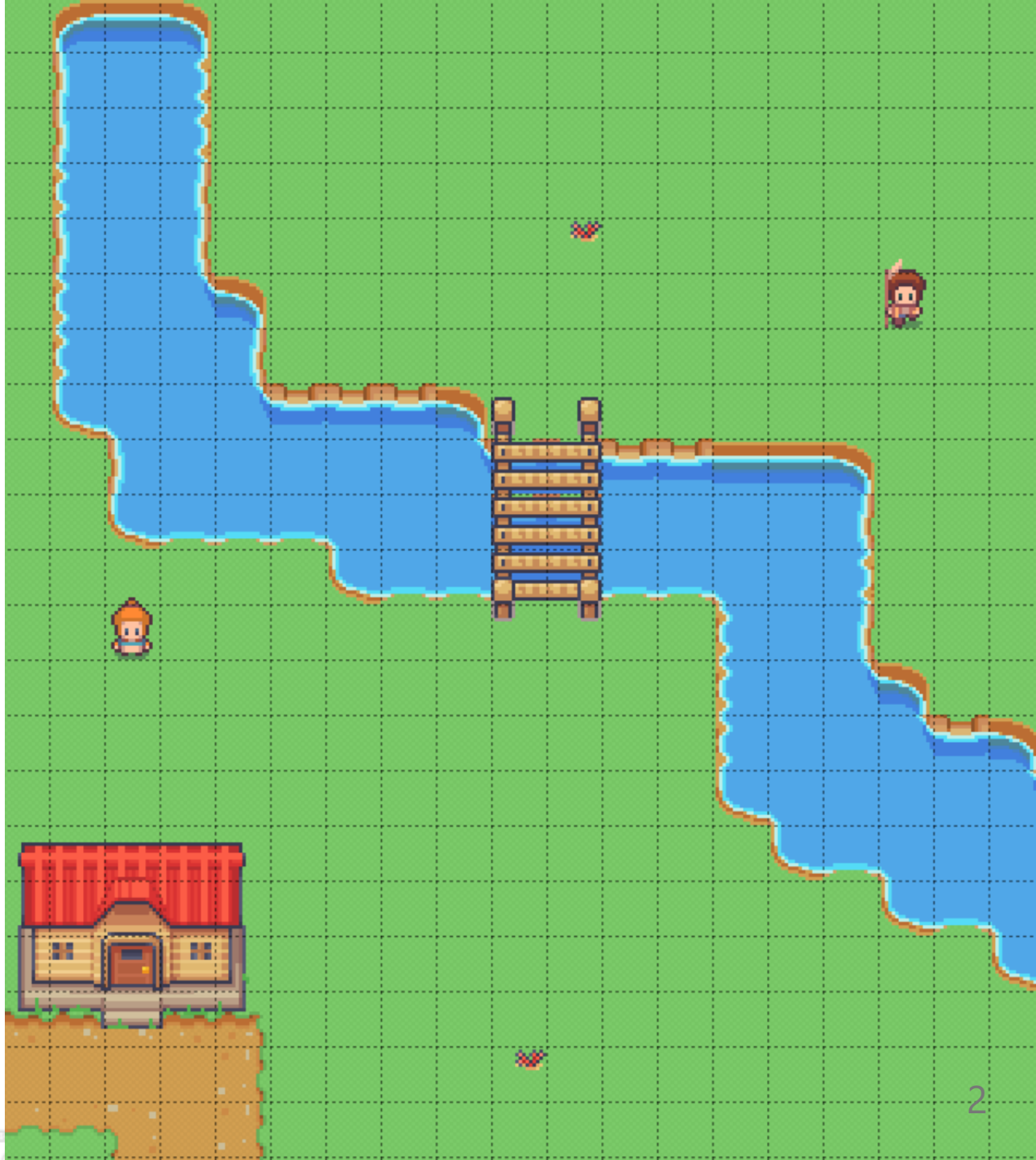
Masters 1 & 2 - Web Mobile - PWA

Projet fil rouge - Jeu au tour par tour

Par Julien SOSTHENE - 2020

Sommaire



1. 👨‍🎓 Objectif pédagogique 👩‍🎓

Le but de ce projet suivi est de valider les compétences acquises pour la mise en place d'une Progressive Web App complète:

