

## Projet : Bomberman

Ce projet doit est censé vous apporter une véritable expérience de développement d'un logiciel relativement conséquent à l'aide d'un EDI en combinant les différents outils fournis vus en cours ou en TP (contrairement aux TP qui ne sont en général qu'une succession d'exercices illustrant un nombre limité de fonctionnalités de l'EDI utilisé pour ce module).

### 1 Présentation

L'objectif du projet est d'implémenter un jeu de type Bomberman :

<http://en.wikipedia.org/wiki/Bomberman>

(dont des variantes sont aussi connues sous le nom de Dynablasters ou XBlast...)

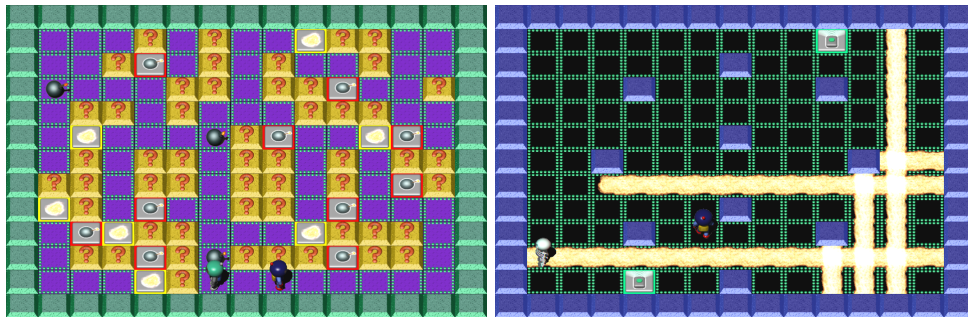


FIG. 1 – XBlast

Ce jeu fait s'affronter deux joueurs armés de bombes (pilotés par des humains ou non) dans une « arène » décomposée en  $(15 \times 13)$  blocs. Chacun de ses blocs peut être soit libre soit plein. Les joueurs ne peuvent se déplacer et déposer des bombes que sur les blocs libres. Certains blocs pleins sont toutefois destructibles à l'aide des bombes (et une fois détruit, deviennent des blocs libres en révélant parfois un bonus caché). Le but du jeu est (bien sûr...) d'éliminer l'adversaire.

### 2 Code existant

À l'adresse suivante :

<http://www.di.ens.fr/~labatut/ED6/projet/Bman.zip>

se trouve une implémentation de la partie graphique, animation et interaction du jeu.

### 3 Ce qui est demandé

Une fois familiarisé avec le code existant, il est demandé :

- de compléter le code existant pour en faire un jeu fonctionnel à deux joueurs humains,
- d'initialiser correctement de manière aléatoire la carte (en plaçant les blocs destructibles et les bonus cachés),
- d'implémenter un algorithme de recherche de chemin optimal dans le graphe de la carte,
- de proposer et d'implémenter un comportement pour le joueur robot.

### 3.1 Initialisation aléatoire des niveaux

La carte sur laquelle s'affrontent les deux adversaires doit être initialisée de manière aléatoire.

Il faudra cependant respecter les contraintes suivantes :

- les blocs bordant la carte sont toujours solides et indestructibles,
- les joueurs ne sont pas isolés l'un de l'autre par des blocs solides indestructibles,
- les positions initiales des joueurs leur permettent de placer des bombes sans avoir à se suicider,
- aucun des joueurs n'a immédiatement accès à des bonus lui donnant un avantage trop important.

### 3.2 Jeu basique à deux joueurs humains

Le code fourni ne fait qu'implémenter l'animation des personnages, il doit être légèrement complété pour en faire un jeu à deux joueurs humains. La première étape consiste à mettre à jour la carte avec la position courante des joueurs (et des bombes). Il faut aussi rajouter l'explosion des bombes, vérifier si l'un des joueurs est à portée ou non de l'explosion et le cas échéant, signaler la victoire d'un des deux joueurs (en utilisant par exemple les images grand format des personnages et les animations pour le perdant et le vainqueur).

D'autre part, les bonus de type *range* (qui augmente la portée des bombes d'un bloc dans les quatre directions) et de type *bomb* (qui permette à un joueur de poser une bombe supplémentaire) ne sont pour l'instant pas gérés.

### 3.3 Recherche de chemin optimal

Pour que votre joueur robot soit capable de se diriger tout seul vers une cible désignée, vous devrez implémenter l'algorithme A\* :

[http://en.wikipedia.org/wiki/A\\_star](http://en.wikipedia.org/wiki/A_star)

Il s'agit d'une généralisation de l'algorithme de Dijkstra (que vous avez déjà vu en cours) intégrant une heuristique permettant (normalement) de réduire le nombre de noeuds à explorer.

Dans un premier temps, vous pourrez vous contenter d'implémenter l'algorithme de Dijkstra<sup>1</sup>. Puis vous pourrez modifier cette implémentation pour en faire un A\* en trouvant une heuristique adaptée.

Vos implémentations d'algorithmes de recherche de chemin optimal devront être capables de travailler en tenant compte ou non des blocs destructibles.

### 3.4 Joueur robot : « Intelligence artificielle »

Vous devrez spécifier le comportement de votre joueur robot puis l'implémenter. Le but étant d'éliminer le joueur adverse, une stratégie triviale consiste à se rapprocher suffisamment de l'adversaire (en calculant et mettant à jour un chemin optimal sans tenir compte des blocs destructibles et en tâchant d'éviter les bombes) pour déposer une bombe à proximité puis aller se placer hors de portée de la bombe.

### 3.5 Si vous voulez en faire plus...

En plus des bonus augmentant la portée des bombes et le nombre de bombes posables, il existe trois autres bonus spéciaux : la télécommande (qui permet de faire exploser la bombe la

<sup>1</sup>Il est fortement conseillé d'utiliser les structures de données et algorithmes déjà implémentés dans le JDK : cf. `java.util.*`

plus ancienne posée), le fantôme (qui permet de passer à travers les blocs pleins destructibles) et l’invincibilité. Gérer ces types de bonus et modifier le comportement de votre joueur robot pour en tenir compte.

## 4 Évaluation

On rappelle qu’il faut porter une attention toute particulière aux points suivants :

- bien présenter le code,
- documenter le code (les modifications que vous apportez au code existant, et les parties délicates de votre code),
- implémenter des tests unitaires (quand cela est pertinent, par exemple, ici pour la génération de carte, et l’algorithme de recherche),
- ne pas (trop) modifier les fonctionnalités du code existant.

La soutenance des projets aura lieu à une date communiquée ultérieurement (entre le 19 et le 23 mai).

Le vendredi 16 mai au plus tard, vous devrez faire parvenir<sup>2</sup>, un rapport de projet décrivant votre implémentation (ce que vous avez réussi à implémenter et comment...) ainsi que le code de votre projet. Apportez beaucoup de soin à ce rapport (et au code), car il servira de base à l’évaluation de votre projet.

Pour la soutenance, vous devrez :

- présenter une partie illustrant les différents points demandés (création de niveau, bonus, recherche de chemin, AI),
- savoir (évidemment) répondre aux questions...

Enfin, le projet est à réaliser en groupe de 2 ou 3 personnes mais la soutenance sera individuelle.

---

<sup>2</sup>par mail, à l’adresse <mailto:labatut@di.ens.fr>