Table des matières

[1 Analyse préliminaire 3](#_Toc516127184)

[1.1 Introduction 3](#_Toc516127185)

[1.2 Objectifs 3](#_Toc516127186)

[1.3 Descriptif 4](#_Toc516127187)

[1.4 Planification initiale 5](#_Toc516127188)

[2 Analyse / Conception 6](#_Toc516127189)

[2.1 Vue d’ensemble 6](#_Toc516127190)

[2.2 MCD 6](#_Toc516127191)

[2.3 Stratégie de test 7](#_Toc516127192)

[2.4 Risques techniques 7](#_Toc516127193)

[2.5 Planification 7](#_Toc516127194)

[2.6 Dossier de conception 8](#_Toc516127195)

[2.7 Use cases 8](#_Toc516127196)

[2.8 Scénarios 9](#_Toc516127197)

[2.8.1 S’authentifier 9](#_Toc516127198)

[2.8.2 Rejoindre une partie 10](#_Toc516127199)

[2.8.3 Jouer 11](#_Toc516127200)

[2.8.4 Consulter les statistiques 15](#_Toc516127201)

[2.8.5 Paramétrage du jeu 15](#_Toc516127202)

[2.9 Maquette 16](#_Toc516127203)

[2.9.1 Page de connexion 16](#_Toc516127204)

[2.9.2 Page d’inscription 17](#_Toc516127205)

[2.9.3 Page d’accueil vue joueur 18](#_Toc516127206)

[2.9.4 Page d’accueil vue administrateur 19](#_Toc516127207)

[2.9.5 Page de paramétrage 20](#_Toc516127208)

[2.9.6 Partie en attente de joueurs 21](#_Toc516127209)

[2.9.7 Partie en cours, vue du joueur qui joue 22](#_Toc516127210)

[2.9.8 Partie en cours avec erreur 23](#_Toc516127211)

[2.9.9 Partie en mode coopératif, relâcher un poisson 24](#_Toc516127212)

[2.9.10 Partie en cours, un autre joueur joue 25](#_Toc516127213)

[2.9.11 Partie gagnée 26](#_Toc516127214)

[2.9.12 Partie perdue 27](#_Toc516127215)

[2.9.13 Consulter les statistiques 28](#_Toc516127216)

[2.9.14 Fin d’une saison 29](#_Toc516127217)

[2.10 MLD 30](#_Toc516127218)

[2.10.1 Particularité 1 30](#_Toc516127219)

[2.10.2 Particularité 2 30](#_Toc516127220)

[2.10.3 Particularité 3 30](#_Toc516127221)

[2.11 Brève explication du jeu 31](#_Toc516127222)

[3 Réalisation 32](#_Toc516127223)

[3.1 Dossier de réalisation 32](#_Toc516127224)

[3.2 Description des tests effectués 33](#_Toc516127225)

[3.3 Erreurs restantes 34](#_Toc516127226)

[3.4 Liste des documents fournis 34](#_Toc516127227)

[4 Conclusions 35](#_Toc516127228)

[5 Annexes 36](#_Toc516127229)

[5.1 Planification initiale 36](#_Toc516127230)

[5.2 Résumé du rapport du TPI / version succincte de la documentation 37](#_Toc516127231)

[5.3 Sources – Bibliographie 37](#_Toc516127232)

[5.4 Journal de bord 38](#_Toc516127233)

[5.5 Manuel d'Installation 39](#_Toc516127234)

[5.5.1 Windows 39](#_Toc516127235)

[5.5.2 Mac 39](#_Toc516127236)

[5.6 Archives du projet 39](#_Toc516127237)

# Analyse préliminaire

## Introduction

Ce projet est réalisé en tant que TPI, le travail est effectué sur les postes disponibles au CPNV. Ce projet permettra de m’évaluer selon les critères donnés dans le cahier des charges.

Auparavant j’ai eu l’occasion de réaliser un poker en ligne pendant mon pré-TPI permettant à plusieurs joueurs en même temps. Je pourrais utiliser les connaissances acquises tout au long de mon pré-TPI afin de réaliser mon TPI.

Lors du TPI, le projet à réaliser traitera d’un jeu de pêche basé sur la stratégie et les conséquences des actions faîtes auparavant.

## Objectifs

Lors de ce projet je veux :

* Améliorer mes connaissances dans l’architecture MVC
* Tenir une planification dans Trello tout au long du projet et à jour à l’aide de scrum for trello
* Créer un jeu paramétrable
* Créer un jeu avec plusieurs joueurs en réseau

## Descriptif

Un utilisateur va sur le site pour :

* S’inscrire sur le site.
* Se connecter au site. Il rejoint alors une page qui montre une liste de parties qui sont soit en cours, soit en attente de joueurs.
* Rejoindre une partie en attente de joueurs, si le nombre voulu de joueurs est atteint, la partie démarre
* Consulter ses statistiques

Contexte de départ du jeu :

* Il y a un lac avec 60 poissons.
* Les joueurs (pêcheurs) sont disposés autour du lac
* Chaque pêcheur a son étang privé, contenant 3 poissons
* Une partie est caractérisée par une stratégie de repeuplement du lac :
  + Coopératif : le nombre qu’il veut (peut-être aucun)
  + Imposition : un nombre calculé en fonction des poissons qu’il possède
  + Imposition avec forfait : un nombre calculé en fonction des poissons qu’il possède, mais au maximum 10

Une partie compte 18 tours de jeu – appelés saison de pêche.

Déroulement d’une saison de pêche :

* Chaque pêcheur relâche des poissons dans le lac selon le type de partie choisi
* Chaque pêcheur pêche tour à tour autant de poissons qu'il veut et les met dans son étang. Il ne peut pas pêcher plus que le nombre de poissons qu’il possède.
* A chaque saison c’est un autre pêcheur qui commence la pêche
* Les poissons se reproduisent : 1 pour 2 en étang, 3 pour 2 dans la nature
* Pour vivre, chaque pêcheur mange 2 poissons de son étang.

La partie se solde par plusieurs scores :

* Le score commun, qui est le nombre de poissons restant dans le lac après toutes les saisons de pêche.
* Le score collectif, qui est le nombre total de tous les poissons
* Le score individuel de chaque pêcheur, qui est le score commun plus le nombre de poissons dans son étang

Classements :

* Le classement général des joueurs est établi en fonction de la moyenne des scores individuels obtenus.
* Lorsque meilleur score commun est battu dans une partie, cette partie compte double dans les statistiques du joueur.
* Un joueur ayant moins de 3 parties à son actif n’est pas classé

Divers :

* Tous les chiffres cités dans le descriptif ci-dessus doivent être facilement modifiables

## Planification initiale

Une méthode agile a été imposée dans le cahier des charges et sera donc gérée à l’aide de l’outil Trello.

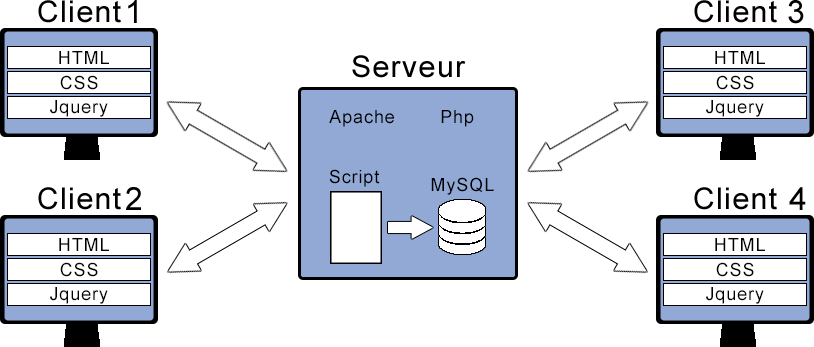
Ce point reprend la liste des sprints disponibles dans le Trello, afin d’avoir plus de détail il faut se rendre sur le Trello donné dans l’adresse ci-dessous. Chaque story point équivaut à une période, soit 45 minutes.

<https://trello.com/b/k0NRKXsM/fishermen-land> (Disponible au point 5.1 en annexe)

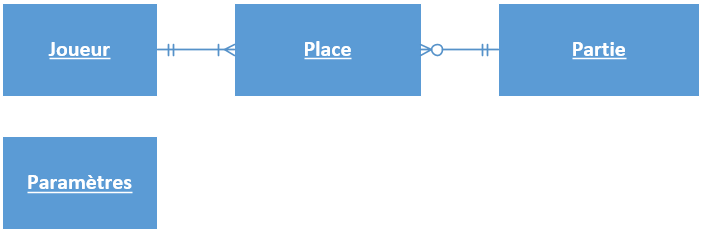
|  |  |  |
| --- | --- | --- |
| Sprint | Description | Date |
| Pré-analyse | Planification et début d’analyse | 09.05.2018 |
| Sprint 1 | Analyse et conception | 18.05.2018 |
| Sprint 2 | Gérer une partie et les rangs | 25.05.2018 |
| Sprint 3 | Gérer le déroulement d’une partie et les scores | 01.06.2018 |
| Sprint 4 | Ajout de l’affichage et déboguer | 07.06.2018 |
| Livraison du TPI |  | 07.06.2018 |

# Analyse / Conception

## Vue d’ensemble

**

## MCD



**Joueur :** Garde les données d’un joueur, comme son pseudo, son mot de passe, s’il est admin ou non et son rang.

**Place :** Permet à un joueur de rejoindre une partie, celle-ci va garder les informations permettant de calculer les statistiques et l’ordre de jeu.

**Partie :** Les parties contiennent toutes les données nécessaires quant au bon déroulement d’une partie et garde certaines valeurs nécessaires pour créer les statistiques.

**Paramètres :** Un administrateur pourra changer certaines valeurs, tel que le nombre de poissons par défaut dans le lac, le nombre de joueurs maximum dans une partie, etc… Les parties seront créés en fonction de ces paramètres.

