

Projet de Programmation Orientée Objet

Cahier des Charges

The Caterpillar Quest

Etudiants : Cyril CARON

Pauline HOULGATTE

Titouan PASQUET

Maxime PINEAU

Professeur Encadrant : Julien COHEN



Établi le lundi 4 mai 2015

Polytech Nantes - Formation Informatique
Site de la Chantrerie
Rue Christian Pauc - BP50609
44306 Nantes cedex 3 - France
Tél. : +33(0)2.40.68.32.00
www.polytech.univ-nantes.fr

Table des matières

1	Présentation générale du problème	2
1.1	Projet	2
1.2	Contexte	2
1.3	Enoncé du besoin	2
1.3.1	Le client	2
1.3.2	Les éléments visuels du jeu	2
1.3.3	Le lancement du jeu	4
1.3.4	Le déplacement du personnage	4
1.3.5	Scrolling, le déplacement de l'écran	5
1.3.6	Les scores	5
1.3.7	La fin de la partie	5
1.3.8	La génération de niveau	5
2	Expression fonctionnelle du besoin	6
2.1	Fonctionnalités principales	6
2.1.1	Jouer une partie	6
2.2	Fonctionnalités complémentaires	8
2.2.1	Déplacer un joueur	8
2.2.2	Génération de niveau / chargement des cartes	8
2.3	Contraintes techniques	8
3	Données métiers	9
4	Date de livraison du produit	10

1. Présentation générale du problème

1.1 Projet

Le but de ce projet est de créer un jeu de plateforme, où un joueur devra déplacer un personnage jusqu'à la fin d'un niveau.

1.2 Contexte

Ce projet est développé dans un cadre pédagogique, ainsi que pour réutiliser les différents concepts du développement objet avec le langage Java et des concepts de génie logiciel. Lors de ce projet, l'enseignant tuteur jouera le rôle du client.

Les différents contributeurs du projet sont le client et les membres du groupe.

1.3 Enoncé du besoin

1.3.1 Le client

Le client désire un jeu de plateforme, où il pourra jouer et déplacer un personnage sur une carte. Le but du jeu sera de déplacer le personnage depuis un point de départ (la zone d'apparition du joueur) jusqu'à une sortie, et ce en mettant le moins de temps possible.

1.3.2 Les éléments visuels du jeu

Le décor du jeu sera composé des éléments suivants :

- Le personnage du joueur
- des cerises
- des guêpes
- des plates-formes
- une ou plusieurs sorties de niveau
- le score du joueur
- une image de fond fixe

Ces éléments sont décrits dans les parties ci-dessous. Le design de chaque élément graphique est indifférent au client, excepté pour le personnage. Le client souhaite valider le design du personnage jouable à partir d'une liste de propositions à fournir.

Le personnage

Le personnage doit être vu de profil.

Il peut se déplacer vers la droite ou la gauche (mais de façon limitée, selon une zone qui sera précisée dans la section Scrolling). Il peut sauter au-dessus de lui (vers le haut) et en diagonale.

Les plate-formes

Des plate-formes seront présentes sur la carte. Elles représenteront soit le sol, soit des zones en l'air accessible par le personnage grâce à un saut.

Les plate-formes seront des objets infranchissables, c'est à dire que ni le joueur, ni les ennemis ne seront capable de les traverser.

Elles pourront être espacées entre elles. Un espacement entre deux d'entre elles qui composent le sol sera défini comme un trou.

Un mur sur la carte sera représenté par un empilement de plate-formes.

La taille d'une plateforme est indifférente au client.

Les cerises

Certains objets pourront être ramassés par le joueur, ceux-ci seront représentés par des cerises. Elles auront une position fixe et disparaîtront au contact du joueur.

Un compteur sera affiché durant le jeu pour indiquer combien de ces dernières ont été ramassées par le joueur.

