



# Cahier des charges

**Développement d'un logiciel de jeu vidéo :  
« A Bug's Life »**



Table des matières :.....	<b>Erreur ! Signet non défini.</b>
Introduction :.....	4
Introduction :.....	4
Objet : .....	4
Contexte du projet : .....	4
Description détaillée du produit :.....	5
Analyse fonctionnelle : .....	5
Etape1 : .....	5
Interface : .....	7
Conditionnement et livraison (Livvable) : .....	8
Conditions de réalisation : .....	8
Délai de réalisation : .....	8
Contraintes de développement : .....	8
Communication avec le client :.....	9
Suivi de réalisation : Diagramme de Gantt :.....	9
ANNEXE :.....	11
Règles du jeu.....	11

## Introduction :

La société VGD (Video Game Development) est spécialisée depuis 18 ans dans le développement de jeux vidéo pour plateformes PC, Playstation 2, et XBox 360. VGD vous assure des prestations de qualité dans les domaines du développement, du Game Design, et du consulting. Notre très bonne connaissance des processus de production nous octroie un haut niveau de prestation et une grande maîtrise des paramètres Coût-Qualité-Délais, ceci aussi bien pour des développements mono-plateforme que multi-plateformes. Nous intervenons sur l'ensemble du cycle de développement, mais nous pouvons également vous aider à approfondir et finaliser vos idées de jeux. VGD travaille régulièrement sur de nouveaux concepts originaux pour les différentes plateformes actuelles, adaptés à différents types de budgets. VGD est actuellement la 3<sup>ème</sup> entreprise de création de logiciel de loisirs d'après le classement de Juillet 2006 ci-contre ( Source : ieMagazine, the newspaper of the interactive entertainment industry <http://www.iemag.com/>).

Crdre	Editeur
1	NINTENDO OF AMERICA
2	VIDEO GAME DEVELOPMENT
3	SONY
4	ACTIVISION
5	VIVENDI UNIVERSAL
6	INFOGRAMES
7	THQ
8	SEGA OF AMERICA
9	UBISOFT
10	TAKE 2 INTERACTIVE
11	MICROSOFT
12	ACCLAIM ENTERTAINMENT
13	KONAMI OF AMERICA
14	NAMCO
15	MIDWAY
16	EIDOS INTERACTIVE
17	THE 3DO COMPANY
18	SQUARE EA
19	CAPCOM USA
20	INTERPLAY PRODUCTIONS

## Objet :

Ce cahier des charges a pour but de définir et présenter un projet de développement informatique. Le but est la réalisation d'un programme de jeu. Ce document fixera par conséquent le contexte du projet, les impératifs de cette mission ainsi que son organisation avec tout ce que cela implique.

## Contexte du projet :

La société VGD propose ainsi la réalisation d'un logiciel de loisir type jeu vidéo exécutable sur une plateforme PC. Ce logiciel ludique doit permettre aux utilisateurs (joueurs) de développer un certain esprit de jeu. Le jeu se présentera sous la forme d'un plateau de jeu en 2 Dimensions sur lequel évolueront des personnages. L'objectif du joueur

