

INFO-F-202
Langages de programmation 2
Université libre de Bruxelles

Projet

John Iacono Alexis Reynouard Simon Renard

1 Sommaire

Le but du projet est de coder à l'aide de Allegro un jeu inspiré de Arkanoid, créé en 1986. Il était basé sur Breakout (1976), qui était lui-même basé sur Pong (1972).

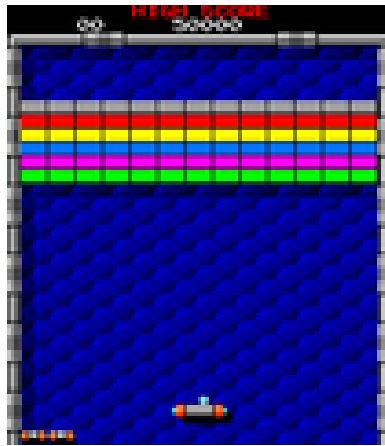


FIGURE 1 – Image du jeu Arkanoid <https://strategywiki.org/wiki/Arkanoid/Walkthrough>

Dans ce jeu le joueur doit casser plusieurs lignes de « briques », en haut de l'écran, grâce à une balle qu'il contrôle en déplaçant une raquette en bas de l'écran. La balle rebondit sur les murs, la raquette et les briques. Dès qu'une brique est touchée par la balle, elle disparaît. Lorsque la balle tombe plus bas que la raquette, elle est perdue et le joueur perd une vie. Le but est de casser toutes les briques avant de perdre toutes ses vies. Il existe une variété de bonus qui tombent lorsque la balle touche une brique et qui sont actifs si vous les attrapez avec la raquette.

Pour référence, voyez :

- la [page Wikipedia de Arkanoid](#),
- la [page strategywiki de Arkanoid](#),
- le [jeu en ligne](#). Notez que cette version du jeu ne correspond pas exactement à ce que vous devez implémenter.

Vous pouvez réaliser le projet **individuellement ou par groupe de deux**. Si vous effectuez le projet en groupe, les deux étudiants recevront des notes identiques (sauf si un étudiant ne se présente pas à la présentation orale).

Vous devez remettre :

- un zip avec votre code, d'autres fichiers nécessaires (vous pouvez utiliser des images ou des fichiers texte) et un Makefile,
- un document au format pdf expliquant votre démarche (voir ci-dessous).

Ces documents sont dûs pour le **5 janvier à 23 h 59** pour la première session.

Un entretien oral aura lieu la semaine du 6 janvier. Vous y ferez une démonstration de votre programme et répondrez à des questions.

Vous devez cliquer [ici](#) pour choisir une heure pour présenter votre projet.

Pour la deuxième session, le projet sera identique et sera rendu en août. La date exacte sera annoncée une fois que le calendrier des examens de la deuxième session sera fixé. Vous pouvez passer l'examen écrit en première session et faire le projet en deuxième session (ou inversement) si vous le souhaitez.

2 Notation

La notation prendra notamment en compte les éléments suivants :

- tâches terminées,
- rapport écrit,
- qualité du code,
- présentation orale et questions,
- retard.

Votre note maximale est 10 si vous ne terminez que les tâches de base et monte à 20 si vous terminez également 10 tâches additionnelles. La qualité de votre code, votre rapport écrit, votre présentation orale et vos réponses aux questions orales déterminent votre note par rapport à la note maximale.

3 Tâches

Les tâches de base consistent en la réalisation d'un niveau fonctionnel de Araknoid. Plus précisément, il faut implémenter les points ci-dessous.

Raquette Le joueur peut bouger sa raquette latéralement en utilisant les touches A ou Q (gauche) et D ou P (droite). Quand la balle rebondit sur la raquette, la direction qu'elle prend dépend de la position du rebond sur la raquette. Si la raquette fait une taille L et que la balle rebondit à une distance x de la gauche de la raquette, alors la balle prend la direction qui forme un angle α avec l'horizontale telle que $\alpha = 30 + 120(1 - \frac{x}{L})$ degrés. La trajectoire de la balle dépend donc uniquement de sa position sur la raquette, pas dans son angle d'arrivée.

