

# Projet Fishermen

---

## Table des matières

1	Analyse préliminaire.....	3
1.1	Introduction.....	3
1.2	Objectifs.....	3
1.3	MATÉRIEL ET LOGICIEL À DISPOSITION.....	4
1.4	PRÉREQUIS.....	4
1.5	Planification initiale.....	5
1.6	Planification détaillé.....	5
2	Analyse / Conception .....	6
2.1	Concept.....	6
2.1.1	Diagramme de flux.....	6
2.2	Maquettes .....	10
2.3	Stratégie de test.....	12
2.3.1	Test Unitaire.....	12
2.3.2	Tests d'acceptations.....	12
2.4	Risques techniques .....	13
2.5	Planification finale.....	13
2.6	Dossier de conception.....	14
3	Réalisation .....	14
3.1	Dossier de réalisation.....	14
3.1.1	Scripts.....	14
3.2	Répertoires.....	16
3.3	Versions.....	17
3.4	Description des tests effectués.....	19
3.5	Erreurs restantes .....	19
3.6	Liste des documents fournis.....	19
4	Conclusions.....	20
5	Annexes .....	21
5.1	Résumé du rapport du TPI / version succincte de la documentation.....	21
5.2	Sources – Bibliographie.....	21
5.3	Journal de travail .....	21
5.4	Manuel d'Installation.....	22
5.5	Manuel d'Utilisation .....	22
5.6	Archives du projet.....	22

---

## 1 Analyse préliminaire

### 1.1 Introduction

*Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.*

Etant très intéressé par le développement et ayant de l'expérience sur Unity, J'ai demandé que mon projet TPI s'effectue dans ce domaine.

Le projet consiste à créer, à l'aide du moteur de jeu Unity, en langage C#, un Serious Game de simulation/gestion économique de pêche. Le joueur pourra pêcher différentes espèces de poissons dans différentes zones, et pouvoir les revendre pour gérer son économie.

Des poissons trop petits ou trop jeune peuvent être pêché. Il faudra, alors les relâcher, Auquel cas, le joueur recevra une amende.

### 1.2 Objectifs

Le jeu contiendra :

- Différentes zones de pêches
  - Le joueur doit payer certaines zones de pêches pour y accéder
- Un système de pêche
  - Quand un poisson est pêché, le moment où il a été pêché est indiqué sur le poisson
  - Chaque poisson pêché aura un âge et une taille aléatoire
  - La possibilité de relâcher un poisson
- Un inventaire
  - Chaque poisson pêché devra être mis dans un inventaire qui pourra être regardé
- 5 espèces de poissons différentes
- Un inventaire

- 
- Chaque poisson pêché devra être mis dans un inventaire. L'inventaire pourra être consulté à n'importe quel moment
  - Système d'économie
    - Le joueur possédera un compteur qui donnera l'état actuel de son argent
  - Un système de Vente
    - Chaque poisson pourra être vendu
    - Si un poisson vendu est trop jeune ou trop petit, le joueur devra payer une amende
  - Chaîne de froid
    - Le moment où le poisson a été pêché est indiqué sur le poisson. Dépassé un certain moment le poisson devra être jeté, sinon le joueur recevra une amende

### 1.3 MATÉRIEL ET LOGICIEL À DISPOSITION

- 1 ordinateur du CPNV
- Unity 2020
- Visual Studio 2020
- Visual Studio Code
- Suite office

