer un coup

L'utilisateur et l'adversaire (un joueur humain ou artificiel) joue un coup à tour de rôle jusqu'à ce que la partie soit nulle ou terminée (c'est-à-dire lorsqu'un des deux rois à été éliminé).



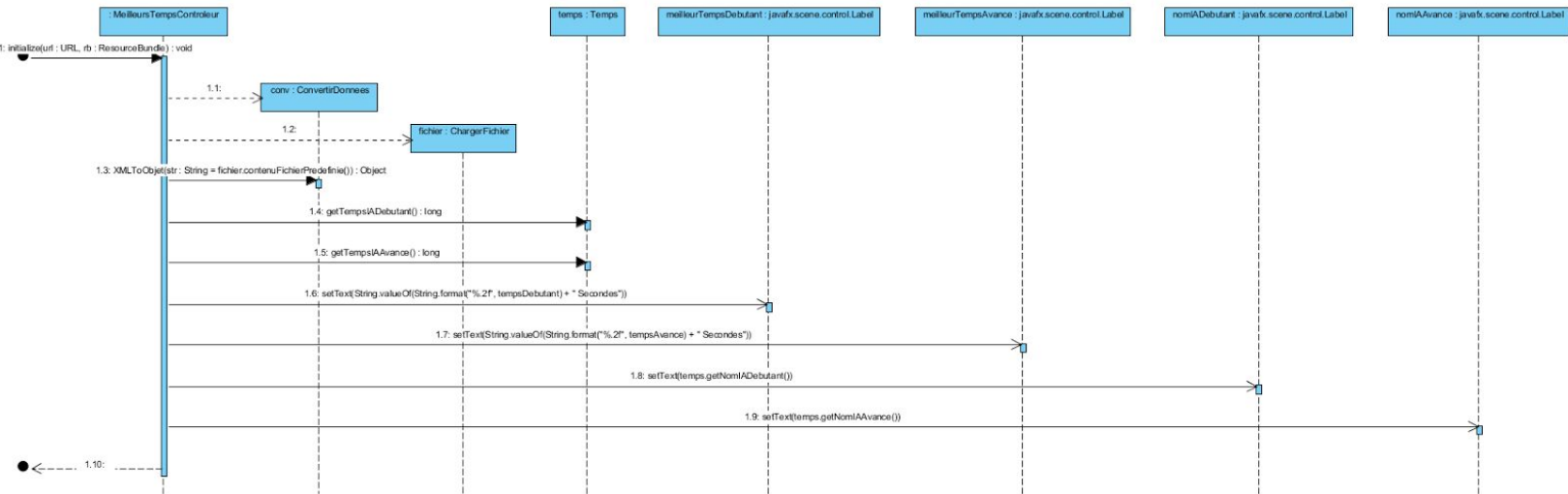
DS4 - Enregistrer un nouveau temps

L'utilisateur a eu un nouveau record de temps dans un niveau de difficulté contre un joueur artificiel. Le nouveau record de temps, le nom de l'utilisateur et la difficulté du joueur artificiel vaincu est enregistré dans un fichier sur la machine de l'utilisateur.



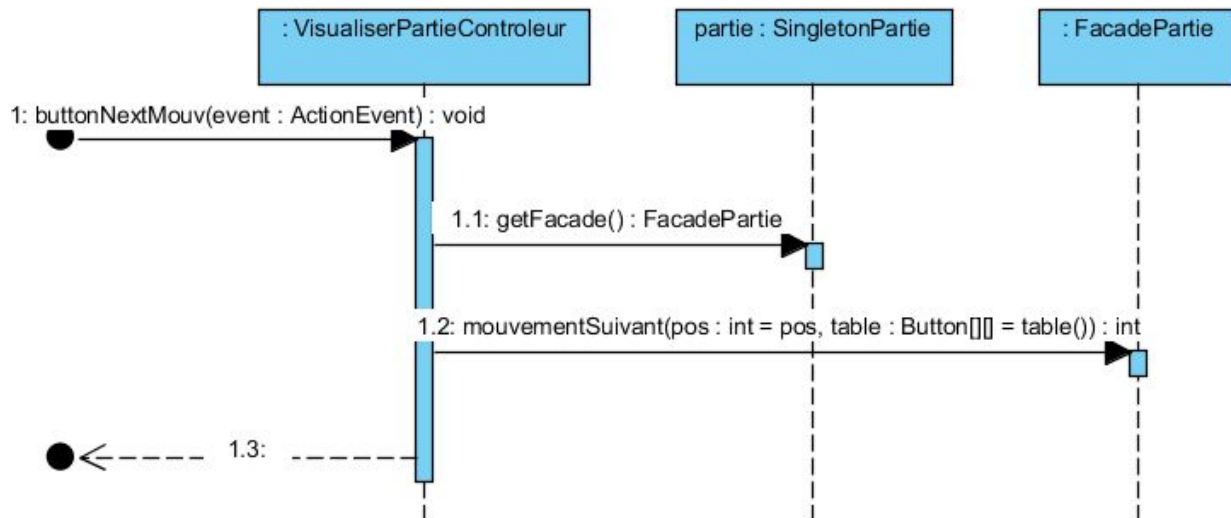
DS5 - Consulter les meilleurs temps

L'utilisateur veut voir les meilleurs temps contre les joueurs artificiels. Cette page va afficher à l'utilisateur le nom du joueur, son meilleur temps et le niveau de difficulté du joueur artificiel vaincu. Cette fenêtre va afficher un meilleur temps pour chaque niveau de difficulté de l'IA.



