

Prototype projet A.F.J.D. pour le jeu de Diplomatie

But:

Ce document présente l'architecture du projet et se focalise sur la partie permettant de jouer.

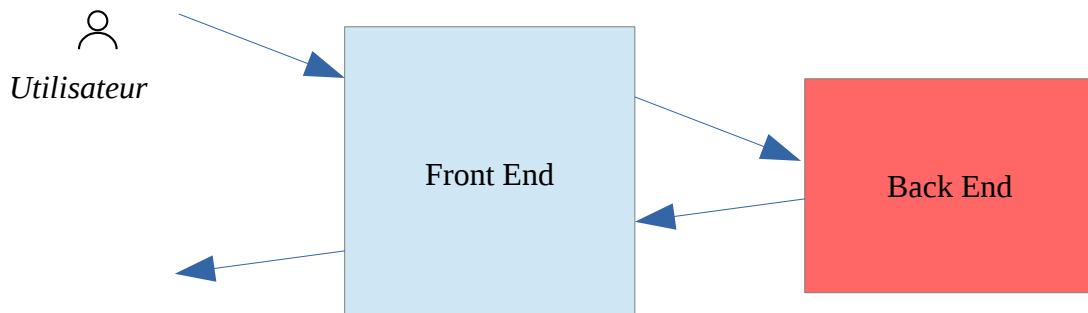
La partie « Front-End »

- ✗ est proto-typée en Python et s'appuie sur Tkinter (Tkinter est un port Python vers Tcl)
- ✗ le prototype fonctionne sous LINUX et WINDOWS pour une utilisation sur PC.
- ✗ est à refaire par les étudiants :
 - pour un client web (navigateur)
 - pour un client Android
 - (éventuellement) pour un client de type Iphone

La partie « Back-end »

- ✗ est réalisée en Python et s'appuie sur Flask (Flask RestFul, Flask Mail, Flask JWT Extended) pour le côté serveur et « request » pour le côté client .
- ✗ ne sera à priori pas à refaire par les étudiants (du moins dans un premier temps).

Vue d'ensemble simplifiée



Différence très importante :

- Un composant « Front-End » est susceptible d'être altéré à des fins malicieuses
- Un composant « Back-End » est hébergé sur un serveur contrôlé par l'association, il ne peut donc pas être altéré à des fins malicieuses (en principe!)

Il faut donc toujours garder cet aspect à l'esprit lors de la conception des composants.

Composants « Front End »

Ce sont en réalité des « I.H.M »

La création de compte (peut être limitée au client Web)

Ce composant permet à utilisateur de créer son compte, préalable à toute activité répétée sur le site.

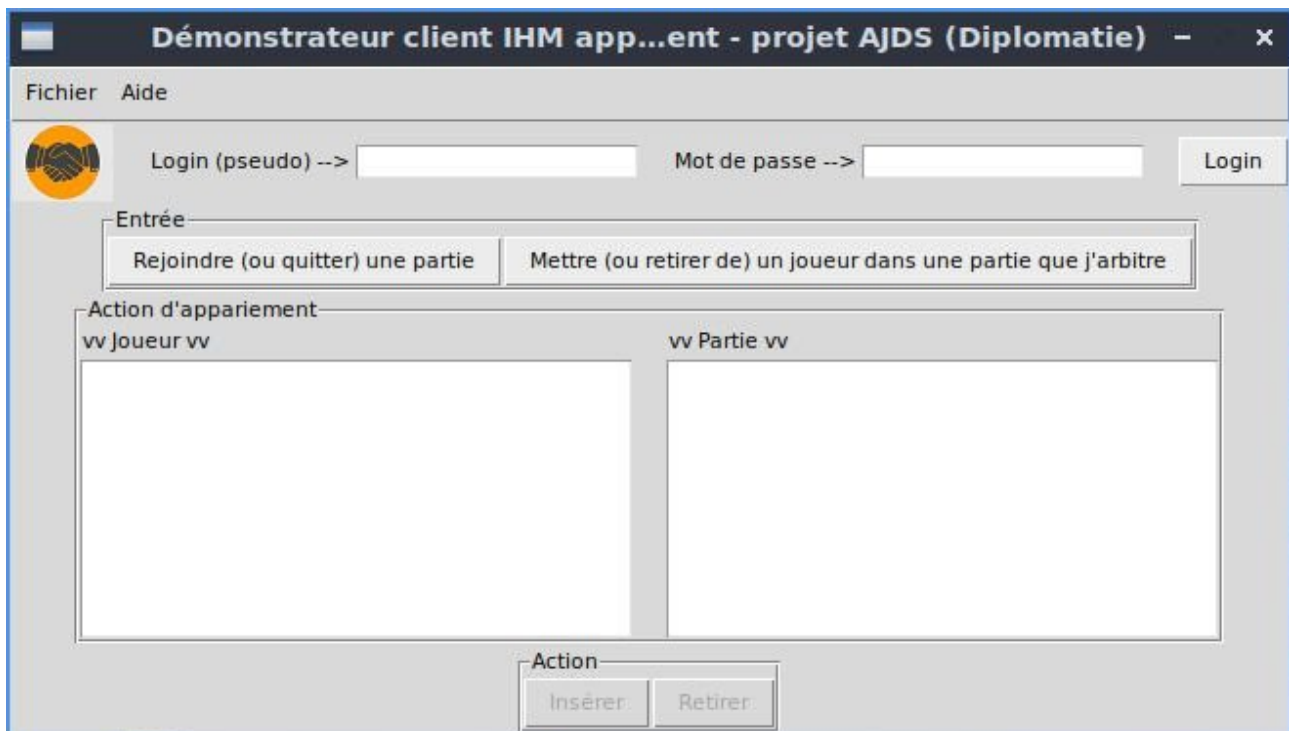
La création de partie (peut être limitée au client Web)

