

Projet Final ISN 2016-2017

Informatique et Sciences du Numérique

Dossier : Alexis Melo da Silva

Autres membres du groupe : Lucas Drieux, Simon Avril

HORRIFYING PIRATE FROM HELL

- 1. Présentation du projet, origine, but et lancement du projet.**
- 2. Développement du projet : moyens et outils produits.**
- 3. Description de la démarche collaborative dans le groupe.**
- 4. Questions posées sur un plan plus sociétal et juridique.**
- 5. Évaluation finale du projet et perspectives de poursuite.**

1. Présentation du projet, origine, but et lancement du projet.

Nous cherchions à créer un jeu **facile d'accès** pour n'importe quelle personne ayant envie de jouer, qui soit **esthétiquement beau** et ayant la possibilité d'être joué en **multijoueur localement**. Le projet devait également être **original** par son thème et son aspect, puisque l'originalité en terme de règles de jeu nous est **limitée** à cause de nos compétences en python qui ne nous permettent pas de faire des jeux totalement **nouveaux et trop ambitieux**.

Nous nous sommes alors tournés vers un **jeu de l'oie revisité**, car il est **simple** à jouer, potentiellement **multijoueur** mais aussi jouable **seul**, et le fait qu'il y ai un plateau avec des cases différentes ainsi que la possibilité d'y ajouter un fond nous permet de **personnaliser** le projet à **volonté**. De plus j'avais déjà l'idée d'un jeu de ce type avant même de commencer le projet, j'ai alors proposé à mes camarades de le réaliser ensemble, ce qu'ils ont accepté.

Pour trouver le thème du projet, nous avons utilisé le site <https://videogamena.me/> qui génère des noms de jeux vidéos **non-existant aléatoires**, cela a permis de nous donner divers sources d'inspiration et de se mettre d'accord sur un thème qui nous plaise à tous. Nous sommes alors tombés sur « **Horrrifying Pirate from Hell** », ont alors découlé les thèmes de la **Piraterie** et de la dualité **Enfer/Paradis**. Étant donné que le titre est **anglais**, on a préféré faire tout le programme en anglais pour avoir une concordance, en plus cela permet d'élargir le nombre de joueurs possible, le programme étant très **intuitif** même une personne non-anglophone peut jouer sans problème. Les règles du jeu sont simples : Un plateau avec des cases numérotées de 1 à 36, le but étant de partir de la case 1 et arriver à la case 36 en lançant un dé à tour de rôle. Certaines cases ont des bonus, d'autres des malus. Pour jouer il suffit de cliquer sur un bouton qui lance le dé.

Le but final est d'avoir un programme **fonctionnel** jouable **d'une à quatre personnes**, prenant en compte le **nom** et la **couleur** des joueurs pour les **différencier**, le programme doit aussi être **beau**, car c'est aux premiers abords ce qui donne **envie d'y jouer**, et totalement **personnalisé**.

2.1 Développement du projet : moyens et outils produits.

Le projet a entièrement été codé avec **Python**. Les modules utilisés sont « tkinter », « random », et la fonction « import os ». Des logiciels de retouche photo et de dessin ont été utilisés, tels que **paint**, **paint.net**, et **photoshop**.

Avant de commencer le projet sous format numérique nous avons d'abord réalisé des **croquis** sur papier. On a tracé le plateau dans un quadrillage, et on a mis sur papier nos idées ainsi que ce qui doit être présent dans le programme.

Illustration détaillée : Annexe 1

Le programme devait alors comporter :

- 1 **page d'accueil** ou **menu principal**
- 1 **page de configuration** avec sélection du nombre de joueur, de leurs noms et de la couleur de leur pion.
- 1 **plateau de jeu**
- 1 **page de fin**

Nous avons alors commencé à développer le **jeu vierge**, avec juste un plateau, un dé, et un pion que l'on pourra personnaliser plus tard. Pour se faire on a utilisé la méthode « **grid** » de python pour disposer les éléments dans des cases en fonction de colonnes et de lignes, le croquis papier nous a bien aidé. On a réussi à créer un pion et le faire avancer à chaque fois que le dé est lancé en utilisant une **fonction** qui ajoute le nombre du dé à une **variable** de base égale à 0, cette variable correspond au numéro de la case sur laquelle atterrit le pion.

