

Rapport Conception Logiciel

Mathieu Goudal, Hugo Bosvy, Valère Sarfati,
Edwin Bertin-Cohonner

25 Avril 2022



UNIVERSITÉ
CAEN
NORMANDIE

Table des matières

1	Introduction	2
1.1	Description du projet	2
1.2	Plan du rapport	2
2	Objectifs du projet	2
3	Fonctionnalités implémentées	2
3.1	Description des fonctionnalités	2
3.2	Organisation du projet	3
4	Architecture du projet	6
5	Expérimentations	6
6	Conclusion	12

1 Introduction

1.1 Description du projet

Nous avons pour but de coder en Python, un jeu qui s'intitule "Le président". Ce genre de jeu est un jeu de cartes où chaque joueur doit se débarrasser le plus rapidement possible de la totalité de ses cartes. Les joueurs ayant été les plus rapides à se débarrasser de leurs cartes ont droit à un avantage pour la partie suivante et inversement les plus lents se verront pénalisés par un handicap.

Le "2" est la carte la plus haute, tandis que le "3" est la plus basse.

Les cartes posées doivent toujours être supérieures à celles déjà jouées.

Lors d'une série, le jeu s'arrête lorsque plus aucun des participants ne peut ou ne veut jouer une carte supérieur. Ainsi, il n'est pas possible de faire une boucle comme par exemple As, 2 et repartir sur 3.

1.2 Plan du rapport

Nous évoquerons d'abord quels étaient nos objectifs de départ, puis nous détaillerons les différentes étapes de la création de notre projet avec les rôles de chacun. Ensuite nous présenterons l'architecture de ce projet et finalement nous présenterons certaines expérimentations et terminerons par une courte conclusion.

2 Objectifs du projet

Nous voulions développer une interface graphique pour jeux de cartes. Nous avons donc choisit le jeu de cartes "Président". Ce jeu peut se jouer à 4 joueurs sur la même machine. Nous avons décidé de proposer 4 thèmes différents : *Basique*, *Shingeki no Kyojin*, *Harry Potter* et *Avengers*.

3 Fonctionnalités implémentées

3.1 Description des fonctionnalités

Les possibilités du jeu sont multiples. A partir de l'accueil, il est possible de choisir le thème du plateau de jeu à partir de différents boutons ; les bou-

tons `Theme_basique`, `Theme_snk`, `Theme_harrypotter`, et `Theme_avengers`. Après avoir choisit son thème, il suffit d'appuyer sur le bouton "Play" pour lancer la partie.

Une fois la partie lancée, un bouton "Paramètre" se trouve en haut à droite de l'écran, ce bouton ouvre une nouvelle fenêtre qui permet de changer de thème en cours de jeu, ou d'accéder aux règles du jeu. Dans cette fenêtre se trouve aussi un bouton "Exit" permettant de retourner au plateau de jeu.

En haut à gauche de l'écran se trouve un autre bouton "Son" permettant de couper et remettre la musique de fond.

Un dernier bouton "Jouer", permet de jouer ses cartes sélectionnées, ou de passer son tour.

3.2 Organisation du projet

Pour commencer, Valère et Edwin ont effectués plusieurs recherches sur la manière de coder en pygame.

Mathieu et Hugo, quant à eux, ont commencés à coder "`class_carte`" qui leur a servit à définir les propriétés des cartes (sa valeur, son nom et sa couleur). Elle leur a aussi permit d'attribuer une image à chaque carte et sa "*hitbox*" (surface cliquable de la carte).

Cette classe contient plusieurs fonctions :

- La fonction "`update()`" permet de définir la taille d'une carte.
- La fonction "`maj_image()`" permet d'attribuer le recto correspondant à sa carte.
- La fonction "`_str_()`" permet de récupérer le nom exact de la carte dans la variable "`image_recto`" de la forme nom + couleurs.

