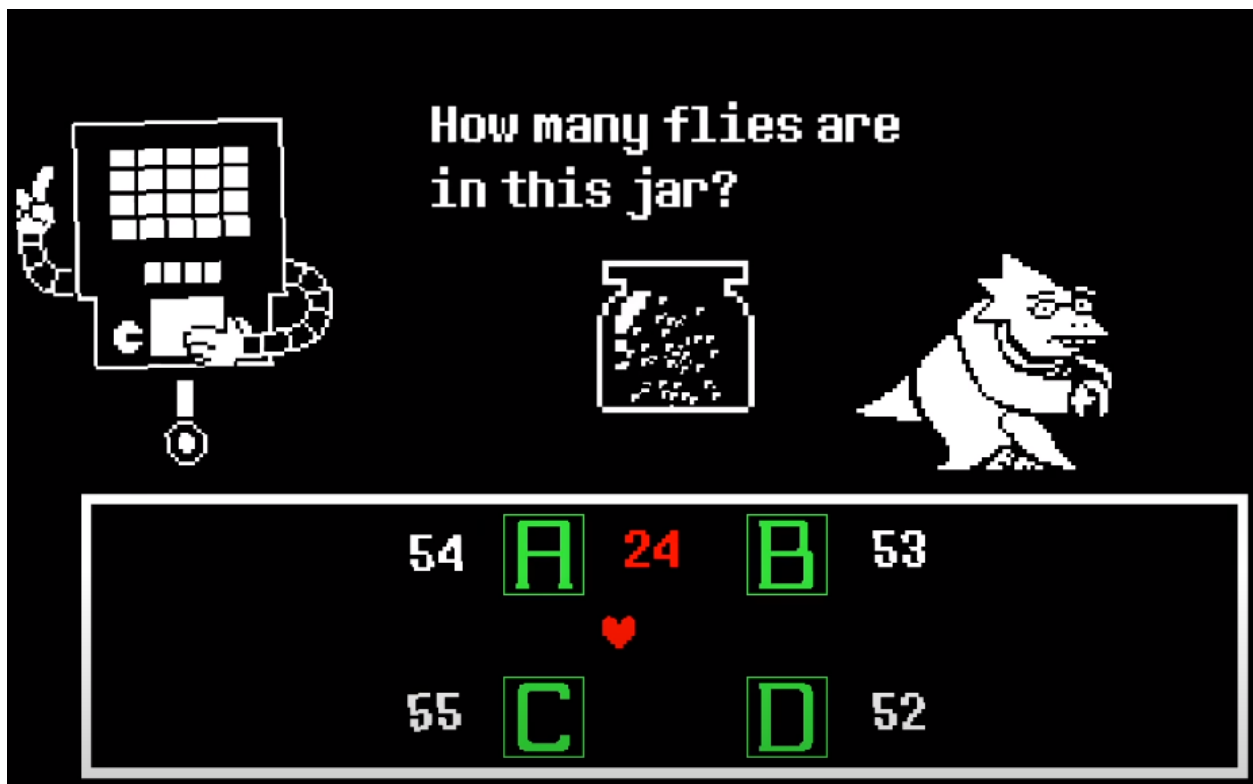


Pour adresser votre inquiétude de plagiat (et pour mieux documenter mes idées), j'ai décidé de mieux documenter mon processus pour ce projet.

Dans le projet de cinéma, je me suis familiarisé avec les modules Pillow et Pygame, ainsi que les objets de canvas dans Tkinter. Il y avait quelques points où que j'avais de la difficulté, mais la plupart du temps il y avait une réponse dans la documentation du module en question.

Dès le début, quelque chose qui m'a inspiré en programmation était le jeu vidéo Undertale, par Toby Fox et Temmie Chang. Ils permettent les personnes à utiliser leurs assets pour leurs propres projets, à moins qu'ils ne récréent pas le jeu entier pour gratuit.

