

Projet Time To Play

Cahier des Charges et de Création

Membres de l'équipe : Gaby FULCHIC, Rémi Mafat, Benoit Jallet, Damien Pommier

Les réalisations de chacun

Nous avons travaillé régulièrement en équipe, il est donc assez difficile de déterminer qui a travaillé sur quoi. Ci-dessous nous avons essayé de lister les éléments sur lesquels un membre a travaillé en particulier tout en sachant que les tâches sont restées très collaboratives.

Damien nous a rejoint lors de l'avant-dernière séance, il a donc participé selon le temps qui lui était imparti.

Gaby :

- Animation et gestion du déplacement des ennemis
- Gestion du score
- Gestion de l'invincibilité
- Intégration de la musique
- Merge du code

Rémi :

- Animation du joueur (poulet) et gestion du déplacement
- Gestion du lancer d'œuf pour éliminer un ennemi
- Ajout d'ennemis multiples
- Gestion de la mort du joueur et game restart
- Création de la musique et des sons en Home Studio
- Intégration du menu
- Debug
- Ecran de victoire
- Positionnement des ennemis du niveau 2 + flèches et tornade
- Ajout des sons

Benoit :

- Création de la map en fichier JSON avec Tiled
- Intégration des tileset et level design
- Gestion du défilement
- Création du menu
- Debug

Damien :

- Défilement du scoring
- Réglages sur l'ATH

Les choix réalisés concernant le framework

Nous avons choisi de travailler avec le framework Phaser car il nous semble assez complet, beaucoup de tutos sont disponibles et la communauté assez large. De plus, c'est un framework particulièrement adapté pour le développement d'un jeu.

Nous avons utilisé Yeoman pour structurer notre projet et gérer plus facilement les states (notamment le menu), ainsi que pour tester notre jeu sur un serveur local.

Les choix du gameplay

Notre objectif est de créer un jeu de plate-formes de type Super Mario.

Le joueur est un poulet qui part à la recherche de ses poussins, perdus lors d'un voyage en Egypte (ne nous demandez pas pourquoi svp). Il devra se frotter à nombre d'obstacles et d'ennemis pour arriver à son but.

Le jeu comporte plusieurs niveaux (tableaux) accessibles depuis le menu principal.

Les difficultés rencontrées

Par où commencer ?

Dès le démarrage du projet, nous nous sommes confrontés à la difficulté de créer un jeu avec un framework que nous ne connaissions pas.

Pour remédier à cela, nous avons suivi chacun le tutoriel Phaser pour apprendre à créer un jeu de base.

Faire des animations

Seconde étape, apprendre à gérer l'animation des sprites (gauche, droite, saut, etc.)

Créer une map défilante

Première grosse difficulté rencontrée dans ce projet, nous voulons créer un jeu dans lequel le joueur se déplace sur un tableau relativement long verticalement, et donc créer un système où la caméra suit le joueur lorsqu'il se déplace.

Nous avons trouvé de l'aide en ligne, créé une map en JSON avec un logiciel (Tiled) et utilisé des fonctions telles que `game.camera.follow()`.

Fixer l'ATH (score, vie, etc.)

Le jeu défile horizontalement, il a fallu donc trouver un moyen de fixer les informations. Nous avons fini par trouver sur les exemples de Phaser, la fonction `fixCamera()`.

Tuer un ennemi

Afin de tuer un ennemi, et pour un peu d'originalité, nous voulions pondre un œuf sur leur tête (si le joueur touche un ennemi, il perd). Nous avons eu du mal à réussir à gérer la ponte correcte d'un œuf (un œuf à la fois, impossibilité de pondre des œufs en mode « mitraillette », etc.) et avons mis en place pour cela un booléen pour savoir si un œuf est déjà présent sur la map et une fonction qui le delete au bout d'une seconde avec `setTimeout()`

Gérer l'invincibilité

Lorsque le joueur attrape un œuf de dragon, il devient invincible pendant 10 secondes. Afin de gérer cela, nous avons dû gérer un nouveau sprite pour le joueur (il passe au rouge) et avec un booléen, modifie le comportement des fonctions de collide avec les ennemis (si elle est à true, l'ennemi est tué au lieu du joueur).

Faire le menu

Point sur lequel nous avons beaucoup perdu de temps. En effet, nous étions partis pour créer un menu nous-même (avec la gestion des states), ce qui était de base plutôt complexe malgré les informations récupérées sur le web.

Puis nous avons migré sur Yeoman, dont la structure propose déjà des states pré-remplies. Il ne restait plus qu'à.

Sauf que l'utilisation de Yeoman nous a causé pas mal de tracas, vu qu'il a fallu intégrer notre code et notre projet dans la structure nouvellement créée (pas facile). Nous avons ainsi passé toute une matinée à intégrer notre code et à le modifier pour faire fonctionner correctement le jeu dans Yeoman.