## Stratégie de test

Tout au long du projet je fais des tests fonctionnels selon mon avancée, afin de s’assurer que les fonctionnalités demandées fonctionnaient chacune de leur côté.

Afin de gérer les tests fonctionnels j’ai souvent utiliser plusieurs navigateurs. Pour connecter 6 personnes j’utilisais une fois le navigateur en mode normal et une fois en mode navigation privé, ceci me permet d’avoir 6 comptes connectés au même temps.

Par la suite j’ai commencé à faire des test d’intégrité afin de s’assurer qu’on puisse parcourir les différentes fonctionnalités du jeu sans avoir de problèmes.

Je présente également chaque semaine au chef de projet une démo de ce qui est fonctionnel et m’assure que ce qui était fonctionnel les semaines d’avant, le reste.

Par la suite je vais faire des tests de robustesse en demandant à plusieurs joueurs de jouer au même temps sur mon réseau local afin de voir s’il n’y a pas de problème lorsque plusieurs joueurs jouent au même temps.

Au final je ferais des tests de sécurité si mon projet est fini avant le délai, afin de s’assurer que toutes les pages et chaînes de caractères soient sécurisées.

## Risques techniques

J’ai un manque de connaissances au niveau de l’architecture MVC car c’est la première fois que celle-ci est utilisée. Afin de combler ce problème j’ai suivi les cours de openclassrooms sur internet le week-end.

Manque de connaissances en Jquery. Quand le moment sera venu d’automatiser mes requêtes SQL je suivrais le cours présent sur openclassrooms et si ceci ne suffit pas, je chercherais un maximum d’informations sur internet.

Il se peut également que le jeu soit mal sécurisé car je manque de compétences dans ce domaine. La sécurité ne sera peut-être pas implémentée, celle-ci n’est pas référencée dans le cahier des charges.

## Planification

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Description | Date | Date effective |
| Pré-analyse | Planification et début d’analyse | 09.05.2018 | 09.05.2018 |
| Sprint 1 | Analyse et conception | 18.05.2018 | 22.05.2018 |
| Sprint 2 | Gérer une partie et les rangs | 25.05.2018 | 25.05.2018 |
| Sprint 3 | Gérer le déroulement d’une partie et les scores | 01.06.2018 | 01.06.2018 |
| Sprint 4 | Ajout de l’affichage et déboguer | 07.06.2018 | 07.06.2018 |
| Livraison du TPI |  | 07.06.2018 | 07.06.2018 |

## Dossier de conception

Matériel et OS utilisé :

* Une tour Dell Optiplex 9010
* Windows 7 entreprise

Programmes utilisés :

* Documentation et communication
  + Word
  + Excel
  + Outlook
* Versioning et sauvegarde des données
  + GitHub
* Planification du projet TPI
  + Trello
* Création des marquettes
  + Balsamiq Mockups 3
* Retouche d’images
  + Photoshop
  + Krita
* Editeurs de texte
  + Brackets
  + Atome
  + Notepad++
* Navigateurs web
  + Google Chrome
  + Firefox
  + Opera
* Base de donnée
  + MySQL Workbench 6.3 CE
* Plateforme de développement web
  + Wampserver 3.0.6 64bit

## Use cases

Dans les use cases, j’ai choisis d’aborder les thèmes suivants :

* S’authentifier
* Rejoindre une partie
* Jouer
* Consulter les statistiques
* Paramétrage du jeu

## Scénarios

### S’authentifier

#### Création d’un compte + erreurs

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| Rentrer l’URL de la page web |  | La page web demande à l’utilisateur de se connecter, de créer un compte |
| L’utilisateur clique sur « Pas encore de compte ? Inscrivez-vous ! » |  | Page d’inscription |
| L’utilisateur rentre des identifiants | Le pseudo existe déjà | Le site refuse de lui créer un compte |
| L’utilisateur change son pseudo | Le mot de passe n’est pas assez long | Le site demande à l’utilisateur de mettre plus de caractères |
| L’utilisateur change le mot de passe selon les critères donnés | Le compte est créé | Son compte est créé et il est authentifié et redirigé vers la page d’accueil |

#### Connexion à un compte + erreurs

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| Rentrer l’URL de la page web |  | La page web demande à l’utilisateur de se connecter, de créer un compte |
| L’utilisateur rentre ses identifiants | Mot de passe incorrect | Le site refuse de se connecter et dit que le mot de passe est erroné |
| L’utilisateur rentre ses identifiants | Pseudo incorrect | Le site refuse de se connecter et dit que le pseudo est erroné |
| L’utilisateur retape ses identifiants | Le compte et mot de passe correspondent à ceux existants | L’utilisateur est authentifié et redirigé vers la page d’accueil |

### Rejoindre une partie

Les parties qui ne peuvent pas être rejointes n’ont pas de bouton « Rejoindre ». Pour qu’une partie puisse être rejointe celle-ci ne doit pas être « En cours ».

#### Rejoindre une partie + début de partie

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| L’utilisateur clique sur un bouton « Rejoindre » | Il y a encore 3 places | Message « En attente de 3 joueurs » qui se charge automatiquement |
|  | Un joueur rejoint la table | Message « En attente de 2 joueurs » qui se charge automatiquement |
|  | Un joueur rejoint la table | Message « En attente de 1 joueurs » qui se charge automatiquement |
|  | Le dernier joueur rejoint la table | La partie commence, le statut passe à « En cours » |

### Jouer

#### Déroulement d’un tour en « coopératif » + perdre par manque de nourriture

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| Le joueur rentre « 30 » dans le champ de texte et appuyé sur le bouton « pêcher » | Le joueur n’as pas assez de poissons dans son étang | Message « Vous ne pouvez pas pêcher plus de poissons que ce que vous possédez déjà » |
| Le joueur rentre « 7 » dans le champ de texte et appuyé sur le bouton « pêcher » |  | 7 poissons sont ajoutés à son étang et 7 sont retirés du lac |
|  | Le joueur peut relâcher des poissons, en pêchant il a passé son tour |  |
| Le joueur rentre « 14 » dans le champ de texte et appuie sur le bouton « relâcher » | Le joueur ne peut pas se nourrir | Le joueur a perdu la partie |

#### Déroulement d’un tour en « imposition » + oublier de jouer

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| Le joueur rentre « 4 » dans le champ de texte et appuyé sur le bouton « pêcher » |  | 4 poissons sont ajoutés à son étang |
| Le joueur passe son tour | Un nombre de poissons est relâché en fonction de son nombre de poissons | 16 poissons sont relâchés |
|  | Le joueur attend son tour |  |
|  | Le joueur à oublier de jouer son tour, un nombre de poissons est relâché en fonction de son nombre de poissons | 14 poissons sont relâchés |

#### Déroulement d’un tour en « imposition avec forfait »

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| Le joueur rentre « 75 » dans le champ de texte et appuyé sur le bouton « pêcher » | Il n’y a pas assez de poissons dans le lac | Message « Vous ne pouvez pêcher que 38 poissons » |
| Le joueur rentre « 38 » dans le champ de texte et appuyé sur le bouton « pêcher » |  |  |
| Le joueur passe son tour | Un nombre de poissons est relâché en fonction de son nombre de poissons | 10 poissons sont relâchés car c’est la limite |

#### Gagner une partie

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| Le joueur pêche les derniers poissons |  | Les prochains joueurs doivent jouer |
|  | Les autres joueurs perdent par manque de nourriture |  |
|  | Le joueur à gagner la partie | Message « Vous avez gagné » et le score « commun, collectif et individuel » est affiché. |

#### Perdre une partie

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
|  | Il n’y a plus de poissons à pêcher dans le lac |  |
| Le joueur passe son tour | Un nombre de poissons est relâché en fonction de son nombre de poissons | 0 poissons sont relâchés |
|  | Le joueur à perdu | Message « Vous avez perdu » |
|  | Le joueur attend la fin de la partie | La saison est terminée. Le score « commun, collectif et individuel » est affiché. |

#### Perdre une partie et quitter

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
|  | Il n’y a plus de poissons à pêcher dans le lac |  |
| Le joueur passe son tour | Un nombre de poissons est relâché en fonction de son nombre de poissons | 0 poissons sont relâchés |
|  | Le joueur à perdu | Message « Vous avez perdu » |
| Le joueur clique sur « Quitter la partie » |  | Le joueur est redirigé vers la page d’accueil |

#### Fin d’une saison

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
|  | La partie est terminée depuis quelques secondes. L’affichage est le même pour tous les joueurs | Deux boutons apparaissent, « Rejouer » et « Quitter la partie » |
| Le joueur clique sur « Rejouer » | L’use case « Jouer » s’applique mais ce n’est pas le même joueur qui commence à jouer |  |
| Le joueur clique sur « Quitter la partie » |  | Le joueur est redirigé vers la page d’accueil |

#### Gérer le classement

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
|  | Le joueur à gagner sa deuxième partie. Il n’as pas assez de parties pour être classé | Une nouvelle partie recommence |
|  | Le joueur à perd sa troisième partie. Il est classé | Une nouvelle partie recommence |
|  | Le joueur gagne sa quatrième partie. Il bat le meilleur score commun, les points sont doublés. Il gagne des place de classement | Une nouvelle partie recommence |
|  | Le joueur perd sa cinquième partie. Il perd des place de classement |  |

### Consulter les statistiques

#### Afficher les statistiques

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| Le joueur clique sur l’icône en haut à droite lorsque la partie a déjà commencer |  | Le nombre de tour joués au total et le nombre de tours nécessaires pour terminer une saison de pêche sont affichés. Le joueur qui joue est mis en évidence. On voit le nombre de poissons de chaque joueur dans leurs étangs et les classements. |
| Le joueur passe la souris au-dessus de la fenêtre de jeu |  | La fenêtre de statistiques se ferme |

