

RAPPORT DE SPRINT

JAMES BOMB

Alves Brandon
Duarte Vincent
Tran Thien My, Elsa



<u>1. INTRODUCTION</u>	<u>2</u>
1.1) OBJECTIFS	2
1.2) VUE D'ENSEMBLE DU PRODUIT	2
1.3) DÉFINITIONS, ACRONYMES ET ABRÉVIATIONS	3
1.4) DOCUMENTS DE RÉFÉRENCES	3
1.5) APERÇU DU DOCUMENT	3
<u>2. DESCRIPTION GÉNÉRALE DU LOGICIEL</u>	<u>3</u>
2.1) VUE D'ENSEMBLE DES FONCTIONS DU PRODUIT	3
2.2) CONTENU DES SPRINTS	5
<u>3. CONTENU DU SPRINT ACTUEL</u>	<u>6</u>
3.1) CAS D'UTILISATION IMPLANTÉS DANS LE SPRINT	6
Améliorations de ce qui était livré avant	6
Use Case Implantés	7
Autre "Fonctionnalités" implantées	7
3.2) Modèle de classe	8
Sprint 1	8
Sprint 3	8
Explications	9
4. Revue	9
4.1) REVUE DU SPRINT	9
Revue technique	9
Revue de gestion	10
4.2) REVUE DU PROJET	11
Revue technique	11
Revue de gestion	12
Développement Agile	12
5. Annexes	12

1. INTRODUCTION

1.1) OBJECTIFS

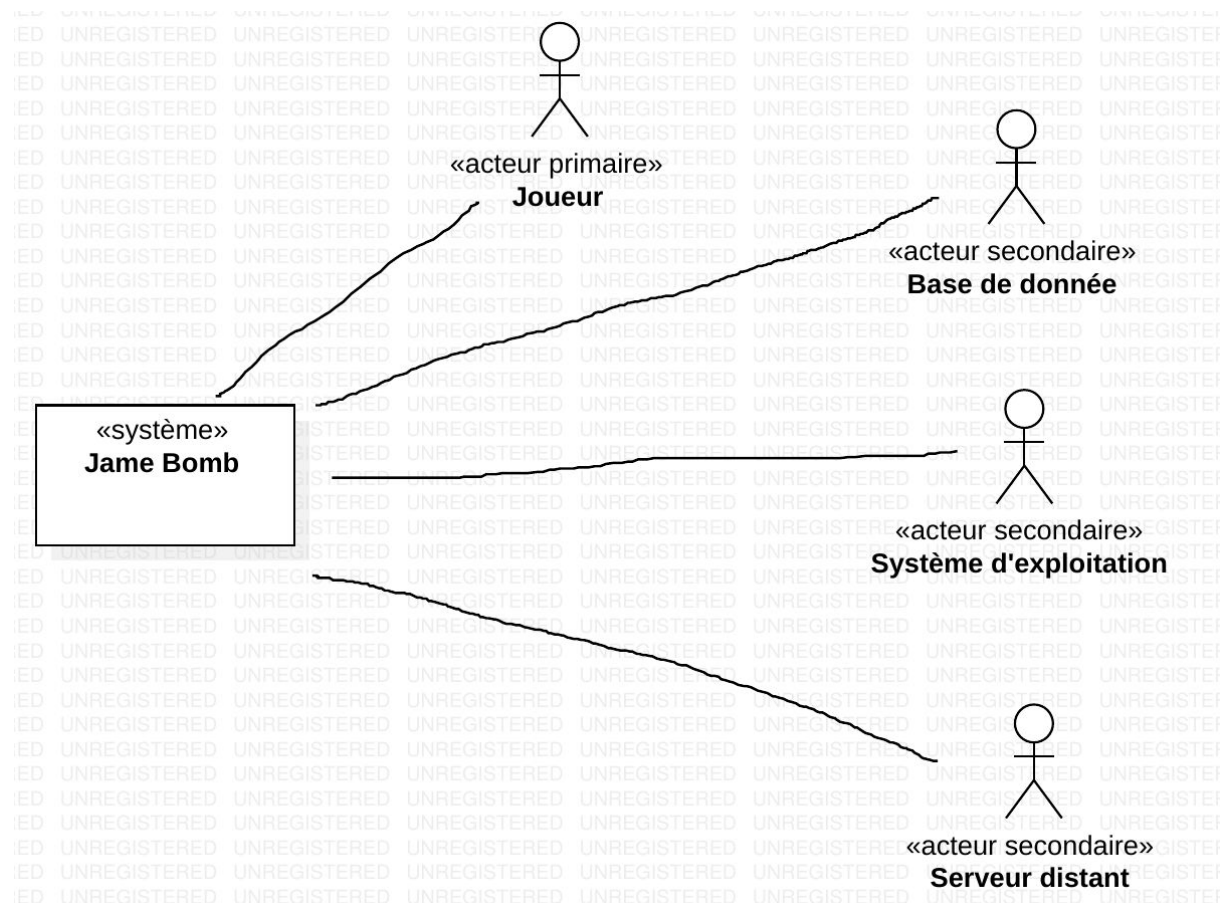
L'objectif de ce document est de décrire ce qui a été fait dans le second/troisième sprint aux commanditaires, comment c'est déroulé le travail d'équipe, les obstacles et difficultés rencontrés et finalement établir un état des lieux du projet dans sa globalité.