Pour créer le paquet de carte, Mathieu et Hugo ont défini "`class_deck`". Ils ont défini une variable "`type`" qui leur permettra par la suite de créer un jeu de 52 cartes ou un jeu vide grâce au fichier "`config`".

Cette classe contient plusieurs fonctions permettant de manipuler le paquet de cartes :

- La fonction `"mélanger()"` appelle la fonction `"shuffle"` de la classe Python `"random"` afin de mélanger le jeu de cartes.
- La fonction `"tirer()"` permet de tirer la dernière carte du jeu et de la supprimer du paquet.
- La fonction `"ajouter()"` permet d'ajouter une carte à un paquet.
- La fonction `"enlever()"` permet d'enlever une carte d'un paquet.

Ensuite ils ont créés des joueurs dans `"class_joueur"`. Ces joueurs possèdent un `"id"` et une `"main"` créée à partir de `"class_deck"`.

Pour le fonctionnement du jeu, toutes les fonctions ont été codées dans `pre-sident"` par Mathieu et Hugo.

Dans cette classe a été créée la `pioche` (à partir de `"class_deck"`). Cette pioche est un jeu de 52 cartes mélangées qui nous sert à construire les mains des joueurs.

La `défausse`, quant à elle, est un deck vide également codé à partir de `"class_deck"`.

`dernier_joueur` est le dernier joueur à avoir joué une ou plusieurs carte(s). `nb_carte_tour` est le nombre de cartes jouées par le joueur qui a ouvert le tour, c'est-à-dire, le nombre de cartes à jouer tout le long du tour.

`joueur_en_cours` sert à savoir qui est le joueur qui est en train de jouer.

Nos 4 joueurs et leurs mains ont été créés ici grâce à de multiples fonctions :

- `"dessine_main_joueur()"` parcourt la main d'un joueur et dessine chaque carte de la main à l'écran. Les mains sont dessinées différemment en fonction de l'id du joueur c'est-à-dire de son placement sur l'écran (haut, bas, droite ou gauche). Cette fonction permet également de monter les cartes lorsqu'elles sont sélectionnées (elles se déplacent encore différemment selon l'id du joueur).
- `"display_main_joueur()"` fait appel à la fonction `"dessine_main_joueur()"` et permet de dessiner toutes les mains de tous les joueurs en même temps. Grâce aux paramètres `x` et `y` de la fonction `"dessine_main_joueur()"` l'affichage des cartes est dynamique, cela veut dire que selon la taille de la main du joueur, les cartes seront toujours centrées. En effet, le `x` ou le `y` changera automatiquement, en fonction de l'id du joueur, quand le joueur jouera une carte. Nous avons attribué une formule au `x` et au `y` qui est : (longueur ou largeur (selon l'id) de l'écran/2)-

((longueur ou largeur (selon l'id) d'une carte + la longueur de la main du joueur*30)/2). Lorsque le joueur va jouer une carte, sa main va se centrer grâce à cette formule.

