

Dustopia

Anouk OMS, Lysandre TERRISSE, Elora ROGET, Thomas DAILLY

Février 2023

Scénario

Nous contrôlons Kabu, une susuwatari. Cette petite boule de suie vient d'une autre planète pour libérer des étoiles que les humains exploitent pour faire fonctionner leurs usines. En explorant la Terre, notre susuwatari découvre que d'autres de son espèce sont enfermées dans le but de faire des recherches scientifique. Les humains ont créé des gardiens balais qui ont comme seul but d'empêcher les susuwataris de s'enfuir. Le but du héros est donc d'aller libérer ses congénères tout en récupérant un maximum d'étoiles sur le chemin, puis de rentrer sur leur planète d'origine.

Table des matières

1	Introduction	3
2	Cahier des charges et planification	4
2.1	Cahier des charges	4
2.2	Planification	5
3	Conclusion	8
4	Bibliographie et Sitographie	9
4.1	Bibliographie	9
4.2	Sitographie	9

1 Introduction

Dans le cadre du cours de projet CMI, nous avons dû réfléchir à un programme à développer (application, jeu...). Le choix de notre groupe s'est porté sur un jeu de plateforme type Mario en deux dimensions.

Nous nous inspirons pour l'univers et les personnages du studio Ghibli, reprenant comme personnage principal une susuwatari.



Les ennemis et obstacles seront en rapport avec notre personnage. Son but sera d'avancer jusqu'à la fin du niveau en évitant les ennemis et en se déplaçant de plateforme en plateforme.

Nous avons décidé de coder en C car d'après nos recherches il s'agit d'un bon langage permettant de coder des jeux et nous enlevant certaines contraintes quant à l'optimisation. On travaillera également avec CSFML, une bibliothèque initialement créée pour C++, et qui nous permettra de faire l'aspect graphique de notre jeu.

2 Cahier des charges et planification

2.1 Cahier des charges

Le jeu sera conçu pour un seul joueur. Lors du lancement, il devra avoir accès à un menu lui permettant de lancer un niveau. Une fois le niveau lancé il se retrouvera directement aux commandes du héros. Le joueur verra alors son interface graphique avec les vies de son personnage, son personnage au sein du niveau, et l'accès à un deuxième menu permettant de modifier certaines options telles que les contrôles, le volume, et l'activation des musiques.

Le joueur devra se déplacer par défaut grâce aux touches directionnelles du clavier. Il pourra se déplacer vers la gauche ou la droite, et pourra sauter. Lorsqu'il sautera, sa trajectoire sera définie par la vitesse à laquelle il sera. En l'air, il sera évidemment soumis à la gravité et retombera selon un arc calculé au cours de sa trajectoire. Lors du heurt avec un obstacle, il perdra toute sa vitesse sans rebondir. En heurtant un mur dans les airs il glissera le long, perdant toute sa vitesse horizontale et gagnant progressivement de la vitesse verticale vers le sol. Les ennemis blesseront le héros sans lui faire perdre sa vitesse.

Le personnage sera positionné au premier tiers de l'écran afin que le joueur ait une bonne vision du terrain devant lui. Le décor sera composé de blocs chargés au fur et à mesure que le joueur avancera ou reculera. Nous devons calculer les zones de heurt des obstacles avec le joueur afin qu'il ne traverse pas les blocs. Ce calcul devra cependant prendre garde à bien repositionner le joueur pour éviter tout bug de collision, bugs qui sont d'ailleurs extrêmement fréquents dans les jeux de plateformes.

Des ennemis seront générés par des éléments du décors. Ces ennemis se déplaceront vers le joueur avec une vitesse prédéfinie et ne pourront pas sauter. Ils seront invincibles et ne dévieront pas de leur trajectoire, forçant le joueur à les esquiver. Lorsque les ennemis toucheront le joueur, celui-ci perdra de la vie et eux continueront leur chemin.

2.2 Planification

Nous avons décidé de nous répartir le travail (ce qui fut un peu compliqué étant donné les absences de Thomas). Ayant rejoint le projet en cours, nous avons dû l'intégrer à nos tâches et avons donc décidé de nous répartir le travail comme suit :