1.2) VUE D'ENSEMBLE DU PRODUIT

“James Bomb” est une application de bureau pour :

- Procurer du plaisir et du divertissement aux utilisateurs
- Créer de la compétitivité entre les utilisateurs
- Développer de la coopérativité entre les utilisateurs
- Rendre le jeu plus accessible (monétaire et localité)
- Attirer de nouveaux utilisateurs en gardant les utilisateurs actuels

Les parties prenantes de notre système sont:



Utilisateur : Les utilisateurs principaux du jeu sont les personnes qui jouent à “James Bomb”.

Base de donnée : La base de donnée interagit avec le système afin de transmettre des données. Le système communique avec la base de donnée afin de sauvegarder de l'information.

Système d'exploitation: Un système d'exploitation est un ensemble de programmes permettant de faire fonctionner et contrôler un appareil informatique. Puisque le jeu est une application de bureau, il est dépendant du système d'exploitation de l'ordinateur sur lequel il sera installer. Ici James Bomb est implémenté en python ce qui le rend opérationnel sur la plupart des systèmes d'opération.

Serveur distant: Le serveur distant communique avec le système afin de fournir le service du jeu en ligne et deux permettre la liaison entre deux acteurs principales "Joueur".

1.3) DÉFINITIONS, ACRONYMES ET ABRÉVIATIONS

1.4) DOCUMENTS DE RÉFÉRENCES

Documentation en lien avec le système actuel :

Wikipédia - Bomberman (<https://en.wikipedia.org/wiki/Bomberman>)

Menu (<https://github.com/ppizarror/pygame-menu#configuration-values>)

YouTube - Gameplay du mode aventure (<https://www.youtube.com/watch?v=rdp1rY8U-f0&t=1935s>)

Sprites

(<https://gamedevelopment.tutsplus.com/articles/enjoy-these-totally-free-bomberman-inspired-sprites--game-dev-8541>)

Musique/sons (https://freesound.org/people/james%20bond%20007/downloaded_sounds/)

1.5) APERÇU DU DOCUMENT

Le document contient la description du projet général et plus particulièrement celle du sprint 2-3. On y développe les fonctionnalités implémentés durant ce sprint. Puis on passe en revue les aspect techniques et de gestion liés à ce sprint. Finalement on synthétise ce qui a été réalisé ou non, comment c'est déroulé l'élaboration du système, les points qui ont portés satisfaction, les points à améliorer, etc... On conclut en parlant de notre expérience vis à vis du développement agile.

2. DESCRIPTION GÉNÉRALE DU LOGICIEL

2.1) VUE D'ENSEMBLE DES FONCTIONS DU PRODUIT

Le jeu comporte deux modes d'opération (Local et En Ligne) qui offrent les mêmes modes de jeux. Le nombre de joueurs maximal est de 2 en local et de 4 en ligne. En ligne, l'utilisateur peut jouer avec d'autres joueurs à distance, on rassemble aléatoirement 4 joueurs et crée une partie. Il peut lui même créé une partie privé et invité des amis ou rejoindre une partie déjà créé par un ami.

