

# **Projet de synthèse**

## **Cahier des charges**

### **Jeu de plate-forme**

# Sommaire

I – Introduction.....	3
I.1 - Nature de la fourniture.....	3
I.2 - Compétences requises.....	3
I.3 - Définitions, sigles et abréviations.....	3
II - Présentation du projet.....	4
II.1 - Contexte.....	4
II.2 - Description générale.....	4
III – Livrables.....	5
IV – Fournitures du Maître d'Ouvrage.....	5
V – Description de la solution technique.....	5
VI – Description du domaine.....	6
VII – Description détaillée du produit.....	7
VII.1 - Interface.....	7
VII.2 - Interactions détaillées.....	8
VII.3 Exigences fonctionnelles.....	8
VII.4 - Exigences non-fonctionnelles.....	8
VIII – Annexes.....	9
Grille de réponse.....	9

# **I – Introduction**

## ***I.1 - Nature de la fourniture***

L'objet du présent cahier des charges est la fourniture d'un jeu de plate-forme en défilement horizontal. Ce jeu sera disponible dans une version pour Windows et une pour GNU/Linux.

## ***I.2 - Compétences requises***

La réalisation du livrable décrit dans le présent cahier des charges requiert de solides connaissances en programmation orientée objet, et plus particulièrement dans le langage C++, et une bonne maîtrise de la syntaxe UML pour la conception. Le projet sera conduit en appliquant les méthodes de gestion étudiées en cours.

## ***I.3 - Définitions, sigles et abréviations***

Scrolling : défilement, dans un jeu en scrolling horizontal, l'écran suit les déplacement du personnage le long d'un axe horizontal.

SFML : Simple and Fast Multimedia Library

Hitbox : zone de contact, la collision entre deux hitboxes déclenche un événement (par exemple, si la hitbox correspondant à l'arme du personnage touche la hitbox globale d'un ennemi, celui-ci est blessé).

# **II - Présentation du projet**

## ***II.1 - Contexte***

Polytech' Marseille est une école d'ingénieurs qui propose un cursus Informatique, Réseaux et Multimédia, anciennement dispensé par l'École Supérieure d'Ingénieurs de Luminy. En fin de deuxième année, il est demandé aux étudiants de réaliser un projet de synthèse faisant appel aux différentes compétences acquises durant cette année.

Dans ce cadre, M. Marc DANIEL, supervise un projet de développement de jeu en C++. Le sujet est libre, mais avec la contrainte de mettre en œuvre de nombreuses interactions entre objets. Le présent cahier des charges décrit le jeu que nous avons choisi de développer pour ce projet.

## ***II.2 - Description générale***

L'objectif du projet est de développer un jeu d'action/plate-forme en scrolling horizontal. Le but du jeu est d'atteindre la fin du niveau située à l'extrémité droite, en partant de l'extrémité gauche.

Le joueur déplace son personnage vers la droite ou la gauche sur des plates-formes, avec la possibilité de sauter pour atteindre un endroit élevé ou franchir un obstacle. La première difficulté est donc de parcourir le niveau sans tomber dans le vide ou dans tout autre piège présent. Dans cette optique, le joueur sera également confronté à des énigmes dans lesquelles il devra réfléchir au moyen de franchir un obstacle. Divers objets rencontrés dans le niveau pourront être déplacés par le personnage pour résoudre ces énigmes.

En plus des obstacles, le personnage est confronté à des ennemis, qui pourront utiliser diverses capacités, dont celle d'interagir avec le décor, pour éliminer le personnage. Ces ennemis pourront être combattus ou simplement évités. Ils se trouvent seuls ou en groupe essayant de coordonner leurs efforts pour arriver à leurs fins.

La réussite du joueur sera mesurée par un score calculé à la fin du niveau en fonction du temps écoulé pour atteindre l'arrivée, du nombre d'ennemis tués, etc.

### **III – Livrables**

Les fournitures suivantes seront livrées à l'issue du projet :

- Le cahier des charges réactualisé
- Deux versions du jeu pour Windows et GNU/Linux
- Les codes sources du jeu (archive + dépôt)
- Le rapport de projet

### **IV – Fournitures du Maître d'Ouvrage**

Le Maître d'Ouvrage fournira un accès aux outils nécessaires à la programmation, à savoir des ordinateurs fonctionnant sous Windows ou GNU/Linux et connectés à Internet. Il mettra également à disposition les supports de cours pouvant aider à la réalisation du projet.

### **V – Description de la solution technique**

Le programme sera réalisé en langage C++ pour répondre aux contraintes imposées par le Maître d'Ouvrage.

La librairie graphique choisie est SFML. Elle est complète, facile d'utilisation, et écrite en C++.

Le choix de l'IDE est encore en cours d'étude, mais se portera vraisemblablement soit sur Devcpp, soit sur CodeBlocks.

# VI – Description du domaine

Ce chapitre décrit de manière conceptuelle, incomplète et non-normalisée les entités du domaine :

- Personnage
  - Position
  - Image
  - Points de vie

Le personnage se déplace en répondant aux commandes du joueur. Ses points de vie indiquent le nombre de fois qu'il peut être blessé avant de mourir.

