

Projet Javascript

The Little Miner



DORET Alexandre – GARDIN Kélian – HANOTEL Guillaume

Table des matières

1) Les Réalisations :	3
2) Le choix du Framework	3
3) Les choix de gameplay.....	3
4) Les difficultés rencontrées	4

1) Les Réalisations :

Pour ce projet, nous avons essayé de nous répartir les tâches pour faire en sorte que chacun progresse, l'un qui avait plus de facilité a pu aider les 2 autres à réaliser leurs tâches.

Guillaume HANOTEL : Modèle du jeu, animation de destruction, structure du jeu, fonction de la classe Block

Kélian GARDIN : Création des différents blocks, animation TNT, Menu Title, fonction de la classe GameModel

Alexandre DORET : Intégration Son/Musique, Animation Dynamite, Menu GameOver, gestion du score

2) Le choix du Framework

Pour réaliser ce projet, nous n'avions pas de contraintes spécifiques nécessitant l'utilisation d'un Framework particulier. C'est pourquoi nous nous sommes tournés vers un outil classique déjà connu pour réaliser des jeux vidéo. Phaser s'est imposé comme le choix pertinent, de par sa documentation fournie, mais aussi par son ancienneté qui a permis la naissance de nombreux projets, dont de nombreux sujets de forum ont découlé, qui nous ont permis de piocher parmi toutes ces ressources pour forger notre jeu petit à petit.

3) Les choix de gameplay

Le gameplay de notre jeu s'inspire d'un jeu déjà existant sur smartphone : Pocket Mine (et Pocket Mine 2). Penser le gameplay d'un jeu vidéo n'est pas une tâche facile, pour éviter de nous confronter à des problématiques liées à toutes les mécaniques qui fait qu'un jeu est équilibré/prenant/suscite de l'intérêt, etc., nous avons choisi de reprendre le concept du jeu, ce qui nous évite de le penser, pour se concentrer sur la phase de création en elle-même.

De plus, le thème du jeu nous a vraiment aidé car proche de l'univers de Minecraft, ce qui a permis de trouver des sprites et images très facilement.

Le jeu a un fonctionnement simple ; notre souris fait office d'une pioche qui possède un nombre limité de coup pour miner les blocs qui se trouvent en profondeur. Au début les seuls blocs visibles sont ceux de la surface, une fois creusé, la caméra défile vers le bas et donne accès aux blocs adjacents qui peuvent, à leur tour être minés. Le but est donc de creuser en profondeur pour aller le plus loin possible, sachant qu'il existe différents types de block pouvant ralentir ou accélérer la progression du joueur. Une fois la pioche épuisée, c'est Game Over et un score vous est donné en fonction de la profondeur jusqu'où vous êtes allé et si vous avez récupéré des minerais en route.

La dimension aléatoire du terrain fait que le jeu n'est jamais le même et que l'on peut toujours essayer d'améliorer son score.

4) Les difficultés rencontrées

Quelques difficultés se sont posées lors du processus de création :

La première est bien évidemment la prise en main du Framework Phaser, il a fallu reprendre les tutoriels et tester les différentes fonctions et paramètres pour comprendre comment se fabriquait un jeu, de son background, à la physique du jeu. L'un des enjeux importants a aussi été de correctement organiser le code selon le modèle MVC sur lequel nous avons un peu de mal au départ, mais qui a pris forme au fur et à mesure de nos expérimentations.

Au départ, même si nous séparions bien le code avec le modèle MVC, tout était dans 1 fichier, nous avons vite corrigé ça en regardant comment séparé le jeu Phaser en « état » : menu/jeu/gameover, puis le modèle du jeu en autant de fichier qu'il y a de classe.

Enfin, vers la fin du développement, ne restant que le personnage initialement prévu à faire, au vu des délais restants, nous avons fait le choix de changer de perspective et d'opter pour un pointeur pioche, pour nous épargner de longues heures de développement pour un résultat pas forcément indispensable. Ce qui nous a permis de finaliser d'autres points et autres bugs.