

ETML

Space Invaders

P_00

Abiram Muthulingam
04/09/2024

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs.....	3
1.3	Gestion de projet	4
2	Analyse / Conception.....	4
2.1	Gameplay	4
2.2	UX.....	5
2.2.1	Conception centrée utilisateur :	5
2.3	Concept	9
2.4	Analyse fonctionnelle.....	9
2.4.1	Création du Menu principal.....	9
2.4.2	Déplacement gauche ou droite.....	9
2.4.3	Capacité de tir.....	10
2.4.4	Nombre de vies.....	10
2.4.5	Tir	10
2.4.6	déplacement des ennemis.....	10
2.4.7	Nombre de vies des ennemies	11
2.4.8	Lancement du fichier exécutable	11
2.4.9	Déplacement de l'obstacle.....	11
2.4.10	Durée de vie de l'obstacle.....	11
2.4.11	Ajouts d'un 2ème obstacle.....	11
2.4.12	Tirs ennemis	11
2.4.13	tirs de l'obstacle	12
2.5	Stratégie de test.....	12
3	Réalisation.....	12
3.1	Points de design spécifiques	12
3.1.1	13
3.1.2	13
3.1.3	13
3.2	Déroulement.....	13
3.3	Mise en place de l'environnement de travail.....	13
3.4	Description des tests effectués.....	14
3.5	Erreurs restantes	14
4	Conclusions	14
5	Annexes.....	15
5.1	Manuel de référence.....	15
5.2	Journal de travail	15

NOTE L'INTENTION DES UTILISATEURS DE CE CANEVAS :

Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.

De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.

1 Analyse préliminaire

1.1 Introduction

Le but de ce projet, est de réaliser un jeu de tirs en programmation orienté objet en C#. Le jeu est fait en 3 projet différent (UX, OO, DB). Dans ce projet, nous implémentons seulement la partie du gameplay et du menu, qui permettra à l'utilisateur de jouer. Durant le projet, l'outil IceScrum est utiliser pour la gestion du projet, ainsi qu'un journal de travail et d'un rapport expliquant les détails du jeu.

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.

Ces éléments peuvent être repris des spécifications de départ.

1.2 Objectifs

L'objectif de ce projet, est de créer un jeu de tirs en programmation orienté objet en C#, avec comme conception des ennemis, le joueur (de l'utilisateur), ainsi que des obstacles qui serviront de protection. A la fin de ce projet, il doit y a avoir un jeu accessible sans bug.

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

1.3 Gestion de projet

L'outil IceScrum, est utilisé durant le projet, afin de pouvoir gérer la gestion de projet. IceScrum permet de planifier les tâches à effectuer, et aussi d'avoir des tests pour vérifier que le jeu est fonctionnel. Un journal de travail est mis à jour à chaque semaine, expliquant le travail effectué.

Ce chapitre décrit la méthode de gestion de projet utilisée, ainsi que les éventuelles particularités requises par le contexte et/ou le chef de projet

2 Analyse / Conception

2.1 Gameplay

Le joueur :

Le joueur contrôlera un vaisseau qui se trouve en bas de l'écran et peut se déplacer à gauche et à droite (flèche gauche, droite ou lettre A, D), mais ne peut pas sortir du cadre du jeu. Le vaisseau n'a pas de munition limitée et peut donc tirer à volonté verticalement. Si le vaisseau se fait toucher par un ennemi, la partie est terminée.

Les ennemis :

Les ennemis apparaissent aléatoirement depuis le haut de l'écran et s'approche du vaisseau pour le toucher afin de le détruire. Il y a 2 types d'ennemis (petit et grand), les petits ont seulement 1 point de vie et les grands ont 3 points de vie.

Les obstacles :

Les obstacles servent de protection et empêchent les tirs du vaisseau (quand le vaisseau tire) et de l'ennemi, mais quand le vaisseau ne tire pas l'obstacle se transforme pour pouvoir tirer en continu (pour aider le joueur) afin de toucher des ennemis. Les obstacles sont destructibles seulement sous transformation de tirs, en recevant 5 tirs (phase de test). Les obstacles se déplacent horizontalement à gauche et à droite.

Niveau 2 :

Le niveau 2 sera plus difficile que le niveau 1 (niveau de base). Les ennemis sont plus rapides et peuvent aussi tirer. Le vaisseau du joueur aura un nombre limité de munition et son projectile est un missile qui fait de dégâts aux alentours, mais qui a un délai plus long que la normal.