### Paramétrage du jeu

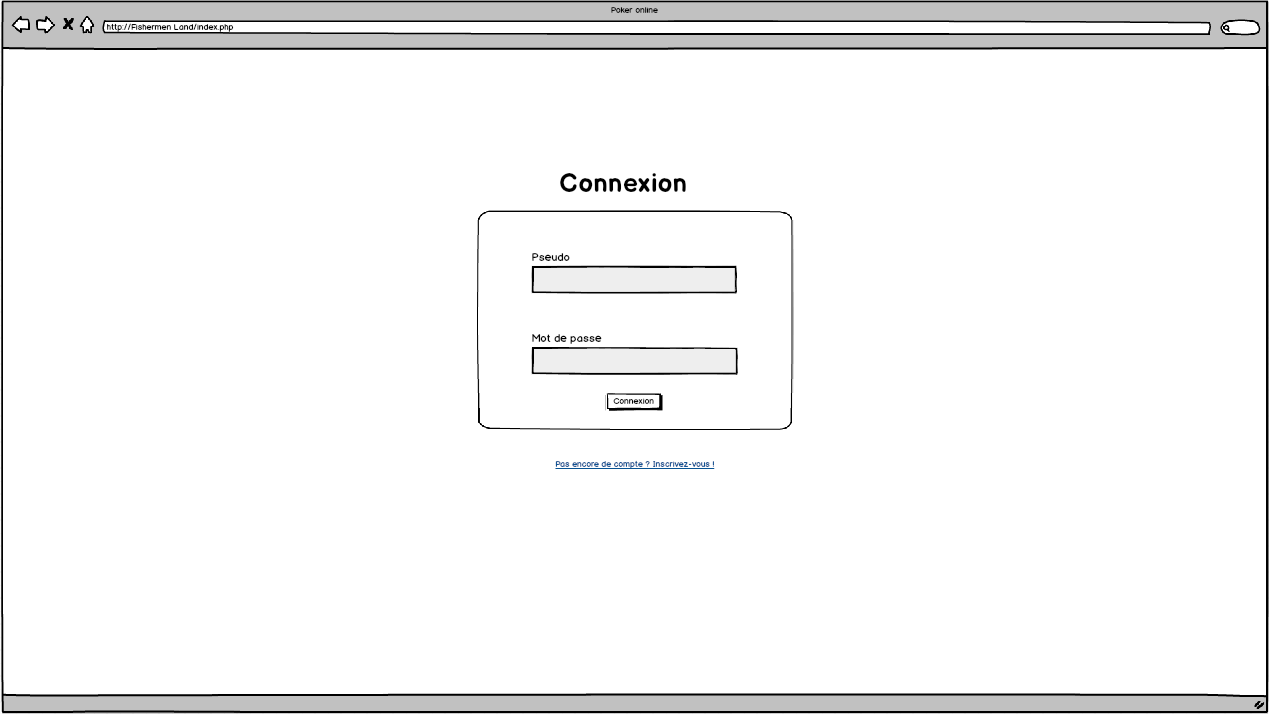
#### L’administrateur est connecté

|  |  |  |
| --- | --- | --- |
| **Action** | **Situation particulière** | **Réaction** |
| L’administrateur clique sur « Paramétrage du jeu » dans la page d’accueil |  | Toutes les données modifiables sont affichées |
| L’administrateur change une valeur et clique sur « Valider » |  | Le paramètre a été changé. Toutes les parties subissent le changement |
| L’administrateur clique sur « Réinitialiser » |  | Les valeurs par défaut sont appliquées |

## Maquette

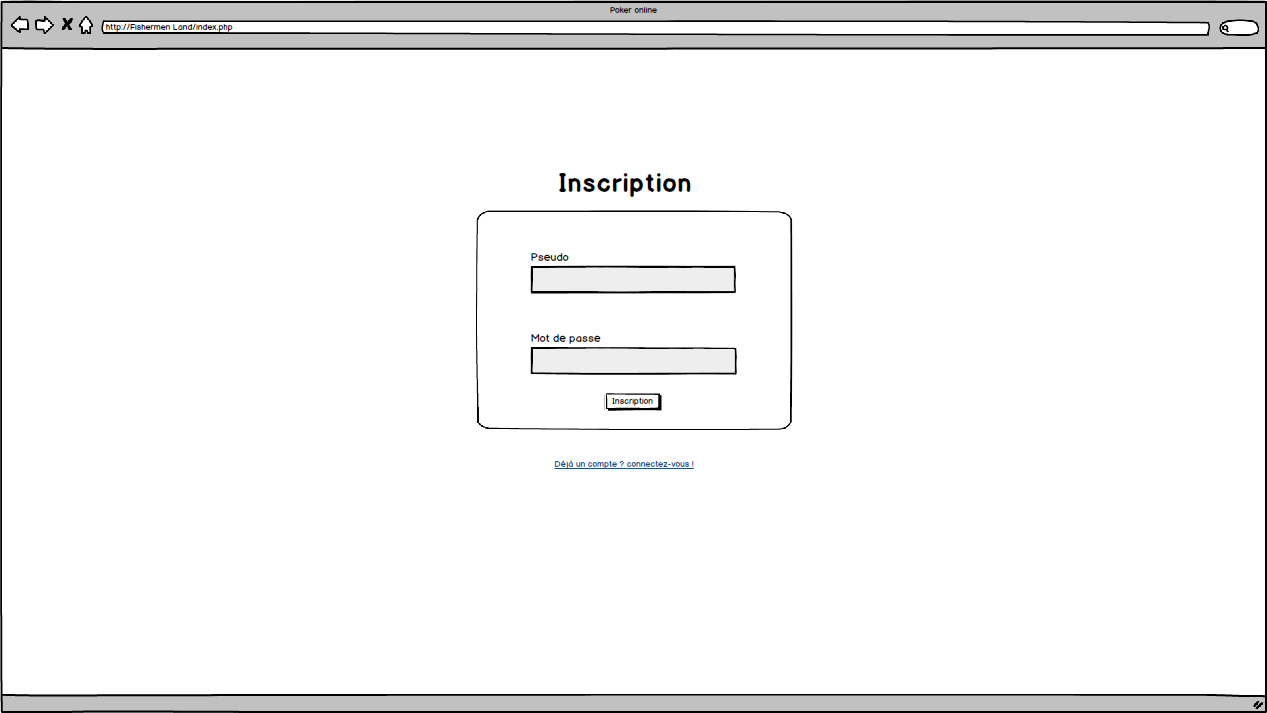
### Page de connexion

Première page affichée pour le joueur, cette page lui permet de se connecter ou d’aller à la page inscription.



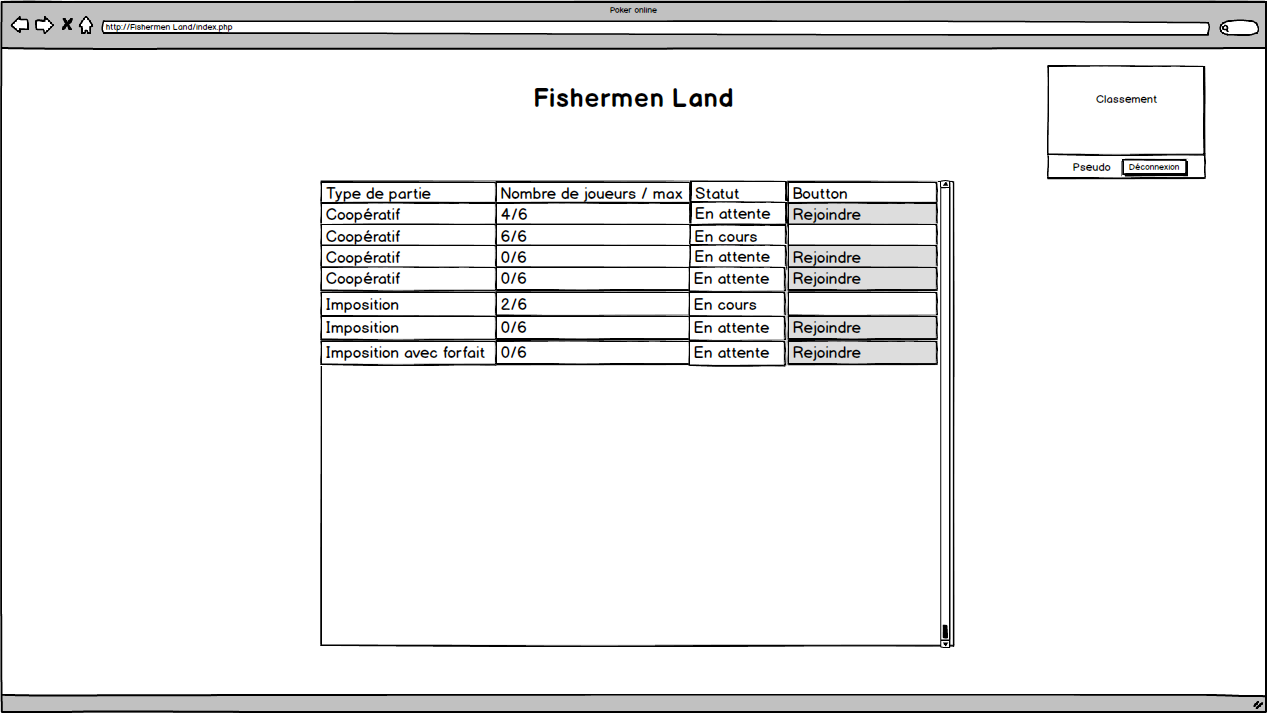
### Page d’inscription

Cette page permet au joueur de s’inscrire ou d’aller à la page de connexion.



