

Cahier des charges

Projet de Bomberman

par CUZIN Kévin (p1909458), NGUYEN Cécilia (p1908025), VO Anh-Kiet (p1907921)

*LIFAP4, Licence d'Informatique, 2^{ème} année
Année 2020-2021, Université Claude Bernard Lyon 1*

Chapitre 1 : Présentation du projet

Dans le cadre d'un enseignement universitaire, Conception et Développement d'Applications (LIFAP4), à l'Université Claude Bernard Lyon 1, il nous est demandé de réaliser un projet à moyen terme dont le sujet est libre.

Nous choisissons donc de concevoir un jeu vidéo sur le modèle du *Bombberman*.

Le cahier des charges est composé :

- d'une description détaillée du produit
- du déroulement du projet
- d'un diagramme de Gantt qui nous permettra de répartir les tâches sur le long terme
- d'un diagramme des classes qui nous donne plus en détails ce qu'il faudra coder

Notre objectif principal est de développer une première version avec un terrain, deux personnages et des bombes fonctionnels. Puis nous tenterons une version où il est possible de jouer seul avec un ennemi contrôlé par une intelligence artificielle.

L'ensemble de ces documents sera maintenu à jour pendant toute la durée de développement du produit.

Chapitre 2 : Description du produit

Principe et règles du jeu

Le jeu est une reproduction du célèbre *Bombberman* et de ses dérivés que l'on peut voir sur les sites de jeux flash.

Vous incarnez un personnage qui est capable de poser des bombes.

Le but du jeu est de faire exploser l'autre adversaire à l'aide des bombes. Si le joueur se fait toucher par une bombe, il meurt et perd la partie.

Le plateau de jeu sera en 2D, constitué de plusieurs carrés qui font office d'obstacle. Le joueur devra les faire exploser au fur et à mesure afin d'obtenir des bonus tels que :

- Des bombes avec des effets différents
- Augmentation de la vitesse du personnage
- Augmentation de la portée des bombes
- Nombre maximum de bombes qu'on peut poser sur le plateau
- ...

Il y aura la possibilité de jouer à deux sur le même écran. Le joueur 1 utilisera les touches directionnelles du clavier. Le joueur 2 utilisera les touches "ZQSD" (si c'est un clavier AZERTY).

Détail de l'interface

Voici ci-contre un exemple de plateau (screen du jeu Bomberman) :



Chapitre 3 : Contraintes

- Le projet sera développé en C++ sous Linux et Windows ;
- Le code sera propre et compréhensible : code indenté, variables significatives... ;
- La bibliothèque SDL2 sera utilisée pour le projet ;
- Le code du projet sera archivé et géré sur le GitLab de l'université Claude Bernard Lyon 1 ;
- Un README.md sera disponible pour expliquer comment jouer ;
- Le code devra être compilé avec un Makefile sous Linux et Windows ;
- Le débogueur GDB et le profiler Valgrind seront utilisés ;
- Le code du projet sera documenté sous Doxygen ;
- Une version texte devra être terminée avant le 30 mars 2021 ;
- Le projet devra être réalisé entre le 1er mars 2021 et le 3 mai 2021, date limite du rendu du projet.

Chapitre 4 : Déroulement du projet

Tâche 0 : Rédaction du cahier des charges

Membres impliqués : tous.

Durée : 1 semaine, rendu final le 11 mars 2021.

Tâche 1 : Définitions du diagramme des classes

Membres impliqués : tous.

Durée : 1 semaine.

Tâche 2 : Développement du mode texte

Durée : 1 mois.

Tâche 2.1 : écriture et test du module Couleur (Couleur.h / Couleur.cpp)

Membres impliqués : Anh-Kiet.

La Couleur est définie par 3 composantes : rouge, vert et bleu.

Tâche 2.2 : écriture et test du module Personnage (Personnage.h / Personnage.cpp)

Membres impliqués : Anh-Kiet.

Le Personnage est défini par son nom, sa couleur, sa position en X et en Y, s'il est vivant ou non, sa vitesse et ses bombes.

Il pourra se déplacer et placer des bombes sur le terrain pour détruire les briques qui l'entourent.

Tâche 2.3 : écriture et test du module Brique (Brique.h / Brique.cpp)

Membres impliqués : Kévin.

Une Brique est définie par sa position en X et Y, par sa Couleur, si elle est détruite ou non, si elle possède un bonus, si oui on aura son type de bonus.

