

Rapport Conception Logiciel

Hocine Bourouih

L1 Info
Groupe 4B

20 Avril 2022



Table des matières

1 Introduction	2
1.1 Description générale du projet	2
2 Objectifs du projet	2
2.1 Description plus précise de ce qu'il fallait faire	2
2.2 Description de travaux existants sur le même sujet	3
3 Fonctionnalités implémentées	3
3.1 Description des fonctionnalités	3
3.1.1 Le Menu	3
3.1.2 Le Plateau	4
3.1.3 Les Règles/Didacticiel	5
3.2 Organisation du projet	6
4 Éléments techniques	7
4.1 États du jeu	7
4.2 Fonctionnement du plateau	8
4.3 Descriptions des structures de données	8
5 Architecture du projet	9
6 Expérimentations et usages	9
6.1 Cas d'utilisation	9
6.2 Résultats quantifiables	11

7 Conclusion	13
7.1 Récapitulatif des fonctionnalités principales	13
7.2 Propositions d'améliorations	14
7.2.1 Pour le Menu	14
7.2.2 Pour le Plateau	14
7.3 Conclusion	15

1 Introduction

1.1 Description générale du projet

Pour notre projet de conception logicielle nous avions choisi le sujet suivant :

- L'interface pour jeu de cartes

A l'origine nous souhaitions faire un jeu de cartes en 3d , cependant nous nous sommes vite rendus compte que ce n'était pas faisable . Donc nous nous sommes concertés , et parvenu à la conclusion suivante qu'il fallait créer un jeu sur Pygame en 2d, avec comme spécificité, l'utilisation d'un Sandbox (Bac à sable).

Pour faire ce jeu, il nous fallait imposer le moins de contrainte aux joueurs afin de lui procurer un sentiment de liberté.

Néanmoins, il ne fallait pas non plus que le joueur se retrouve au dépourvu devant notre jeu. De cette manière le joueur peut jouer à n'importe quel jeu de cartes de son choix : Solitaire, Bataille ...

Le jeu est donc en vue du dessus et possède de nombreuses fonctionnalités qui seront énoncées et expliquées plus tard .

Par ailleurs ce jeu s'appelle Card Game

2 Objectifs du projet

2.1 Description plus précise de ce qu'il fallait faire

A l'origine nous avions plusieurs choix disponibles :

- Faire un jeu de cartes avec des points de vie et d'attaque.
- Faire un jeu de cartes avec des cartes dites classiques : As Roi Reine etc...
- Faire un jeux de société type : Uno , Mille Bornes etc ...

Nous avons opté pour l'option numéro 2, il s'agit donc d'un jeu de 54 cartes (Joker pouvant être retiré).

Il nous fallait donc créer une structure à ce jeu. Le jeu est donc divisé en trois grand Axes :

- L'accueil qui permet d'accéder aux deux autres parties.
- La zone de jeu , l'endroit où le joueur est sensé passer le plus de temps.
- La zone d'explication qui explique la quasi-totalité des fonctionnalités du jeu

2.2 Description de travaux existants sur le même sujet

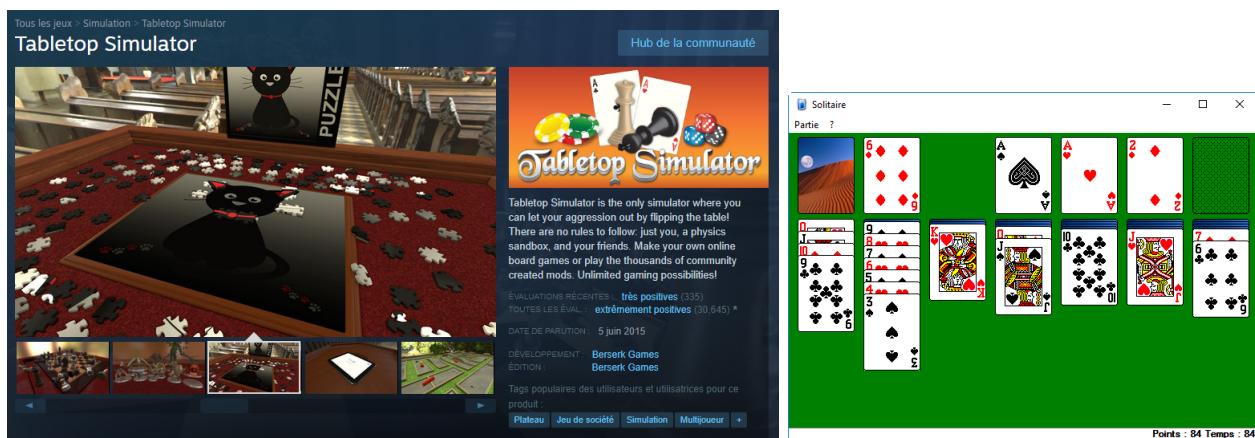
Pour faire ce projet nous nous sommes inspiré d'un jeu déjà existant , il s'agit de "Tabletop simulator" . Ce jeu est une simulation de jeux de société qui permet de jouer à quasiment tout ce qui existe (en l'occurrence les cartes) . Cependant ce jeu est bien différent du nôtre, en effet il possède une interface 3d , il est même compatible avec la VR .

