# INFO-H-200 : Programmation orientée objet
# Projet

Professeur : Hugues Bersini
http://cs.ulb.ac.be/public/teaching/infoh200

Année académique 2014-2015

---

## Sujet

On vous demande la réalisation en Java d'un jeu de Bomberman qui sera jouable entre plusieurs joueurs, sous la forme d'interface graphique en utilisant une approche orienté objet. Bomberman est un jeu vidéo où le joueur incarne un poseur de bombes, le but étant de faire exploser les adversaires/ennemis pour gagner .

### Au début de la partie

Le plateau a une taille paramétrable par l'utilisateur (ici 15x15 blocs). Les joueurs sont disposés dans les coins. Les deux emplacements à côté d'eux sont libres (un joueur ne peut pas être directement bloqué). Des blocs incassables sont disposés de manière régulière, comme dans un damier (voir image 1). Des blocs cassables, ici en blanc, (+/- 100 pour un plateau de 15x15) sont disposés de manière aléatoire.

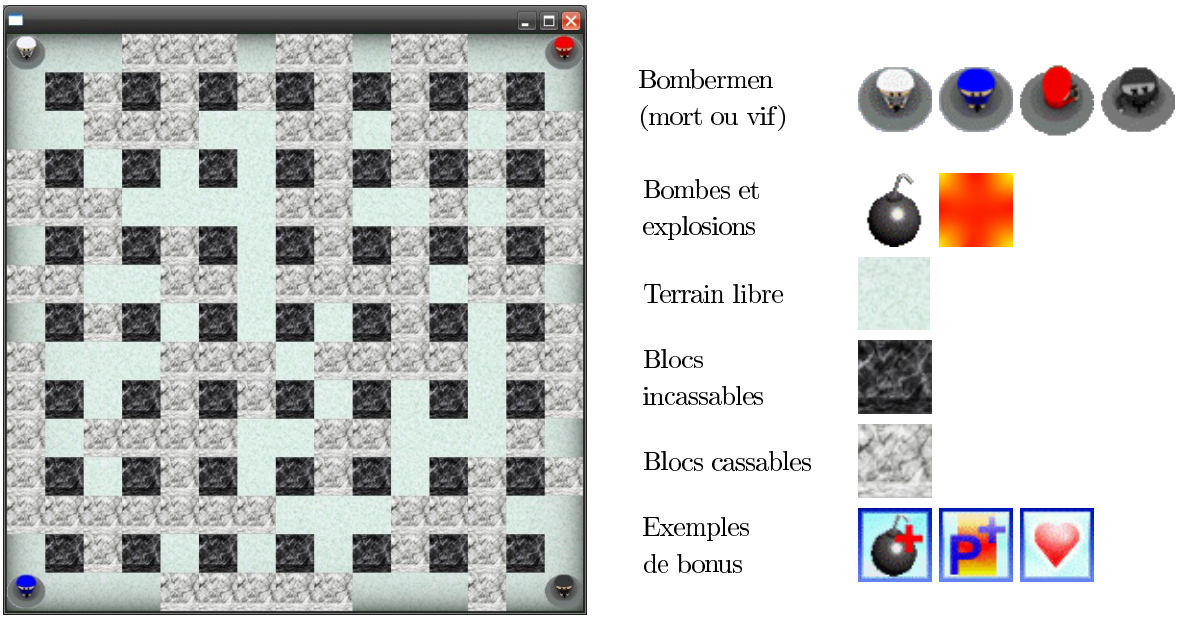


FIGURE 1 – Exemple de composition d'un plateau de jeu

Tous les joueurs en début de partie ne peuvent déposer qu'une bombe à la fois (la bombe placée doit exploser avant de pouvoir en replacer une autre) et la bombe explose dans un rayon d'une case autour de son emplacement (voir images 2).

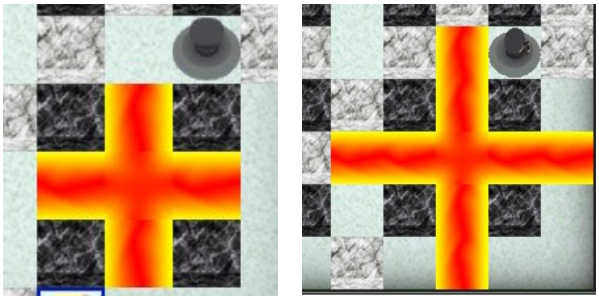

FIGURE 2 – Rayon d'explosion pour une puissance de 1 (à gauche) ou de 2 (à droite)

### Pendant la partie

Les joueurs peuvent se déplacer dans les 4 directions mais sont bloqués par les bords du plateau, les blocs (cassables ou incassables) ainsi que les bombes des autres joueurs.
Un joueur peut placer une bombe s'il en a encore le droit (au début maximum une bombe à la fois, cette limite peut évoluer au cours de la partie, par exemple grâce à un bonus). La bombe prend alors un certain temps avant d'exploser. Attention ! Une bombe touchée par une autre explose à son tour et ce immédiatement (voir image 3).

