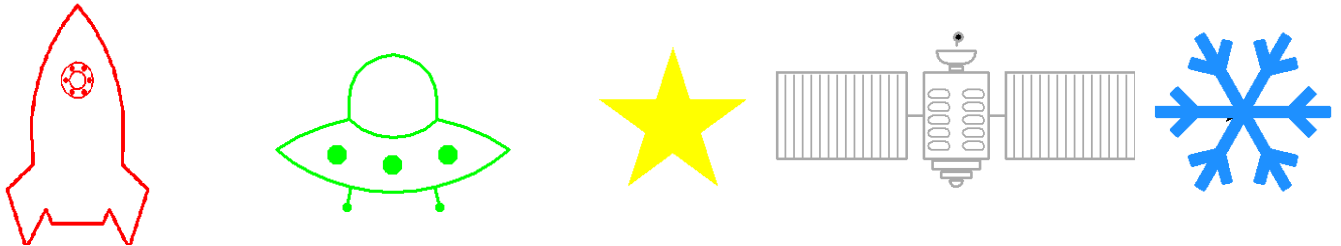


Pour ce projet nous avons décidé de prendre le thème de l'espace/ciel, les formes sont alors une fusée, un ovni, une étoile, un satellite et un flocon.



Nous avons également décidé de ne pas changer les règles mais de simplement changer le système de score qui devient alors un temps qui s'incrémente d'un certain nombre de secondes suivant la difficulté pour une bonne réponse et l'inverse pour une mauvaise réponse.

Pour la structure du code, nous avons choisi de créer deux modules. Un pour les dessins de bases (Fusée, étoile, etc...) et un autre module qui utilise ce premier module pour créer les graphismes qui vont être gérés dans le programme principal. Par exemple, pour l'affichage de la carte, on utilise les dessins du premier module puis on les affiche grâce au second module. On a ensuite décidé de créer plusieurs objets Turtle. Par exemple l'arrière-plan n'est pas dessiné avec le même objet que la carte ou le temps. Cela nous permet de gérer chaque objet séparément et de ne pas tout effacer et redessiner si nous passons d'un menu au jeu par exemple. Cela permet également d'avoir une meilleure optimisation. Cependant, le choix d'avoir plusieurs objets Turtle complexifie un peu le code avec beaucoup de variables globales pour chaque objet.

Pour la répartition des tâches, une personne fait tous les graphismes, et l'autre personne s'occupe de gérer ces graphismes pour les mêler avec la version textuelle du jeu. Ceci nous a permis d'être plus efficace puisque la première personne fait les graphismes sans penser au fonctionnement du jeu, et inversement pour la seconde personne, elle fait le jeu sans se préoccuper des graphismes pouvant être déplacé là où il veut. Au final, c'est Benjamin qui a fait les graphismes et Antoine l'implantation dans le jeu.



Dans ce projet, nous avons rencontré quelques difficultés. La toute première fut de prendre en main le fonctionnement d'une interface graphique qui est un peu contre intuitive d'un point de vue algorithmique. On a dû remplacer la boucle « while » dans notre version textuelle par la méthode « mainloop » de Turtle qui permet de faire cette boucle, et on a alors également dû placer la majeure partie du jeu, c'est-à-dire la création des Cartes et affichage des cartes, la récupération de la réponse et la vérification de celle-ci dans notre « clic ».

Une des autres difficultés que nous avons rencontré fut avec la création des boutons, que ce soit dans le menu principal comme dans le jeu, puisque pour envoyer la réponse dans notre version du jeu, il suffit de cliquer sur le bouton correspondant à la réponse voulue, cette notion a été un peu fastidieuse à prendre en main au départ mais avec un petit peu d'entraînement la suite n'en fut que plus rapide. Lorsque nous avions une version presque terminée basique, elle n'était vraiment pas accessible et ne pouvait être ouverte à des modifications que très difficilement. Nous avons alors décidé d'améliorer la partie code mais aussi nous en avons profité pour rajouter d'une part un timer au début de chaque manche (nous étions encore avec des scores) puis un menu et enfin des boutons accessibles tout le temps pour revenir au menu principal ou quitter le jeu.

La plus grande difficulté que nous avons eu était le timer que nous voulions absolument mettre en place (idée de Leyan Floriot G2). En effet, nous avons passé beaucoup de notre temps sur ce timer. Tous d'abord, nous avons utilisé la méthode 'sleep' du module 'time' mais ça ne marchait pas puisque le programme se mettait alors en pause toutes les secondes et il était impossible de cliquer sur les boutons. Nous avons ensuite essayé la même chose avec cette fois la méthode 'delay' de Turtle mais sans réussite non plus. Nous avons alors décidé au final d'utiliser la méthode 'ontimer' de Turtle. En effet cette méthode nous permet d'exécuter une fonction au bout d'un certain temps sans pour autant mettre en pause le programme. Ainsi, il nous a suffi de créer 2 fonctions, une qui le lance une fois avec

un nombre de fois au départ et une autre qui va s'appeler plusieurs fois. Mais la difficulté ne s'est pas arrêtée là puisque une fois fini, nous avons découvert un bug qui faisait que lorsqu'on lançait notre programme, on revenait au menu en pleine partie et lorsqu'on en relançait une, l'ancien timer était toujours en mémoire donc le temps se décrémenait 2 fois plus vite. Cependant, quelques tests avec des variables ont permis de remédier à ce problème.

Au niveau des modifications que nous pourrions apporter, qui peuvent certes être nombreuses et peuvent différer en fonction des points de vues et exigences de chacun, nous aurions pu par exemple opter pour un plus large choix au niveau des options, de ce fait le joueur aurait pu choisir le nombre exact de secondes qu'il possède au lancement du jeu, ou bien nous aurions pu afficher des résultats plus complets à la fin du jeu (nombre de bonnes réponses, temps moyen de réponse...) ou encore mieux optimiser l'affichage pour qu'il s'adapte a tous types d'écrans.

Enfin la dernière difficulté après toutes ces longues modifications fut la factorisation du code au maximum. Nous avons beaucoup de fonctions se ressemblant énormément comme par exemple une fonction qui permettait de revenir au menu après avoir gagné et une autre permettant la même chose mais depuis la partie en cours puisqu'on ne devait pas effacer les mêmes choses.



A gauche, on doit enlever moins de chose que celle d'en bas

