

RAPPORT DU PROJET 3

L'idée du projet est de concevoir une interface graphique sous forme de labyrinthe, dans laquelle personnage de Mac Gyver doit pouvoir se déplacer, attraper un certain nombre d'objets et rejoindre le client pour pouvoir sortir.

Difficultés rencontrées

Au début de mon projet j'ai rencontré quelques difficultés dans l'acquisition des connaissances très variées : Pygame ; environnement virtuel ; programmation orientée objet.

Affichage de l'interface

- La première étape consistait à l'affichage des éléments dans la fenêtre du jeu ; ces éléments qu'on appelle les « sprites ». Pour afficher ces éléments, on doit tout d'abord charger l'image grâce à la méthode Load ; ensuite on doit afficher cette image et la placer à l'endroit approprié.

le déplacement et la positionnement de joueur

- La deuxième étape consiste à faire bouger un sprite : le sprite se déplace de 40 pixels en ordonnée ou en abscisse, pour cela j'ai donc mis pour les cases une dimension de 40 pixels par 40.

Le déplacement et la positionnement de joueur

- cette étape consiste à gérer les collisions avec les murs ; le joueur doit respecter le bon cheminement, et pour cela il faut gérer correctement les chemins de déplacement.

Après, il fallait définir dans un fichier texte le niveau de jeu, qui est représenté ainsi :

- Le X représentant un mur
- Le O représentant une case où le joueur est permis à se déplacer dessus

Quand on va lancer le programme, le fichier sera lu automatiquement ceci afin de retrouver les cases qui sont autorisées et les stocker.

Pour ce qui est des éléments à récupérer par le joueur, le programme va simplement piocher au hasard trois cases dans la liste des cases autorisées. Ensuite le programme se chargera de les afficher à l'écran.

Les classes :

1-AgentScreen

Possède en attribut un constant élément qui contient la liste des images à charger au

Démarrage du programme.

2-class ActionGame:

Gère tout le mécanisme du jeu, à savoir le déplacement du joueur, ainsi

Que la vérification des objets à récupérer. Lorsque le joueur se présente devant le gardien, la méthode `check_end_game()` est appelée pour contrôler si le joueur a tous les objets. Si

Oui, alors il gagne la partie, autrement il perd la partie.

3-class Labyr:

Se contente simplement de récupérer le niveau du labyrinthe à partir du

fichier data. Puis fournira la liste des cases autorisées sous forme d'une liste de tuples à la

classe **AgentScreen** via la méthode `get_map()`



Lien de code :

<https://github.com/pic0de/pygame/tree/master>