- Ennemi
  - Type
  - Position
  - Image
  - Points de vie
  - Point faible

Un ennemi se déplace pour blesser le personnage, selon son type (qui détermine entre autres sa manière de se déplacer et d'attaquer), et selon la présence d'autres ennemis avec lesquels il va se coordonner. Ses points de vie indiquent le nombre de fois qu'il peut être blessé avant de mourir. Son point faible est la zone dans laquelle il doit être touché pour être blessé.

- Plate-forme
  - Position
  - Image

Une plate-forme est un objet statique qui n'a d'autre but que de permettre aux autres objets de se tenir dessus (les objets, à l'exception des plates-formes, sont soumis à la gravité).

- Décor interactif
  - Position
  - Image
  - Interaction

Un décor interactif est fixe, mais le personnage ou un ennemi peut accomplir une action particulière s'il se tient à proximité (par exemple, une échelle offre la possibilité de grimper).

- Objet interactif
  - Position
  - Image
  - Poids

Un objet interactif peut être déplacé par le personnage ou les ennemis. Le poids indique les interactions possibles : un objet léger peut être soulevé et lancé, un objet lourd peut être poussé ou tiré.

## **VII – Description détaillée du produit**

### ***VII.1 - Interface***

L'écran est configuré pour suivre les déplacements du personnage. Ainsi, le joueur voit toujours son personnage dans la partie centrale de l'écran. Une étroite bande verticale au centrée constitue une zone de liberté dans laquelle le personnage peut se mouvoir sans faire avancer ou reculer le scrolling. Lorsqu'il arrive en butée de cette zone, l'écran commence à suivre le déplacement du personnage de manière à ce sa position relative à l'écran reste centrée.

Le joueur contrôle son personnage à l'aide du clavier. Les commandes seront par défaut les suivantes :

- Flèche droite : déplacer le personnage vers la droite
- Flèche gauche : déplacer le personnage vers la gauche
- Flèche haut : faire grimper le personnage (uniquement à côté d'une échelle), naviguer dans le menu pause
- Flèche bas : faire descendre le personnage (uniquement à côté d'une échelle), ramasser un objet, naviguer dans le menu pause
- Espace : sauter
- A : attaquer (si aucun objet porté), lancer l'objet porté
- Entrée : mettre le jeu en pause, sélectionner une option du menu pause

Outre ces commande en jeu, les flèches directionnelles ainsi que les touches Entrée et Echap serviront à naviguer dans le menu principal.

## ***VII.2 - Interactions détaillées***

Les interactions entre objets seront gérées par un système de hitboxes et de collisions.

- Les objets autres que les plates-formes ne sont pas sujet à la gravité s'ils sont en collision avec la face supérieure d'une plate-forme
- Un ennemi est blessé lorsque la hitbox de l'arme du personnage le touche
- Un ennemi est blessé lorsque la hitbox du personnage entre en collision avec la sienne par le haut
- Dans tous les autres cas de collision entre la hitbox d'un ennemi et celle du personnage, c'est ce dernier qui est blessé
- Le personnage meurt lorsque sa hitbox entre en collision avec la limite inférieure de l'écran
- L'interaction avec un décor/objet interactif est possible si sa hitbox est en contact avec celle du personnage ou de l'ennemi
- Un objet interactif, lorsqu'il est lancé, est détruit lorsque sa hitbox rentre en collision avec celle du personnage ou d'un ennemi, qui est alors blessé

## ***VII.3 Exigences fonctionnelles***

- Gérer les actions du joueur

Cette fonctionnalité comporte la gestion des événements claviers provoqués par le joueur.

- Gérer les collisions

Cette fonctionnalité sera assurée par la gestion des hitboxes. Le programme vérifie en permanence si des hitboxes sont en contact, et résout la collision le cas échéant.

- Gérer la gravité

Cette fonctionnalité consiste à maintenir en mémoire un vecteur de force orienté vers le bas, qui s'applique au déplacement de chaque objet lorsqu'il n'est pas posé sur une plate-forme.

## ***VII.4 - Exigences non-fonctionnelles***

- Le jeu devra répondre aux commandes du joueur en moins de 20ms
- On utilisera au maximum les fonctions présentes dans la librairie SFML



## VIII – Annexes

### *Grille de réponse*

<b>Livrable</b>	<b>Durée estimée</b>	<b>Prix estimé</b>
Cahier des charges	1j	160€
Conception	2j	320€
Programme <ul style="list-style-type: none"><li>• Objets</li><li>• Personnages</li><li>• Ennemis</li><li>• Interactions</li></ul>	11j <ul style="list-style-type: none"><li>• 2j</li><li>• 2j</li><li>• 2j</li><li>• 5j</li></ul>	1760€ <ul style="list-style-type: none"><li>• 320€</li><li>• 320€</li><li>• 320€</li><li>• 800€</li></ul>
Graphisme	2j	320€
Tests & revues	3j	480€
Rapport	1j	160€
<b>Total</b>	<b>20j</b>	<b>3200€</b>

*Tarifs calculés sur la base de 22,86€/h couvrant la rémunération de deux jeunes ingénieurs à 10€/h.*