Illustration détaillée : Annexe 2

Ensuite nous avons réalisé un **menu principal** avec une image de fond, un bouton qui permet de « jouer » et un bouton « crédits » qui permet de voir les règles du jeu ainsi que quelques informations.

Illustration détaillée : Annexe 3

Hors il nous fallait un **intermédiaire** entre le menu principal et le plateau de jeu, alors nous avons réalisé la **page de configuration**. Elle permet de choisir le nombre de joueurs, grâce à des flèches, le nom des

joueurs qu'ils peuvent être eux-même grâce à des Label [Entry], et un choix de couleurs propre à chaque personnage lui aussi possible grâce à des flèches.

Illustration détaillée : Annexe 4.1 et 4.2

Le menu principal, la page de configuration et le plateau sont réalisés sur trois fichier python différent pour faciliter la mise en place et limiter les **erreurs possibles**. Il est possible de lancer un programme depuis un autre grâce au module « **import os** ». Le programme du menu principal et la page de configuration ont été réalisés sur des canvas, ils permettent une mise en place plus **précise** des images et des éléments (au pixel près), et il permettent de **mieux répondre** à notre **demande** par exemple quand nous voulons utiliser une partie d'une image pour en faire un bouton ou insérer des « Entrées » pour écrire le nom des joueurs. Le plateau quant à lui est réalisé avec une **grille** (grid) car cela **facilite** la mise en place des cases et des autres éléments même si cela amène moins de **précision**. Les déplacements du pion se font en ajoutant/diminuant la position du pion grâce aux colonnes et lignes (la fenêtre est découpée comme un échiquier invisible).

Lors d'un tour, le pion peut atterrir sur 5 types de cases :

- **Case Bonus** ; le pion avance de 3 cases
- **Case Malus** : le pion recule de 3 cases
- **Case raccourci** : le pion avance de 6 cases
- **Case Recul** : la variable qui définit le numéro de la case où se trouve le pion est divisée par 2, la position du pion est aussi divisée par 2
- **Case arrivée** : Le plateau se ferme, la fenêtre de fin s'ouvre

Lors d'un lancé de dé une **fonction** s'**active** et ajoute le nombre du dé à une **variable** qui défini le « **Rang du pion** » (correspond aussi au **numéro de case**, ou à **tout les lancés de dé cumulés**), à chaque fois il y a une **vérification** pour voir si cette variable n'est pas **égale** au numéro d'une case spéciale prédéfini, si elle l'est, alors une autre fonction est lancée et elle effectue **l'effet spécial** de la case.

2.2 Les problèmes rencontrés

- Le premier problème a été sûrement le plus **compliqué** à régler, il fallait faire se déplacer le pion selon la grille du plateau et surtout pendant les **virages**, le pion devant se déplacer dans toutes les directions. Parfois le pion se retrouvait sur des cases **en dehors** du plateau et sur des cases qui ne

correspondaient pas au bon chiffre.

Solution : en se référant au schéma de l'Annexe 1, il fallait découper la **méthode de déplacement** du pion pour en faire des **différentes** entre les cases 1 à 10, 11 à 12, 13 à 23, et ainsi de suite. On se retrouve avec une longue fonction mais peu importe le programme fonctionne parfaitement à ce niveau là.

- Le second problème a été de faire en sorte que l'arrivée se fasse **seulement** sur la case 36 et que si le joueur dépasse ce nombre il soit **renvoyé** en arrière, hors nous n'arrivions pas à faire coïncider le « surplus » de cases et de renvoyer le pion au bon endroit.