- *Le joueur*
- *Les ennemis*
- *Les déplacements*
- *Les niveaux*
- *Le tir*
- *La gestion des vies*
- ...

2.2 UX

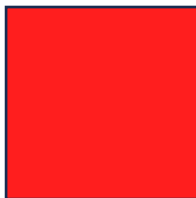
2.2.1 Conception centrée utilisateur :

Création de 2 personas, âgés de 18 et 12 ans pour diverses raisons de jouer au jeu.

Personas 1 : Lisa, âgée de 18 ans, voulant réussir à jouer pour le fun dans un jeu pas très compliqué.

Personas 2 : Marius est un jeune homme de 12 ans, qui a pour but de persévérer dans les jeux, et de devenir le meilleur.

2.2.2 Choix de la palette graphique :



FF1E1E

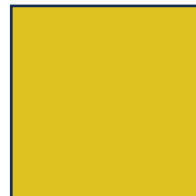


A01A1A

Ces teintes de rouge sont appliquées aux titres, aux textes principaux, et aux éléments de sélection, afin d'assurer une bonne visibilité grâce à un fond noir.

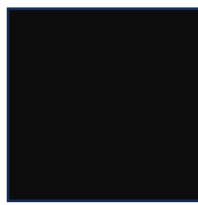


E6EED6



DEC222

Ces couleurs sont réservées au texte dans la section *High Score*, créant un contraste permettant de lire facilement sur le fond sombre.



0D0C0C



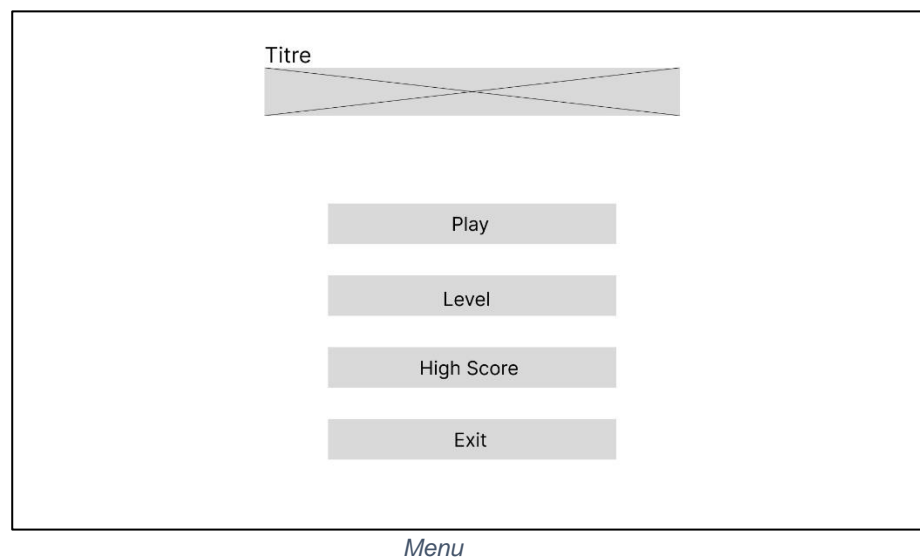
FFFFFF

Le noir est utilisé pour le background et le blanc est utilisé sur le texte pour les choix de sélection.

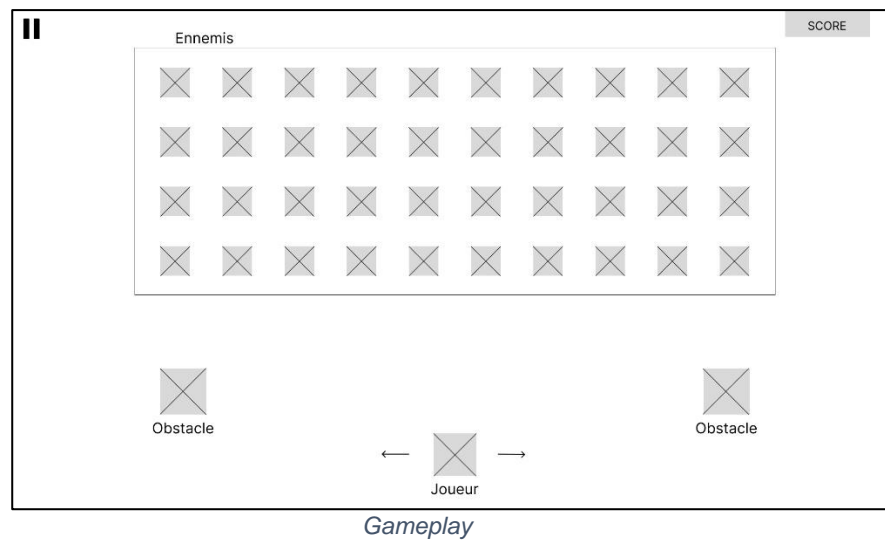
2.2.3 Accessibilité :