- Concevoir une application complète et choisir des technologies pertinentes
- Mettre en place une interface front-end navigable, fluide et adressable
- Mettre en place un back-end pertinent pour l'application
- Mettre en place une politique d'installation et de mise à jour à travers l'utilisation du cache
- Respecter l'approche progressive au maximum à travers de fonctionnalités "opt-in"
- Faire fonctionner des notifications web

2. 🎮 Présentation du projet 

2.1 Présentation d'ensemble

Le but de ce projet est de mettre en place, à travers une Progressive Web App, un petit jeu au tour par tour jouable avec des amis.

Le jeu sera complètement asynchrone, c'est à dire que chaque joueur jouera quand il lui plaira dans un laps de temps de 24 heures lorsque son tour survient pour des parties qui peuvent s'étendre sur des semaines entières.

⚠️ **L'important n'est pas d'avoir mis l'intégralité du gameplay en place ni de faire quelque chose d'esthétiquement beau, mais d'avoir pu démontrer l'acquisition de toutes les compétences!**

Il vaudra donc mieux privilégier la mise en place des fonctionnalités (navigation, lancement de partie, déplacement, installation pour usage hors ligne et notifications) que de l'intégralité du gameplay ou de la finition graphique.

💡 Des cases colorées, et de simples images fonctionnelles suffisent graphiquement!

2.2 Etapes du projet:

Etape 1 - Conception

Il s'agira ici de concevoir l'architecture logicielle prévisionnelle de l'application:

- Listage des vues et des composants
- Choix des technologies front-end
- Répartition du travail initial
- Formalisation avec des outils adaptés

Etape 2 - L'interface

Il s'agira d'abord de créer une interface front-end fluide et navigable, c'est à dire des vues adressables par des URL, ainsi qu'un gameplay au tour par tour se jouant entièrement en local depuis un seul terminal.

L'intégralité du gameplay ne devra pas encore être finalisée. Il pourra être complété au fur et à mesure. L'application servira de point initial pour mettre en place le reste des étapes.

Etape 3 - Jouer hors ligne

Il s'agira de mettre en œuvre la politique d'installation et la gestion du cache pour pouvoir installer l'application et jouer hors ligne.

Etape 4 - Back-end

Il s'agira de mettre au point le back-end pour permettre le jeu en ligne, avec gestion des utilisateurs, lancement de parties et enregistrement des tours.

A cette étape, il s'agira de laisser le joueur consulter l'état de la partie, mais il ne pourra plus jouer tant qu'il ne sera pas en ligne et que ce ne sera pas son tour.

Etape 5 - Notifications

Il s'agira ici de prévenir l'utilisateur que c'est à son tour en lui envoyant une notification lorsque c'est le cas (s'il a accepté les notifications).

Une seconde notification le préviendra s'il est proche du temps limite pour jouer.

Etape 6 - Bonus

On peut à présent détecter la disponibilité de l'API Background Sync et permettre au joueur le cas échéant de jouer hors-ligne de manière à envoyer les données de son tour au serveur dès qu'il aura de nouveau accès à Internet.

2.3 Règles du jeu 🧙‍♂️

Principe

Il s'agira d'un jeu de "Battle Royale", de 2 à 5 joueurs, asynchrone et au tour par tour, pour lequel plusieurs parties pourront se jouer simultanément.

Le but du jeu est d'attaquer les autres joueurs jusqu'à ce qu'il n'en reste plus qu'un!

