

# Développer des jeux

## en HTML5 & JavaScript

Multijoueur temps-réel avec Node.js et intégration dans Facebook

**Samuel Ronce**



© Groupe Eyrolles, 2013,  
ISBN: 978-2-212-13531-2

**EYROLLES**

# Avant-propos

---

## Pourquoi concevoir un jeu en HTML5/JavaScript ?

Depuis des années, le langage Javascript était utilisé avec parcimonie pour quelques interactions avec l'utilisateur. Toutefois, les nouvelles versions de navigateurs web ont accéléré l'interprétation du Javascript, autorisant la création d'applications plus sophistiquées, et particulièrement de jeux. Et surtout, HTML5 a fait son apparition ; il propose de nouvelles balises et API pour JavaScript, afin de manier notamment la vidéo et l'audio, ainsi que la possibilité de dessiner en 2D ou 3D.

### **COMPATIBILITÉ** Les balises de dessin

Les balises de dessin (`<canvas>`) existaient déjà dans Safari en 2004. Cependant, elles étaient très peu exploitées ; car, d'une part, les autres navigateurs ne les utilisaient pas et, d'autre part, les navigateurs n'étaient pas assez puissants pour réaliser des jeux. La technologie Flash était encore bien présente.

HTML5 permet de dynamiser des pages web comme Flash le faisait antérieurement. Alors pourquoi ne pas rester sur des développements Flash, accessibles pour 99 % des utilisateurs, quand environ 40 % de ces derniers comprennent HTML5 ? La question se pose moins en termes d'accessibilité qu'en termes d'évolution : développer maintenant en HTML5, c'est anticiper sur la mise en place de produits sur de nouveaux marchés. Il suffit de constater l'explosion du marché des Smartphones – 1 milliard dans le monde prévu en 2013 – et des tablettes tactiles. L'intérêt de HTML5 réside dans sa disponibilité pour de nombreuses plates-formes. Votre jeu fonctionnera aussi bien sur iOS ou Android que sur les télévisions connectées.

**CONCURRENCE** Que devient Flash ?

Adobe a abandonné Flash Player pour les mobiles et déclaré sur son blog : « *HTML5 est désormais pris en charge par les principaux appareils mobiles, dans certains cas exclusivement. Cela en fait la meilleure solution pour créer et diffuser des contenus dans le navigateur sur différentes plates-formes mobiles* ».

Adobe a tout de même tenté d'implanter sa technologie AIR qui permet de « cross-compiler » vers ces nouvelles plates-formes. AIR étant sans doute plus mature que HTML5 aujourd'hui, il pourrait être envisagé de continuer avec *Actionscript*. Néanmoins, si le jeu est porté sur navigateur et essayé sur iPad, comment le joueur peut-il l'essayer sans HTML 5 ?

**REMARQUE** Jeu natif

Il est possible de faire tourner un jeu nativement en utilisant PhoneGap ou Adobe AIR 3.

Certes, HTML5 est encore jeune et en cours de standardisation, mais cela n'empêche pas des facteurs importants du Web d'en faire la promotion dès maintenant, voire de l'utiliser. Par exemple, il est possible de développer des applications en HTML5 sur Windows 8 UI de Microsoft. Inutile d'hésiter, utiliser HTML5 n'est pas une décision délicate pour une entreprise. Toutefois, selon le public visé, pensez à adapter votre application pour des navigateurs plus anciens tels que Internet Explorer 7 et 8.

**ANCIENNES VERSIONS** La rétro-compatibilité pour des jeux

La balise `<canvas>` pour les jeux HTML5 n'existait pas du tout sur la plupart des anciens navigateurs. ExplorerCanvas est un code JavaScript obligeant IE8 à interpréter cette balise.

Il est conseillé de demander à l'internaute de mettre à jour son navigateur, pour que son expérience du jeu soit plus agréable.