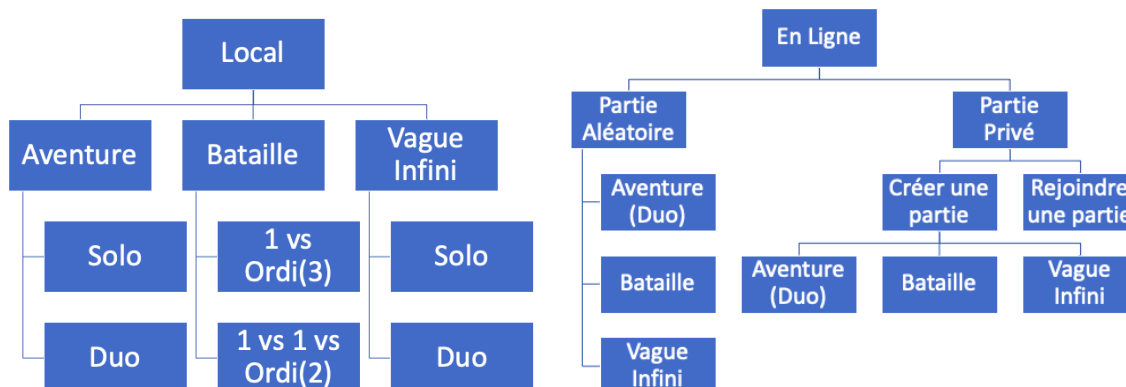


Figure 2 - Schéma des modes de jeu et leurs sous-modes

Pour tous les modes de jeux, plusieurs obstacles bloquent le passage sur l'ensemble du terrain et les joueurs ont la possibilité de les détruire avec leur bombe pour atteindre leur objectifs. Sachant que la destruction d'un obstacle a une chance de faire apparaître un objet bonus (les mêmes que le système actuel) pour augmenter les compétences du personnage. Les modes de jeux sont disponible en local et en ligne avec un nombre de joueurs variable.

Mode Aventure : Ce mode ressemble au mode aventure du système actuel. Le joueur (solo) ou les joueurs (duo local ou en ligne) doivent tuer tous les ennemis présent sur le terrain et trouver la sortie de tableau qui le mènera au tableau suivant. En duo, les joueurs jouent en coopération. Plus on avance dans l'aventure, plus les tableaux deviennent difficile. Les niveaux sont composé chacun de 10 tableaux et chaque fin de niveau comporte un boss à l'issue duquel les joueurs gagnent un accessoire décoratif pour leur personnage.

Mode Bataille : Ce mode ressemble au mode bataille du système actuel. Le joueur doit éliminer les 3 autres joueurs ou ordi avant la fin du temps imparti. Lorsqu'il reste peu de temps au chronomètre, le terrain se rétrécit vers le centre, les personnages sont éliminés s'ils ne se rabattent pas assez vite. En ligne, si le joueur gagne la partie, il gagne aussi un accessoire décoratif pour son personnage. Cf. figure 3.

Mode Vague Infini : Le ou les joueurs doivent tuer le plus d'ennemi avant de mourir, ce mode est coopératif. Les ennemis apparaissent par vague, chaque vague a une difficulté plus élevé que la précédente. La difficulté des vagues est défini par le nombre et la difficulté des ennemis. Le score des joueurs est déterminé par leurs temps de survie.

En ce qui concerne les ennemis du mode aventure et les ordinateurs, ils sont plus intelligent grâce à des algorithmes d'intelligence artificiel avec différents niveaux de difficultés.

Customization des personnages : Lorsque le joueur gagne une partie dans le mode bataille multijoueur ou lorsqu'il trouve un coffre bonus dans le mode aventure, il débloquent un accessoire qui peut être équipé sur son personnage pour modifier son apparence.

Les catégories d'accessoires sont les suivantes :

- Les couvre-chefs
- Les visages/masques

- Les hauts
- Les bas
- Les souliers
- Les accessoires portés sur le dos (sac à dos, instruments, etc..)

Pour le menu d'option, le joueur peut changer les touches de commandes et la langue du jeu (Anglais ou Français). Il pourra aussi consulter ses statistiques de jeu. En partie local, le joueur peut mettre le jeu en pause à tout moment, ce qui affiche un menu pause avec les choix d'arrêter ou de reprendre la partie.. Lorsque le joueur charge une ancienne partie en mode aventure, il recommence au début du dernier tableau à partir duquel il a sauvegardé.

James Bomb est multi-plateforme grâce à Python (langage de programmation) qui est très en vogue en ce moment et installable sur les principaux systèmes d'exploitation.

En mode bataille et vague infini, la taille des terrains sont variables et aléatoire puisqu'ils sont générés par un algorithme.

Notre système utilisera 3 principales librairies :

- [pygame](#) (principalement utilisé pour l'affichage et la gestion des évènements)
- [pygame-menu](#) (utilisé pour la création simple de menus)
- [mysql.connector](#) (utilisé pour permettre à python de communiquer avec la base de données)

2.2) CONTENU DES SPRINTS

Pour le premier sprint nous avons prévu d'implémenter :

- le mode aventure, c'est à dire d'implémenter :
 - la création des labyrinthes,
 - le déplacement du personnage,
 - la gestion des bombes et leur explosion,
 - la gestion des collisions,
 - la gestion des objets bonus,
 - la création d'une histoire,
 - la création et la gestion des ennemis,
 - la création et la gestion des boss,
 - la vie du personnage,
- le menu

Pour le second sprint, nous avons prévu d'implémenter :

- la sauvegarde des parties et le chargement de celles ci,
- la personnalisation du personnage avec des accessoires afin de modifier son apparence

