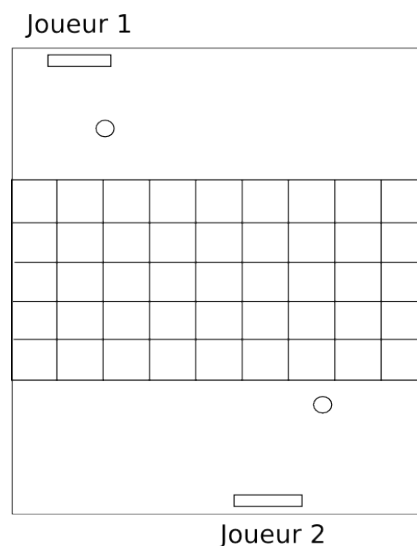


ARKANOPONG

L'objectif de ce projet est la réalisation d'un jeu mélangeant le célèbre Pong et son cousin le Casse Brique. L'application sera réalisée à l'aide des bibliothèques OpenGL et SDL.

1 Description du gameplay

Le jeu se joue à deux. Un joueur est placé en haut de l'écran et le second en bas. Un ensemble de briques les sépare comme dans la configuration suivante :



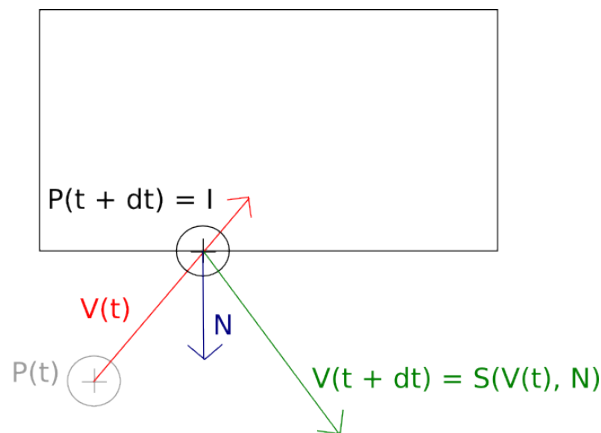
Chacun des deux joueurs possède sa propre balle. Comme dans le jeu du casse-brique, les balles peuvent détruire les briques. Tant que suffisamment de briques n'ont pas été détruites pour créer un chemin du joueur 1 au joueur 2, le jeu est un casse-brique classique. La balle de chaque joueur doit être rattrapée par ce dernier sous peine de perdre un point de vie.

Dès qu'il existe un chemin entre le joueur 1 et le joueur 2, le jeu peut être vu comme un pong (auquel on a ajouté des briques) et chaque joueur doit rattraper la balle de l'ennemi en plus de la sienne.

Le jeu se termine lorsque le nombre de points de vie d'un des deux joueurs atteint 0. L'autre joueur est alors déclaré vainqueur et le premier doit lui donner 10 euros.

1.1 Comportement des balles

Une balle peut être considérée comme un point mobile pouvant entrer en collision avec des rectangles (briques ou barre d'un des joueurs). Elle est donc caractérisée par sa position $P(t)$ et sa vitesse $\vec{v}(t)$ à l'instant t . A chaque image (tour de la boucle d'affichage), elle se déplace le long de son vecteur vitesse d'un facteur proportionnel au temps écoulé depuis la dernière image ($P(t + \Delta t) = P(t) + \Delta t \vec{v}(t)$). Il faut également tester les collisions avec les briques et les barres. Si une collision est détectée au point I , la position de la balle doit être placée en I ($P(t + \Delta t) = I$) et sa vitesse doit être modifiée pour devenir le symétrique $S(\vec{v}(t), \vec{N})$ de la vitesse à l'instant t par rapport à la normale \vec{N} au point d'impact :



dt : temps écoulé entre l'image i et l'image $i + 1$

1.2 Comportement des briques

Une brique est un élément rectangulaire statique. Elle bloque la progression des balles jusqu'à être détruite. On peut imaginer plusieurs types de briques :

- Des briques indestructibles.
- Des briques disparaissant après le premier impact.
- Des briques apportant des bonus après destruction.
- Des briques apportant des malus après destruction.
- etc.

1.3 Contrôle des barres

Chaque joueur contrôle sa barre à l'aide de deux touches du clavier. Chaque barre peut se déplacer à droite ou à gauche mais ne peut pas sortir de la zone de jeu.

2 Fonctionnalités de base

Vous réaliserez ce jeu en langage C à l'aide des bibliothèques SDL et OpenGL. La SDL sera utilisée uniquement pour le fenêtrage, la gestion des événements et le chargement d'images. OpenGL sera utilisée pour l'affichage. Voici la liste des fonctionnalités à implanter :

- Un niveau jouable à deux (sur le même clavier avec 4 touches de contrôle) dont la structure est similaire au premier schéma. Le joueur du haut utilisera les touches A et Z pour se déplacer à gauche ou à droite. Celui du bas utilisera les flèches directionnelles.
- Au moins 4 types de bonus pour les briques (par exemple : agrandissement de la barre du joueur, ajout d'un point de vie au joueur, retrait d'un point de vie à l'ennemi, ajout d'une balle ne blessant pas le joueur lorsqu'elle est perdu).
- Affichage de briques texturées (vous choisirez les images en fonction de l'ambiance que vous souhaitez donner à votre jeu).
- Affichage du nombre de points de vie de chacun des joueurs (avec des coeurs par exemple).
- Le niveau doit être chargé depuis un fichier. Le format de fichier à respecter est le suivant :

```
largeur hauteur  
brique_0 brique_1 brique_2 [...] brique_largeurxhauteur
```