### 1.4 PRÉREQUIS

- Formation de base du CPNV
- Connaissances en POO
- Maîtrise de Unity

## 1.5 Planification initiale

Fishermen  
Kevin Gacon

lun, 05.02.2022

1

Kevin Gacon

				2 mai 2022							9 mai 2022							16 mai 2022							23 mai 2022							30 mai 2022			
				2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
				l	m	j	v	s	d	l	m	j	v	s	d	l	m	j	v	s	d	l	m	j	v	s	d	l	m	j	v	s	d	l	m
PROGRESSION	DÉBUT	FIN	JOURS																																
PLANIFICATION																																			
Planification initiale	0%	02.05.2022	02.05.2022	1																															
Discussion avec l'expert	0%	02.05.2022	02.05.2022	1																															
Création des Uses Cases et Scenario	0%	03.05.2022	03.05.2022	1																															
Création des diagrammes de flux	0%	03.05.2022	05.05.2022	3																															
Création des Sprints sur JiraScrum	0%	03.05.2022	05.05.2022	3																															
Conception du visuel du jeu	0%	05.05.2022	05.05.2022	1																															
Création de l'environnement de travail	0%	05.05.2022	05.05.2022	1																															
PROJET																																			
Recherche d'élément qui me serviront pour les décors	0%	05.05.2022	06.05.2022	2																															
Système d'argent pour le joueur	0%	06.05.2022	09.05.2022	4																															
Création des zones de pêches	0%	06.05.2022	09.05.2022	4																															
Système de pêche	0%	09.05.2022	12.05.2022	4																															
Création des différents poissons	0%	12.05.2022	13.05.2022	2																															
Création de l'inventaire	0%	16.05.2022	19.05.2022	4																															
Système de vente de poisson	0%	19.05.2022	20.05.2022	2																															
Création de la gestion du froid pour les poisson	0%	23.05.2022	24.05.2022	2																															

## 1.6 Planification détaillée

## 2 Analyse / Conception

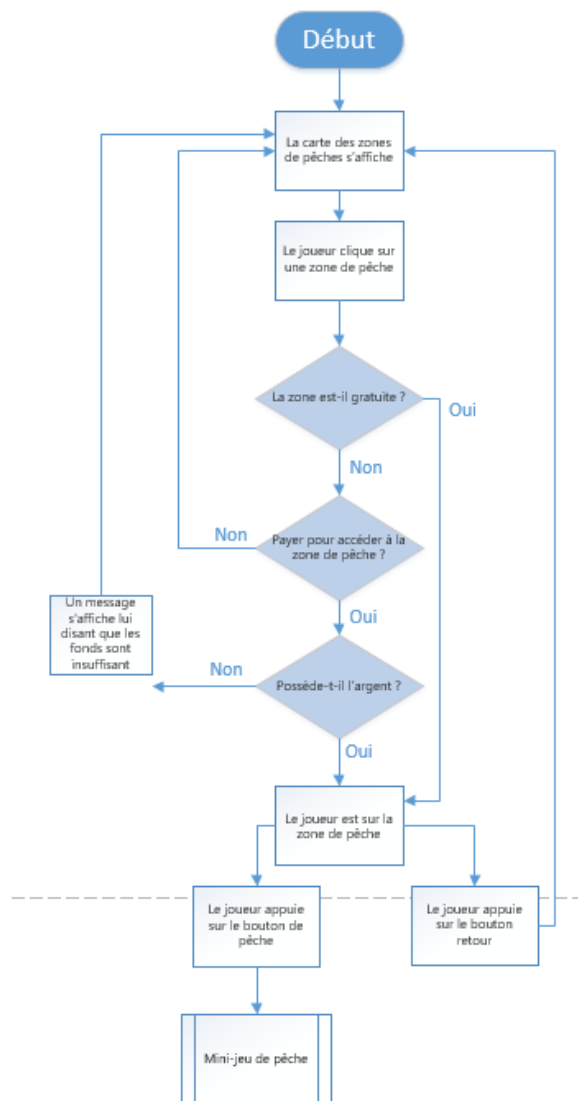
### 2.1 Concept

#### 2.1.1 Diagramme de flux

Afin de mieux comprendre les fonctionnalités demandées, j'ai créé plusieurs diagrammes de flux, pour avoir une représentation graphique. Et pouvoir savoir comment implémenter les fonctionnalités.