Les guêpes

Le joueur devra également éviter des ennemis représentés sous forme de guêpes. Ces dernières auront une trajectoire fixée à l'avance sous forme de va-et-vient, et se déplaceront soit verticalement, soit horizontalement (une guêpe ne pourra pas changer de trajectoire). Les guêpes ne pourront pas être tuées par le joueur, ne pourront pas entrer en collision avec les plate-formes et passeront au travers des cerises.

La sortie du niveau

La sortie d'une carte sera représentée par une porte fixe, et ne sera pas animée.

Le fond

Le décor contiendra aussi une image de fond, qui ne fera pas partie des zones jouables et qui sera donc indépendante du reste.

1.3.3 Le lancement du jeu

Le jeu se lancera en ligne de commandes. Le nom (ou pseudo) du joueur sera à indiquer en paramètre lorsque le joueur démarrera le jeu.

Deux possibilités de lancement via la ligne de commandes :

```
Jeu NomJoueur fichierNiveau
```

```
Jeu NomJoueur
```

Dans le deuxième cas, un niveau par défaut est sélectionné.

1.3.4 Le déplacement du personnage

Les touches utilisées

Les touches utilisées pour récupérer les actions que veut effectuer le joueur sont les suivantes :

- → : déplace le personnage du joueur vers la droite si cela est possible
- ← : déplace le personnage du joueur vers la gauche si cela est possible
- ↑ : fait sauter le personnage vers le haut.
- ← ↑ ou ↑ ← : fait sauter le personnage vers la gauche.
- → ↑ ou ↑ → : fait sauter le personnage vers la droite.

Toute autre touche ou combinaison de touches n'est pas prise en compte, elle ne modifie donc pas le déplacement du personnage s'il était en train de se déplacer, et n'a aucune autre interaction sur le jeu ou sur la fenêtre.

Exemple : Si une touche →, ← ou ↑ est appuyée dans un premier temps, puis une autre touche qui ne permet pas de faire une combinaison comme ← ↑ ou → ↑, le personnage continue sa trajectoire comme si la deuxième touche n'était pas prise en compte.

La zone de déplacement du personnage

Le personnage ne pourra se déplacer que sur des plate-formes. Pour accéder à une plateforme en l'air, et comme les plate-formes sont infranchissables (voir partie plateforme), le joueur devra faire effectuer un saut en diagonale à son personnage.

1.3.5 Scrolling, le déplacement de l'écran

L'écran se déplacera seulement vers la droite, et devra rester fixe tant que le personnage du joueur n'a pas dépassé le premier tiers de l'écran puis se décaler vers la droite en fonction du déplacement indiqué par le joueur.

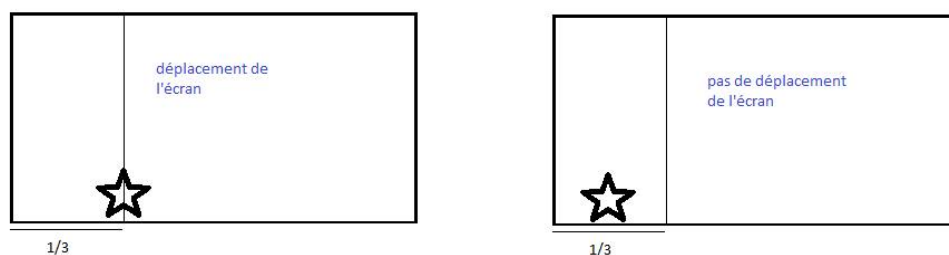


FIGURE 1.1 – schéma représentant le déplacement de l'écran

1.3.6 Les scores

Le score d'un joueur sera composé du nombre de cerises ramassées par ce joueur, ainsi que du temps de parcours du niveau.

Pour comparer deux scores, on comparera tout d'abord le nombre de cerises. En cas d'égalité, le départage s'effectuera en fonction du temps de parcours du niveau.

