Algorithmique et programmation :

Rapport sur le projet Bomb'ISEP

Fait par :

- David **FENG (G4D)**
- Soukaina **KAMEL (G4E)**
- Kenza **KETTANI (G4C)**

Sommaire :

Introduction

L'objectif du projet de cette année est de concevoir, implémenter et tester un jeu vidéo écrit en langage Java. Il s'agira de coder un jeu de type "Bomberman" jouable par deux joueurs au clavier.

Le Bomberman que nous avons codé se compose d'un plateau de jeu de 21 x 17, composé de trois types de blocs :

- Les blocs verts qui sont les seuls blocs sur lesquels le personnage peut se déplacer,
- les blocs gris qui sont infranchissables et indestructibles,
- et les blocs orange qui sont infranchissables mais destructibles.

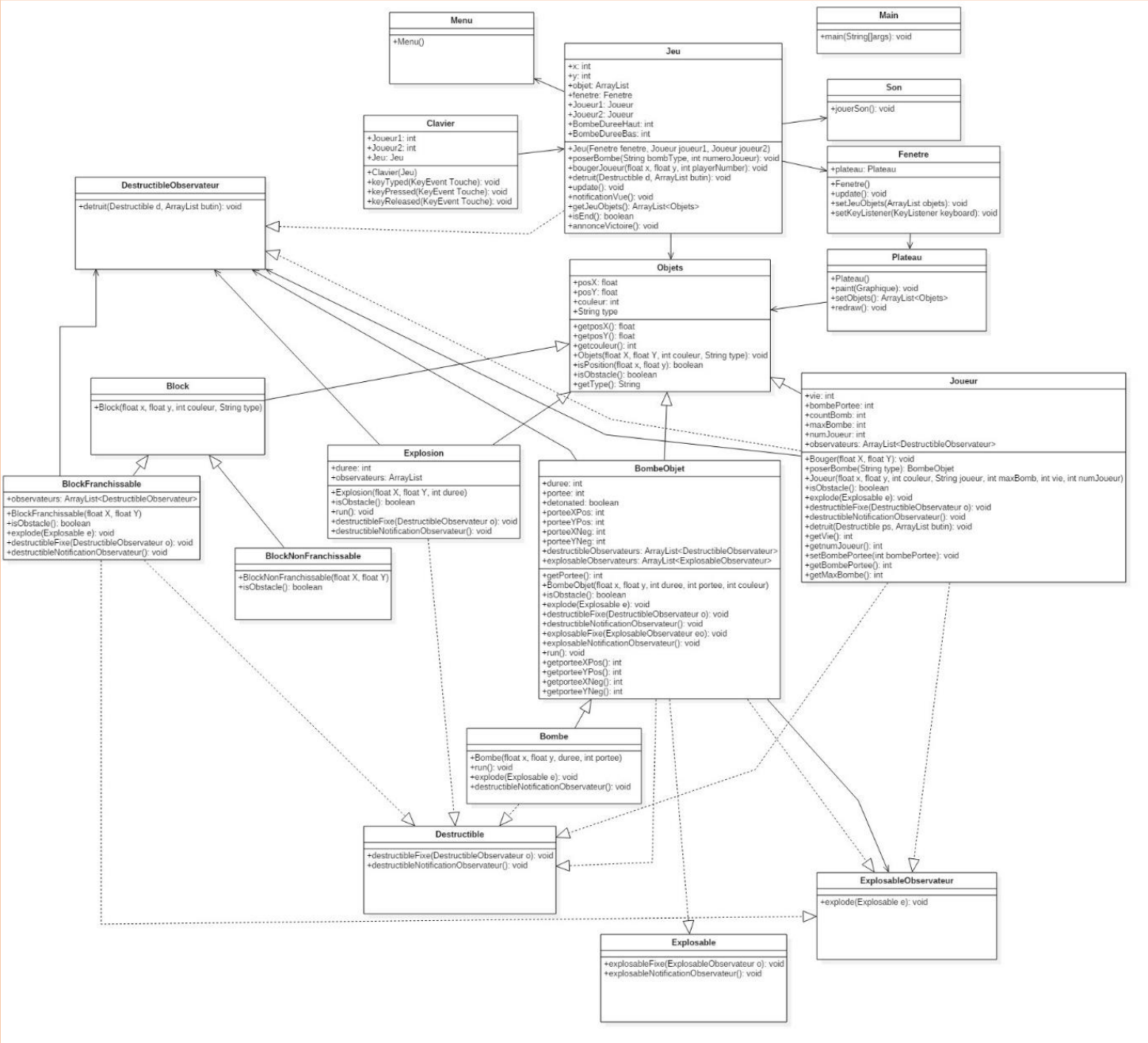
En début de partie, le Joueur 1 démarre en bas à gauche le joueur 2 en haut à droite. Notre jeu propose également tout un ensemble de bonus permettant d'améliorer les possibilités de notre Bomberman. Ces bonus se trouvent dans les blocs destructibles et apparaissent une fois ceux-ci détruits. On trouve ainsi des bonus permettant d'améliorer les bombes posées et d'autres améliorant les caractéristiques du Bomberman.

