

# PROJET « SAVE THE KITTEN »

CAHIER DES CHARGES, SPÉCIFICATIONS FONCTIONNELLES,  
SPÉCIFICATIONS TECHNIQUES

JEREMY BIDEY; MELODY GALMICHE

# Table des matières

1	Introduction.....	3
2	Cahier des charges.....	3
2.1	Cible .....	3
2.2	Éléments.....	3
2.3	Règles .....	3
3	Spécifications fonctionnelles.....	4
3.1	Interface .....	4
3.2	Menu .....	4
3.2.1	Jouer .....	4
3.2.2	Règles .....	4
3.3	Sélection des niveaux .....	4
3.4	Espace de jeu.....	4
3.4.1	Canons.....	4
3.4.1.1	Canon simple .....	4
3.4.1.2	Canon rafale .....	4
3.4.1.3	Canon double fût.....	4
3.4.2	Filets .....	5
3.4.3	Chats.....	5
3.4.3.1	Chat lambda .....	5
3.4.3.2	Chat gymnaste.....	5
3.4.3.3	Chat griffu.....	5
3.4.4	Bombes.....	5
3.4.4.1	Bombe explosive .....	5
3.4.4.2	Bombe vortex .....	5
3.4.5	Murs.....	5
4	Spécifications techniques.....	6
4.1	Organisation .....	6
4.1.1	Core .....	6
4.1.2	GUI.....	6
4.1.3	Main.....	6
4.1.4	Tests.....	6
4.2	Architecture.....	6
4.2.1	Package.....	6
4.2.2	Classes .....	6

4.2.2.1	Core .....	6
4.2.2.2	GUI .....	7
4.2.2.3	Main.....	7
4.2.2.4	Tests.....	7
5	Annexes .....	8

## 1 Introduction

Ce document a pour objectif de présenter le projet de Java de M1. Ce projet consiste en la réalisation d'un jeu nommé « Save the Kitten ». Tous les éléments décrits ci-après constituent la version 1.0 du projet. Tous les éléments apparaissant dans le jeu mais ne figurant pas dans les écrits suivants seront considérés comme contenu additionnel et auront une version supérieure à 1.0. Toutefois, des ajouts pourront être insérés dans le document et seront précédés de leur version.

## 2 Cahier des charges

### 2.1 Cible

Un jeu de plateforme à plusieurs niveaux pour les enfants âgés de 8 ans et plus. Ce jeu met en scène des chats, qu'il faut sauver au moyen de bombes ( ? ^^' ).

### 2.2 Éléments

Une plateforme désigne l'espace de jeu.

Dans cette plateforme il y aura :

- Un canon qui projette les chats dans une direction.
- Des filets pour attraper les chats.
- Des murs pour bloquer les chats.
- Des bombes pour modifier la trajectoire des chats.
- Des chats, évidemment, à sauver d'une mort lente et épouvantable.

### 2.3 Règles

Le but du jeu est de sauver tous les chats à travers les niveaux. Un canon les envoie dans l'espace (pas celui qui est galactique) dans une direction prédéfinie. Pour les sauver, le joueur devra placer des bombes sur le terrain à des endroits stratégiques tout en tenant compte des éventuels murs. Il pourra choisir le moment de déclenchement des bombes. Lorsque tous les chats sont arrivés dans des filets distincts, il ne peut y avoir deux chats dans un même filet, le niveau est validé et il peut passer au suivant. Dans le cas contraire, le joueur a échoué et peut recommencer le niveau.

Le déroulement d'un niveau se passe en deux phases :

1. Phase de préparation
2. Phase de lancement

Durant la phase de préparation, le joueur peut placer ses bombes et choisir leur déclencheur.

Dans la deuxième phase, le joueur appuie sur le bouton « Play » qui va verrouiller le terrain de jeu et lancer les chats. L'utilisateur ne peut plus influencer la partie dans cette phase.

## 3 Spécifications fonctionnelles

### 3.1 Interface

Le jeu se lancera dans une résolution fixe mais en plein écran. Pour les grandes résolutions, la taille ne changera pas et l'écran sera complété avec des bandes noires autour. Pour les petits écrans, le comportement de l'affichage n'est pas défini.

Résolution définie : 1024x768 (4:3)

v1.2 : la résolution s'adapte aux écrans plus petits sans jamais dépasser la résolution maximale actuelle.

### 3.2 Menu

Le menu comporte deux boutons.

#### 3.2.1 Jouer

Le bouton « Jouer » permet de lancer l'écran de sélection des niveaux.

#### 3.2.2 Règles

Le bouton « Règles » permet d'afficher l'écran contenant les règles.

v1.1 : un tutoriel est disponible afin d'apprendre les règles en s'exerçant.