Alors, lorsque j'ai vu le projet de jeu questionnaire, je savais que je voulais le baser sur une scène du jeu où que le personnage principal est forcé à participer à un [game show](#). Je trouvais le fait que la présentation du projet parlait d'une option de "hint" drôle puisque dans la scène originale, après quelques secondes Alphys montre la correcte réponse. Cependant, j'ai décidé de ne pas utiliser Alphys, et au lieu, un bouton qui donne l'utilisateur un choix de l'utiliser pour une pénalité d'HP.



Classes et modules

J'ai commencé ce nouveau projet en vouloir explorer les classes. Une ressource que j'ai utilisée pour apprendre le format général des classes en Python est [w3schools](#). Puisque j'ai un peu d'expérience en C#, après avoir consulté ce site, c'était pas mal facile à travailler de là. Je voulais aussi diviser le projet en plusieurs fichiers, puisque le seul script du dernier était un peu encombré, et je savais que celui-ci serait encore plus complexe. La combinaison de ces deux avancements a fait le développement de ce projet (et les diagrammes OPE) plus intéressant et mieux organisé.

Sprite/Player/Collider

Mon premier sprite était le player. J'avais appris comment utiliser `fenetre.bind()` lors du dernier projet, alors établir les inputs était facile. Mais, lorsque j'essayais de bouger le player, il y avait un bizarre glitch visuel que je n'étais pas capable à recréer, alors j'ai ajouté `spr.adjust()`, qui utilise `Image.new()` et `Image.paste()` de Pillow pour créer un "buffer" entre les frontières du sprite. J'ai trouvé un exemple sur [StackOverflow](#) et consulté la documentation de Pillow pour le compléter.

Pour l'indicateur d'HP du player j'ai consulté [ce site](#) pour le format des nombres sous 10.

Pour `spr.color()`, j'ai trouvé `Image.getdata()` et `Image.putdata()` sur [GitHub](#). Encore, j'ai consulté la documentation de Pillow, j'ai expérimenté, et j'ai complété cette fonction.

Lors de la conception du class Question, j'ai créé le class Collider en utilisant rien de nouveau.

Gif

C'est un peu d'un désastre, mais ça fonctionne. J'avais appris de `image.seek()` d'un thread de [StackOverflow](#), de qui j'ai utilisé une version modifiée de leur méthode de try, except pour extraire les frames. Je voulais utiliser `image.putalpha()` pour modifier l'opacité des derniers frames, mais je n'avais pas considéré que cette fonction modifie aussi des pixels qui sont déjà transparentes. Alors, j'ai utilisé une méthode similaire à `sprite.color()` pour remplacer ces nouveaux pixels. C'est peut-être inefficace, mais je ne pouvais pas trouver une autre solution.

En rétrospective, c'était beaucoup de travail pour quelque chose entièrement cosmétique.

Oh well.

MTT

Plutôt facile à animer grâce à mon expérience avec `Image.rotate()` et `fenetre.after()` dans le formatif, j'avais seulement à bien aligner les images des bras (que j'ai été capable à trouver sur [SpritersResource](#)) avec mon ordi à la maison, puisque je ne suis pas habitué à modifier des images avec un trackpad comme celui sur mon portable d'école.

Dialogue

La première chose à faire était de figurer comment utiliser des fonts custom. J'ai découvert [Pyglet](#) dans les réponses de [ce thread de StackOverflow](#). De là, j'ai ajouté Pyglet à la série d'installations dans `modules/setup.py`. Prochain, pour ajouter des pauses appropriées à l'effet de "typewriter" du texte, j'ai utilisé [any\(\)](#) pour déterminer si le caractère courant est de ponctuation. Pour les sons de voix, j'ai utilisé le mixer de Pygame encore, c'est à ce point que le warning de Pyglet commençait à apparaître dans la console. Je ne suis pas entièrement certain de comment prévenir ceci.

Question/Option/UI/etc.

Pas mal simple, essentiellement une mise en application des autres classes.