

Implication individuelle :

Nous avons réalisé en équipe le cahier des charges pour se mettre d'accord sur le projet et les actions principales que nous voulions avoir dans notre jeu de Tower Defense.

Le découpage fonctionnel a été réalisé tous ensemble, nous avons essayé d'avoir une vue globale mais nous n'étions pas toujours en accord sur le placement de certaines fonctions dans les différentes classes. Notre découpage fonctionnel a été amélioré au moment du codage, il est donc différent sur certains points de celui rendu à mi-projet.

Concernant l'implication individuelle de chacun, voici la répartition des tâches principales :

- Omar s'est principalement occupé du code principal (déroulement du jeu) et les classes Ennemis et Tours
- Jérémie a créé les fichiers classes Mapp et Niveaux, tout en créant les fichiers excels associés. A aussi aidé pour les classes Tours et Ennemis.
- Hugo a mis en forme la map sur l'interface graphique ainsi que l'interaction avec la souris.
- Juliette a réalisé l'interface graphique de la page d'accueil et la page de jeu sur tkinter.

Nous considérons que nous nous sommes tous autant investis durant les séances de projet afin de mettre en œuvre notre application, chacun s'occupant de son domaine avec ses compétences, à son rythme.

Notre implication individuelle est donc chacun de 25%.

Utilisation de l'intelligence artificielle :

- Pour la création du code, le fichier Map.py a été en partie créé à partir de chat GPT, afin d'accélérer la création du code pour l'ouverture d'un fichier xlsx et sa lecture avec une extension qu'on ne connaissait pas.

Prompt utilisé :

« Peux-tu me donner un code python lisant un fichier Excel et le transformant en liste 2D avec l'extension openpyxl. »

De plus l'IA m'a donné les commandes possibles pour récupérer la couleur d'une case

- Le fichier Wave.xlsx a été généré par chat GPT afin de créer des vagues d'ennemi évoluant de manière exponentielle au niveau de la difficulté.

Prompt utilisé :

“Je fais un tower defense en Python. Je veux générer 50 vagues d'ennemis dans un fichier Excel. Chaque ligne = 1 tic, avec une seule colonne à 1 (ça indique l'ennemi qui spawn). Y'a 6 types d'ennemis avec des PV : [30, 50, 40, 100, 80, 300]. Les gros ennemis (types 4, 5, 6) arrivent plus tard (vague 10, 15, 20). Je veux que les PV des vagues augmentent de façon exponentielle. Les 5 premières vagues seront remplies de plein de petits ennemis. Ensuite, on alterne entre ça et des vagues avec peu d'ennemis mais des gros. Parfois on peut mélanger. Et je veux aussi des tics vides de temps en temps. A la vague 20, il faut donc qu'il y est 100 ennemis environ.”

- Pour économiser du temps, nous avons également en partie généré les dockstrings grâce à l'IA puis nous avons apporté quelques corrections lorsque l'ia avait du mal à comprendre l'intérêt de certaines fonctions.

Prompt utilisé :

« Peux-tu générer les dockstrings des fonctions dans ce code ? »

- Pour améliorer l'interface avec l'utilisateur, nous avons également demandé de l'aide à une IA pour générer le code permettant de créer des sprites à partir d'une palette de sprites de taille 16x16 ou 20x20 pixels trouvée en ligne.

Prompt utilisé :

« Peux-tu générer le code permettant de créer des widgets de taille 16x16 et de les placer sur un canvas Tkinter à partir d'une palette de sprites sous format png ? »

- Nous avons également généré la palette « tilset3.png » et l'image « sang.png » grâce à l'IA.

Prompt utilisé :

« crée-moi un tileset de tower defence avec des sprite de 20 pixels par 20 pixels il doit contenir des forêts, montagnes, chemins, pont, tour d'archer, tour de mage, baliste, tour de feu et muraille »

« crée-moi une image 20x20 représentant une tête de mort et du sang a afficher sur les cases où des ennemis sont morts »

Utilisation de ressources externes :

- Pour l'algorithme A*, nous avons repris le code du P2i6, des membres de l'équipe en font partis. Nous avons repris le code de cet algorithme, en changeant la condition interne au choix du prochain sommet.
- Nous avons trouvé les fichiers « tilset.png » et « fond_ecran.png » sur Itch.io.