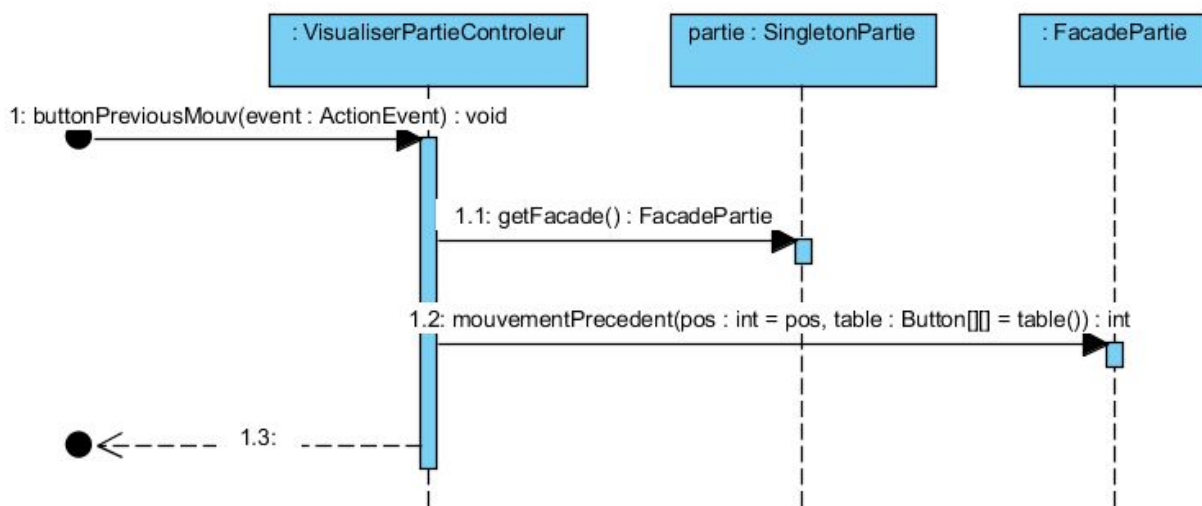
DS6 - Visualiser le mouvement suivant d'une partie

À la fin d'une partie, l'utilisateur veut revoir la partie dans son intégralité. Une fois la visualisation activée, un tableau d'échec d'une partie vierge est affiché. Puis, après chaque fois que l'utilisateur appuie sur le bouton pour afficher le coup suivant, le tableau d'échec se modifie avec les mouvements précédemment enregistrés.



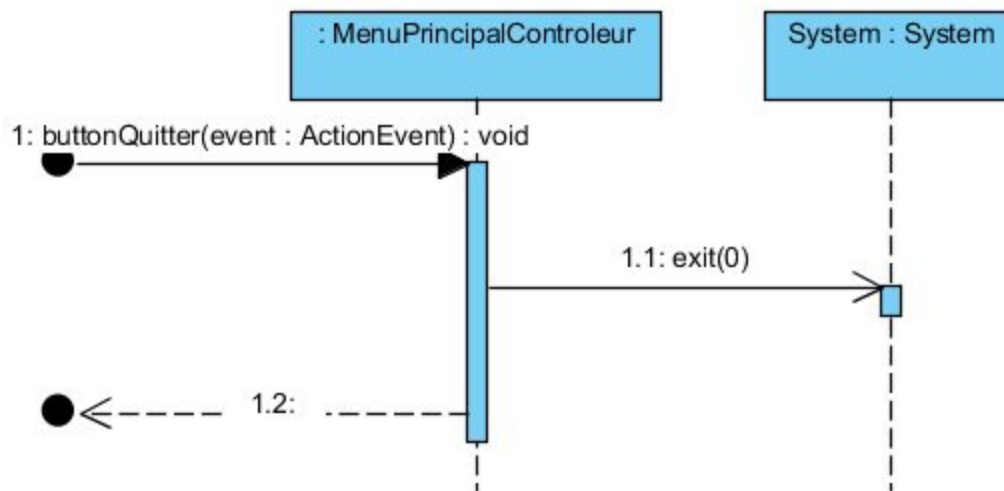
DS7 - Visualiser le mouvement précédent d'une partie

À la fin d'une partie, l'utilisateur veut revoir la partie dans son intégralité. Une fois la visualisation activée, un tableau d'échec d'une partie vierge est affiché. Puis, après que l'utilisateur est appuyé au moins une fois sur le bouton suivant, l'utilisateur peut appuyer sur le bouton précédent pour lui permettre de visualiser les coups joués dans le sens inverse jusqu'à l'arrivée de la table vierge de début.



