



Doodle Jump Project

Doodle Jump, le jeu

Doodle Jump ([cliquez ici](#) !) est un jeu de plate-forme qui a connu son heure de gloire il y a 10 ans, ainsi qu'une résurrection remarquée en 2018. Si vous ne savez pas en quoi consiste ce classique, vous pouvez jeter un oeil sur quelques vidéos youtube (par exemple [là](#)) ou bien tout simplement télécharger le jeu sur votre mobile. Le but de ce projet est de coder en Javascript (Vanilla !) votre propre Doodle Jump.

Pour être recevable, votre projet doit implémenter a minima les fonctionnalités listées ci-dessous. De plus, au niveau de votre implémentation, vous choisirez et respecterez un pattern objet pour la gestion de tous vos éléments (pseudo-classiques, ES6, ou bien pattern OLOO). Concernant l'affichage, vous êtes libres de choisir un rendu DOM classique, ou bien un rendu "canvas".

Caractéristiques demandées

- **Progression**

Vous aurez compris que la devise du jeu est "toujours plus haut". Ce type de jeu n'a jamais vraiment de fin, mais plus l'on progresse, plus la difficulté augmente. Vous devez respecter ce principe. Une version "sans fin" suppose que les plate-formes et autres éléments soient générées dynamiquement (en faisant bon usage de nombres aléatoires) au fur et à

mesure que le joueur s'élève, sachant qu'il faut toujours s'assurer que depuis une plate-forme, il soit toujours possible d'atteindre au moins une autre plate-forme plus haute. On pourra accepter que vous utilisiez un format de données (au hasard JSON) pour pré-définir des niveaux, mais ce n'est probablement pas l'option qui vous donnera le moins de fil à retordre. Sans compter que la définition des niveaux risque d'être très chronophage sans que ça n'apporte de plus-value à votre travail. Donc réfléchissez bien.

- **Le Gribouilleur / Doodler**

Le Gribouilleur est le personnage du jeu. Son contrôle se fera avec les flèches. La possibilité de tirer avec la barre d'espace n'est pas imposée mais sera appréciée à sa juste valeur. Au cours de sa progression, le Gribouilleur pourra collecter les bonus suivants :

- une casquette à hélice qui le fait voler pendant 2 secondes
- un jetpack qui le fait voler (plus vite que la casquette) pendant 4 secondes
- une fusée que le fait voler (plus vite que le jetpack) pendant 6 secondes
- une potion d'invincibilité (contre les monstres) valable pendant 15 secondes

- **Les plates-formes**

Il en existe différentes sortes :

- des plates-formes "normales" fixes et sans mauvaise surprise
- des plates-formes

pourries qui cèdent
si le Gribouilleur
saute dessus
(entraînant sa chute)

- des plates-formes
mobiles qui
oscillent
horizontalement
- des plates-formes
mobiles qui
oscillent
verticalement
- des plates-formes
avec ressort qui
permettent au
Gribouilleur de
sauter 2 fois plus
haut
- des plates-formes
avec trampoline qui
permettent au
Gribouilleur de
sauter 4 fois plus
haut

- **Les monstres**

Les monstres sont létaux pour le Gribouilleur.
On partira du principe qu'on ne gère pas de
collision entre les monstres et les plates-
formes. Seulement avec le Gribouilleur. 3
types de monstres peuvent être rencontrés :

- des monstres
immobiles
- des monstres
mobiles
horizontalement
- des monstres
mobiles
verticalement
- et ce n'est pas
imposé mais une
4ème catégorie de
monstres qui
laissent tomber des
pierres sera
appréciée à sa juste
valeur

Quelques indications

Voici quelques conseils pour vous éviter de partir dans des choses inutilement compliquées.

- **Choisir un repère**

D'une manière ou d'une autre, il vous faudra un repère pour placer vos éléments et effectuer vos calculs de collision. Le plus simple est certainement de définir un repère cartésien fixe, le sol étant l'altitude 0. Le score du joueur sera naturellement l'altitude maximale atteinte par le Gribouilleur. La zone visible de jeu peut alors se concevoir comme une "fenêtre" de vue dont la position est relative à celle du Gribouilleur, afin d'accompagner sa progression (ou sa chute).

- **Les collisions**

Vous trouverez sur le web différentes façons de faire de la détection de collision. N'optez pas pour des choses complexes. Considérant la simplicité des formes des différents éléments du jeu, on peut raisonnablement se baser sur leur rectangle englobant pour effectuer des tests de collision (qui deviennent alors très simples)

- **Votre hiérarchie d'objets**

La façon dont vous allez concevoir votre hiérarchie peut faire de ce projet une lettre à la poste. Ou pas. Donc réfléchissez bien avant de coder vos classes. Interrogez-vous si vous constatez de nombreuses redondances. Si tout s'assimile à un rectangle, quelle conséquence sur votre hiérarchie d'objets ?

Evaluation

Votre projet sera évalué sur 2 aspects. D'une part la qualité de votre jeu et le respect des fonctionnalités demandées. N'hésitez pas à en

ajouter d'autres si vous le souhaitez. D'autre part, la qualité de votre implémentation. C'est à dire le bon usage des patterns objets et d'héritage, le bon usage des mécanismes de délégation, la bonne organisation de votre code et l'efficacité de vos algorithmes