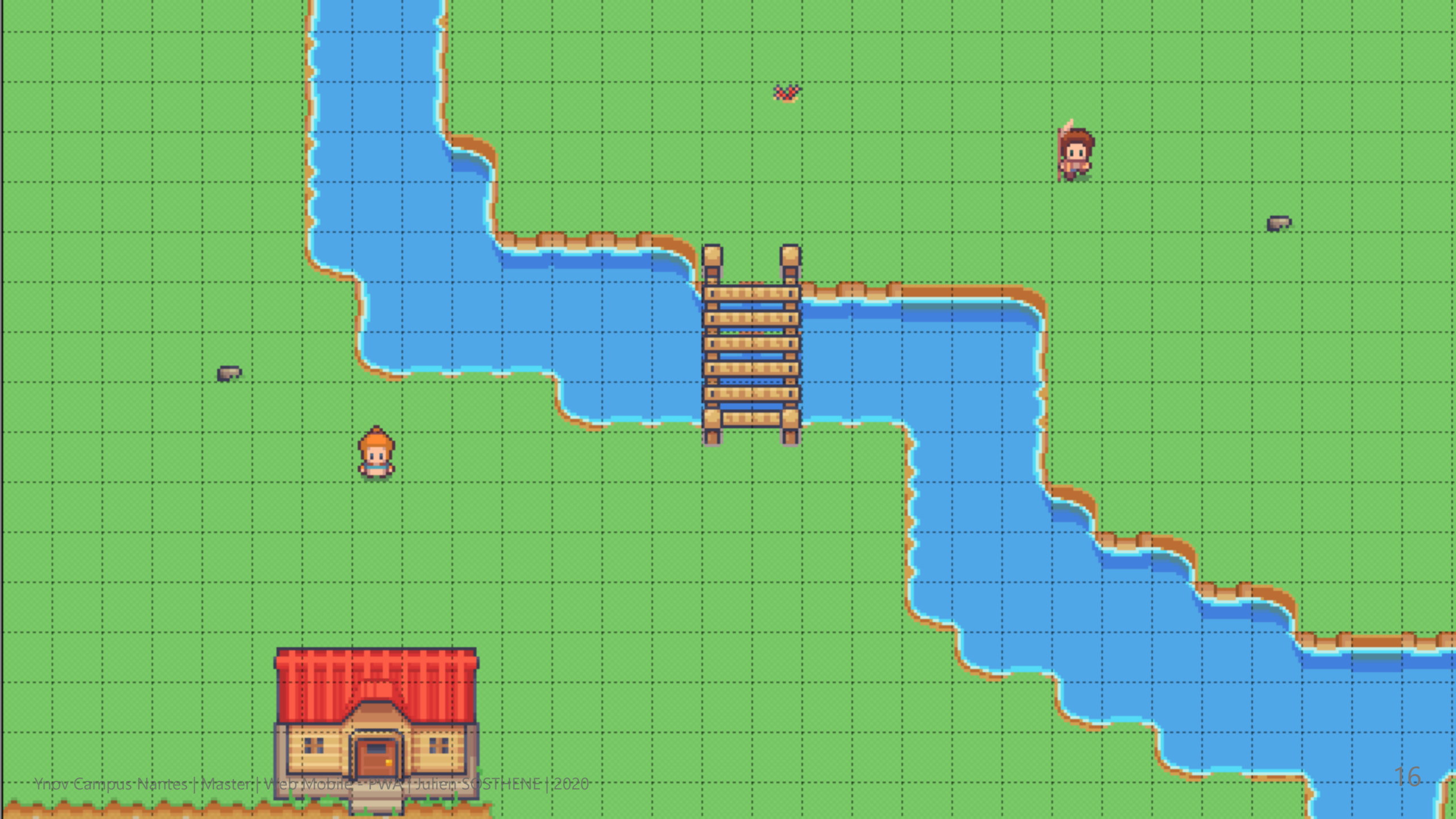
1. On peut créer une partie ou la rejoindre avec un code (ex: Jitsi Meet). Un ID est fourni à chaque navigateur se connectant, qui sera enregistré dans un lobby.
2. Le créateur de la partie peut la lancer dès qu'il y a deux joueurs et plus. La partie est lancée automatiquement si 5 joueurs sont présents.
3. Le jeu commence, avec un premier tour
    i. phase de déplacement
    ii. Phase d'attaque si joueur à portée.
4. Les tours s'enchaînent
5. Quant un joueur a gagné, la partie s'arrête.

💡 Comme rien n'empêche de rejoindre plusieurs parties en même temps, il est souhaitable d'avoir une vue pour lister les parties auxquelles l'utilisateur participe!