### 3.3 Sélection des niveaux

L'écran de sélection des niveaux permet de voir les niveaux déjà terminés et de choisir le niveau parmi ceux débloqués. Un niveau terminé donne accès au suivant. Ainsi si le joueur a terminé le niveau 1 et le niveau 2, il pourra choisir de rejouer les deux premiers niveaux ou d'essayer le troisième.

### 3.4 Espace de jeu

#### 3.4.1 Canons

Il existe trois types de canons. L'espace de jeu fixe la position du canon au début du niveau. Il se trouve toujours sur la gauche au centre du terrain et pointe vers le mur opposé de façon perpendiculaire.

v2.1 : Il est possible d'orienter le canon dans une direction différente.

##### 3.4.1.1 Canon simple

Ce canon tire tout droit et un chat à la fois. L'angle ne peut être choisi.

##### 3.4.1.2 Canon rafale

Ce canon tire tout droit et tous les chats successivement à intervalles réguliers. L'angle ne peut être choisi. L'intervalle est prédéfini et n'appartient pas au joueur.

v2.0 : Il est possible de choisir la fréquence de tir du canon.

##### 3.4.1.3 Canon double fût

Ce canon tire deux chats simultanément dans deux directions opposées de 45°. L'angle ne peut être choisi.

### 3.4.2 Filets

Il n'existe qu'un seul type de filet dans le jeu. Le filet doit accueillir exactement un chat, ni plus ni moins. Si un filet reste vide, la partie est considérée comme perdue. Si un chat arrive dans un filet déjà occupé, il est « envoyé dans un autre espace temps » par le premier chat et la partie est considérée comme échouée.

Il y a autant de filet dans l'espace de jeu qu'il n'y a de chats dans le « chat-rgeur ».

### 3.4.3 Chats

Il existe trois types de chats. Les chats sont projetés à la même vitesse mais sont rapidement agacés de se déplacer. Ils ralentissent avec le temps pour finir par s'arrêter définitivement.

#### 3.4.3.1 Chat lambda

C'est le chat « classique ». Il n'a aucun talent particulier, mais il sait particulièrement bien s'écraser contre les murs.

#### 3.4.3.2 Chat gymnaste

Le chat gymnaste est très souple comme son nom l'indique et chaque obstacle est pour lui comme un arbre à chat où il peut s'élancer sans difficultés. Il rebondit sur les murs.

#### 3.4.3.3 Chat griffu

Le chat griffu lui est bien plus teigneux. Avec ses grosses griffes, il s'accroche à tous les obstacles qu'il rencontre. Il s'agrippe aux murs, se déplace sur l'extrémité la plus proche et se lance tout droit. Comme tous les autres chats, il s'épuise rapidement.

### 3.4.4 Bombes

Afin de modifier la trajectoire des chats, il existe deux types de bombes.

#### 3.4.4.1 Bombe explosive

La bombe explosive, explose au bout d'un certain temps. L'explosion provoque un souffle, qui ne fait pas une bouillie de chats, mais expulse les chats à proximités dans une autre direction. La direction est calculée selon l'inverse à la normale. L'explosion de la bombe provoque un regain d'énergie pour le chat et lui permet d'aller plus loin. La puissance de la bombe varie selon la distance entre le chat et la bombe au moment de l'explosion.

#### 3.4.4.2 Bombe vortex

La bombe vortex, implose au bout d'un certain temps. Cette implosion a pour effet d'aspirer tous les chats à proximités, modifiant ainsi leur trajectoire. L'implosion ne provoque pas d'effets secondaires tels qu'un regain d'énergie. La puissance d'absorption varie avec la distance entre le chat et la bombe au moment de l'implosion.

### 3.4.5 Murs

Dans l'espace de jeu, zéro, un ou plusieurs murs peuvent être présents. La position, la taille et l'inclinaison de ces murs est fixée avant le début du jeu. Elle appartient au niveau et l'utilisateur ne peut influencer dessus.

## 4 Spécifications techniques

### 4.1 Organisation

La réalisation du projet est divisée en 4 parties.

#### 4.1.1 Core

Le « Core » correspond à toute la mécanique du jeu :

- le calcul des trajectoires
- les effets des events : explosion, implosion
- les objets représentant les différents éléments du jeu : chats, bombes, canons, filets, murs
- la gestion du travail parallèle

#### 4.1.2 GUI

La « GUI » définit l'implémentation des éléments graphiques du jeu et fera le lien entre les events et les animations.

#### 4.1.3 Main

Le « Main » instancie le jeu, les niveaux, les éléments et lance les routines du jeu.

