



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à info@trophees-nsi.fr.

NOM DU PROJET : ShiFuMi

> PRÉSENTATION GÉNÉRALE :

- Nous avons simplement en tête de recréer le jeu célèbre du « Pierre Feuille Ciseaux » mais via un programme python et une interface graphique, grâce au module tkinter.

A la différence du simple jeu, nous voulions créer 2 modes : 'libre' (le joueur arrête la partie quand il le souhaite) et 'challenge' (le joueur choisit un nombre manches à gagner par le vainqueur). Nous avons également pour espoir de créer une variante « Pierre Feuille Ciseaux Léopard Spock » avec, elle aussi, 2 modes d'exécutions, mais à notre grand regret le temps nous a manqué.

D'autres idées comme une application avec beaucoup de mini jeux condensés (dont le shifumi) nous avait heurté mais la date butoir approchait.

- Par manque d'originalité et un intérêt pour les jeux, l'ensemble du groupe c'est dessiné à réaliser un jeu célèbre dans la vraie vie mais pas sur ordinateur.

Dans un sens nous voulions créer un programme divertissant et agréable à la réalisation et plus ludique pour nous familiariser avec le langage de programmation python en temps qu'apprentis.

> ORGANISATION DU TRAVAIL :

- *Présentation de l'équipe =*

- Vayana : cheffe de projet, a englobé et surveillé la majeure partie du projet et donnait les directives.
- Jérémy : aide principal à la programmation et supervision secondaire.
- Antoine : aide à la programmation et réalisateur de la vidéo de présentation.
- Max : co-réalisateur de la vidéo de présentation et designer de l'application.

- *Répartition des tâches =*

- Vayana : avait pour rôle la conception des fonctions principales du jeu et de relier le programme python et l'interface tkinter
- Jérémy : avait pour rôle la conception des fonctions liées au fonctionnement de l'interface (ex : quitter la fenêtre, validation d'un radiobutton sélectionné)
- Antoine : pouvait soulager Vayana et Jérémy de quelques tâches liées à la programmation mais était principalement dédié à la réalisation de la vidéo de présentation.
- Max : avait pour rôle l'imagination de l'interface graphique de la réaliser et soulageait Antoine de certaines tâches liées à la réalisation de la vidéo (ex : enregistrement vidéo d'une partie du jeu, capture d'écran de certaines fenêtres du jeu) .

- *Organisation du travail =*

Nous nous voyons environ 2 fois par mois au CDI du lycée, mais les réunions n'étaient pas réellement nécessaires en raison de la communication via le code commun sur google doc (qui n'était pas très pratique mais cela nous permettait d'être au courant de chacune des nouveautés). De plus via les messages et petites conversations en personne nous avons su nous y retrouver sans trop se voir en groupe.

LES ÉTAPES DU PROJET :

- *Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet) =*

Concernant le programme :

Au début de la réalisation du projet nous avons procédé à une réunion pour décider l'idée de jeu du programme, puis de la réalisation avec ou sans module, avec ou sans interface graphique du jeu. Nous en avons profité le même jour, une fois décidé, de se mettre d'accord sur l'agencement de l'interface graphique du ShiFuMi sur papier.

Vayana a commencé dès les débuts la programmation du jeu lui même (pierre gagne contre ciseaux etc), de même concernant Max, pour la recherche des images représentatifs des «armes» (= pierre, feuille ou ciseaux) et notamment pour donner une idée plus concrète sur ordinateur l'interface souhaitée.

Un peu plus tard, une fois l'idée de l'interface fixée Vayana a commencé à relier le fonctionnement du jeu à l'interface mais partiellement car Jérémy travaillait en parallèle sur les boutons et fonctions liées au bon fonctionnement du jeu et à la fluidité de ce dernier.

Après des tonnes de problèmes liées à l'interface, l'affichage des images et le contenu des fenêtres, le groupe a finalement réussi à relier la version du mode 'challenge', sa fenêtre de choix, et la version du mode 'libre' à un menu de choix des modes !

Enfin les dernières finalisations des commentaires du programme pour le rendre plus lisible et compréhensible a été réalisé par Antoine.

Concernant la vidéo :

Une fois le programme terminé, Max et Antoine se sont attelés à la réalisation de la vidéo, ce qui n'a pas pris trop de temps.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- *Avancement du projet =*

Le mode 'libre' est entièrement terminé avec le menu de choix des modes, de même pour le menu des choix des manches pour la version challenge mais le mode challenge rencontre des 'bugs' mais appartient au programme sans le choix des manches le code tourne comme souhaité.

Il resterait à faire un menu d'accueil principal

- *Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet*

Des tests hors du programme principal, des fonctions seules par exemples, ont été réalisés le long de la conception.

Concernant l'intuitivité de notre programme nos proches ont été livrés à eux-mêmes et à leur intuition pour comprendre le fonctionnement du jeu, à notre grande surprise ce test a été passé haut la main (=même chez les plus âgées).