La raison pour laquelle nous nous sommes inspirés de ce jeu est simple :

Ce jeu est le plus gros jeu Sandbox en terme de fonctionnalités , ce qui explique notre intérêt envers lui .

En ce qui concerne l'inspiration graphique et ses fonctionnalités notre jeu se rapproche plutôt d'un solitaire sur window xp .

Néanmoins je tiens à rappeler que notre jeu n'est pas un solitaire mais plutôt un jeu qui pourrait simuler un solitaire ou n'importe quel jeu de cartes existant (à partir des 54 cartes de base) ce qui explique les différences.



3 Fonctionnalités implémentées

3.1 Description des fonctionnalités

Comme dit précédemment le jeu se divise en 3 grandes parties , nous verrons donc en détail leurs contenus :

3.1.1 Le Menu

- Le menu est la partie centrale du jeu cependant il ne s'agit pas de la partie la plus aboutie ou contenant le plus de code .

On y retrouve néanmoins de nombreuses fonctionnalités :

Pour commencer nous remarquons que ce menu ne contient que des boutons.

En effet ces derniers permettent aux joueurs de pouvoir faire de nombreuses interactions que nous allons présenter :

Premièrement : Dans le coin de l'image (situé en dessous) supérieur droit nous avons 3 boutons qui permettent respectivement de :

- Quitter le jeu .
- Couper la musique du jeu .



FIGURE 1 – Screenshot du Menu de Card Game

- Changer la soundTrack (la liste de musique que contient le jeu)

conception logiciel > interface-carte-hgmx > musiques				
Nom	N°	Titre	Interprètes ayant p...	Album
music2		Fluffing a Duck	Kevin MacLeod	Royalty Free
music3		Fallen leaves		
music4		Jingle Bells	Mystery Mammal	
musique1		Casinò Music 4 Poker Ga...	Sensual Music Clu...	
musique2		Casinò Music 4 Poker Ga...	Sensual Music Clu...	
musique3		Casinò Music 4 Poker Ga...	Sensual Music Clu...	
musique4				

Deuxièmement : Dans le coin de l'image supérieur gauche on retrouve les Règles ou plus communément appelé le didacticiel .

Il faut savoir que l'idée qui à conduit à faire cette élément du jeu est insolite car en réalité il s'agissait d'un moyen de diviser l'équipe .

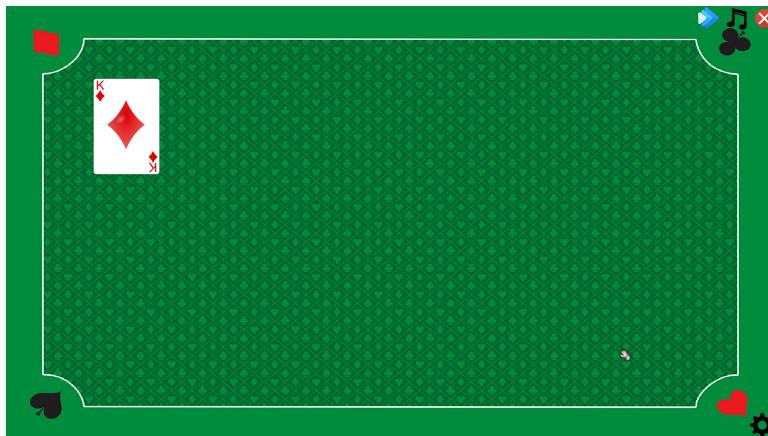
En effet afin que chacun puisse travailler sur sa partie tout en évitant au maximum les soucis de corruption de données, qui avait lieu lorsque qu'une personne faisait un commit sans update au préalable (ce qui est survenu de nombreuses fois)

Troisièmement : Dans le centre de l'image on retrouve le bouton qui permet d'accéder à la zone de jeu qui sera appelé le "Plateau" (Il s'agit en effet du nom de son fichier Python : Plateau.py).