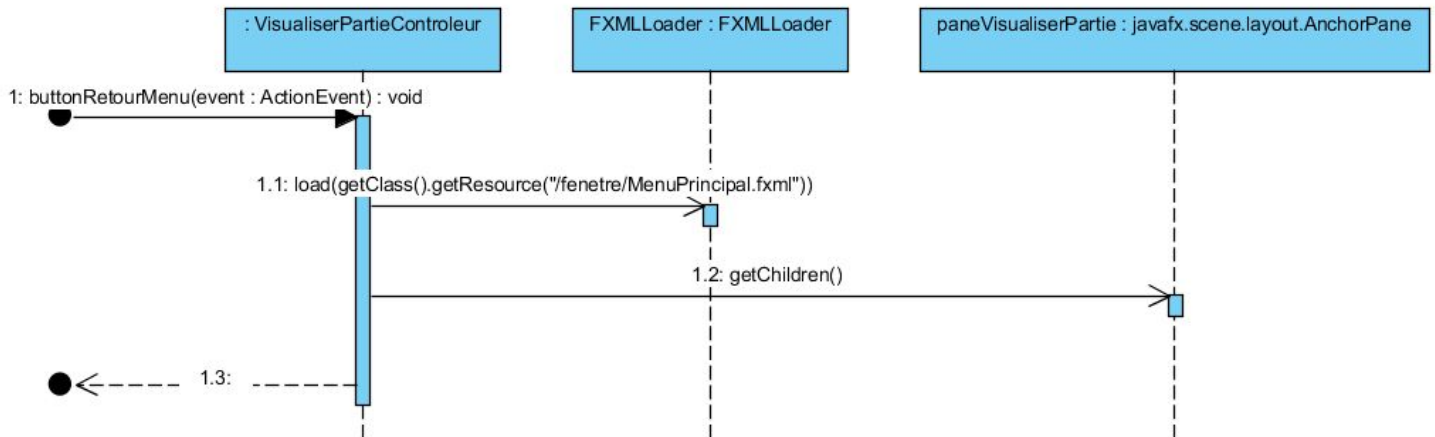
DS8 - Quitter le jeu

Lorsque l'utilisateur se trouve au menu principal, il a la possibilité de quitter le jeu. Lorsque l'utilisateur quitte le jeu, la fenêtre se ferme.



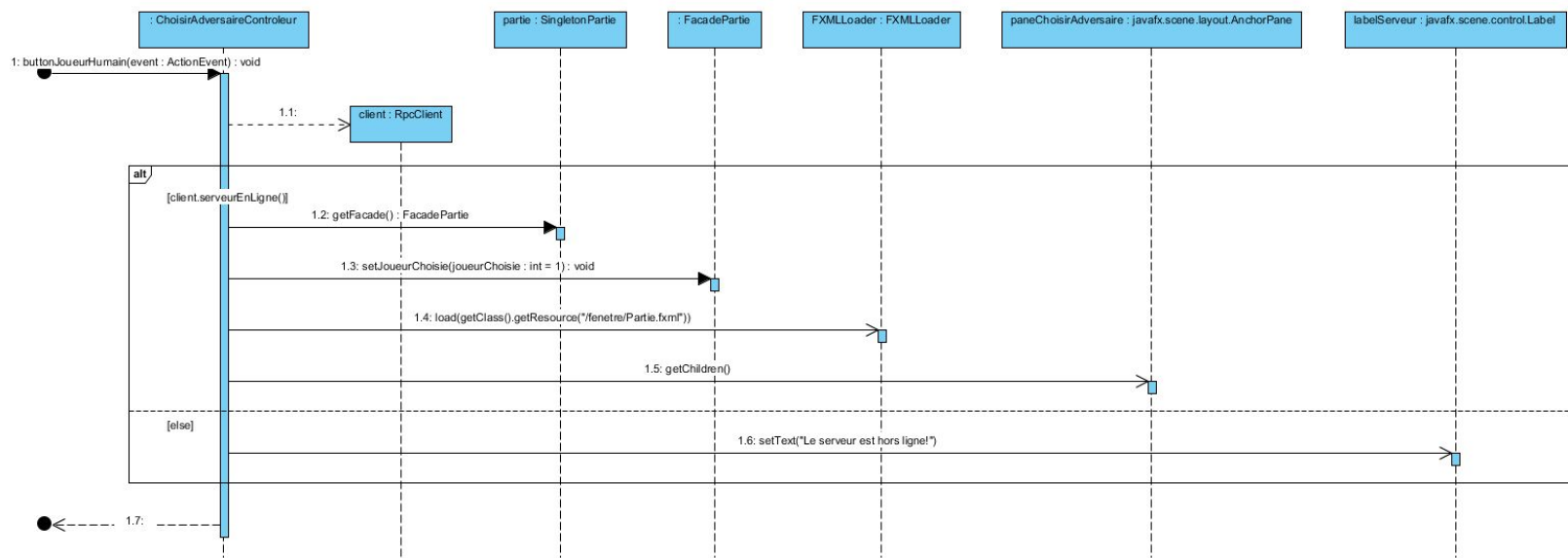
DS9 - Retourner au menu principal

Lorsque l'utilisateur se trouve sur la fenêtre de la partie d'échec, la fenêtre pour visualiser une partie ou la fenêtre pour consulter les meilleurs scores. L'utilisateur peut retourner au menu principal.