- *Difficultés rencontrées et solutions apportées =*

-Lorsque nous avons essayé de relier les modes à la fenêtre de choix des modes via un bouton, l'interface et le jeu cessaient de fonctionner, c'est ici que nous avons appris l'importance des variables globales et locales et leurs placements. Nous avons donc réussi à régler le problème après quelques tests et recherches sur internet.

-Ou encore le choix des manches pour la version challenge, l'idée principale était de créer une «entree = entry» pour récupérer sa valeur, malheureusement par manque d'expérience et de pratique nous avons pas réussi à récupérer la valeur de l'«entree» même avec `entree.get()`, car notre fonction détruisait l'entree et sa valeur avant d'être récupérer. Nous savions l'origine mais pas comment régler le problème en lui même, par conséquent nous avons fini par imposé des choix de nombres de manches (5, 10 ou 15 manches).

> OUVERTURE :

- *Idées d'améliorations =*

Comme cité précédemment nous avons eu comme idée en fin de projet d'insérer notre programme dans un autre programme avec plein de mini jeux(ex : morpion, pendu, 4 à la suite etc...), afin de créer une fenêtre principale avec en choix tous ces mini jeux.

Mais également d'ajouter notre variante 'Pierre Feuille Ciseaux Lezard Spock' avec les 2 modes à notre programme.

- *Stratégie de diffusion pour toucher un large public =*

Nous pourrions créer une chaîne youtube montrant notre avancée sur notre grand projet des «mini-jeux», en interaction avec le public en leur proposant des idées de jeux et de solutions pour les éventuels problèmes rencontrés.

La réalisation d'un quizz via l'exemple de notre jeu serait aussi intéressant.

Et si notre chaîne rencontre un fort succès, pourquoi pas en faire une émission télé ? Dans laquelle on apprendrait à coder de petit programme de manière ludique et ouvert à tous.

- *Analyse critique du résultat =*

Nous pensons qu'il aurait falut plus représenter notre projet sur papier et de ne pas être passer à la pratique «directement», il aurait falut plus de théorie au début pour éviter et contourner un maximum de problèmes (ce qui nousn aurait été bien utile).

DOCUMENTATION

- *Spécifications fonctionnelles :*

Ouvrir le fichier python «SHIFUMI», vérifier qu'il se situe dans le même dossier que les images fournies, puis lancer le programme. De ce fait une fenêtre tkinter ouvrira et JOUER !

C'est à vous de découvrir le jeu, notre objectif était qu'il soit le plus intuitif possible, si ce n'est pas le cas nous avons échouez, à vous d'en jugé !

(au cas où : dans le menu principale, choisissez votre mode entre 2 bouton 'LIBRE' et 'CHALLENGE', la fenêtre se réinitialisera en fonction du mode choisis :

-Si vous avez choisis le mode 'LIBRE' le jeu s'affiche directement, cliquez sur n'importe quelle radiobutton (pierre, feuille ou ciseaux) et cliquez sur valider, le résultat s'affichera dans zone «pt(s) joueur :» et «pt(s) ordinateur:»

pour rejouer une manches tout en gardant le compte de vos point cliquez sur «rejouer»

-Si vous avez choisis le mode 'CHALLENGE' une fenêtre de choix de manche s'ouvrira vous avez le choix entre 5,10 et 15 manches, choisissez l'option qu'il vous convient puis cliquez sur 'valider'.

Maintenant le même fonctionnement s'opère que pour le mode 'LIBRE' mais à la différence que le jeu s'arrête quand le vainqueur atteint le nombre de manches choisis précédemment.

Si vous souhaitez faire tourner le jeu avec le mode challenge sans bugs dans le programme principal, vous devez ouvrir le fichier 'SIFUMI_challenge' et vérifier qu'il soit compris dans le même fichier que les images transmises.

Un message en fonction de votre statut s'affiche (statut = gagner ou perdu).

Ps: il faudra cliquer une fois de plus sur valider lorsque vous avez joué la manche gagnante pour obtenir le message en question.)

- *Spécifications techniques* =

Nous avons utilisé le langage de programmation python avec le module tkinter pour la réalisation du projet.

- *Illustrations, captures d'écran, etc* =

Partie de la partie du programme qui génère l'accueil du jeu :

```
def accueil_shifumi():  
    global fen1  
    zone_1= Frame(fen1, bg='#7DCEA0',height=45,width=350).grid(row=0,column=0,columnspan=3)  
    Label(zone_1,text='-CHOISIS TON MODE DE JEU-',bg='#7DCEA0',font=('Arial 17 bold'),fg='FFFFFF').grid(row=0,column=0,columnspan=3)  
    zone_2= Frame(fen1, bg='FFFFFF',height=25,width=350).grid(row=1,column=0,columnspan=3)  
  
    zone_3= Frame(fen1, bg='#7DCEA0',height=25,width=350).grid(row=2,column=0,columnspan=3)  
    Label(zone_3,text = '•LIBRE',bg='#7DCEA0',font=('Arial 10 bold'),fg='FFFFFF').grid(row=2,column=0)  
    Label(zone_3,text = '•CHALLENGE',bg='#7DCEA0',font=('Arial 10 bold'),fg='FFFFFF').grid(row=2,column=2)
```

ce qui en ressort :