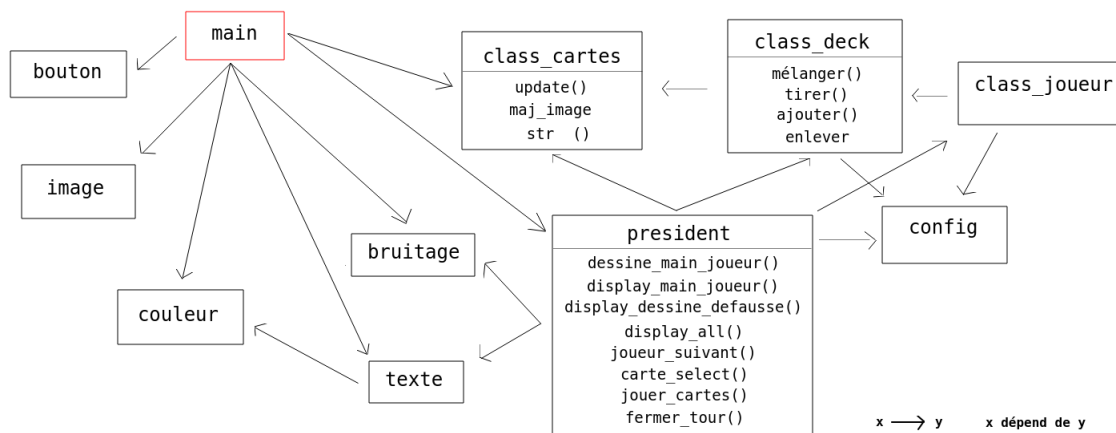
- `"display_dessine_defausse()"` permet de dessiner la défausse, c'est-à-dire, l'endroit où les cartes jouées sont placées. Cette défausse est parcourue et chaque carte qu'elle contient est dessinée et placée au centre du plateau.
- `"joueur_suivant()"` permet de changer de joueur.
- `"carte_select()"` permet au joueur en cours de sélectionner les cartes qu'il veut jouer. Cette fonction fait en sorte que le joueur puisse seulement sélectionner des cartes de mêmes valeurs. En effet, si le joueur sélectionne des cartes de valeurs différentes, alors les premières cartes sélectionnées seront automatiquement désélectionner.
- `"jouer_carte()"` permet de placer les cartes sélectionnées dans la défausse, si elle respecte les conditions fixées :
 1. Le premier joueur à ouvrir le tour, détermine le nombre de cartes qu'il faudra jouer tout le reste du tour grâce à `nb_carte_tour`.
 2. Si le nombre de cartes n'est pas respecté, un message s'affiche et le joueur a la possibilité de ré-essayer de jouer.
 3. Si un joueur joue la carte "2", le tour se ferme automatiquement et c'est encore à lui de ré-ouvrir le nouveau tour.
 4. Si les 4 dernières cartes posées dans la défausse sont de mêmes hauteurs, le tour se ferme ; c'est la règle du *carré magique*.
 5. Si 3 joueurs passent leur tour, le tour se ferme et la main revient à `dernier_joueur`.
- `"fermer_tour()"` permet de vider la défausse.
- `"fin_game()"` nous informe de la fin de la partie, pour pouvoir lancer la fenêtre de fin de jeu.
- `"main_vide()"` permet de passer automatiquement le tour d'un joueur n'ayant plus de cartes.

Mathieu a choisit de ranger les différents éléments graphiques dans différentes classes. Il y a donc une classe "bouton", "couleur", "texte", "image", "bruitage".

Enfin, Valère et Edwin se sont occupés du rapport, tandis que Mathieu de l'interface graphique.

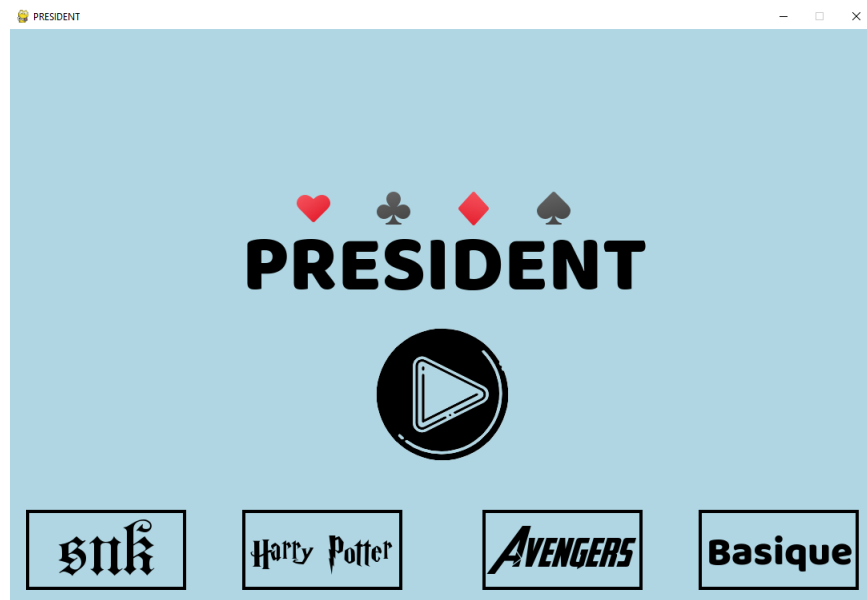
4 Architecture du projet

Diagramme de toutes les classes (et leurs dépendances) :

