



History Treasures

« Faites un saut dans le temps avec History Treasures ! »

« Mon premier est une lettre

Mon second est ordonné

Mon troisième est la réponse

Mon tout est l’Histoire

Qui suis-je ? »

Spécifications

Version

31/03/2015

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 12/04/2016 | 1.0 | Début du document | Julie |
| 12/04/2016 | 1.1 | Ajout cas d’utilisation | Yacine |
|  |  |  |  |
|  |  |  |  |

Sommaire

1 Introduction 4

1.1 Mission 4

1.2 Objectifs 4

1.3 Glossaire 4

2 Description générale 5

2.1 Acteurs 5

2.2 Cas d’utilisations 5

3 Spécifications fonctionnelles 6

3.1 Carte de navigation 6

3.2 Détails des cas d’utilisations 6

4 Spécifications non fonctionnelles 9

4.1 Contraintes de conception et d’implémentation 9

4.2 Documentation utilisateur 9

Les spécifications sont à la base de l’accord entre le client et l’équipe réalisatrice sur ce que le système devra faire et ce qu’il ne devra pas faire.

Ce document est adapté à une description des fonctionnalités sous forme de cas d’utilisation.

Critères d’évaluations lors de l’avant-projet ou avant chaque début d’itération (\*) :

Spécifications fonctionnelles :

* Les acteurs (visiteur, administrateur, …) sont identifiés
* Les cas d’utilisations sont listés
* Les cas d’utilisations ont été détaillés + maquette (\*\*)
* La carte de navigation du logiciel est présente (\*)

Spécifications non-fonctionnelles

* Les contraintes de conception et d’implémentation ont été listées

(\*) Dans le cas d’un processus itératif, les spécifications détaillées (paragraphe 3) peuvent concerner l’itération courante ou plusieurs itérations

(\*\*) Dans le cas d’un processus itératif, les spécifications détaillées sont validés par le client à chaque début d’itération

# Introduction

## Mission

Proposer et développer un jeu en 2D isométrique à caractère pédagogique sur le thème de l’histoire.

## Objectifs

1 – Mettre au point une interface claire et attrayante.

2 – Définir trois thèmes contenant quatre sous-parties.

3 - Recherche historique afin de l’intégrer au sein du jeu.

4 – Mettre en place le gameplay.

5 – Mettre le jeu à disposition de la communauté en tant que jeu indépendant.

## Glossaire

Définir les termes propres au domaine de l'application, les sigles et les abréviations, les termes techniques nécessaires à une bonne compréhension et interprétation du document

### Termes du domaine

Nous n’avons aucuns termes concernant le domaine.

### Termes techniques

Gameplay : Ensemble des mécanismes de jeu.

Gamedesign : Ensemble des graphismes de jeu.

Map : Carte du jeu

# Description générale

## Acteurs

Présenter les différentes familles d’utilisateurs de la solution

|  |  |
| --- | --- |
| **Nom** | **Description** |
| Le joueur | Il s’agit de l’utilisateur final de notre projet, le joueur représente toute personne utilisant le jeu. |

## Cas d’utilisations

Lister les cas d’utilisations principales du produit, les détails seront donnés plus bas

### Diagramme de cas d’utilisation

Insérer ici le diagramme de cas d’utilisation représentant les cas d’utilisation (détaillés dans le 3) associé aux acteurs. Structurer éventuellement le diagramme en utilisant des relations d’inclusion, d’extension ou de généralisation.

### Listes des cas d’utilisations

Nommer et décrire brièvement les cas d’utilisation répondant aux exigences fonctionnelles de votre système.

Pour identifier les cas d'utilisation appropriés, commencez par réfléchir à ce que chaque acteur attend du système. Pour chaque acteur, humain ou non, posez-vous les questions suivantes :

* Du point de vue de l'acteur, quelles sont les principales tâches que le système doit exécuter ?
* L'acteur va-t-il créer, stocker, modifier, supprimer ou lire des données dans le système ?
* L'acteur sera-t-il amené à informer le système de changements extérieurs soudains ?
* L'acteur doit-il être avisé de l'occurrence de certains événements dans le système ?
* Une fois les cas d’utilisation identifiés, hiérarchisez-les en tenant compte des deux facteurs suivants :
* La priorité fonctionnelle
* Le risque technique
* L’objectif ensuite est de planifier en premier (premières itérations) la réalisation des cas d’utilisation ayant un risque élevé et une priorité haute.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom du cas d’utilisation** | **Acteur(s)** | **Priorité** | **Risques** |
|  |  |  |  |

# Spécifications fonctionnelles

## Carte de navigation

L'objectif de la carte de navigation est d'exprimer les chemins d'interface principaux dans le système. Ces chemins sont les chemins principaux à l'écran et ne reprennent pas la totalité des possibilités. Elle peut être considérée comme une carte routière de l'interface utilisateur.

Un diagramme de navigation peut être représenté grâce à un diagramme d’activité

## Détails des cas d’utilisations

Certains cas d’utilisation complexe avec de nombreuses extensions et/ou répétitions peuvent être illustrés par un diagramme d’activité.

### Nom du cas

Le nom du cas doit commencer par un verbe. Il correspond à l’objectif à atteindre.

#### Description :

Résumé succinct qui présente le scénario de base (succès) dans un paragraphe.

Peut être rédigé lors de la première étude de besoins et avant la rédaction détaillée du cas d’utilisation pour se faire une idée du sujet et de son périmètre.

#### Acteur principal :

Celui qui fait appel au système pour atteindre un but.

Un acteur n’est pas un utilisateur : une même personne peut jouer plusieurs rôles, plusieurs personnes peuvent jouer un même rôle.

Un acteur peut être un système, une machine.

