

Roulette avancé :

Le projet :

Le projet roulette est un projet où l'utilisateur peut jouer à un jeu de roulette, inspiré des casinos et suivant un énoncé très précis :

- L'utilisateur fixe la quantité d'argent disponible
- Il choisit un numéro sur lequel il parie une mise
- L'ordinateur lance la roulette et affiche les résultats (si les chiffres sont égaux, il remporte la mise * 3, si ils sont tous les 2 pairs ou impairs, il remporte la mise * 1.5, sinon il ne remporte rien)
- Il peut soit continuer à jouer soit quitter (sauf si il est ruiné, ou il ne peut que quitter)
- Il peut voir à la fin la quantité d'argent qui lui reste ainsi que ces bénéfices/pertes

Choix de programmations :

Au début du programme, les différents modules utilisés sont importés. Il s'agit de math, pygame, rouletteModule (module du projet sans fenêtre graphique), sys, time et une deuxième fois pygame. Cette deuxième importation de pygame permet de lancer le programme pour la fenêtre dès la ligne 8. A la ligne 11, une fonction nommée strToArray permettant de transformer une chaîne de caractères donnée en paramètre est définie. Elle retourne s sous forme de tableau. A la ligne 14, une fonction f est définie. Elle permet de renvoyer à partir d'un nombre x l'image sur une fonction formant des courbes très régulières est ressemblant à des cercles. Leur représentation graphique est proche de celle des fonctions sinusoïdales. La taille de la fenêtre est définie dans un tuple à la ligne 30. Le numéro de la page est défini dans une variable ligne 40. La pagination est un système permettant d'afficher différents éléments sur la fenêtre selon leur page. Elle peut prendre plusieurs valeurs : -1 (page de paramètre non définie pour l'instant), 0 (page d'accueil), 1 (page où rentrer la quantité d'argent disponible), 2 (page où le joueur choisit son numéro et sa mise), 2.3 (pareil que la page 2 mais après que la roulette soit tirée), 2.6 (pareil que la page 2.3 mais après que la roulette est finie de tourner et au moment de l'actualisation de l'argent du joueur), 2.65 (pareil que 2.6 mais sans actualisation d'argent) et 3.0 (page d'au revoir où sont affichés les résultats. Ensuite, à la ligne 44, l'instance de la fenêtre est créée et stockée dans une variable fenetre. A la ligne 47, l'angle de la roulette est stocké. Il permet de savoir comment la roulette est tournée. A la ligne 49, l'angle de la roulette depuis le début de la rotation est stocké. Il permet de savoir depuis quand la roulette tourne. Ensuite, ligne 51 est défini l'angle limite de la roulette, pour ne pas qu'elle tourne indéfiniment. Il est défini à l'équivalent de 15 tours. Ensuite, ligne 53, l'angle de rotation réel de la roulette est défini. En effet, l'angle de rotation réel sera légèrement différent de l'angle de rotation limite car il faut l'aligner avec le bon chiffre choisi aléatoirement. L'image d'arrière plan est définie à la ligne 55, dans une variable arrierePlan. L'image est chargée dans le dossier assets à assets/arrierePlan.png. Elle a été réalisée grâce à Pixlr, logiciel gratuit de retouche photo, avec un dégradé de rouge qui tend

vers du jaune pour un aspects de richesse. Ensuite ligne 57 est défini le `deltaTime`, ou Δtime . Il s'agit du temps qu'a duré la dernière itération de la boucle `while` principale. Ensuite ligne 59 est défini le `deltaTimeConcret`, qui correspond au temps de `deltaTime` réel, soit de celui de l'itération de boucle qui a précédé celle qui se passe au moment de l'acquisition de `deltaTime`. Ensuite, la variable `elementsGUI` est défini ligne 61. Il s'agit d'un dictionnaire où se trouve tout les éléments appartenant à la fenêtre graphique. La clé est leur nom et la valeur l'image chargé dans le dossier `assets`. Les boutons sont stockés dans le dossier `assets/boutons`. Différents booléens correspondants au focus des différentes interfaces d'entrées sont défini dès la ligne 78. La largeur des traits de la roulette est défini à la ligne 82. La rotation du bouton paramètres (qui évolue lors du focus) est stocké dans une variable défini ligne 84. La vitesse de cette rotation est stocké dans une variable constante défini ligne 86 est initialisé à 0,5 tours par secondes. A la ligne 88 est défini la variable qui contient la taille de police des textes chiffres de la roulette. Pour une roulette à 37 cases, une police de 8 est conseillé. Pour 17 case, une police de 24 est conseillé. Les différents `rect`, c'est à dire tuples contenant la position et taille, des interfaces d'entrée sont définis ligne 90. Les tailles maximales de texte dans ces interfaces d'entrées sont définis ligne 94. Le temps écoulé entre la page 2.6 et 2.65 est défini ligne 98. Le temps que le trait de saisis des interfaces d'entrées sera affichés sont stockés ligne 100. Le temps depuis lequel le trait est visible/invisible est stocké ligne 104. Le texte contenu dans les interfaces est stockés ligne 108. La visibilité des traits de saisis des interfaces d'entrées sont stockés dans des booléens défini ligne 112. La vitesse de rotation de la roulette est défini ligne 116 et la coordonnées y du bouton valider de la page 1 est stocké ligne 118. Nous pouvons remarquer beaucoup de variables nécessaire au fonctionnement des interfaces d'entrées. Elles sont du au fait qu'elles ont besoin de beaucoup de ressources pour fonctionner. La police est défini ligne 120. Cependant, comme une grande partie des variables énumérer ici, elle n'est pas une constante et est voué à être modifié plus tard dans le code. La boucle infini démarre dès la ligne 123. A la ligne 125, `deltaTime` est actualiser au temps en nanosecondes depuis bien longtemps. Cela permet de préparer l'établissement plus précis d'un vrai `deltaTime` à la fin de la boucle. La police est ré-initialisé ligne 127 pour palier de potentiel modification durant la boucle précédente. La position de la souris est obtenu ligne 130 et le curseur final est défini ligne 132. Il est fait pour être modifié plus tard.

