



# 11

## La création de jeux vidéo

### Au sommaire de ce chapitre

- La phase préparatoire en amont
- Le prototypage avec GameSalad
- Créer un jeu tactile sur iPhone
- Développer un jeu communautaire avec Game Kit
- Communiquer avec un autre périphérique en Bluetooth
- Développer un moteur physique au sein d'un jeu
- Pour aller plus loin

Figurant assurément parmi les catégories les plus populaires de l'App Store, les jeux vidéo épousent un cycle de développement très particulier. Même si l'histoire de la boutique en ligne d'Apple est émaillée de success-stories enthousiasmantes, qui ont propulsé des développeurs indépendants au sommet du classement et assuré leur gloire et leur fortune, la plupart des titres commerciaux sont le fruit d'un travail collaboratif au sein d'un véritable studio. Dès lors, il paraît illusoire de chercher à concurrencer l'expérience et le savoir-faire de prestigieux éditeurs qui, comme *Electronic Arts*, *Ubisoft* ou *Rockstar* ont bien compris l'intérêt de décliner leurs titres-phares au mobile et à la tablette d'Apple (voir Figure 11.1). Leurs plus gros succès impliquent la participation de dizaines d'experts, des graphistes 3D aux animateurs, en passant par les spécialistes en moteur physique ou *les level-designers*.

**Figure 11.1**

Grand Theft Auto : Chinatown Wars sur iPad a été développé par les équipes d'origine de Rockstar : il s'agit même de la version bénéficiant de la plus haute résolution, toutes consoles confondues.



Mais ne baissez pas les bras pour autant : avec son kit de développement complet et intuitif, iOS supporte des jeux de tout type d'ampleur. Il suffit de parcourir les étals de l'App Store pour s'en convaincre : les titres les plus marquants reposent parfois sur un code source très succinct, mais mettent en scène une idée diablement originale. Le musée des jeux vidéo est constellé d'histoires de ce genre, à commencer par *Tetris* par exemple. Conçu en 1984 par Alexei Pajitnov, il n'aura nécessité qu'un mois de développement pour aboutir au produit final. Mais en revanche, une vie toute entière ne suffit pas nécessairement à en trouver le concept ! En développant vos propres jeux pour iPhone ou iPad, vous renouez avec les balbutiements de cette industrie et vous figurez parmi les pionniers d'une plateforme d'un type nouveau, simple, ludique et conviviale.

À la lumière des enseignements des chapitres précédents, vous êtes d'ores et déjà capable de développer la plupart de vos idées : quiz s'articulant autour d'une base de données, intégration de séquences vidéo ou de musique, réactions aux gestes de l'utilisateur, interfaces constellées de boutons et de champs de texte... Vous pouvez exploiter tous les objets de base du kit de développement de l'iOS pour aboutir à un résultat simple et efficace, qui séduira rapidement des milliers d'adeptes. Mieux encore : depuis la version 4.1 d'iOS, vous pouvez même profiter de Game Center, le "réseau social" mis en place par Apple pour soutenir sa communauté de joueurs. Comparaison de scores, parties multi-joueurs et échanges en ligne : Apple prend en charge tout cet aspect de votre projet et vous aide à donner une autre ampleur à votre application. Dans le même ordre d'idée, des éditeurs indépendants ont mis au point des outils révolutionnaires capables d'aboutir à un jeu complet au prix d'efforts minimes.

Débridez votre imagination et envisagez le développement de ce type d'applications. Non seulement elle constituera la synthèse de tous les enseignements entrevus jusqu'alors, mais vous mettrez également toutes les chances de votre côté pour rentabiliser votre apprentissage en proposant un produit au fort potentiel commercial.

## La phase préparatoire en amont

Le développement d'un jeu vidéo, aussi simple soit-il, ne laisse nullement la place au hasard. Plus que jamais, vous devez plancher sur papier et mettre soigneusement au point les règles de votre jeu, tout en envisageant de multiples pistes afin de renforcer son aspect addictif ou son intérêt auprès de la communauté de joueurs. Il n'y a rien de pire qu'un titre bancal, dont on fait rapidement le tour avant de jurer que l'on ne nous y reprendra plus jamais ! À titre d'exemple, imaginez le cycle de développement du célèbre *Fruit Ninja*, développé par Halfbrick Studios (voir Figure 11.2).

