**Projet Casse-Brique**

**Lysa Gramoli – ClÉment Rouault**

# Spécifications

Dans cette partie, nous allons décrire les fonctionnalités de notre programme. Le programme doit pouvoir interagir avec l’utilisateur.

## Fonctionnalité liée à l’interaction avec l’utilisateur

Le jeu devra :

* Indiquer au joueur lorsqu’il aura gagné ou perdu
* Créer une partie
* Quitter une partie
* Indiquer le score du joueur
* Utiliser la caméra afin de savoir si le joueur souhaite aller à gauche ou à droite avec sa main

Le programme devra aussi respecter les règles du jeu :

* Lorsque la balle touche un mur ou le palet, elle rebondit
* Lorsque la balle touche une brique, elle rebondit et supprime la brique
* Lorsque la balle touche le mur du bas, le joueur perd un point de vie

## Interface graphique

L’outil utilisé pour la représentation graphique est OpenGL. Le jeu comportera une zone ou les briques seront affichées.

# Conception

Classes utilisées :

* **MyGLWidget :**

Classe contenant une instance de Model, et permettant de gérer l’affichage des différents objets.

* **Model :**

Classe contenant tous les objets du jeu ( briques, balles, palet, joueur, murs) et des variables nécessaires à certaines fonctions.

* **Reflector :**

Classe abstraite pour tous les objets sur lesquels les balles peuvent rebondir.

* **Brick :**

Classe représentant une brique. On retient la coordonné en bas à droite de la brique pour la positionner.

* **Paddle :**

Classe représentant le palet. On retient la coordonné en bas à droite de la brique pour le positionner.

* **Wall :**

Classe représentant les murs. On n’a finalement pas besoin de 4 instances de cette classe, et même pas besoin de cette classe pour jouer, mais nous l’avons gardée car elle contient la fonction de représentation des 4 murs.

* **Ball :**

Classe représentant une balle. On retient la coordonné du centre de la balle pour la positionner.

* **Player :**

Classe représentant les données d’une partie en cours.

DIAGRAMME DE CLASSES