Briques Le jeu doit comporter 8 lignes de 14 briques. Pour la tâche de base, elles peuvent être toutes de la même couleur.

Score Lorsque le joueur casse une brique, il gagne 1 point. Le score doit être visible durant la partie.

Vies Le joueur a 3 vies. Son nombre de vie doit être affiché. Lorsque le joueur a perdu toutes ses 3 vies (lorsqu'il n'a plus de balle), le jeu s'arrête et un message de défaite s'affiche avec le score du joueur.

Victoire Un message de victoire apparaît quand le joueur a cassé toutes les briques. Son score est affiché et le jeu peut être démarré en appuyant sur n'importe quelle touche.

Les tâches additionnelles sont à réaliser une fois les tâches de base terminées. Vous devez d'abord terminer la première tâche, « Niveaux », avant d'effectuer l'une des tâches suivantes. Vous pouvez terminer le reste dans n'importe quel ordre. Chaque tâche vaut 1/20.

Niveaux. Une fois que vous avez terminé un niveau, un nouveau niveau doit commencer. Chaque niveau doit avoir une nouvelle disposition des briques et doit être encodé dans un fichier. Vous pouvez utiliser **les niveaux de jeu réels** comme source d'inspiration. Pour pouvoir faire facilement une démonstration, vous devez disposer de raccourcis clavier pour avancer et reculer d'un niveau. Les niveaux doivent être conçus pour pouvoir faire rapidement une démonstration de toutes les fonctionnalités que vous avez implémentées.

Souris / Trackpad. Vous contrôlez la raquette en utilisant la souris plutôt que le clavier.

Briques colorées et meilleur score. Les briques peuvent avoir plusieurs couleurs. Chaque couleur rapporte un nombre de points différent. Un meilleur score s'affiche à l'écran et est stocké dans un fichier. Vous devriez avoir un raccourci clavier pour réinitialiser le meilleur score à 0.

- Blanc : 50 points
- Orange : 60 points
- Cyan : 70 points
- Vert : 80 points
- Rouge : 90 points
- Bleu : 100 points
- Magenta : 110 points
- Jaune : 120 points

Briques argentées et dorées. Les briques argentées doivent être frappées deux fois avant de disparaître. Les briques dorées ne disparaissent jamais (et n'ont pas besoin d'être retirées pour terminer le niveau). Les briques argentées valent 200 points. Après avoir touché une brique argentée une fois, la couleur de son contour devrait changer (ce qui est différent du jeu réel).

Bonus (power-up). Dans le jeu réel, les bonus apparaissent de manière aléatoire. Cependant, pour faciliter le processus de démonstration, chacun des bonus peut être attaché à un bloc spécifique dans le cadre de l'encodage d'un niveau. Le bonus sur un bloc doit être indiqué par une petite lettre sur le bloc. Chaque bonus tombe d'une brique et si vous l'attrapez, il prend effet. Chaque bonus annule l'effet de tous les bonus précédents actuellement en vigueur.

Laser Vous pouvez tirer des lasers vers le haut de l'écran en appuyant sur la barre d'espace. Les lasers peuvent être utilisés contre toutes les briques sauf les briques dorées.

Agrandir Récupérez la capsule bleue pour augmenter la largeur de la raquette.

Attraper Récupérez la capsule verte pour obtenir la capacité d'attraper. Lorsque la balle touche la raquette, elle collera à la surface. Appuyez sur la barre d'espace pour libérer la balle. La balle se relâchera automatiquement après un certain temps.

Ralentissement Récupérez la capsule orange pour ralentir la vitesse de la balle. La collecte de plusieurs capsules orange a un effet cumulatif et la vitesse de la balle peut devenir extrêmement lente. Cependant, la vitesse de la balle augmentera progressivement pour revenir à la vitesse d'origine.

Interruption Récupérez la capsule cyan pour que la balle se divise en trois instances d'elle-même. Les trois balles peuvent être maintenues en l'air. Il n'y a pas de pénalité pour la perte des deux premières balles. Aucune capsule colorée ne tombera tant qu'il y a plus d'une balle en jeu. C'est le seul bonus qui, tout en étant en vigueur, empêche les autres bonus de tomber.