Enfin, lorsqu'un joueur se fait toucher par une explosion, il meurt et le dernier survivant gagne le jeu.

I) Conception

Vous pouvez retrouver ci-dessous le diagramme UML de notre projet. Ce diagramme UML aura encore des modifications jusqu'au jour de la présentation.

Par soucis de lisibilité, nous joindrons en plus de ce rapport, un autre fichier PDF pour que vous puissiez accéder au diagramme.



II) Fonctionnalités attendues

Le joueur :

- **Le déplacement des joueurs** : Les deux joueurs se déplacent grâce aux touches (q,s,d,z) ou aux flèches du clavier.
- **Les vies des joueurs :** Chaque joueur démarre le jeu avec trois vies. Le nombre de vies pourra être augmenté ou diminué grâce à d'autres fonctionnalités du jeu.
- **La perte de la partie :** Le joueur dont la vie arrive à zéro perd la partie.

Les bombes :

- **Poser une bombe :** Chaque joueur peut poser une bombe grâce aux touches du clavier. Une seule bombe peut être posée par case et au maximum 3 bombes peuvent être posées simultanément. Ces bombes sont des objets infranchissables.
- **Explosion de bombe :** Chaque bombe posée explose au bout de 4 secondes. Les bombes ont une portée de 3 cases (3 au-dessus, 3 en dessous, 3 à droite, 3 à gauche).
- **Destruction d'objets :** Si une explosion a lieu, celle-ci touche le premier objet destructible rencontré qui est détruit et ne va pas plus loin. En revanche si l'explosion rencontre un objet indestructible, celle-ci ne va pas plus loin et l'objet n'est pas atteint. Si une bombe est prise dans l'explosion d'une autre, elle est détruite à son tour.

Les bonus de bombes :

- **Flamme Bleue :** Diminue de 1 la portée des bombes du joueur (minimum 1).
- **Flamme Jaune :** Augmente de 1 la portée des bombes du joueur (maximum 10).
- **Flamme Rouge :** Les bombes du joueur ont une portée de 10.
- **Bombe rouge :** Les objets destructibles n'arrêtent plus la portée de l'explosion et il est donc possible de tuer un adversaire derrière un mur.

Les bonus de joueur :

- **Bonus_Vie :** Donne une vie supplémentaire au joueur qui passe dessus.
- **Speed_Up :** Augmente la vitesse de déplacement du joueur.
- **Speed_Down :** Diminue la vitesse de déplacement du joueur.
- **Bombe_Plus :** Augmente de 2 le nombre maximal de bombes posées que le joueur peut avoir (maximum 7).
- **Bombe_Moins :** Diminue de 2 le nombre maximal de bombes posées que le joueur peut avoir (minimum 2).
- **Afficher bonus :** Le rôle de cette fonctionnalité est d'afficher tous les bonus codés.