A la ligne 134 début la boucle de vérification des évènement `pygame`. Si un évènement de type `quitter` est détecter (clique sur la croix), le programme est fermé. Si un évènement de type `click` est détecter, selon la page actuel, les variables de focus vont d'abord être mis à faux pour ne pas fausser les focus des différentes interfaces, puis, selon la page et la coordonnée de la souris, le programme va savoir sur quel bouton l'utilisateur a cliqué. Si il s'agit du bouton jouer de la page 0, alors la page est mis à 1 pour signifier que l'utilisateur passe à la page de rentrée d'argent pour commencer à jouer. Si le bouton cliqué est le bouton paramètres, alors la page est mise à -1. Comme rien n'est défini pour cette page, le programme ne va rien afficher sur la fenêtre, sauf les éléments récurrents à toutes les pages. Si le joueur click sur le bouton valider de la page 1, alors l'argent du module `rouletteModule` est actualiser à l'argent entrée par l'utilisateur. Cependant, l'utilisateur doit avoir entrée une valeur qui peut être converti en nombre (décimal) pour fonctionner. Sinon il ne se passe rien. Si tout est bon, alors les variables `argent`, `argentDebut` (qui ne sera modifié que ici) et `page`

sont modifié. La page devient 2 pour permettre à l'utilisateur de rentrer sa mise et la case qu'il choisi. Si l'interface d'entrée d'argent est clické, le focus est mis sur elle pour que l'utilisateur puisse modifier son contenu. Si l'une des interfaces de la page 2 est clické, alors le focus est mise sur cette dernière pour pouvoir modifié son contenu. Si le bouton lancer est clické, alors l'argent est actualiser pour que la mise soit soustrait, tout les paramètres de la roulette sont ré-initialisé et un nombre au hasard est choisis. La variable de l'angle de rotation limite réel est défini là selon le nombre choisis pour que la roulette choisisse ce nombre en particulier. La page passe à 2.3 pour dire que la roulette a été tiré. Si le bouton rejouer de la page 2.6 ou 2.65 est clické, alors la page est remis à 0 est le contenu de la mise et de la case est vidé pour ne pas créer de potentiel erreur. Si il click sur arrêter, alors la page est mise à 3 et les résultats sont affichés.

Si un évènement de type touche de clavier pressé est détecter, alors le programme va pouvoir identifier quel éléments va être modifié selon le focus et la page. Si une interface d'entrée est focus et qu'il lui reste de la place dans son texte, alors l'évènement va lui être appliqué. Si le nombre est un chiffre, alors il va simplement l'ajouter au texte de l'interface en question. Si il s'agit de la touche pour supprimer un caractère, alors le dernier caractère de l'interface est supprimé (si l'interface à plus de 1 caractère). Il est transformé en tableau de caractère via `strToArray`, son dernier élément est supprimé et le tout est join pour formé un str. Si l'interface est celle de l'entrée d'argent et que la touche entrer est pressé, alors ce passera les mêmes actions que si le bouton valider de la page 1 était clické. Dans le cas de l'interface d'entrée argent et mise, alors des nombres décimaux peuvent être rentré. Cependant, seul le point fonctionne est il ne sera rentrée que si l'interface ne compte aucun autre point et n'est pas vide.

Les éléments récurrents vont ensuite être afficher sur la fenêtre dès la ligne 281. Premièrement, l'arrière plan puis le titre son affiché. Le titre est un texte provenant de Pixlr, comme tout les éléments de la GUI, en dégrade bleu et jaune pour représenter la couleur de l'euro. Un texte est généra via la police ligne 284 pour faire l'éloge de moi et de Léo. Les affichages par page commencent dès la ligne 287.