#### 4.1.4 Tests

La phase de « Tests » permet de vérifier le bon comportement du code en toute circonstance.

### 4.2 Architecture

#### 4.2.1 Package

Les packages ont pour préfixes `fr.upem` et pour suffixes `core`, `gui`, `main`, `tests`.

#### 4.2.2 Classes

##### 4.2.2.1 Core

Chaque élément du jeu est représenté par un objet dont le comportement est défini par une classe.

Une classe `Cat` définit le comportement commun de tous les chats. Les trois types de chats sont définis dans la classe `Cat` qui a un champ `type` de type énumération qui fait référence au type de chat : `lambda`, `gymnaste` ou `griffue`. C'est ce champ qui définit le comportement spécialisé du chat. Un champ `celerity` définit la vitesse de déplacement du chat.

Une classe `Bomb` définit le comportement commun à toutes les bombes. Les deux types de bombes sont définis dans la classe `Bomb` qui a un champ `type` de type énumération qui fait référence au type de la bombe : `vortex` ou `explosive`. La classe contient également un champ `timer` de type `long` qui définit le temps en milliseconde du déclencheur choisi par le joueur. Un champ `potency` de type `long` qui définit la force attractive (négatif) ou répulsive (positif) des bombes.

Une classe `Canon` définit le comportement général des canons. Un champ `type` de type énumération permet de distinguer chaque type de canon et leur comportement associé : `simple`, `rafale`, `double fût`. Un champ `timer` définit le temps entre chaque tir. Un champ `power` définit la puissance de projection des chats (vitesse initiale).

Une classe `Net` définit le comportement d'un filet. Cette classe n'a qu'un champ `empty` de type booléen qui est à `false` si le filet est occupé par un chat, sinon à `true`.

Une classe `Wall` représente les murs. Cette classe ne contient pas de champs particuliers.

#### 4.2.2.2 GUI

Le package `GUI` contient les classes qui permettent de mettre en place graphiquement les différents éléments du jeu.

Il contient notamment une classe `Element` qui possède un champ `position` indiquant la position en  $(x, y)$  de l'élément dans l'espace de jeu. Les classes `Cat`, `Bomb`, `Wall`, `Net`, `Canon` héritent de cette classe. Pour chaque instance de la classe `Element`, un thread est assigné et exécute un code associé à son type.

- `Cat` : un thread mettant à jour la position et la vitesse du chat.
- `Bomb` : un thread exécutant le décompte du déclencheur et lançant l'événement associé (explosion, implosion).
- `Canon` : un thread exécutant le décompte du prochain tir et l'événement associé (un chat, rafale de chat, double chat en cône).
- `Net` : un thread vérifiant le statut du filet (`empty` or `filled`).
- `Wall` : pas de thread, objet statique

Suivant la librairie graphique qui nous est mise à disposition, la `GUI` contient également :

- Une classe `MainWindow` qui définit l'apparence (résolution, layout) de la fenêtre principale du jeu.
- Une classe `Menu` qui définit l'apparence du menu (position, bouton).
- Une classe `LevelGrid` qui définit l'apparence de la fenêtre de sélection des niveaux.
- Une classe `Rules` qui définit l'apparence de la fenêtre des règles du jeu.
- Une classe `Level` qui définit l'apparence de l'espace de jeu (éléments, layout).

#### 4.2.2.3 Main

Le package `Main` contient une classe `Main` avec la méthode « `main` » qui est exécutée par le programme. Cette classe instancie tous les éléments (`Core` et `GUI`) du jeu.

Elle lance les interfaces graphiques (fenêtre, menu, sélection des niveaux, niveau). Si le joueur décide de quitter un niveau en cours, le `main` interrompt alors tous les threads des éléments et libère les instances des classes du niveau. Puis revient à l'écran de sélection des niveaux. De là il peut revenir au menu principal, puis quitter le jeu.

#### 4.2.2.4 Tests

Les tests sont majoritairement unitaires et vérifient principalement l'exactitude des calculs de trajectoires dans l'espace de jeu.

D'autres tests permettent de vérifier la coordination des threads.



## 5 Annexes

### 1 Menu du jeu



*2 Espace de jeu*

3 Canon simple



4 Canon rafale



5 Canon double fût



6 Filet



7 Chat lambda



8 Chat gymnaste



9 Chat griffu

10 Bombe  
explosive  
désactivée11 Bombe explosive  
active12 Bombe  
explosive sur le  
point de péter13 Bombe  
explosive  
explosée14 Bombe  
vortex fermé15 Bombe vortex  
ouvert

*16 Mur*



17 Gros chat lambda



*18 Gros chat gymnaste*



19 Gros chat griffu

