

Spong
Bertine Thomas

I. Le départ

L'idée de coder un jeu tout seul et pour la première fois était excitante mais aussi terrifiante pour moi. Je voulais donc rendre un hommage au jeu pong mais je n'avais aucun repère sur Pygame en dehors d'un usage un peu "mathématiques" de Python. Je repoussais ce fameux départ, passant des jours à me documenter, à bloquer devant mon premier fichier Python vide dans lequel seul Pygame était importé. Ce départ était quelque peu difficile pour moi.

Je réalise maintenant que la NSI nécessite une méthode de travail différentes de ma part car cette matière demande un investissement important en temps (surtout quand on est étranger à ce milieu) que je n'ai pas bien su donner. Ainsi s'explique mon retard, je n'ai pas réussi à bien m'organiser pour rallier mes cours à mon projet, par exemple, un DS repoussait le moment de ce mettre au travail. Toutefois, je me suis documenté de manière à maîtriser certaines bases qui m'ont permis d'avancer beaucoup plus vite ses derniers mois malgré que le jeu demeure inachevé.

Ce projet est bien plus que ça, c'est un premier pas dans l'informatique et également une espèce d'aventure.

NB : Mon regard sur la conception des jeux vidéo a changé.

II. Les difficultés rencontrées

Mon voyage dans la programmation orientée objet a été ponctué de nombreux obstacles qui m'ont mis dans tous mes états. Premièrement, après avoir conçu un fichier TMX qui avait pour but de simplifier la tâche de travail n'a fait que l'aggraver. Des objets rectangulaire pouvait être utilisé pour créer des collisions mais le passage de Tiled a été une catastrophe, me forçant finalement à faire des rects manuellement.

De plus, l'animation fût une véritable découverte, car je comprenais qu'une fonction/méthode qui s'active après un évènement devait continuer à fonctionner sans ce dernier pour qu'on l'on est l'illusion d'un mouvement et non d'une téléportation instantanée. Ainsi j'ai donc eu l'idée d'implémenter des arguments dans toutes les méthodes qui impliquaient des mouvements puis de faire retourner ces même arguments par ces méthodes. De ce fait les méthodes ont toujours un argument qui leur permet de marcher même après un évènement (par exemple la balle qui a un mouvement continue après qu'elle touche un mur. Par ailleurs, ce n'était pas vraiment une difficulté mais le fait d'avoir oublié de mettre la méthode `save_location` pour le deuxième joueur m'a profondément marqué, je passais des heures à chercher une petite erreur dans un code qui prenait de grandes proportions pour moi. Mais ce n'est pas fini, car les problèmes de batterie de l'ordinateur de la région se sont également manifestés, j'ai réussi à perdre 2h de travail sur une malheureuse batterie qui coupe l'ordinateur vers les 60%.

III. Un jeu inachevé

Malheureusement, mon projet est inachevé : Contre Jean est mal animé (d'où le problème de hitbox), il n'y a pas les zones qui devaient infliger des dégâts, etc... Néanmoins, il a tout

de même était remanié pour offrir une expérience de jeu : les joueurs doivent éliminer leurs adversaires en leur envoyant la balle dessus.

Conclusion

Ce projet m'aura appris beaucoup de chose sur la programmation orientée objet comme le concept de classe, de méthodes, d'instances, de threads (même s'il n'y en a pas). J'ai eu l'occasion de discuter avec des gens qui m'ont beaucoup aidé comme les personnes du serveur discord Pygame mais aussi mes chers camarades.