**speech\_bubble.py**

**Provenance :**

Création entièrement originale.

**Utilités :**

* Afficher les dialogues à l’écran

**Fonctionnement :**

La classe « Dialogues » est celle utilisée dans l’importation des fichiers texte dans le programme « settings.py ». Un dialogue est caractérisé par le lieu et l’étape où il doit être dit et par son texte. Il peut également faire avancer l’histoire ou donner un indice au joueur avec les méthodes « AddStep » et « AddHint ».

La classe « SpeechBubble » est légèrement plus compliquée. Cette dernière s’occupe entièrement de l’affichage des boîtes de dialogues.

Sa première méthode, « initialize », permet de créer un nouvel affichage de dialogue, en utilisant presque toutes les autres méthodes de la classe. On va en effet calculer les bonnes proportions à respecter lors de l’affichage des boîtes de dialogue. On va également créer une police d’écriture, puis séparer les répliques en boîtes de réplique et ces boîtes en lignes. Ensuite, on affichera la première boîte de dialogue avec l’image du personnage qui parle.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Comme vous avez pu le comprendre, « create\_font » crée une police d’écriture pour les dialogues, « split » découpe les répliques en boîtes de dialogues et en lignes en fonction de la place qu’il y a (si un mot dépasse de l’écran, il y a un retour à la ligne, ou même un changement de boîte de dialogue, et « create\_text » permet de créer le rendu de 3 lignes de répliques.

La méthode « update », quant à elle, actualise le dialogue en ajoutant 1 à l’index de la boîte de dialogue, tout en vérifiant si l’on n’est pas arrivé à la fin du dialogue.

Ensuite, la méthode « create\_face » permet d’afficher l’image du personnage qui parle, c’est-à-dire le prénom au début de chaque répliques.

Enfin, la méthode « draw » permet d’afficher la boîte de dialogue entière : on affiche les 3 lignes à leur juste place pour que tout le dialogue rentre dans l’écran.