

Programmation créative - TD2 - Mini-jeu interactif

Nikita MELNIKOV et Florian VACHER

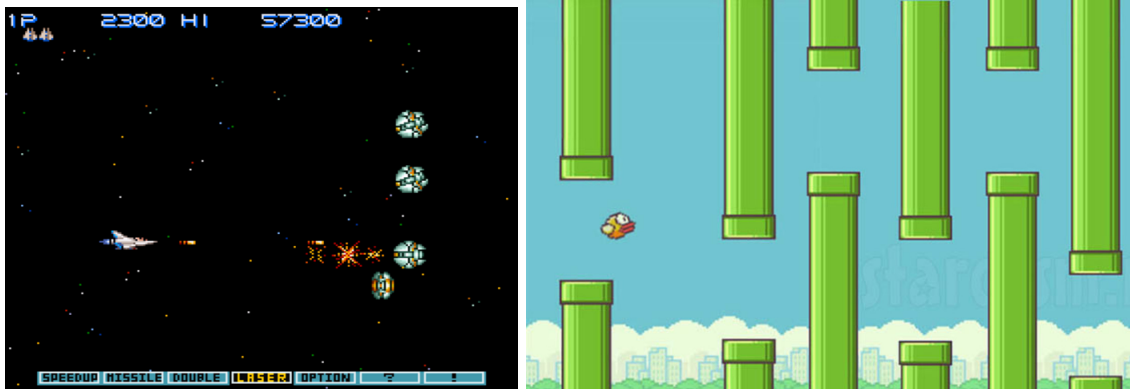
09/11/2022

1 Notes d'intention

L'objectif a été de produire un jeu d'obstacles très simple où le joueur doit faire éviter à son véhicule des éléments défilant vers lui de manière horizontale de la droite vers la gauche.

L'inspiration première se base sur le célèbre jeu mobile qu'a été Flappy Bird, un jeu vidéo d'obstacles développé au Viêt Nam par Nguyễn Hà Đông, développeur de jeu indépendant. Ce dernier est sorti en mai 2013 pour iPhone sur iOS 6. Le gameplay repose sur l'agilité du joueur qui doit faire avancer un oiseau dans un environnement à défilement horizontal en tapotant sur l'écran tactile tout en évitant des tuyaux présents en haut et en bas de l'écran.

A l'instar du jeu mobile, nous avons repris les mêmes règles où toucher un tuyau fera terminer la partie. Le joueur reçoit des points pour chaque tuyau traversé. Cependant, le véhicule contrôlé au sein de notre jeu s'apparente davantage à ceux que l'on retrouve dans le format de type "shoot them up" (un genre de jeu vidéo dérivé du jeu d'action dans lequel le joueur dirige un véhicule ou un personnage devant détruire un grand nombre d'ennemis à l'aide de projectiles) comme dans le célèbre Gradius (jeu vidéo de type shoot 'em up développé et édité par Konami en 1985 sur borne d'arcade).



2 Principales fonctionnalités

2.1 Ajout d'images

Afin de pouvoir apporter une thématique visuelle à notre jeu, nous avons fait usage des méthodes `preload()` et `loadImage()` dans le but de charger et d'insérer des éléments graphiques à l'écran. Ainsi, le véhicule, le fond et les obstacles sont habillés d'une image.

2.2 Génération des tuyaux

Qui dit jeu d'obstacles, dit génération d'obstacles. C'est pourquoi il a été nécessaire que d'intégrer au code une façon que de générer des objets se déplaçant dans l'axe X de droite vers la gauche que le joueur doit éviter. Les tuyaux sont générés toutes les X millisecondes avec une position aléatoire selon une plage de positions possibles. Une fois qu'un tuyau est sorti de l'écran, l'espace mémoire qu'il occupait est libérée dans un souci d'optimisation.

2.3 Déplacement de l'hélicoptère

Puisqu'il est question d'un jeu d'obstacles, le joueur se doit de les faire éviter au véhicule qu'il contrôle. De ce fait, il a été question que de gérer le déplacement de l'hélicoptère par l'intermédiaire des touches directionnelles du clavier (haut, bas, gauche et droite). De plus, une interface a été ajoutée en complément sur l'écran reprenant l'allure des touches directionnelles du clavier afin de permettre au joueur de faire usage de ces fonctionnalités dans le cas où il ferait usage d'un support tactile (smartphone, tablette) et non d'un ordinateur.

2.4 Détection des collisions

Une fois les deux précédents points gérés, il fallait les mettre en lien. Nous avons donc dû gérer la collision entre l'hélicoptère et les tuyaux en mouvements. Pour cela, les tuyaux à l'écran sont stockés dans un tableau. A chaque affichage, un test est fait pour voir si les positions de l'hélicoptère ne rentrent pas en conflit avec une des positions des tuyaux. Si c'est le cas, le jeu s'arrête et le joueur est invité à recommencer.

2.5 Ajout des sons

Dans la plupart des jeux, nous pouvons retrouver un son d'ambiance qui permet à l'utilisateur d'être en immersion dans le jeu. C'est pour cela, que nous avons décidé de rajouter différents sons modélisant différentes actions ;

- Etre sur le menu principal (son d'ambiance)
- Etre en pleine partie (son d'ambiance)
- Le fait que l'hélicoptère se crache contre un tuyau
- Le fait que l'hélicoptère passe correctement entre deux tuyaux
- Le fait d'avoir gagné

2.6 Ajout de pop-up

Afin de différencier le fait que le joueur ai perdu ou gagner, nous avons décidé d'implémenter des pop-up s'affichant en cas de victoire ou de défaite. En plus de ça, le score est affiché et deux possibilités s'offrent à l'utilisateur : soit de retourner à la page d'accueil, soit de recommencer le niveau.

2.7 Animation de fin

Afin de rester dans le thème du vol et du fait que l'hélicoptère (dans la vie réelle) est très manipulable, nous avons décidé que notre hélicoptère fasse un looping suite à une victoire. Nous avons fixé la position de l'animation au 3/4 de l'écran afin qu'elle s'effectue juste avant qu'il ne quitte l'écran. Le "looping" s'effectue grâce à une rotation de l'image autour d'un point situé au-dessus de la trajectoire de l'hélicoptère (afin d'éviter d'avoir un effet saccadé entre le déplacement et le "looping"). Une translation doit être bien sûr effectuée afin d'éviter que l'hélicoptère ne tourne autour du point (0,0) et donc de permettre qu'il tourne bien autour du point défini juste avant.