- **Anouk : codage des monstres**
 - Déplacements : lorsqu'un monstre apparaît, il choisit une direction unique en fonction de la position du joueur : si celui-ci se tient à sa gauche alors il ira vers lui mais ne changera jamais de direction.
 - Attaques : lorsqu'un monstre touche le joueur, il lui fait perdre une vie.
 - Apparition des monstres : les monstres proviennent soit des poubelles implémentées partout sur la carte, soit ils sont déjà présents sur la carte (et sont figés tant qu'ils ne sont pas visibles sur l'écran). Lorsqu'une poubelle apparaît sur l'écran, celle-ci génère un ennemi de manière régulière (un tout les dix secondes) et arrête lorsqu'elle a fait apparaître dix ennemis afin de ne pas trop charger la carte. Une poubelle arrête de générer des monstres lorsque le joueur s'éloigne d'elle (définir la distance nécessaire entre une poubelle et le joueur, hors-champ)
 - Disparition lorsque le joueur est trop loin d'eux (définir la distance nécessaire entre un monstre et le joueur, hors-champ)
 - Aider au codage du menu
- **Lysandre : interface graphique, sons et système de collision**
 - Création de deux formats PixelArt et ObjectArt. PixelArt serait un format avec des valeurs rgb séparées par des points virgules, tandis que ObjectArt serait un format où l'on spécifie une forme que l'on veut avec certains paramètres.

Création des fonctions `calculateImage` et `useImage`, qui permettent de calculer la texture d'une image, et de la calquer à un endroit précis et avec une échelle précise sur le buffer d'écran.

- Gestion du son et création des musiques.
- Implémentation du système de collisions. Ce système de collision sera fait en calculant le point d'intersection entre la droite du vecteur de vitesse du joueur et les droites correspondant aux côtés des blocs.

- **Elora : graphisme et codage en soutien**

- Création des images : design du personnage, des monstres dont des balais et plumeaux, création du background et des textures, design de la barre de vie.
- Les fonctions touchant à la vie du personnage (initialisation de la barre de vie, décrémentation d'une vie lors d'une attaque de monstre, mort du joueur si la barre atteint 0).
- Aide sur les fonctions qui demandent beaucoup de travail et/ou de réflexion comme la fonction à propos des collisions.
- Création du menu

- **Thomas : soutien au codage**

- Aide sur la fonction de collision.
- Aide à l'interface graphique.
- Aide à l'écriture des niveaux.

Pour ce projet, nous avons créé un Github qui puisse accueillir nos codes:

<https://github.com/Delreya-o/CMI-Project.git>

Nous nous sommes répartis les tâches à l'aide d'un diagramme de Gantt.

Fonctions & sous-fonctions	06/02 – 12/02	13/02 – 19/02	20/02 – 26/02	27/02 – 05/03	06/03 – 12/03	13/03 – 19/03	20/03 – 26/03	27/03 – 02/04	03/04 – 09/04	10/04 – 16/04	17/04 – 18/04
		Programmations des fonctions de bases					Améliorations & niveaux			Rendu du projet et préparation de l'Oral	
Organisation											
Attaques											
Apparition :											
Création											
Intervalle											
Disparition											
Déplacement											
Création niveaux											
Création menu											
Préparer l'oral											
Organisation											
Background											
Design											
Personnage											
Ennemis											
Obstacles											
Barre vie											
initialiser											
décrémenter											
mourir											
Création menu											
Préparer l'oral											
Organisation											
Format PixelArt											
calculerImage & drawImage											
Collisions											
ObjectArt											
Implémenter son											
Création son											
Musiques											
Création niveau											
Préparer l'oral											
Aide au code											
Préparer l'oral											

3 Conclusion

Nous avons de nombreuses fonctionnalités que nous aimerions améliorer lorsque le projet sera suffisamment fonctionnel pour être testé. Nous cherchons pour l'instant à obtenir une version permettant une expérience agréable sur quelques niveaux, et avec un nombre minimal d'éléments. Une fois que nous aurons une preuve que notre démo fonctionne, nous améliorerons certains éléments et rajouterons plus de fonctionnalités, comme des niveaux de plusieurs thèmes, des boss, de nouveaux ennemis et des objets à ramasser, comme des pièces.

4 Bibliographie et Sitographie

4.1 Bibliographie

Bibliographie de SFML (celle de CSFML est très incomplète)

<https://www.sfml-dev.org/documentation/2.5.1/annotated.php>

4.2 Sitographie

Quelques vidéos de Handmade Hero pour apprendre C

<https://guide.handmadehero.org/>

Quelques vidéos de Formation Vidéo pour apprendre C

<https://www.youtube.com/watch?v=90hGCMC3Chc&list=PLrSOXFDHBtfEh6PCE39HERGgbbaIHhy4j>

W3school pour apprendre C

<https://www.w3schools.com/c/index.php>

Image du susuwatari

<https://sketchfab.com/3d-models/susuwatari-3c4ec8341f9847f28ae776ebd1624dde>

Le lien du projet sur GitHub

<https://github.com/Delreya-o/CMI-Project.git>