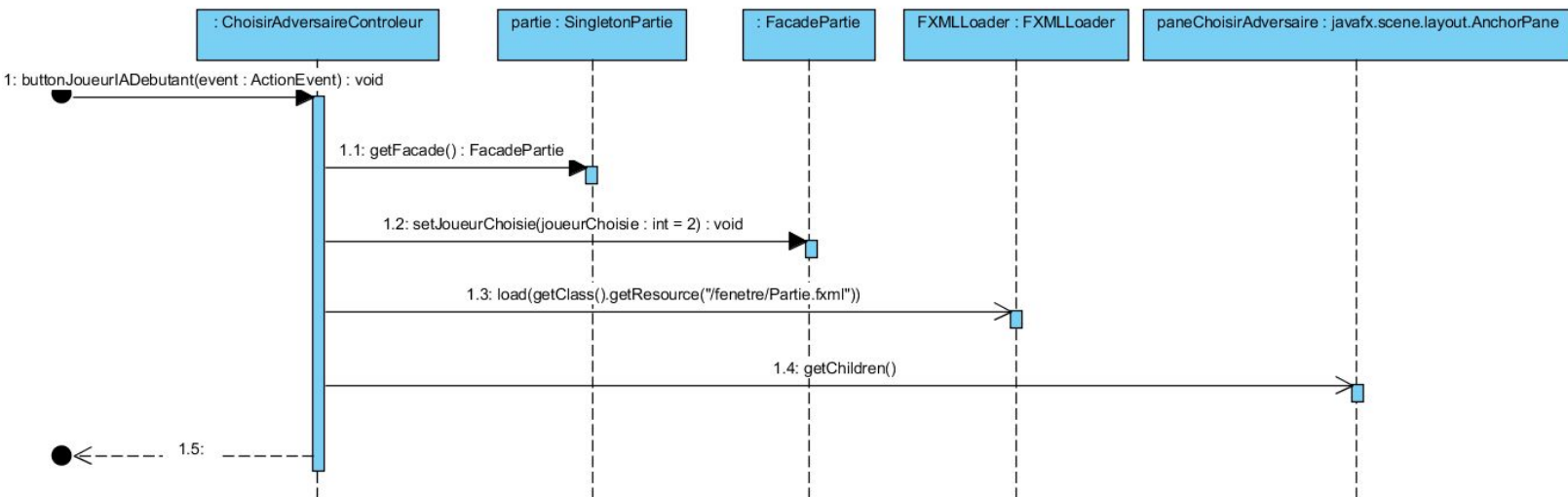
DS10 - Choisir de jouer contre un humain

L'utilisateur peut choisir de jouer contre un joueur humain, une IA au niveau débutant ou une IA au niveau avancé. Si l'utilisateur choisit de jouer contre un humain et que le serveur est en ligne, la partie commence et attend qu'un autre joueur soit attribué à notre partie. Le premier joueur attribué à une partie aura la couleur blanche et le deuxième joueur aura la couleur noire.



DS11 - Choisir de jouer contre une IA au niveau débutant

L'utilisateur peut choisir de jouer contre un joueur humain, une IA au niveau débutant ou une IA au niveau avancé. Si l'utilisateur choisit de jouer contre une IA au niveau débutant, la partie commence. La couleur attribué à l'utilisateur sera blanc et noir pour l'IA.



DS12 - Choisir de jouer contre une IA au niveau avancé

L'utilisateur peut choisir de jouer contre un joueur humain, une IA au niveau débutant ou une IA au niveau avancé. Si l'utilisateur choisit de jouer contre une IA au niveau avancé, la partie commence. La couleur attribué à l'utilisateur sera blanc et noir pour l'IA.

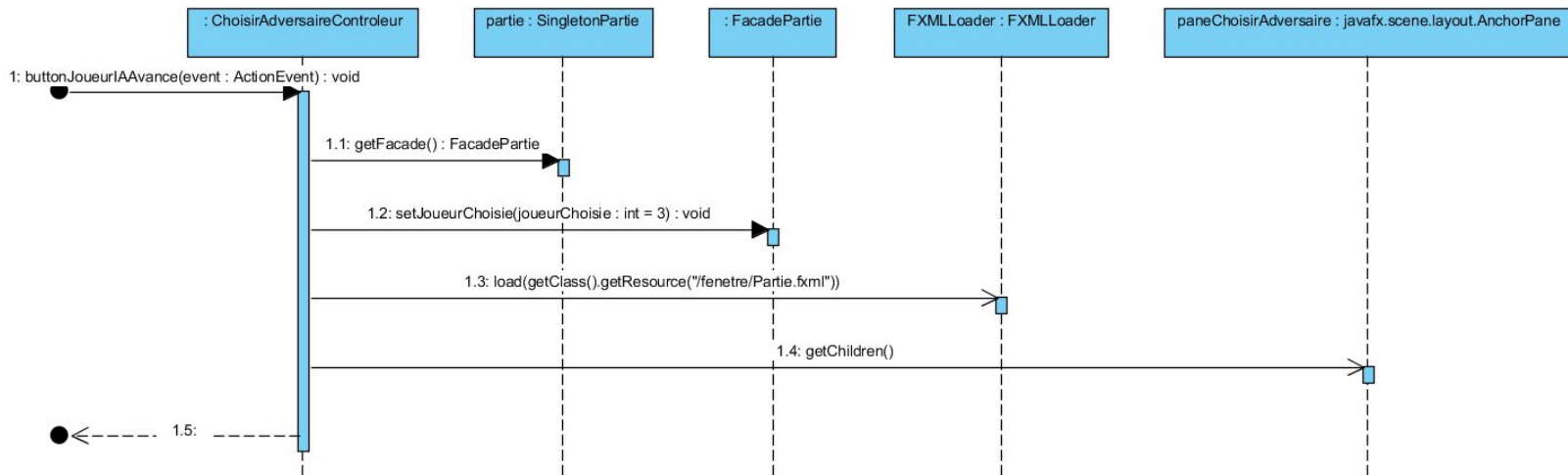


Diagramme de packages

Le diagramme suivant établit les packages du système et leurs liens. Pour le développement du logiciel, nous utilisons l'architecture en couche détaillée. Chaque couche du diagramme de package représente en ordre: la présentation, l'application, le domaine, les services techniques et la fondation. Les classes sont indiquées pour chaque package.