Pour le troisième sprint, nous avons prévu d'implémenter :

- les parties multijoueurs,
- la personnalisation des commandes de control du jeu

Après le sprint 1 nous avons décidé d'implémenter pour le sprint 2 :

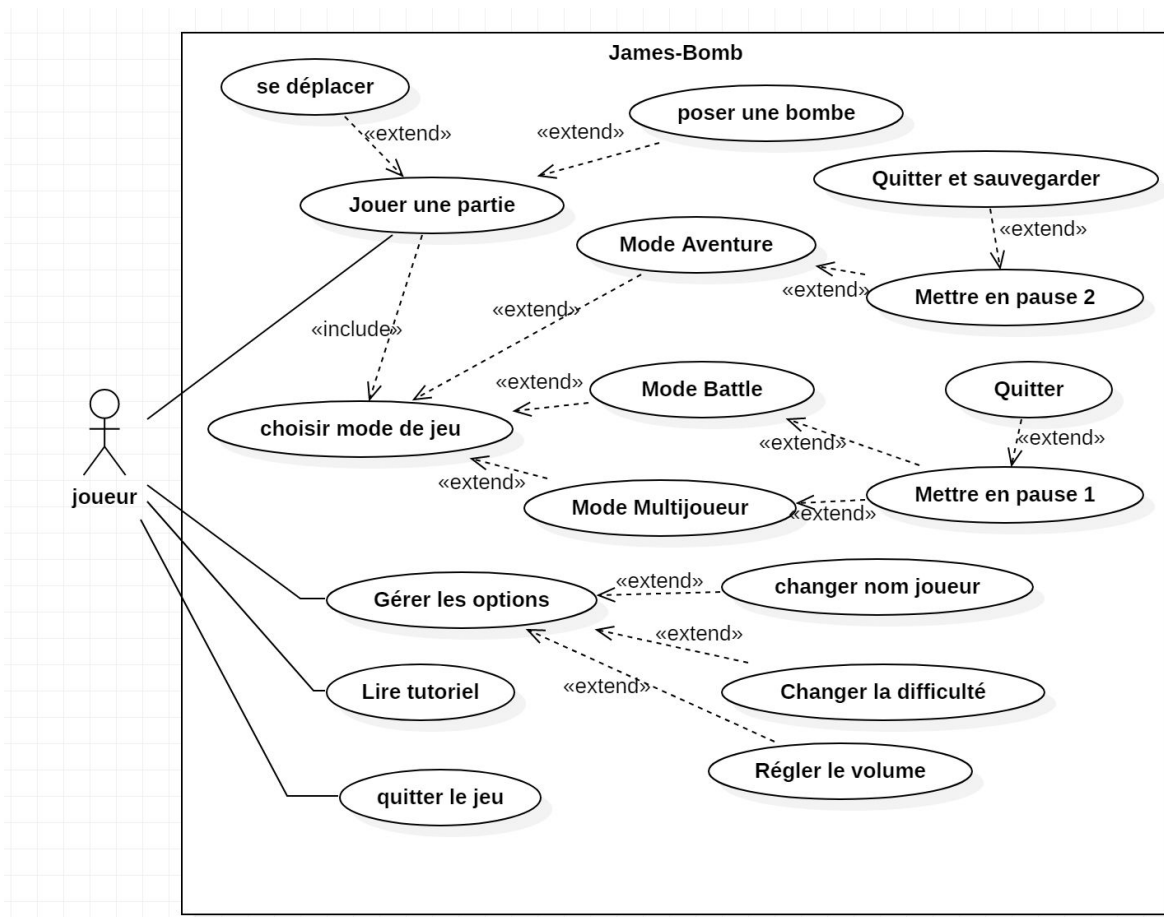
- les bonus,
- les parties multijoueurs,
- personnalisation de personnage,

- un boss final,
- la vie du personnage,
- sauvegarde et chargement des parties

Car nous n'avions pas terminé la gestion des bonus et que nous avons pensé que le mode de jeu aventure pourrait rajouter une plus value à notre système, retardant ainsi la personnalisation des personnage (que nous n'aurons finalement pas implémenté). De plus nous avons décidé d'évacuer la personnalisation des commandes de control du jeu estimant que cette fonctionnalité n'était pas essentielle pour le moment pour faire valoir notre produit.

3. CONTENU DU SPRINT ACTUEL

3.1) CAS D'UTILISATION IMPLANTÉS DANS LE SPRINT



AMÉLIORATIONS DE CE QUI ÉTAIT LIVRÉ AVANT

Comme dans le précédent Diagramme de Use-Cases, on peut retrouver les cas d'utilisations :

- Jouer une partie

Ce Use-Case en soi ne change pas totalement du premier sprint. C'est la fonctionnalité global qui a été amélioré et qui sera détaillé plus bas.