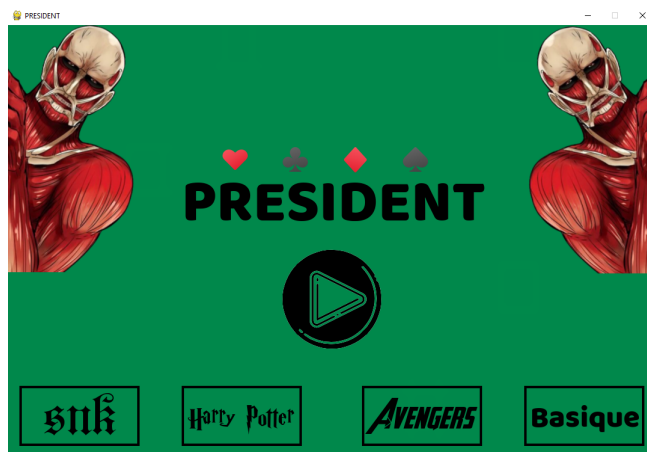


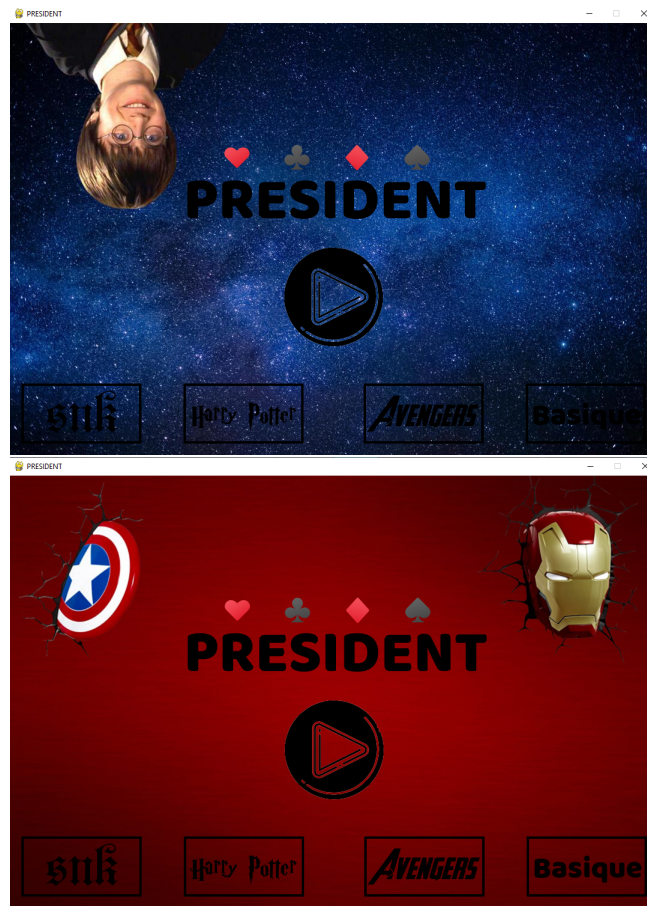
5 Expérimentations

Lorsque vous lancez le jeu, vous arrivez sur une fenêtre d'accueil avec 5 boutons, le bouton **Play** et les boutons des 4 thèmes :

