

Pièges !

L'objectif de ce projet est d'implémenter un jeu de stratégie multi-joueur appelé *Pièges !* (*Stay alive!* en anglais), mis au point par Milton Bradley en 1971¹.



Figure 1: Un exemple de partie

1 Règles du jeu

Le jeu est constitué d'une grille 7×7 , dotée de 14 *tirettes*, 7 horizontales (oranges) et 7 verticales (blanches). Chaque tirette a un nombre fixé de trous, et peut être placée dans trois positions, en poussant ou en tirant sur ses extrémités. À chaque point de la grille, lorsque deux trous sont superposés (celui de la tirette verticale et celui de la tirette horizontale), la bille située au-dessus tombe, et le joueur perd sa bille. Le jeu se joue à 2, 3 ou 4 joueurs, chacun disposant de 5

¹Stay Alive, site officiel du jeu. Voir aussi la pub française de 1994.

billes de sa couleur choisie. L'objectif du jeu est d'être le dernier joueur à avoir au moins une bille sur le plateau.

1.1 Phase préliminaire

Au début de la partie, les joueurs placent les tirettes dans des positions aléatoires. Ensuite, chacun à leur tour, ils placent une bille sur la grille, aux endroits où il n'y a pas de trou. Lorsque toutes les billes sont placées, le jeu peut commencer.

1.2 Partie

Le jeu se joue au tour par tour. À son tour, un joueur choisit une tirette et la déplace d'un cran dans le sens qu'il souhaite dans le but de faire tomber les billes des joueurs adverses.

1.3 Fin de partie

Le gagnant est le dernier joueur à avoir des billes en jeu.

2 Objectif intermédiaire (V1)

Vous pouvez dans un premier temps travailler sur une version plus simple du jeu (appelée V1), dont la réalisation vous garantit la moyenne à l'UE. Dans cette version :

- Il n'y a qu'un seul joueur, et son objectif est d'éliminer les billes en le moins de coups possibles. Il vous faut donc compter le nombre de coups joués.
- Les tirettes sont circulaires, c'est-à-dire que vous pouvez les faire avancer et reculer autant que vous le souhaitez.
- La position des billes et la position des tirettes sont *hardcodées* : c'est vous qui définissez les variables dans votre code, et vous n'avez pas à implémenter la phase préliminaire du jeu (positionnement aléatoire des tirettes et positionnement des billes par les joueurs).
- Le jeu se joue dans le terminal.

Dans la suite, cette version est appelée V1, et la version qui implémente toutes les règles du jeu est appelée V2.

3 Travail demandé

3.1 Moteur du jeu

Dans cette première tâche, vous devez choisir des structures de données pour représenter l'état du jeu à tout moment de la partie :

- La position des billes sur la grille

- La position des tirettes et la position de leurs trous

Il vous faudra ensuite implémenter les fonctions de base du moteur du jeu :

- Une fonction qui déplace une tirette
- Une fonction qui met à jour les positions des billes, en faisant tomber les billes situées au-dessus de trous

Dans la V2, il vous est de plus demandé : - Une fonction qui génère aléatoirement une tirette, avec au moins un trou (sinon on ne peut pas gagner !) - Une fonction qui permet de placer une bille d'un joueur sur la grille (pour la phase préliminaire)

Dans cette tâche, il vous est demandé de **ne pas** réaliser d'actions de saisie, d'affichage ou de dessin : cela viendra plus tard.

L'évaluation prendra en compte la documentation du code (*doctstrings* et *doctests*) : type des paramètres, valeurs autorisées, effets secondaires de la fonction, erreurs possibles ,etc.

3.2 Interface

Dans un second temps, vous pourrez mettre en place l'interface du jeu, c'est-à-dire tout ce qui permet à un joueur de l'utiliser. On sera attentif à l'*ergonomie* du jeu (il doit être agréable à utiliser), à sa *stabilité* (il ne doit pas planter même quand le joueur se trompe ou réalise des actions imprévues), à son *esthétique* (il doit être joli, ou au moins pas trop repoussant...).

Pour la réalisation de cette tâche, plusieurs options peuvent être envisagées. Il est possible d'associer comme on le souhaite ces différentes options de réalisation d'interface si cela a du sens (par exemple proposer un affichage graphique et une saisie au terminal). Pour la V1, seul l'affichage en mode texte est demandé.

3.2.1 Mode d'affichage du jeu

Le programme peut autoriser l'une ou plusieurs des options d'affichage suivantes :

- *Affichage en mode texte.* L'ensemble des informations du jeu (messages d'accueil, succession des tours des joueurs, état du plateau, règles utilisées, messages d'erreur, invites) doit être affiché sur le terminal. On veillera à la clarté des informations affichées. L'orthographe et la syntaxe doivent être soignées.