Il s'agit par ailleurs de la partie la plus travaillée du jeu .

3.1.2 Le Plateau

- Le plateau est la partie contenant le plus de code comme dit précédemment. Le plateau dans son état initial ressemble à ceci (voir image si dessous).



On y retrouve donc un curseur , ce curseur se situe sur les 3 parties (Plateau, Menu, Règles). Il sert bien évidemment à manipuler les éléments qui se situent sur le jeu .

Je tiens tout de même à préciser que les différentes instance de sprite du curseur, ont été créées de toutes pièces, à l'exception du curseur dans son état initial.

Celui ci provient du jeux Dragon Quest 9 Les Sentinelles du Firmament .

On retrouve par ailleurs les 3 boutons situés dans la partie supérieur droite que j'ai déjà expliqué précédemment .

On peut aussi apercevoir un paquet de 54 cartes situés en haut à gauche.

De plus amples informations seront données afin d'expliquer comment est constitué chaque paquet, notamment dans la partie 4 sur les structures de données

Il est possible sur le plateau de :

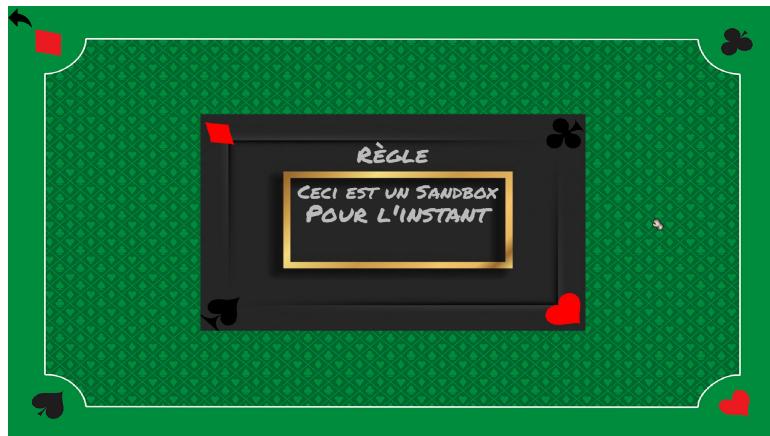
- Déplacer des cartes grâce au curseur.
- Tourner les cartes grâce au clic droit .
- Mélanger les cartes avec une Touche (La touche M).
- Empiler des cartes ce qui crée de nouveaux paquets.
- Avec Ctrl + clic gauche, il est possible de déplacer l'entièreté d'un paquet.
- Avec Ctrl + clic droit, il est possible de retourner l'entièreté d'un paquet.
- La touche Tab permet de faire apparaître de nouveaux paquets .

Enfin le plateau possède un bouton dans la partie inférieure droite qui permet de faire apparaître un petit menu qui contient :

- Un bouton pour réinitialiser le plateau .
- Un bouton pour revenir au Menu .
- Un bouton pour revenir sur le plateau (retour arrière).

3.1.3 Les Règles/Didacticiel

Le didacticiel aussi appelé Règle en raison du nom du fichier python (Regle.py) est la partie qui permet de guider le joueur / l'introduire dans Card Game .



En premier lieu on remarque que Règle explique au joueur le principe fondamental du jeu : Il s'agit d'un Sandbox.(voir image ci dessus)

Une fois que le joueur à cliqué sur l'image plusieurs choses apparaissent .

- Une boite de dialogue (Trouvée sur le site : "<https://pngdaunrodo.web.app/text-box-png-game.html>").
- 2 à 3 lignes de texte avec la police d'écriture : "Permanent Marker".
- 8 animations faites en pygame qui représentent ce qui est possible de faire sur le Plateau.

Enfin il existe un bouton situé dans la partie supérieur gauche qui permet de revenir à tout moment au Menu .

A savoir : Lorsque toutes les animations ont été affichées le jeu retourne automatiquement sur le Menu .