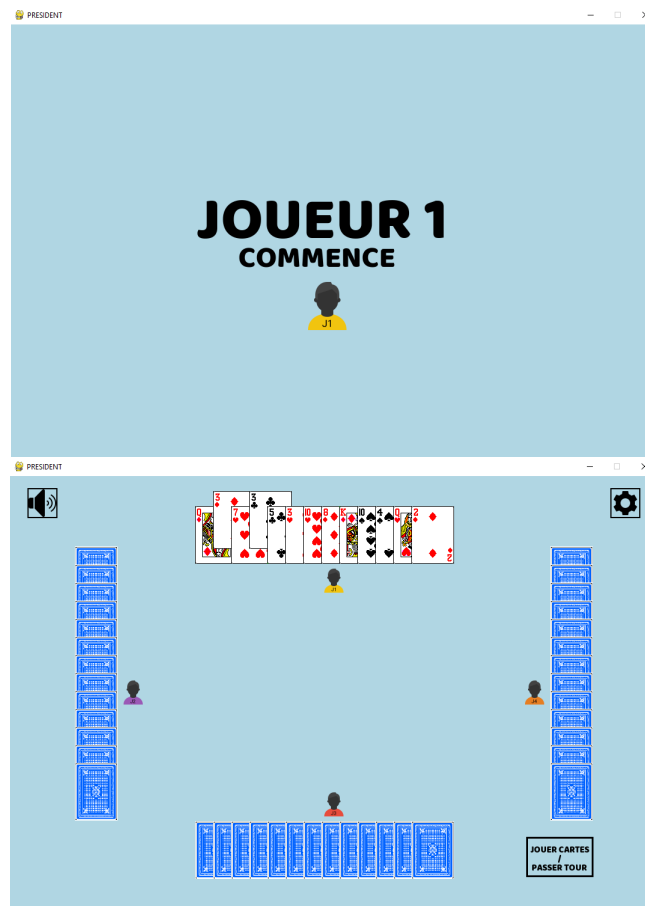


Vous pouvez choisir de changer de thème. Le fond, la musique, et les cartes seront différents :



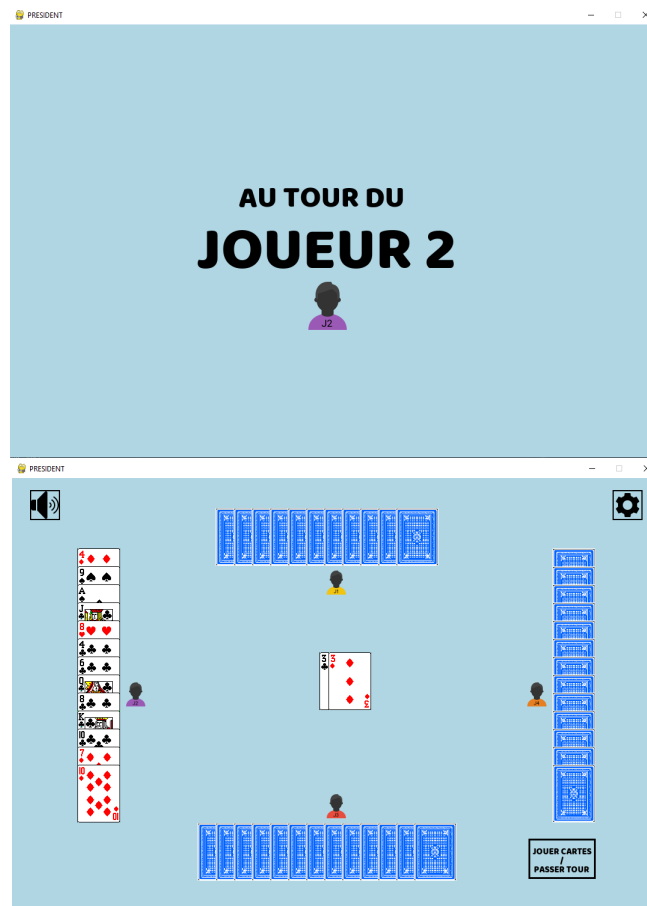


Après avoir choisit votre thème et cliqué sur le bouton **Play**, une page vous annoncera que le joueur 1 débute la partie. Cliquez n'importe où sur l'écran afin de passer au plateau de jeu :



