

CAHIER DES CHARGES

Projet Worms

Version: 2.2

Auteur: ORNON Thomas, PITUELLO Cédric

ISEN Toulon - Yncréa
Maison du Numérique et de l'Innovation
Place Georges Pompidou
83000 Toulon

Description du document

Type		Version	Confidentialité	
Cahier des charges		2.2	Usage externe	
	Nom	Fonction	Date	Visa
Rédacteur	ORNON Thomas PITUELLO Cédric			
Vérificateur				
Approbateur				
Destinataire		Fonction	Organisme	
Public			ISEN	

Révisions du document

[illegible]

Sommaire

1. Présentation du projet.....	7
A) Contexte du projet.....	7
B) Objectifs du projet	7
C) Description de l'existant	7
D) Acceptation du produit	8
2. Expression fonctionnelle des besoins.....	8
A) Besoins fonctionnels	9
B) Besoins non fonctionnels	9
3. Contraintes	11
A) Budgétaires.....	11
B) Temporelles.....	11
C) Normatives.....	11
4. Déroulement du projet	11
A) Planification.....	11
B) Documentation	11
C) Equipe projet et responsabilités	11

Index des illustrations

Illustration 1: Screenshot d'une partie du jeu Worms 6

Index des tables

REFERENCES

Référence	Description	Nom
[1]		
[2]		

DEFINITIONS

Sans objet

ABBREVIATIONS

ISEN : Institut Supérieur de l'Electronique et du Numérique

1. Présentation du projet

A) Contexte du projet

« Joué au tour par tour, en 2 dimensions, il oppose deux équipes de vers de terre armés jusqu'aux dents, selon le principe du jeu d'artillerie. Le concept de jeu a été inventé par Andy Davison.

Le but étant de détruire tous ses ennemis, on y trouve des armes destructrices à souhait, plus ou moins conventionnelles, allant du bazooka, jusqu'au fusil en passant par le coup de poing de feu ou le mouton explosif.

Ces armes enlèvent de la vie au(x) ver(s) touché(s) ; réduire les points de vie des vers ennemis étant le principal moyen de gagner.

Cependant, on peut aussi envoyer les lombrics dans l'eau où ils se noient aussitôt. Certaines armes sont d'ailleurs prévues pour ça, comme la très utilisée batte de baseball. Sortir un ver des limites du jeu permet également de l'éliminer. Un inventaire d'outils permet également de creuser le décor, de s'agripper au mur ou encore de poser des mines. »

Source : Wikipédia



Illustration 1: Screenshot d'une partie du jeu Worms Armageddon

B) Objectifs du projet

L'objectif du projet est donc de développer un clone de Worms. Les aspects importants à gérer seront la physique des vers avec le décor, la gestion et l'animation des tirs, la map destructible et l'interface, entre autres. Le langage de programmation est C#. Nous utiliserons le framework Monogame, avec peut-être Velcro Physics (anciennement Farseer Physics) pour le moteur physique.

C) Description de l'existant

Le jeu Worms est sorti en 1995, édité par Ocean Software, sur Amiga et PC. A ce jour ils existent une multitude de générations de jeux Worms.

D) Acceptation du produit

Le logiciel sera considéré comme acceptable s'il vérifie tous les tests spécifiés dans le cahier de recette.

2. Expression fonctionnelle des besoins

Les exigences décrites ci-après ont pour but de décrire les caractéristiques du projet.

Elles sont verbalisées en trois parties : objet-verbe-attribut, où :

- objet désigne le sujet sur lequel porte l'exigence ;
- attribut désigne la caractéristique que le verbe impose à l'objet.

Numérotation

Les exigences sont numérotées sur quatre chiffres, de dix en dix pour pouvoir éventuellement insérer, après accord de toutes les parties concernées, de nouvelles exigences à côté de précédentes exigences.

Un numéro d'exigence abandonné ne pourra pas être réutilisé.

Les exigences sont de plusieurs types, chaque type décrit ci-après étant représenté par une lettre caractéristique. Cette lettre prefixera chaque exigence.

Les exigences peuvent être facultatives ou indispensables. Les lettres F et I suffixeront respectivement les numéros d'exigences décrites ci-dessus.