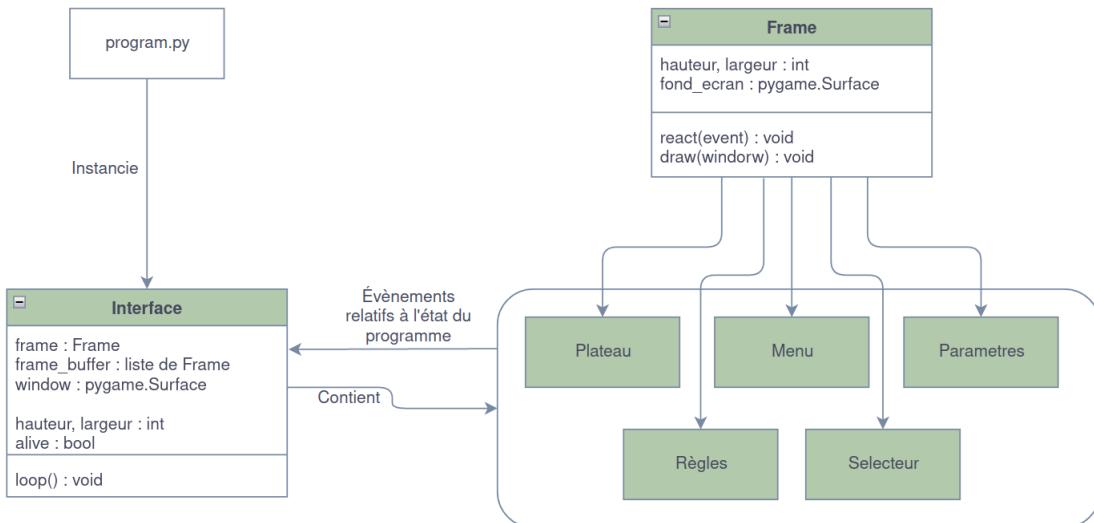
3.2 Organisation du projet

4 Éléments techniques

Dans cette partie, nous allons voir les attributs et les fonctions de chaque classes, ainsi que la manière dont chaque classes communiquent avec les autres.

4.1 États du jeu

Dans un premier temps, voici comment le jeu passe d'un état à un autre :



Afin d'exécuter le jeu, nous demandons à l'interpréteur python de lancer le fichier "program.py" ce qui va permettre d'instancier un objet de type "Interface". Cette classe permet de créer la fenêtre, elle contient plusieurs variables : la variable "alive" qui permet de contrôler la fin du programme, la hauteur et la largeur, ainsi que "frame" et "frame_buffer". La fonction `loop` de cet classe permet de continuer à exécuter le programme.

L'objet "frame" est un objet d'une des classes héritant de la classe "Frame", cet objet réagit en fonction des événements donnés par la fonction "loop" de l'interface grâce à la fonction "react", puis elle est dessinée grâce à la fonction "draw".

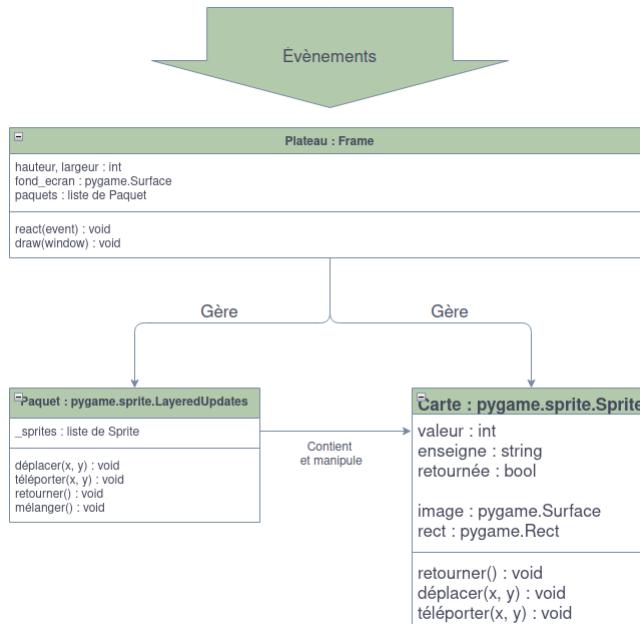
Cela permet d'avoir fenêtres interchangeables grâce aux événements pygame.

Par exemple, lorsqu'on lance le programme, l'interface frame devient un objet de type "Menu", puis lorsque l'on appuie sur le bouton "Jouer", l'interface envoie l'événement capté dans la fonction `react(event)` de frame (qui est de type `Menu`). Enfin, pygame envoie un événement d'utilisateur pour changer l'état du programme en mode plateau, donc la variable `frame` qui était un objet `Menu` devient un objet `Plateau`.

La liste `frame_buffer` permet de garder des frames en arrière plan, elle sera donc dessiné avant la frame principale et il n'est également pas possible d'interagir avec la plupart du temps.

4.2 Fonctionnement du plateau

Nous allons à présent nous intéressé à la manière dont fonctionne le plateau de carte.



