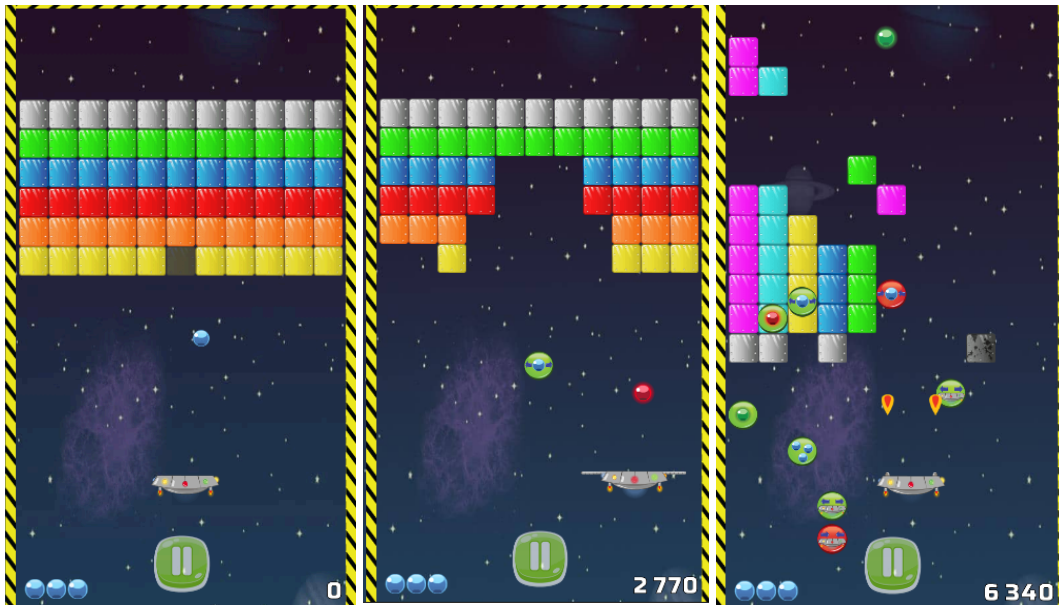


## Projet de Programmation Avancée: Remake Casse Brique!

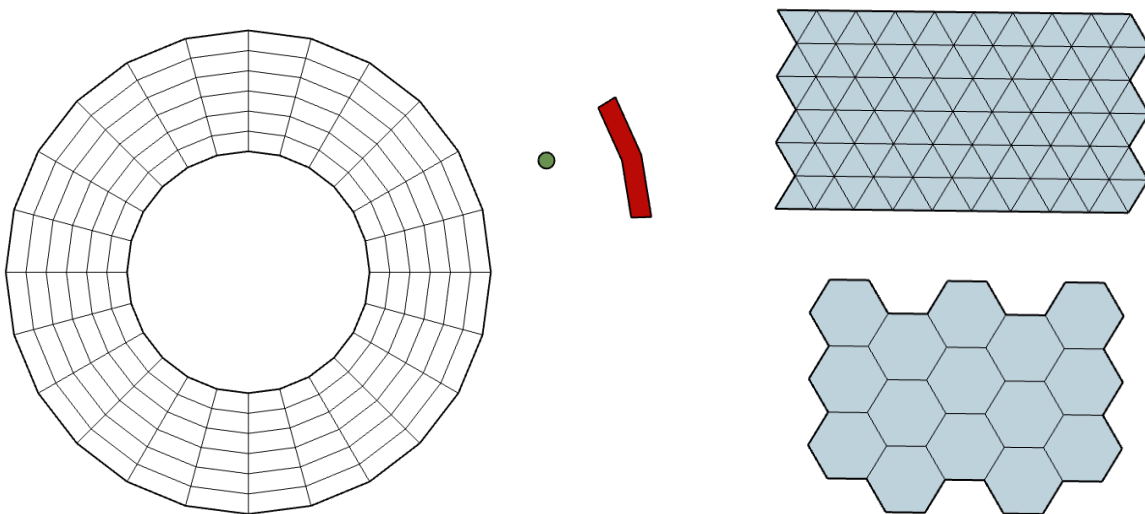


Dans ce projet, vous serez amenés à programmer en binômes votre propre version du jeu classique Casse Brique, en C++ moderne avec un affichage en SDL2. Vous trouverez un exemple du jeu au lien suivant: <https://poki.com/en/g/brick-breaker>

Il existe de nombreuses variantes du jeu, avec différents bonus, différentes interactions, et différents styles. Vous êtes libres d'être un peu créatif tant que vous respectez les critères suivants:

- une plateforme déplaçable au clavier ou à la souris.
- une balle qui rebondit en fonction de son angle d'impact.
- des briques de différents types avec différentes résistances.
- au moins trois types de bonus/malus dont le multi-balle.
- différentes grilles chargées depuis un fichier ASCII.

Pour compléter le projet vous implémenterez l'une des deux possibilités ci-dessous:



Soit 1) une grille circulaire avec la plateforme en orbite autour, soit 2) supporter différentes formes de grilles de briques: hexagonales et triangulaires (<https://www.redblobgames.com/grids/hexagons/>) .

Un exemple de code SDL2 est disponible ici: [http://igg.unistra.fr/people/thery/Prog\\_Avancee/](http://igg.unistra.fr/people/thery/Prog_Avancee/)  
Vous pourrez utiliser les fonts de <https://paulbourke.net/dataformats/hershey/> pour l'interface.

**L'évaluation prendra en compte le contenu, mais surtout la modernité, la propreté, et l'organisation du code.**

Vous respecterez les critères suivants:

- Utilisez la STL au maximum.
  - Pas de tableau! des conteneurs STL.
  - Pas de pointeur! des unique/shared/weak\_ptr.
  - Utilisez les algorithmes de la STL
- Ecrivez des classes pour vos objets et encapsulez la SDL2.
- Utilisez les possibilités du C++ 11/14/17/20/etc... (Lambda, for range).
- Utilisez de l'héritage et du polymorphisme.
- Soyez const-correct et efficaces, pas de copie inutile!
- Documentez votre code.
- Utilisez une convention de codage.