Ce choix d'implémentation est considéré comme plus facile.

- *Affichage graphique.* L'état du jeu est affiché dans une fenêtre graphique gérée grâce au module *ftk*. Il est demandé de ne pas utiliser un autre module graphique, même pour rendre le jeu plus joli ou plus performant. L'utilisation de *ftk* fait partie des consignes obligatoires du projet.

Ce choix d'implémentation est considéré comme plus difficile.

3.2.2 Mode de saisie des actions des joueurs

Le programme peut autoriser l'une ou plusieurs des options de saisie suivantes ou un mélange des deux. Pour la V1, seule la saisie dans le terminal est demandée.

- *Saisie des actions sur le terminal.* Toutes les actions des joueurs se font en tapant du texte sur le terminal. Les invites doivent être lisibles et faciles à comprendre. Saisir un texte imprévu ne doit pas faire planter le programme. L'orthographe et la syntaxe doivent être soignées.

Ce choix d'implémentation est considéré comme plus facile.

- *Saisie des actions à l'aide de la souris ou du clavier.* Les joueurs interagissent avec le jeu en cliquant sur les objets représentés sur la fenêtre, en pressant des touches du clavier ou en combinant ces deux types d'actions. Les événements sont gérés avec les fonctions fournies par `fltk`. Bien entendu, cela n'a de sens que si le jeu propose une interface graphique.

Ce choix d'implémentation est considéré comme plus difficile.

4 Améliorations et variantes (optionnelles)

Il est demandé de n'implémenter les améliorations et variantes que si vous avez terminé la V2. Si votre V2 ne fonctionne pas, vous n'obtiendrez pas de points avec les améliorations.

4.1 Variante : Positionnement aléatoire des billes

À la place de la phase préliminaire, on peut considérer une variante du jeu où les billes des joueurs sont initialement disposées aléatoirement sur la grille. Implémentez une fonction qui permet de disposer les billes aléatoirement. Dans un premier temps, on peut autoriser à ce que les billes tombent sur des trous (auquel cas le joueur correspondant est désavantagé). Dans un second temps, on s'assurera que les billes tombent sur des cases sans trou.

4.2 Améliorations de l'ergonomie

Une première piste d'amélioration possible consiste à ajouter des éléments d'ergonomie au jeu. On pourra par exemple ajouter des menus graphiques, un habillage, pourquoi pas une animation des déplacements des pions ou d'autres améliorations visuelles.

D'autres améliorations possibles concernent les fonctionnalités du jeu proprement dites. On pourra par exemple ajouter de nouvelles actions liées à des touches du clavier : annulation ou rétablissement des coups précédents, sauvegarde ou chargement de partie (voir le paragraphe suivant), etc.

4.3 Codage et sauvegarde des parties

Comme dans d'autres jeux de stratégie (notamment les dames ou les échecs), il est possible de représenter toute l'information concernant une partie sous une forme symbolique, par exemple en désignant les positions des pions à l'aide de leurs coordonnées. Une première tâche concernant cette perspective d'amélioration consiste à mettre au point une manière de transformer la suite des coups joués et toutes les autres informations importantes en une chaîne de caractères, dans un format approprié.

Une fois ce codage mis au point, il devient possible de sauvegarder l'information d'une partie sous la forme d'un fichier texte. On peut ainsi sauvegarder la partie en cours avant de quitter le jeu, et si nécessaire la reprendre au même point au prochain lancement du programme.

Enfin, une fois ces fonctionnalités mises en place, il est envisageable de permettre de sauvegarder l'ensemble des parties jouées, d'afficher leur liste, des statistiques sur les joueurs, et pourquoi pas de proposer un « replay » de chaque partie coup par coup.

4.4 Jeu contre l'ordinateur

Même si le développement d'une IA complète dépasse largement le cadre de ce projet, les étudiants qui le souhaitent peuvent essayer de concevoir un mode de jeu contre l'ordinateur. Il sera pour cela nécessaire de mettre au point, pour chaque phase de jeu, des critères qui permettent de décider du prochain coup joué par l'ordinateur.

Dans sa version la plus simple, il est envisageable que le joueur-ordinateur joue de manière purement aléatoire (tout en respectant les règles bien entendu). Des règles plus raffinées ou heuristiques peuvent permettre de privilégier certains coups (par exemple ceux qui permettent de faire tomber une bille immédiatement, ou de faire se rapprocher une bille d'un trou). Les étudiants les plus motivés peuvent effectuer des recherches et pourquoi pas implémenter des algorithmes de jeu plus avancés.