Types d'exigences

Les types d'exigences sont les suivants (avec entre parenthèses la lettre servant de préfixe au numéro d'exigence) :

- **exigences Fonctionnelles (EF)** : portent sur ce que le produit doit être capable de faire ;
- **exigences de Design (ED)** : portent sur tout ce qui a trait à la forme au travers de laquelle on interagira avec le produit ;
- **exigences d'Implémentation (EI)** : portent sur les contraintes techniques liées à la réalisation du produit ;
- **exigences de Performance (EP)** : portent sur des niveaux quantitatifs qui doivent être atteints dans des conditions à préciser ;
- **exigences de Maintenance (EM)** : portent sur tout le support qui est à mettre en œuvre pour assurer le bon fonctionnement du produit ;
- **exigences de Validation (EV)** : portent sur les actions qui permettent de valider des exigences ;
- **exigences Sécuritaires (ES)** : portent sur les mesures à prendre pour assurer la sécurité et la sûreté des informations ;
- **exigences Légales (EL)** : portent sur les mesures à prendre pour assurer la légalité des actions du produit.

A) **Besoins fonctionnels**

Les besoins fonctionnels décrivent les différentes fonctions que le logiciel devra assurer. Il est préférable de les classer par catégories, afin d'en simplifier la lecture.

Exigences fonctionnelles

EF_0100_I

Gérer les Actions

EF_0110_I

Gérer les Déplacements

EF_0111_I

Aller à gauche

EF_0112_I

Aller à droite

EF_0113_I

Sauter

EF_0114_I

Attendre

EF_0120_I

Gérer les Tirs

EF_0121_I

Viser

EF_0122_I

Tirer

EF_0200_I

Gérer la Physique

EF_0210_I

Gérer les collisions

EF_0220_I

Gérer la gravité

EF_0230_I

Gérer la destruction de la carte

Exigences de design

ED_0100_I

Gérer les Animations

ED_0110_I

Gérer les Animations du Worms

ED_0111_I

Aller à gauche

ED_0112_I

Aller à droite

ED_0113_F

Attendre

ED_0114_F

Sauter

ED_0115_I

Tirer

ED_0116_F

Mourir

ED_0120_I

Gérer les Animations du projectile

ED_0121_I
Gérer le déplacement du projectile
ED_0122_F
Gérer l'explosion du projectile

ED_0200_I
Gérer la Carte
ED_0210_I
Dessiner la carte
ED_0220_I
Mettre à jour la carte

ED_0300_I
Gérer les Interfaces
ED_0310_F
Afficher le menu 1
ED_0320_F
Afficher le menu 2
ED_0330_F
Inventaire
ED_0331_F
Afficher l'inventaire
ED_0332_F
Se déplacer dans l'inventaire
ED_0333_F
Choisir son arme

ED_0400_F
Gérer l'Audio
ED_0410_F
Gérer le bruit des armes
ED_0420_F
Gérer la musique du jeu

B) Besoins non fonctionnels

Les besoins non fonctionnels sont tous ceux qui n'ont pas trait aux fonctions du logiciel, comme par exemple le système d'exploitation sur lequel il devra fonctionner.

Exigences d'implémentation

EI_0010_I
Le logiciel devra fonctionner sous système d'exploitation Windows.
EI_0020_I
Le logiciel devra être écrit en langage C#.
EI_0030_I
Le logiciel devra utiliser le FRAMEWORK MONOGAME.
EI_0040_F
Le logiciel devra utiliser le FRAMEWORK FARSEER.

Exigences de performance

EP_0010_I
Le logiciel devra consommer moins de 20% du processeur.

EP_0020_I

La phase de réflexion de l'ordinateur ne devra pas excéder 20 secondes.

Exigences de maintenance

EM_0010_I

Le logiciel sera maintenu par l'équipe projet pendant 1 an.

EM_0020_F

Les sources du logiciel seront commentées au format Doxygen.

Exigences de validation

EV_0010_I

L'interface utilisateur sera validée manuellement.

EV_0020_I

Le jeu sera jouable à plusieurs sur le même ordinateur.

Exigences sécuritaires

Aucune exigence sécuritaire.

Exigences légales

Aucune exigence légale.

3. Contraintes

A) *Budgétaires*

Le budget alloué pour ce projet sera de 0€. Les moyens matériels mis à disposition sont trois ordinateurs reliés à internet.

B) *Temporelles*

La livraison du produit fini est prévue en mai 2018. La documentation relative à l'analyse du logiciel devra être livrée fin décembre 2017.

C) *Normatives*

Le code source du projet devra être documenter pour faciliter la compréhension et lecture du code.

4. Déroulement du projet

A) *Planification*

Le projet se déroulera sur 9 mois.

La première semaine sera consacrée à l'analyse de l'existant et à la rédaction du CDC et du WBS dans un premier temps.

B) *Documentation*

Le projet livré sera accompagné d'un cahier de conception général et détaillé.

C) *Equipe projet et responsabilités*

L'équipe sera constituée d'un chef de projet, et de 1 développeur.