##### 2.1.1.1 Système de zones de pêche

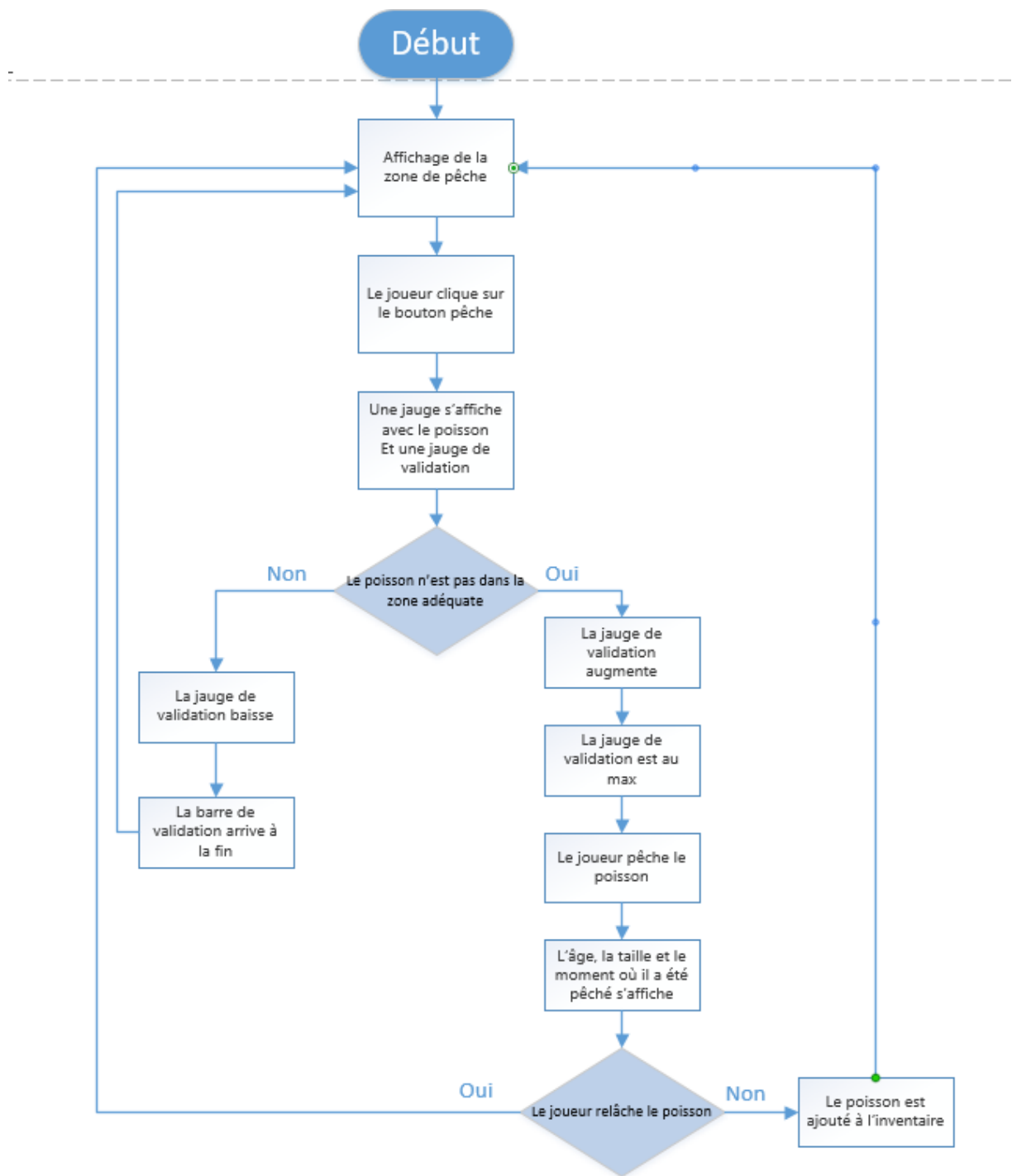
Pour pouvoir pêcher, le joueur doit accéder au zone de pêche, des conditions peuvent être présente avant d'accéder à ces zones. Certaines zones nécessitent d'être payé pour y accéder.