Terrain de jeu

Le terrain est une grande grille (taille à déterminer) où les joueurs sont initialement placés.

Sur le terrain, des obstacles et objets divers sont disséminées. Leur placement pourra être au choix fixe, choisi parmi une sélection de cartes fixes, ou entièrement aléatoire.



⚠️ Sur mobile, il faudra s'arranger pour que la carte ne soit pas visible en entier pour permettre aux cases d'être clickables!

Pensez à une zone scrollable par exemple.

💡 **Du code vous sera fourni pour gérer l'affichage de la grille.**

Tours de jeu

L'ordre des tours est déterminé aléatoirement au lancement de la partie, et chaque joueur ne joue qu'une fois que c'est son tour.

Un tour consiste en deux étapes:

- Un déplacement
- Une attaque

Chaque étape pourra être passée si le joueur ne désire pas se déplacer ou accomplir d'action, mais elles auront forcément lieu dans cet ordre.

Joueur

Chaque joueur démarre avec un nombre égal de points de vie et une arme par défaut.
Ils doivent être différentiables grâce à un indicateur graphique (image, couleur...).

Cliquer sur un joueur visible sur la grille permet de consulter sa fiche joueur avec ses PV et ses objets.

S'il est à portée, un bouton "attaquer" permet de l'attaquer.

Déplacement

Le déplacement du joueur peut se faire sur une zone l'entourant en terme de nombre de cases.

Pour se déplacer, il faut cliquer sur une case **libre**, puis confirmer son déplacement.

On ne pourra pas se déplacer sur la case d'un autre joueur ou une case contenant un obstacle.

On peut imaginer que la portée du déplacement évolue selon les objets ramassés, mais rien n'est obligatoire ici.

Attaque

Une attaque a une portée selon l'arme du joueur (ramassée parmi les objets) et un nombre de dégâts.

L'arme par défaut des joueurs a une portée d'une case, c'est à dire qu'elle ne peut toucher que les cases directement adjacentes

Objets

Les seuls objets obligatoires sont des armes, possédant une portée et un nombre de dégâts.

On peut imaginer également des potions qui rétablissent de la santé lorsqu'un joueur les ramasse, des objets donnant des points d'armure, de bottes permettant de se déplacer plus loin...

Le joueur ramasse un objet automatiquement s'il s'arrête sur la case où était l'objet.

💡 Cliquer sur l'objet permet de voir ses propriétés avant de confirmer le déplacement.

⚠️ Ramasser une nouvelle arme pourra faire tomber l'arme actuelle du joueur à terre, ou la supprimer complètement. Ce choix de gameplay vous appartient.

Attaque et fin du jeu.

Lorsqu'un joueur attaque un autre joueur, celui-ci perd une quantité de PV correspondant aux dégâts de son arme.

Si les PV d'un joueur tombent en dessous de zéro, alors il perd la partie, et les autres la poursuivent jusqu'à ce qu'il n'en reste plus qu'un, qui est alors déclaré vainqueur.

💡 **Quand un joueur vainc l'un de ses adversaires, une partie de ses PV est restaurée. (Par exemple, 30%).**

3. Déroulé du projet 🗓

3.1 Lancement et composition

Le projet est lancé dès les premières séances et s'étalera sur l'intégralité du module, avec un suivi groupe par groupe durant les heures dédiées.

Les groupes seront composés de 4 membres.

3.2 Durée

De son lancement à son évaluation, le projet durera 8 semaines.

La version finale devra être rendue une semaine après la soutenance.

3.3 Rendu intermédiaire

En dehors du suivi classique, un rendu intermédiaire est demandé au milieu du projet, qui donnera lieu à une première note.

Il devra contenir le gros du front-end de l'application, avec la mise en place des vues, et tout ce qui le rendra installable (Service worker/manifeste)

L'ensemble du gameplay n'est pas exigible ici, mais l'affichage de la grille et le déplacement du joueur sont souhaitables.