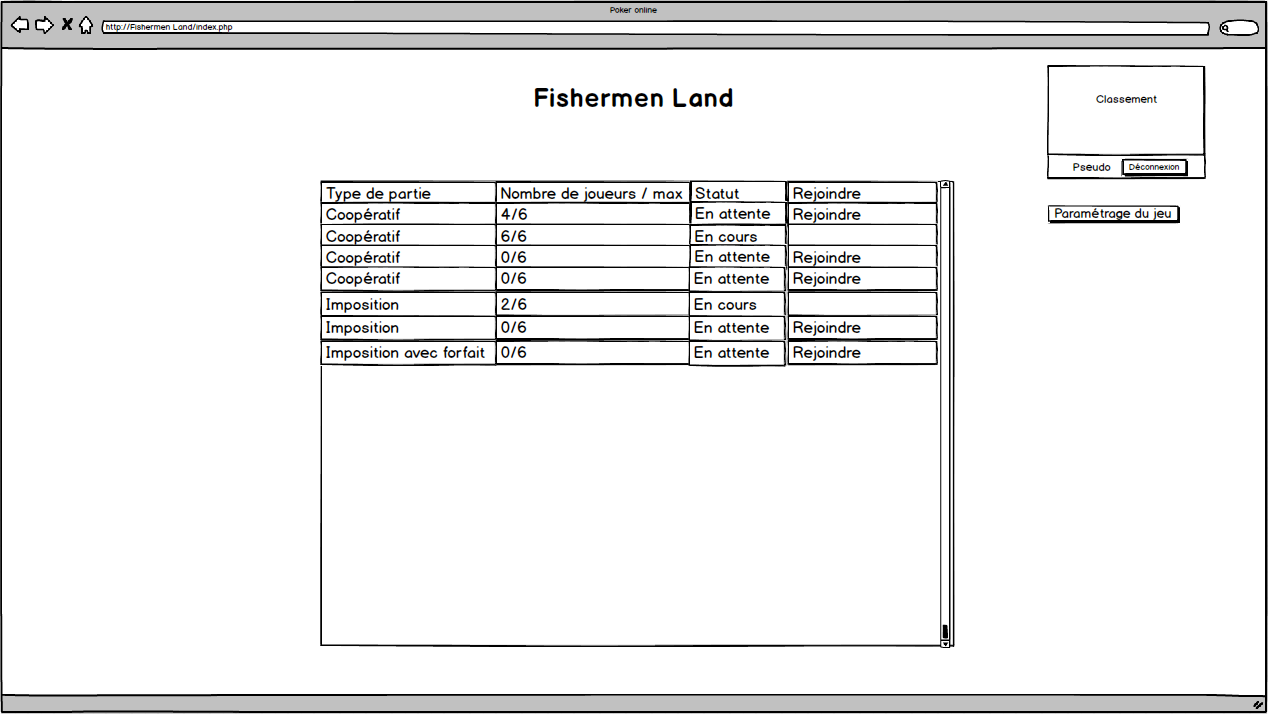
### Page d’accueil vue joueur

Lorsqu’un joueur est connecté, il est redirigé vers cette page, il peut voir les parties et les rejoindre, sont classement et se déconnecter.



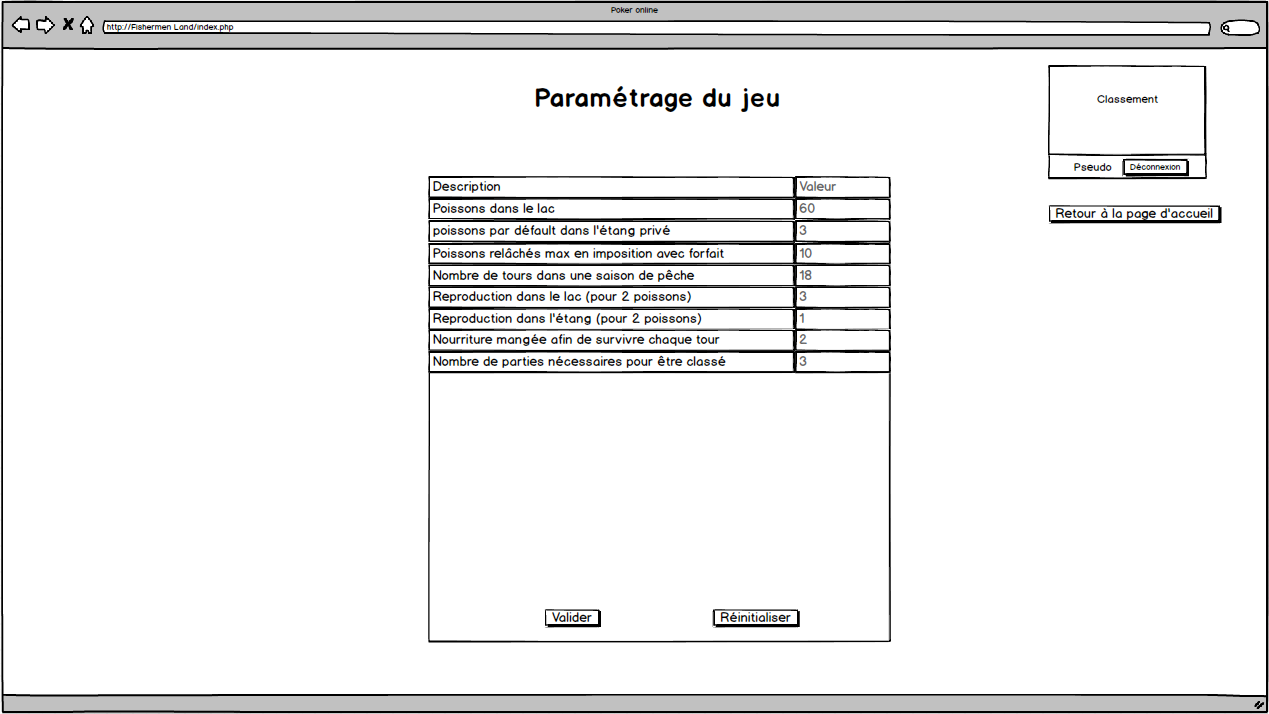
### Page d’accueil vue administrateur

Lorsque l’admin se connecte, il voit ce qu’un utilisateur peut voir, mais il a un bouton en plus, lui permettant de modifier quelques paramètres qui modifierons le fonctionnement d’une partie.



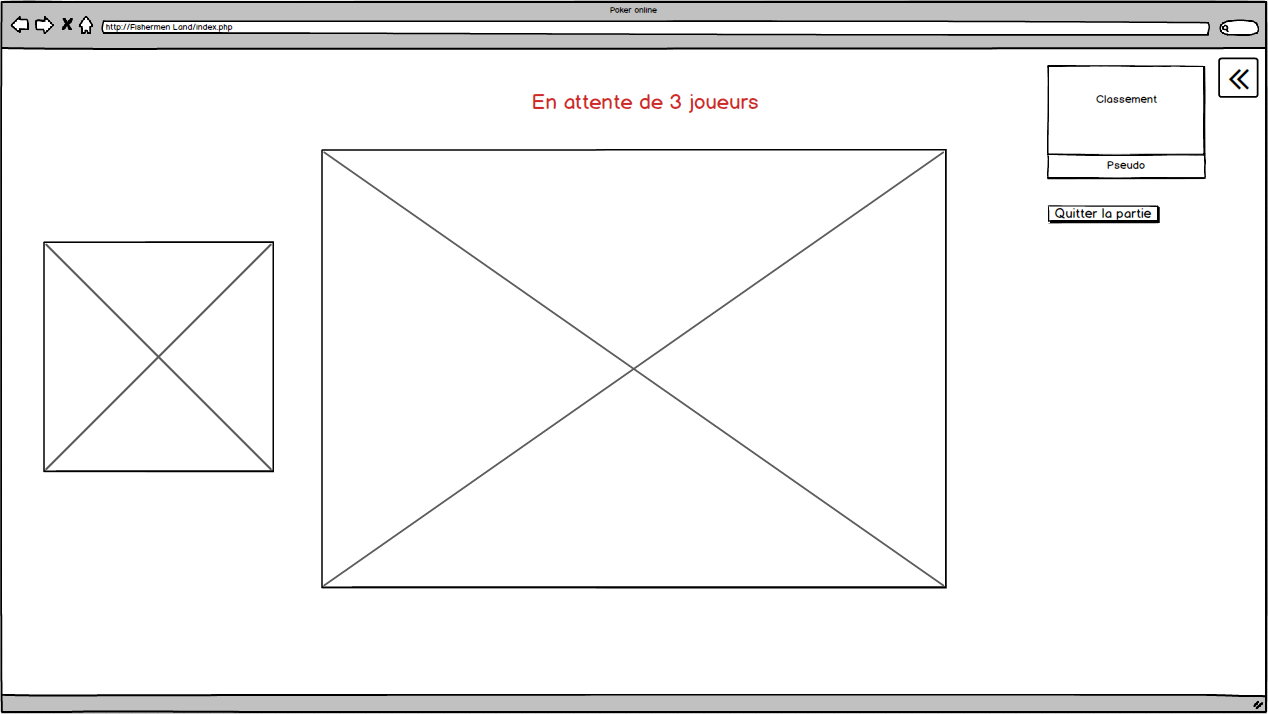
### Page de paramétrage

Lorsque l’admin à appuyer sur le bouton « Paramétrage du jeu » il voit cette fenêtre, qui lui permet de modifier directement les paramètres de parties.



