

# Compte Rendu : Bomberman

Tanguy THOMAS

17 mai 2016

## Table des matières

<b>1</b>	<b>Cahier des Charges</b>	<b>1</b>
1.1	Présentation . . . . .	1
1.1.1	Règles du jeu . . . . .	1
1.1.2	Environnement . . . . .	2
1.1.3	Ce que le joueur peut ou doit faire . . . . .	2
1.1.4	Ce que le joueur ne peut pas faire . . . . .	2
1.1.5	Ce que les bombes peuvent ou doivent faire . . . . .	2
1.1.6	Ce que les bombes ne peuvent pas faire . . . . .	2
1.1.7	Ce que l'explosion peut ou doit faire . . . . .	2
1.1.8	Bonus . . . . .	3
1.2	Architecture du Programme : Fonctions . . . . .	3
<b>2</b>	<b>Compte-Rendu</b>	<b>4</b>
2.1	Recherche documentaire . . . . .	4
2.2	Travail d'équipe . . . . .	4
2.2.1	Github . . . . .	4
2.3	Répartition du travail . . . . .	4
2.3.1	Clément Guin . . . . .	4
2.3.2	Titouan Rannou . . . . .	4
2.3.3	Tanguy Thomas . . . . .	5
2.4	Mon rôle dans le projet . . . . .	6
2.5	Conclusion . . . . .	6

## 1 Cahier des Charges

### 1.1 Présentation





Le but de ce projet est de créer un clone du jeu vidéo *Bomberman*. Il s'agit d'un jeu multijoueur qui se joue à 2 joueurs. Voici les règles :

#### 1.1.1 Règles du jeu






Imaginez une arène (vue de dessus) composée de blocs incassables, de blocs cassables (briques) et de chemins praticables. Le jeu se joue à 2 joueurs, ce sont tous les 2 des Bombermans. Il peut se déplacer et poser des bombes. Après quelques secondes elles explosent et détruisent les blocs cassables proches. Les règles sont simples : il ne doit en rester qu'un. Il faut utiliser toute sa ruse, récupérer des items bonus et poser plusieurs bombes afin de venir à bout de votre adversaire.

### 1.1.2 Environnement

Nous décrivons dans cette section l'arène du jeu : La construction de l'arène se fait à l'aide des images suivantes :

- Des blocs *incassables* 
- Des blocs *cassables* 
- Des 2 personnages  

Il y aura aussi les items suivants dans l'arène :

- Les 2 bombes pour chaque personnage :  
- Le bonus de portée de la bombe 
- Le bonus qui donne un point de vie 
- Le bonus qui augmente le nombre maximal de bombes posés en même temps 

Chaque joueur commence à des coins opposés de l'arène. Ils peuvent se déplacer dans 4 directions, Nord, Sud, Est, Ouest. Ils ont chacun 3 vies en début de partie. Ils en perdent une s'ils sont dans la portée de la bombe. Si un joueur n'a plus de vie il a perdu.

### 1.1.3 Ce que le joueur peut ou doit faire

- Ne rien faire
- Se déplacer dans l'arène sur les cases vides
- Pouvoir récupérer des bonus
- Aller sur la même case qu'un joueur, une bombe ou un bonus
- Poser des bombes

### 1.1.4 Ce que le joueur ne peut pas faire

- Sortir de l'arène
- Traverser des obstacles (blocs et briques)
- Jouer une fois mort
- Poser plus de bombes en même temps que son nombre maximal

### 1.1.5 Ce que les bombes peuvent ou doivent faire

- Exploder au bout de un certain temps secondes
- Être traversées par les joueurs

### 1.1.6 Ce que les bombes ne peuvent pas faire

- Casser des blocs (éléments incassables)

### 1.1.7 Ce que l'explosion peut ou doit faire

- Enlever un point de vie au joueur qui a posé la bombe ou/et au joueur adverse

- Casser des briques (éléments cassables) correspondant à la portée du joueur
- Faire apparaître différents bonus