Diagramme de packages - TP2

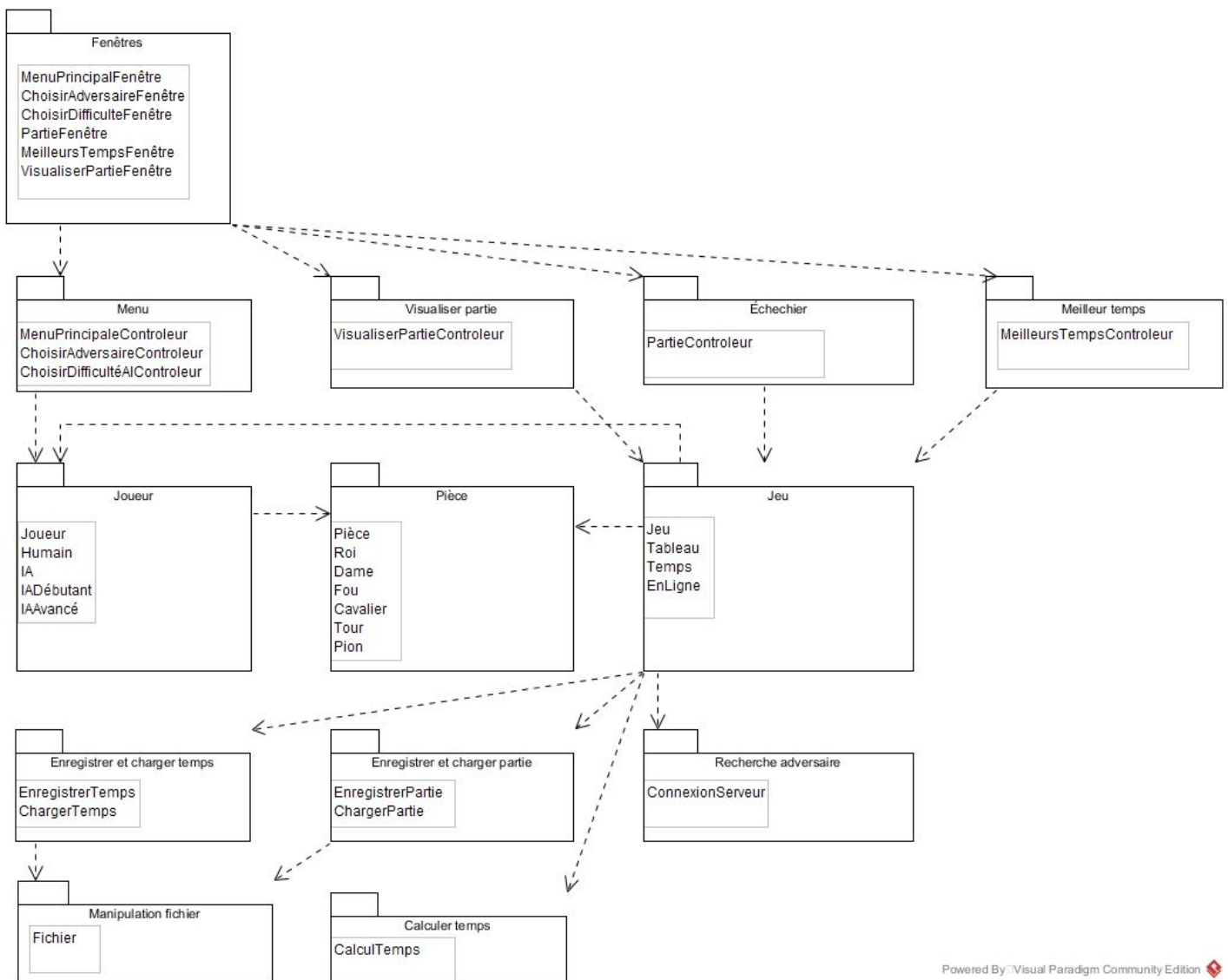


Diagramme de packages - TP3

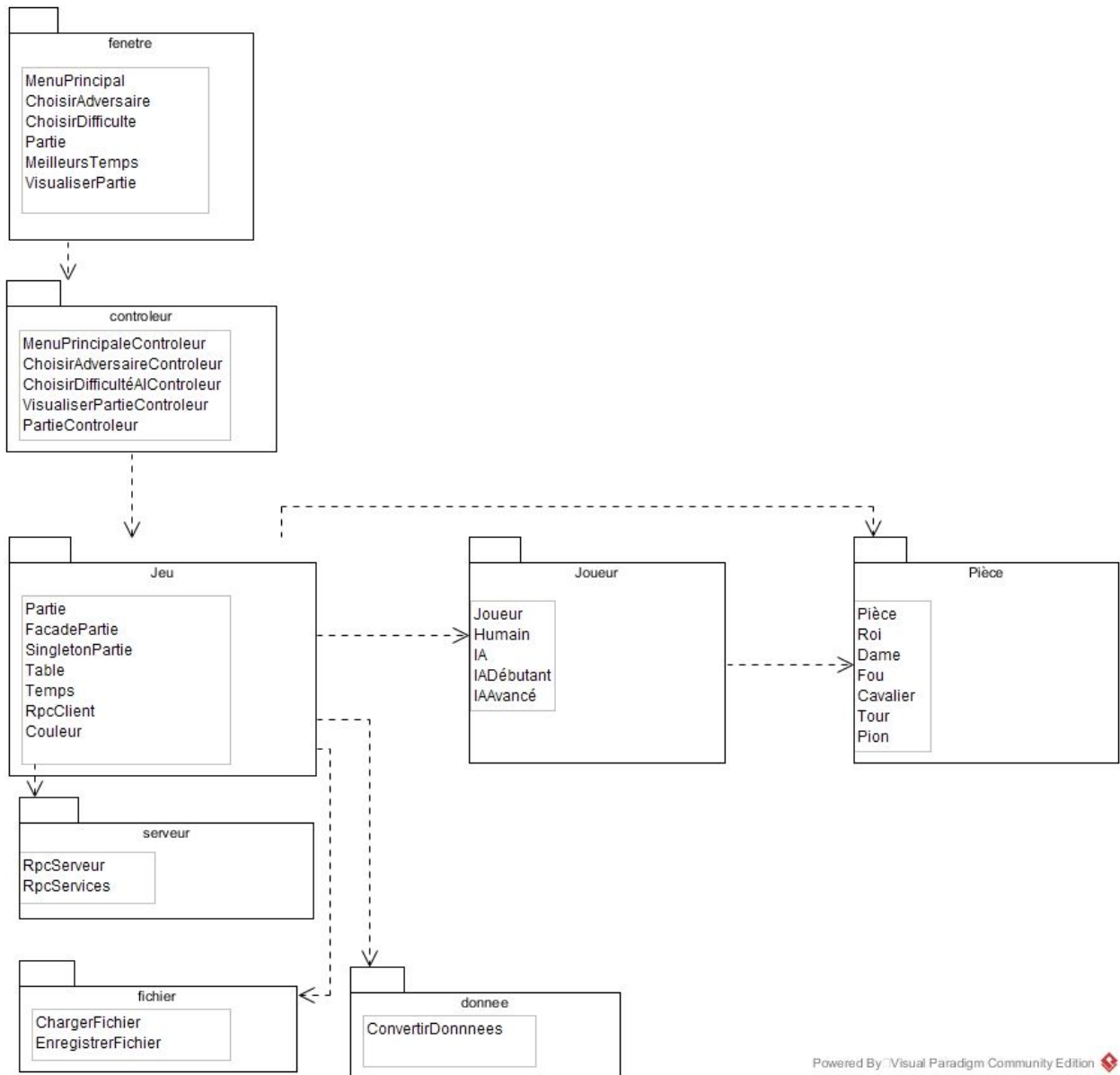


Diagramme de classes

Le diagramme de classe illustrant la conception détaillé d'un jeu d'échec. Un utilisateur peut jouer une partie d'échec contre un joueur humain en établissant une connexion à un serveur ou jouer contre un joueur artificiel localement. Le diagramme est composé des classes des couches: application, du domaine, services techniques et fondation.