Le jeu est conçu pour être utilisable uniquement avec les touches du clavier et peut se jouer d'une seule main. Pour les personnes malvoyantes, le contraste des couleurs a été soigneusement choisi pour rendre le texte facilement visible et suffisamment grand pour être lisible. Un bouton haut-parleur pourrait être également ajouté pour permettre une lecture audio sur les textes affichés. Chaque bouton est clairement étiqueté avec un mot décrivant précisément sa fonction.

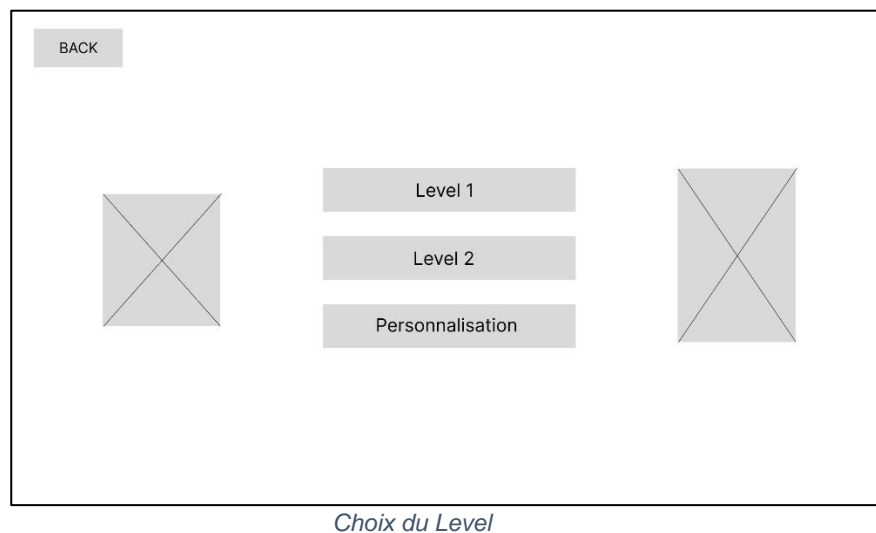
2.2.4 Chapitre de conception :



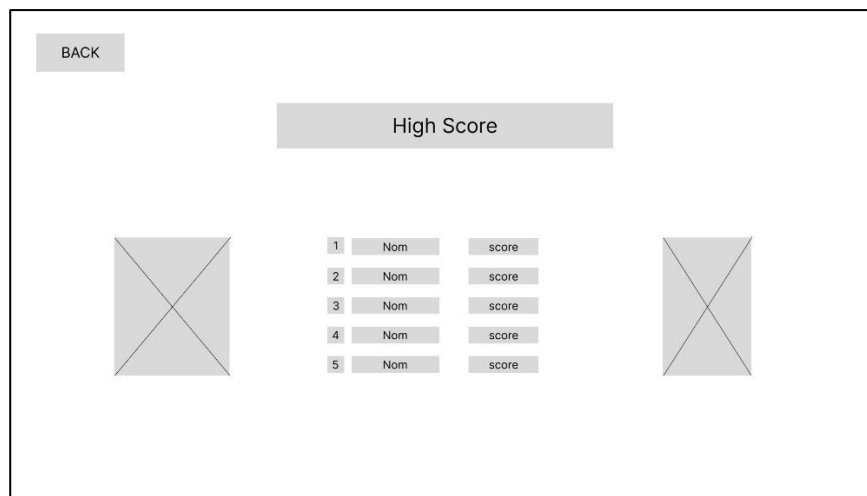
Au lancement du jeu, nous retrouvons le menu permettant d'accéder à d'autres page ou de commencer à jouer.