### 1.1.8 Bonus

- Vie
- Augmentation de la portée de la bombe

## 1.2 Architecture du Programme : Fonctions

Dans cette partie nous présenterons les différentes fonctions qui permettent le bon fonctionnement du jeu :

**menu** : Fait apparaître le menu

**startgame** : Démarre le jeu

**perdu** : Permet à l'utilisateur de rejouer ou de quitter la partie

**dessinermap** : Dessine l'arène de jeu

**ajouter\_brique** : Dessine une brique

**ajouter\_bloc** : Dessine un bloc

**bombe1** : Pose une bombe qui explose au bout d'un certain temps pour le joueur 1

**bombe2** : Pose une bombe qui explose au bout d'un certain temps pour le joueur 2

**explosion** : Fait exploser la bombe

**destruction\_animation\_explosion** : Détruit l'animation de la bombe

**bonus\_bombe** : Pose un bonus qui augmente la portée de la bombe aléatoirement

**verif\_bonus\_bombe** : Vérifie si le joueur 1 est sur le bonus de portée de bombe

**verif\_bonus\_bombe1** : Vérifie si le joueur 2 est sur le bonus de portée de bombe

**bonus\_vie** : Pose un bonus de vie de la bombe aléatoirement

**verif\_bonus\_vie** : Vérifie si le joueur 1 est sur le bonus de vie

**verif\_bonus\_vie** : Vérifie si le joueur 2 est sur le bonus de vie

**bonus\_recharge** : Pose un bonus qui augmente le nombre maximal de bombes

**verif\_bonus\_recharge** : Vérifie si le joueur 1 est sur le bonus de recharge

**verif\_bonus\_recharge1** : Vérifie si le joueur 2 est sur le bonus de recharge

**enleve\_vie** : Enlève un point de vie à un joueur

**rajoute\_vie** : Rajoute un point de vie à un joueur

**personnages** : Crée les 2 personnages

**animdroite** : Déplace le joueur 1 vers la droite

**animgauche** : Déplace le joueur 1 vers la gauche

**animbas** : Déplace le joueur 1 vers le bas

**animhaut** : Déplace le joueur 1 vers le haut

**animdroite2** : Déplace le joueur 2 vers la droite

**animgauche2** : Déplace le joueur 2 vers la gauche

**animbas2** : Déplace le joueur 2 vers le bas

**animhaut2** : Déplace le joueur 2 vers le haut

## 2 Compte-Rendu

### 2.1 Recherche documentaire

Passionnés d'informatique, nous connaissons le site openclassrooms (anciennement site du zéro), qui est un site de cours en ligne gratuit, orienté vers la programmation informatique. Nous avons donc suivi le cours d'initiation à python pour voir des notions que nous n'avions pas vu en cours, ou réviser et approfondir celles que nous avons vu en cours.

### 2.2 Travail d'équipe

#### 2.2.1 Github

Tout d'abord nous nous sommes heurtés à des difficultés pour travailler ensemble, partager et mettre en communs nos travaux. Nous avons donc cherché une solution : Git. Git est un logiciel de gestion de versions, c'est à dire qu'il permet de stocker nos fichiers en conservants la chronologie de toutes les modifications qui ont été effectuées dessus. Pour un travail d'équipe efficace nous avons utilisé le service web git hub, qui est un réseau social utilisant le système Git. Git Hub nous a permis de nous organiser, ne pas perdre la trace de nos anciennes modifications. Par exemple, lorsque l'utilisateur fait une modification dans son code, et le code ne marche plus, on peut revenir à une version antérieur ou le code marchait pour pouvoir identifier le problème et le régler.

### 2.3 Répartition du travail

Dans notre groupe, nous avons essayé au maximum de répartir le travail de façon équitable, afin d'être le plus efficace possible. Ainsi, nous avons réalisé, chacun de notre côté, des fonctions, que nous avons ensuite rassemblé, au fur et à mesure de l'élaboration de notre projet.