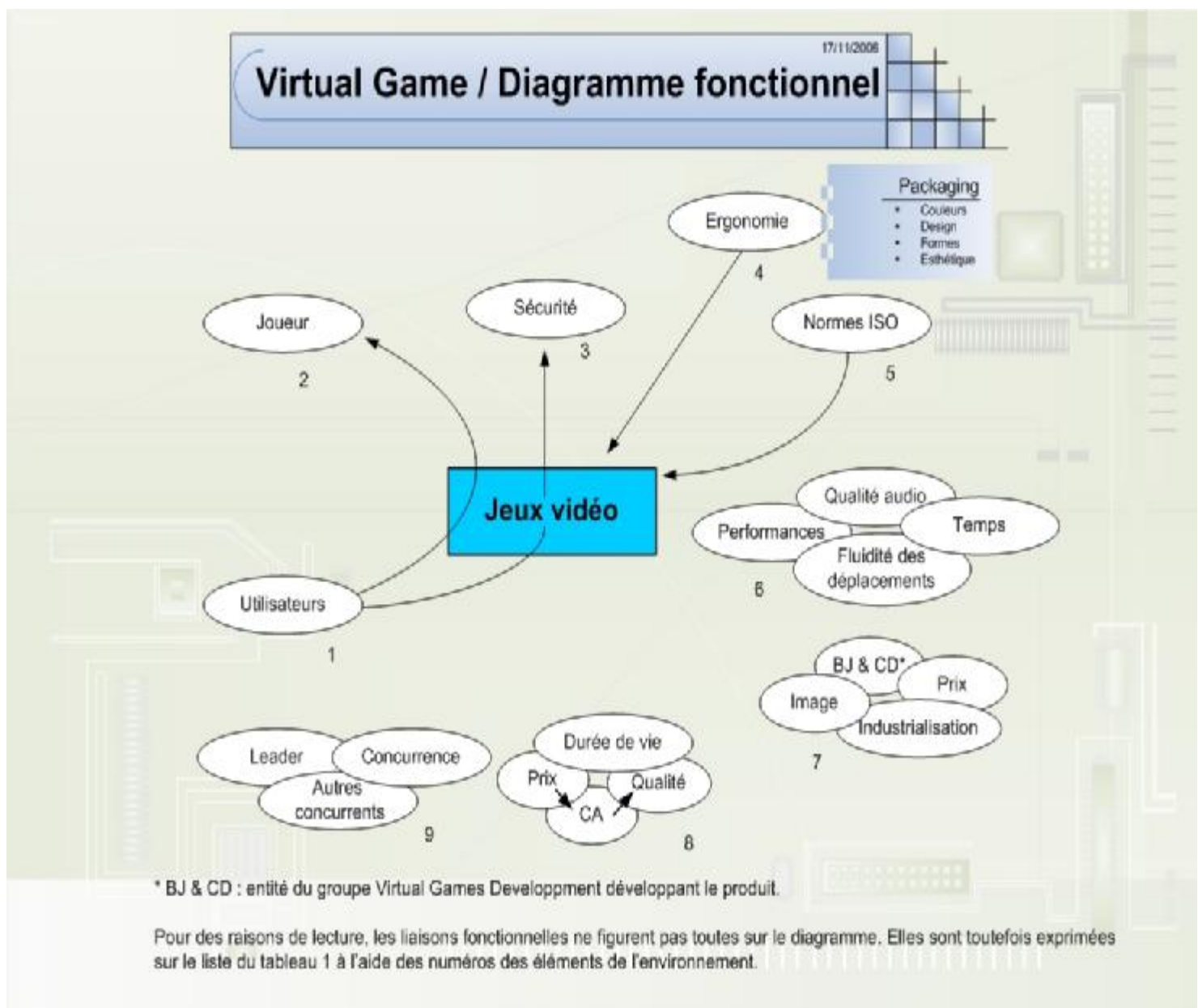
sera de prendre successivement le contrôle de l'un d'entre eux dans l'optique de le nourrir et de le faire évoluer. Tous ces différents paramètres relatifs à la jouabilité seront présentés en annexe dans une plus large mesure (Règles du jeu). Le jeu s'intitulera « A bug's life ».

## Description détaillée du produit :

### Analyse fonctionnelle :

#### Etape1 :

On propose d'illustrer l'aspect fonctionnel du système par un diagramme.



