

Projet: Bataille des boules.

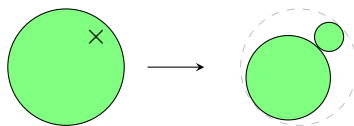
Le but de ce projet est de réaliser un jeu de placement de boules. Chaque utilisateur joue avec une couleur. Le but du jeu est d'occuper la plus grande aire coloriée avec sa couleur. Les joueurs contrôlent la souris chacun à leur tour, et on décide à l'avance du nombre de tours.

Le projet est à réaliser en plusieurs phases. Lisez bien le sujet dans son intégralité avant de commencer le travail, car les choix que vous effectuerez au début auront un impact sur la suite.

1 Le programme à réaliser : le jeu de base

Dans sa version de base, l'aire de jeu est initialement vide. Le jeu montre à qui est le tour. À la fin, il doit indiquer qui a gagné. Vous devrez implémenter le jeu avec les règles suivantes :

1. Les intérieurs de deux boules de couleurs différentes ne peuvent pas s'intersecter. Si un joueur essaye d'inclure une boule qui intersecte l'intérieur d'une boule de la couleur opposée, rien ne se passe et il perd son tour.
2. Si un joueur clique dans l'intérieur d'une boule de l'adversaire, il la transforme en deux boules de même couleur plus petites. Une des deux boules a pour centre l'endroit où l'on a cliqué, et est tangente à la boule d'origine, et l'autre est tangente à la boule d'origine et à la nouvelle boule. Voici un exemple où le \times indique l'emplacement du clic du joueur :



Vous devrez décider comment gérer le bord de l'aire de jeu.

2 Variantes

Dans un deuxième temps, vous devrez fournir une version du jeu avec un menu initial qui permet d'utiliser une combinaison des alternatives suivantes. Les choix d'implémentation sont laissés à votre appréciation.

1. **Sablier** : chaque joueur a un temps prédéterminé pour jouer à chaque tour ; s'il ne réagit pas à temps, il perd son tour.
2. **Scores** : un joueur peut vérifier quelle aire ses boules totalisent à chaque instant en appuyant sur la touche 's'. On affiche le score pour les deux joueurs, et il disparaît après 2 secondes.
3. **Taille des boules** : à chaque tour, on demande au joueur le rayon de la boule qu'il veut introduire. Il commence avec un certain budget fixé au préalable, et pour chaque boule posée, son budget diminue du rayon de la boule qu'il pose.
4. **Version dynamique** : les rayons de toutes les boules s'incrémentent à chaque tour en respectant les règles données (dès que deux boules de couleurs différentes se touchent, elles arrêtent de grandir).
5. **Terminaison** : ajouter une règle permettant à un joueur de décider une fois par partie que le jeu se termine dans 5 tours.
6. **Obstacles** : le tableau commence avec certains obstacles que les boules ne peuvent pas toucher. Vous pouvez décider du type et de la forme des obstacles.

3 Les bonus et améliorations

Finalement, vous devrez implémenter au moins deux des améliorations suivantes :

1. **IA** : ajoutez un mode où l'ordinateur joue tout seul, que l'on peut lancer avec une option spéciale de la ligne de commande. Il peut y avoir un ou deux joueurs virtuels (donc soit on joue contre l'ordinateur, soit l'ordinateur joue contre lui-même). Dans chaque cas, réfléchissez à des stratégies et motivez vos choix.
2. **Classement** : si vous implémentez l'IA, rajoutez de quoi mémoriser / afficher le classement des meilleurs scores.
3. **Mode billard** : dans cette variante, les boules bougent selon les règles d'un billard. La direction initiale de chaque boule doit être aléatoire, quand deux boules de couleurs différentes se touchent elles rebondissent. Quand elles touchent le bord, elles rebondissent aussi. Attention, pour chaque boule, vous allez devoir choisir la vitesse de déplacement, c'est-à-dire définir de combien de pixels elle se déplace entre deux rafraîchissements de l'écran de jeu (c'est le vecteur vitesse). Afin que le jeu soit jouable sans être trop difficile, cette vitesse doit absolument être aisément paramétrable, car l'effet réel peut être très différent d'une machine à l'autre. Vous choisirez donc une vitesse par défaut et laisserez la possibilité à l'utilisateur d'en choisir une autre en paramètre de la ligne de commande. Les boules n'arrêtent jamais de bouger.
4. **Création des obstacles** : rajoutez la possibilité de charger une aire de jeu agrémentée d'obstacles à partir d'un fichier texte contenant leur description et leurs emplacements. Vous choisirez un encodage pour les obstacles et leurs propriétés.
5. **Pause et sauvegarde** : rajoutez la possibilité de mettre en pause le jeu et de reprendre la partie, et de sauvegarder l'état du jeu à ce moment-là, ainsi que la possibilité de charger une partie sauvegardée en début de jeu.
6. **Sauvegarde des paramètres** : rajoutez un fichier de configuration pour le jeu afin de fixer les valeurs des différents paramètres (taille de la fenêtre, de l'aire de jeu, vitesse, ...).

