

## **- CAHIER DES CHARGES -**

### **Projet de Génie Logiciel : « Le gardien de parc »**

**Université Cergy Pontoise**

Auteurs : Karabouali Makoura  
Phetramphand Antoine  
Tangara Ramatoulaye

## Table des matières

I. Présentation du projet.....	3
1. Contexte.....	3
2. Objet.....	3
3. Organisation.....	3
a) analyse.....	3
b) spécification.....	3
c) conception.....	3
d) programmation.....	3
e) tests.....	4
f) validation.....	4
g) maintenance.....	4
4. Environnement.....	4
5. Planification.....	5
II. Objectif.....	6
1. Description brève du logiciel.....	6
2. Le but du jeu.....	6
3. Liste des fonctionnalités.....	6
4. Contraintes.....	7
III. Livraisons attendues	
.....	8
IV. Conception.....	9
IV. Glossaire.....	10

# **I. Présentation du projet**

## **1. Contexte**

L'intelligence artificielle est de plus en plus utilisée dans le monde d'aujourd'hui. Le multi-agents réactifs est une méthode de programmation pour créer des intelligences artificielles. Dans le cadre de notre formation à l'*Université de Cergy Pontoise*, les étudiants en Licence informatique deuxième année réalisent des projets personnels afin de valider leur année scolaire. Ce type de réalisation exige donc un certain temps, un investissement personnel et l'usage de toutes les ressources disponibles. Ici, nous allons réaliser notre projet sous la forme d'un jeu vidéo appelé « *le gardien de parc* ».

## **2. Objet**

Le but est la réalisation d'une application qui consiste à créer un jeu qui permet de chasser les intrus dans un espace donné et de les éliminer.

## **3. Organisation**

Notre équipe est composée de trois membres à la charge du projet : *Karabouali Makoura*, *Phetramphand Antoine* et *Tangara Ramatoulaye*.

Pour mieux appréhender le projet, nous allons entamer des recherches et approfondir les non acquis et mettre le problème en œuvre techniquement en utilisant le processus « *modèle en cascade* ». Il s'agit du modèle le plus simple des processus logiciels en termes de complexité et de facilité de mise en œuvre et repose sur les étapes suivantes :

### **a) analyse**

On essaye de comprendre le problème, puis d'analyser les besoins fonctionnels (contrainte technique).

### **b) spécification**

On décrit les besoins du client, puis on traduit les besoins en fonctionnalités.

### **c) conception**

On transforme le problème en solution.

### **d) programmation**

On passe du résultat de la conception à un ensemble de programmes traduit en un langage de programmation (Écriture des textes des programmes).

**e) tests**

On recherche des erreurs dans une spécification ou programme.

**f) validation**

Le système répond aux exigences du client et s'assure de l'adéquation des résultats de l'analyse et de la spécification.

**g) maintenance**

On veille au bon fonctionnement des programmes du projet.

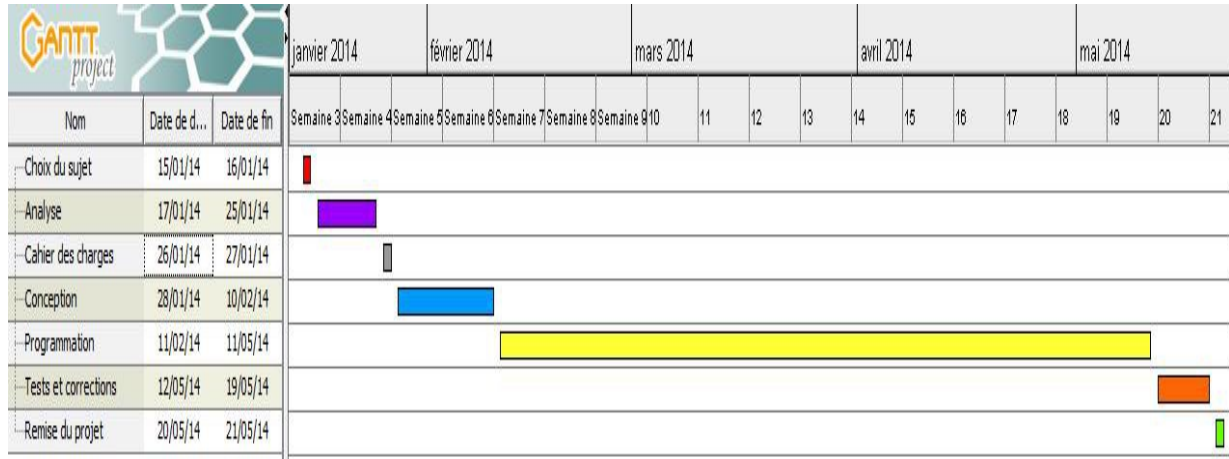
## **4. Environnement**

Les ressources que nous disposons pour le développement de notre projet sont :