► <http://code.google.com/p/explorercanvas>

Devant l'enthousiasme général soulevé par HTML5, de nombreux projets d'adaptation ont été pratiquement abandonnés : dernier *commit* en 2010 pour ExplorerCanvas, dernière *release* de FlashCanvas en 2011.

Créer un jeu en HTML5, est-ce possible ? On est loin de la petite interaction de la part de l'utilisateur. Non seulement des algorithmes doivent être codés, ce qui implique une interprétation plus ardue du JavaScript, mais l'affichage graphique est également plus poussé avec des rafraîchissements continuels. Heureusement, les navigateurs web ont été largement améliorés et donnent désormais au développeur la possibilité de concevoir un jeu en HTML5.

Cependant, cela n'exclut pas les limites de l'interprétation et ne remplace pas la puissance de langages comme C ou C++, plus adaptés pour des jeux de très grande qualité que l'on retrouve sur des consoles de salon. Restons-en pour l'instant à la 2D en HTML5 ; programmer en 3D est actuellement prématuré, mais le jour viendra... Bien entendu, cela ne vous empêche pas de travailler sur la 3D avec WebGL.

**APPROFONDIR WebGL et Three.js**

La dernière annexe de l'ouvrage présente succinctement la conception d'éléments 3D avec Three.js.

Réaliser un jeu en 2D, n'est-ce pas un problème stratégique si l'on souhaite attaquer le marché des jeux vidéo ? Cela dépend de votre positionnement. A priori, si vous créez un jeu en HTML5, il sera distribué en ligne, ainsi que sur les nouvelles technologies (smartphones et tablettes tactiles). En ligne, soit vous créez votre propre site contenant votre jeu, soit vous utilisez les réseaux sociaux. On ne peut que conseiller la deuxième option (ou les deux) puisque les réseaux sociaux, tels que Facebook ou Google+, ont des millions d'utilisateurs qui sont autant de joueurs occasionnels ne cherchant pas la 3D HD mais l'amusement temporaire, seuls ou entre joueurs.

**STATISTIQUES Qui sont les joueurs des réseaux sociaux ?**

69 % des joueurs sont des femmes et ont en moyenne 43 ans (PopCap Games 2011). Ces statistiques sont loin des clichés sur les joueurs adolescents.

Un public varié, des plates-formes différentes... un jeu 2D, s'il est bien conçu, n'aura aucun mal à se placer dans ce marché, même en pleine génération 3D HD. Prenons des exemples concrets :

- Angry Birds, réalisé par Rovio, a engendré plus de 50 millions d'euros de chiffre d'affaires pour un budget d'environ 100 000 euros. Pourtant, le concept du jeu est très simple et la 3D n'est pas présente.
- FarmVille de Zinga, simulateur de vie agricole, est une application très populaire sur Facebook avec plus de 82 millions d'utilisateurs actifs.

Bien sûr, les échecs sont également nombreux dans le monde des jeux, mais c'est moins en raison de la 2D que pour des critères de « jouabilité » : sans mobiliser un budget conséquent, il suffit d'éditer un jeu addictif et amusant, pour lequel la compétitivité entre amis (faire le meilleur score, la plus grosse ville, etc.) devient très intéressante quand les réseaux sociaux s'en mêlent.

## À qui s'adresse cet ouvrage ?

Que vous soyez amateur ou professionnel, cet ouvrage vous fournira les bases pour la création d'un jeu en HTML5. Toutefois, il s'adresse aux personnes connaissant le langage JavaScript. Si vous souhaitez attaquer le marché des jeux vidéo, le livre complètera tout l'aspect technique par un chapitre sur les stratégies de monétisation pour récompenser vos efforts.

## Structure du livre

Le livre s'organise en trois parties dans un ordre logique :

- 1 la réalisation d'un concept de jeu ;
- 2 le développement technique ;
- 3 l'intégration dans les réseaux sociaux.