Pour la page 0, le bouton jouer est affiché. Tout les boutons sont présents en 2 exemplaires : une version focus et une version non-focus. Le programme va calculer la position de la souris pour voir si le bouton est focus ou non. La différence entre les 2 versions sont le curseur et la taille du texte. Cela ne sera pas précisé partout mais ce système est identique pour TOUT les boutons présents dans le programme. Le bouton de paramètres est aussi affiché avec l'animation de rotation si nécessaire. L'animation est géré par la variable de statue de rotation paramètre et la vitesse de rotation paramètres, qui elle même dépend du `deltaTime` concret pour donner une animation fluide.

Pour la page 1, un texte pour indique au joueur qu'il doit rentrer sont argent est généré via la police. L'interface d'entrée d'argent est aussi dessiné. Elle est composé de 2 rectangles les 2 dans les autres, un blanc et un noir. Dans le rectangle blanc, le contenu de l'interface est affiché via la fonction `render` de la police. Le trait est aussi dessiné après le texte selon la présence du focus ou non. De même, ce ne sera pas précisé, mais ce système est le même pour toutes les interfaces présentes dans le programme. Le bouton valider est affiché en dessous pour valider l'argent.

Pour la page 2, l'interface d'entrée de la mise et de la case sont affichés. Elles sont tout le temps visible. Si la page est exactement à 2, alors un bouton lancer est placé pour lancer la roulette. Si la page est à 2.3, alors l'animation de la roulette est joué. La roulette est dessiné grâce au fonctions trigonométriques cos et sin. Elles permettent d'anticiper le placement des traits et de faire quelque chose de propre. Les traits font la taille défini selon la variable de la largeur des traits. Le texte est aussi affiché grâce à la même méthode, même si le tout est inversé par rapport à celui du calcul des traits, pour ne pas que le texte sortent du rond. Lors de la rotation, on a juste rajouté la valeur de l'angle de la roulette dans les fonctions cos et sin pour faire une rotation fluide. Quand la rotation est fini, soit que l'angle de la roulette est égal à l'angle limite - 0.25 (et pas à l'angle limite exactement car il tend vers 0 mais ne l'atteint pas vraiment avant longtemps), alors la page passe à 2.6 et l'argent est actualiser selon la case et la mise. La mise totale est affiché ainsi que le nombre tiré est les relations avec celui choisi. Pour ne pas que ce procédé est lieu indéfiniment, la page passe à 2.65 et les boutons rejouer et arrêter apparaissent. Si le joueur continu, la page passe à 2 et le cycle recommence. Sinon, la page passe à 3.

Pour la page 3, l'argent total de l'utilisateur, les gains ou pertes et un petit commentaires sont affichées. Un message de remerciement est aussi affiché et un bouton quitter permet de quitter le jeu.

Après tout ça, le curseur final est assigné ligne 503 selon la variable curseur modifié ou pas dans la boucle, l'écran est actualiser ligne 506 et deltaTime est mis à jour selon le temps qu'a pris tout ce processus. DeltaTimeConcret est défini à ce moment là pour être utiliser dans la prochaine itération de la boucle.

Difficultés rencontrées :

Au début, je voulais utiliser tkinter pour faire la fenêtre graphique. Ayant lu vite fait la documentation de tkinter, une chose ne me plaisait pas. En effet, pour utiliser tkinter, il faut utiliser des canvas/widgets, qui sont beaucoup plus puissant avec des objets. Cependant, ma connaissances des classes python est assez limité contrairement au classes c++ et c#. J'ai donc choisis d'utiliser pygame, qui ne nécessite pas vraiment de classe pour fonctionner correctement. Ensuite, lors de la génération de l'image de la roulette, il fallait calculer le point pour placer le trait et le texte sur l'image. J'aurai pu utiliser une image fixe, mais j'ai décidé de créer une image pour pouvoir utiliser une roulette avec des nombres de cases différents. J'ai premièrement voulu utiliser des fonctions trigonométriques mal exploitées, puis une fonction créant un demi cercle parfait ($\sqrt{x^2 - r^2}$ ou r est le rayon du demi cercle), puis le théorème de Pythagore, avant de simplifier avec de la trigonométrie, qui permet d'utiliser la rotation sans difficultés. Une autre difficultés est l'utilisation des radians dont les fonctions mathématiques de la bibliothèque math. Étant habituer au degrés « normaux » ça a fait un choc.

Améliorations possibles :

L'amélioration la plus flagrante possible est l'ajout de la page -1, soit des paramètres, qui ne sont pas disponibles. Dans cette version, certains paramètres comme le nombre de case, les gains... peuvent être configurés. De plus, une partie règles peut aussi être appréciable dans la page d'accueil. Une amélioration de la texture des boutons est aussi envisageable, ainsi que de celle de la roulette. Le triangle dessiné sous la roulette peut aussi être animé pour augmenter le réalisme. Les interfaces d'entrées peuvent aussi bénéficier d'une ligne de saisie active, qui peut bouger selon le désir de l'utilisateur.