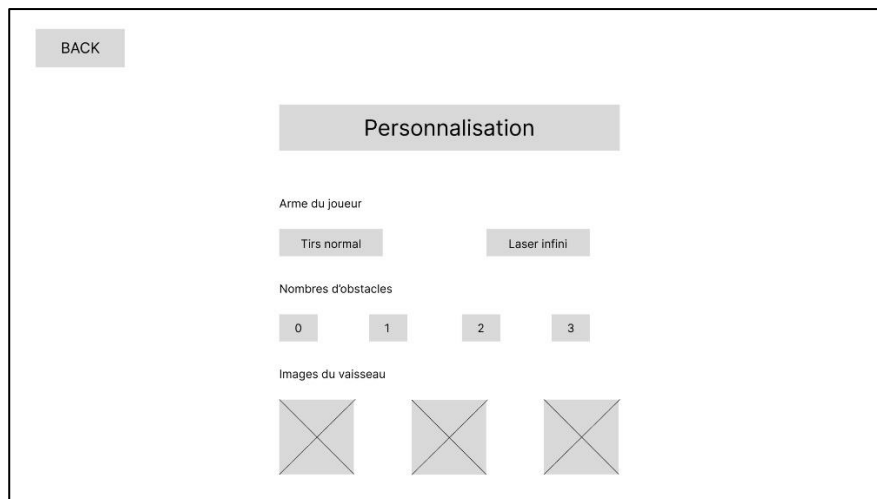
En sélectionnant le bouton PLAY dans le menu, le joueur accède à Space Invaders. L'objectif est d'éliminer tous les ennemis sans subir de dommages, en utilisant les obstacles pour se protéger.



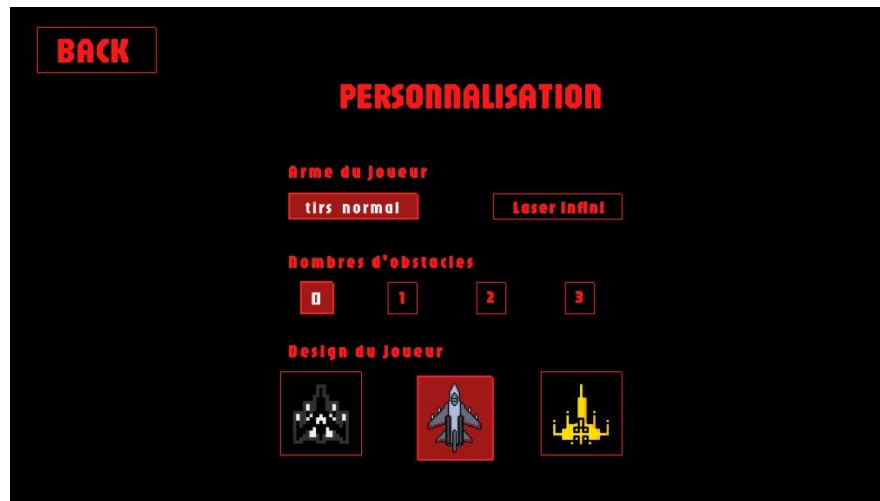
En appuyant sur le bouton **LEVEL** depuis le menu, l'utilisateur a 2 choix de niveau (niveau 1, niveau 2) ou il peut aussi personnaliser le niveau.

*High Score*

En sélectionnant le bouton high score dans le menu, il est possible de consulter les scores des 5 meilleurs joueurs du jeu.

*Editeur de jeu*

En choisissant de personnaliser, l'utilisateur peut sélectionner son arme (missile ou laser), le nombre d'obstacles pour se protéger pendant le jeu, ainsi que l'option de modifier l'image de son vaisseau.



Editeur de jeu

2.2.5 Choix effectués :

Les couleurs sont sélectionnées en fonction du jeu Space Invaders. Les options de sélection sont affichées au centre, en grand format, afin d'être facilement accessibles, avec une police Jaro qui évoque le style de texte des jeux vidéo.

2.3 Concept

- *Diagramme de classe*
- *Diagramme(s) d'état*

2.4 Analyse fonctionnelle

2.4.1 Création du Menu principal

En tant qu'utilisateur, je souhaiterais avoir un menu, afin de pouvoir accéder au jeu	
Tests d'acceptance:	
Naviguer entre les boutons verticalement	Dans le menu Quand je presse flèche du haut ou flèche du bas je peux choisir de jouer, changer de niveau ou de quitter le programme
En appuyant enter	Dans le menu Quand j'appuie sur la touche ENTER je valide la proposition dont j'ai choisi et cela me lance le jeu, choix du niveau ou l'arrêt du programme

2.4.2 Déplacement gauche ou droite

En tant qu'utilisateur, je souhaiterais pouvoir bouger mon vaisseau de gauche à droite.