- **Éclipse** : un logiciel de développement. Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation permettant de réaliser notre projet en utilisant le langage Java.
- **SVN** : Subversion est un logiciel de gestion permettant de stocker , partager des fichiers texte informatique, et dispose d'un mécanisme intelligent de fusion des modifications apportées. C'est un outil très utilisé pour le développement de logiciels.
- **Latex** : c'est un système de production de documents. Il ne s'agit pas de composer son texte, mais il s'agit d'utiliser un jeu de commandes décrivant ce que l'on souhaite obtenir.

## 5. Planification

**Délai** : le projet doit se finir avant le 21/05/2014



## **II. Objectif**

### **1. Description brève du logiciel**

L'objectif du projet consiste à développer un jeu ou une application basé sur un système de type «*multi-agents réactifs*» mettant en place comme éléments : un gardien, des intrus et des éléments du décor ou de l'interface graphique (eau, arbre, mur...).

### **2. Le but du jeu**

Cela consiste à l'utilisateur à prendre en main un gardien qui a pour but d'attraper les intrus présents dans un environnement donné (ici un parc). L'idée est divertir l'utilisateur afin qu'il soit immergé dans le cadre du divertissement.

### **3. Liste des fonctionnalités**

Cette application permettra de réaliser les fonctionnalités suivantes :

- **Présenter au joueur une interface d'entrée dans le jeu** : cette interface expliquera à l'utilisateur la tâche qu'il devra accomplir,
- **Mettre en évidence des éléments importants de la scène en les surlignant** : on met en valeur, en premier plan, les gardiens et les intrus afin de savoir pour l'utilisateur qui on peut contrôler et qui sont les cibles à éliminer ; puis en second plan les obstacles du jeu(eau, arbre, mur...), afin de permettre au joueur de prendre en compte ces éléments et leurs fonctions,
- **Offrir une simplicité d'utilisation** : le jeu devra être facile à utiliser et à manipuler, c'est-à-dire moins de touches du clavier à utiliser, suivi des touches de la souris.
- **Offrir une rapidité de mise en œuvre** : le jeu doit permettre à des joueurs de mesurer leurs réflexes mentaux afin qu'il puisse prendre des décisions (ex : trouver un chemin le plus court possible pour arriver jusqu'à l'intrus) le plus rapidement possible lors de la partie du jeu.
- **Assurer des temps d'exécution faibles** : lors du lancement du logiciel, les temps de chargement des programmes doivent être rapides.
- **Assurer des temps d'interactions joueur / machine faibles** : durant l'exécution du logiciel, l'utilisateur ne doit sentir aucune latence, que ce soit avec le clavier ou la souris. Les déplacements du joueur doivent être fluides.

#### **4. Contraintes**

- Problème de repérage des intrus par le gardien lié a son champ de vision limité.
- Détermination de la position des intrus
- Présence de multiple intrus dans l'espace.
- Déplacements et simulation des intrus.

