Brick Breaker – Rendu 2

1 Objectifs du rendu 2

Intégrer le modèle validé au rendu 1 dans l'interface graphique fournie.
Ce rendu se concentre sur la visualisation et les interactions avec l'interface graphique.

2 Contraintes techniques

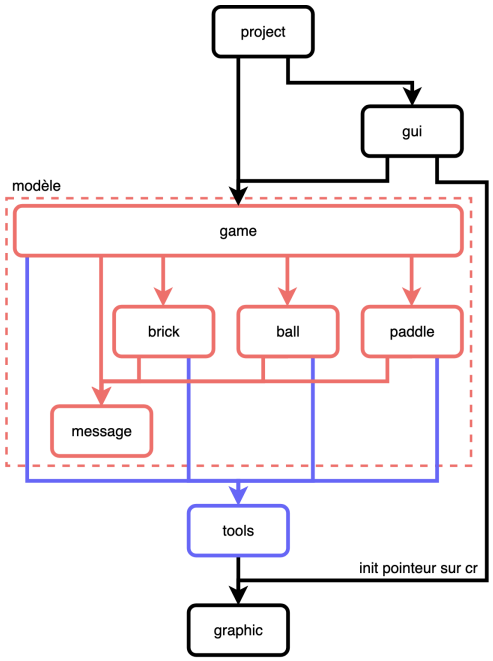

FIGURE 1 – Architecture générale du projet

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

3 Comportement attendu

— **Gestion des erreurs :** Le programme ne doit plus se fermer en cas d'erreur de lecture. Il doit afficher le message d'erreur (identique à celui du rendu 1), vider l'arène (canvas blanc) et rester ouvert.

En cas d'absence de fichier, l'interface graphique doit s'ouvrir avec un canvas blanc et des compteurs à 0. Aucun message d'erreur n'est alors affiché.

— **Affichage :** Utiliser le module `graphic`, par l'intermédiaire du module `tools`, pour dessiner les briques, la raquette, les balles et l'arène.

— **Interactions :** Implémenter le déplacement de la raquette via la souris. Seule la raquette se déplace ; les collisions entre la raquette et les balles pendant le jeu sont ignorées.

Le bord de l'arène ainsi que les briques empêchent le déplacement de la raquette.

Les boutons **Exit**, **Open**, **Save**, **Restart**, **Start** et **Step** doivent être fonctionnels.

4 Livrable

À partir du rendu 2, il est demandé de mettre en place une hiérarchie de classes afin de gérer les différents types de briques.

Un rapport d'une page maximum doit être fourni, décrivant l'organisation du groupe et la structure du code du modèle, en particulier :

— Activité individuelle de chaque membre du groupe

— La hiérarchie de classes des briques : quels attributs et quelles méthodes sont gérés par la superclasse et par les classes dérivées ? Optionnel : si vous utilisez le polymorphisme, quelles méthodes sont déclarées virtuelles afin de le mettre en œuvre ?

— La structuration des données des autres entités du modèle : quels sont les attributs de la classe `Game` ? Où et comment sont mémorisés les différents ensembles gérés par le jeu ?

— Brève description des types définis dans le module `tools`.