Le concept est une réflexion préalable sur la réalisation du jeu. Le chapitre 1 vous aide dans votre étude du marché et à vous diriger vers un contexte de jeu (Gameplay) attrayant.

Le chapitre 2 mentionne le chargement des ressources graphiques et audio avant le jeu et les différents écrans qui donneront la première impression au joueur.

Le chapitre 3 vous apprend à construire une structure pour l'importation des données dans le jeu afin de réaliser une carte ou un niveau et de s'aider d'un éditeur OpenSource.

Le chapitre 4 explique comment concevoir des animations à partir d'une image, ou déformer un élément selon une frise temporelle virtuelle.

Le chapitre 5 expose la réalisation du Gameplay avec le clavier, la souris, l'accéléromètre ou la manette. Il enseigne comment diriger un personnage en se souciant de son accélération, sa gravité, etc.

Le chapitre 6 détaille les méthodes de défilement de la carte pour pouvoir se déplacer dans l'intégralité du décor.

Le chapitre 7 montre comment créer les interactions entre le joueur et les décors pour gérer les collisions ainsi que les interactions entre le joueur et les autres personnages pour déclencher des événements.

Le chapitre 8 aide à mettre en place le concept avec l'application des règles du jeu.

Le chapitre 9 vous apprend à amplifier l'ambiance du jeu par des musiques et effets sonores ainsi qu'avec des effets graphiques.

Le chapitre 10 explique la conception des adversaires et le calcul des dégâts lors d'une attaque entre le joueur et l'adversaire.

Le chapitre 11 retrace les moyens pour sauvegarder et charger les données du jeu.

Le chapitre 12 résume les chapitres précédents par un cas pratique : la création d'un jeu plate-forme.

Le chapitre 13 mentionne l'installation de Node.js pour réaliser un jeu multijoueur.

Le chapitre 14 explique la création des modèles côté serveur pour le partage des données en temps réel.

Le chapitre 15 montre comment intégrer le jeu sur le réseau social Facebook avec authentification et autorisation du jeu.

Le chapitre 16 détaille la récupération des informations du joueur pour les partager avec ses amis et les inviter à leur tour de jouer.

Le chapitre 17 explique comment utiliser la monnaie virtuelle de Facebook comme stratégie de monétisation.

L'annexe *Rappels sur HTML5 Canvas* traite des différentes méthodes de l'élément `canvas` de HTML, utiles dans le dessin et l'affichage dans le jeu. Nous vous conseillons de la lire attentivement si vous n'êtes pas encore familiarisé avec cette balise.

Il existe plusieurs frameworks pour réaliser des jeux en HTML5. L'annexe *Frameworks Javascript* en mentionne deux : Easel.js et RPG.js.

La dernière annexe évoque la création de jeux 3D en WebGL avec l'aide du framework Three.js.

## Remerciements

Je remercie :

- l'équipe des éditions Eyrolles pour la publication de cet ouvrage, et particulièrement Muriel Shan Sei Fan pour le suivi de l'écriture ;
- David Dany pour quelques illustrations tirées d'un jeu coproduit ensemble ;
- Romain Pouclet pour la relecture technique ;
- Anne Bougnoux et Laurène Gibaud pour la relecture générale.

# Table des matières

---

<b>Avant-propos</b> .....	V
<b>Pourquoi concevoir un jeu en HTML5/JavaScript?</b> .....	V
<b>À qui s'adresse cet ouvrage?</b> .....	VII
<b>Structure du livre</b> .....	VIII
<b>Remerciements</b> .....	IX
 CHAPITRE 1	
<b>Mettre au point le concept du jeu</b> .....	1
<b>Étude de marché</b> .....	2
Quelques géants du Social Gaming .....	2
Étude démographique et comportementale .....	3
Étude technologique .....	5
Votre étude de marché .....	6
<b>Positionnement</b> .....	6
Joueurs ciblés .....	6
Prix du jeu .....	8
<b>Type du jeu</b> .....	8
Action .....	9
Jeu de rôle .....	9
Aventure .....	10
Action-Aventure .....	10
Simulation .....	10
Sport .....	11
Réflexion .....	11
<b>Choisir un framework</b> .....	11

