



**NOM DU PROJET : Casino Royal**

## > PRÉSENTATION GÉNÉRALE :

*Ce projet est entièrement codé par Marie HOIZEY—CANGIOLONI, Hugo DETHOREY et Xinyu LI, élèves de 1<sup>ère</sup>8 au lycée Marc Chagall, à Reims.*

Ce projet est basé sur l'utilisation de la bibliothèque turtle en langage Python. Avec les critères imposés en classe, nous sommes parvenus à l'idée d'un casino qui proposerait plusieurs jeux interactifs. Pour cela, nous avons utilisé la bibliothèque random.

Le casino est composé de deux jeux :

Le jeu de dés dans lequel le joueur affronte l'ordinateur. Chaque participant possède deux dés. Le participant ayant la plus grande somme de dés gagne la partie. Les règles de ce jeu s'affichent avant le lancement du jeu.

La bataille de cartes constituée de plusieurs rounds au choix (max 10). Le joueur joue contre l'ordinateur. Chaque participant possède 26 cartes au début de chaque partie. Le participant possédant le plus de cartes à la fin des rounds gagne.

Les jeux du casino fonctionnent avec un système de mise. La somme de départ est de 50£. Le joueur choisit la somme à miser. S'il gagne, cet argent sera doublé. S'il perd, cet argent sera retiré. S'il y a égalité, l'argent est conservé. Lorsque le joueur ne possède plus d'argent, il est contraint à quitter le casino.

Le joueur peut changer de jeux à la fin de chaque partie en repassant par l'accueil avec un bouton *home* qui permet aussi de quitter le casino directement.

## > ORGANISATION DU TRAVAIL :

Pour créer ce projet, nous avons constitué un groupe de trois personnes : Marie, Hugo et Xinyu.

Xinyu s'est occupé de la modélisation du paysage dont la page d'accueil et le fond des jeux (en plus de la machine à sous qui n'a pas été intégrée dans le casino car le jeu n'a pas été entièrement développé par manque de temps). Elle a aussi codé le programme aléatoire des dés dans le jeu de dés.

Marie s'est occupée en partie de la modélisation des dés et des cartes. Elle a aussi modélisé les règles du jeu pour le jeu de dés.

Hugo a modélisé le système de mise, les lettres qui annoncent si le joueur a gagné ou non, l'animation des dés et des cartes, le bouton d'accueil et l'animation qui permet de défiler les règles du jeu en appuyant sur la barre espace. Il s'est aussi occupé de la mise en commun des bouts de programmes de chaque membre du groupe.

Le travail a été effectué la plupart du temps individuellement. Les informations sont communiquées sur un groupe Instagram et les documents à partager ont été envoyés par l'ENT (code, images, etc.)

Nous avons surtout travaillé pendant les vacances. Des réunions ont été faites pour des mises en communs importantes en extérieur, ainsi que pendant nos cours de NSI. La fréquence de travail était très élevée au début du projet. Nous avons enchaîné pendant des heures (jusqu'à 10 heures par jour) et parfois avons pris tout le weekend

pour travailler sur le projet. En revanche, le travail a été mis de côté davantage pendant les jours de la semaine et pendant quelques semaines en raison des cours.

## **> LES ÉTAPES DU PROJET :**

L'idée initiale était de faire un jeu de dés dans un casino, Xinyu a commencé par créer le fond avec les fleurs de lys et le programme aléatoire qui fera tourner les dés. Pendant ce temps, Hugo a épluché la documentation python pour pouvoir ensuite créer le jeu. Marie s'est alors chargée de la modélisation des dés, ainsi que l'animation qui les font trembler et change leurs chiffres. Hugo a alors modélisé le système de mise qui permet de gérer l'argent parié et de générer les lettres « win », « lost » et « draw ».

Nous avons dans un deuxième temps modélisé l'accueil et les fonctionnalités qui permettent d'accéder aux jeux. En parallèle nous avons modélisé les règles du jeu pour le jeu de dés.

Dans un troisième temps, nous avons créé le deuxième jeu, la bataille. Nous avons repris le fond du jeu de dés et avons créé le jeu de cartes en se basant partiellement sur les programmes d'avant.

Ensuite nous avons modélisé les cartes nous-mêmes. Elles ont ensuite été converties en gif pour pouvoir les insérer dans le code. Chaque carte étant associées à une valeur, nous avons pu réaliser une vraie bataille. Chaque élément de notre code est authentique, nous n'avons en effet rien copié d'internet.