**Figure 11.2**

*Fruit Ninja est un jeu extrêmement prenant, qui a été téléchargé plusieurs millions de fois.*



Envisagez ses caractéristiques principales et inspirez-vous-en :

- **Un concept simple et intuitif.** Dans *Fruit Ninja*, des fruits déferlent du haut de l'écran : avec votre seul doigt en guise d'arme, vous devez les trancher pour en extraire la substantifique pulpe. Librement adapté d'un mini-jeu disponible sur la Wii de Nintendo, ce titre tire partie à merveille des possibilités de l'iPhone et de l'iPad : vous en maîtrisez le fonctionnement dès les premières secondes, vous profitez de l'interface tactile des deux périphériques et même les très jeunes enfants peuvent s'y essayer.
- **Une identité graphique remarquable.** Avec des graphismes colorés et mignons, le jeu bénéficie d'une identité propre et pose les bases d'un genre à part entière. Afin de mettre en scène leur concept simple et intuitif, les développeurs ont imaginé cette histoire de samouraïs et de guerriers japonais : c'est un prétexte très habile pour "déguiser" cette bonne idée et profiter de l'écran tactile de l'iPhone et de l'iPad.
- **Un aspect communautaire.** Compatible avec les réseaux OpenFeint et Game Center, *Fruit Ninja* renforce son aspect communautaire : en solo, vous comparez facilement vos scores avec ceux des meilleurs joueurs mondiaux et vous disposez d'un mode multijoueurs particulièrement prenant. Le développement de ce type d'aspect communautaire est aujourd'hui simplifié depuis iOS 4.1 et l'apparition de Game Center.
- **Une prise en compte de l'iPad.** Le titre de Halfbrick Studios figure parmi les jeux les plus populaires de l'iPad. Rien d'étonnant à pareil succès : cette seconde version tire pleinement partie de la résolution de la tablette d'Apple et a été entièrement réécrite. Deux joueurs peuvent ainsi s'affronter face-à-face, avec un écran partagé en deux. Une excellente idée,

qui justifie à elle seule la commercialisation d'une version spécifique et un véritable gage du sérieux des développeurs !

- **Un excellent rapport qualité/prix.** Même s'il paraît fortuit de se préoccuper du tarif de votre jeu alors que vous n'en êtes qu'à sa phase préparatoire, sachez que les titres les plus populaires sont généralement proposés à un prix très bas, de l'ordre de 0,79 euros. C'est d'ailleurs le choix de Halbrick Studios et la raison partielle de son immense succès ! Bien entendu, les titres les plus ambitieux, qui sont des déclinaisons de véritables jeux pour Playstation 3 ou Xbox 360, sont proposés à un tarif plus élevé mais là encore, les studios se montrent raisonnables ; *Grand Theft Auto : Chinatown Wars* coûte 7,99 euros sur iPad alors que le jeu s'échangeait aux alentours de 35 ou 40 euros sur Nintendo DS et PSP.

Avant d'aller plus loin, vous devez ainsi définir les éléments de base de votre projet en vous inspirant de la démarche de ces studios professionnels. Commencez par jeter sur papier vos principales idées et choisissez le type de jeu que vous souhaitez développer. Chaque catégorie implique ses propres codes et langages qui, comme nous l'avons vu, participent au succès et à l'intérêt de votre application.

- **Un jeu d'aventure.** Si vous envisagez de développer un RPG massivement multijoueurs, autour d'une communauté forte et d'une interface en 3D, réfrérez vos ardeurs : ce type de projets implique des ressources démesurées, qui ne sont à la portée que des studios les plus prestigieux. À ce titre, *Final Fantasy* de Square Enix est un modèle du genre : il s'appuie sur un moteur graphique directement hérité des consoles de salon et met en scène des effets spéciaux spectaculaires qu'il vous sera difficile de reproduire (voir Figure 11.3). En revanche, vous pouvez parfaitement envisager le développement de titres très prenants qui s'articulent autour d'un scénario intrigant et méticuleusement rédigé. C'est le pari d'Agharta Studio avec sa collection de titres *1112* : les graphismes sont de pures illustrations et vous manipulez des objets à l'écran, dans la grande tradition des *point'n click*. Mieux encore : le récent *Hacker Run*, développé par un programmeur indépendant, s'appuie sur un véritable tour de passe-passe. Il vous met dans la peau d'un pirate informatique, qui doit cracker un système sécurisé : par le biais de commandes que vous saisissez dans le clavier virtuel, vous tentez tout type de mots de passe, vous parcourez de faux e-mails et vous affichez des structures de fichiers fictifs. L'interface n'est composée que de libellés et de champs de texte et le projet tout entier ne nécessite que des connaissances techniques minimales pour assurer son développement de bout en bout (voir Figure 11.4).

**Figure 11.3**

*Final Fantasy est un jeu très ambitieux qui s'inscrit dans la veine de la prestigieuse saga pour consoles de salon.*



**Figure 11.4**

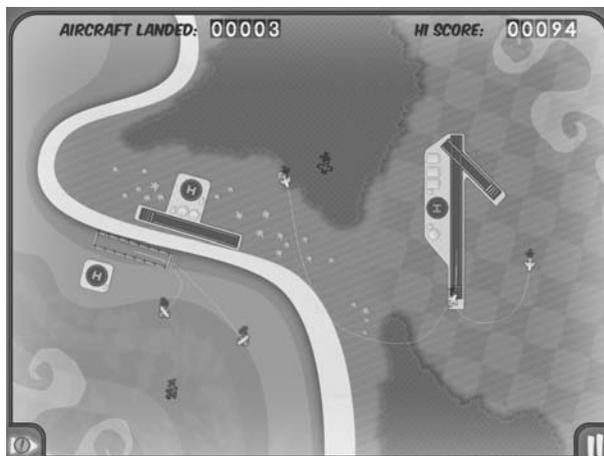
Hacker Run illustre à merveille l'intérêt d'un bon scénario dans un jeu d'aventure : il vous permet de vous dispenser d'une interface graphique haute en couleurs.