17/11/2006

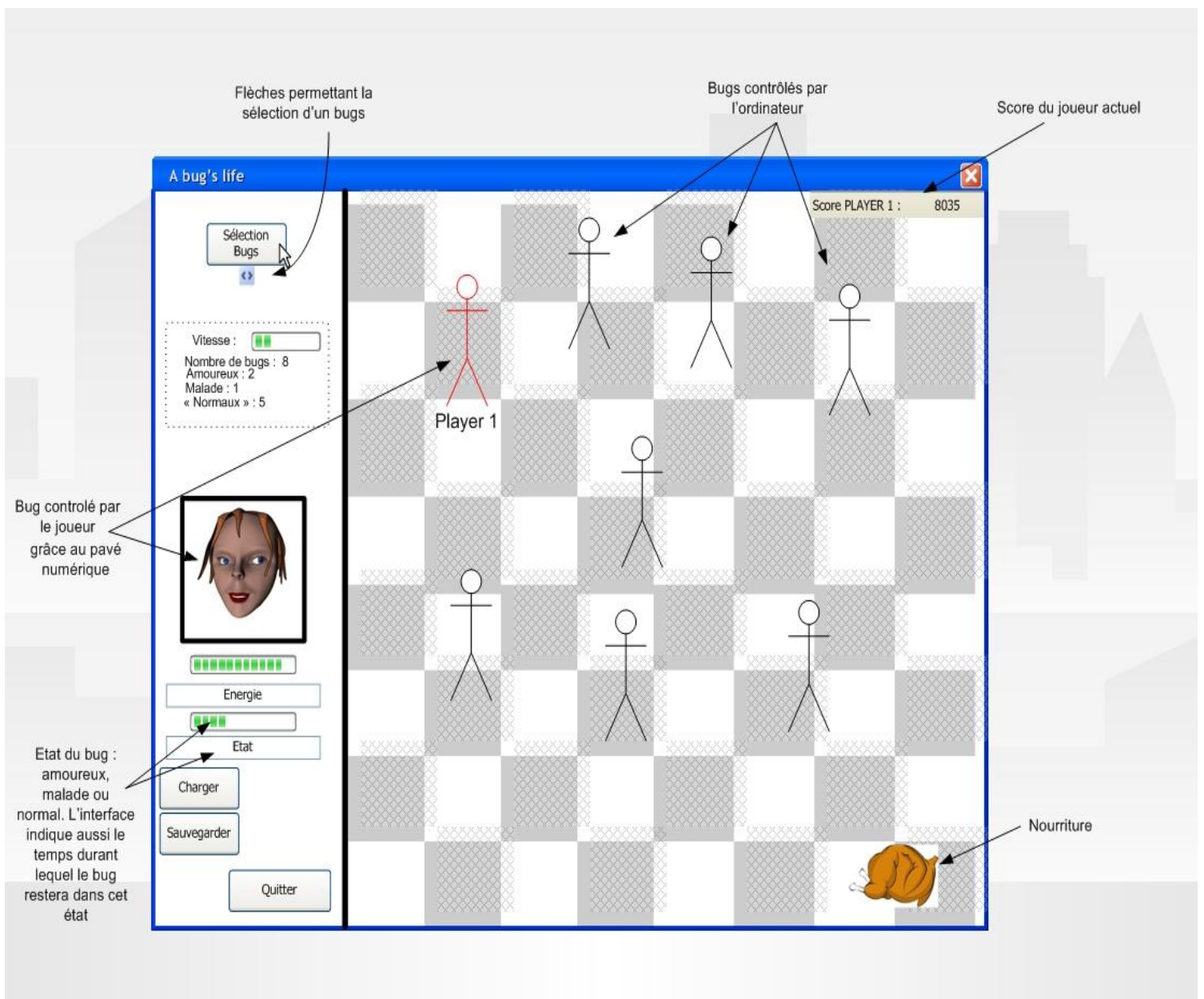
Virtual Games / Tableau 1 - Liste des fonctionnalités issues du diagramme fonctionnel

- F.1.2.** Offrir une simplicité d'utilisation (utilisation)  
Offrir une rapidité de mise en œuvre (rapidité)  
Offrir une standardisation d'utilisation (universalité des plateformes de jeux type PC)
- F.1.3.** Assurer la protection contre le piratage (Clef d'utilisation – Cryptage DES )  
Assurer l'identification grâce au chiffrement DES (ident.)
- F.1.2.** Assurer des temps de transactions faibles – Sauvegarde / Chargement de parties -.
- C.5.** Respecter les normes ISO 9000 (normes)
- F.1.6** Assurer des temps d'exécution faibles (lancement du jeu) (performance)  
Assurer des temps d'interactions joueur / machine faibles (fluidité)  
Assurer une qualité audio (son)  
Assurer une prise en main du jeu aisée (contrôle)
- F.1.7** Offrir une conduite d'utilisation (guide)  
Montrer et maintenir une bonne image
- F.1.8** Assurer une grande fiabilité (fiabilité)  
Assurer une robustesse d'usage (robustesse)  
Remplir des conditions de qualité (qualité)
- F.1.9** Offrir un produit supérieur à celui de la concurrence (concurrence)  
Analyser la concurrence – cibler le meilleur produit actuel
- F.2.7** Réussir un prototypage à la date demandée (date échantillon)  
Réussir l'industrialisation à la date du marché (date d'industrialisation)  
Offrir le prix attendu (prix)
- F.6.8** Garantir une durée de vie objective
- F.7.9** Assurer le leadership des jeux vidéo sur PC
- Total des fonctionnalités : 11**

## Interface :

Considérée comme l'intermédiaire entre l'utilisateur et le système, la définition de l'interface est cruciale. Il s'agit de ce que le client verra du système dans un contexte d'utilisation.