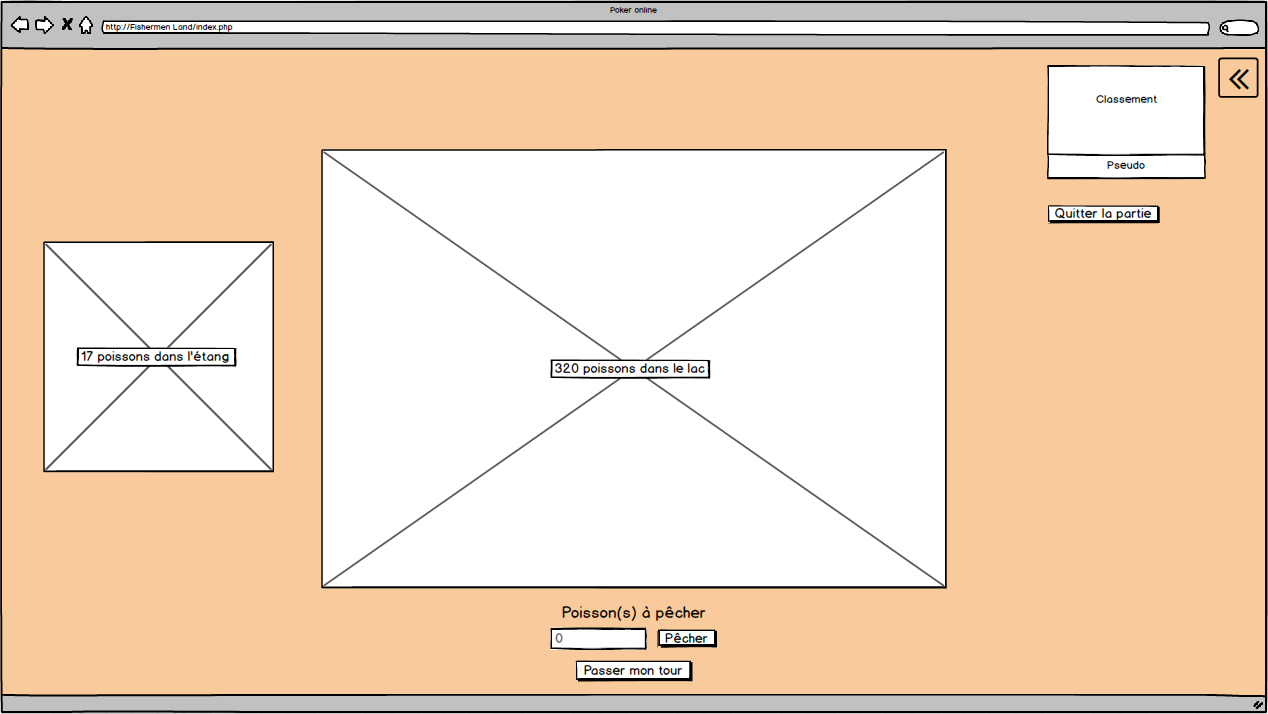
### Partie en attente de joueurs

Lorsqu’un joueur rejoint une partie et que celle-ci n’est pas pleine, on lui indique combien de joueurs sont manquants.



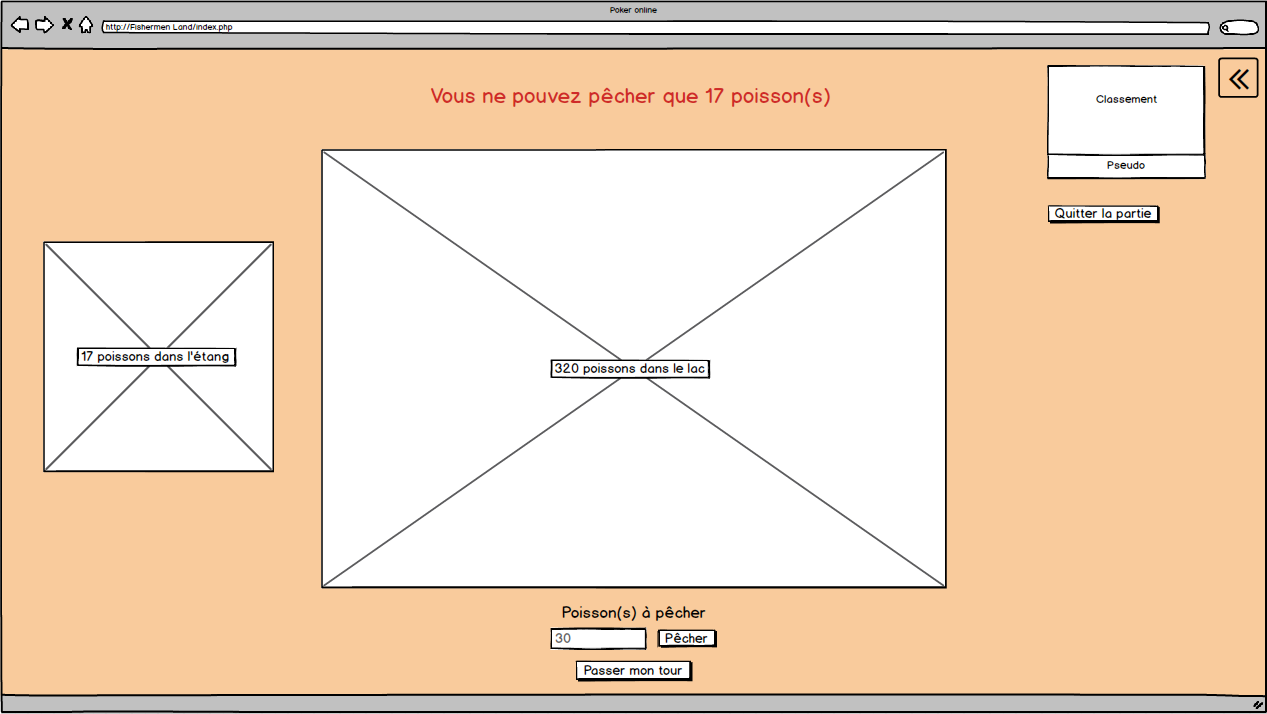
### Partie en cours, vue du joueur qui joue

Lorsque c’est au tour du joueur, la vue est différente de son écran d’attente, afin d’attirer son attention.



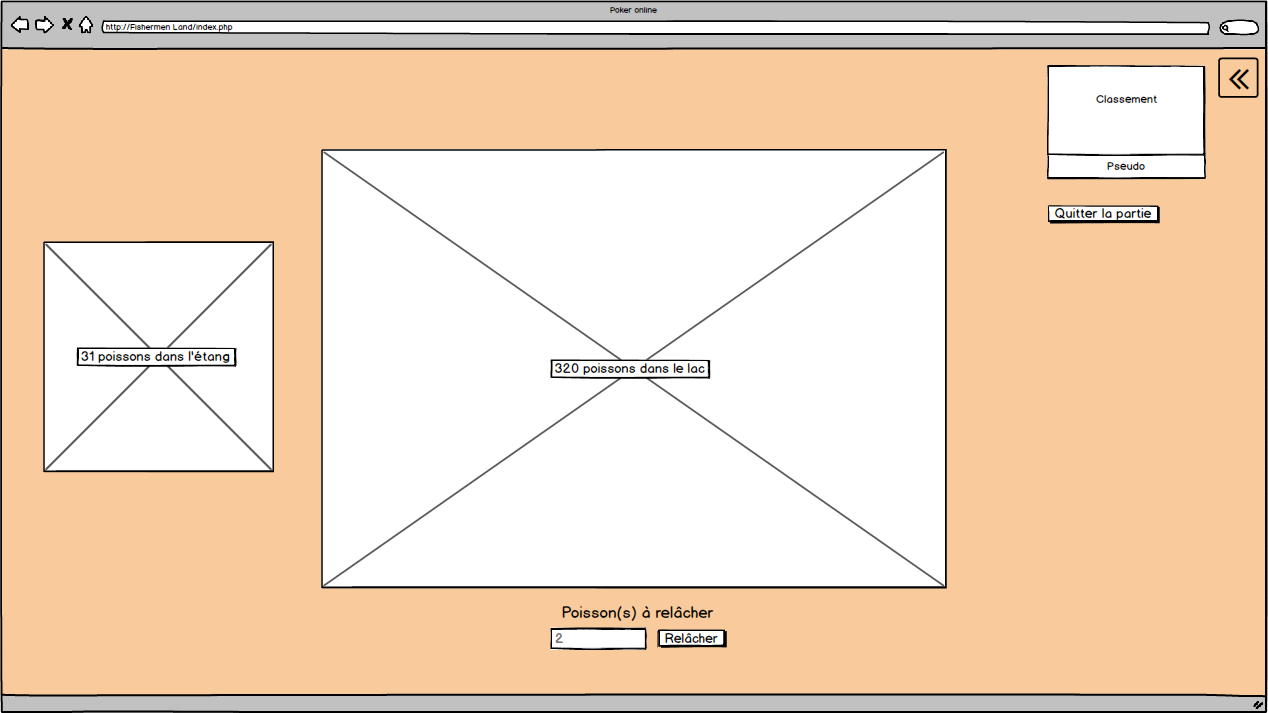
### Partie en cours avec erreur

Si le joueur essaye de pêcher plus de poissons que ce qu’il possède, on lui affiche un message d’erreur. Ceci est également valable quand il essaye d’en relâcher plus qu’il en possède.