Solution : Rajouter un « if » quand le dé est lancé pour les cases **supérieures** à 36, on crée une **seconde variable** égale à la première (celle qui correspond à tout les lancer de dé cumulés et au numéro de la case) mais **à laquelle on retire** 36 (le nombre total de cases), il reste alors le surplus, on enlève ensuite ce surplus à la toute **première variable** et le pion peut être placé facilement avec la **nouvelle valeur** déduite.

- Le troisième était de ne pas avoir d'espace entre les cases du plateau car quand une image est plus grande, la case de la grille (et donc de toute la ligne + toute la colonne) s'y adapte et tout est décalé.

Solution : on fait 4 images « fix » **invisibles** que l'on place dans les coins du plateau.

- Le quatrième était d'obtenir la couleur, le nombre et les noms des joueurs dans le plateau car la page de configuration et le plateau sont 2 **programmes différents**.

Solution : on utilise les **fichier texte**, quand on appuie sur « Start » dans la page de configuration pour lancer le plateau, le nombre de joueurs, les couleurs des joueurs(que l'on fait correspondre à un chiffre entre 1 et 4) et leurs noms sont écrits dans 3 fichier textes différent **utilisables** depuis le second programme.

- Le cinquième était de faire **correspondre** la couleur du joueur à la couleur du pion, soit le programme **plantait** soit il y avait un problème à cause d'une variable non-défini en « **global** »

Solution : on crée une fonction qui se lance **une seule fois** dans le programme qui crée les pions puis les fait apparaître sur le plateau, avant nous avions mis les « if » **en dehors** d'une fonction et donc cela ne marchait pas.

- Le sixième était de mettre les pions dans des coins pour ne pas qu'ils se **superposent** en multijoueur

Solution : il y avait un problème de syntaxe, il faut utiliser l'option « sticky » et les paramètres Nord-Ouest, Sud, etc..

3. Description de la démarche collaborative dans le groupe.

Lucas s'est principalement occupé des **images** grâce à photoshop et des qualités de dessinateur il a pu nous fournir des dessins en **concordance** avec les besoins meilleurs que ce que j'aurais pu fournir avec paint. Lucas et Simon ont travaillé sur d'autres tâches qu'on avait défini en groupe en classe.

Je me suis occupé de la **majeure partie de la programmation**, j'ai d'abord commencé par **intégrer l'image de Lucas** pour le menu principal. J'ai découpé une petite partie pour en faire un **bouton** qui permet de lancer le programme de la page de configuration. J'ai également fait **toute la partie de la page de configuration** excepté les images des pirates, j'ai d'abord fait une première fenêtre qui permet de choisir le nombre de joueur en cliquant sur des flèches.

Illustration détaillée : Annexe 4.1

Puis une fois le nombre choisit, on peut cliquer sur « Entrer », cela vide le canvas et nous affiche d'autres **paramètres réglables** ; la couleur du personnage en fonction de chaque joueur (réglable avec des flèches cliquables) et le nom de chaque joueur qui doit être entré **manuellement** dans des cases. Une fois le tout réglé on appuie sur « Start » et le jeu se lance.

Illustration détaillée : Annexe 4.2

Je me suis également occupé du plateau. J'ai codé la **partie fonctionnelle en complétant** le dé de Lucas et Simon, en ajoutant les **conditions** qui permettent le déplacement du pion. J'ai également fait les fonctions qui assignent les couleurs des fichiers texte, les noms dans les fichier textes, et le nombre de joueurs aux **endroits appropriés** sur le plateau (à droite). J'ai jugé bon d'**ajouter une colonne** avec les noms et les couleurs de chaque joueur pour mieux s'y retrouver avec à leur droite le « **rang** » qui correspond au **numéro de la case** où ils se trouvent pour savoir qui est **en tête**. J'ai aussi ajouté une partie en bas du plateau qui **change en fonction de la case** où se trouve le pion avec une petite **citation** et l'**influence** de la case sur la position du pion. J'ai aussi dessiné les 4 motifs possibles sur les cases. J'ai également redessiné le dé pour qu'il soit **en accord** avec notre thème, j'ai redessiné les différents pions aussi, et j'ai ajouté une fonction qui affiche sur le plateau **à qui est le tour** de jouer.