Une interface utilisateur résultant de l'analyse précédente qui conviendrait au projet en cours pourrait se définir de la manière suivante :





## **Conditionnement et livraison (Livrable) :**

L'intégrité du produit doit être assurée.

Le produit devra comporter :

- Le nom des fabricants.
- L'identification du produit (sa référence).

### **Précision sur le chiffrement nécessaire au dispositif anti-piratage**

Conformément aux instructions de la Législation française sur les chiffrements symétriques avec des clefs aussi grandes que 128 bits, le produit devra prendre en compte la décision du Conseil constitutionnel n° 2004-496 DC relatif à la loi du 10 juin 2004 sur la liberté de communication en ligne.

Le fournisseur s'engage donc à fournir lors de chaque livraison un certificat de conformité pour chaque lot.

Par ailleurs, la mise en œuvre éventuelle en cours d'année par le client de techniques nouvelles agréées découlant du standard de chiffrement fera l'objet d'une adaptation en conséquence du présent cahier des charges.

## **Conditions de réalisation :**

### ***Délai de réalisation :***

Dans le contexte d'appel d'offre évoqué précédemment, les délais sont fixés par l'Ecole Supérieure des Ingénieurs de Luminy (ESIL). Le cahier des charges devra être rendu le 17 novembre 2006, le rendu du projet finalisé devra s'effectuer le 18 janvier 2007.

### ***Contraintes de développement :***

Le maître d'ouvrage se doit de respecter les règles imposées par le maître d'œuvre en plus des règles de développement spécifique à son activité de programmeur. Il s'engage donc à respecter les contraintes de programmation suivantes :

- Les algorithmes utilisés seront traduits entièrement en C++ avec l'utilisation de la bibliothèque OpenGL et OpenAL. (imposé par le client)



-Le codeur s'engage à respecter une syntaxe spécifique aux commentaires dans le but de produire une documentation finale compatible « Doxygene ».

-Le programme doit bien évidemment respecter les fonctions de contraintes évoquées précédemment (Performance, interface...).

### **Communication avec le client :**

La communication « fournisseur - client » découlera des divers rendez vous organisés par le fournisseur et le client. La communication avec le client sera ainsi incluse dans les plages horaires préalablement définies dans le cadre de la réalisation du projet.

### **Suivi de réalisation :** **Diagramme de Gantt :**

ID	Noms de tâche	Début	Terminer	Durée	nov. 2006			déc. 2006				janv. 2007				
					11	11	11	12	12	12	12	12	12	12	12	12
1	Objectif Général Détermination de l'objectif général du projet	09/11/2006	09/11/2006	1j												
2	Analyse / Impératifs du logiciel Analyse des besoins Ébauche des spécifications de logiciel préliminaires Développement du budget préliminaire Développement d'un délai de livraison Rédaction du cahier des charges	09/11/2006	16/11/2006	6j												
3	Conception Examen des spécifications du logiciel Développement des spécifications fonctionnelles Développement du prototype basé sur les spécifications fonctionnelles Obtention des approbations de poursuite du projet Conception achevée	16/11/2006	07/12/2006	16j												
4	Développement Examen des spécifications fonctionnelles Développement du code Testing des développeurs (débogage primaire) Développement achevé	07/12/2006	03/01/2007	20j												
5	Tests Développement des plans de test en tenant compte des spécifications du produit Examen du code Test des modules des composants par rapport aux spécifications du produit Identification des anomalies par rapport aux spécifications du produit Modification du code Nouveau test des codes modifiés	07/12/2006	11/01/2007	26j												
6	Documentation Développement des spécifications des manuels de l'utilisateur Développement des manuels de l'utilisateur Documentation achevée	07/12/2006	15/01/2007	28j												
7	Examen post-exécution Synthèse des leçons reçues Distribution aux membres de l'équipe Examen post-exécution achevé	18/01/2007	19/01/2007	2j												

