Le code est divisé en deux : Le code du jeu en général et le code gérant la carte et le joueur. Dans le code principal, on retrouve :

- main.py est le fichier principal du jeu. C’est celui qu’il faut exécuter pour pouvoir jouer au jeu. Il contient la boucle principale du jeu :

- On retrouvera dans la plupart des fichiers les trois parties qui sont gérées dans cet ordre : 1.Gestion des inputs du joueur, 2.Mise à jour de ce qui doit être affiché et 3.Affichage.

- Menu.py est le fichier gérant tout ce qui touche au menu d’accueil du jeu. Les commentaires précisent dedans les 5 parties du fichier :

- L’initiation de la classe avec le chargement des image et police d’écriture.

- Le chargement du menu, permettant de charger le menu à tout moment, que ce soit en lançant le jeu ou en quittant la partie en cours.

- Les trois suivantes sont les trois parties décrites ci-dessus et les fonctions correspondantes sont appelées à partir de main.py.

- Game.py est le fichier gérant tout ce qui peut se passer lorsque la partie est lancée. Il est divisé de façon identique au fichier Menu.py. C’est ce fichier qui appellera tous les autres fichiers utiles pendant la partie.

- Savemgr.py est le fichier gérant les sauvegardes. Il ne faut pas le confondre avec le fichier suivant :

- reset.py est un fichier permettant d’altérer les sauvegardes hors du jeu. Il n’est appelé par aucun autre fichier et il peut être utilisé notamment pour remettre toutes les sauvegardes dans un état souhaité (notamment pour reset le jeu), avec sa fonction set\_all().

- Les fichiers Audio.py et Characters\_sprites.py s’occupent de stocker et gérer respectivement les fichiers audios joués par le jeu et différents sprites utilisés.

- Commands.py n’est utile que pour le développement ou le test du jeu. Il gère les commandes. Il est appelé et traité directement par le fichier Game.py à l’aide de la fonction execute().

- Le fichier Cinematics.py contient les fonctions permettant de créer les cinématiques et dialogues du jeu, notamment avec la fonction cinematic\_frame(). Il contient également les fonctions lançant les cinématiques, qui peuvent être appelées directement depuis le fichier. Le fichier contient une mini boucle de jeu indépendante, car les cinématiques sont des programmes à part entière.

- Les fichiers mini-jeux sont également des programmes à part entière. Ils sont séparés dans leur propre fichier.

- Gameplay.py gère les lancements des phases de gameplay en fonction des circonstances.

- Loading.py sert à afficher des images sur l’écran pendant le chargement du jeu.

- Autres fichiers utilitaires.

Dans le code de la carte(map/src), la plupart du code est emprunté (et modifié) du youtubeur Graven, on retrouve :

- game.py qui contient également une boucle de jeu secondaire. Il est à noter que la carte ne fonctionne pas de façon autonome, bien que cela sera possible prochainement. Le fichier gère les autres fichiers en .py présents.

- map.py s’occupe de l’importation de la carte, de la gestion des éléments de ces cartes et de leurs interactions.

- player2.py s’occupe du joueur et prochainement des PNJ (personnages non joueurs) qui seront présents sur la carte.

- animation.py s’occupe d’animer les sprites.

- Interactions.py gère les interactions des PNJs

- Map\_objects.py est l’ensemble des classes représentant tout ce qui peut être affiché sur la carte.