1.3.7 La fin de la partie

Le jeu peut s'arrêter de deux façons :

- le joueur gagne la partie, c'est à dire qu'il a atteint la porte de fin de niveau de la carte sans toucher d'ennemis.
- le joueur perd la partie, soit parce qu'il a touché un ennemi, soit parce qu'il est sortie de la zone affichée à l'écran et est donc tombé dans un trou.

On affichera à la fin d'une partie le score du joueur, ainsi que les noms et scores des cinq meilleurs joueurs dans le terminal ayant lancé le jeu.

1.3.8 La génération de niveau

Le programme devra être capable d'interpréter un fichier texte fourni et de créer un niveau si la syntaxe du fichier est correcte. Le client nous a laissés le choix de cette dernière.

2. Expression fonctionnelle du besoin

Les différentes fonctionnalités qu'un utilisateur pourra effectuer sont répertoriées dans le diagramme de cas d'utilisation suivant :

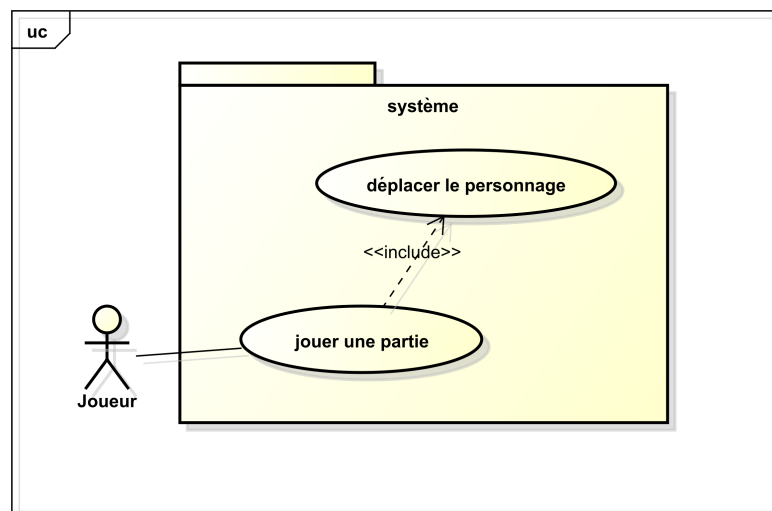


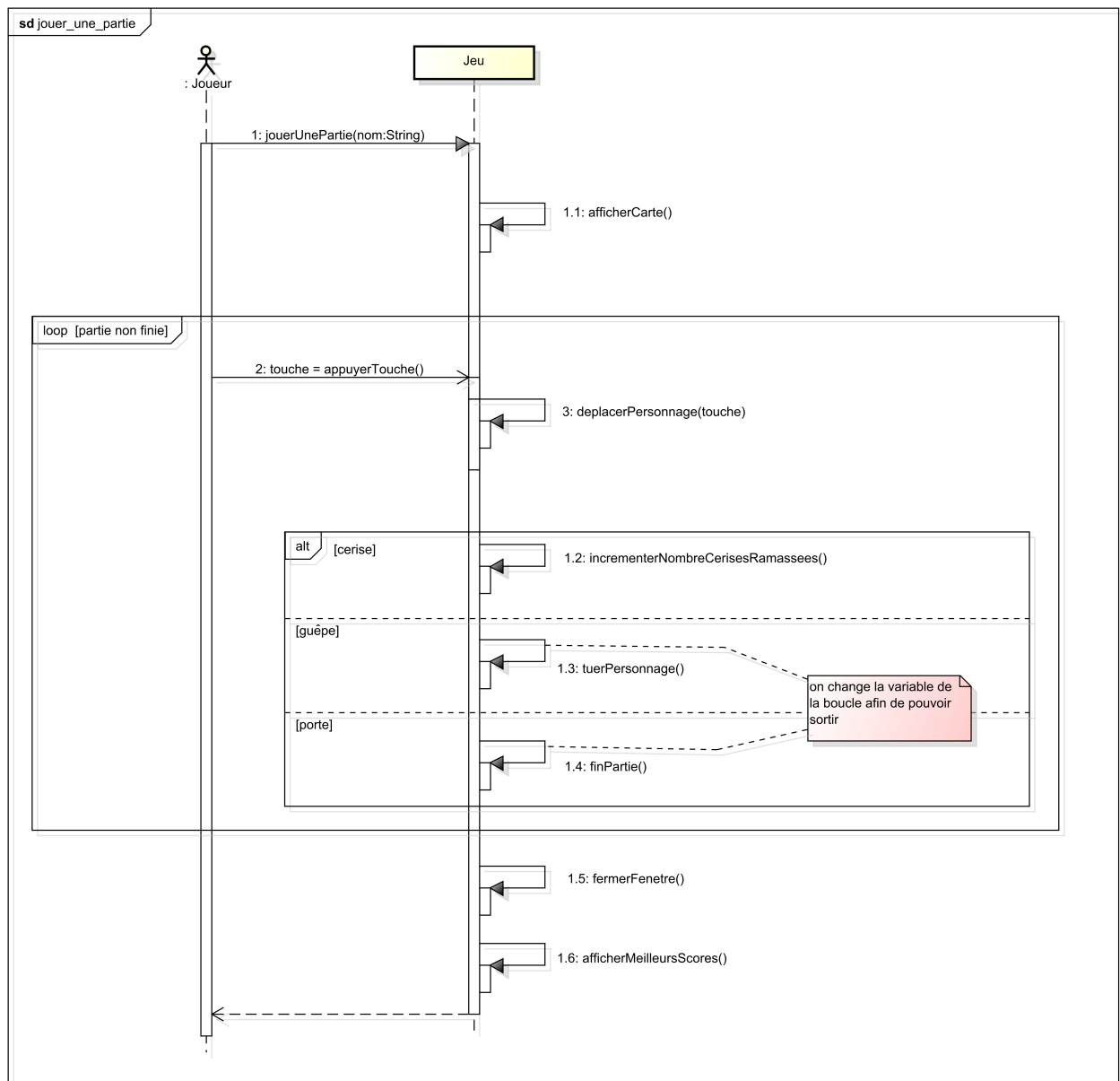
FIGURE 2.1 – Diagramme de cas d'utilisations du programme

- **jouer une partie** : Cette fonctionnalité permettra au joueur de lancer une nouvelle partie.
- **déplacer le personnage** : Cette fonctionnalité est appelée lors de la fonctionnalité "jouer une partie", elle permet de déplacer sur l'écran le personnage du joueur en fonction de l'appuie d'une touche.