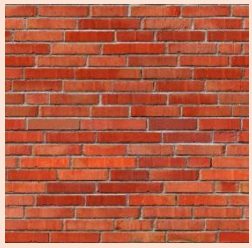
III) Fonctionnalités codées

- ✓ Tout d'abord, on a codé le plateau de 21 x 17 cases avec les trois types de blocs : verts (où les joueurs peuvent se déplacer), oranges (infranchissables mais destructibles) et gris (blocs indestructibles infranchissables).

- ✓ On a aussi codé les deux joueurs et les bombes.

✓ On a implémenté les images suivantes pour améliorer l'aspect visuel de notre jeu :

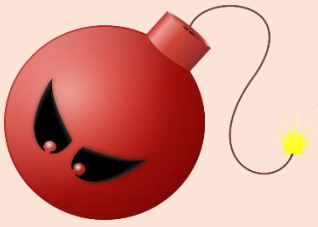
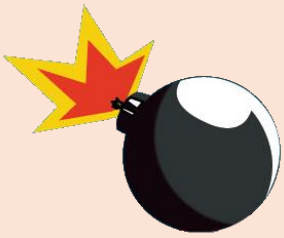
Les blocs :

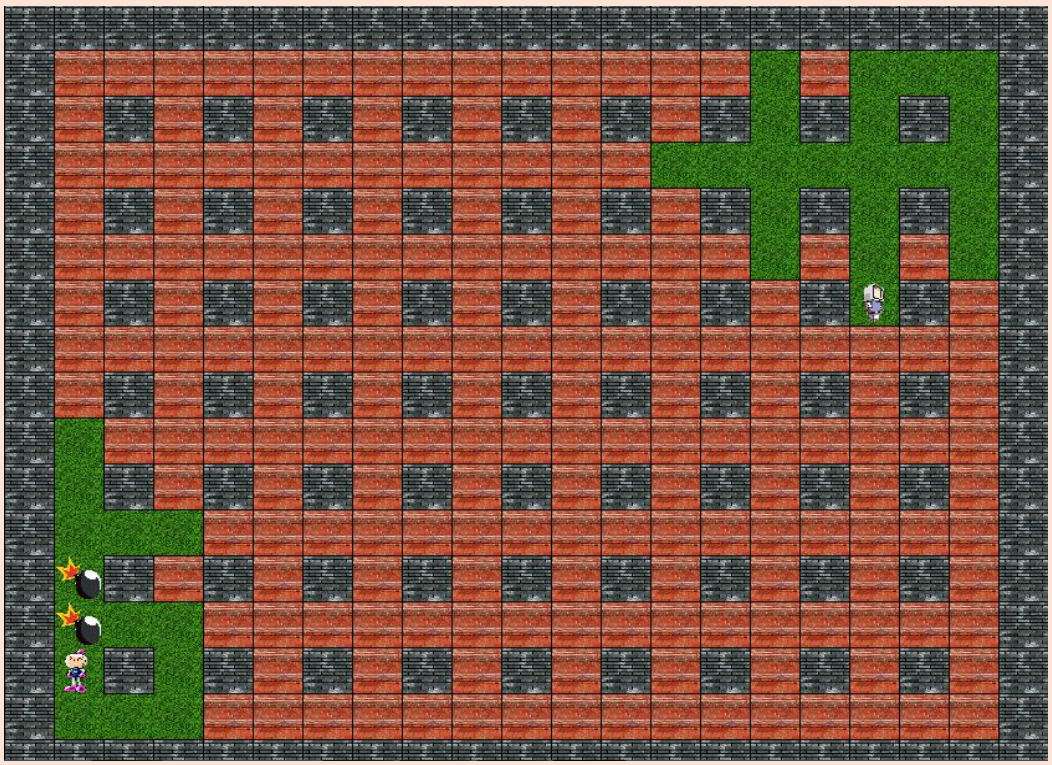

Les flammes :

Les deux joueurs :


Les bombes/détonations :