Ce composant permet à un utilisateur inscrit de créer une partie de Diplomacy/Diplomatie (ou d'une variante)

L'appariement (peut être limitée au client Web)

Ce composant permet à un utilisateur inscrit de lier un utilisateur à une partie. Il n'est pas question pour le moment d'allouer un rôle donné dans la partie, mais juste de regrouper des utilisateurs dans des parties

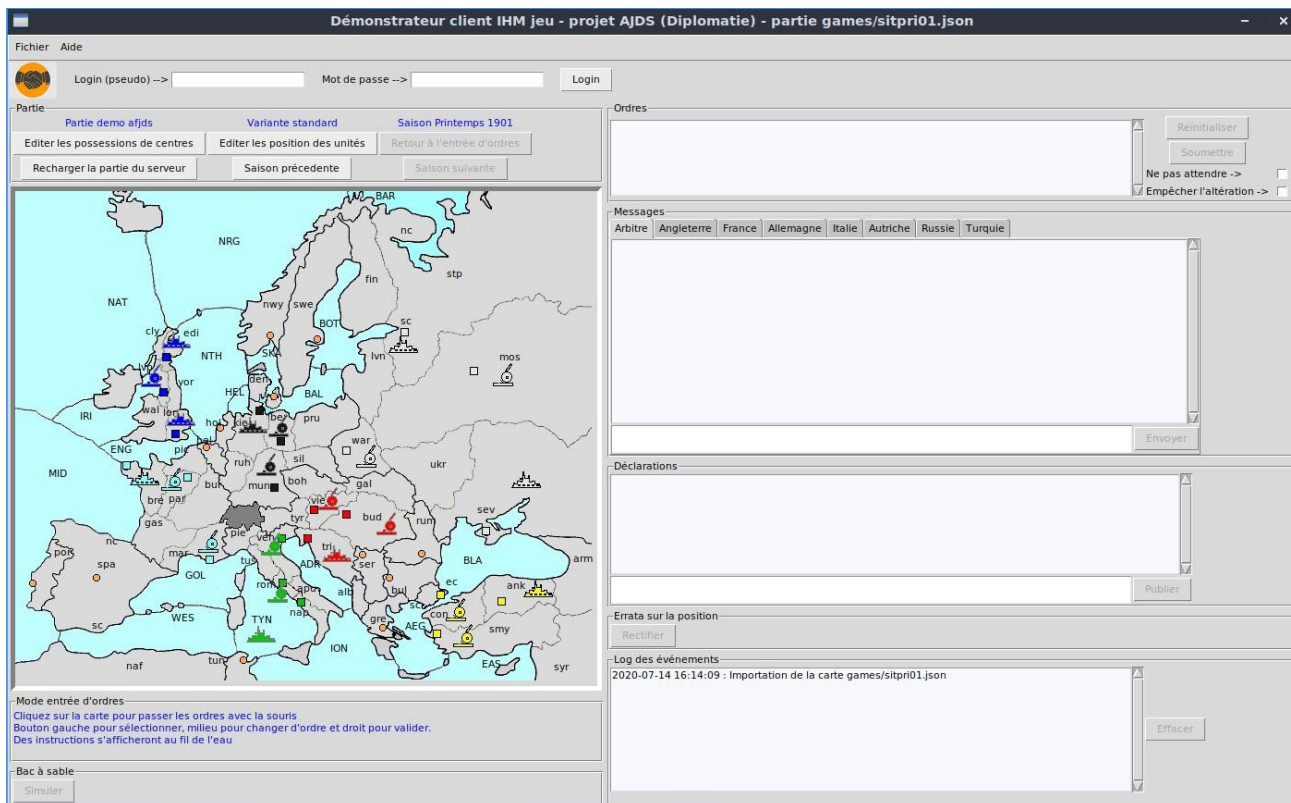
À noter : non indispensable pour commencer mais très pratique, il faudra prévoir un affichage de l'avancement des appariements.