Easel.js .....	11
Kinetic.js.....	12
Crafty.js .....	13
Caat.js.....	13
CanvasEngine.js .....	13
<b>Installer CanvasEngine .....</b>	<b>14</b>

## CHAPITRE 2

<b>Créer l'écran titre .....</b>	<b>15</b>
<b>Qualité des images.....</b>	<b>16</b>
<b>Chargement initial .....</b>	<b>16</b>
Création de la scène de chargement .....	17
Schéma des données.....	17
Affichage de la barre de progression .....	18
<b>Écran titre.....</b>	<b>21</b>
Création de la scène de l'écran titre .....	21
Initialisation des boutons .....	22
Association des événements.....	24
<b>Écrans additionnels.....</b>	<b>25</b>
Options.....	25
Niveaux.....	26

## CHAPITRE 3

<b>Affichage des décors.....</b>	<b>27</b>
<b>Level Design .....</b>	<b>28</b>
Insertion des données.....	28
Une carte .....	29
Un niveau .....	34
<b>Utiliser Tiled Map Editor.....</b>	<b>38</b>
Créer la carte .....	38
Intégrer la carte dans la scène .....	39
Obtenir des données de la carte.....	40
<b>Objets principaux : les sprites.....</b>	<b>40</b>
Ensembles d'éléments graphiques ou Spritesheets.....	41
Cas particulier .....	45

## CHAPITRE 4

<b>Animer les éléments sur l'écran</b> .....	47
<b>Déformer pour animer</b> .....	48
<b>Animation en boucle</b> .....	49
<b>Animer en réaction à une action</b> .....	52
<b>Animation temporaire</b> .....	53

## CHAPITRE 5

<b>Concevoir le Gameplay</b> .....	57
<b>Mouvement</b> .....	58
État d'un élément .....	58
Exemple : plate-forme mobile .....	58
<b>Contrôle du joueur</b> .....	61
Clavier .....	63
Souris .....	64
Accéléromètre .....	65
<b>Accélération et décélération</b> .....	71
Accélération .....	72
Décélération .....	73
<b>Gravité pour le saut</b> .....	75
Initialisation .....	76
Gravité .....	77
Saut .....	77

## CHAPITRE 6

<b>Avancer le joueur avec des défilements</b> .....	81
<b>Défilement classique</b> .....	82
Définir les éléments à déplacer .....	82
Rafraîchissement du déplacement .....	83
<b>Défilement différentiel</b> .....	83

## CHAPITRE 7

<b>Interaction avec les objets</b> .....	85
<b>Collision</b> .....	86

Au bord de la carte . . . . .	87
Sur le décor et les objets . . . . .	89
Collision sur des objets . . . . .	93
<b>Interaction</b> . . . . .	96
Déclenchement automatique au contact . . . . .	96
Selon une action . . . . .	99
CHAPITRE 8	
<b>Mise en place des règles du jeu</b> . . . . .	101
<b>Situation initiale du joueur</b> . . . . .	102
Inventaire . . . . .	103
<b>Application du concept</b> . . . . .	104
Affichage des points de vie . . . . .	104
Explication des règles au joueur . . . . .	106
<b>Affichage du score</b> . . . . .	108
<b>Fin de partie : gagnée ou perdue</b> . . . . .	109
Le joueur termine un parcours . . . . .	109
Le joueur sort de l'écran vers le bas . . . . .	110
CHAPITRE 9	
<b>Ambiance du jeu</b> . . . . .	111
<b>Ajouter des effets sonores</b> . . . . .	112
HTML5 Audio . . . . .	112
SoundManager . . . . .	113
Effectuer des fondus musicaux . . . . .	114
<b>Dynamiser avec des effets graphiques</b> . . . . .	115
Ton de l'écran : effet jour/nuit . . . . .	115
Flash visuel . . . . .	115
CHAPITRE 10	
<b>Les adversaires</b> . . . . .	117
<b>Paramètres des adversaires</b> . . . . .	118
Modèle et affichage des points de vie . . . . .	118
Calcul des dégâts . . . . .	121
Zones spécifiques d'interaction . . . . .	124