- Poser une bombe

À la différence du premier sprint, la gestion des bombes est maintenant mieux gérée. Vis-à-vis de leur portées, de leur quantité, et ce même en fonction des joueurs qui la pose (Cf Mode Multijoueur).

- Se déplacer

Ce Use-Case ne change pas par rapport au premier sprint.

- Quitter le jeu

Ce Use-Case ne change pas par rapport au premier sprint.

USE CASE IMPLANTÉS

- Lire tutoriel

Le joueur a la possibilité d'accéder à un tutoriel écrit pour comprendre les touches claviers du jeu.

- Gérer les options

Le joueur a la possibilité de changer des options du jeu (son pseudonyme, la difficulté, le son).

- Changer nom joueur

Pour sauvegarder une partie dans le mode aventure, le joueur peut choisir d'entrer un nom s'il veut retrouver sa progression plus tard.

- Changer la difficulté

Le joueur peut choisir l'intensité de la difficulté pour les parties en cours en fonction de son aisance de au jeu (inopérant pour le moment).

- Régler le volume

Le joueur peut couper ou mettre en marche la musique présente dans le jeu.

- Choisir Mode de jeu

Comme prévu initialement dans notre ConOps, le joueur a le choix entre plusieurs modes de jeu détaillés ci-dessous.

- Mode multijoueur

Le joueur peut lancer une partie qui se joue à 1 contre 1 sur la même machine.

- Mode Battle

Le joueur peut lancer une partie où le seul but sera de sortir du labyrinthe présent dans le jeu.

- Mode Aventure

Le joueur peut lancer une partie où il sera confronté à un scénario avec un enchaînement de stage finissant sur un combat contre un boss. Si le joueur n'a pas encore commencé le mode aventure, une cinématique se présentera à lui. Sinon il la reprendra là où il s'était arrêté (le joueur est identifiable par un pseudonyme).

- Mettre en pause 1

Depuis le Mode Battle ou dans le Mode Multijoueur, on peut mettre le jeu en pause pour le reprendre ensuite ou au contraire l'arrêter, détaillé ci-dessous.

- Mettre en pause 2

Depuis le Mode Aventure, on peut mettre le jeu en pause pour le reprendre ensuite ou au contraire le sauvegarder et l'arrêter, détaillé ci-dessous.

- Quitter

L'option quitter du menu pause des Modes Battle et Multijoueur quitte tout simplement la partie et laissera la place à une nouvelle lors de la reprise du jeu.

- Quitter et sauvegarder

L'option quitter du menu pause du mode aventure permet au joueur de quitter le jeu mais en sauvegardant sa progression, ce qui lui permettra de la reprendre où il en était lors de sa prochaine session de jeu.

AUTRE "FONCTIONNALITÉS" IMPLANTÉES

- Fonctionnalité de la musique du jeu

Lors de la conception de notre logiciel, on a trouvé bonne l'idée de rajouter une musique de fond à notre jeu pour lui donner plus de vie et qu'il soit plus agréable à utiliser. Cependant on a quand même décidé de laisser au joueur le choix de la couper ou non.