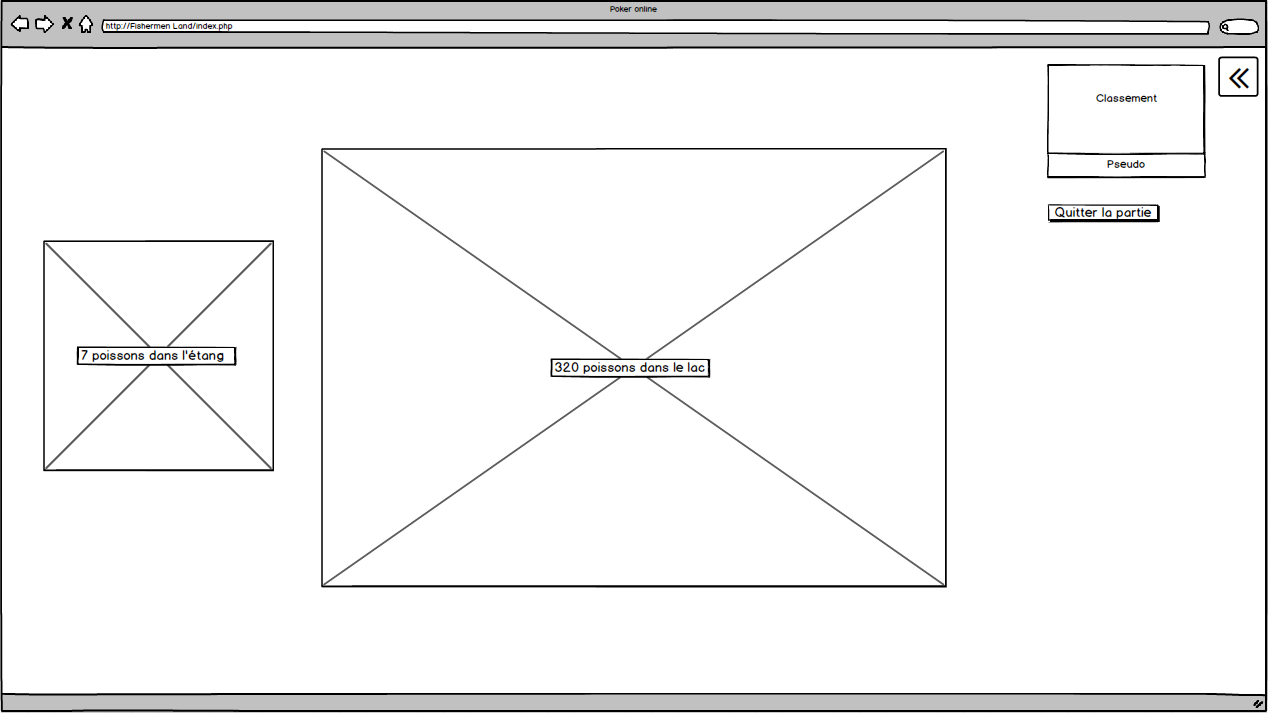
### Partie en mode coopératif, relâcher un poisson

En mode coopératif le joueur peut choisir le nombre de poissons qu’il veut relâcher.



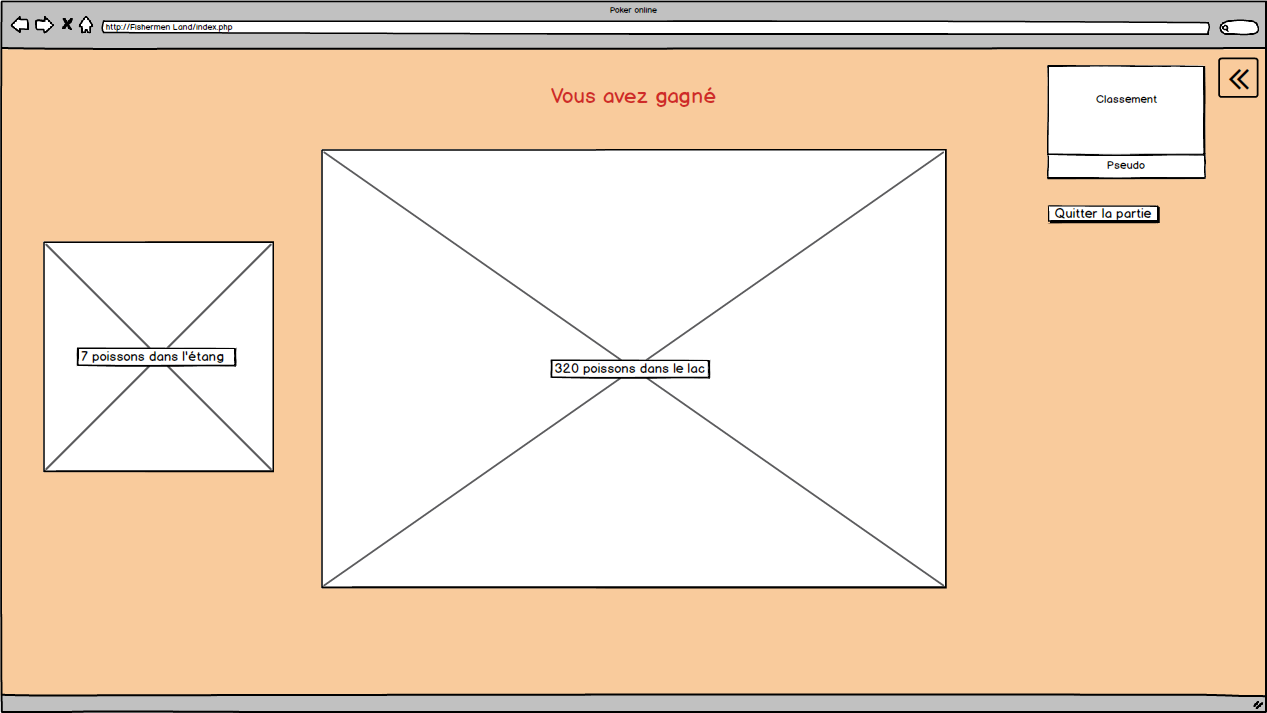
### Partie en cours, un autre joueur joue

Lorsque le joueur ne joue pas, sont affichage change afin de dire que ce n’est pas son tour.



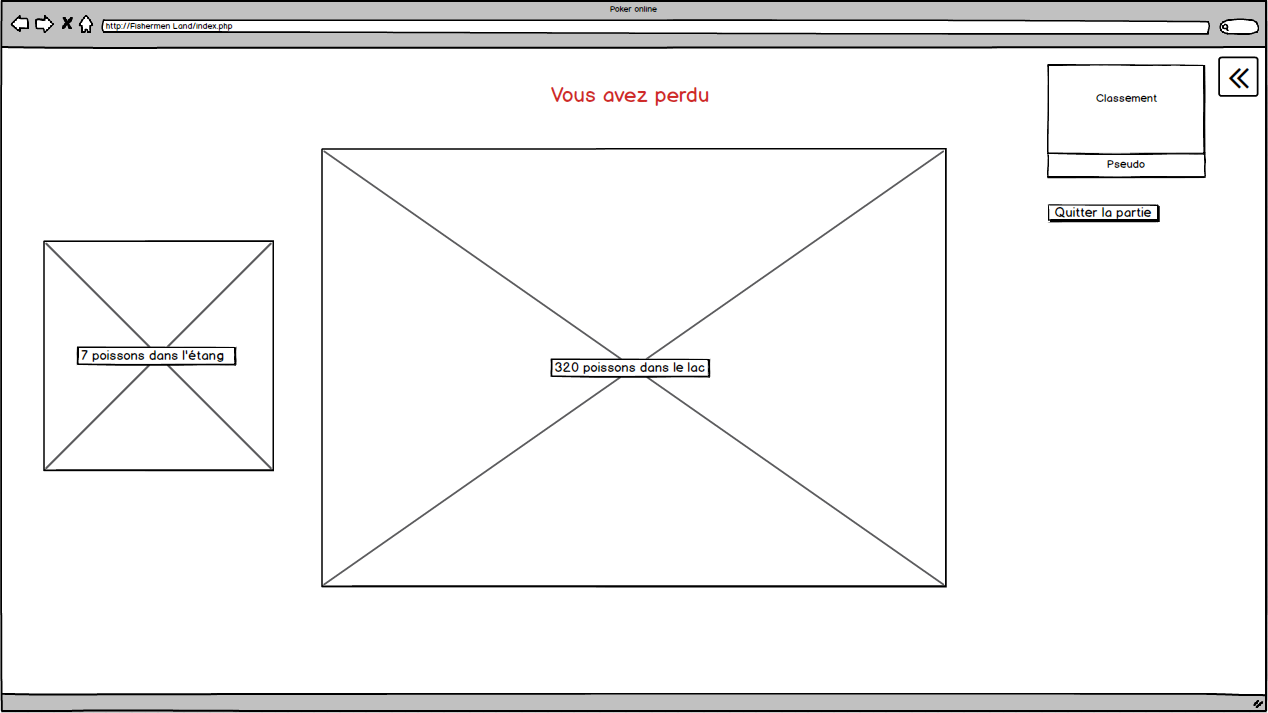
### Partie gagnée

Lorsque la partie est gagnée, un message est affiché quelques brefs instants.