Voici les fonctions que nous avons donc réalisé, ainsi que leur répartition entre chaque membre du groupe :

#### 2.3.1 Clément Guin

- menu
- startgame
- perdu
- animdroite
- animgauche
- animbas
- animhaut
- animdroite2
- animgauche2
- animbas2
- animhaut2
- personnages
- enleve\_vie

#### 2.3.2 Titouan Rannou

- bonus\_bombe
- verif\_bonus\_bombe

- `verif_bonus_bombe1`
- `bonus_vie`
- `verif_bonus_vie`
- `verif_bonus_vie1`
- `bonus_recharge`
- `verif_bonus_recharge`
- `verif_bonus_recharge1`
- `dessiner_map`
- `ajouter_brique`
- `ajouter_bloc`

### **2.3.3 Tanguy Thomas**

- `bombe1`
- `bombe2`
- `explosion`
- `destruction_animation_explosion`
- `rajoute_vie`

## 2.4 Mon rôle dans le projet

J'ai tout d'abord créé 2 fonctions :

- Bombe1 pour le joueur 1
- Bombe 2 pour le joueur 2

Les 2 fonctions admettent le paramètre event pour leur dédier une touche. Après avoir vérifié si le joueur n'a pas dépassé le nombre de bombes maximal que le joueur peut poser en même temps. Si oui, elles posent une bombe aux coordonnées du joueur lorsque que l'utilisateur appuie sur la touche dédiée à chacune des 2 fonctions et on décrémente de 1 le nombre de bombes maximal que le joueur peut poser en même temps. Après 2,5 secondes, la fonction explosion est appelée.

La fonction explosion, qui tient une place importante dans le jeu, admet 3 paramètres : les 2 coordonnées de l'endroit où l'explosion doit se dérouler et de quel joueur il s'agit. Dans un premier temps on définit la valeur du bonus en fonction du joueur et on incrémente de 1 le nombre de bombes maximal que le joueur peut poser en même temps.

Dans la direction choisie, on affiche l'animation de l'explosion jusqu'à ce qu'on trouve un bloc, une brique ou que la portée de la bombe soit atteinte. S'il y a une brique, on ajoute son id à la liste destroy et on affiche aussi l'image de l'animation de l'explosion. S'il y a un bloc, on arrête d'afficher l'animation d'explosion de la bombe. S'il y a un joueur on lui enlève un cœur avec la fonction enleve\_vie. Et ceci est exécuté pour toutes les directions possibles, soit en haut, en bas, à droite et à gauche. On supprime tous les éléments de la liste destroy qui correspondent aux briques qui doivent être détruites. On supprime l'image de bombe que le joueur a posée, et on la remplace par une image d'animation d'explosion et pour finir on appelle la fonction destruction\_animation\_explosion

La fonction destruction\_animation\_explosion qui prend comme seul paramètre le joueur qui a posé la bombe. Elle s'occupe de détruire les images d'animations d'explosions pour chaque joueur.

La fonction rajoute\_vie qui a pour paramètre le joueur qui doit gagner un cœur. On incrémente de 1 le nombre de cœurs du joueur en question et dans le canvas score on affiche un cœur de la couleur du joueur correspondant à un point de vie à l'emplacement qui lui correspond. Si le joueur à 4 cœurs on décrémente de 1 sa vie pour limiter le nombre de cœurs à 3.

## 2.5 Conclusion

Etant passionné d'informatique, j'ai apprécié travailler sur ce projet d'ISN. Le travail de groupe a été une expérience intéressante. De plus le travail de groupe est indispensable dans le milieu de l'informatique, cela nous a donc permis d'avoir une approche du travail de groupe. Mes camarades étaient sérieux et impliqués dans leur travail. Nous avons donc eu une bonne cohésion de groupe qui nous a permis de travailler efficacement. Ce projet m'a convaincu que j'aimerais faire de l'informatique mon métier.