<b>Champ de vision</b> .....	127
Zone de détection .....	127
Réaction .....	128
CHAPITRE 11	
<b>Réaliser la sauvegarde</b> .....	131
Sérialisation des classes .....	132
Chargement des données .....	133
CHAPITRE 12	
<b>Cas pratique : créer un jeu plate-forme</b> .....	135
Règles du jeu .....	136
Initialisation et création des classes .....	136
Chargement du niveau .....	138
Création des animations .....	139
Défilement de la carte .....	140
Gestion des collisions .....	141
Mouvement, gravité et saut .....	143
Effectuer un mouvement selon une touche .....	145
CHAPITRE 13	
<b>Configurer le serveur pour le multijoueur</b> .....	149
Utiliser le serveur via SSH .....	150
Télécharger Node.js .....	151
Installer NPM .....	152
Installer Socket.io .....	152
Tester l'installation .....	153
CHAPITRE 14	
<b>Utilisez Node.js pour votre jeu multijoueur en temps réel</b> .....	155
Comment fonctionne Socket.io ? .....	156
Fonctionnement dans CanvasEngine .....	157
Définir les événements .....	157

Composer la structure du jeu . . . . .	158
<b>Créer des modules dans Node.js . . . . .</b>	<b>159</b>
Créer un modèle . . . . .	159
<b>Base de données . . . . .</b>	<b>162</b>
Schéma . . . . .	162
Données dans le modèle . . . . .	163
<b>Gérer les connexions et déconnexions . . . . .</b>	<b>164</b>
<b>Données communes . . . . .</b>	<b>165</b>
Partage des données entre joueurs . . . . .	166
<b>Sauvegarde et chargement avec Mongoose . . . . .</b>	<b>168</b>
Installation de MongoDB et Mongoose . . . . .	168
Connexion à la base et schéma . . . . .	169
Sauvegarder des données . . . . .	169
Chargement . . . . .	170
 CHAPITRE 15	
<b>Intégration du jeu à un réseau social : Facebook . . . . .</b>	<b>171</b>
<b>Déclaration du jeu dans Facebook . . . . .</b>	<b>172</b>
<b>Authentification et autorisation . . . . .</b>	<b>173</b>
 CHAPITRE 16	
<b>Implémentation de la partie Social Gaming . . . . .</b>	<b>177</b>
<b>Intégration du jeu HTML5 . . . . .</b>	<b>178</b>
Intégration du SDK et initialisation . . . . .	178
L'utilisateur est-il connecté ? . . . . .	179
Scène pour demander une connexion . . . . .	180
<b>Inviter des amis à jouer . . . . .</b>	<b>181</b>
Supprimer la notification . . . . .	182
<b>Afficher un score et le partager sur le mur de l'utilisateur . . . . .</b>	<b>183</b>
<b>Système de badges . . . . .</b>	<b>185</b>
<b>Récupérer des informations (amis, groupes, etc.)     pour les utiliser dans le jeu . . . . .</b>	<b>186</b>
Données générales . . . . .	187
Liste des groupes . . . . .	188