`largeur` et `hauteur` sont les dimensions de la grille de briques. Viens ensuite la liste des briques. Chaque entrée `brique_n` est un entier identifiant le type de brique. Puisque vous devez au moins implanter 4 types de bonus, il faudra pouvoir mettre des entiers au moins entre 0 et 4 inclus (0 correspondant à la brique sans bonus et 1, 2, 3, 4 aux différents types de briques que vous aurez choisis). Le fichier doit être encodé au format ASCII et non binaire. Voici un exemple de fichier valide :

```
2 2  
0 1 2 0
```

Il correspond à un niveau composé de 4 briques. La première ligne est formé d'une brique simple et d'une brique de bonus 1. La seconde ligne est formé d'une brique de bonus 2 et d'une brique simple. A noter que nous utiliserons un fichier de niveau écrit par nos soins pour tester votre chargeur de niveaux, il faut donc respecter le format !

L'implantation de tout ces éléments vous apportera une note entre 10 et 15. Pour avoir plus il faudra ajouter des extensions.

3 Extensions

Voici quelques idées d'extensions :

- Une intelligence artificielle controlant la deuxième barre (pour les gens associables qui n'aiment pas jouer à deux).
- La possibilité de changer le thème graphique (les textures associées aux briques).
- La généralisation du jeu à un nombre quelconque de joueurs (on peut imaginer qu'ils soient répartis sur un polygone régulier).
- La connexion en réseau pour jouer sur internet (beaucoup plus difficile que les autres bonus, mais tellement fun!).
- etc.

4 Quelques conseils

N'essayez pas de développer ce jeu en une seule fois. Cela doit se faire par itération, fonctionnalité par fonctionnalité. Voici un exemple de feuille de route à suivre :

- Déplacement d'une unique balle dans l'écran rebondissant contre les murs.
- Ajout d'une barre contrôlable pouvant repousser la balle.
- Ajout de la deuxième barre et de la deuxième balle.
- Ajout de la gestion des points de vie de chacun des joueurs et du fait que les balles sortent de l'écran par le haut et par le bas.
- Ajout de la gestion des briques (blocage et destruction seulement).
- Ajout des bonus associées aux briques.

Chaque fonctionnalité doit être codée, testée et validée avant de passer à la suivante.

5 Evaluations

Une version implantant uniquement les fonctions de base sera notée sur 12 points. Le reste des points sera attribué aux extensions.

Attention : les extensions ne seront pas prises en compte si les fonctionnalités de base n'ont pas été implantées.

6 Rapport

Vous fournirez en version papier et en version électronique un rapport de 10 pages maximum. **Consacrez suffisamment de temps au rapport car il représente une bonne partie de la note finale.** Essayez de respecter au mieux les directives suivantes :

- **Votre rapport doit comprendre une page résumant vos travaux avec les indications suivantes :**
 - éléments demandés et codés qui fonctionnent.
 - éléments demandés et codés qui ne fonctionnent pas.
 - éléments demandés mais pas codés.
 - éléments non demandés et codés qui fonctionnent.
 - éléments non demandés mais pas codés ou qui ne fonctionnent pas.
- Ne perdez pas de temps à réexpliquer le sujet du projet, l'enseignant le connaît déjà, faites seulement un bref résumé de quelques lignes. De manière plus générale, ne détaillez pas des méthodes déjà expliquées dans l'énoncé à moins que vous les ayez modifiées.
- Un rapport sert surtout à montrer comment vous avez fait face aux problèmes (d'ordre algorithmique). Certains problèmes sont connus (on en parle dans l'énoncé), d'autres sont imprévus. Montrez que vous les avez remarqués et compris. Donnez la liste des solutions à ce problème et indiquez votre choix. Justifiez votre choix (vous avez le droit de dire que c'est la méthode la plus facile à coder).
- Il ne doit figurer aucune ligne de code dans votre rapport. Un rapport n'est pas un listing de votre programme où vous détaillez chaque fonction. Vous devez par contre détailler vos structures de données et mettre du pseudocode pour expliquer vos choix algorithmiques.
Il est autorisé d'utiliser des "raccourcis" tels que "initialiser le tableau `tab` à 0" plutôt que de détailler la boucle faisant la même chose.
- n'hésitez pas à mettre des images dans votre rapport pour illustrer vos propos et vos résultats.
- Enfin, il est **très important** de faire la liste de ce que vous avez fait, de ce qui fonctionne correctement et la liste des dysfonctionnements. Précisez quand il s'agit d'options que vous avez rajoutées en plus de ce qui était demandé.

7 Rendu

Vous devrez nous rendre votre projet comportant le rapport, le Makefile, les fichiers .c et .h ainsi que l'exécutable (le tout taré et zippé). Votre archive portera vos 2 noms.

Exemple : `arkanopong_nozick_biri.zip`

Votre projet devra compiler sur les machines de l'université.

Bon courage !