Chacun des événements sont pris en charge par la fonction react(event) de la frame. Par exemple, si un clic gauche est détecté, la fonction react cherche à savoir si le clic a touché un bouton ou une carte, puis réagir en conséquence.

Nous avons également du prendre en compte les combinaisons de touches (pour prendre tout un paquet par exemple), ce qui a amené à la création d'un dictionnaire pour savoir quels touches étaient enfoncés.

4.3 Descriptions des structures de données

Les paquets de Plateau.py sont un bon exemple de comment les structures de données sont employé dans notre code.

Concrètement lorsqu'un paquet est généré par la touche Tab : Plateau.py va générer une nouvelle liste contenant une liste de 54 cartes (varie en fonction des jokers).

Si une carte est retirée du précédent paquet, l'ancien paquet va perdre la carte sélectionnée. Mais de l'autre coté le Plateau va de nouveau créer une liste contenant une liste avec un seul élément : la carte qui a été déplacée.

Avec des fonctions comme Append ou bien Empty il devient alors très facile de faire naviguer des cartes d'un paquets à un autre .

5 Architecture du projet

Le projet a été organisé par type de fichiers :

```
> __pycache__
> images
> musiques
> police
> Rapport Conception Logiciel
  bouton.py
  carte.py
  event.py
  exceptions.py
  frame.py
  groupeBouton.py
  interface.py
  menu.py
  musique.py
  paquet.py
  parametre.py
  plateau.py
  program.py
  regle.py
  selecteur.py
  souris.py
```

Les fichiers sources python sont directement dans la racine du projet, tandis que les ressources sont inclus selon leur types (fichiers .mp3, .mp4 dans musique, fichiers .jpg, .png, .bmp dans images, etc...).

Le dossier " __pycache__ " contient des versions pré-compilés du code source que l'on a écrit afin d'aider l'interpréteur à exécuter le programme de manière plus efficace.

6 Expérimentations et usages

6.1 Cas d'utilisation

Après avoir lancer le jeu avec la commande « python3 program.py », la fenêtre du jeu s'ouvre en plein écran, pour cela on a adapté notre code pour un bon affichage sur les différentes taille de l'écran, et cela en évitant d'utiliser des valeur fixe en pixel, mais plutôt de valeur en fonction de la longueur et la largeur de la fenêtre, comme le positionnement des Sprite par exemple (largeur/2, hauteur/2 pour placez un Sprite au milieu de n'impose écran).



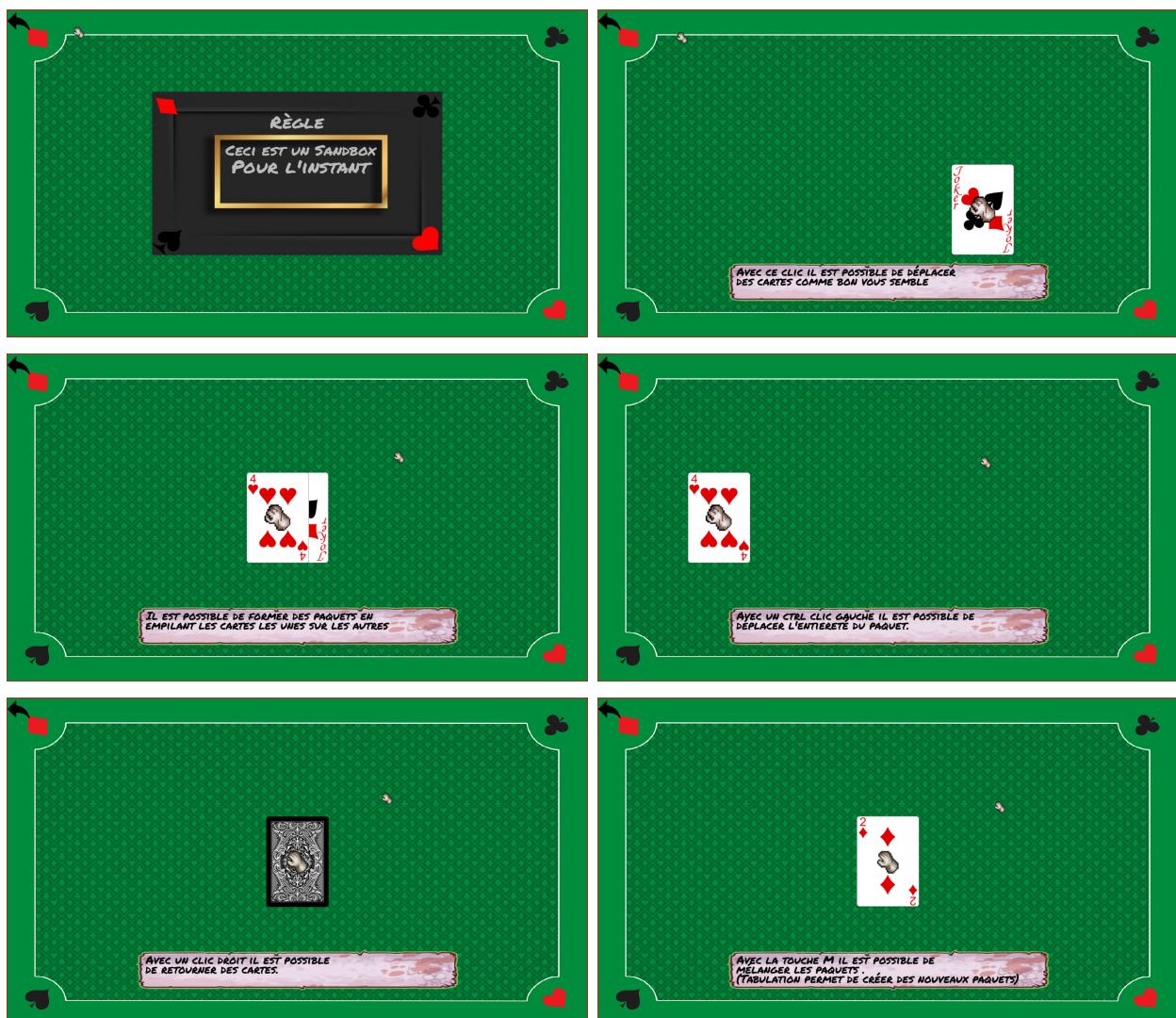
On se retrouve sur notre menu on peut directement commencez à jouer, et aussi on a accès à certain paramètre comme le bouton règle, le bouton activer/désactiver la musique ou même

