

# Brick Breaker – Rendu 3

## 1 Objectifs du rendu 3

Finaliser le projet en implémentant la gestion des déplacements, du score, des vies et des conditions de fin de partie.

Prendre également du recul afin d'évaluer les forces et les faiblesses rencontrées lors du travail en groupe.

## 2 Contraintes techniques

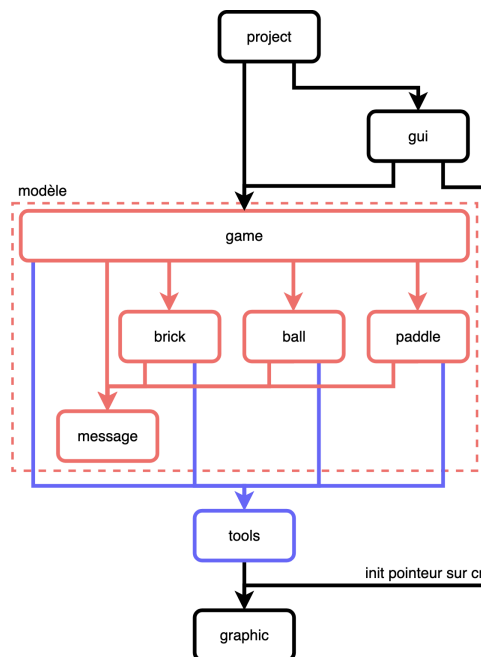


FIGURE 1 – Architecture sans interface graphique

- **Interface** : interface graphique utilisant GTKmm (partiellement fournie).
- **Compilation** : `make`.
- **Exécution** : `./project test.txt` ou `./project`.

## 3 Comportement attendu

- **Gestion des erreurs** : identique au rendu 2.
- **Affichage** : identique au rendu 2.
- **Interactions** : toutes les interactions doivent être fonctionnelles (interactions avec la fenêtre graphique et entre les éléments du jeu).

## 4 Livrable

En plus du code, un rapport (2 pages maximum) doit être fourni.

Le rapport ne contient ni page de titre ni table des matières. Il est rédigé avec une police de taille comprise entre 11 et 14, en interligne simple. Il peut être écrit en français ou en anglais ; une orthographe ou une grammaire défailante peut induire les correcteurs en erreur.

Le rapport ne doit pas dupliquer les précédents rendus. Il est organisé en deux parties :

— **Première partie (1 page maximum) :**

— Optionnel : liste des éventuels changements dans la structuration des données ou des algorithmes par rapport au rendu précédent.

— Description de l'exécution du programme pour les événements suivants (insérer des captures d'écran pertinentes — pas de photographies — et décrire brièvement le comportement du programme) :

— Collision de deux balles de rayons différents (deux images avant et deux après afin d'observer l'évolution de la vitesse).

— Destruction de chaque type de brique (image avant et après, avec la balle incidente).

— **Deuxième partie (1 page maximum) :**

— Activité individuelle de chaque membre du groupe.

— **Méthodologie :**

— Organisation du travail en groupe et outils utilisés (Git, VSCode, etc.).

— Ordre de développement des modules et méthodes de test associées.

— Proportion de travail simultané (côte à côte ou en ligne sur le même code) par rapport au travail indépendant. Avec le recul, modifieriez-vous cette proportion ?

— Bug le plus fréquent : causes et résolution.

— Usage éventuel d'un outil d'IA : gain ou perte de temps (quantification approximative).

— **Conclusion :** brève auto-évaluation du travail réalisé (robustesse et performance du code) et de l'environnement mis à disposition (points forts, points faibles, améliorations possibles).