2.1 Fonctionnalités principales

2.1.1 Jouer une partie

La fonctionnalité "Jouer une partie" permettra au joueur de lancer une partie. Lors de ce lancement, une nouvelle fenêtre s'ouvrira sur l'écran de l'utilisateur.



Scénario nominal : Le joueur lance une partie. La carte s’affiche à l’écran. Le joueur déplace le personnage à l’aide des touches directionnelles. Une fois que le personnage arrive à la fin du niveau, la partie se termine. La fenêtre se ferme et la liste des cinq meilleurs scores s’affiche dans le terminal.

Scénario alternatif 1 : Pendant un déplacement, le personnage touche une cerise. Le score du joueur s’incrémente et la cerise disparaît.

Scénario alternatif 2 : Le personnage entre en collision avec un ennemi, ce qui provoque la fin de la partie.

Scénario alternatif 3 : Le personnage tombe dans un trou et quitte l’écran, ce qui provoque la fin de la partie.

2.2 Fonctionnalités complémentaires

2.2.1 Déplacer un joueur

Lorsque le joueur joue sa partie, il peut appuyer sur certaines touches pour déplacer son personnage sur les côtés, vers le haut, ou en diagonale.

2.2.2 Génération de niveau / chargement des cartes

Le client aura la possibilité de créer lui même (ou via une tierce personne) un niveau à l'aide d'un fichier texte, en respectant une syntaxe particulière. Nous devons mettre au point cette syntaxe.