Tests d'acceptance:	
déplacement à l'horizontal	En cours de partie Quand je presse flèche gauche ou flèche droite, le vaisseau se déplace dans la direction de la flèche
ne sort pas du cadre du jeu	En cours de partie quand je déplace le vaisseau jusqu'au bord, il est bloqué
déplacement à l'horizontal	En cours de partie Quand je presse A ou D, le vaisseau se déplace à gauche ou à droite
Maintenir la touche pour bouger à gauche ou droite	En cours de partie Quand je maintiens la touche pour bouger à gauche ou à droite le vaisseau avance en continu dans la direction choisi, une fois la touche relâchée le vaisseau s'arrête.

2.4.3 Capacité de tir

En tant qu'utilisateur, je souhaiterais savoir les capacité de tirs(direction des tirs, munitions) du vaisseau afin de pouvoir l'utiliser contre les ennemies.	
Tests d'acceptance:	
Tirer est vérifier le cooldown	En cours de partie quand un missile à été tiré Je peux pas en tirer un deuxième avant qu'il à été disparu

2.4.4 Nombre de vies

En tant qu'utilisateur je voudrais savoir le nombre de vie du vaisseau, afin de savoir quand sera la fin de la partie	
Tests d'acceptance:	
Mourir 1 fois	En cours de partie Quand je me fais toucher par un ennemi le vaisseau est détruit et la partie est terminée.

2.4.5 Tir

En tant qu'utilisateur, je souhaiterais savoir la direction des tirs, afin de pouvoir tirer sur les ennemis.	
Tests d'acceptance:	
Le vaisseau tirs	En cours de partie Quand je presse sur espace, le vaisseau tire verticalement
déplacement du missile	En cours de partie une fois que j'ai tiré, le missile monte tous seul
disparition du missile	En cours de partie quand le missile arrive en haut de l'écran le missile disparaît.

2.4.6 déplacement des ennemis

En tant qu'utilisateur, je souhaiterais savoir par où viennent les ennemis, afin de pouvoir se protéger.	
Tests d'acceptance:	
Déplacement vertical	En cours de partie Quand je joue les ennemis apparaissent depuis le haut de l'écran et va jusqu'en bas

Les ennemis ne s'entremêlent pas	En cours de partie Quand les ennemis apparaissent Les ennemis ne se touchent pas entre eux (il n'y a pas de collision)
----------------------------------	--

2.4.7 Nombre de vies des ennemies

En tant qu'utilisateur, je souhaiterais savoir le nombre de point de vie de chaque ennemie, afin de pouvoir les contrôler.	
Tests d'acceptance:	
Tirs sur les ennemis commun	En cours de partie Quand je touche un ennemi avec mon projectile, Il disparaît de l'écran.
tirer sur les ennemis rare	En cours de partie Quand je touche 3 fois un ennemi de type rare, il disparaît de l'écran

2.4.8 Lancement du fichier exécutable

En tant qu'utilisateur, je souhaiterais lancer un fichier exécutable, afin de pouvoir accéder au menu du jeu.	
Tests d'acceptance:	
Lancer le fichier exécutable	Lancer le fichier et voir un menu qui s'affiche

2.4.9 Déplacement de l'obstacle

En tant qu'utilisateur je souhaiterais savoir le déplacement de l'obstacle, afin de pouvoir me protéger.	
Tests d'acceptance:	
Déplacement à l'horizontal	En cours de partie automatiquement, l'obstacle se déplace horizontalement (droite, gauche)
ne sort pas du cadre du jeu	En cours de partie Quand l'obstacle s'approche du bord de l'écran, il repart dans le sens opposé

2.4.10 Durée de vie de l'obstacle

(Auteur: Abiram Muthulingam)

En tant qu'utilisateur je souhaiterais savoir à quel moment l'obstacle disparaît, afin de pouvoir l'utiliser correctement.	
Tests d'acceptance:	
Résistance de l'obstacle	En cours de partie, Quand l'obstacle se fait toucher Il est toujours en fonctionnement