4 Consignes de rendu

Le projet se déroulera en trois grandes phases : un premier rendu aura lieu début novembre 2022, un second rendu fin novembre, et finalement un rendu final début janvier 2023. Pour les rendus, il sera impératif de suivre les consignes précisées ci-dessous, sans quoi vous perdrez des points. En particulier :

- Ce projet est à faire en binôme (s'il est nécessaire de faire une exception, contactez les enseignants). Vous devrez sélectionner votre binôme sur e-learning dans la semaine suivant la publication du sujet.
- **Vous DEVEZ utiliser upemtk**. L'utilisation de modules autres que les modules standards de Python est interdite ; en cas de doute, n'hésitez pas à poser la question.
- **Votre programme devra impérativement s'exécuter sans problème sur les machines de l'IUT**. Prévoyez donc bien de le tester sur ces machines en plus de la vôtre et d'adapter ce qui doit l'être le cas échéant (par exemple, les vitesses de déplacement). Il sera donc utile de permettre à l'utilisateur de préciser certaines options auxquelles vous attribuez des valeurs par défaut plutôt que de devoir modifier le code à chaque exécution.

4.1 Premier rendu

L'objectif général à atteindre pour le premier rendu est d'avoir un jeu fonctionnel et robuste avec :

- Le jeu de base (section 1) complet.
- Au moins une variante de la liste donnée en section 2.

Ne commencez pas les variantes de la section 2 avant d'avoir réalisé le jeu de base.

La date limite pour ce premier rendu est le **jeudi 10 novembre 2022 à 23h55**. Passé ce délai, la note attribuée à votre projet sera 0. Les séances de SAÉ de la semaine suivante seront consacrées à la vérification de votre rendu et à amorcer la seconde partie. Vous déposerez sur la plate-forme e-learning une archive contenant :

1. **les sources de votre projet**, commentées de manière pertinente (quand le code s'y prête, pensez à écrire les doctests). De plus :
 - les fonctions doivent avoir une longueur raisonnable ; n'hésitez pas à morceler votre code en plusieurs fonctions auxiliaires pour y parvenir ;
 - plus généralement, le code doit être facile à comprendre : utilisez des noms de variables parlants, limitez le nombre de tests, et veillez à simplifier vos conditions ;
 - quand cela se justifie, regroupez les fonctions par thème dans un même module ;
 - chaque fonction et chaque module doit posséder sa *docstring* associée, dans laquelle vous expliquerez leur fonctionnement, et pour les fonctions, ce que sont les paramètres ainsi que les hypothèses faites à leur sujet.
2. un **fichier texte** `README.txt` expliquant :
 - les bonus qui ont été implémentés (au moins un) ;
 - l'organisation du programme ;
 - les choix techniques ; et
 - les éventuels problèmes rencontrés.

Vous pouvez y ajouter tous les commentaires ou remarques que vous jugez nécessaires à la compréhension de votre code.

Si le code ne s'exécute pas, la note de votre projet sera 0. Vous perdrez des points si les consignes ne sont pas respectées, et il va sans dire que si deux projets trop similaires sont rendus par 2 binômes différents, la note de ces projets sera 0.

4.2 Second rendu

L'objectif pour le second rendu est toujours d'avoir un jeu fonctionnel et robuste avec le jeu de base et toutes les variantes.

Les consignes données pour le premier rendu restent valables pour le second rendu. La date limite pour le second rendu est le **jeudi 8 décembre 2022 à 23h55**.

4.3 Troisième rendu

L'objectif est toujours d'avoir un jeu fonctionnel et robuste avec le jeu de base et toutes les variantes et au moins deux bonus de la section 3

La date limite pour le second rendu est le **jeudi 5 janvier 2023 à 23h55**. Des mini-soutenances seront organisées la semaine suivante, au cours desquelles vous ferez une démonstration sur une machine des salles de TP et serez interrogés sur le projet (les choix techniques et algorithmiques que vous avez faits, les difficultés rencontrées, l'organisation de votre travail, ...). Les contributions de chaque participant seront évaluées, et il est donc possible que tous les participants d'un même groupe n'aient pas la même note.