1 : diagramme de zones de pêche

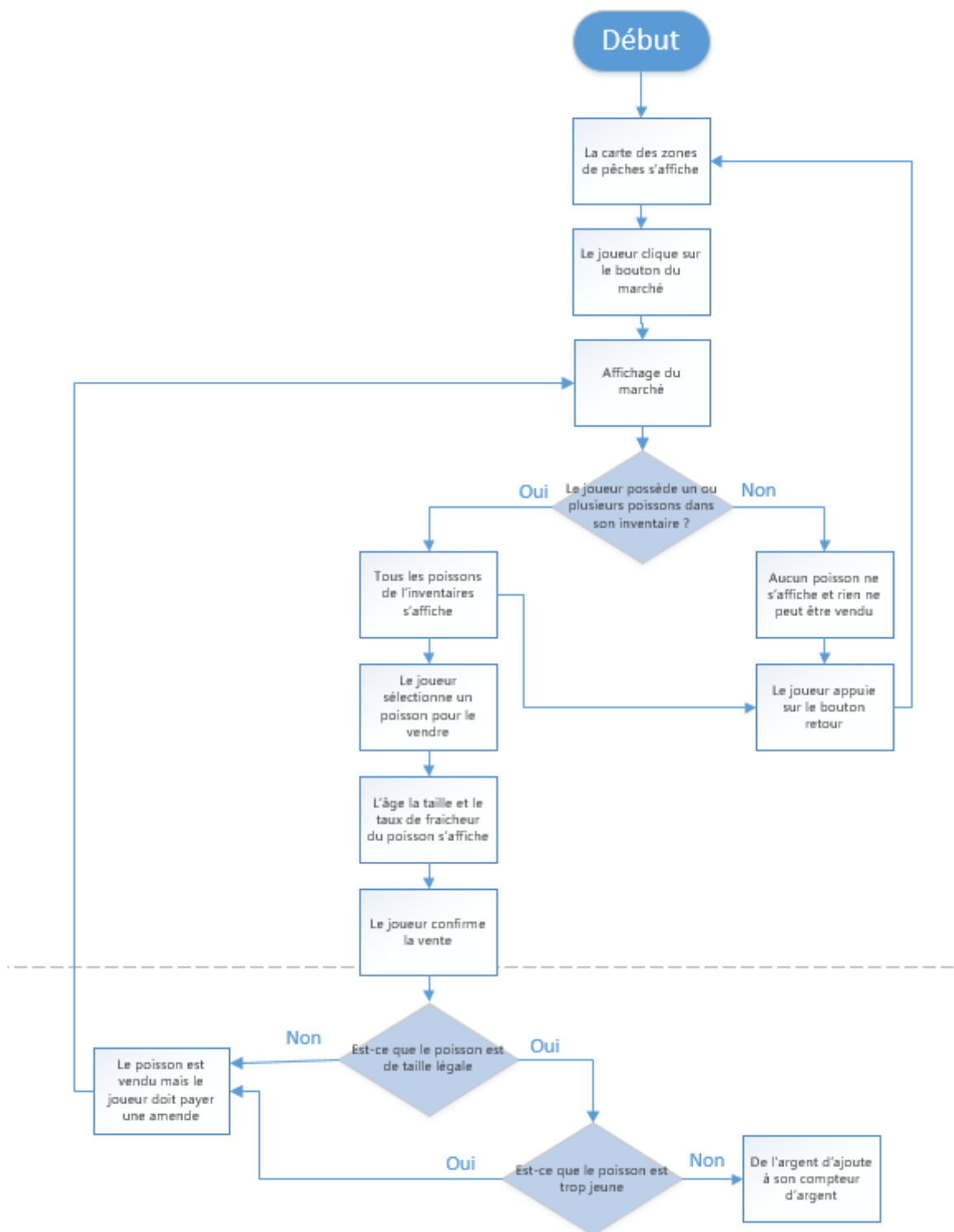
### 2.1.1.2 Jeu de la pêche

Lorsque le joueur se trouve sur une zone de pêche, il peut commencer à pêcher.



