Projet Intégrateur 2022
Cahier des charges

Équipe 1A

Janvier 2022

Sommaire

1 Présentation et contexte du projet

1.1 Équipe

L'équipe de notre projet est composée de 12 étudiants en L3 :

- ALAHMAD Rawan
- AMODEO Mélissa
- BAKIR AGHA Vincent
- BIACHE Yaël
- BODELE Ladislas
- CHIKRI Intissar
- CHRISTMANN Emmanuel
- DEROUSSEAUX–LEBERT Nathanaël
- DIABY Ibrahim
- DIAKITE Mariam
- DIATTARA Amadou
- DIEBOLT Loïc

1.2 Problématique

Dans le cadre de l'UE **Projet Intégrateur**, notre équipe est chargée de **revisiter un jeu de plateau** de manière à le rendre **multijoueur** en réseau.

Le jeu sera dans un premier temps développé sur des clients lourds interagissant avec un serveur central. Dans un second temps, il sera adapté pour être jouable sur matériel mobile sans fil et/ou via une page web.

L'authentification, la sauvegarde des scores, les données de jeu, seront réalisées via une base de données sur le serveur central.

1.3 Situation actuelle

Notre équipe de développement est composée d'étudiants ayant déjà utilisé certains outils, langage ou framework, que nous allons utiliser pour ce projet.

Le jeu que nous avons choisi est **Diplomatie**, un **jeu de stratégie militaire** orienté sur le **dialogue entre les joueurs**.

Chef de projet	
CHRISTHMANN Emmanuel	
Backend	**Frontend**
DEROUSSEAUX–LEBERT Nathanaël (Chef d'équipe)	BAKIR AGHA Vincent (Chef d'équipe / Designer en chef)
DIAKITE Mariam	ALAHMAD Rawan
BODELE Ladislas (Secrétaire)	BIACHE Yaël
CHIKRI Intissar	AMODEO Mélissa
DIATTARA Amadou	DIEBOLT Loïc
	DIABY Ibrahim

Rôle du chef de projet :

Le chef de projet suit et **supervise** le projet, **planifie** les réunions et **organise** le temps de travail pour les 2 équipes, il peut également **aider** à la réalisation du projet si besoin.

Le chef de projet ne doit pas nécessairement être l'expert du groupe. Le codage n'est pas sa priorité. Il doit en effet pouvoir **fixer les objectifs** à atteindre par l'équipe, gérer la **communication** interne ou externe, veiller à ce que tout soit fait dans les temps et pouvoir **adapter** le planning de l'équipe dans le cas contraire.

En plus d'être un **chef d'orchestre** ayant une bonne connaissance du projet, il doit être un **leader charismatique** qui sait trouver des compromis, **motiver** son équipe et qui veille à ce qu'une bonne entente et un climat sain règne dans le groupe.

Mot du chef de projet :

Je voudrais que les équipes **travaillent ensemble**, que les membres s'**entraident** dans chaque équipes et que les équipes de Front et de Back puissent s'**échanger des membres**, je veillerais à la bonne réalisation du projet sur ce sujet que j'ai lancé et dont j'ai eu l'idée. La **répartition du travail** entre chaque membre sera équitable et la Charge horaire par personne sera de 4h de travail en groupe obligatoire + 8h de travail personnel par semaine à partir du 31/01.

1.4 Les objectifs de l'application

- Modéliser le jeu de plateau Diplomatie avec une interface graphique;
- Un chat avec des canaux (général, privé, spectateur);
- La résolution automatique des ordres par l'application;
- Avoir une expérience utilisateur agréable.

2 Design de l'application

Nous utilisons Figma et Bubble pour définir notre première charte graphique, celle-ci est en **constante amélioration**.

Interface de lancement d'une partie :

Figure 1: Inspiration pour le design de notre interface de lancement de partie sur ordinateur

