

Master 1  
Sciences et Ingénierie du Réseaux, de l'Internet et des Système

Programmation Avancée

# Rapport

Projet  
Arkanoid

Constantin DIVRIOTIS

# Tables des Matières

- 1. Manuel d'utilisation et implémentation**
- 2. Difficultés rencontrées et améliorations possibles**

# 1. Manuel d'utilisation et implémentation

Le projet Arkanoid a rendre pour le lundi 17 Décembre 2018, se lance en effectuant plusieurs commandes :

1. `cmake CMakeList.txt`
2. `make`
3. `./Arkanoid`

**Remarque :** Il est nécessaire de lancer, dans le même répertoire.

Un simple résumé de chaque classe permettra de mieux comprendre les fonctionnalités de chacune des classes :

- `Ball.cpp` : gestion de l'affichage de la balle, gestion de sa vitesse et de sa direction,
- `Bonus.cpp` : gestion des de l'affichage de chaque type de bonus et gestion du bonus actif (un seul bonus peut être actif),
- `Brick.cpp` : gestion de l'affichage de chaque type de brique, initialisation de la vie de la brique et du score obtenue si on détruit cette brique, gestion des briques difficiles et des briques indestructibles,
- `Grille.cpp` : gestion de l'affichage du plateau à partir du fichier `Grille.txt` (fichier permettant la création du plateau de chaque niveau du jeu) et création des briques selon le plateau,
- `Mobject.cpp` : gestion de l'affichage SDL (classe hérité par chacune des classes nécessitant un affichage),
- `Player.cpp` : gestion de la vie du joueur, de son score et du bonus actif,
- `Vaus.cpp` : gestion de l'affichage du vaisseau et des bonus liés à ce dernier,
- `WriteManager.cpp` : permet l'affichage de lettre et/ou chiffres via le fichier `sprites.bmp`,
- `Game.cpp` : fichier principale permettant d'afficher la fenêtre de jeu, d'afficher un menu et le choix du joueur, d'initialiser le jeu avec toutes les variables, de gérer toutes les collisions (notamment vaisseau/ball et

ball/brique), de gérer les effets des bonus, d'afficher le score du joueur et de quitter le jeu proprement.

- `main.cpp` : initialise une classe Game et lance ainsi le jeu.

Le projet respecte la plupart des consignes exigées. L'héritage de la classe `Mobject` apporte beaucoup de modularité, elle permet de gérer la partie SDL facilement. La plupart de la modularité passe également par la découpe en différentes classes et la déclaration de `#define` dans les fichiers `.h` : un seul changement est nécessaire pour que la modification soit effective partout.

Pour ma part, j'ai implémenté la partie de gestion du plateau (`Grille.cpp`), des briques (`Brick.cpp`), des joueurs (`Player.cpp`) et des collisions balle/brique (`Game.cpp`). J'ai géré les changements de niveau et une partie de la gestion des bonus dans le fichier `Game.cpp`. Govindaraj s'est occupé du vaisseau (`Vaus.cpp`), de la bonne gestion des containers, de tout l'affichage SDL avec le `sprite.bmp` (`WriteManager.cpp` et `Mobject.cpp`) et une partie des bonus (`Bonus.cpp`).

Pour le plateau de jeu (et en général), une modification du fichier `Grille.txt` et des `#define` présent dans le fichier `Grille.h` est la modification sera appliquée dans le jeu.

Il est important de comprendre que nous avons placé des ID sur chaque brique (voir fichier `Grille.txt`) pour pouvoir les gérer différemment chacune dans la classe `Brick.cpp`. L'ID -1 correspond à une case vide c'est-à-dire sans brique. Les ID vont de 0 à 13 correspondant chacune à un type de brique. Par exemple, la brique 12 correspond à la brique difficile ("HardWall") et la brique 13 correspond à la brique indestructible. L'implémentation d'une "vie" pour les briques était la manière la plus simple de gérer les différentes briques. Ainsi, lors des collisions, si la vie d'une brique tombe à 0, elle disparaît et le joueur reçoit le score associé à cette brique. Le score est mis à jour automatiquement et affiché grâce au `WriteManager.cpp`.

L'utilisation des containers est nécessaire pour la gestion de plusieurs balls et/ou plusieurs vaisseaux, c'est pourquoi nous gérons des vectors dans le fichier `Game.cpp` afin de pouvoir gérer ces bonus.

Les bonus ont été implantés en respectant les règles de l'Arkanoid. Nous nous sommes référés principalement à un site<sup>1</sup>, car le sujet proposé n'offrait pas toutes les informations nécessaires.

**Remarque :** Le code fourni par M. Théry a été le point de départ de notre projet.

## 2. Difficultées rencontrées et améliorations possibles

La principale difficulté rencontrée n'est rien d'autre que le temps. Les différents projets de ce semestre ne nous ont pas permis de consacrer autant de temps que nous le voulions pour le projet Arkanoid.

Nous allons énumérer et expliquer explicitement les différents problèmes auxquelles nous nous sommes confrontés :

- **gestion des templates** : dès le début du projet, nous voulions appliquer des notions vues en TP telles que les templates. Mais, nous nous sommes vite rendu compte de la complexité de la chose et de la difficulté à le mettre en pratique sur le projet (utilisation inutile sur les sprites ou sur les déplacements/positions des différentes classes).
- **collisions** : les collisions fonctionnaient correctement assez rapidement, mais lorsque la balle touchait le coin d'une brique, la balle rebondissait anormalement. Cette gestion de la collision d'un coin d'une brique a été fastidieuse et plutôt longue.
- **containers** : lorsque nous avons commencé les containers (c'est-à-dire peu après l'implémentation des bonus), des erreurs mémoires sont apparus à notre grand désarroi. C'est mon camarade qui a géré ce problème notamment, mais il nous a fait perdre du temps, car les différentes erreurs n'étaient pas visibles au premier coup d'oeil.

La principale amélioration que nous aurions voulu proposer, si notamment nous aurions plus de temps à consacrer à ce projet, est une application réseau du jeu. Nous étions d'accord pour tenter d'implémenter cette option dès le début, car cette partie nous intéressait énormément, malheureusement, nous n'avons même pas eu le temps d'essayer de l'implémenter.

[1] - [https://strategywiki.org/wiki/Arkanoid/Getting\\_Started](https://strategywiki.org/wiki/Arkanoid/Getting_Started)