

Programmation

Travail d'évaluation

L'évaluation de l'UE programmation se fera via un travail individuel à réaliser et à défendre oralement.

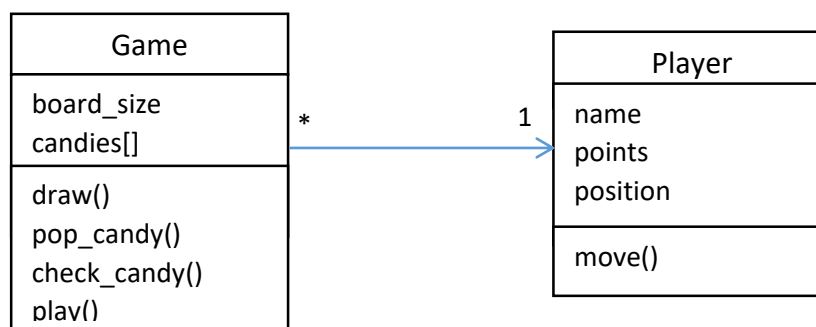
Le travail consiste à améliorer un jeu qui vous est donné. Il va de soi que le travail doit être personnel et ne doit plagier aucun autre travail.

La défense nous permettra de vérifier que c'est bien vous qui avez travaillé. Vous devrez être capable d'expliquer n'importe quelle ligne de code (y compris dans le code qui vous est donné au départ), de justifier vos choix et de discuter d'améliorations possibles.

Enoncé du jeu

Il s'agit d'un petit jeu de plateau (de 10 cases par défaut) sur lequel un joueur se déplace et doit récupérer des bonbons qui apparaissent aléatoirement. Le joueur a une minute pour en récolter le plus possible.

Voici le diagramme UML du jeu qui vous est fourni :



Le code python correspondant au jeu est disponible sur moodle sous le nom « base_travail.py » dans la section « Projet ».

Dans un premier temps, il vous est fortement conseillé de jeter un œil au code et de tester le jeu pour bien le comprendre.

Consignes du travail

Le jeu de base vaut **5 points sur 20**. Chaque amélioration apportée au jeu peut vous rapporter des points. La valeur de votre jeu final définira la note de base avant la défense orale, note qui sera revue à la hausse ou à la baisse suite à cette défense¹.

Voici une liste d'idées améliorations possibles et le nombre de points qu'elles peuvent vous rapporter.

- Empêcher le joueur de revenir en arrière [1]
- Ajouter un second joueur [3]
- Ajouter des ennemis qui se déplacent aléatoirement [2]
- Donner des valeurs différentes aux bonbons récoltés (éventuellement malus) [1]
- Ajouter des bonbons bonus (par exemple faire apparaître pleins de bonbons) [2]
- Encapsuler les classes [1]
- Utiliser des exceptions [2]
- Permettre de mettre le jeu en pause [3]
- Améliorer les ennemis en leur donnant un peu d'intelligence (IA) [2]
- Proposer un niveau de difficulté différent (plus de temps, des pénalités...) [2]
- Empêcher de sortir du plateau [1]
- Permettre de passer à travers les murs et de réapparaître de l'autre côté [2]
- Enregistrer les tops joueurs [1] (dans un fichier [3])
- Améliorer le jeu pour en faire un snake [4]

Il ne s'agit ici que d'idées pour ceux qui n'en auraient pas mais vous êtes libre de mettre en place vos propres améliorations ou de pousser ces idées un peu plus loin. De plus, les points indiqués le sont à titre indicatifs. En effet, si vous ne réalisez pas l'amélioration totalement ou qu'elle aurait pu être mieux faite, vous n'aurez peut-être pas la totalité des points. Inversement, si vous poussez l'amélioration à son maximum (voir au-delà), vous aurez peut-être plus de points. De plus, certaines améliorations combinées amèneront peut-être des points bonus.

Il est important de noter que **le jeu final doit être fonctionnel** ! Il vaut donc mieux faire moins mais le faire bien que de faire trop et que le jeu ne fonctionne plus.

Remarque : *pour ajouter des fonctionnalités, vous pouvez modifier certaines parties du code qui vous a été fourni si cela se justifie.*

Besoin d'aide ?

Nous restons disponibles par mail en cas de besoin. Cependant, nous ne corrigerons ni n'écrirons de code à votre place. Vous pouvez nous poser des questions précises sur un morceau de code sur lequel vous êtes bloqué et nous vous aiderons à comprendre votre erreur. Si la question n'est pas assez précise, nous vous le ferons savoir.

¹ Un travail réussi ne garantit pas une réussite au final ! En effet en cas d'oral médiocre, votre note pourra être vue à la baisse au point de vous mettre en échec.