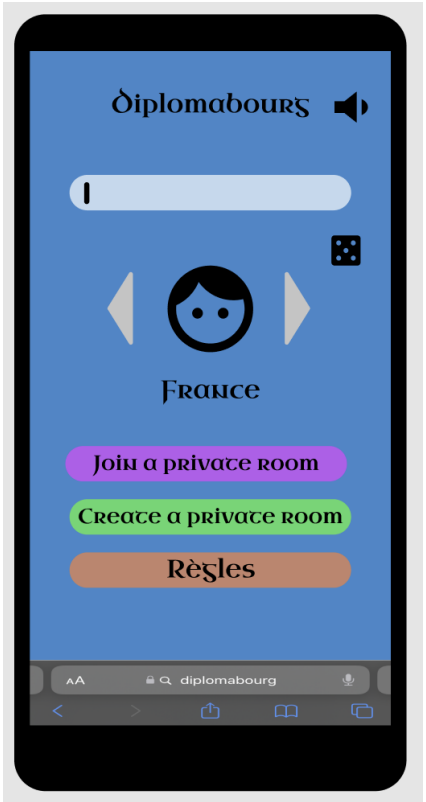

Figure 2: Première idée d'interface acceuil, version mobile

Figure 3: Première idée d'interface du salon d'attente des joueurs, version mobile

Interface de jeu :

Figure 4: Interface de jeu, version mobile

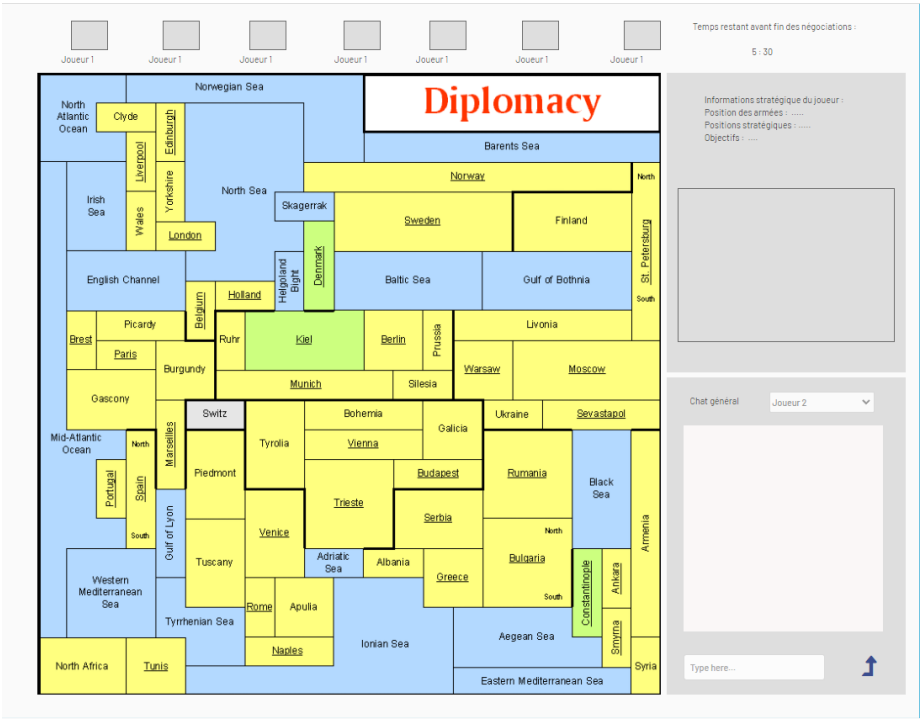

Figure 5: Interface de jeu, version ordinateur

3 Fonctionnalités et spécifications techniques

3.1 Fonctionnalités

Nous découpons les fonctionnalités par phase de jeu et par interface / objets.

3.1.1 Phase 1 : écran d'accueil

Lancement de l'application

Au lancement de l'application le logo du jeu s'affiche pendant une petite période de temps, puis la page d'accueil s'affiche.

Accueil

- Carte du jeu en fond
- Nom du jeu
- Formulaire avec choix du pseudo, de la couleur et d'une icône
  Contraintes pour le choix du pseudonyme:
  - Le pseudo ne doit pas être une chaîne vide.
  - Il doit contenir au moins une lettre de l'alphabet latin.
- Musique d'ambiance

- Bouton pour créer un salon
- Bouton pour rejoindre un salon
- Bouton qui indique les règles du jeu

Salon d'attente du début de la partie
Interface :

- Paramètres du jeu: langues, temps de dialogue, nombre de joueurs
- Génération d'un lien
- Espace d'attente pour que les joueurs soient au complet et quelle puissance ils vont jouer
- Lancement du chat en attendant le début de la partie

Objets :

- Page d'accueil

Rejoindre la partie

- Si le joueur n'a pas de données sauvegardées il est redirigé vers le formulaire de création de joueur.
  - il rejoint le salon si le formulaire est complet et valide

3.1.2 Phase 2 : Début du jeu

Début du jeu
Au lancement d'une partie, il y a entre 2 et 7 joueurs qui ont un interface personnelle. Après le lancement de la partie, aucun nouveau joueur ne sera accepté.