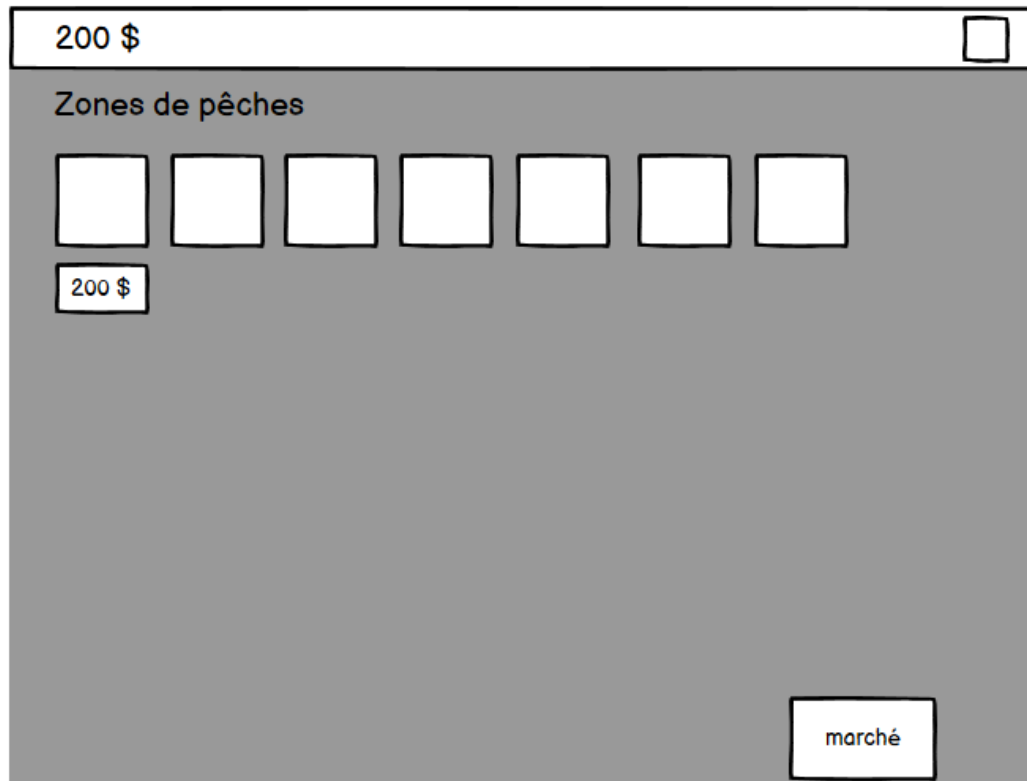
2 : Mini-Jeu de pêche



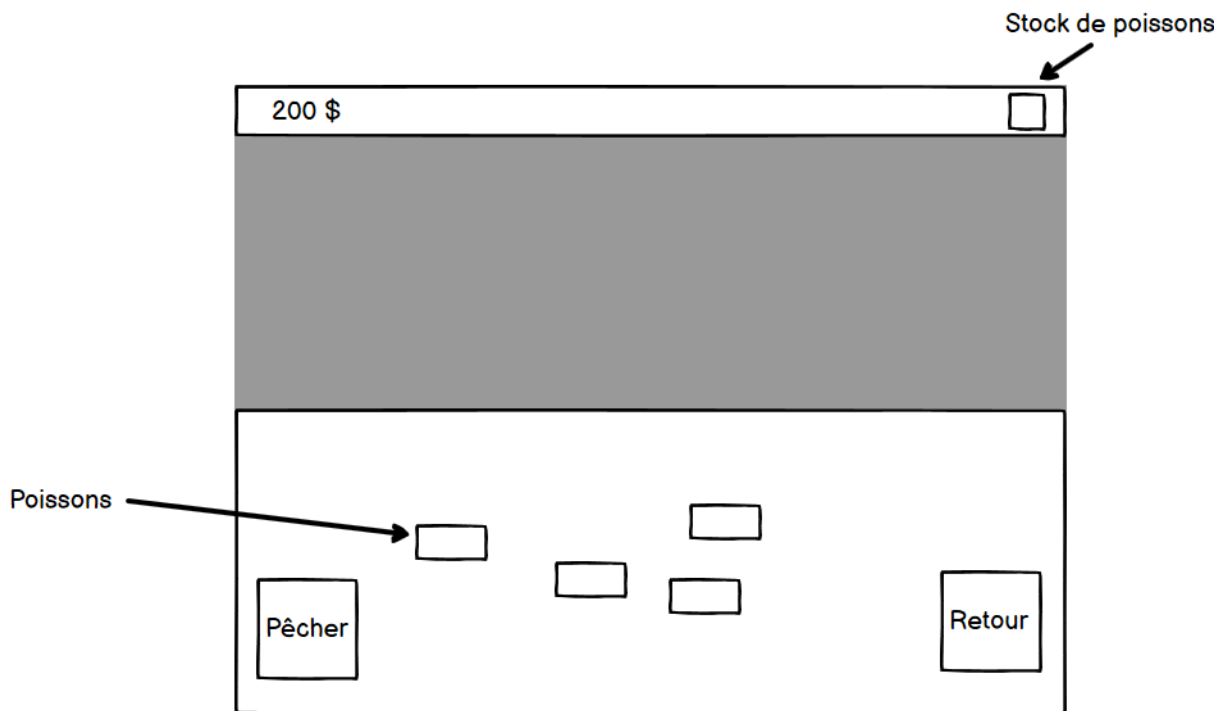


3 : Vente de poissons

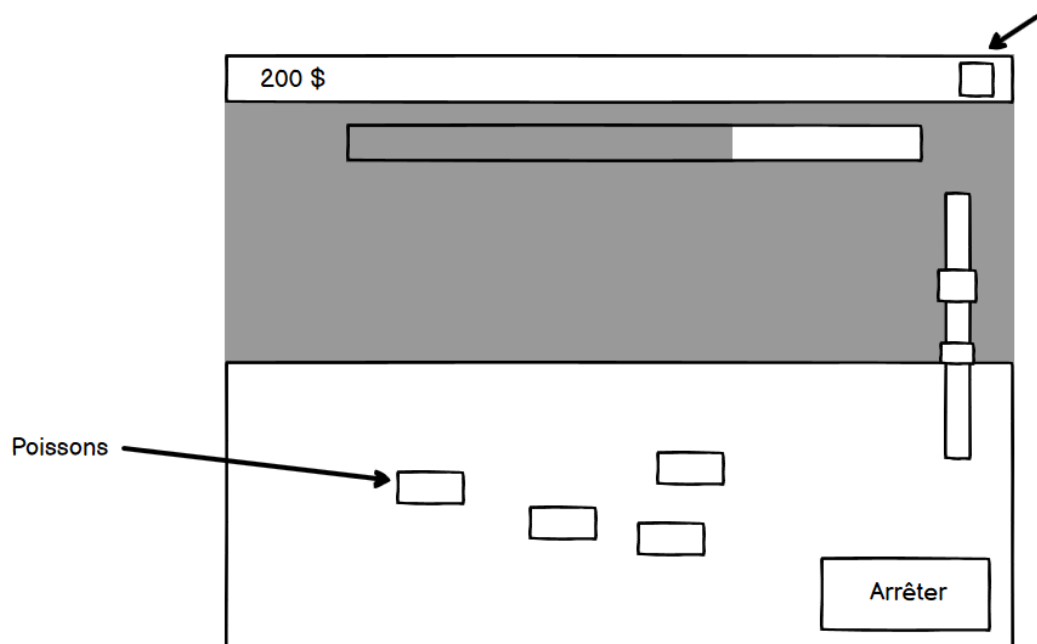
## 2.2 Maquettes



4 : Maquette menu de sélection de Zones de pêches



5: Maquette d'une zone de pêches



6 : Menu de ventes de poissons

200 \$

Stock de poissons

Nom du poisson

Nom du poisson

Info du poissons

Relacher

## 2.3 Stratégie de test

### 2.3.1 Test Unitaire

Afin de correctement tester une fonctionnalité, un test unitaire est implémenter et sera effectué sur la fonctionnalité de vente de poissons. Car c'est la partie qui a le plus de vérifications. Le poisson doit avoir des caractérisés suffisante pour être vendu.

### 2.3.2 Tests d'acceptations

Plusieurs tests d'acceptations sont inscrits sur IceScrum. Je demanderais à des camarade de classe et proches de tester et de me faire un retour sur le jeu (éventuel bug et prise en main du jeu).