2.3 Contraintes techniques

Plusieurs contraintes techniques nous ont été imposées pour ce projet :

- le jeu devra se lancer via l'utilisation du terminal.
- le jeu devra être développé en Java.
- le jeu devra pouvoir fonctionner sur une distribution Linux.

3. Données métiers

Différents types d'informations seront manipulés par le programme, comme la position de chaque élément graphique par exemple. Ces informations sont regroupées dans le diagramme de classes métiers suivant :

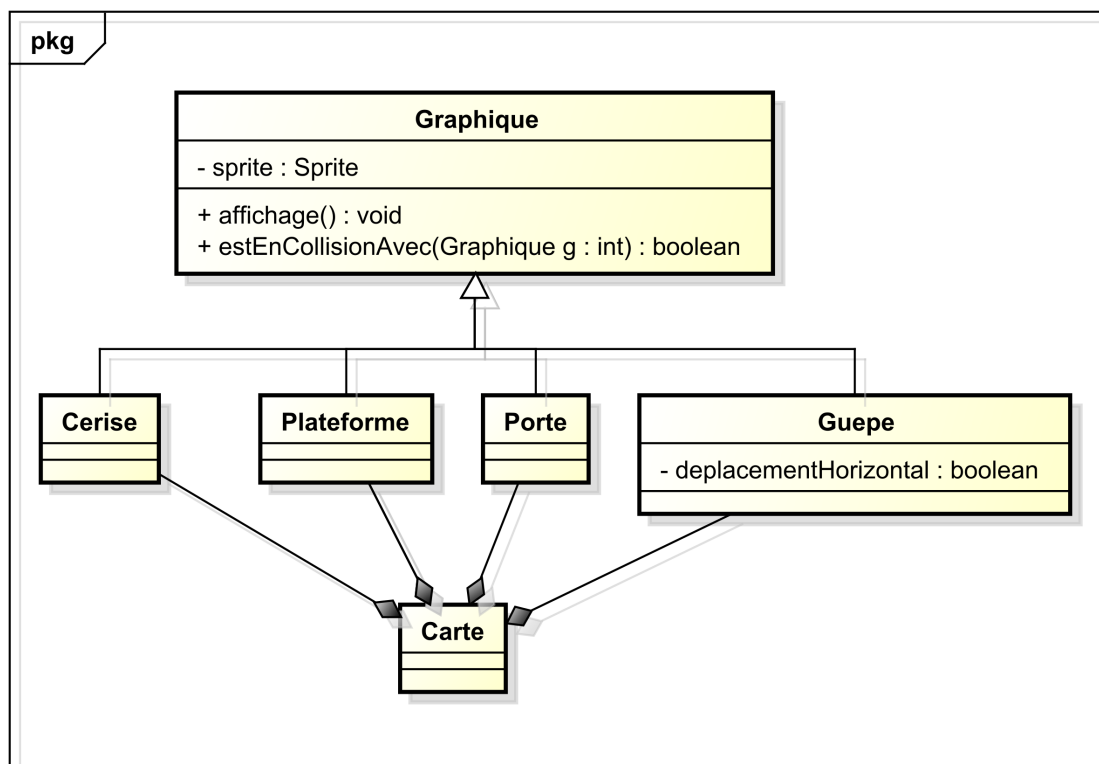


FIGURE 3.1 – Diagramme de classes métiers

Toutes les classes représentant des objets graphiques hériteront d'une classe Graphique, permettant ainsi à ces objets d'avoir tous une méthode affichage() et entreEnCollisionAvec().

Une classe Cerise représentera un objet cerise ramassable. En suivant le même principe, une classe Guêpe représentera une guêpe. Un attribut booléen permettra de savoir si la guêpe se déplace horizontalement ou verticalement.

4. Date de livraison du produit

La livraison du produit est prévue pour le 5 juin 2015 (lancement du projet le 7 avril 2015).