Diagramme de classes - TP2

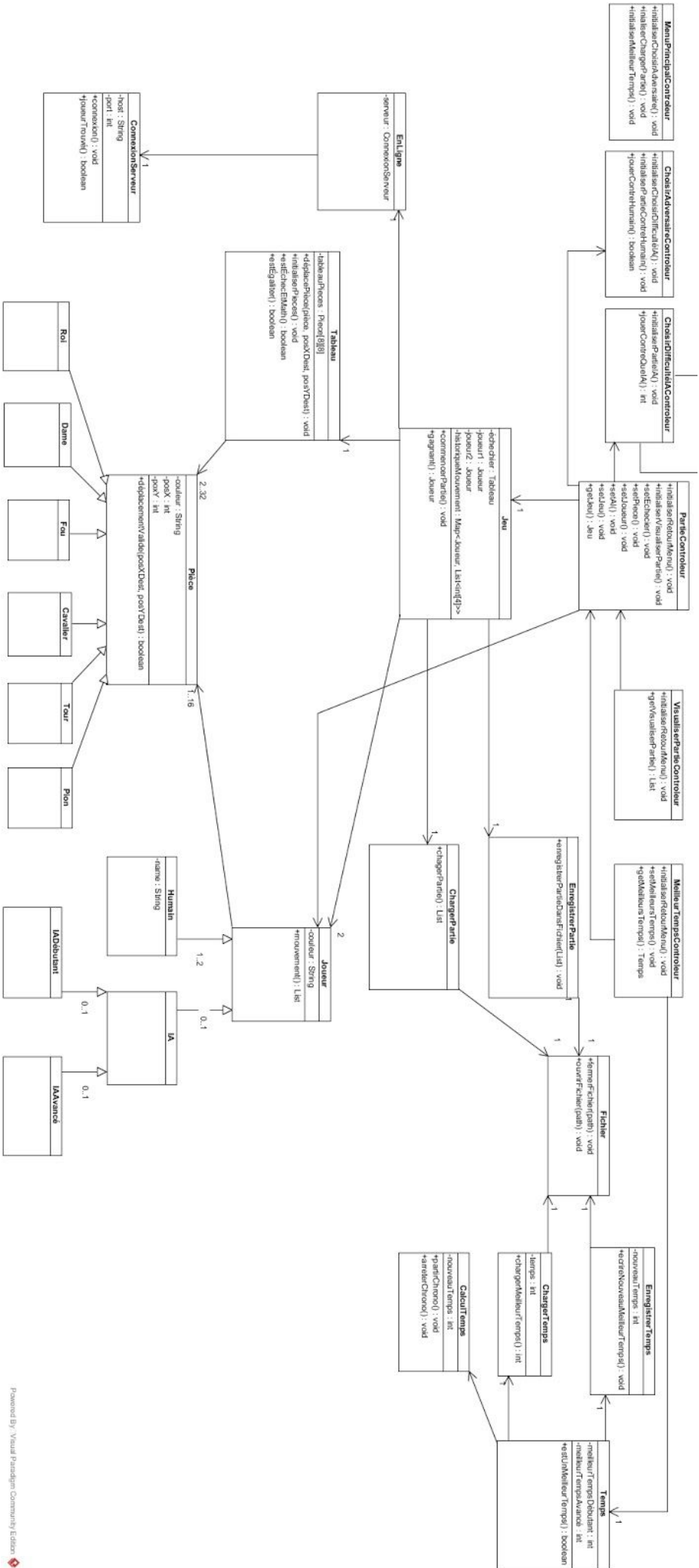
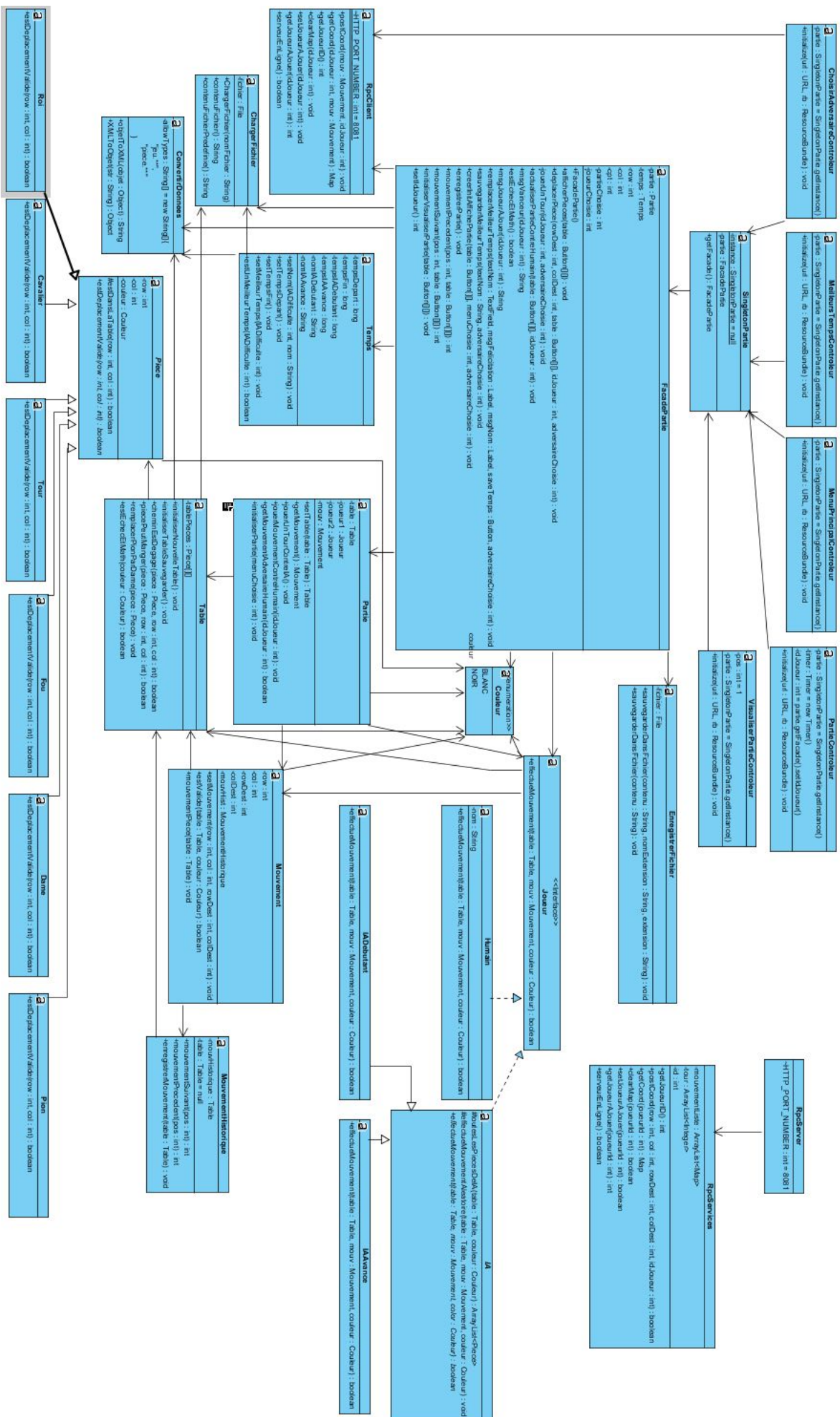