Illustration détaillée : Annexe 5

J'ai finalement ajouté une **exception** sur la case finale (36) qui quitte le plateau et affiche l'écran final quand un joueur a gagné.

J'ai aussi pris soin d'ajouter des boutons « quitter » et « menu principal » dans **chaque partie** du programme pour qu'il soit jouable en **boucle** et qu'on puisse le **quitter ou revenir en arrière** à tout moment.

Il existe en tout **4 programmes** qui servent à faire le plateau, un pour chaque nombre de joueurs(entre 1 et 4 suivant le besoin), cela évite d'avoir un fichier python **surchargé** de toutes les fonctions, et cela améliore la **lisibilité**. De toute façon le fait d'ajouter des joueurs n'est pas compliqué, mais cela prend **énormément de temps** car il faut adapter toutes les fonctions et tout copier/coller dans chaque programme.

4. Questions posées sur un plan plus sociétal et juridique.

Selon la loi (source : <https://www.photobox.fr/contenu/propriete-intellectuelle-droit-image>) :

« Toute représentation ou reproduction intégrale ou partielle, traduction, adaptation, transformation, arrangement d'une œuvre réalisée sans le consentement de l'auteur ou de ses ayants droit est illicite (article L 122-4 du CPI). La personne qui reproduit sans autorisation de l'auteur une œuvre sur un serveur Internet pour mettre celle-ci à la disposition du public commet un acte de contrefaçon (articles L 335-2 et L 716-9 du CPI). La contrefaçon est un délit civil et un délit pénal. La contrefaçon couvre toutes les reproductions et diffusions illicites c'est-à-dire non autorisées. »

Pour **respecter les lois** prévues pour la protection de toute marque déposée et éviter tous **problèmes de copyright**, chaque dessin de notre projet a été réalisé **par nos soins**, la police d'écriture sur les menus est **libre de droits** également. Tout a été réalisé comme dit précédemment pour obtenir un **projet personnel et unique** qui reflète bien l'image que nous avons du jeu au départ. Le jeu de l'oie était un **classique** des jeux de société à l'époque, son **concept** est aujourd'hui totalement libre de droit lui aussi, de nombreuses variantes existent. On ne prétend pas créer un jeu **révolutionnaire** mais nous sommes **satisfait** d'avoir réussi à **transformer** un jeu **traditionnel** français en jeu **numérique original** de par sa forme et par le support sur lequel il est présenté (informatique).

5. Évaluation finale du projet et perspectives de poursuite.

Évaluation finale du projet

A l'heure où nous rendons le dossier le projet n'est pas **entièrement fini**, ce qui est assez dommage. Le jeu est **entièrement jouable à 1 joueur** mais il manque la réalisation des programmes pour jouer à 2,3 et 4. Néanmoins je suis assez satisfait du résultat final, même si il reste des choses que je pourrais améliorer avec plus de temps et des **compétences** en python supplémentaires voir même l'aide d'autres personnes ou le passage sur un **autre langage de programmation**. Le projet est **original** comme je l'avais imaginé, on reconnaît bien le principe du jeu de l'oie mais l'aspect graphique particulier me plaît. J'ai appris à travailler en groupe même si la répartition du travail était parfois **inéegale** mais cela m'a appris à **redoubler d'efforts** et à chercher des solutions par **moi-même**. De plus j'ai appris de nombreuses choses sur python utiles pour ma poursuite d'études (DUT Informatique)

Perspectives de poursuite

Perspectives qui vont être sûrement réalisées après avoir rendu le dossier :

- Finir entièrement le programme pour qu'il soit fonctionnel en multijoueur, au moins à deux.
- Ajouter le nom du joueur gagnant sur la fenêtre de fin