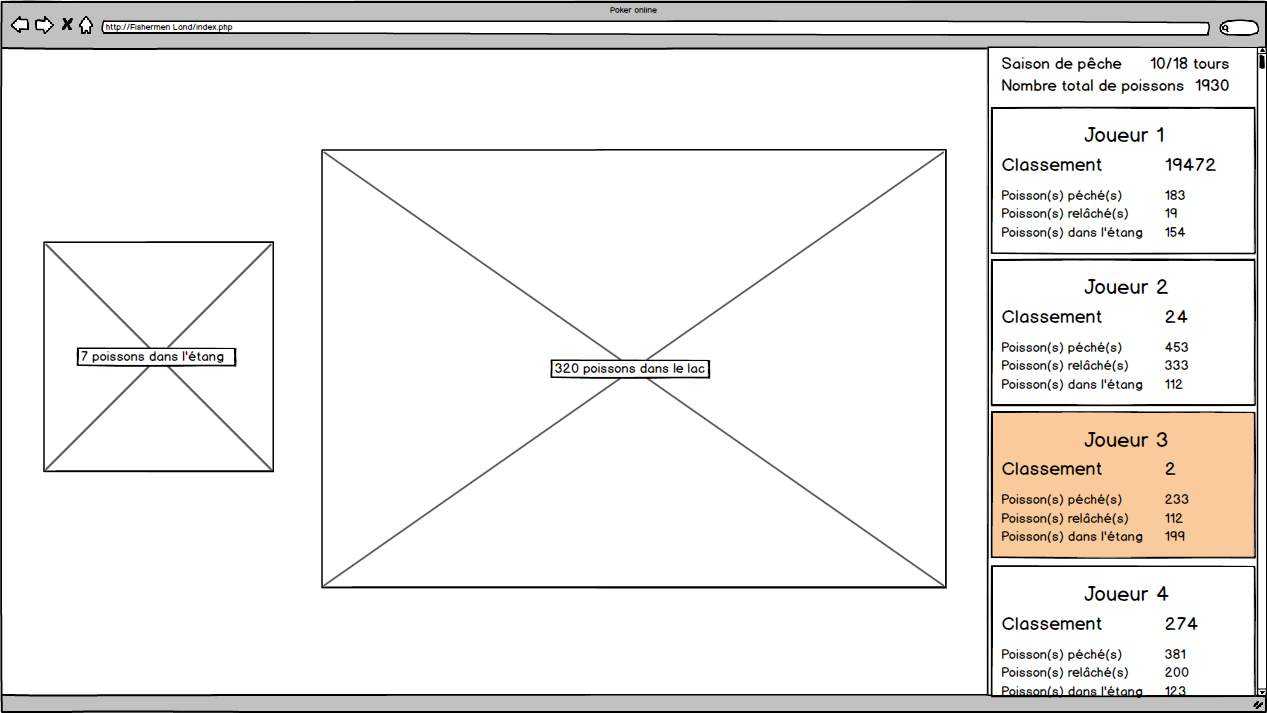
### Partie perdue

Lorsque la partie est perdue, un message est affiché quelques brefs instants.



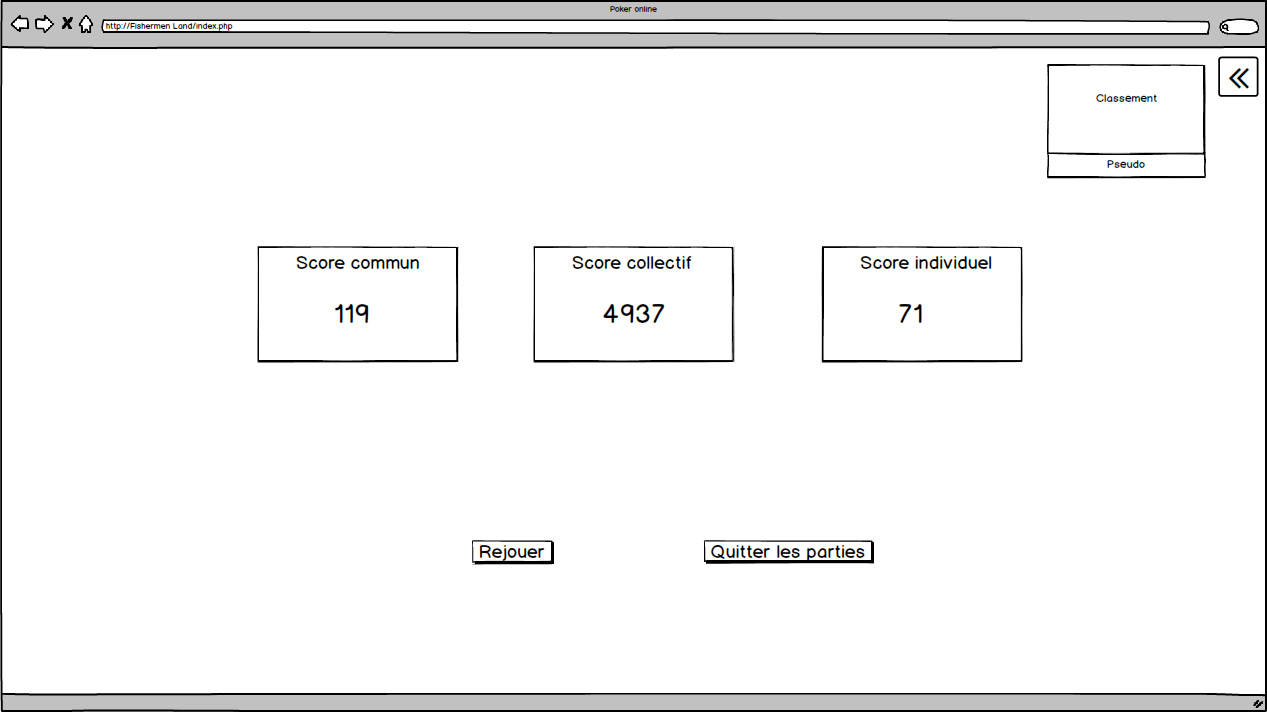
### Consulter les statistiques

Quand le joueur affiche les statistiques, une fenêtre s’ouvre par-dessus la partie, le joueur qui joue est mis en surbrillance.



### Fin d’une saison

Lorsque la partie est finie pour tout le monde, la page affichant les scores est visible et le joueur à le choix de rejouer ou de quitter la partie.



## MLD

**Status :** Contient les différents statuts qui sont attribuables à une place, dans laquelle se trouve le joueur. Contient des entrées comme « Joue », « Relâche des poissons », « En attente », etc…

**Type :** Contient les types de parties existants, tel que « Coopératif », « Imposition » et « Imposition avec forfait ».

**History :** Sauvegarde le score individuel de toutes les parties faites par les joueurs. Permet également de calculer le classement du joueur.

### Particularité 1

Les données stockées dans la table « Settings » permettent des paramétrer les parties. Lorsqu’un paramètre est modifié directement sur le jeu, les données stockées dans la table « Game » sont automatiquement modifiées, cependant les parties qui sont en cours ne se verront pas modifiées avant la fin de la partie.

### Particularité 2

Lorsqu’une partie est complète, c’est au premier joueur d’initialiser la partie. Dans le code on va donc regarder que le « OrderPlace » corresponde à 0, ceci nous permet également de vérifier si ce joueur était le premier joueur précédemment grâce au « FirstPlayerGame ».

### Particularité 3

L’ordre des joueurs est géré avec un modulo (MaxPlayersGame). Dans le cas où un joueur quitte la partie, il va changer le « OrderPlace » de tous les joueurs jouant après lui. Si le joueur est le premier à joueur il va incrémenter d’un tous les « OrderPlace » et se servir du modulo.

## Brève explication du jeu

Une partie est entièrement paramétrable.

Par défaut il existe 3 types de parties :

* Une partie coopérative permet au joueur de relâcher le nombre de poissons qu’il veut à la fin de son tour.
* Une partie en imposition relâche automatiquement à chaque tour un nombre de poissons en fonction des poissons dans l’étang du joueur.
* Une partie en imposition avec forfait se comporte comme la partie en imposition, sauf qu’il ne peut relâcher que 10 poissons maximum par tour.

Dans une partie un joueur doit donc pêcher des poissons dans le lac afin de les mettre dans son étang, il peut également ne pas pêcher, lorsqu’il passe son tour il relâche les poissons selon son mode de jeu puis mange 2 poissons.

Les joueurs jouent toujours à tour de rôle et une fois qu’ils arrivent à la fin d’une saison de pêche (18 tours par personne), ils voient les scores puis on leur attribue un classement si c’est leur troisième partie, sinon on change son classement en fonction de la moyenne des scores individuels obtenus.

Les joueurs peuvent en tout temps consulter les statistiques de la partie. Si le meilleur score commun est battu dans une partie, cette partie compte double dans les statistiques.

# Réalisation

## Dossier de réalisation

Tous les documents produits tout au long du projet se trouvent sur GitHub.

<https://github.com/AlexandreJunod/Fishermen-Land.git>

Voici l’arborescence et une brève explication de l’utilité des dossiers et fichiers.

**Fisherman Land** : Dossier contenant tous les documents produit pendant le TPI

* **Code** : Contient tout ce qui permet de faire fonctionner le jeu
  + **controller** : Partie logique du code
    - **frontend.php**
  + **function**: Contient du code pour éviter la répétition
    - **function.php**
  + **model** : Exécute des requêtes
    - **frontend.php**
  + **public**: Contient le CSS/JS et images
    - **css**
      * **style.css**
    - **images**
    - **js**
  + **view** : Contient les différentes vues que l’utilisateur pourra afficher
    - **frontend**
      * **AdminView.php**: Contient le bouton de « paramétrage »
      * **EndGameView.php**: Contient la page de scores
      * **GamePlayingView.php**: Bouton pour pêcher
      * **GameReleasingView.php**: Bouton pour relâcher
      * **GameView.php**: Vue de la partie, quand on en a rejoint une
      * **HomeView.php** : Vue de la page d’accueil
      * **InfoPlayerView.php** : Affiche les infos du joueur loggé
      * **LoginView.php** : Vue de la page de connexion
      * **SettingsView.php**: Vue de la page de paramétrage
      * **SignupView** : Vue de la page d’inscription
      * **Template.php**: Est appelé par les vues afin de les afficher
  + **fishermenland.sql**: Script de création de base de donnée
  + **index.php** : Routeur, vérifie que l’affichage demandé peut-être fourni
* **Documentation**: Contient toute la documentation du TPI
* **Readme.md**: Manuel d’installation, créer automatiquement par GitHub

## Description des tests effectués

|  |  |  |  |
| --- | --- | --- | --- |
| Scénario | 30.05 | 05.06 | 07.06 |
| 2.8.1.1 Création d’un compte | Dév => OK | Dév => OK | Dév => OK |
| 2.8.1.2 Connexion à un compte | Dév => OK | Dév => OK | Dév => OK |
| 2.8.2.1 Rejoindre une partie | Dév => OK | Dév => OK | Dév => OK |
| 2.8.3.1 Déroulement d’un tour en « coopératif » | Dév => KO | Dév => OK | Dév => OK |
| 2.8.3.2 Déroulement d’un tour en « imposition » | Dév => KO | Dév => OK | Dév => OK |
| 2.8.3.3 Déroulement d’un tour en « imposition avec forfait» | Dév => KO | Dév => OK | Dév => OK |
| 2.8.3.4 Gagner une partie | Dév => KO | Dév => OK | Dév => OK |
| 2.8.3.5 Perdre une partie | Dév => KO | Dév => OK | Dév => OK |
| 2.8.3.6 Perdre une partie et quitter | Dév => KO | Dév => OK | Dév => OK |
| 2.8.3.7 Fin d’un saison | Dév => KO | Dév => OK | Dév => OK |
| 2.8.3.8 Gérer le classement | Dév => KO | Dév => OK | Dév => OK |
| 2.8.4.1 Afficher les statistiques | Dév => KO | Dév => KO | Dév => OK |
| 2.8.5.1 L’administrateur est connecté | Dév => KO | Dév => KO | Dév => KO |

Les tests effectués le 30.05 ont été effectués sur 2 pc différents au sein du même réseau. Le premier pc utilisait Chrome, Firefox et Opera et était connecté sur 3 comptes différents, le deuxième pc était sur Chrome. Les 4 joueurs étaient dans la même partie.