<https://cloud.icescrum.com/p/FISHERMEN/#/planning/200015/sprint/200016/story/506811/tasks>



*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

## 2.6 Dossier de conception

*Fournir tous les document de conception:*

- *le choix du matériel HW*
- *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
- *le choix des outils logiciels pour la réalisation et l'utilisation*
- *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...*
- *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
- *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

## 3 Réalisation

### 3.1 Dossier de réalisation

#### 3.1.1 Scripts

##### 3.1.1.1 DontDestroyOnLoad

Ce script sert à garder les objets des poissons entre chaque scène. Ces objets gardent les données des poissons (tailles, âges, prix...).

##### 3.1.1.2 DayTimeSystem

Ce script sert à afficher le temps et à définir le moment où le poisson a été pêché.

##### 3.1.1.3 EconomySystem

Ce script sert à garder en mémoire l'argent actuel du joueur. Il contient, également, les fonctions qui permettent d'ajouter et de retirer de l'argents.

#### 3.1.1.4 FlshData

Script qui permet de créer des ScriptableObjects de poissons directement dans les assets (dossier du jeu). Un ScriptableObjects sert à enregistrer plusieurs données d'un objet indépendamment des instances de classe. Il me permet ainsi d'utiliser les mêmes données pour plusieurs objets différents (Une espèce de poissons).

#### 3.1.1.5 FishingArea

Ce script sert à la gestion de sélection de zone de pêche. Il permet de savoir si une zone est payante ou non. Si l'argent est suffisant ou non. Et savoir quelle scène de zone de pêche il faut charger.

#### 3.1.1.6 FishingGame

Ce Script gère tout la partie mini-jeu de la pêche. Il vérifie si le joueur a gagné ou non. Et il ajoute un poisson lorsque le jeu est gagné.

#### 3.1.1.7 GUINavigation

Permet de charger les bonnes scènes lorsque le joueur navigue dans les différents menus. Egalement, il permet d'ouvrir son inventaire.

#### 3.1.1.8 InventoryFishes

Script qui contient les fonctions qui permettent d'actualiser l'inventaire. Cette fonction est appelée à chaque fois que l'inventaire est ouvert dans le script GUINavigation.

#### 3.1.1.9 MyFishSpecificData

Script qui permet de générer des valeurs que je ne peux pas stocker dans FishData. Car les données de « tailles » et d' « âges » doivent être aléatoire. J'ai, donc, stocker des valeurs maximum et minimum dans FishData et je génère des valeurs aléatoires entre la valeur maximum et minimum que je stock dans MyFishSpecificData.

#### 3.1.1.10 SelectFish

Ce script contient la fonction qui permet de sélectionner plusieurs poissons dans la shop. Ça permet de vendre plusieurs poissons en même temps.

#### 3.1.1.11 SellFishesSystem

Script qui gère tout l'affichage de la partie vente de poissons. Et contient la fonction pour vendre tous les poissons sélectionnés.

#### 3.1.1.12 SwimmingFish

Ce script permet de faire bouger aléatoirement les poissons dans l'océan

## 3.2 Répertoires

### 3.2.1.1 Assets

Dossier principal du jeu généré par Unity. Il contient tous les dossiers, fichiers et scripts utilisé pour la création du jeu.

### 3.2.1.2 Fishes

Dossier qui contient les playerprefs des différentes espèces de poissons. Chacun possède le nom de l'espèce, l'âge, la taille...

### 3.2.1.3 Prefab

Dossier qui contient tous les objets préfabriqués utilisé pour le jeu.

### 3.2.1.4 Ressources

Répertoire qui contient les dossiers d'élément utiles pour le jeu (Images, Fonts...)

#### 3.2.1.4.1 Fonts

Ce répertoire contient tous les polices d'écriture que j'utilise sur la plupart de mes projets.

#### 3.2.1.4.2 FishImages

Contient toutes les images de poissons.

#### 3.2.1.4.3 Images

Contient toutes les autres images utilisé dans le jeu.

### 3.2.1.5 Scènes

Répertoire qui contient l'ensemble des scènes utiliser pour le projet.

### 3.2.1.6 Scripts

Ce répertoire contient tous les scripts du jeu.