- Fonctionnalité du tutoriel

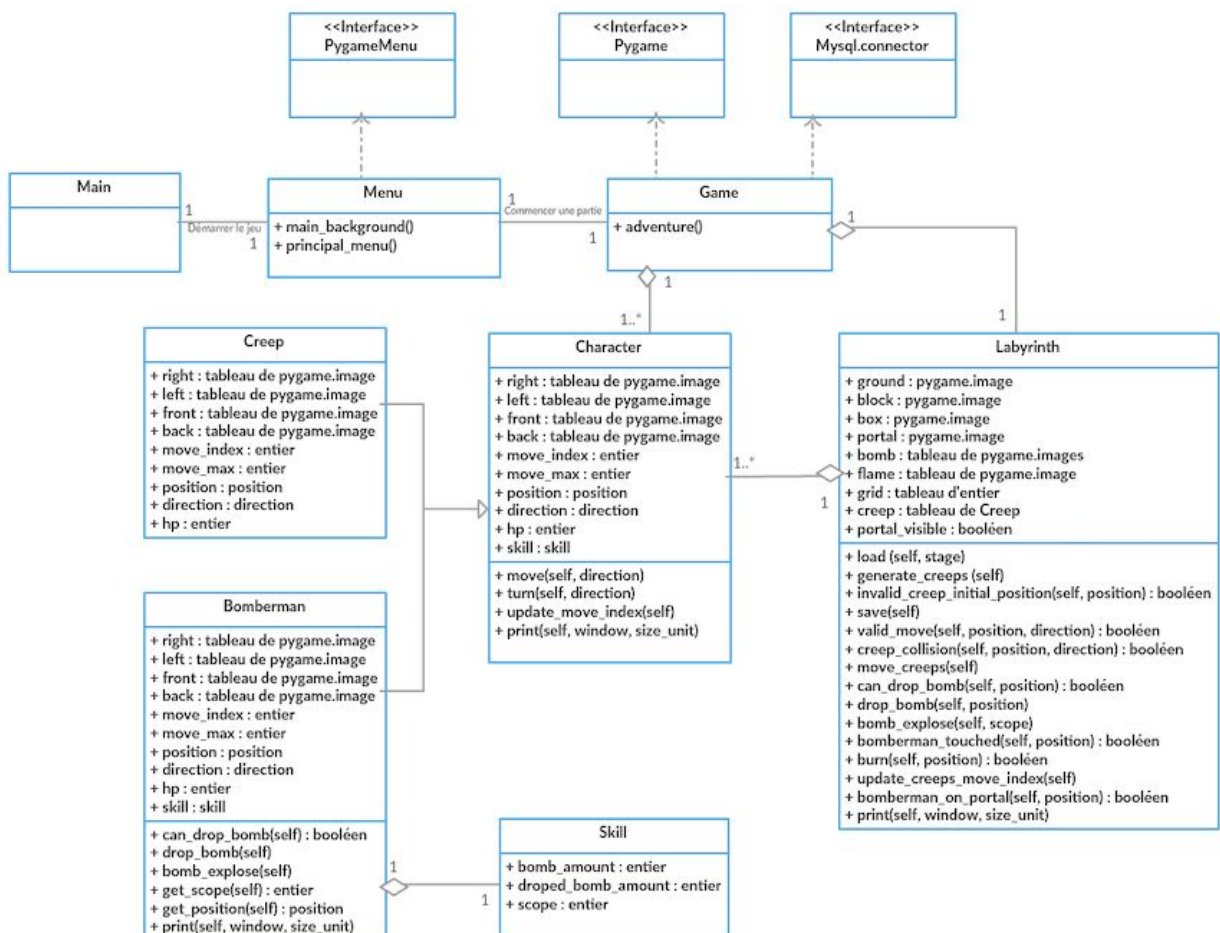
De même que pour la musique, il nous a semblé bon de mettre un didacticiel dans le jeu pour que l'utilisateur du système puisse le prendre en main plus facilement et rapidement.

- Fonctionnalité du menu pause

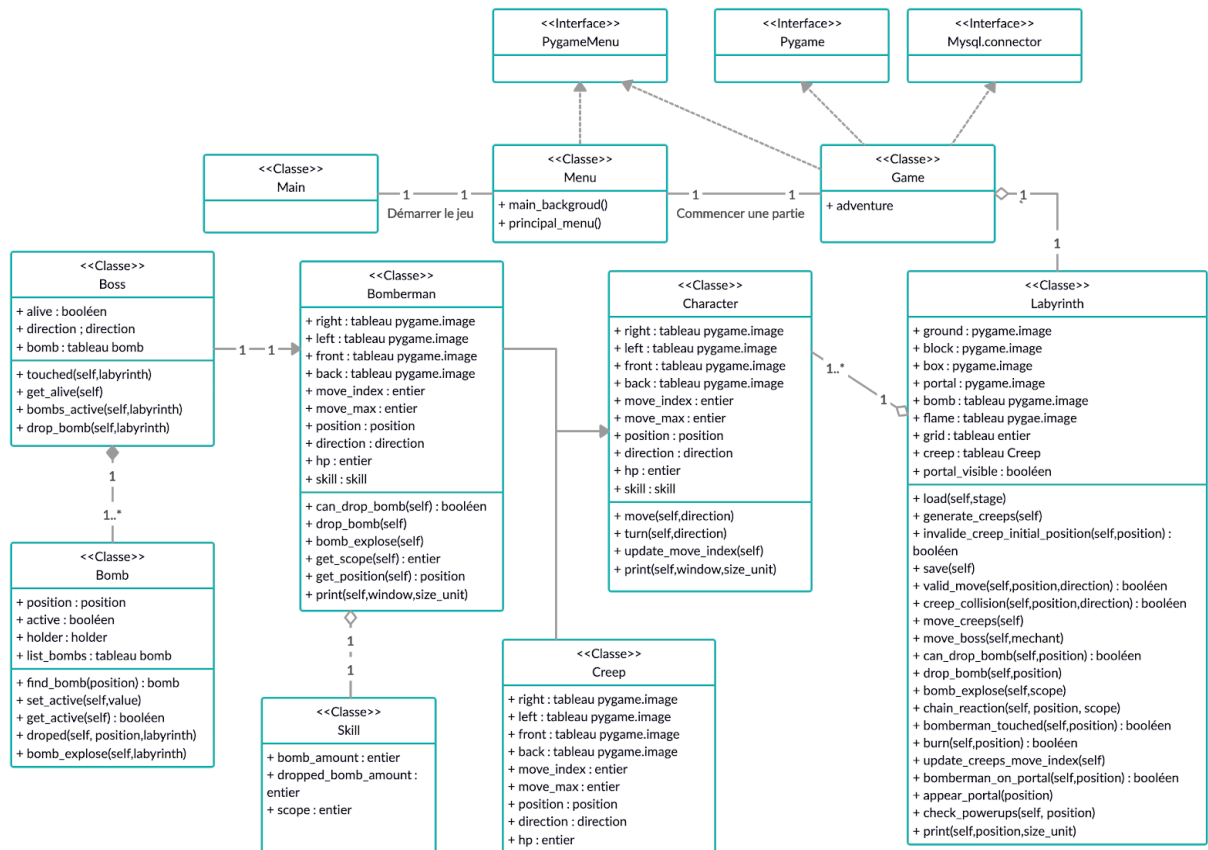
En testant notre logiciel, on s'est rendu compte que c'était plus agréable de pouvoir mettre pause pour marquer un temps mort dans notre partie, sans pour autant devoir la quitter. Et ainsi éviter de la reprendre depuis le début.