3.4 Priorités

⚠️ **La priorité absolue n'est pas le gameplay ni le rendu graphique, mais la mise en place des diverses fonctionnalités.**

Si vous sentez du retard sur votre progression, il vaut mieux retirer des mécaniques de gameplay et les détails esthétiques. Il s'agit avant tout d'un prétexte à la mise en place d'une Progressive Web App complète.

Minimal Viable Product (MVP)

Le rendu minimum souhaité est une application avec:

- La totalité des vues nécessaires au lancement d'une partie
- La totalité des éléments nécessaires à l'installation de l'application
- La vue de Personnage avec possibilité de se déplacer et d'attaquer un autre joueur
- Un backend responsable de centraliser les données et de distribuer les tours.
- Des notifications de tour sur la PWA.

4. Contraintes techniques

4.1 Mise en place d'une PWA

Votre application devra

- Etre installable
- Fonctionner hors ligne et conserver le dernier état de la partie récupéré en ligne
- Etre capable de recevoir des notifications
- Passer les tests de Google Lighthouse.

4.2 Navigabilité

- Chaque vue devra correspondre à une URL différente (comprendre: le clic sur un lien doit nous amener au bon endroit dans votre application)
- Un bouton de retour sera implémenté si c'est nécessaire pour les utilisateurs d'iPhone

4.3 Gestion de versions

Votre projet devra être versionné sous Git et l'historique des commits devra être fournie.

4.4 Technologies

💡 Le choix des technologies est libre pour le front-end et pour le back-end. Le code source doit en revanche être consultable.

Vous pouvez choisir les technologies évoquées en cours si vous le désirez mais ce n'est pas obligatoire.

5. Livrables

5.1 Livrables du rendu intermédiaire

- Un document de conception front-end comprenant
  - Une liste des vues,
  - Une liste des composants nécessaires
  - La répartition des tâches du groupe
  - Un planning prévisionnel réalisé en début de projet.
  - La justification technique des technologies choisies.
- Le dépôt complet du projet contenant son historique sur Git
- Un fichier Readme expliquant comment lancer le projet en local.

5.2 Livrables de la soutenance

💡 La soutenance aura lieu une semaine avant le rendu final.

- Un support de présentation (sous forme de PDF)
- Un bilan du projet comprenant:
  - L'état du projet au moment de la soutenance
  - Une description des étapes restantes et de l'approche envisagée pour les mettre en place.

5.3 Livrables finaux

- Lien de l'application (hébergé sur une plateforme de votre choix)
- Une archive contenant votre code ainsi que l'historique Git (`/.git`); le lien du dépôt ne suffit pas.
- Un court document décrivant vos modifications depuis la soutenance et les possibles améliorations que vous auriez pu apporter.

6. Modalités de rendu et d'évaluation

6.1 Modalités de rendu

Les rendu se feront via Teams, sur 3 devoirs (rendu intermédiaire, rendu de soutenance, rendu final)

Tout rendu en retard fera l'objet de points de pénalité.

6.2 Modalités d'évaluation

L'évaluation se fera sur 2 notes:

- Le rendu intermédiaire (Coefficient 1)
- Le rendu final et la soutenance (Coefficient 2)

6.3 Modalités de soutenance

- La soutenance sera composée de
  - 15 à 20 minutes de présentation de votre part
  - 10 minutes de questions/réponses
- Attention à votre gestion du temps et au partage de la parole
- Vous devrez démontrer l'essentiel de votre conception, de vos choix techniques et faire une démonstration.

⚠️ Le but de cette soutenance est de prouver l'acquisition des compétences présentées au début de ce document.

Veillez donc à l'organiser de manière à privilégier la justification de ces compétences!

6.4 Critères d'évaluation

Vous serez évalué selon les critères suivants :

- Respect des consignes et fonctionnement de votre PWA
- Sérieux, pertinence et propreté des rendus écrits et de la soutenance
- Propreté et organisation du code
- Pertinence de la répartition des tâches
- Niveau de complétion du gameplay / Finition générale
- Des points bonus seront accordés pour ceux qui surpassent les attentes du projet.

⚠️ S'il est constaté un trop gros déséquilibre dans la participation aux commits, des notes différentes pourront être délivrées au sein d'un même groupe.

C'est parti! Bon courage!