Les tests effectués le 05.06 ont été effectués de la manière que les précédents. Certains messages ne sont pas totalement égaux à ceux cités dans les scénarios, par exemple on ne dit pas au joueur combien il peut pêcher, on lui dit qu’il ne peut pas pêcher plus que ce qu’il possède. Le site ne dit pas également à un joueur qu’il a gagné une partie lorsqu’il est le dernier en vie, car il n’y a pas de gagnant.

Les tests effectués le 07.06 ont été effectués de la même manière que les précédents. Le dernier objectif est KO, car les paramètres ne s’appliquent pas directement après le changement d’une valeur.

## Erreurs restantes

* Lors d’une fin de partie, le joueur qui joue en dernier enregistre la partie 2 fois. S’il fait un record il l’enregistre 4x. Il y a une erreur dans l’architecture qui fait que le code s’exécute une fois de trop.
* Au niveau de l’affichage, lorsque le joueur joue il exécute le code une deuxième fois et change la div à sa valeur par défaut, il y a donc 1 joueur qui disparaît à l’affichage.
* Solution : Trouver la fonction qui exécute deux fois le code du joueur qui joue et rajouter des conditions afin que le code ne s’exécute qu’une fois
* Lorsque tous les joueurs sont éliminés, la partie ne se termine pas directement, le joueur n’est pas éliminé et peut encore jouer une fois. Ceci est un problème d’architecture mais peut-être utiliser comme un « Bonus » pour le joueur qui survit
* Solution : Au lieu de mettre la vérification qui indique que la partie au tour de jeu du joueur, on la déplace avant que le joueur passe, donc au changement de tour.
* Il n’y a pas de vrai sécurité dans le jeu, celle-ci n’as pas été implémentée afin que tout mon temps soit consacré au jeu. Un simple guillemet dans les formulaires de connexion et inscription font crash le code
* Solution : Rajouter un quote
* Suite à un changement de paramètres, les parties ne se mettent pas à jour automatiquement, il faut jouer une partie afin de réinitialiser la partie.
* Solution : Exécuter la réinitialisation des parties vides également quand on change des paramètres

## Liste des documents fournis

* Planification initiale (inclus dans la documentation)
* Un rapport de projet (inclus dans la documentation)
* Un journal de travail
* Le code du jeu
* Procédure d’installation (inclus dans la documentation)

# .Conclusions

Le projet c’est dérouler correctement, j’ai tenu une planification à jour tout au long du projet sur Trello, j’ai eu l’occasion d’utiliser scrum for Trello et le mettre à jour régulièrement et j’ai rendu le jeu paramétrable au maximum.

Au niveau de l’architecture MVC j’ai beaucoup progressé et je comprends mieux le fonctionnement et le but de cette méthode de programmation, mais je me suis rendu compte qu’au long terme j’ai commencé à rajouter un peu trop de logique dans plusieurs fonctions différentes et ceci m’a fait perdre du temps.

Si c’était à refaire je me concentrerais plus sur l’architecture, j’éviterais d’avoir trop de variables inutiles et des requêtes redondantes. Je pourrais également faire le maximum de logique sur un minimum de fonctions, et me servir des autres fonctions seulement afin d’exécuter du code et non vérifier la permission de l’exécuter.

J’ai eu beaucoup de difficultés au niveau de jQuery, et j’ai fini par prendre une autre solution avec de ne pas perdre de temps. L’autre solution était d’utiliser du JavaScript, et j’ai eu l’occasion d’en découvrir plus sur ce langage et ce qu’il est possible de faire et j’y ai pris beaucoup de plaisir à l’utiliser, et je pense par la suite en utiliser plus souvent.

La première amélioration à faire est d’ajouter un bouton « Réinitialiser » dans la page de paramètres afin de mettre toutes les valeurs en valeurs par défaut, car celui-ci était dans l’use case « 2.8.5 Paramétrage du jeu »

Le projet peut encore continuer d’évoluer, je pense qu’il serait bien de commencer par permettre à l’administrateur de gérer les différents messages d’erreurs dans la page de paramétrage.

Permettre également à l’admin de choisir le nombre en % de poissons relâchés dans les autres modes, de rajouter des administrateurs directement depuis le site, de créer des nouveaux modes de jeux et de crées et supprimer des parties manuellement.

# Annexes

## C:\wamp64\www\Fishermen Land\Documentation\Images\Trello 15.05.2018.PNGPlanification initiale

## Résumé du rapport du TPI / version succincte de la documentation

Lorsque le TPI a débuté j’ai reçu comme consigne de créer un jeu, celui-ci est stratégique, le but de ce TPI est de voir comment les joueurs gèrent le lac sans communication. Pour réaliser correctement ce je devais donc comprendre comment fonctionnait le jeu.

J’ai tout d’abord commencer par faire l’analyse, définir les différents scénarios et uses cases et m’assurer que le jeu fonctionne à ce que le client voulait. Une fois que le fonctionnement du jeu correspondait à ce que le client voulait, j’ai commencé à créer le jeu.

En tout premier lieu j’ai créé les différentes pages et je n’ai pas fait de mise en page, par la suite j’ai implémenté la partie logique afin de pouvoir jouer une partie, puis j’ai améliorer au maximum que je pouvais le jeu tout en corrigeant les bugs en parallèle.

La documentation a continué d’évoluer tout au long du projet, celle-ci a été livrée 2 par semaine au experts et au chef de projet et au client et une démo par semaine était planifiée avec le chef de projet.

Le fonctionnement du jeu correspond aux attentes qui m’ont été données, mais il reste un bug qui vient gêner la vue de l’utilisateur.

## Sources – Bibliographie

Aide externe : M. Carrel

<https://openclassrooms.com/>

<https://stackoverflow.com/>

<https://www.w3schools.com/>

<https://pbnaigeon.developpez.com/>

<https://developer.mozilla.org/fr/>

## Journal de bord

|  |  |
| --- | --- |
| Date | Evénement |
| 08.05.2018 | Avec le client nous avons décidé qu’une partie commencerait lorsqu’il y a 6 joueurs dans une partie. Cette valeur devra être modifiable plus tard |
| 09.05.2018 | La planification initiale a été approuvée par M. Montemayor |
| 09.05.2018 | Suite au retour de M. Gruaz, l’extension web scrum for trello est utilisé sur le trello |
| 09.05.2018 | Tentative de planification de la démo avec M. Egger |
| 15.05.2018 | Envoi par mail des use cases, scénarios et maquette a M. Carrel |
| 17.05.2018 | Expliquer à M. Carrel le déroulement prévu pour une partie et lui demander si ceci correspond à sa vision des choses. Dans une partie on commence par pêcher, relâcher, manger puis ensuite de nouveaux poissons apparaissent |
| 18.05.2018 | Démo présentée à M. Egger et prendre notes des retours reçus |
| 23.05.2018 | Rencontre avec M. Gruaz et prise en note des différents points à aborder dans la documentation |
| 25.05.2018 | Démo présentée à M. Egger |
| 29.05.2018 | Demande d’aide au maître de classe, qui est également le client (M. Carrel). L’aide a été demandée pour des problèmes techniques |
| 01.06.2018 | Présentation de la démo à M. Egger |

## Manuel d'Installation

Afin de pouvoir utiliser le jeu réaliser lors de ce TPI, il existe deux procédures, une prévue pour les postes en Windows, l’autre pour les Mac.

### Windows

* Installer WampServer
* Démarrer WampServer
* Télécharger le dossier de réalisation et le placer dans C:\wamp64\www
* Lancer phpMyAdmin depuis Wamp
* Importer le script fishermenland.sql qui se trouve dans Fishermen Land\Code
* Ouvrir un navigateur et se rendre à l’adresse : [localhost/Fishermen%20Land/Code/](http://localhost/Fishermen%20Land/Code/)

### Mac

* Installer Mamp
* Démarrer Mamp
* Télécharger le dossier de réalisation et le placer dans Applications\M\htdocs
* Lancer phpMyAdmin à l’adresse localhost:8888/phpMyAdmin/
* Importer le script fishermenland.sql qui se trouve dans Fishermen Land\Code
* Ouvrir la page frontend.php dans Fishermen Land\Code\model
* Modifier la ligne 10 pour avoir : $password = 'root';
* Ouvrir un navigateur et se rendre à l’adresse : [localhost:8888/Fishermen%20Land/Code/](http://localhost/Fishermen%20Land/Code/)

Pour se connecter avec le compte administrateur, il faut utiliser le pseudo : Admin et le mot de passe : Admin

## Archives du projet

Le projet est enregistré sur 3 CD, il est également archivé sur le GitHub