Compte Rendu : Bomberman

Clément Guin

17 mai 2016

Table des matières

1 Cahier des Charges **1**
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

2 Compte-Rendu **4**
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 5
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 régles :

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 régles 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

— Exploser au bout de un certain temps secondes
— Être traversées par les joueurs

1.1.6 Ce que les bombes ne peuvent pas faire

— Cassser 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 adversaire

— 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 apparaitre le menu

startgame : Démarre le jeu

perdu : Permet a 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 : Verifie si le joueur 1 est sur le bonus de portée de bombe

verif_bonus_bombe1 : Verifie 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 une 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 connaissions 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 avions 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 trâce 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

Voici les aspects et les fonctions qui m'ont été confiés.

Premièrement lorsque l'on démarre le jeu, on tombe sur un menu, avec 2 items sur lesquels on peut cliquer :

— nouvelle partie permet de commencer une partie
— quitter permet de quitter en détruisant la fenêtre

Une fois la partie lancée, la fonction startgame est appelée :

— elle crée la fenêtre de jeu dont la taille est de 550pixels sur 550pixels. C'est donc dans ce Canvas que se déroule le jeu
— elle crée également le canvas situé en bas, contenant les barres de vies, le nombre de bombes à disposition des joueurs ainsi que la portée de leurs bombes

Ensuite la fonction personnage crée les personnages. Nous avons donc 2 personnages :

— le joueur 1 (en rouge) est situé en haut à gauche de la fenêtre de jeu, aux coordonnées xj,yj=0,0
— le joueur 2 (en bleu) est lui situé en bas à droite, aux coordonnées xj2,yj2=500,500

Les deux joueurs sont des images au format PNG (toutes les images ont le même format) Ces personnages doivent bien évidemment être capables de se déplacer. Pour cela, j'ai codé les fonctions permettant le déplacement dans 4 directions (haut,bas,gauche,droite),étant donné que nous sommes dans un jeu en 2D. Elles fonctionnent toutes de la même façon :

— elles utilisent les coordonnées des joueurs (xj,yj,pour le joueur 1 et xj2,yj2, pour le joueur2)
— elles vérifient pour chaque élément se trouvant dans la direction vers laquelle le joueur veut se déplacer si l'élément est une brique ou un bloc :
  — si oui on empêche le mouvement
  — si non, le déplacement a lieu

Pour appeler les fonctions de déplacement, j'ai lié les différentes directions à des touches du clavier grâce à la méthode .bind que j'ai appliqué au canvas can, qui est l'espace où le jeu se déroule :

— Joueur 1 :

    - haut : flèche du haut
    - bas : flèche du bas
    - gauche : flèche de gauche
    - droite : flèche de droite
- Joueur 2 :
    - haut : Z
    - bas : S
    - gauche : Q
    - droite : D

Les déplacements ont lieu au moment où les touches sont relâchées grâce à l'option « KeyRelease- » située avant la touche.

La fonction enleve_vie permet de gérer la vie des deux personnages. La fonction reçoit en paramètre un joueur, pour déterminer à quel joueur il faut enlever un point de vie :

- si le joueur reçoit des dégâts de bombe alors il perd un point de vie (représenté par les cœurs dans la barre de vie) et le cœur est remplacer par un carré blanc
- si le joueur tombe à 0 point de vie, alors le joueur gagnant est affiché à l'écran et on fait appel à la fonction perdu

Elle permet également de faire apparaître un écran « Game over » lorsqu'un des deux personnages n'a plus de vie :

- elle crée 3 images :jeufini, rejouer,et quitter,qui sont des items rendus cliquables en appelant la fonction perdu qui permet de détecter le clic gauche de la souris, grâce au paramètre event, et aux variables, event.x et event.y.

Enfin, nous avons donc la fonction perdu :

- 2 choix s'offrent à nous :quitter, ce qui ferme le jeu en détruisant la fenêtre, ou nous pouvons rejouer, ce qui réinitialise toutes les variables telles que la vie des personnages, leur nombre de bombes ainsi que la portée de l'explosion des bombes, et qui appelle la fonction startgame pour recréer les éléments.

Durant ce projet, j'ai également pris en charge la réalisation des graphismes. Pour créer les différents éléments je me suis servi du logiciel paint.net

2.5 Conclusion

Ce projet d'ISN a été pour moi une expérience enrichissante. En effet ce n'est que la deuxième fois que je suis amené à travailler en groupe : la première fois était en 1ère, lors des TPE. C'est un travail, qui certes demande du temps ainsi qu'un investissement personnel important, car il s'ajoute aux autres matières, mais c'est également une tâche passionnante car elle nous permet d'avoir une approche du travail en groupe, qui est je l'imagine, souvent sollicité dans le domaine de l'informatique car il permet une plus grande efficacité.

Pour ce qui est de l'ambiance du groupe, je dirais qu'elle est agréable et propice au travail. Nous avons su faire preuve de coordination, en répartissant le travail équitablement, ce qui nous a permis de travailler efficacement, sans nous disperser et ainsi sans perdre de temps. Personnellement je pense avoir travaillé plus efficacement chez moi, plutôt qu'en cours d'ISN, sans doute car je n'arrivais pas à me concentrer dans le bruit, mais aussi car j'ai la possibilité de me renseigner directement sur internet en cas de difficulté, en étant chez moi, ce qui est moins évident en classe.

De plus, ce projet en groupe, m'a donné l'envi d'enrichir mes connaissances en programmation, en apprenant d'autres langages informatiques, et pourquoi pas en programmant d'autres petits jeux ou applications.