Galli Evan
Lecard Maxence
Groupe 5

Rapport projet : Memory

Répartition et organisation du travail :

Nous avons utilisé l'IDE Visual studio 2022 afin de profiter de la richesse de ses outils et de son confort d'utilisation.

Nous avons aussi utilisé GitHub afin de nous faciliter la tâche en termes de communication du code et notamment pour le suivi du travail grâce à un historique nous permettant de suivre les modifications apportées par notre binôme.

Nous avons aussi essayé de respecter au maximum les conventions de code, nous avons pour cela utilisé pylint, un outil de vérification de code qui nous a rapporté les changements nécessaires. Pylint attribue à notre projet une note de 9.48 / 10

Pour la répartition du travail, Evan s'est occupé du système de jeu, alors que Maxence s'est occupé du décor. Nous avons aménagé des horaires de travail pendant les vacances, notamment par appels discord.

Bien que ce rapport soit destiné à apporter des explications quant à notre programme, le code contient de nombreux commentaires afin d'expliquer plus en détail certains passages.

Thème graphique et niveaux de difficulté :

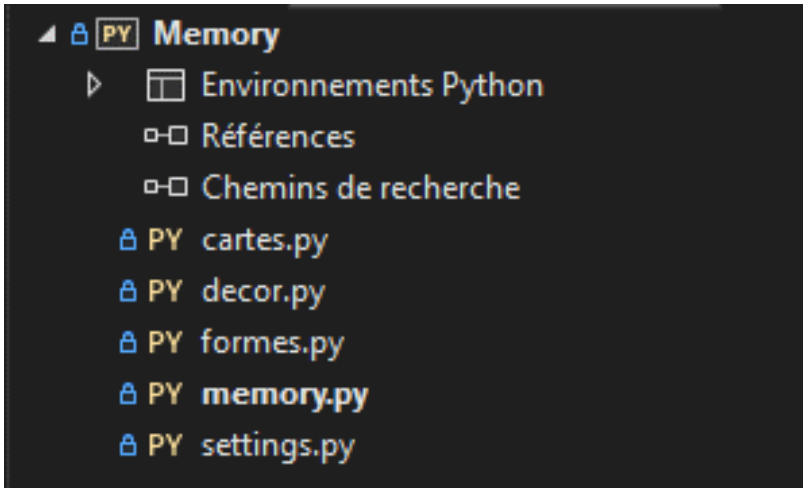
Nous avons choisi le niveaux 2+ pour le projet, l'utilisateur peut donc cliquer sur l'objet de son choix. Nous avons aussi implémenté une barre de progression permettant au joueur de visualiser le nombre de tentatives restantes (nous reviendrons plus en détail sur ce point).
Le thème de notre jeu est l'hiver.
Concernant les éléments du décor nous avons mis deux bonhommes de neige au premier plan et les éléments du décor qui se répètent sont les étoiles, celles-ci scintillent et sont positionnées aléatoirement à chaque redémarrage du jeu. **Les éléments graphiques comme le reste du système de jeu ont été créé par nos soins.**

Règle du jeu :

Le joueur clique une première fois pour retourner la carte, puis il clique sur une seconde carte. Si les deux cartes ont la même forme et la même couleur alors le couple de carte reste retourné, (le cas échéant, les cartes se retournent après un délai d'une seconde). On continue ainsi de suite jusqu'à avoir trouvé toutes les paires de la partie.
Il y a un nombre maximum d'essai qui dépend du nombre de cartes dans la partie (il correspond au double du nombre de couple de cartes), on peut visualiser ce nombre d'essai grâce à la barre de progression située au milieu en bas de l'écran, au-delà de ce nombre d'essai, la partie s'arrête, et on a perdu.

Structure du code :

Notre projet est structuré de la façon suivante :
Nous avons le code principal (memory) qui appelle la fonction main du décor ainsi que l'ensemble des fonctions nécessaires au fonctionnement du jeu.
Nous allons voir un peu plus en détail les fonctions et les différents modules de notre projet.



Programme principale :

Le programme principal se trouve dans le fichier memory.

Au début de notre programme, nous importons les modules et les librairies dont nous besoin. Nous avons désactivé le redimensionnement de la fenêtre de jeu car celle-ci nous aurait posé problème pour le positionnement du décor. Nous nous sommes servis de trois tortues différentes pour tracer le décor, les cases et la barre de progression. C'est aussi dans ce programme que nous appelons la fonction main qui trace le décor.