## **> FONCTIONNEMENT ET OPÉRATIONNALITÉ :**

Le Casino propose jusqu'à maintenant deux jeux complets. Un troisième jeu est en cours de programmation : la machine à sous. Le fond a été programmé et il nous reste à ajouter les cartes et les autres images, l'animation des images et le système de mise. Nous ajouterons ensuite les pages de règles du jeu. Après cette étape, nous programmerons éventuellement de nouveaux jeux tels que la roulette ou le Black Jack. Nous créerons aussi une nouvelle interface sur laquelle nous pourrions directement cliquer sur le jeu souhaité, ce qui remplacerait l'input de la page d'accueil.

Des « print » ont été ajoutés aux points clés du programme permettant ainsi de vérifier les valeurs et la réaction apportée du programme en temps réel, et donc de voir si le programme agissait comme il fallait. Les programmes ont d'abord été pensés comme des jeux textuels plutôt que visuel, ils peuvent donc fonctionner sans interface, il a alors suffi d'y rajouter la modélisation et toutes les commandes du module turtle de python.

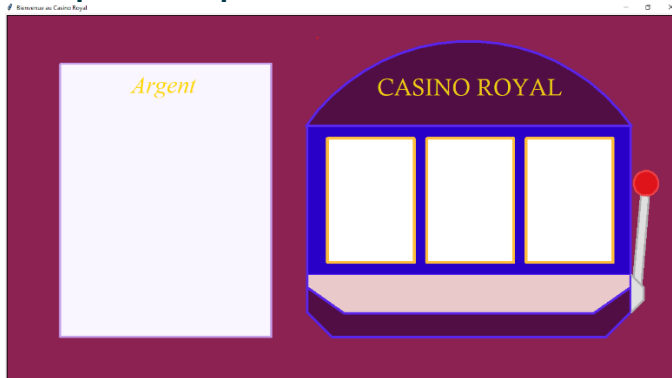
Les difficultés rencontrées lors de la programmation sont :

- La découverte et l'adaptation au langage python et de la bibliothèque python. Nous avons dû nous approprier la bibliothèque turtle avec les différentes commandes et les tester.
- Il a aussi été compliqué pour le jeu de cartes de permettre aux deux joueurs de piocher dans leur propre jeu de cartes plutôt que de piocher dans le paquet entier.

- La mise en commun de nos différentes parties a été assez compliquée à réaliser, étant donné qu'au début nous n'étions pas très à l'aise avec Python. Cependant, nos cours de NSI nous ont permis de partager nos difficultés et de trouver des solutions ensemble.
- De plus, les mesures d'angles et de longueurs nous ont parfois fait obstacle. Nous avons donc décidé d'effectuer nos modélisations sur papier millimétré et ensuite de convertir les mesures en pixel.
- La plus grande difficulté était que la commande « `clearscreen()` » supprime tout dessin à l'écran mais aussi toutes les tortues, les fonctions de dessin ne pouvaient donc être placées qu'à l'intérieur d'autres fonctions, puisqu'on ne peut pas simplement charger les tortues au tout début du programme car elle se supprimeraient au moindre « `clearscreen()` »

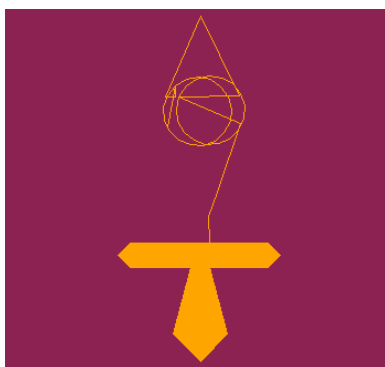
## > OUVERTURE :

Une machine à sous aurait dû être implémentée mais elle n'a pas vu le jour par manque de temps.



Un court-métrage a été tourné, il rend ainsi le projet plus attractif et peut convenir à tous les âges. Ce projet peut simplement servir de passe temps mais aussi permettre d'analyser la manière que chacun a d'économiser ou de dépenser face à un jeu d'argent.

Une fois notre projet complètement achevé (même s'il restera toujours des choses à améliorer), nous souhaiterions le partager aux élèves et professeurs de notre lycée. Ce dernier organise souvent de nombreux concours de dessin, de photographie ... Alors pourquoi pas un concours d'idées. En effet, nous aimerions organiser cela afin que les élèves puissent proposer d'autres jeux de casino qu'ils auront codés. Le meilleur jeu pourrait par la suite être implémenté à notre Casino Royal.