Joueur Récupérez la capsule grise pour gagner une vie supplémentaire.

4 Qualité du code

Votre note de qualité de code sera basé, entre autres, sur la façon dont vous codez en utilisant les principes de la programmation orientée objet. Voici des points que l'on attend dans votre projet :

- Décompositions en classes et en fichiers
- Pas de logique répétée
- Aucun **nombre magique**

- Aucun nombre magique répété!
- Pas de variables globales non constantes.
- Commentaires et documentation du code
- Méthodes et fonctions courtes
- Nom des fonctions, méthodes et variables clairs
- Utilisation correcte des constructeurs et des destructeurs
- Utilisation du modèle de conception MVC (couvert dans l'avant-dernière leçon)

Les classes de dessin de base qui appellent les fonctions allegro doivent être séparées de votre code spécifique à Breakout.

Votre code doit être aussi modulaire que possible.

5 Questions orales

Vous aurez une présentation orale pendant laquelle il vous sera demandé de faire une démonstration de votre programme et de votre code. Durant cette présentation nous vous poserons quelques questions à ce sujet. Nous vous demanderons peut-être d'apporter une modification mineure au code pour voir si vous comprenez votre propre code. Si vous avez réalisé le projet en équipe de deux, les deux membres de l'équipe doivent se présenter ensemble.

6 Rapport écrit

Vous devez inclure un rapport écrit au format PDF, d'au maximum 10 pages. Il n'y a pas de minimum de longueur, mais le rapport doit au minimum comprendre les points suivants

- Tâches. Indiquez les tâches que vous avez accomplies.
- Classes. Pour chaque classe, fournissez l'interface (pas le corps des méthodes) et décrivez brièvement le rôle de chaque classe et comment elle se rapporte aux autres classes.
- Logique du jeu. Décrivez en détail ce qui se passe dans votre code entre le lancement du programme et le moment où la balle touche la première fois une brique.
- Modèle-Vue-Contrôleur. Avez-vous utilisé ce modèle de conception? Si c'est le cas, expliquez comment vos classes correspondent à ce modèle de conception. Sinon, expliquez vos choix.

Prenez garde au fait vous que ce rapport fera partie des critères qui influencent votre note (bien que ce ne soit pas le critère principal), et pas un document optionnel à bâcler. En particulier, essayez de rendre celui-ci propre et clair pour le lecteur.

7 Retard

Vos points seront réduit de 5% pour chaque heure de retard.

8 Ce sur quoi vous n'êtes PAS noté

Vous êtes notés sur votre capacité à créer un programme volumineux, complexe et orienté objet. Vous n'êtes pas noté sur votre capacité à avoir un beau jeu. Normalement, un graphiste créerait de jolis éléments visuels, et vous êtes des informaticiens, pas des graphistes. L'aspect visuel devrait être assez bon pour que les différents éléments soient reconnaissables et que les animations soient raisonnables, mais aller au-delà n'aidera pas votre note.

9 Plagiat

Nous devons aborder le problème du plagiat dans le code. Ce n'est pas facile, car il est tout à fait normal de couper et coller de petits morceaux de code. Cependant, il n'est pas acceptable que vous copiez un gros morceau de code spécifique à ce jeu d'autres étudiants ou du Web.

La règle suivante s'applique : si vous avez plus d'une seule ligne de code que vous avez obtenue ailleurs, vous devez inclure un commentaire avec l'endroit où vous l'avez obtenue (comme l'URL). Si vous utilisez l'IA, vous devez indiquer quel système vous avez utilisé et la question que vous avez posée. Si vous utilisez un système intégré comme copilote, vous devez inclure une section du rapport où vous décrivez comment vous l'avez utilisé. Cette section doit être rédigée par vous-même sans l'aide de l'IA.

Toute violation de cette politique, même minime, peut entraîner un 0 sur le projet.

Nous ne ferons pas de distinction entre ceux qui ont fourni du code et ceux qui ont copié du code ; si deux groupes partagent le code sans attribution, les deux obtiendront un 0. Vous pouvez éviter cela en insérant simplement un commentaire !