Interface :

- Bouton pour les options
  - Rappel des règles du jeu
  - Modification du son
  - Langues
- Chronomètre pour chaque phase d'une partie (diplomatique et tactique)
- Carte du jeu
- Chat avec canaux de discussion

Objet :

- Génération de la carte, positionnement des unités de chaque joueur

3.1.3 Phase 3 : Jeu

Chat général, privé et spectateur

- Le chat général se sert de 3 canaux de discussion :
  - Le canal général permet de parler avec tous les joueurs en même temps, les spectateurs peuvent voir ce canal, mais pas interagir avec.
  - Le canal privé permet d'envoyer des messages directs à un ou plusieurs joueurs.
  - Le canal spectateur permet de communiquer avec tous les spectateurs en même temps, les joueurs ne peuvent pas voir ce canal.
- En cliquant sur la bulle d'un joueur, le message cible ce joueur. Si l'on clique sur plusieurs bulles de joueur, le message sera envoyé aux joueurs sélectionnés.
- Les canaux peuvent être filtrés pour afficher un certain type de message.

Sons d'ambiance

- Musique de fond;
- Message reçu;
- Nouveau tour;
- Victoire et défaite.

Phases de jeu
Phase diplomatique :

- Durée de la phase : 10min
- Discussion entre les joueurs grâce au chat

Phase tactique :

- Durée de la phase : 5min
- **Déroulement :**
  1. Rédaction des ordres
  2. Résolution automatique des ordres
  3. Retraite des unités
  4. Recrutement d'unités

Ordres
Ordres aux armées :

- Attaquer
- Tenir
- Tenir et soutenir sur place

- Tenir et soutenir une attaque

Ordres aux flottes :

- Attaquer
- Tenir
- Tenir et soutenir sur place
- Tenir et soutenir une attaque
- Tenir et convoyer

3.1.4 Phase 4 : Fin du jeu

Fin du jeu :
Interface :

- Classement des joueurs
- Historique des chats
- Récapitulatif de fin de partie

Objets :

- Affichage des informations de la partie fini

3.2 Technologies et architecture logicielle

3.2.1 Architecture logicielle

Afin de respecter la séparation Client/Serveur, notre frontend mettra en forme les données renvoyées par le backend. Il n'aura aucun calcul à faire et toutes les règles seront **implémentées dans le backend**. Lorsqu'une nouvelle donnée sera disponible, le backend mettra au courant le frontend au moyen d'un **websocket**.

Le backend et le frontend seront **déployés sur un serveur Apache**, hébergé sur un raspberry.

3.2.2 Frontend

Pour la partie frontend, nous allons utiliser le langage JavaScript couplé avec le **framework Vue.js**. Il nous a semblé judicieux car il correspond en tout point à notre projet, étant donné que son architecture est **adaptable incrémentalement**.

Du côté performance, il est **très léger** par rapport à d'autres framework : environ 20Kb, contre 43Kb pour React et 143Kb pour Angular. Il est très facile d'approche car ce sont des technologies connues par l'équipe de développement (HTML et JavaScript), ce qui permettra un gain de temps considérable au niveau de la phase d'apprentissage. Malgré une communauté plus petite que celle des autres, Vue.js dispose d'une documentation très détaillée.

Au niveau développement, de nombreux outils sont fournis avec le framework, permettant un développement rapide et efficace. En effet, un des outils permet de générer un projet, ce qui nous fera gagner du temps sur la **configuration initiale du projet**. Il y a également une intégration des outils pré-configurés avec des valeurs par défaut pour que l'on puisse directement se concentrer sur l'application en elle-même. Vue.js simplifie le **débogage de l'application** en prenant en charge les tests unitaires.

3.2.3 Backend

Pour la partie backend, nous avons choisi d'utiliser le langage Python avec le **framework Pyramid**. Il fait partie des frameworks python les plus populaires en open source. On a opté pour Pyramid puisqu'il est plus **flexible** que Django, mais également parce qu'un membre de l'équipe a déjà de l'expérience. Son point fort est sa capacité à bien fonctionner avec de petites et grandes applications. De plus, il s'agit du framework le plus rapide connu. Il permet aussi de prendre en charge la génération d'URL, élément dont on aura besoin dans notre projet. Le framework est simple à apprendre, avec une documentation riche et précise ainsi qu'une large communauté active. On peut aussi décider du langage du template, des bibliothèques et de la couche de base de donnée.