Génération aléatoire des paires de cartes :

Le nombre de carte est déterminée à partir du nombre de colonnes et de lignes de la grille.
Le code génère une liste de l'ensemble des couples forme/couleur.
La génération aléatoire privilégie les couples dont la couleur et la forme ne sont jamais sortis (pour éviter de jouer avec des objets qui sont tous de même couleur ou tous de la même forme). Puis on sélectionne parmi les couples forme/couleur non tirée le nombre manquant de couples.
Nous avons alors généré l'ensemble des couples forme/couleur, il nous suffit de dupliquer la liste obtenue pour obtenir des paires de cartes. Enfin on mélange la liste de manière à répartir les cartes sur la grille.

Le jeu :

Le programme de jeu est contenu dans une fonction nommée `clickCases`.
On définit la fonction `clickCases` qui est appelée lorsque l'utilisateur effectue un clic.
Elle détecte tout d'abord si le clic s'est produit sur une carte et qui plus est non retournée. Si ce n'est pas le cas, elle s'arrête selle révèle le contenu de la carte.
Elle est aussi en charge de la vérification des couples : Une fois un couple sélectionné, on attend que le contenu soit de nouveau caché avant d'effectuer d'autre commande, si la case est déjà retournée ou que le joueur clique en dehors de la case il ne se passe rien
Une fois la première carte sélectionnée par le joueur, le programme affiche le contenu de la case en traçant l'objet qui se trouve sous la carte puis attend l'instruction de joueur pour la deuxième carte.
Si les cartes retournées sont de même couleur et de même forme alors le joueur a trouvé une paire et on garde donc les cartes retournées, sinon on attend une seconde pour laisser le temps au joueur de visualiser le contenu avant retourner les cartes (c'est à dire redessiner la carte sans l'objet).
Enfin si les cartes ont toutes été retournée alors la partie est gagné, un écran de fin apparait.
Si au contraire, le joueur a dépassé le nombre de tentatives autorisées, la partie est perdu et autre écran de fin apparait.
Dans les autres cas, le jeu se prépare à recevoir un nouveau couple et à répéter les mêmes instructions.

Le décor :

Le programme décor est appelé dans le programme du jeux grâce à la fonction main. Concernant la structure du programme décor, celui-ci est composé de plusieurs fonctions qui ont pour rôle de tracer des objets. Le programme principal est indépendant du décor il serait donc possible de créer plusieurs décors sans avoir à changer le programme (on pourrait créer par exemple plusieurs saisons), pour cela il suffirait de changer le module décor.
Le fichier décor est décomposé en plusieurs fonctions :

- La fonction main : c'est la fonction qui est appelée par le programme principale et qui appelle les différentes fonctions chargées de tracer le décor.
- La fonction generateEtoiles qui créer une liste de n tuples, en respectant un espacement minimum entre les étoiles afin d'éviter qu'elle se superposent.
- La fonction etoiles, cette fonction permet de tracer les étoiles en fonction des n tuples obtenus grâce à la fonction generateEtoiles.
- La fonction et carotte qui est appelée dans la fonction bonhommeDeNeige afin de tracer le nez du bonhomme de neige.
- La fonction bonhommeDeNeige : trace un bonhomme de neige :

Les lignes 91 à 125 sont dédiées à la construction du bonhomme de neige, pour cela nous nous sommes servis de la fonction rond dans le fichier formes pour tracer les corps du bonhomme de neige ainsi que ses yeux et les boutons qui sont sur son corps. Nous nous sommes servis de la fonction triangle qui se trouve dans le fichier formes, afin de tracer un triangle orange à l'horizontale qui représente la carotte qui correspond au nez du bonhomme de neige. Enfin, nous avons utilisé la fonction rectangle dans le fichier formes pour tracer les bâtons qui forment les bras du bonhomme de neige, cette fonction trace un rectangle marron incliné.
Concernant la fonction rond celle-ci trace un rond circonscrit dans un carré (imaginaire) à partir du point en bas à gauche du carré dont on donne les coordonnées.
Puis nous avons commencé à tracer les bâtons qui servent de bras au bonhomme de neige au niveau du milieu de la deuxième boule du bonhomme de neige, nous avons tracé ces deux bâtons avec un angle de 180 et 50 degrés. Ensuite les boules qui forment le corps, pour cela on appelle une première boule puis une deuxième avec un diamètre qui est égal à 75% du diamètre de la première boule et enfin on trace la troisième boule avec les bonnes coordonnées et un diamètre égal à 50% du diamètre de la première boule.
Puis on finit par les petits éléments sur le bonhomme de neige comme ses yeux que nous traçons avec les bonnes positions, avec comme abscisse la position de la troisième boule plus 1/6 du diamètre de la troisième boule moins le rayons des yeux (car nous venons de déterminer le centre du cercle or la fonction trace le rond à partir du bas à gauche du carré, donc on retire rayon yeux pour le recentré et bien le tracer) et comme ordonnée la hauteur de la troisième boule du bonhomme de neige plus 2/3 du diamètre de la troisième boule pour le premier œil et la comme abscisse du deuxième œil la troisième boule plus 5/6 du diamètre de la troisième boule moins rayon œil (pour la même raison) et la même ordonnée que le premier œil.