Une Brique pourra être détruite par une Bombe d'un Personnage et potentiellement contenir un bonus.

Tâche 2.4 : écriture et test du module TabBrique (TabBrique.h / TabBrique.cpp)

Membres impliqués : Kévin.

Le tableau de briques contient toutes les briques du terrain avec leurs propriétés et le nombre total de briques.

Des tests pourront être faits sur les briques pour vérifier si le code du module Brique n'a pas de problème.

Tâche 2.5 : écriture et test du module Bombe (Bombe.h / Bombe.cpp)

Membres impliqués : Cécilia.

Une bombe est définie par une position en X et Y et par sa portée d'explosion.

Tâche 2.6 : écriture et test du module TabBombe (TabBombe.h / TabBombe.cpp)

Membres impliqués : Cécilia.

Le tableau de bombes contient toutes les bombes du jeu avec leurs propriétés et le nombre total de bombes.

Des tests pourront être faits sur les bombes pour vérifier si le code du module Bombe n'a pas de problème.

Tâche 2.7 : écriture et test du module Terrain (Terrain.h / Terrain.cpp)

Membres impliqués : tous.

Le terrain possède des dimensions en X et en Y et un tableau 2D de caractères pour afficher le terrain en version texte.

Tâche 2.8 : écriture et test du module Jeu (`Jeu.h` / `Jeu.cpp`)

Membres impliqués : tous.

Le module Jeu est décrit par un Terrain, un Personnage et un tableau de Brique.

Une fonction de ce module va permettre au personnage (module `Personnage`) de se déplacer sur le terrain (module `Terrain`) quand une touche sera pressée. C'est également grâce à ce module que l'on va tester si tous nos autres modules marchent.

Tâche 2.9 : écriture du module JeuModeTexte (`JeuModeTexte.h` / `JeuModeTexte.cpp`)

Membres impliqués : tous.

Affichage au format texte de toutes les données du Terrain, Personnage et Brique.

On aura un tableau en 2D de caractères pour représenter les briques du jeu tout autour de la fenêtre, une lettre pour le personnage et pour les bombes.

Tâche 3 : Développement d'une 2ème version en mode graphique sous SDL2

Durée : 1 mois.

Tâche 3.1 : temps de réflexion pour savoir par où commencer au niveau du code.

Membres impliqués : tous.

Tâche 3.2 : écriture du module JeuModeGraphique (`JeuModeGraphique.h` / `JeuModeGraphique.cpp`)

Membres impliqués : tous.

Codage des procédures init (pour la fenêtre SDL2), boucle (pour les événements du clavier) et de destruction d'une fenêtre.

Tâche 3.3 : modification du diagramme des classes et entraînement présentation

Membres impliqués : tous.

Tâche 3.4 : test et debug

Membres impliqués : tous.

Debug avec gdb sur le jeu et la fuite de mémoire avec Valgrind.

Chapitre 5 : Diagramme de Gantt

Légende : K (Kévin), A (Anh-Kiet), C (Cécilia), X (tous)

	Description	01/03/21	08/03/21	15/03/21	22/03/21	29/03/21	05/03/21	12/03/21	19/03/21
Tâche 0	Cahier des charges	X	X						
Tâche 1	Diagramme des classes	X	X						
Tâche 2.1	Module Couleur		A						
Tâche 2.2	Module Personnage		A						
Tâche 2.3	Module Brique		K	K					
Tâche 2.4	Module TabBrique			K					
Tâche 2.5	Module Bombe		C	C					
Tâche 2.6	Module TabBombe			C					
Tâche 2.7	Module Terrain			X	X				
Tâche 2.8	Module Jeu			X	X	X			
Tâche 2.9	Module JeuModeTexte			X	X	X			
Tâche 3.1	Temps de réflexion					X			
Tâche 3.2	Module JeuModeGraphique					X	X	X	X
Tâche 3.3	Modification diagramme des classes et entraînement présentation							X	X
Tâche 3.4	Test et debug	X	X	X	X	X	X	X	X

	Description	26/03/21	03/04/21
Tâche 0	Cahier des charges	X	X
...			
Tâche 3.2	Module JeuModeGraphique	X	X
Tâche 3.3	Modification diagramme des classes et entraînement présentation	X	X
Tâche 3.4	Test et debug	X	X

Chapitre 6 : Diagramme des classes