3.2) MODÈLE DE CLASSE

SPRINT 1



SPRINT 3



EXPLICATIONS

Les classes Boss et Bomb ont été ajoutées pour la gestion des boss qui peuvent poser des bombes tout comme le personnage Bomberman. Boss hérite de Bomberman.

Grâce à cette modification, on rattache les bombes à la personne qui les ont posées, ayant ainsi la possibilité de poser des bombes sur le terrain n'ayant pas les mêmes attributs comme la portée par exemple.

La classe Labyrinth a quelques fonctions supplémentaires dont :

- Move_boss qui gère le déplacement du boss,
- Chain_reaction qui fait exploser les bombes touchées par l'explosion d'une bombe adjacente

Les classes centrales de notre projet sont Game, Labyrinth et Bomberman. Labyrinth s'occupe de la gestion du labyrinthe (ennemis, différents types d'unités de terrain, explosions, bonus, etc...). Bomberman s'occupe de la gestion du personnage du joueur. Game permet à ces deux derniers de communiquer suite à des événements déclenchés par des pressions de touches du clavier ou des comptes à rebours.

Creep est une classe qui représente les ennemis qui vadrouillent au sein du labyrinthe. Ils appartiennent à un labyrinthe.

Skill est une classe qui permet de stocker les caractéristiques d'un bomberman (portée des bombes, nombre de bombes posées, portée des bombes). Un bomberman possède un skill.

Character est une classe abstraite qui permet de factoriser du code commun à Bomberman et Creep

4. REVUE

4.1) REVUE DU SPRINT

REVUE TECHNIQUE

- Tâches complétées
 - Sauvegarde des labyrinthes
 - Chargement des labyrinthes
 - Gestion du son
 - Amélioration du menu
 - Correction de bugs dans la gestion des explosions (réactions en chaînes)
 - Mode aventure
 - Mode multijoueur
 - Affichage des touches de jeux
 - Gestion des bonus
- Tâches non complétées
 - Le redimensionnement de la fenêtre (nous avons décidé de ne pas nous attarder là dessus)
 - Scénario
 - Customisation des personnages
 - IA (initialement dans le ConOps puis retiré à partir du sprint 1)
 - Jeux en ligne (initialement dans le ConOps puis retiré à partir du sprint 1)
- Ce qui a bien fonctionné
 - L'utilisation de la bibliothèque pygame (a facilité la gestion de l'affichage, et des évènements)
 - L'utilisation de la bibliothèque pygame-menu (a grandement simplifié la création d'interfaces pour les menu)
 - L'utilisation de la bibliothèque mysql-connector
 - Le choix du langage python
- Ce qui a mal fonctionné
 - Le module pygame ne permet pas à l'utilisateur de créer une quantité illimitée d'évènements (ce qui nous a un peu compliqué la tâche pour la gestion des bombes)
- Obstacles rencontrés et solutions trouvées
 - Apprentissage du langage python
 - S'organiser et se coordonner pour travailler ensemble. La solution a été de travailler avec la plateforme GitHub afin de suivre l'avancement de chacun.
 - Le confinement a eu des effets contre motivateurs. Il a été pour nous plus difficile de travailler dans cette situation.