Pour la carotte on la trace avec une abscisse qui correspond à l'abscisse de la troisième boule plus la moitié du diamètre de la troisième boule et une ordonnée qui correspond à la hauteur de la troisième boule plus 1/3 du diamètre de la troisième boule
Enfin, pour les boutons sur son corps nous nous servons d'une boucle qui appelle la fonction rond qui se trouve dans le fichier formes et qui va tracer trois boutons avec des cordonnées différentes (on ajoute 25 à l'ordonnée).
Ainsi nous avons appelé les fonctions dans cet ordre de manière à superposer les éléments les uns sur les autres et de manière à avoir un bonhomme de neige

Formes :

Le fichier formes est composé de nombreuses fonctions qui ont pour objectifs les différents éléments graphiques du jeu comme le décor ou bien les objets sous les cartes (le dos de la carte). Ce fichier contiens notamment la fonction dessine qui est appelée pour chacune des fonctions du fichier formes, cette fonction dessine permet d'initialiser la position, couleur, l'épaisseur et l'orientation de la tortue. Chacune de fonctions sont assez détaillées et nécessite pas forcement beaucoup plus d'explications que celle fournies dans le code

Settings :

C'est le fichier qui contient les paramètres du jeux (comme le nombre de cartes, la couleur et les formes des objets de la carte) et que l'utilisateur peut modifier.

Cartes :

C'est le fichier qui dessine les cartes et/ou les effaces afin d'afficher son contenu lorsqu'on sélectionne la case.
Dans le fichier il y a plusieurs fonctions :

- La fonction `dessineCase` qui dessine la carte.
- La fonction `positionCase` qui calcule la position des cartes.
- La fonction `obtenirCase` qui calcule l'index de la carte présente aux coordonnées en paramètre.
- La fonction `afficheContenu` qui redessine le contenu des cartes non sélectionnée et qui révèlent le contenue de la carte sélectionnée (ne dessine pas le contenu de la carte sélectionnée) et celle qui ont déjà été révélée (les paires de cartes déjà trouvée).

Informations supplémentaires :

La durée du projet : ce projet nous aura pris environ 6 semaines pour environ une soixantaine d'heures (au moins).

Problèmes rencontrés : durant le développement de ce projet nous avons rencontrés quelques problèmes :
-le clique de la souris pour choisir les cases pour le traçage des cercles, puisque les cercles étaient tracés à partir d'un carré (imaginaire) depuis le coin à gauche et non le centre de celui-ci

Rajoutes tes problèmes ici

Améliorations possibles :

Une amélioration possible pour notre jeu serait de pouvoir choisir la difficulté au démarrage, celle-ci pourrait par exemple consister en une diminution du le nombre de tentatives autorisées, ou encore par une augmentation du nombre de cartes dans la partie.
Une autre amélioration possible serait de pouvoir recommencer la partie sans avoir à redémarrer le jeu.
Enfin, on pourrait imaginer implémenter un système de saison dans le jeu qui changerait les cartes et le décor en fonction de celles-ci.

Ce que le projet nous a apporté :

Finalement ce projet a été une réelle façon de s'entraîner sur python, avec recherche de solution à tous nos problèmes auxquels nous avons dû faire face, cela nous a aussi permis d'imaginer le travail de développeur en entreprise avec un travail précis à réaliser avec une date limite. De plus cela nous a aussi permis d'utiliser et de découvrir certains logiciels et plateforme de programmation. Mais encore cela nous a aussi permis d'apprendre le travail en équipe et nous appris à être coordonnée par exemple en répartissant les tâches de travail et en mettant en place des horaires de travail en communs notamment par appel discord pendant les vacances par exemple. Enfin ce projet nous à permit d'améliorer notre niveau en python peut import notre niveau de base.