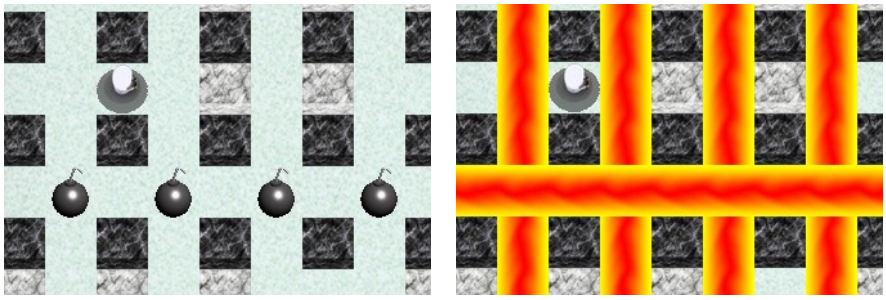

FIGURE 3 – Explosion en chaîne

Lorsqu'un bloc cassable est détruit, il peut soit simplement être détruit, soit en plus d'être détruit laisser une option. Le type des options est également aléatoire. Les meilleures options apparaitront de manière plus rares que les options banales... Les bombermen peuvent alors ramasser les options en passant dessus.
Un bomberman perd une vie lorsqu'il est touché par une explosion et meurt lorsqu'il n'a plus de vie. Par défaut un bomberman n'a qu'une seule vie. Le gagnant est le dernier bomberman en vie.

### Bonus et amélioration

Nous faisons confiance à votre imagination pour le choix des bonus :
- Augmenter le nombre de vie
- Augmenter la portée des explosions
- Augmenter le nombre de bombes
- Possibilité de pousser ou de shooter dans les bombes
- Système de piège
- Utiliser d'autres armes (rockets, grenades,...)
- ...

A ces bonus, vous pouvez également y ajouter différentes améliorations majeurs, tel que
- IA pour jouer contre l'ordinateur
- Jeu en réseau
- Effets sonores et/ou bande son
- ...

## Consignes

On vous demande d'implémenter une application graphique en Java en respectant et en illustrant les concepts orientés objet vus au cours et aux TPs (exceptions, interface, héritage, généricité, polymorphisme, design pattern...). Les éléments à considérer en priorité sont l'architecture et la modélisation de votre programme.
Le projet peut être réalisé individuellement ou par groupe de 3 étudiants maximum, du même groupe de TP (même jour), mais la défense, qui aura lieu sur machine (de la salle informatique ou personnelle), comportera des questions individuelles. Chaque groupe doit envoyer un mail à l'adresse aschenke@ulb.ac.be contenant les noms et adresses email de chaque membre du groupe, avant le **20 mars 2015**.
Le projet comprend la remise d'un premier rapport d'analyse du problème et de conception (décrit ci-dessous), de la remise des délivrables (bref rapport, code source complet et commenté), ainsi que d'une défense orale de votre projet.

## Rapport d'analyse du problème et de conception

Un rapport (max. 5 pages) vous est demandé portant sur votre phase d'analyse (fonctionnelle), de conception et de modélisation, et devra contenir les éléments suivants :
- une description et motivation de votre architeture orientée objet et en particulier des Designs Patterns que vous comptez mettre en oeuvre.
- une description des fonctionnalités qui seront implémentées.
- le résultat de votre modélisation sous forme de diagrammes UML (diagramme objet, de classes, de séquences...)

Ce rapport est à envoyer pour le **vendredi 10 avril 2015** via email à l'adresse aschenke@ulb.ac.be.

## Délivrables du projet

Les délivrables du projet sont les suivants :

1. Le code source complet et commenté (avec tous les images nécessaires)
2. Un exécutable
3. Un rapport décrivant votre approche du problème et la manière dont vous vous êtes réparti le travail

Le rapport demandé devra contenir également les éléments suivants :
- une description synthétiques des fonctionnalités qui sont implémentées
- les diagrammes UML suivants : diagramme objet, diagramme de classe et diagramme de séquence (sur une partie identifiée de votre programme).
- une description et motivation de votre architeture finale et en particulier des Designs Patterns que vous avez mis en oeuvre.

Les délivrables seront remis au plus tard le **vendredi 8 mai** à Arnaud Schenkel sous forme numérique (envoyez un email à aschenke@ulb.ac.be avec pour sujet « INFOH200 : nom_etudiant_1 nom_etudiant_2...» à partir de votre compte ULB). Aucune remise de projet au-delà de cette date ne sera prise en considération.

## Evaluation

La défense du projet consistera en une brève démonstration de votre programme ainsi que quelques questions relatives à votre code. L'évaluation aura lieu la **semaine du 11 mai** et tiendra compte notamment des critères suivants :
- l'architecture initiale proposée (premier rapport) et celle implémentée
- le respect des règles de bonnes pratiques de la programmation orientée objet
- la maîtrise des concepts vus au cours et aux TPs
- la qualité de votre implémentation et la structure du code
- les possibilités de paramétrage de l'application
- l'originalité des bonus et des améliorations complémentaires
- le contenu de votre rapport de fin de projet
- la qualité de vos réponses aux questions lors de la défense