Pour la programmation de la fleur de lys, nous ne nous sommes pas assez informés et avons pris plus de temps pour la programmer et le résultat n'est pas tout à fait symétrique. En effet, nous ne savions pas que l'utilisation d'un nombre négatif pour l'étendue dans la commande « `turtle.circle()` » permettrait de dessiner le cercle dans le sens horaire, ce qui aurait permis une symétrie plus cohérente au niveau de la fleur de lys. Ce manque d'information nous a fait perdre du temps car nous avons dû tester des mesures d'angles pour que le cercle dessiné soit à peu près symétrique à celui dessiné au préalable.

## DOCUMENTATION

- Télécharger le fichier contenant le fichier python et les 53 .gif (ne rien modifier dans le fichier)
- Exécuter le fichier python intitulé « Casino Royal Hugo D, Marie H, Xinyu L.py » puis exécuter le programme (ctrl+f9 sur edupython)
- Cliquer sur la porte et suivre les instructions apparaissant dans la case en haut à gauche de la fenêtre turtle.

Si vous avez entré « card » :

- Cliquer sur le carré vert en bas à gauche de l'écran pour lancer une partie.
- Sélectionner ensuite le nombre de round(s) que vous désirez (l'input modifie le nombre de fois où les joueurs piochent. Comparer leurs cartes avant que le jeu ne s'arrête) puis la mise que vous mettez en jeu.
- Attendre la fin de l'animation pour recevoir le double de votre mise si vous avez gagné.
- Pour revenir à l'accueil, cliquer sur le bouton en forme de maison en haut à gauche de la fenêtre turtle.

Si vous avez entré « dice » :

- Appuyer sur la barre espace après avoir lu chaque page de règles.
- Cliquer sur le carré vert en bas à gauche de l'écran pour lancer une partie.
- Indiquer la mise que vous voulez mettre en jeu.
- Attendre la fin de l'animation, si les deux dès du bas ont une somme supérieure à ceux du haut, vous recevez le double de votre mise.
- Pour revenir à l'accueil, cliquer sur le bouton en forme de maison en haut à gauche de la fenêtre turtle.

Vous pouvez à tout moment quitter le programme en écrivant « quit » après avoir cliqué sur un bouton maison ou sur la porte de l'accueil.

Le programme est organisé en différentes fonctions, chacune gérant une page différente du programme, ainsi il existe une fonction pour la page d'accueil, une autre pour la page de règles et deux autres pour les deux jeux. Chaque grosse fonction contient de plus petites fonctions et toutes les tortues dont elle a besoin pour fonctionner. Une grosse fonction est toujours quittée à l'aide d'un « clearscreen() » supprimant ainsi les dessins et les tortues associés.

Pour le jeu de batailles, le nom des cartes est stocké par famille dans 4 listes. 4 autres listes servent à définir quel joueur les possède.

Une carte étant notée « True » appartient au joueur 1, les cartes « False » appartiennent à l'ordinateur.

```
o Coeur=("2C.gif","3C.gif","4C.gif","5C.gif","6C.gif","7C.gif","8C.gif","9C.gif","10C.gif","VC.gif","QC.gif","RC.gif","1C.gif")
o CoeurIf=[True,True,True,True,True,True,True,True,True,True,True,True,True]

o Carreau=("2K.gif","3K.gif","4K.gif","5K.gif","6K.gif","7K.gif","8K.gif","9K.gif","10K.gif","VK.gif","QK.gif","RK.gif","1K.gif")
o CarreauIf=[True,True,True,True,True,True,True,True,True,True,True,True,True]

o Pic=("2P.gif","3P.gif","4P.gif","5P.gif","6P.gif","7P.gif","8P.gif","9P.gif","10P.gif","VP.gif","QP.gif","RP.gif","1P.gif")
o PicIf=[True,True,True,True,True,True,True,True,True,True,True,True,True]

o Trefle=("2T.gif","3T.gif","4T.gif","5T.gif","6T.gif","7T.gif","8T.gif","9T.gif","10T.gif","VT.gif","QT.gif","RT.gif","1T.gif")
o TrefleIf=[True,True,True,True,True,True,True,True,True,True,True,True,True]
```

Les cartes sont ensuite distribuées au début du jeu de bataille.

Le système de pioche va chercher aléatoirement une carte selon sa famille et sa valeur, si elle est notée comme « False » alors que le joueur 1 la tire, le programme en choisit une autre, et ainsi de suite jusqu'à tomber sur une carte « True ». (Vice versa pour l'ordinateur)

**Pour le jeu de dé, la valeur de chaque dé est déterminée avant l'animation, les dés se contentent alors d'afficher la valeur qui leur est attribuée. Le résultat se calcule simplement en comparant la somme des valeurs définies au départ.**