

# Projet de programmation fonctionnelle avancée

## Metroidvania

Adrien Chesneau  
Antonin Gallot

Notre jeu présente toutes les fonctionnalités décrites dans les paliers du sujet sauf la caméra. Nous avons également fait quelques bonus, le son et un power-up. On utilise les touches directionnelles pour diriger le personnage et les touches A et E pour tirer respectivement à gauche et à droite. La touche M permet de revenir instantanément au menu.

Nous avons commencé le développement en nous familiarisant avec Tsd, comprendre comment gérer les différents composants de la fenêtre, afficher des images, gérer la taille de ces images...

Nous nous sommes ensuite lancés dans la programmation du jeu lui-même. Nous avons commencé par afficher un personnage sans gravité avec des commandes pour le déplacer dans les quatre directions. Nous avons ensuite rajouté la gravité et les collisions avec des objets immobiles (mur ou plate-formes) afin d'avoir une première idée du ressenti lors du jeu. Nous avons ensuite rajouté des projectiles et un menu à partir de là, puis des ennemis. Nous avons alors essayé d'implémenter notre premier algorithme de collision gérant tous les objets, qui fonctionnait comme suit : lorsqu'on veut déplacer un objet, on compare sa position après déplacement avec la position après déplacement de tous les autres objets de la scène. Si il y a une collision, on replace les deux objets en fonction du sens de collision.

Notre chargé de TD nous a rapidement dit que cet algorithme n'était pas le plus efficace ni le plus simple, et nous avons donc changé pour l'algorithme suivant : on déplace tous les objets puis on gère toutes les collisions. Malgré que cet algorithme soit plus simple à implémenter que le précédent, cela nous a tout de même pris un certain temps car nous devions réécrire une partie de notre code. En effet jusque là nous différencions nos objets dans la scène en les plaçant dans différentes listes. Nous avons donc une liste d'ennemis, une liste de plate-formes et une liste de projectiles en attribut de la scène. Nous avons changé cela pour avoir une seule liste d'objets dans la scène et stocker dans le type de chaque objet dans un de ses attributs.

Une fois ce changement fait il a fallu réadapter nos fonctionnalités (tir, déplacement, saut...) à ce nouveau fonctionnement.

Une fois cela fait nous avons pu nous lancer dans les animations, ce qui a été relativement rapide. A partir de là nous avons juste corrigé quelques bugs mineurs et ajouté quelques fonctionnalités bonus.

Nous n'avons pas fait la caméra car nous nous étions lancés dans la réécriture du code pour le nouvel algorithme de collision et n'avons pas eu le temps.

La répartition du travail a été la suivante :

- Antonin a fait le chargement des fenêtres, textures etc., le déplacement, la gestion des différents types d'objets, les collisions, la fonction de tir adaptée au nouvel

algorithme de collision

- Adrien a fait le menu, les animations, le son, les power-ups, la première fonction de tir, la partie graphique avec découpage de tous les sprites, design du niveau