REVUE DE GESTION

Au niveau de la cohésion/gestion de l'équipe, tout c'est bien déroulé. Il y a une bonne entente au sein de l'équipe. Nous nous sommes bien réparti les tâches. Nous avons utilisé git et GitHub pour la gestion et le partage du code source, ce qui nous a permis de suivre l'avancement de chacun, reporter les bugs, commenter et discuter les ajouts/modifications de chacun..

L'utilisation d'une base de donnée c'est bien passé. Nos structures de données nous ont permis de réutiliser notre code pour les modes de jeux aventure et multijoueur.

Pour le mode multijoueur nous nous sommes basé sur un clavier QWERTY pour les touches de jeu. Ce qui est dérangent pour les personnes utilisant un clavier AZERTY. Une idée serait de détecter le type de clavier utilisé par l'utilisateur et d'adapter les touches en fonction.

Nous avons ajouté du son (musique du menu et musique de jeu). Une amélioration possible serait d'ajouter des sons pour les différents évènements (bombe posée, bombe qui explose, obtention d'un bonus, etc...).

Pour le mode aventure nous avons commencé à préparer un scénario mais nous n'avons pas eu le temps de l'intégrer entre chaque niveaux. Ceci serait sans doute un point intéressant à ajouter.

Nous avons permis de sélectionner le niveau de difficulté dans le menu mais celui ci n'est pas considéré pour le moment. Le but serait de permettre de donner une intelligence aux ennemis dans une version future du logiciel.

On a commencé l'implémentation plutôt tard par manque de motivation et dû à certaines difficultés rencontrées vis-à-vis de la situation actuelle. Mais on a quand même réussi à bien avancé le projet et rattraper notre retard.

4.2) REVUE DU PROJET

REVUE TECHNIQUE

- Tâches complétées
 - Génération d'un terrain
 - Déplacement du personnage
 - Déplacement des ennemis
 - Poser des bombes
 - Gestions des explosions
 - Création d'un portail de sortie
 - Sauvegarde des labyrinthes
 - Chargement des labyrinthes
 - Gestion du son
 - Amélioration du menu
 - Correction de bugs dans la gestion des explosions (réactions en chaînes)
 - Mode aventure
 - Mode multijoueur
 - Affichage des touches de jeux
 - Gestion des bonus
 - Inclusion d'un scénario
- Tâches non complétées
 - Le redimensionnement de la fenêtre (nous avons décidé de ne pas nous attarder là dessus)
 - Customisation des personnages
 - IA
 - Jeux en ligne
- Ce qui a bien fonctionné
 - Le choix du langage python
 - Cf 4.1, revue technique
- Ce qui a mal fonctionné
 - Les fonctionnalités qui ne fonctionnent pas totalement se reflètent plus dans ce sprint 3.
 - Cf 4.1, revue technique
- Obstacles rencontrés et solutions trouvées
 - Cf 4.1, revue technique
 - Temps d'apprentissage afin de maîtriser le langage python
 - S'organiser et se coordonner pour travailler ensemble. La solution a été de travailler avec la plateforme GitHub afin de suivre l'avancement de chacun.
 - La difficulté de trouver du temps libre avec la quantité de cours suivi par chacun.
 - La difficulté de se synchroniser dû à la situation actuelle

REVUE DE GESTION

Notre projet se reflète principalement dans ce sprint 2-3. La revue de gestion du projet a donc été globalement traitée ci-dessus.

Cf. 4.1 : Revue de Gestion.

Ce projet nous a également permis d'apprendre un nouveau langage. De plus il nous a permis de mettre en pratique nos connaissances et d'expérimenter le développement agile. Et ainsi nous avons pu avoir une démonstration de ce qui peut se faire en entreprise.

DÉVELOPPEMENT AGILE

Nous avons trouvé le développement agile intéressant. Celui ci nous a permis de nous organiser : implémenter les fonctionnalités de manière organisée, structurée, incrémentale. Le développement agile nous a aussi permis de nous adapter plus facilement aux changements et imprévus plutôt que de suivre aveuglément un plan stricte et rigide. Par exemple nous nous étions attardé sur le redimensionnement de la fenêtre dans le sprint 1 mais nous avons finalement changer de point de vue et nous l'avons enlevé du sprint 3 en estimant que celui ci n'était pas une fonctionnalité indispensable pour le moment. Il nous a aussi permis d'obtenir un logiciel opérationnel plus rapidement afin de faire valoir notre avancement plutôt que de penser que tout se passe bien et d'arriver à la date d'échéance avec un logiciel inopérationnel. Enfin la méthode agile nous a permis de nous adapter plus facilement à la satisfaction du client (rôle que nous avons nous même joué en testant notre jeu) et de l'adapter en conséquence.

5. ANNEXES

NB: Les product backlog, sprint backlog 1 et 3 sont des feuilles dans le fichier excel BACKLOGS.