Le jeu

Ce composant est le plus complexe de tous, il permet de jouer à Diplomatie ou à une de ses nombreuses variantes.

Il devra être disponible sous un maximum de plate-forme, et constitue le cœur du système.



Composants « Back End »

Ce sont, sauf exception, en réalité des serveurs REST. Pour information, ils s'appuient sur une base de données sqlite3. Les tables sont de deux sortes :

- tables « objets » pour lesquelles une rangée correspond à un objet (exemple : un compte utilisateur). Chaque objet possède un identifiant numérique unique.
- tables « relations » qui servent à relier des objets entre eux (exemple : l'appartenance à une partie). Pour ce faire, l'identifiant des objets est utilisé.

Le gestionnaire de parties

Il sait créer une partie, il sait la faire avancer.

Implémentation : quasi complète

Reste à faire : Le démarrage de la partie avec toutes les tables en base de données à remplir.

Le gestionnaire de comptes utilisateurs

Il sait créer un compte utilisateur. Il vérifie l'adresse de l'utilisateur par le biais d'un code de 4 chiffres que l'utilisateur doit saisir pour prouver qu'il a bien reçu le mail.

Implémentation : quasi complète.

Reste à faire : L'aspect récupération de mot de passe oublié. Cet aspect est de loin le plus délicat, car il est susceptible de compromettre la sécurité du site si mal conçu.

Le module identification/authentification (non REST)

Ce module est une brique fondamentale de cette architecture.

Il sait créer et conserver un couple pseudo/mot de passe (le mot de passe n'est pas conservé en clair)

Il sait fournir un jeton lorsqu'un composant lui présente un pseudo/mot de passe qu'il reconnaît

Il sait authentifier un jeton qui lui est présenté.

Il logue soigneusement dans un fichier les événements qui se produisent.

Implémentation : quasi complète, à base de flask-jwt-extended .

Reste à faire : Le renouvellement du jeton n'a pas encore été pris en compte (pour ne taper son mot de passe qu'une fois par semaine au lieu d'une fois toutes les 15 minutes).

IMPORTANT : Dans cette architecture, les modules « sociaux » devront s'appuyer sur ce module pour vérifier l'identité des inscrits. Il est donc primordial de s'assurer que la solution retenue pour les modules « sociaux » permet de le faire.

Le module résolution

Ce module est la brique la plus complexe de cette architecture. Il est extrêmement improbable qu'il soit demandé aux étudiants de le réaliser.

Il sait réaliser une résolution pourvu qu'on lui présente :

- une situation,
- des ordres.

Il renvoie alors :

- un statut,
- des ordres tagués,
- une nouvelle situation le cas échéant.

Implémentation : non commencée, sera vraisemblablement basée sur une alternative parmi les suivantes :

- un logiciel réalisé en Java par Romain Parmantier
- un logiciel réalisé en Langage C par Jérémie Lefrançois
- un logiciel récupéré sur Internet

Dans tous les cas de figure, le logiciel utilisera la méthode de résolution « D.A.T.C ».

Reste à faire : « sur couchage » de solveur existant. (le solveur en lui-même ne sera pas développé dans un premier temps, voire – c'est le plus vraisemblable — pas du tout).