**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 variable globales, des états, lancement du menu
  + Game.js => Classe principale du jeu, qui contient et gère les états
  + Meteor.js => Classe qui défini et gère les météores
  + Gem.js => Classe qui défini et gère les gemmes

**Création de moi-même**

* La logique / le style de jeu
* L’animation du joueur
* 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 réapparaitre 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)

**De quoi parler pour la présentation**

* Les différents états -> fonctions preload(), create(), update()
* Weapon + tir (géré par la lib)
* Les collisions -> prendre exemple le catchGem()