### 3.2.1.7 TextMeshPro

Dossier généré par le package installé TextMeshPro

## 3.3 Touches de DEBUG

Pour tester plus rapidement certaines



Décrire la réalisation "physique" de votre projet

- les répertoires où le logiciel est installé
- la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)
- les versions des systèmes d'exploitation et des outils logiciels
- la description exacte du matériel
- le numéro de version de votre produit !
- programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.

NOTE : Evitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

### 3.4 Versions



7 : Version de Unity



8 : Version de Windows

À propos de Microsoft Visual Studio

? X

# Visual Studio

[État de la licence](#)  
[Termes du contrat de licence](#)

Microsoft Visual Studio Enterprise 2019  
Version 16.11.9  
© 2021 Microsoft Corporation.  
Tous droits réservés.

Microsoft .NET Framework  
Version 4.8.04084  
© 2021 Microsoft Corporation.  
Tous droits réservés.

Produits installés :

Microsoft MI-Based Debugger – 1.0  
Outils Azure App Service v3.0.0 – 16.11.94.52318  
Outils C# – 3.11.0-4.21602.3+fc14a1355c0461af5110b74b26f6478f22d26565  
Outils TypeScript – 16.0.30526.2002  
Outils Visual Basic – 3.11.0-4.21602.3+fc14a1355c0461af5110b74b26f6478f22d26565  
Package hôte de l'adaptateur de débogage de Visual Studio Code – 1.0  
ProjectServicesPackage Extension – 1.0  
Visual F# Tools – 16.11.0-beta.21514.6+b6c2c4f53ea3a08fa603ea93d6d2f808a62a21d1  
Visual Studio Tools pour Unity – 4.11.4.0

Copier les informations

Informations système

DxDiag

Détails sur le produit :

ASP.NET and Web Tools 2019

Avertissement : ce logiciel est protégé par la loi relative au droit d'auteur et par les conventions internationales. Toute reproduction ou distribution partielle ou totale de ce logiciel, par quelque moyen que ce soit, est strictement interdite. Toute personne ne respectant pas ces dispositions se rendra coupable du délit de contrefaçon et sera passible des sanctions pénales prévues par la loi.

OK

9 : Version de Visual Studio

---

### 3.5 Description des tests effectués

#### 3.5.1 Tests d'acceptations

Le choix de zones de pêches fonctionne correctement

- Lorsque le joueur choisit une zone de pêche ^

*Pour chaque partie testée de votre projet, il faut décrire:*

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

### 3.6 Erreurs restantes

- Diagrammes de flux
- Journal de travail
- Planification initiale
- Maquettes

*S'il reste encore des erreurs:*

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

### 3.7 Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

## 4 Conclusions

*Développez en tous cas les points suivants:*

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

---

## 5 Annexes

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

### 5.2 Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur) ... Et de toutes les aides externes (noms)*

Nom des zones de pêches : <https://www.normandiefraicheurmer.fr/recettes-et-si-on-cuisinait/origine-des-produits-de-la-mer/43-zones-de-peche-monde.html>

Image du bateau : [http://www.clipartpanda.com/clipart\\_images/silver-king-custom-marine-42502007](http://www.clipartpanda.com/clipart_images/silver-king-custom-marine-42502007)

Image ombre de poisson :  
<https://www.deviantart.com/absurdwordpreferred/art/Little-Fish-png-147138899>

<https://fr-academic.com/dic.nsf/frwiki/2022609>

<https://www.lapechetechnique.fr/tailles-legales-de-capture-en-mer-2022/>

<https://www.futura-sciences.com/planete/questions-reponses/poisson-poissons-10-especes-plus-pechees-9426/>

images des poissons : <https://www.hiclipart.com/>

info pour les poissons : <http://wpbp.sinica.edu.tw/>

<https://en.wikipedia.org/wiki/Anchovy>

Temps de fraîcheur <https://www.bloc-notes-culinaire.com/2019/08/criteres-de-fraicheur-du-poisson.html>

### 5.3 Journal de travail

## **5.4 Manuel d'Installation**

## **5.5 Manuel d'Utilisation**

## **5.6 Archives du projet**

*Media, ... dans une fourre en plastique*

A inclure

<https://github.com/KevinGacon/Fishermen-TPI>