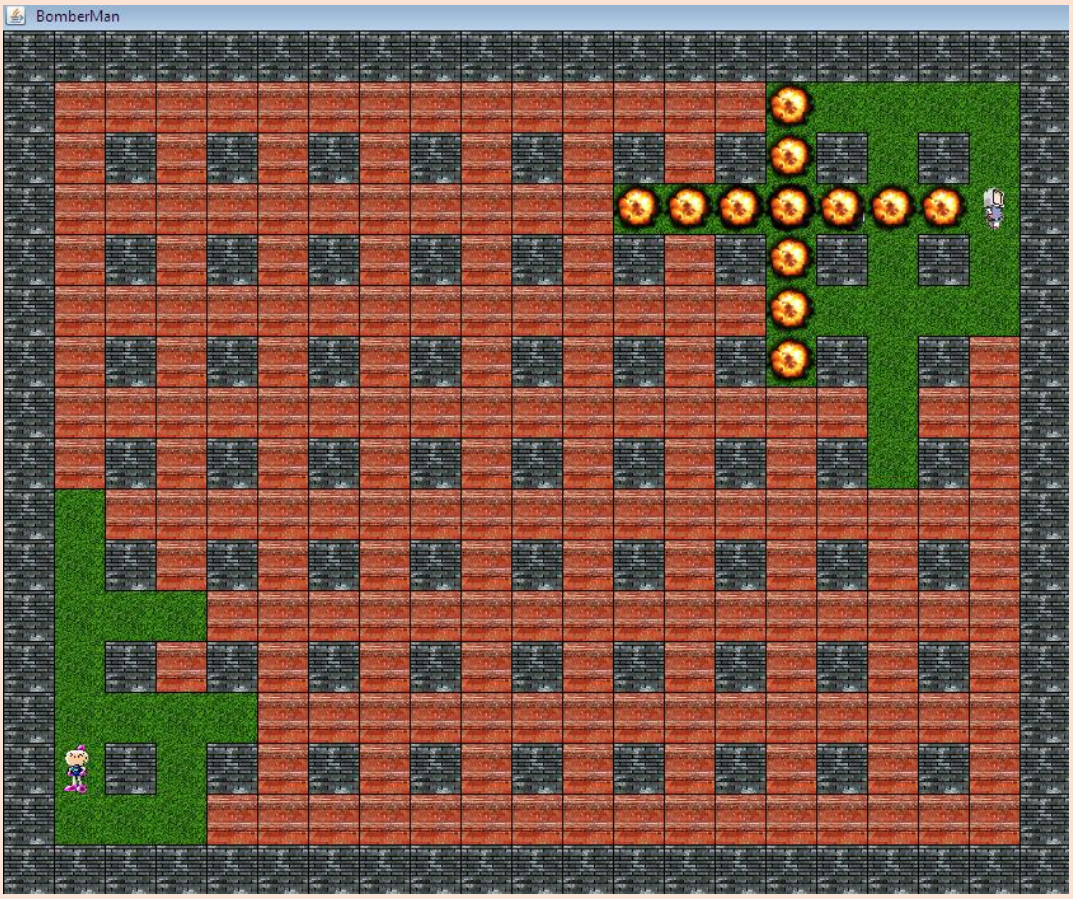

✓ On a aussi le déplacement des joueurs et la pose des bombes :



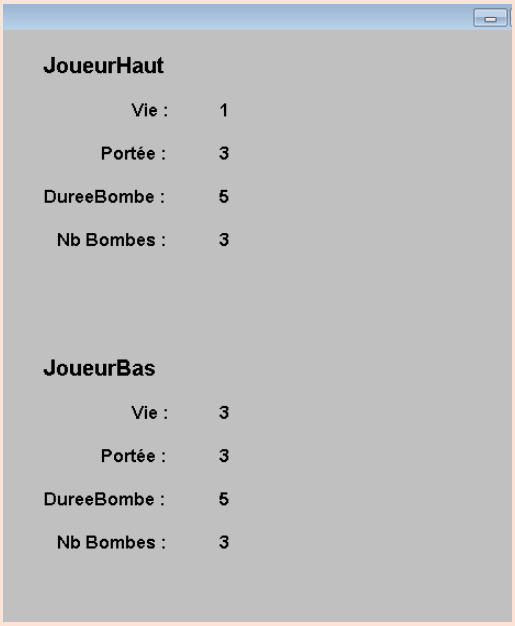
- ✓ La bombe explose bien 5 secondes après avoir été posée (confère démo)
- ✓ La bombe est également infranchissable

