



---

**NOM DU PROJET : The Little Fram**

## > PRÉSENTATION GÉNÉRALE :

- Dans un premier temps, l'idée est venue de créer un jeu de gestion simple en deux dimension car c'est un style de jeu qui semblait relativement abordable à programmer.
- Après réflexion nous sommes arrivé à la conclusion qu'un jeu de gestion de ferme (d'agriculture) semblait être une bonne idée accessible et compréhensible par tous.
- Nous avons donc décidé de créer un jeu de ferme en vu du dessus permettant de faire pousser du blé et de le récolter, cependant cela paraissait trop simple, nous avons donc décider qu'il y aurait aussi des corbeaux qui viennent picorer les champs, portant atteinte à vos précieuses récoltes.

## > ORGANISATION DU TRAVAIL :

- Notre groupe :

Filipe → A proposé l'aspect du jeu et les éléments possibles d'incorporer.

Étienne → Apport d'idées au projet, motivation et gestion

Marcus → Programmation du jeu

Dorian → graphisme, audio, vidéo / compte rendu

- Les taches de notre groupe on été reparties selon la motivation et les compétences de chacun.
- Nous avons échanger durant des heures de cours pour faire un point sur l'avancée de notre projet. Le Weekend le travail était reparti pour faire avancer le projet (difficulté pour Filipe sans ordinateur). Dorian et Marcus ont réalisé des vocaux discord de concertation. L'ensemble du groupe échangeait sur un serveur discord et déposait ces documents sur un Google drive, ainsi que des fiches d'avancée sur un Google docs que tout le monde consultait.

## LES ÉTAPES DU PROJET :

- (idées trouvées comme dit précédemment)

Par la suite une concertation a eu lieu sur comment faire, que doit on utiliser...

Dorian a ensuite commencé à faire des graphismes (images a intégrer au jeu) pour permettre la visualisation de certaines fenêtres du jeu, Marcus a commencé à creuser l'aspect animation des fenêtres, actionnement des bouton, ajout de son, insertion d'image...

Étienne a été mis sur la tâche de la création d'une horloge, nous nous sommes rendu compte plus tard que ce n'était pas nécessaire car la bibliothèque python que nous utilisons pour le jeu (pygame) s'occupe déjà de cela. Filipe quand à lui a essayé d'imaginer a quoi pourrait ressembler le jeu au fur et à mesure de l'avancée.

**1<sup>ère</sup> étape** : Dans un premier temps, nous avons cherché une idée de projet qui nous convenait à tous : nous avons donc décider de faire un jeux de gestion de ferme.

**2<sup>ème</sup> étape** : Dans un second temps, il nous a fallu imaginer le jeu, à quoi il ressemblerait, ce qu'il y aurait à faire à l'intérieur etc...

**3<sup>ème</sup> étape** : Par la suite, nous avons réfléchi à comment réaliser le jeu, quels outils utiliser pour programmer le jeu : nous avons décider d'utiliser Pygame car c'est une bibliothèque python qui semblait simple d'utilisation à première vue. Il a donc fallu que Marcus (le programmeur du projet) apprenne à utiliser la bibliothèque Pygame.

**4<sup>ème</sup> étape** : Ensuite, nous avons programmé le jeu en lui donnant vie avec des graphisme fini et ajout de l'audio.

**5<sup>ème</sup> étape** : Puis, nous avons fait de nombreuses améliorations (surtout en terme de performance), résolu de nombreux bugs, et nettoyé le dossier du jeu.

**6<sup>ème</sup> étape** : Et pour finir nous avons rédigé ce compte rendu.

## > FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- Pour notre part lors de la programmation nous avons préféré rester sur un jeu plus simple dans la limite de temps que sur la frustration de ne pas avoir abouti sur notre jeu ou laisser une étape en cours de programmation. Évidemment une multitude d'améliorations auraient pu avoir lieu et nous ne manquons pas d'idées, c'est le temps qui nous manque.
- Pour éviter les bugs une batterie de test a été effectuée et il semblerait que tout fonctionne de la manière dont il a été programmé. Nous avons fait en sorte que le jeu soit assez simple en terme d'interface graphique ce qui permet aux personnes jouant de bien se repérer dans l'espace graphique et de ne pas se perdre dans la complexité.
- En terme de difficulté nous avons cruellement manqué de temps en commençant en mars et au milieu des épreuves bac, oraux et bac blanc. Il y a aussi eu le souci des compétences de chacun certain était moins motivés et en faisaient le moins possible quand a d'autres qui y passaient la nuit. L'égalité de travail ne fut pas respectée mais que faire quand il y a une date butoir il faut la respecter et la patiente peut avoir des limites. D'autre part nous avons TRÈS fortement sous-estimé la difficulté à programmer un jeu vidéo, même si il s'agit « uniquement » d'un petit jeu de gestion.

## > OUVERTURE :

- En terme d'améliorations possibles nous avons de nombreuses idées que nous n'avons pas eu le temps de mettre en place: des cultures plus variées, des bâtiments avec des fonctionnalités différentes, une fluctuation des prix des récoltes, d'autres fermiers concurrents etc...
- Évidemment si nous devons promouvoir notre jeu nous utiliserions les réseaux sociaux, car ce sont des plateformes qui permettent de toucher un large public. The Little Farm est un jeu ludique. Il peut intéresser certaines personnes attirées par les jeux de gestion peu compliqués et est accessible car il ne nécessite pas une puissance de calcul phénoménale.
- Si c'était à refaire nous aurions aimé avoir plus de temps. Nous aurions pu nous organiser mieux et travailler plus efficacement en groupe car certains ont eu une charge de travail bien plus élevée que d'autre, il y avait donc un problème de gestion et/ou de motivation chez certains. Sur notre jeu certain points graphiques ne sont pas super beau. Et évidemment ne pas être bloqué que sur python !

## DOCUMENTATION

- Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres) :

Pour pouvoir lancer le programme vous devrez disposer de python et de sa bibliothèque Pygame installés. Le programme se lance avec le fichier mainV2.py qui fait appel à la classe Jeu (classe principale du jeu, elle contient les boucles de détection d'input des différentes fenêtres et surtout la boucle principale du jeu), chaque écran du jeu dispose de sa propre classe qui contiens toutes les définitions de boutons, de widgets, de methodes nécessaires dans cette classe, les fonctions des boutons etc..

- Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc.) :

Nous avons décidé de programmer le jeu en langage python, d'une part car c'est celui que nous maîtrisons le mieux (même si nos connaissances restent limités), d'autre part car le temps limité qui nous a été confié pour compléter le projet (environ 2 mois entre-mêlé de révisions pour le bac, bac blanc et tout simplement les cours) ne nous a pas permis d'apprendre et d'utiliser un langage plus adapté aux jeux vidéos (je pense au C#, C++ et Java).

Nous avons utilisé Pygame, qui est une bibliothèque python permettant de créer relativement facilement une fenêtre de jeu, de capter les input clavier/souris, d'intégrer des sons et des images.