

Sujet de Projet Informatique : *Casse briques* et jeux de balle

Sujet proposé par Aldric Degorre.

Dixit Wikipédia : *Le casse-briques est un genre de jeu vidéo souvent classé dans la catégorie arcade, apparu en 1976 avec le jeu Breakout. Il est directement inspiré de Pong. Le principe général est de détruire, au moyen d'une ou plusieurs balles, un ensemble de briques se trouvant dans un niveau pour accéder au niveau suivant.*

Objectif : L'objectif est de programmer un jeu de casse-briques, en se focalisant sur deux aspects : la modularité (diverses options et modes de jeu, diverses versions des briques) et la simulation physique de la balle (laquelle doit être elle-même encapsulée dans un module réutilisable dans d'autres jeux à balle en 2D : flipper, pong, air hokey...)

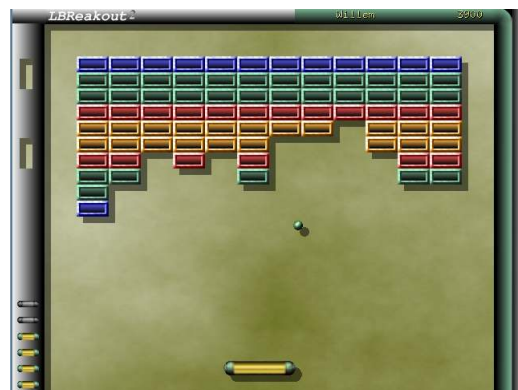


FIGURE 1 – Capture d'écran du jeu Breakout

Principe de base du jeu : Le terrain est typiquement de la taille de l'écran. En haut (environ la moitié supérieure) se situe un mur de briques rectangulaires. En bas, une raquette contrôlée par le joueur peut se déplacer latéralement. Entre les deux, une balle se déplace en trajectoire rectiligne, et rebondit sur les surfaces (briques, murs, raquette).

Lorsque la balle rebondit sur une brique la brique se casse.

Lorsque la balle atteint le bas de l'écran, le joueur perd une vie au lieu de rebondir.

L'objectif est de finir de casser toutes les briques avant d'avoir perdu toutes ses vies.

Principes typiques du jeu : Le jeu de base est typiquement enrichi avec d'autres principes :

- briques solides dont la destruction nécessite plusieurs impacts
- briques indestructibles
- briques libérant un bonus (qui tombe vers le bas et qu'il faut rattraper)
- bonus vie supplémentaire
- bonus balles multiples
- bonus d'élargissement de raquette
- ...

Il y a généralement plusieurs niveaux avec différentes dispositions des briques (de différents types).

Autres possibilité de jeu :

- physique plus aboutie : pesanteur, effets sur la balle (qui a une rotation propre), balle non-circulaire et briques non rectangulaires
- jeu multi-joueur (par exemple, inspiré de Pong) ; plusieurs possibilités : via une IA, plusieurs contrôles humains sur une même machine, réseau,
- jeu infini : au lieu d'une succession de plateaux à vaincre, le jeu se présente sous la forme d'un plateau infini (ou très long) qui descend lentement vers le bas, risquant de coincer la

balle. On peut envisager soit un scénario écrit à l'avance, soit de la génération procédurale (ou un hybride).

- d'autres types d'éléments dans le plateau (briques qui bougent, voire ennemis, ...)
- jeu en 3D : la balle est une boule, les briques et la raquette des pavés droits ; la raquette se déplace dans 2 dimensions et le "plateau" est quant à lui une construction spatiale.

Physique : une balle est un objet indéformable (en première approximation) de forme sphérique (ou circulaire, dans un univers en 2D), muni de caractéristiques physiques (rayon, masse, coefficient de frottement, ...) et d'un état (position, vitesse, et peut-être angle et vitesse de rotation). Le moteur physique doit être à même de simuler les évolutions de l'état d'un tel objet dans un environnement physique : un espace bi- ou tri-dimensionnel exerçant différentes forces sur lui (gravitation, frottements, magnétisme, contraintes solides, ...) et le soumettant à des chocs (modèle du choc élastique, ou modèles plus évolués, en fonction des caractéristiques de la balle¹).

Réalisation technique En Java, avec la bibliothèque graphique JavaFX, ou bien Swing (autres choix techniques possibles, après discussion).

Méthodologie Une certaine rigueur sera demandée pour la physique : le modèle physique ne décrira pas forcément la physique "vraie"², mais il devra respecter un certain nombre d'équations mathématiques prédéfinies (bilan des forces, choc élastique, etc.).

Des tests devront être écrits pour s'assurer que la simulation respecte ces équations avec une certaine marge d'erreur³.

Pour bien séparer la partie moteur physique et jeu, il s'agira de créer deux projets séparés⁴, le second dépendant du premier.

Tout le code devra être pensé en des termes de bonne programmation objet, de sorte à limiter au maximum les répétitions (exactes ou simples similarités) dans le code.

Quantité attendue Les principes de base ainsi que tous les principes typiques listés dans le sujet doivent être implémentés. On s'attend à ce que les étudiants aient le temps de s'attaquer aussi à plusieurs des possibilités de la catégorie "Autres".

1. https://en.wikipedia.org/wiki/Collision_response

2. Dans le sens où il s'agit d'une version 2D d'un phénomène spatial 3D dans le monde réel ; et dans le sens qu'il s'agit seulement d'une simulation en temps réel et discret avec des calculs en précision limitée.

3. Erreur inévitable du fait des limites de la simulation.

4. Il peut s'agir de modules Gradle.