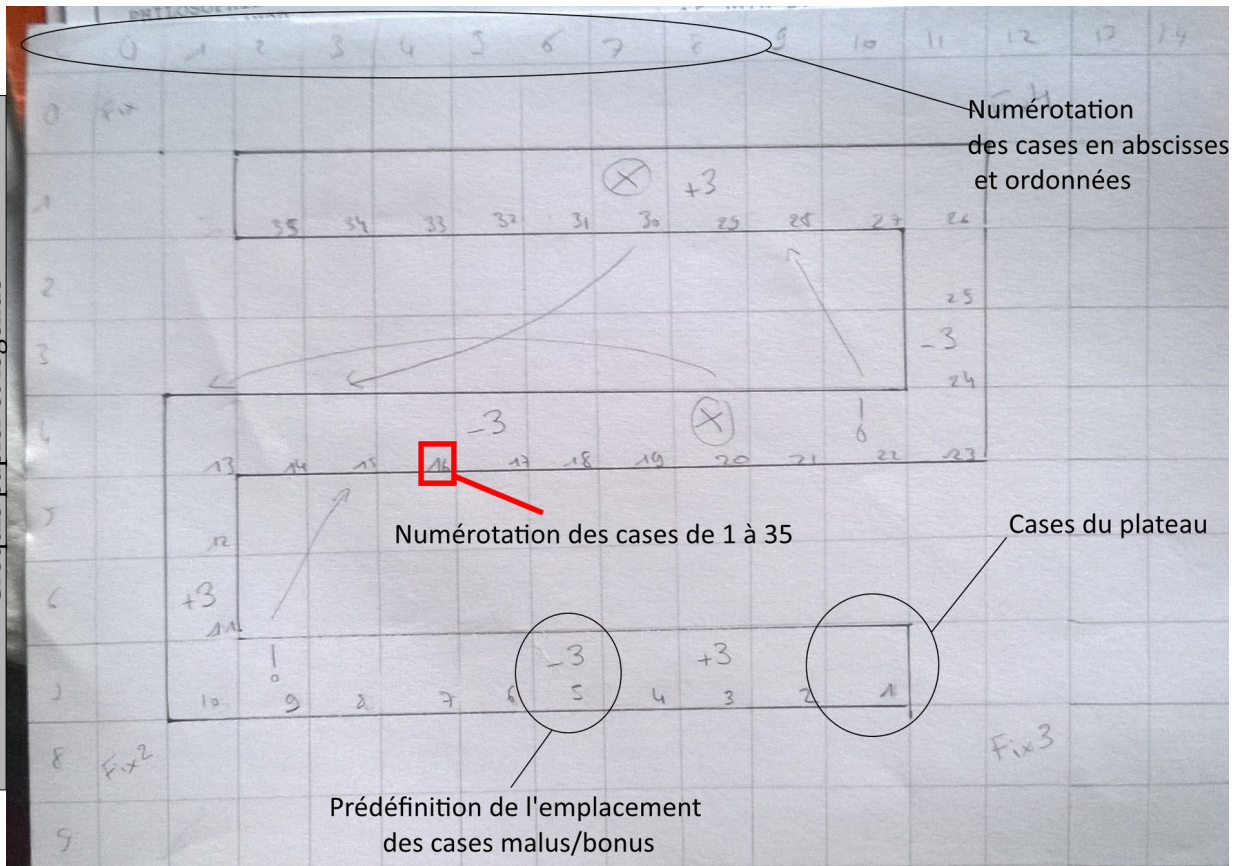
Perspectives qui auraient pu être mises en place avec plus de connaissances et de temps :

- Ajouter des types de cases telles que des cases mini-jeux.
- Rassembler les 4 programmes du plateau en un seul, voir même rassembler tous les programmes en un seul
- Créer un .exe pour lancer le programme directement sans avoir à préciser lequel lancer avant.
- Ajouter une musique d'ambiance
- Ajouter des sons quand on clique sur les boutons
- Ajouter des animations pour le pion (marquer un petit temps de latence pour qu'on puisse remarquer qu'il avance case par case)
- Refaire tout les dessins en meilleure qualité et si possible en format .png plutôt qu'en .gif comme imposé par la version de python du lycée
- Ajouter plus de couleurs disponibles