changer la musique.

Et lorsque on jeux à un jeux on aime bien savoir comment ça marche, pour cela on a décidé de mettre une démo avec des animations pour qu'il soit bien claire, ou on présente tout le fonctionnement du jeu, les différentes méthodes du jeu et les touches qu'il faut utiliser (accès : en cliquant sur le bouton règle haut à gauche de la fenêtre).

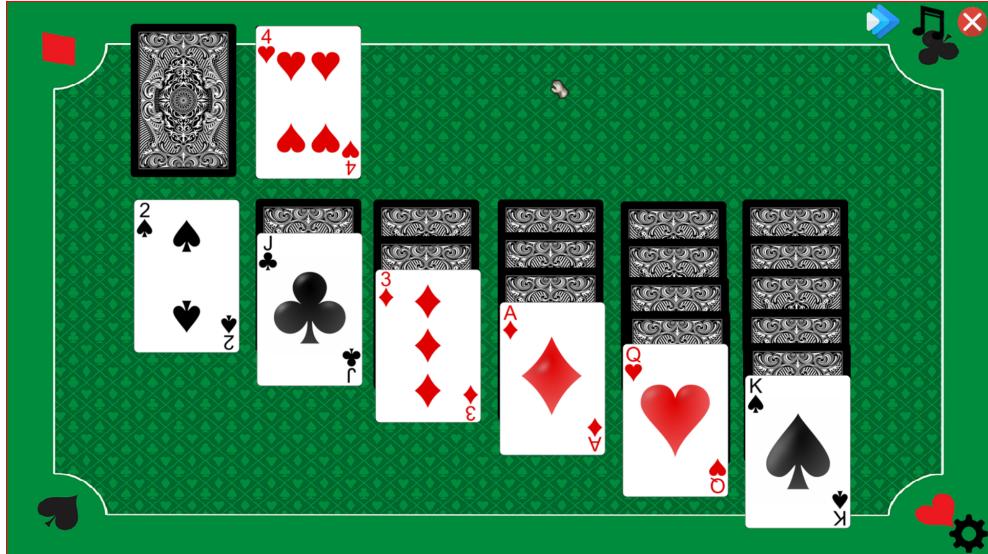
Pour cela :

- Avec le « click gauche + mouvement de la sourie » on peut déplacer les cartes, avec ce déplacement on peut fusionner deux cartes ensemble ou plus et cela nous forme un nouveau paquet
- Avec un « CTRL + clic gauche » on peut déplacer l'entièreté du paquet
- Un « clic droit » permet de retourner les cartes
- La « touche M » sert à mélanger les cartes du paquet
- On a aussi la possibilité d'ajouter de nouveau paquet à partir de la touche « tab »



Maintenant si vous décidez de jouer, vous sera dirigé au plateau de jeu de carte, sur cette interface, commencez à jouer toute en appliquant les règles expliquer dans la démo précédente.