On utilisera les **bibliothèques Cornice et Marshmallow** pour donner à notre API une **architecture REST**. Évidemment, comme notre application demandera du temps réel, elle ne pourra pas être REST-full car elle implémente des websockets. Cette contrainte nous à aussi orienté dans le choix du framework, car Pyramid **supporte nativement les websockets**.

3.2.4 Gestion du projet

On utilisera gitlab pour la **gestion des sources**. Google Drive nous servira à échanger des documents, et à travailler dessus en collaboration. Discord nous servira d'**espace d'échange**, car il permet de créer plusieurs salons organisés en thématiques. Ce qui est primordial pour travailler en collaboration dans une équipe de cet effectif.

3.3 Les contraintes

3.4 Contraintes générales

- Le lien partagé reste actif jusqu'à la fin de la partie
- Au début de la partie le nombre de joueurs est défini et limité; Si il y'a des clients supplémentaires ils auront un statut de spectateur
- Lorsque qu'un joueurs quitte la partie il y'a 2 cas possible:
  - Si 0 visiteurs le joueurs est en mode désordre civil;
  - Si il y'a au moins un visiteur, le premier spectateur peut choisir de remplacer le joueur parti
- Jeu non jouable hors connexion

3.5 Sécurité

- Chiffrement des flux
- Antispam
- Analyse de logs

4 Versions

4.1 Versions Alpha

- 1er acceuil:
  - Bouton création de partie
- 2eme acceuil (aprés création de partie):
  - liste des joueurs présent
  - bouton lancement de partie
  - lien pour rejoindre le salon de partie
- Instance de jeu:
  - bouton options
  - timer
  - carte europe avec les différentes faction
  - Interactions minimales avec le plateau de jeu
- Fin de partie:
  - Pop-up du gagnant de la partie

4.2 Versions Béta

- 1er acceuil:
  - Musique de fond avec bouton On/Off
  - Choix du pseudo
  - Choix du pays avec un bouton au hasard
  - Bouton choix de map de jeu (Europe, Strasbourg)
  - Bouton création de partie
  - Bouton règles du jeu
- 2eme acceuil (aprés création de partie):
  - Musique de fond avec bouton On/Off
  - Liste des joueurs présent et choix de faction
  - Réglages (Langue, temps de dialogue, nombre de joueurs, choix dialogue Appli et/ou Discord)

  - Possibilité de spectateurs (Max: 2x le nbr de joueurs)(si désistement joueur remplacement de l'un des spectateurs en joueur)
  - Bouton lancement de partie
  - Bouton règles du jeu
  - Lien pour rejoindre le salon de partie et/ou génération QR code
  - Lien pour rejoindre le salon de discussion discord
  - Espace de dialogue entre joueur dans l'application
- Instance de jeu:
  - Bouton options
  - Timer entre les phases de jeu
  - Carte europe avec les différentes faction
  - Espace de dialogue joueurs
  - Résolution des ordres des joueurs de manière automatique et visuelle
- Fin de partie:
  - Pop-up du gagnant de la partie
  - Historique de la partie (Toutes les discussions affichées, déroulement de la partie)

5 Architecture système

6 Améliorations

Diplomatie est un jeu de plateau plutôt vieux, le revisiter pour en faire un jeu a naturellement demandé des **changements et améliorations**. Les améliorations de bases sont les idées qui définissent cette version du jeu. Tandis que les améliorations futures reflètent notre désir de faire un jeu unique et agréable.

6.1 Améliorations de base

- Partie en ligne privée, grâce à la génération d'un lien
- Carte de jeu de Strasbourg
- Chat avec canaux, général, privé et spectateur
- Pions modélisés et placés sur la carte
- Chronomètre des phases de jeu

6.2 Améliorations futures

- Différentes carte de jeu
- Créer sa propre carte de jeu
- Carte de jeu générée procéduralement
- Possibilité de jouer contre des IA
- Partie en ligne publique
- Partie rapide, les chronomètres de phases sont plus rapides
- Création de compte avec score générale du joueur et historique des parties
- Création de groupe de chat
- Lien restant actif avec des joueurs considérés comme visiteurs