Galli Evan Rapport projet : Memory

Lecard Maxence

Groupe 5

**Répartition et organisation du travail :**

Nous avons utilisé le logiciel Visual studio 2022 pour nous faciliter la tâche en termes de programmation, et la plateforme GitHub afin de nous faciliter la tâche en termes de communication et notamment pour se transmettre le travail en effet grâce à GitHub nous avions un historique nous permettant de suivre les modifications apportées par notre binôme.

Pour la répartition du travail, nous avons travaillé en binôme, pour cela Evan s’est occupé du jeu, et Maxence s’est occupé du décor.

Bien que ce rapport soit destiné à apporter des explications quant à notre programme, les codes contiennent de nombreux commentaires afin d’expliquer plus en détail certains passage.

**Thème graphique et niveaux de difficulté :**

Nous avons choisi le niveaux 2(plus) pour le projet, l'utilisateur clique sur l'objet de son choix.

Le thème de notre jeu est l’hiver.

Concernant les éléments du décor nous avons mis les deux bonhommes de neige au premier plan à différents endroits et les éléments du décor qui se répètent sont les étoiles, celles-ci scintillent et sont positionnée aléatoirement à chaque redémarrage du jeu.

Capture d’écran du jeu

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 même couleur alors la carte reste retournée puisque la paire a été trouvée, (le cas échéant, les cartes se retournent à nouveaux), 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 (correspond au double du nombre de couple dans la partie), 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 la partie

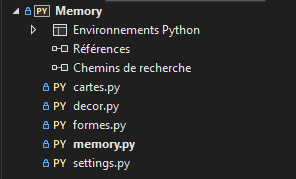
**Structure du code :**

Notre projet est structuré de la façon suivante :

Nous avons le code principal qui appelle le décor (la fonction main) et le jeu est appelé fonction par fonction dans le programme principale.

Nous allons voir un peu plus en détail les fonctions et les différents programmes de notre projet, pour plus de détail, il y a aussi plusieurs commentaires dans le programme qui permettent de mieux comprendre le code si besoin.

Capture d’écran du dossier



**Programme principale :**

Le programme principal se trouve dans le fichier memory.

Au début de notre programme, nous appelons les modules et les librairies que nous besoin.

Nous avons désactivé le redimensionnement de la fenêtre de jeux car celle-ci nous a posé problème pour le décor. Nous nous sommes servis de trois tortues différentes pour tracer le décor, les cases et la barre de progression. Ensuite nous appelons main qui trace le décor.

**Génération aléatoire des paires de cartes :**

On génère ensuite des paires, le nombre de carte est déterminée à partir du nombre de carte sur une colonne et le nombre de carte sur une ligne.

Puis On 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 de même forme). Puis on ne garde que les couples forme/couleur non tirée et on sélectionne les couples forme/couleur restant puis on duplique les cartes obtenues pour obtenir des couples et enfin on mélange le contenu de la liste (case) de manière aléatoire pour bien les répartir dans le jeu.

**Le jeu :**

Le programme de jeux est contenu dans une fonction nommée clickCases.

On définit la fonction clickCases qui est appelée lorsque l’utilisateur clique sur une case, et elle détecte si le clic s’est produit sur une case et révèle son contenu, 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, puis on force l’actualisation du contenu si les cartes retournées sont de même couleur et de même forme alors le joueur a trouvé une paire et donc on garde les cartes retournées, sinon on attend une seconde pour visualiser le contenu et on retourne les cartes autrement dit on redessine la case sur l’objet. Enfin si les cartes ont toutes été retournée **alors on ferme le jeu**, sinon le jeu se prépare à recevoir un nouveau couple et à répéter les mêmes instructions pour un autre couple sélectionné par le joueur.

**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 :

De la ligne 128 à 161 sont dédiée 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 avons créé une fonction carotte qui trace un triangle rectangle orange à l’horizontale. 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. Pour chacune de ces fonctions, nous appelons la fonction dessine qui se trouve dans le fichier formes au début de la fonction, en effet, celle-ci initialise la position, couleur, l’épaisseur et l’orientation de la tortue.

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.

**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).

**Ce que le projet nous a apporter :**

Finalement ce projet a été une réelle façon de s’entrainer sur python, avec la recherche de solution à tous nos problèmes auxquels nous avons fait face, cela nous a aussi permis d’imaginer le travail de développer en entreprise avec un travail précis à réaliser avec une durée limite. De plus cela nous a permis d’utiliser et de découvrir certains logiciel et plateforme de programmation. Enfin ce projet nous à permit de travailler en équipe et d’améliorer notre niveau en python peut import notre niveau de base.

**Améliorations possibles :**

* Choix de la difficulté
* Permettre de recommencer sans fermer le jeu