2.4.11 Ajouts d'un 2ème obstacle

(Auteur: Abiram Muthulingam)

En tant qu'utilisateur, je souhaiterais avoir plusieurs obstacles afin de pouvoir me protéger	
Tests d'acceptance:	
Pas de collision avec l'obstacle 1	En cours de partie, Quand les 2 obstacles se rapprochent, il repart dans le sens opposé

2.4.12 Tirs ennemis

(Auteur: Abiram Muthulingam)

En tant qu'utilisateur, je souhaiterais avoir des ennemis qui puisse tirer, afin de pouvoir jouer	
Tests d'acceptance:	
Les ennemis tirs	En cours de partie Les ennemis tirs horizontalement pour toucher le joueur
Collision avec le vaisseau ou l'obsacle	En cours de partie Quand les ennemis tirs sur le vaisseau ou l'obstacle le tirs doit disparaître

2.4.13 tirs de l'obstacle

(Auteur: Abiram Muthulingam)

En tant qu'utilisateur, je souhaiterais savoir comment l'obstacle m'aide, afin de gagner la partie	
Tests d'acceptance:	
l'obstacle tirs	En cours de partie, Quand le joueur ne tirs pas l'obstacle se transforme pour pouvoir tirer automatiquement
l'obstacle ne tirs pas	En cours de partie, Quand le joueur tirs, l'obstacle se transforme en murs qui se déplace

Reprendre le contenu des User Stories d'IceScrum : Story + tests d'acceptance (avec IceTools) + maquettes

2.5 Stratégie de test

Création des tests expliqué clairement, en trouvant des problèmes de fonctionnalités en se mettant à la place d'un utilisateur. Effectuer ces tests une fois qu'une fonctionnalité est terminée, et vérifier qu'aucun problème persiste.

Décrire quels sont les MOYENS utilisés pour faire les tests, ne pas décrire les tests à effectuer !!!

Décrire l'environnement dans lequel se fait la sprint review

Décrire la stratégie globale de test :

- types de des tests et ordre dans lequel ils seront effectués.
- les moyens à mettre en œuvre.
- couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).
- données de test à prévoir (données réelles ?) **et comment elles seront mises en place.**
- les testeurs extérieurs éventuels.

3 Réalisation

3.1 Points de design spécifiques

Ce chapitre est constitué de plusieurs sous-chapitre.

Chaque sous-chapitre explique un point de design technique particulier, quelque chose que vous avez dû inventer pour répondre au besoin et qui ne peut pas s'expliquer par de simples commentaires dans le code.

Il s'agit d'explications techniques sur le fonctionnement du système. Les explications sont appuyées par des diagrammes, ou de très brefs éléments de code.

NE PAS mettre ici des pratiques usuelles que tout professionnel de la branche connaît déjà. Par exemple, n'EXPLIQUEZ PAS ICI CE QU'EST LE PATTERN MVC.

Exemple (simplifié à l'extrême) : Protection contre des formulaires mal intentionnés ou modifiés

- ***Au moment de générer le formulaire, le script php :***
 - ***Concatène les noms de tous les champs contenus dans le formulaire***
 - ***Calcule un hash SHA256 de la chaîne obtenue***
 - ***Ajoute un input nommé « CSRF » de type hidden dans le form***
- ***A la réception du POST du formulaire***
 - ***Concatène les noms des indices de \$_POST***
 - ***Calcule un hash SHA256 de la chaîne obtenue***
 - ***Vérifie que la valeur du champ CSRF correspond***

3.1.1 ...

3.1.2 ...

3.1.3 ...

3.2 Déroulement

Résumer comment s'est passé la réalisation de chaque story, ses difficultés, les alternatives envisagées mais rejetées, ses surprises, ...

3.3 Mise en place de l'environnement de travail

- ***Comment accéder au code source***
- ***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***

Ce chapitre décrit précisément comment un employé qualifié peut recréer l'environnement dans lequel vous avez effectué ce travail

3.4 Description des tests effectués

Reprendre les tests d'acceptance d'IceScrum au moyen de la feuille ad hoc d'IceTools

3.5 Erreurs restantes

S'il reste encore des erreurs :

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

Reporter la dette technique connue. S'appuyer sur la pratique des // TODO

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 Manuel de référence

Issu de la génération automatique à partir des commentaires

5.2 Journal de travail