- ✓ La bombe possède bien avec un rayon d'explosion de 3 cases, et la bombes portée est bien limitée à une case par obstacle :

- ✓ Un bloc non franchissable bloque bien l'explosion
- ✓ Le bloc franchissable est bien détruit par l'explosion et elle n'ira pas plus loin
- ✓ Un bloc orange détruit devient franchissable par les joueurs :



- ✓ Affichage des joueurs : On peut afficher pour chaque joueur sa vie, la portée de sa bombe ainsi que la durée avant son explosion :



- ✓ Annonce du vainqueur : Cette fonctionnalité détectera lorsqu'un joueur a perdu et l'annoncera par une phrase au deux joueurs,
- ✓ On a aussi codé le son de l'explosion (confère la démo),
- ✓ Enfin, on a codé le Game Over qui annonce la fin de la partie une fois qu'un des deux joueurs meurt :

V) Fonctionnalités envisagées

Ces fonctionnalités ne sont, à ce jour, pas opérationnelles mais nous faisons de notre mieux pour qu'elles le soient le jour de la présentation.

Les bonus de bombes :

- **Flamme Bleue :** Diminue de 1 la portée des bombes du joueur (minimum 1).
- **Flamme Jaune :** Augmente de 1 la portée des bombes du joueur (maximum 10).
- **Flamme Rouge :** Les bombes du joueur ont une portée de 10.
- **Bombe rouge :** Les objets destructibles n'arrêtent plus la portée de l'explosion et il est donc possible de tuer un adversaire derrière un mur.

Les bonus de joueur :

- **Bonus_Vie :** Donne une vie supplémentaire au joueur qui passe dessus.
- **Speed_Up :** Augmente la vitesse de déplacement du joueur.
- **Speed_Down :** Diminue la vitesse de déplacement du joueur.
- **Bombe_Plus :** Augmente de 2 le nombre maximal de bombes posées que le joueur peut avoir (maximum 7).
- **Bombe_Moins :** Diminue de 2 le nombre maximal de bombes posées que le joueur peut avoir (minimum 2).

Le Son Du Jeu :

Cette fonctionnalité permettra au joueur d'avoir un son tout au long du Jeu.

Le Menu :

Nous souhaitons instaurer un menu qui donnerait au joueur le choix de joueur tout simplement grâce à un bouton « jouer ». Le joueur aura également la possibilité de choisir son avatar mais également de couper le Son du Jeu.

Affichage des vies et de la portée :

Le nombre de vies de chaque joueur ainsi que la portée de leurs bombes sont affichées sur le côté du jeu. En revanche, la diminution de ces valeurs se fait parfois de 2 en 2. Nous envisageons de corriger ce bug d'ici la présentation.

Rejouer :

Nous envisageons de mettre un bouton « rejouer » à la fin de chaque partie afin de permettre au joueur de rejouer s'il le souhaite.

L'apparition des bonus :

Les bonus, quel que soit leur type, ont 20% de chance d'apparaître après l'explosion d'une bombe.

Conclusion

En conclusion de ce projet, nous nous sommes rendu compte de l'intérêt certain de la Programmation Orientée Objet (POO) dans le codage d'un jeu vidéo. En effet, toutes les parties de notre code sont représentées sous forme d'objets qui ont chacun des attributs (ou variables) ainsi que des méthodes (fonctions).

De plus, on a combiné l'architecture MVC (Modèle Vue Contrôleur) à la POO qui a permis de mieux structurer notre code. On retiendra notamment que la POO a permis de factoriser le code (notamment grâce à l'héritage) et de le clarifier (la notion d'objets avec leurs attributs et méthodes garantit un bon découpage du code).