Astuce : utilisez la touche « M » pour mélanger les cartes et obtenir les carte souhaiter, Vous pouvez ainsi ajouter de nouveau paquet à partir de la touche « tab ».



Pendant tout le jeu, une musique marche en boucle, avec la possibilité de désactivé le son ou aussi de swiper entre les différente chanson de la playlist.



On a mis à votre disposition un bouton paramètre, à partir de ce dernier vous pouvez mettre en pause le jeu, reprendre le jeu, refaire la partie de nouveau, et aussi revenir à la démo de règle du jeu.



6.2 Résultats quantifiables

Côté performance, le jeu se lance bien, fluidité et rapidité, pas de ralentissement ni bug. Avec cette ligne de code « print(self.clock.get-fps) » on récupère le nombre de FPS ce qui représente

le nombre de frame ou image par seconde.

Tout au long du jeu le nombre de FPS varie entre 120 et 190 images par seconde, ce qui donne une fluidité, une clarté et une excellente résolution au jeu. Et cela est totalement logique car on est largement au dessu de la moyenne, dans le jeu vidéo particulièrement sur les supports consoles comme la PlayStation 4 on est sur des 60 FPS (pour 60 images par seconde) et la Xbox One on est 30 FPS (pour 30 images par seconde).

```
Run: program x
128.2051239013672
128.2051239013672
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
147.05882263183594
```

On remarque une légère descente de performance du jeu mais cela juste lorsque on restart le jeu de nouveau, et cela parce que toutes les frames, les Sprite, les events et les image sont redessiné même celle qui non pas changer comme l'arrière-plan du plateau par exemple, et malheureusement on a pas réussi à optimiser et redessiner seulement les frames qui ont changer, par conséquence on remarque un petit ralentissement du jeu et des images un peu bruité.

Mais cela est résolu avec le conseil de notre prof, on a coder une fonction qui bloque tous les évènements non utiliser, et cela nous a aider à enlever tout les bugs et le ralentissement du jeu, et a passer de 10 FPS à 30 FPS par seconde ceux qui est l'équivalent dans les jeux vidéo de la console Xbox One.

```
interface.py x parametre.py x program.py x musique.py x
print
77
78     def blockUselessEvents(self):
79         pygame.event.set_blocked(pygame.JOYAXISMOTION)
80         pygame.event.set_blocked(pygame.JOYBALLMOTION)
81         pygame.event.set_blocked(pygame.JOYHATMOTION)
82         pygame.event.set_blocked(pygame.JOYBUTTONDOWN)
83         pygame.event.set_blocked(pygame.AUDIODEVICEADDED)
84         pygame.event.set_blocked(pygame.AUDIOVICEREMOVED)
85         pygame.event.set_blocked(pygame.FINGERMOTION)
86         pygame.event.set_blocked(pygame.FINGERDOWN)
87         pygame.event.set_blocked(pygame.FINGERUP)
88         pygame.event.set_blocked(pygame.MULTIGESTURE)
89         pygame.event.set_blocked(pygame.TEXTEDITING)
90         pygame.event.set_blocked(pygame.DROPBEGIN)
91         pygame.event.set_blocked(pygame.DROPCOMPLETE)
92         pygame.event.set_blocked(pygame.DROFILE)
93         pygame.event.set_blocked(pygame.DROPTEXT)
94         pygame.event.set_blocked(pygame.MIDIIN)
95         pygame.event.set_blocked(pygame.MIDIOUT)
96         pygame.event.set_blocked(pygame.CONTROLLERDEVICEADDED)
97         pygame.event.set_blocked(pygame.CONTROLLERDEVICEREMAPPED)
98         pygame.event.set_blocked(pygame.CONTROLLERDEVICEREMOVED)
99         pygame.event.set_blocked(pygame.JOYDEVICEADDED)
100        pygame.event.set_blocked(pygame.JOYVICEREMOVED)
101
102        pygame.event.set_blocked(pygame.WINDOWHITTEST)

Interface > blockUselessEvents()
```

7 Conclusion

7.1 Récapitulatif des fonctionnalités principales

En Conclusion notre jeu possède les éléments nécessaire pour être qualifié de Sandbox. Le joueur peut simuler n'importe quel jeu de cartes (qui découle des 54 cartes). Sans ressentir les contraintes techniques d'un jeu en 2d. Grâce aux faculté présente :