Diagramme de classes - TP3



Patron Singleton

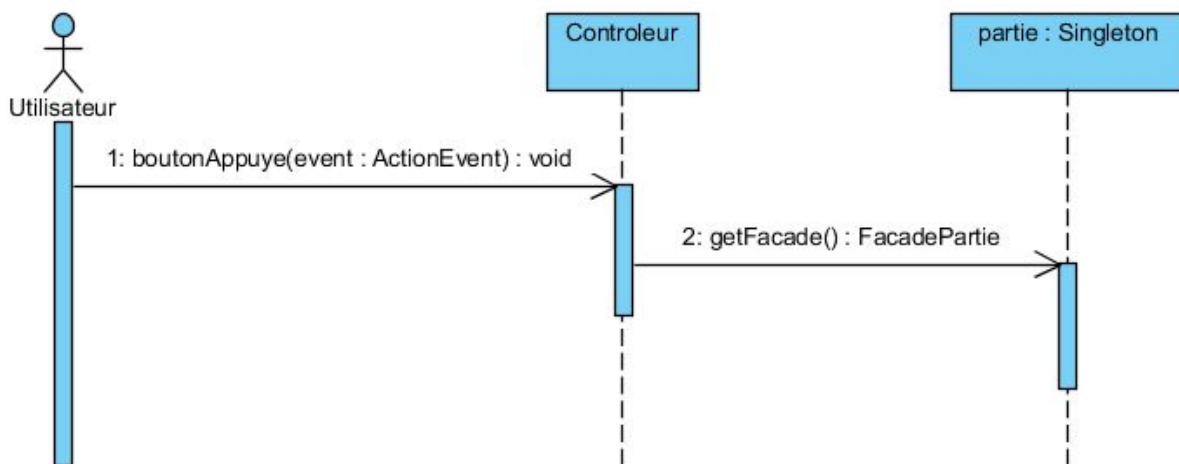
La classe SingletonPartie permet de s'assurer que l'objet FacadePartie ne possède qu'une seule instance dans le jeu d'échec et fournit un point d'accès global à cet objet.

Permet aux controleurs (Par exemple: PartieControleur et VisualiserPartieControleur) d'utiliser le même objet FacadePartie sans avoir à utiliser une variable globale.

Diagramme de classes



Diagramme de séquences



Patron Façade

La classe FacadePartie permet de créer une seule interface simplifiée pour faciliter la manipulation d'un ensemble de fonctionnalités se retrouvant dans les classes : Joueur, Partie, Temps, RpcClient, ChargerFichier, EnregistrerDonnees, ConvertirDonnees et Couleur.

Diagramme de séquences



Différence entre l'implémentation et la conception

1- Dans la conception on y retrouvait les packages "Enregistrer et charger temps" et "Enregistrer et charger partie". Les classes de ses packages utilisaient du code dupliqué et non nécessaire. Le package "Manipulation fichier" dans la couche fondation est suffisant.

2- La classe "Fichier" se trouvant dans le package de "Manipulation de fichier" a été remplacé pour deux classes. Une classe pour enregistrer une chaîne de caractères dans un fichier "EnregistrerFichier" et une autre pour charger dans une chaîne de caractères le contenu d'un fichier "ChargerFichier". Cela permet d'augmenter la cohésion.

3- Lors de la conception, nous n'avions jamais travaillé avec un modèle client serveur pour établir les parties entre joueurs au travers d'un serveur. Au niveau de la couche des services techniques, on y retrouvait le package "Recherche adversaire" qui a été renommé pour "serveur". Un nom qui est plus approprié. Nous avons dans ce package une seule classe "ConnexionServeur" qui a été remplacé lors de l'implémentation par deux classes. La classe "RpcServer" qui consiste à démarrer le serveur sur le port fournis et donner accès aux services de la classe "RpcServices". En effectuant cette séparation de classe, la cohésion augmente.

4- Lors de l'implémentation, deux nouvelles classes ont été ajoutées suite aux exigences. Ses deux nouvelles classes implémentent chacune d'eu un patron Gang of Four, le patron Singleton et le patron Façade. Les bénéfices de l'ajout de ses patrons sont expliqués dans leur onglet respectif.

5- La classe "Jeu" a changé de nom pour "Partie", un nom plus significatif.

6- La classe "EnLigne" a changé de nom pour "RpcClient", un nom plus significatif.

7- Étant donné du peu de contrôleurs qu'on retrouve dans le projet, nous avons fusionné tous les packages des contrôleurs en un seul.

8- Lors de l'implémentation, l'utilisation des couleurs des pièces du jeu d'échec (blanc et noir) arrivent de façons régulière pour être implémentées ou pour effectuer des validations . Au lieu d'utiliser une chaine de caractère à chaque fois, nous avons créé une classe d'énumération "Couleur" pour corriger ce problème.

9- Désormais, les contrôleurs ne s'appellent pas entre eux. Ils ont seulement la méthode "initialize()" de public et appellent la façade pour effectuer un échange de données avec le logiciel. Cela permet d'éviter des variables globales et de diminuer le couplage.

10- La classe "Mouvement" a été créée lors de l'implémentation pour affecter les coordonnées de départs et d'arrivées d'une pièce. Cela permet de diminuer le nombre de variables qu'on passe en paramètre à nos méthodes qui utilisent les coordonnées. Ses coordonnées vont être validées dans cette classe. Si le mouvement est valide, un joueur ou une IA va utiliser la méthode "mouvementPiece()" pour faire le déplacement d'une pièce.

11- La variable "historiqueMouvement" qui se trouvait dans la classe "Jeu", lors de la conception, a été remplacée par sa propre classe "MouvementHistorique". Cela permet d'améliorer la cohésion.

12- Dans les classes du package "joueur", la méthode "mouvement()" a été remplacée par "effectuéMouvement()", un nom plus significatif.

13- Nous avons ajouté la classe "ConvertirDonnees" pour convertir un objet en une chaîne de caractère au format XML ou bien convertir une chaîne de caractères au format XML en objet. Cette classe est utilisée pour la conversion de l'objet temps ou le tableau de pièces.

14- Des méthodes ont été rajoutées dans diverses classes pour satisfaire aux exigences non pensées lors de la conception.

Glossaire

- IA : Une intelligence artificielle.
- Menu: Un menu est une fenêtre constituée seulement de boutons.