## **COUTS**

D'après le diagramme de Gantt présenté ci-dessus, le coût de développement du logiciel « A Bug's life » est évalué à 35K€

Le présent Cahier des charges (après ratification et accord des deux parties) fait état d'un coût total (Développement + Services) d'un coût total de 60K€

## ANNEXE :

### *Règles du jeu*

#### Introduction

Cette simulation de bugs mélange deux intérêts majeurs qui sont l'adresse et la rapidité du joueur. La particularité de ce jeu est que le joueur ne contrôle pas un seul personnage (bug) mais une colonie entière de bugs.

Bien que le bug soit une créature ressemblant à une « bestiole » difforme, n'ayez aucune crainte, ces petits monstres sont de gentilles petites créatures sympathiques vivant en société.

Comme tous petits organismes de ce monde, les bugs sont vulnérables et peuvent tomber malades. Par ailleurs, ce sont des créatures riches en émotions qui peuvent tomber amoureuses.

On distinguera ainsi quatre types de bugs :

- Le bug « amoureux ».
- Le bug « malade ».
- Le bug « normal ».

#### But du jeu :

Votre mission est de nourrir à tour de rôle l'ensemble des individus de la colonie. Vous vous rendrez rapidement compte que cela n'est pas une tâche facile. Dès que le quatrième décès survient, le jeu se finit. Vous marquez des points lorsque vous arrivez à nourrir un bug.

#### Les règles:

- Un **Bug** trop affamé est un **Bug** prêt à mourir. Et c'est justement ce qui ne doit pas arriver car vous en avez la responsabilité !!!
- Les **Bugs** malades peuvent contaminer leurs camarades normaux ou amoureux lors d'un contact. Le plus grand danger avec un **Bug** malade est qu'il peut contaminer la gamelle commune et la rendre inutilisable pendant une certaine période.

Un **Bug** malade redevient normal à partir d'un certain temps. Il ne faut donc absolument pas que sa maladie dure plus longtemps que sa capacité à tenir entre deux prises de repas. Sinon c'est la mort ! Vous aurez donc certainement le dilemme suivant à résoudre : sauver le **Bug** malade en le faisant quand même manger, en risquant de voir mourir un autre **Bug** affamé ou bien ne pas prendre ce risque et le laisser à son destin inéluctable.

- Il n'y a pas de feux sans poudre, on ne peut tomber amoureux sans rencontrer l'âme sœur. Le problème est que quand ça arrive à un **Bug** cela prend des proportions démesurées. Un **Bug** amoureux perd la raison et surtout l'appétit, même si vous l'emmenez à la gamelle, celui-ci ne se nourrira pas. Sans se nourrir, un **Bug** est de toute façon voué à mourir. Il ne faut donc absolument pas que son amour dure plus longtemps que sa capacité à tenir entre deux prises de repas. Dans le cas contraire, il mourra pour la plus belle des causes, l'amour.

Un **Bug** amoureux qui rencontre un autre bug amoureux se voit réinitialiser son compte pendant lequel il ne pourra aller manger

Un **Bug** normal qui touche un **Bug** amoureux devient amoureux pendant un certain temps, le **Bug** amoureux, quant à lui ne se voit affecter d'aucun changement.

Un **Bug** amoureux n'ayant pas rencontré depuis son coup de foudre un autre **Bug** amoureux ou un **Bug** malade finira par redevenir normal.

### Interface :

Lors de la création de la partie, le joueur choisit un niveau de difficulté :

- Easy
- Medium
- Hard

Ces préférences influent sur :

- le nombre de **Bugs** sur la grille
- la proportion de la population malade et amoureuse
- le temps que peut survivre un **Bug** entre deux repas
- la durée de la maladie d'un **Bug**.
- la durée durant laquelle un **Bug** reste amoureux.
- la durée pendant laquelle la gamelle peut rester contaminée

Bien sur, vous disposez de tous les indicateurs qui vous permettent de connaître l'état de la population en temps réel. Vous pourrez ainsi intervenir rapidement et prendre le contrôle de n'importe quel individu et éviter ainsi le fameux message « GAME OVER » d'apparaître...