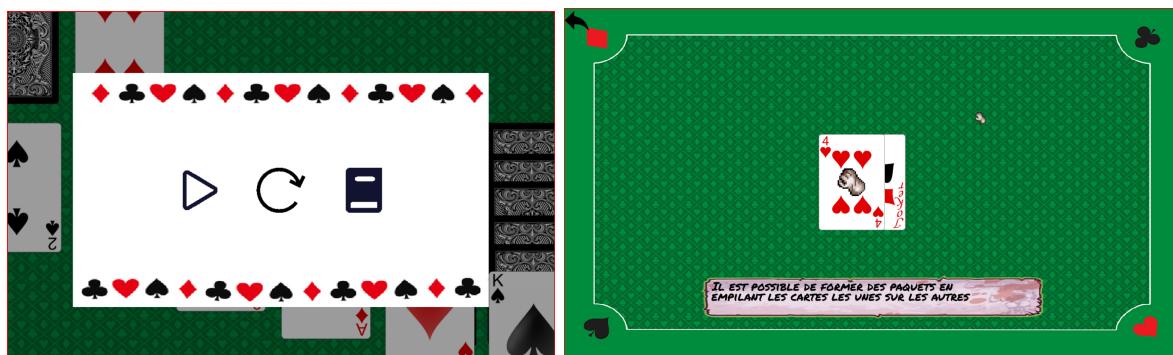
- Bouger les cartes.
 - Créer des paquets de carte.
 - Retourner des cartes.
 - etc ...

Ainsi que les éléments qui ont été mis en place afin de rendre l'expérience du joueur plus satisfaisante .

- Le bouton Paramètres dans Plateau.
 - Regle.py avec ses animations.
 - Les boutons de musiques.
 - etc ...

Nous estimons que le joueur possède une grande liberté ce qui colle avec notre idée de faire un jeu Sandbox .





7.2 Propositions d'améliorations

Cependant de nombreuses choses aurai put être améliorées voir ajoutées .

7.2.1 Pour le Menu

En ce qui concerne le Menu :

- Un fond d'écran qui se déroule aurai pu être codé à la place de l'image actuel , c'est à dire une image qui défile vers la gauche ou la droite afin de créer du mouvement dans cette partie du jeu (à l'instar de Minecraft qui fait bouger l'image de son menu vers la droite).
- Le visuelle des boutons aurai pu être amélioré par ailleurs . Par exemple en mettant en place une zone d'une certaine couleur qui les délimiterai du reste du jeu, ce qui les aurai rendu plus visible et dans une moindre mesure plus beau visuellement .

7.2.2 Pour le Plateau

Pour le Plateau :

Le Plateau dans son état initial est plutôt vide . - Des dés aurai pu être codé afin de laisser la possibilité au joueur de simuler d'autre jeu de carte .

Je prend pour exemple un jeu nommé le Royaume, un jeu qui utilise un dés afin de savoir combien de carte pioché etc ...

Malheureusement ce jeu ne peut pas être simulé sur Card Game par manque de dé .

- Le manque de preset pour notre jeu est aussi une partie qui a du être évincer du projet par manque de temps . En effet la présence d'un bouton qui mets un place un solitaire en divisant les paquets ou bien tout autre jeu de carte aurai pu être fait. Cette fonctionnalité aurai permis au joueur de pouvoir jouer directement à ses jeux favoris sans même devoir les mettre en place.

-L'ajout d'autre paquets de carte (différent de celui dit classique) aurai aussi pu êtres mis en place . Après tout le jeu est un sandbox la présence de carte avec des points de vie aurai pu être rajouté . Cela aurai permis au joueur de s'ouvrir à un tout autre style de jeu .

-L'ajout d'autres éléments que des cartes, tel que des dés, des jetons de poker, etc... Cela aurait sûrement pu ajouter un peu de diversité.

- Enfin il reste l'optimisation du jeu qui n'est vraiment pas très bonne , dépasser les 3 paquets de 54 cartes à l'écran le jeu n'arrive plus à suivre et ce mets à ramer .
Malheureusement cela aurai pu être réglé en changeant le fond d'écran car il doit être recalculer à chaque tick .

7.3 Conclusion

Pour un premier vrai projet dans le monde de l'informatique et plus précisément python , notre jeu tient plus que la route . Les objectifs de base ont été remplis, en effet il y a bien des choses à améliorer mais l'essentiel y est . Voila ce qui conclut le rapport.

FIN.