6. Annexes

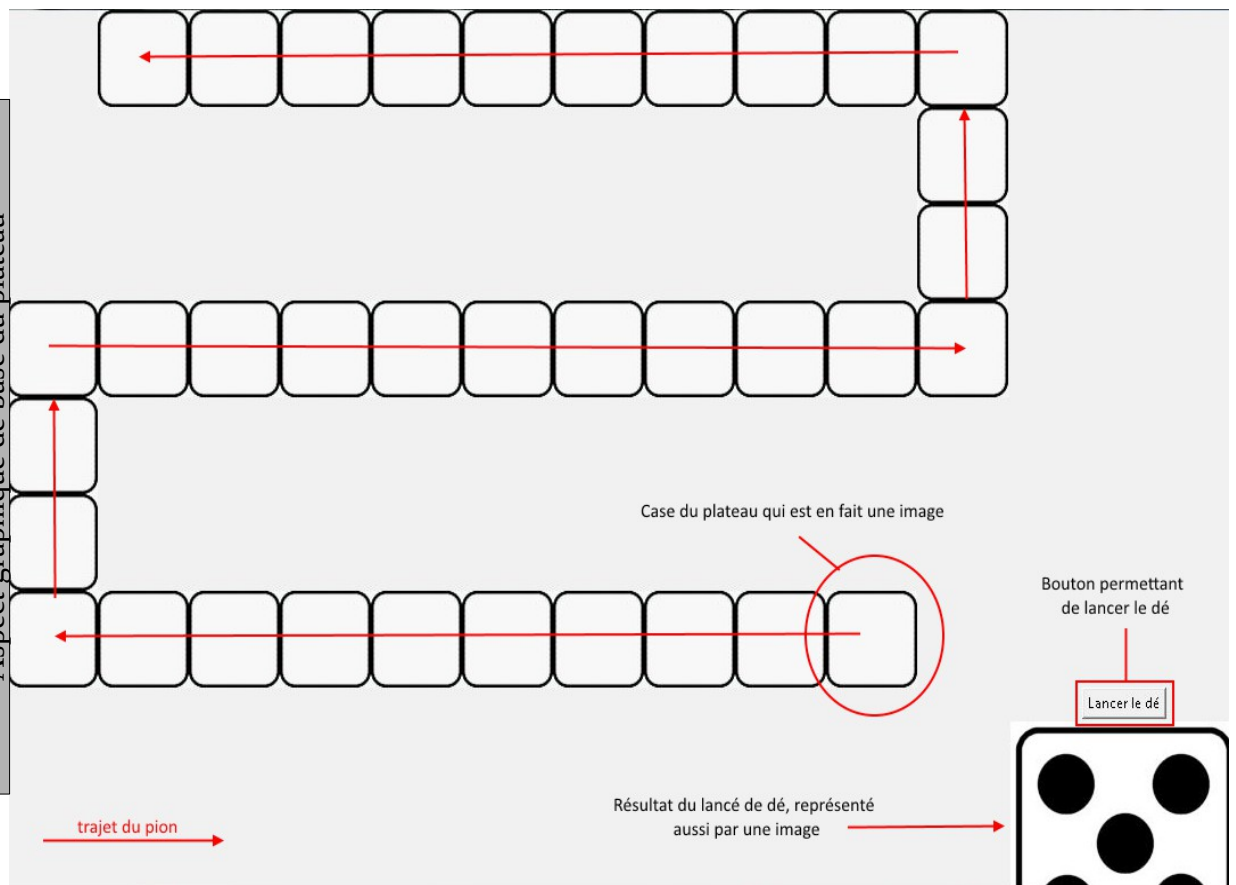
1

Croquis papier et légende



2

Aspect graphique de base du plateau



3

Image du menu principal



5

Plateau de jeu