### **III. Livraisons attendues**

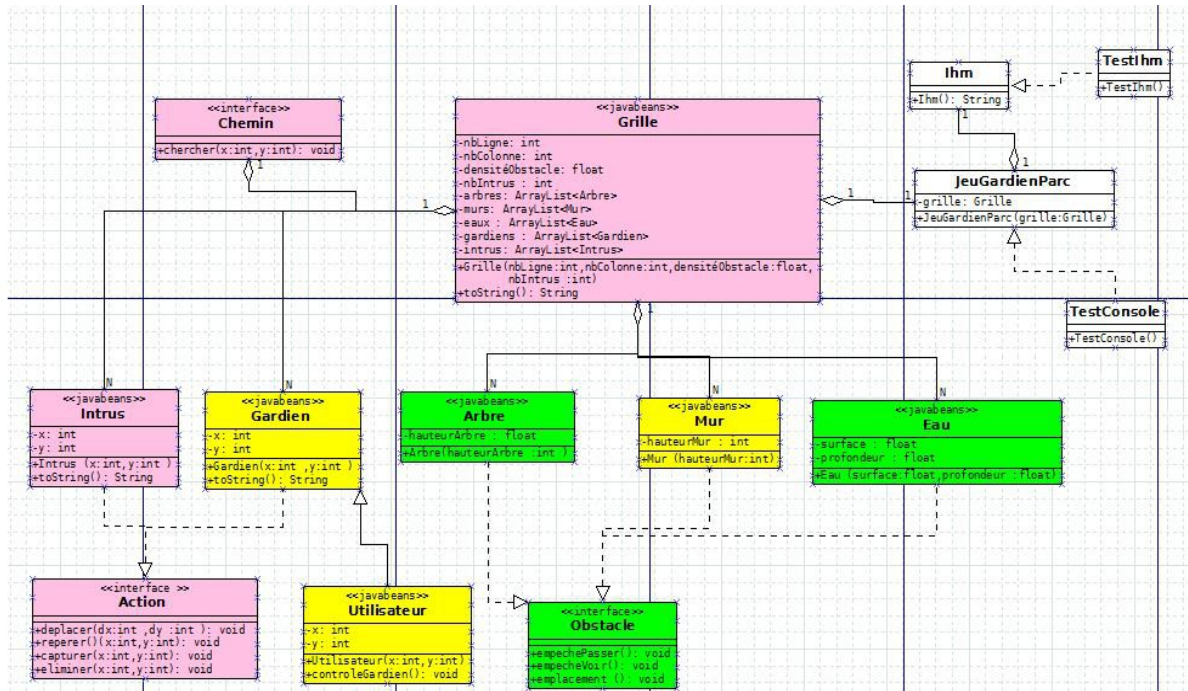
À la fin de notre projet, nous pensons rendre les documents suivants :

- des programmes en langage *java* comportant une interface graphique
- l'exécutable jar des programmes
- manuel d'utilisation du logiciel
- un rapport
- un cahier des charges final
- un planning (en cours)
- *javadoc*



## IV. Conception

Voici notre diagramme de classes ci-dessous :



## **IV. Glossaire**

**Agent** : Entité virtuelle autonome qui interagit avec son environnement. Ici, un agent appartient à un groupe et un rôle et peut communiquer avec les autres agents de la communauté.

**Cahier des charges** : document visant à définir simplement les spécifications d'un produit ou d'un service à réaliser.

**Eclipse** : un environnement de développement (*IDE*) Open Source principalement destiné au langage *JAVA* et fonctionnant à base d'extensions (*plugins*).

**Java** : langage de programmation orienté objet.

**Javadoc** : outil développé par *Sun Microsystems* permettant de créer une documentation d'*API* en format *HTML* depuis les commentaires présents dans un code source en Java. Il est le standard industriel pour la documentation des classes *Java*.

**Jeu vidéo** : il s'agit d'un jeu électronique impliquant une interaction humaine avec une interface «utilisateur» afin de générer un retour visuel sur un dispositif vidéo.

**Latence** : en informatique et plus spécifiquement sur les réseaux informatiques, aussi appelée «*lag*», délai entre le moment où une information est envoyée et celui où elle est reçue. De façon plus générale, la latence peut aussi désigner l'intervalle entre la fin d'un événement et le début de la réaction à celui-ci.

**Modèle en cascade** : il s'agit d'un cycle de vie d'un logiciel, soit un ordonnancement de différentes étapes du processus du développement. Pour ce type de modèle, cela exécute des phases qui ont pour caractéristiques : produire des livrables définis au préalable ; se terminer à une date précise ; ne se terminer que lorsque les livrables sont jugés satisfaisants lors d'une étape de validation-vérification.

**Partie (jeu vidéo)** : cela correspond à la durée de l'appropriation du monde virtuel par le joueur. Cette partie peut se dérouler jusqu'à ce que le jeu soit fini ou jusqu'à ce que le joueur se lasse.

**Plug in** : Application étendant les capacités et les fonctionnalités d'une autre application.

**Système Multi-agents réactifs** : un système composé d'un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations. Un agent est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome. Cela peut être un processus, un robot, un être humain, etc.