Liste des amis . . . . .	188
CHAPITRE 17	
<b>Stratégie de monétisation</b> . . . . .	191
<b>Monnaie virtuelle</b> . . . . .	192
Pourquoi une monnaie virtuelle ? . . . . .	192
Configuration et déploiement . . . . .	193
CHAPITRE A	
<b>Rappels sur HTML5 Canvas</b> . . . . .	197
<b>Initialiser et charger le canvas</b> . . . . .	198
<b>Dessiner dans le canvas</b> . . . . .	199
Les lignes . . . . .	199
Les arcs . . . . .	200
Chemins . . . . .	201
Les formes . . . . .	201
Recadrer . . . . .	202
Les dégradés . . . . .	203
<b>Afficher un texte</b> . . . . .	204
Couleurs . . . . .	205
<b>Les ombres</b> . . . . .	205
<b>Composite</b> . . . . .	206
<b>Insérer des images</b> . . . . .	207
Redimensionner . . . . .	208
Couper . . . . .	208
Répéter l'image en fond . . . . .	209
<b>Transformation</b> . . . . .	210
Translation . . . . .	210
Rotation . . . . .	210
Redimensionnement . . . . .	211
Transformation personnalisée . . . . .	211
<b>Manipulation des pixels</b> . . . . .	212

## CHAPITRE B

<b>Frameworks JavaScript</b> .....	215
<b>Easel.js</b> .....	216
Installer .....	216
Premiers pas .....	216
Ajouter des formes .....	216
Appeler le canvas en boucle .....	217
Afficher un texte .....	217
Ajouter des conteneurs .....	218
Insérer et animer une image .....	219
<b>RPG JS : créez des jeux de rôle</b> .....	220
Premiers pas .....	220
Transférer le joueur sur une autre carte .....	222
Créer un événement .....	222
Commandes événements .....	224
Ajouter dynamiquement des événements .....	225
Ajouter une animation .....	226
Créer des actions .....	229

## CHAPITRE C

<b>3D avec Three.js</b> .....	231
<b>Installation</b> .....	232
<b>Créer une scène</b> .....	232
<b>Ajouter un objet</b> .....	233
Former un groupe d'objets .....	234
<b>Source de lumière</b> .....	235
<b>Rendu</b> .....	235
<b>Bouger la caméra avec la souris</b> .....	236

## Symboles

### 3D

Three.js 232

## A

### adversaire

affichage des dommages 124

barre de vie 122

calcul des dégâts 121

champ de vision 127

paramètres 118

réaction 128

suppression 123

zone de détection 127

zone d'interaction 124

### ambiance du jeu 111

### animation 48, 51, 139

déformation 48

Ease (classe) 49

en boucle 49

réaction à une action 52

temporaire 53

## B

barre de progression 18, 19

### bouton

affichage 23

événement 24

initialisation 22

## C

### canvas HTML5

afficher un texte 204

arc 200

charger 198

chemin 201

composition 206

couleur 205

couper une image 208

dégradé linéaire 203

dégradé radial 203

dessiner 199

initialiser 198

insérer une image 207

ligne 199

manipulation de pixels 212

ombre 205

recadrer 202

rectangle 201

redimensionnement 211

redimensionner une image 208

répéter une image 209

rotation 210

transformation personnalisée 211

translation 210

### carte du jeu 29

affichage 32, 39

création 38

Tileset 30, 32, 39

### collision 141

### concept du jeu 1

### condition binaire 92

- contrôle 61
  - accélération 71, 72
  - accéléromètre 65
  - bouton de manette 69
  - clavier 63, 64
  - décélération 71, 73
  - écouteur clavier 73, 79
  - gravité 75, 77
  - joystick de manette 70
  - manette 67
  - saut 75, 77
  - souris 64
- création d'un niveau 138
- créer un modèle de données 28
- créer un niveau 34
  - schéma de données 35

## D

- décor du jeu 29
  - affichage 32, 39
  - création 38
  - Tileset 30, 32, 39
- défilement 140
  - classique 82
  - différentiel 83
  - du décor 81
- deviceorientation (écouteur) 67

## E

- Easel.js
  - canvas en boucle 217
  - conteneur 218
  - forme 216
  - image 219
  - installation 216
  - texte 217
- écran
  - niveaux 26
  - options 25
  - titre 15, 21
- effet graphique 111, 115
  - effet jour/nuit 115
  - flash visuel 115