Le joueur 1 pourra alors sélectionner ses cartes (mais que les cartes de même hauteur afin de constituer des simples, des paires, des brelans ou des carrés). Une fois ses cartes choisies, il suffira d'appuyer sur le bouton **Jouer cartes** pour les poser au centre du plateau. Une nouvelle fenêtre s'ouvrira afin d'informer les joueurs que c'est au tour du deuxième, pour qu'ils puissent se préparer à changer de joueurs. La main affichée sera celle du joueur 2 et les cartes du joueur 1 seront posées au centre :

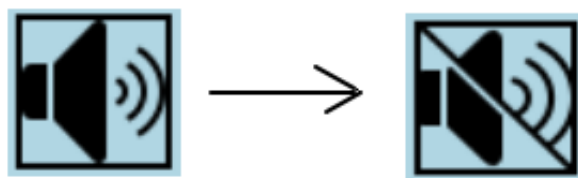




Si un joueur ferme le tour en cours, une fenêtre l'en informera :



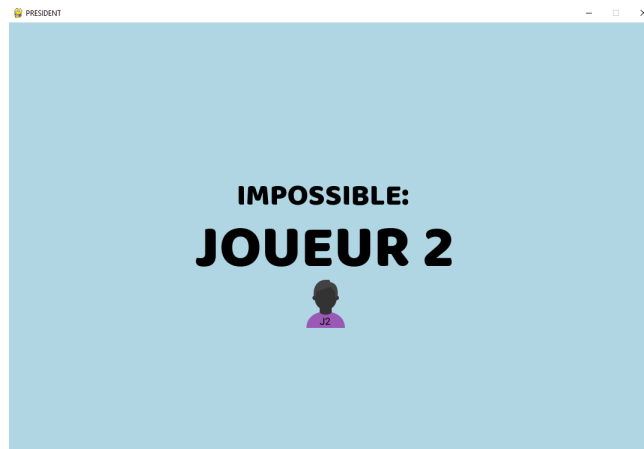
Durant la partie, il sera possible de couper ou rallumer la musique à l'aide d'un bouton situé en haut à gauche. Selon sa forme, vous saurez si le son est coupé ou non :



Il est également possible d'accéder aux paramètres avec un autre bouton situé en haut à droite. Une page s'ouvrira avec les règles du jeu et la possibilité de changer de thème en pleine partie. Cliquez sur la flèche en haut à droite afin de revenir à la partie :



Pendant la partie, si vous jouez des cartes qui ne sont pas possible (cartes trop basses ou nombres de cartes non respecté) une fenêtre vous informera que c'est impossible et vous aurez la possibilité de ré-essayer d'autres cartes ou de passer votre tour (avec le même bouton que pour jouer ses cartes) :



Lorsqu'il ne reste qu'un joueur en jeu, la partie se finit ; une page s'ouvre avec un bouton pour fermer la fenêtre Pygame :



6 Conclusion

Nous avons donc un jeu complet avec plusieurs fonctionnalités et choix de thèmes. Nous avons également réussi à implémenter presque toutes les règles telles que le "*carré magique*", le joueur qui ferme le tour, ouvre le suivant, le "2" est la meilleure carte, l'obligation de poser le même nombre de cartes que le joueur précédent...etc

C'était bien l'objectif que nous nous étions fixés et nous en sommes fiers, mais pour améliorer notre jeu nous aurions pu faire un système de partie en

boucle (c'est-à-dire, le président de la première partie devrait donner 2 de ses pires cartes au trou du cul à la seconde partie), ou encore la règle du "*ta gueule*"