Space shooter

Space shooter est un jeu vidéo sur navigateur web. Le joueur contrôle un vaisseau spatial qui peut se mouvoir à l'aide des flèches directionnelles. Le but est de faire le plus de points possible en récupérant les gemmes vertes réparties aléatoirement sur la carte. Pour cela, il va falloir éviter les météores ou les détruire à l'aide de vos missiles (touche espace). 3 vies sont à disposition et le nombre de niveau est infini. Chaque gemme attrapée remporte 50 points et la collision avec un météore fait perdre une vie.

Technologies

Le jeu est entièrement fait en javascript avec l'aide du framework Phaser, puis intégré dans une simple page en HTML5 / CSS3.

Installation

1. Récupérez l'archive de la dernière version du jeu disponible ici puis, décompressez-la dans le répertoire www (ou équivalent) de votre serveur web.
2. Ouvrer le fichier index.html (en passant par votre serveur web) avec firefox ou google chrome (version récente) *

** La compatibilité avec Internet Explorer ou des vieilles versions de Chrome et Firefox n'est pas assurée dû à l'utilisation de javascript ES-6 et il n'est pour le moment pas retranscrit en ES-5.*

Organisation des fichiers

- JS
  - states
    - MenuState.js => L'état menu du jeu
    - GameState.js => L'état jeu du jeu
    - GameOverState.js => l'état game over du jeu
  - vendor
    - phaser.min.js => La librairie phaser
  - main.js => Définition des variables globales, des états, lancement du menu
  - Game.js => Classe principale du jeu, qui contient et gère les états
  - Meteor.js => Classe qui définit et gère les météores
  - Gem.js => Classe qui définit et gère les gemmes

Inspiration

Deadzone

Qu'est-ce que c'est ?

La deadzone est un rectangle au centre de l'écran de jeu où le joueur peut se déplacer librement sans suivi de caméra. Une fois arrivé contre un bord de cette zone, si le joueur continu de se déplacer contre celui-ci, la caméra va le suivre et le personnage se déplace dans le monde et non plus dans la deadzone.

Qu'est-ce que j'ai repris ?

J'ai entièrement repris le code permettant la gestion de la caméra et de la deadzone. C'est quelque chose de fourni par la framework Phaser, donc il est plus judicieux de le reprendre que de refaire soit même un système de caméra.

Example + code source disponible ici.

Déplacement joueur + système de tir

Qu'est-ce que j'ai repris ?

J'ai repris un des systèmes de déplacement très original proposé par le framework phaser. C'est un système qui permet seulement l'accélération du player. Pour aller sur la gauche ou la droite nous allons appliquer une vitesse angulaire sur le personnage, ce qui va lui faire faire une rotation. Une fois la rotation effectuée, il suffit de remettre de l'accélération pour avancer. Ce système reproduit un mouvement plutôt réaliste d'un vaisseau spatial qui n'a aucun frottement ni de surface d'attraction.

De plus, le système de tir a aussi été repris. C'est un système basique, lui aussi proposé par le framework, qui permet l'utilisation d'une arme qui tire des projectiles en fonction de l'orientation du joueur. Il est facilement personnalisable (nombre de projectile, fréquence de tir, etc.)

Exemple + code source disponible ici.

Création custom

Voici une liste de toutes les fonctionnalités que j'ai entièrement créées, sans m'inspirer de choses déjà faites.

- La logique / le style de jeu
- L'animation du joueur quand il accélère
- La création de chaque météore (3 de base et 1 de plus chaque 3 seconde)
- Le style et la logique des météores.
    - Chaque météore a un style graphique choisi aléatoirement parmi 4
    - Chaque météore à une rotation et une trajectoire aléatoire
    - Les météores peuvent traverser la zone de jeu et réapparaître à l'opposé
- La génération des gemmes
    - 5 gemmes sont placées aléatoirement dans la zone de jeu
    - Les gemmes ont une animation de rotation
- Les collisions
    - Entre les tirs du joueur et les météores
        - L'animation d'explosion
    - Entre les météores et le joueur
        - L'animation d'explosion
        - La perte d'une vie
    - Entre le joueur et les gemmes
        - Le score qui augmente
- Les passages d'un état de jeu à un autre (ex : du jeu au game-over)
- La gestion de tous les audios (musiques, effets spéciaux)

Chaque fonctionnalité est décrite et commentée dans le code source du jeu disponible ici.

Captures d'écran