#### Acteurs secondaires :

Optionnel.

Autres participants du cas d’utilisation.

#### Préconditions :

Ce qui doit toujours être vrai avant le début d’un scénario. En principe, une précondition implique que le scénario d’un autre cas d’utilisation s’est déroulé normalement (ex : « Le Caissier est identifié et authentifié »).

#### Postconditions (garantie en cas de succès) :

Ce qui doit être vrai lorsque le cas d’utilisation se termine avec succès, qu’il s’agisse du scénario principal ou d’un scénario alternatif.

#### Scénario nominal :

Il décrit le scénario type qui satisfait les intérêts de l’acteur principal et des acteurs secondaires. Il ne comprend pas de conditions ni de branchement. Les traitements conditionnels sont à reporter dans la section Extensions.

1. Le cas débute lorsque …

La première étape indique l’événement qui déclenche le scénario.

Exemple pour le cas d’utilisation Traiter une vente : « Le Client arrive à la caisse avec les articles qu’il souhaite acheter »

1. L’Utilisateur {verbe d’action} …
2. Le Système {verbe d’action} …

Le nom des acteurs doivent commencer par une majuscule.

Chaque étape d’actions est rédigée comme une action simple « dans laquelle le sujet est actif ». On peut la comparer à la description d’un match de football : « la pêrsonne1 envoie le ballon à la personne2 ; la personne 2 dribble ; la personne2 envoie le ballon à la personne3 ».

Le scénario est composé d’étapes, lesquelles sont de trois sortes :

1. Une interaction entre deux acteurs (le système étant considéré comme un acteur). Exemple : « Le client saisit son adresse », « Le Système affiche le prix de l’article »

2. Une étape de validation pour protéger les intérêts d’un acteur. Exemple : « Le Système valide le code secret ».

3. Un changement interne pour satisfaire aux intérêts d’un intervenant. Exemple : « Le Système enregistre la vente ».

Il peut être parfois utile d’utiliser des mots clés permettant d’exprimer la répétition

Exemple :

Pour chaque article trouvé

Le système affiche une image du produit

Le système affiche le prix du produit

…

1. Pour {expression d’itération}
2. Faire quelque chose
3. Faire quelque chose d’autre
4. Tant que {expression booléenne}
5. Faire quelque chose
6. Faire quelque chose d’autre

#### Extensions:

Les extensions permettent de décrire les autres scénarios ou branchement possibles, tant en cas de succès qu’en cas d’échec.

3a . Si {*Condition d’extension de l’étape 3 du scénario nominal– à compéter}*  alors:

Comme pour la première étape du scénario nominal, la condition d’extension indique l’événement qui déclenche le scénario.

Exemple : « Si le code de l’article est invalide (non trouvé par le Système) alors »



Description des étapes permettant de prendre en charge la condition d’extension

#### Interface utilisateur:

Représenter les écrans de l’application. Il convient de rappeler que la maquette n’est pas le produit final !

Vous pouvez pour cela dessiner les écrans à la main ou utiliser un outil spécifique de création d’interface utilisateur (exemple : <http://www.mockupscreens.com/>, http://c2.com/cgi/wiki?GuiPrototypingTools)

#### Fréquence :

Optionnel.

Fréquence moyenne estimé d’apparition de ce cas d’utilisation par jours

#### Spécifications particulières :

Optionnel.

Besoin non fonctionnel, attribut de qualité ou contrainte se rapportant spécifiquement à ce cas d’utilisation. Il peut s’agir de questions de performance, de fiabilité ou d’ergonomie. Exemple : « Interface utilisateur à écran tactile sur grand écran plat. Le texte doit être visible à un mètre. ».

#### Questions en suspens:

Optionnel.

# Spécifications non fonctionnelles

## Contraintes de conception et d’implémentation

Décrire tous les facteurs qui vont restreindre la liberté des concepteurs, donner la raison de chaque contrainte

Types de contraintes :

* Technologies, outils, langages de programmation, bases de données, spécifiques imposées ou interdites
* Restriction sur la version du système d’exploitation ou du navigateur Web
* Règles ou conventions imposées dans l’entreprise : structure de la documentation technique destinée à une future maintenance par exemple
* Compatibilité avec des produits plus anciens
* Limitations imposées par les règles du métier
* Limitations liées au matériel disponible : temps de réponse, taille mémoire, vitesse du processeur, taille, poids, matériaux, coûts
* Conventions liées à l’interface utilisateur, lors d’une extension par exemple
* Format standards d’échange d’information : XML par exemple

Une de nos plus grosses contraintes est le temps, même si à l’heure actuelle nous ne pouvons pas nous avancer mais le temps est dans beaucoup de projet une contrainte très embêtante et très restrictive.

En terme de contrainte technique, nous allons concevoir notre jeu à l’aide de Windows Forms ainsi que du C#. L’utilisation de GitHub est également imposée.

En ce qui concerne les règles ou convention imposées dans l’entreprise nous nous imposerons une convention de codage qui devra être respecté par l’équipe. Les enseignants suiveurs vérifieront que ce détail a bien été respecté.

Il n’y aura aucune règle de compatibilité avec des produits plus anciens.

Nous n’avons aucune compétence nécessaire pour pouvoir indiquer clairement une limitation lier au matériel disponible.

Il faudra respecter l’interface utilisateur existante en cas de création d’extension.

Nous n’avons aucun format standard d’échange d’information

## Documentation utilisateur

Lister les composants de la documentation à destination des utilisateurs qui sera livrée avec le produit

Ceci comprend : les manuels utilisateurs, l’aide en ligne et les tutoriaux

Identifier pour chaque document le format, les standards à respecter, les outils à utiliser