- effet sonore 111, 112
  - fondu musical 114
  - HTML5 Audio 112
  - SoundManager 112, 113
- état d'un élément 58
- étude de marché 2
- événement 24, 157

## F

- Facebook 177
  - SDK 178
- framework 11
  - Caat.js 13
  - CanvasEngine.js 13, 14
  - Crafty.js 13
  - Easel.js 11, 216
  - Kinetic.js 12
  - RPG JS 220

## G

- Gamepad.js 68, 69
- Gameplay 57
- gravité 143

## I

- image
  - chargement initial 16
  - charger 20
  - qualité 16
- Image (classe) 20
- initialisation 136
- Input (classe) 61
- interaction personnage/décor
  - bord de la carte 87
  - collision 86, 87, 89, 90, 93
  - déclenchement automatique 96
  - déclenchement sur action 99
- Hitbox 86
  - sur des objets 89, 93
  - sur le décor 89, 90

**J**

jeu  
 authentification dans Facebook 173  
 autorisation dans Facebook 173  
 composer la structure 158  
 configuration du paiement 193  
 déclaration dans Facebook 172  
 jouabilité 57  
 joueur ciblé 6  
 JSON 17, 32, 133, 221

**K**

keyDown 63  
 keyPress 63

**L**

level design 28  
 ligne temporelle virtuelle 48

**M**

Math.max 122  
 Math.min 122  
 monétisation 191  
 MongoDB 169  
 monnaie virtuelle 192  
 mouvement 58, 143, 145  
 multijoueur  
 afficher un badge 185  
 afficher un score 183  
 configurer le serveur 150  
 connexion 180  
 connexion/déconnexion 164  
 gestion des données 162, 163  
 intégration du SDK Facebook 178  
 inviter des amis 181  
 Node.js 150, 151, 155, 159  
 NPM (Node Packaged Modules) 152  
 récupérer des informations 186  
 Social Gaming 177  
 Socket.io 152, 156, 164  
 statut du joueur 179  
 tester l'installation 153

**N**

Node.js 155  
 module 159

**P**

pause 108  
 plate-forme mobile 58  
 position de la souris 25  
 positionnement du jeu 6  
 prix du jeu 8

**Q**

qualité d'image 16

**R**

rafraîchissement 19, 22  
 règle du jeu 101, 136  
 affichage des points 104  
 affichage du score 108  
 application du concept 104  
 boîte de dialogue 106  
 fin de partie 109  
 inventaire 103  
 situation initiale du joueur 102

**RPG JS**

action 229  
 animation 226  
 carte 221, 222  
 événement 222, 224, 225, 226  
 jeu de rôle 220

**S**

saut 143  
 sauvegarde 133  
 chargement des données 133  
 Marshal (classe) 132, 134  
 sérialisation des données 132  
 Scrolling 81  
 Scrolling (classe) 82  
 situation initiale du joueur 102  
 Social Gaming 177  
 Socket.io 156

SoundManager 112, 113  
Sprite 40, 43  
Spritesheet 41

## T

Three.js  
  contrôleur 236  
  groupe d'objets 234  
  installer 232  
  objet 233  
  rendu 235  
  scène 232  
  source de lumière 235  
Tiled Mad Editor 32, 38  
Timeline 48  
type de jeu 8  
  action 9  
  action-aventure 10

Action-RPG 10  
  aventure 10  
  Beat Them All 9  
  combat 9  
  FPS (First Person Shooter) 9  
  Japanese-RPG 9  
  jeu de rôle 9, 220  
  MMO-RPG 10  
  multijoueur 29  
  Occidental-RPG 9  
  réflexion 11  
  RPG (Role Playing Game) 9  
  simulation 10  
  sport 11  
  Tactical-RPG 10

## V

variable privée 36