GRAVITATION ORAL PRESENTATION  
1 English Abstract

WEB MULTIPLAYER VIDEO GAME  
 ALL DEVICES  
OLD SCHOOL 2D GAMEPLAY  
SOLO  
MULTIPLAYER  
ACCOUNT  
RELATIONSHIP  
CHAT  
LOBBY  
RANKING

2 Définition de l’expression de besoin

3 FIRST SPRINT : PHASER  
🡺 apprendre Phaser.io  
Prototype 🡺 jeu Solo  
Comportement du vaisseau  
du tir, de la caméra  
🡺 VIDEO Grav2.avi

MULTIJOUEUR ?  
🡺 WEBSOCKETS  
Communication bidirectionnelle  
Mode push 🡺 Envoi des données au client a l’initiative du serveur

4 SECOND SPRINT : SOCKET.IO

🡺 Apprendre socket.io  
🡺Node.js / Express / Socket.io  
Assez long à mettre en place  
Première implémentation foireuse  
Résultat non concluant

Slack simplon 🡺 Meteor

5 THIRD SPRINT: METEOR  
Fait un POC sur meteor 🡺 Le problème venait de la manière dont je gérait les données communes et que ce n’était pas un probleme technique.

Meteor présente beaucoup d’avantage pour le reste du developpement de l’application.  
Gros framework 🡺 implémente les websockets nativement <3 ( POC 2x + Rapide )  
<3 Client/Serveur 🡺 Isomorphique  
<3 Monolanguage ( JS <3 ).  
<3 React as View 🡺   
<3 ES6 <3

6 FOURTH SPRINT : The Application !

🡺 Meteor  
🡺 React  
🡺 User Account  
🡺 implémentation Phaser Modulaire ( class ES6, import export ).

Structure de fichier de l’application

REMINDER

Application Front 🡺 Mobile

Server 🡺 API ( Application Programming Interface ).  
Controller CRUD Create, Read, Update, Delete

BDD 🡺 MongoDB  
{ title : ‘toto’,  
content : ‘tatatitiatia’}