- **Un puzzle.** Figurant parmi les sous-catégories les plus populaires, les *puzzle-games* les plus célèbres (*Tetris*, *Columns*, *Puzzle Bubble...*) ont tous été portés sur iPhone et iPad. Rien d'étonnant : ils ont été conçus à la base pour des consoles de jeu portables et trouvent ainsi un prolongement naturel à travers l'interface tactile d'iOS. Avec ce type de projets, vous devez particulièrement soigner les règles du jeu et renforcer le challenge. Pour y parvenir, trois solutions essentielles s'offrent à vous : limiter la durée d'une partie (vous augmentez ainsi la pression !), limiter le nombre de coups que peut choisir le joueur (il doit ainsi minutieusement étudier ses déplacements) ou limiter l'espace dont dispose le joueur au fur et à mesure de la partie (à l'instar de *Tetris*, dont les blocs remplissent l'écran). Plus important encore : votre jeu ne doit pas souffrir du moindre défaut ergonomique et les joueurs de tout âge doivent être capables de le prendre en main en quelques secondes. C'est le pari que réussit haut la main *Flight Control*, qui vous met dans la peau d'un aiguilleur du ciel (voir Figure 11.5).

**Figure 11.5**

Avec *Flight Control*, vous êtes limité en temps et l'aire de jeu se réduit considérablement au fur et à mesure de la partie. Stress garanti !



- **Un jeu tactile.** Dans la mesure où l'iPhone et l'iPad s'appuient sur un écran tactile multipoints, vous auriez tort de ne pas en profiter : vous renforcez ainsi l'ergonomie de votre titre et vous pouvez développer un jeu tout entier qui repose sur les gestes de l'utilisateur. Doit-on encore présenter *Angry Birds* ? Développé par Rovio Mobile, ce jeu a fait le succès de ce studio et se décline aujourd'hui en d'innombrables clones (voir Figure 11.6). Pour gérer au mieux ce type d'interfaces, inspirez-vous des enseignements du chapitre précédent. Vous devez pleinement justifier l'utilisation de l'interface tactile : elle doit véritablement servir votre jeu et ne pas souffrir du moindre défaut.

**Figure 11.6**

*Décliné en de multiples épisodes et clones, Angry Birds est l'un des plus grands succès de l'App Store.*



**Figure 11.7**

*Avec Plants vs. Zombies, PopCap Games a créé un véritable défouloir humoristique, classé 9+.*



- **Un jeu d'action.** Dans le même ordre d'idée que les jeux d'aventure, les jeux d'action les plus ambitieux impliquent la participation d'une grande équipe de développement. Les conversions (réussies !) des épisodes de *Call of Duty* ou de *Street Fighter* en sont de parfaits exemples : tous les éléments des titres d'origine ont été exploités et l'expérience de jeu s'apparente alors à celle offertes par les consoles de salon. Dans un registre moins exigeant, l'App Store est constellé de purs défouloirs dans lesquels vous dézinguez à tout va des ennemis qui déferlent sur l'écran. Avec ce type de projets, n'hésitez pas à renforcer le côté loufoque, à l'image de *Plants vs. Zombies* : si le principe consiste bien à dégainer sur des sprites, les personnages sont totalement surréalistes (voir Figure 11.7). Crises de rire

garanties ! Par ailleurs, en mettant en scène ce type de bizarreries, vous avez l'assurance de vous adresser au plus grand nombre : comme vous le verrez au chapitre suivant, Apple ne plaisante pas avec le système PEGI (public conseillé) et votre jeu pourrait être interdit au moins de 17 ans s'il met en scène de véritables tueries. Songez-y à deux fois.

- **Un quiz.** C'est l'un des projets à travers lequel nous avons entrevu le développement d'applications iOS : les simples quiz ne mettent en jeu que des connaissances techniques minimales et restent très populaires sur l'App Store. Choisissez un thème très spécifique et soignez la rédaction des questions – pour habiller votre interface, vous n'avez besoin que d'éléments graphiques minimalistes et d'une poignée de boutons et de libellés. À ce titre, essayez d'adapter notre premier exemple à la lumière de vos connaissances actuelles : vous pouvez notamment ajouter une bande-son en arrière-plan ou des effets sonores qui accompagnent une bonne réponse.
- **Un jeu éducatif.** Bien qu'il s'agisse de coûteux périphériques, réservés en premier lieu à un usage professionnel, l'iPhone et l'iPad ont vite trouvé leur place au sein du foyer familial. Pour faire passer plus vite un long voyage en voiture ou accompagner nos chères têtes blondes dans la découverte de l'alphabet ou des merveilles du monde, de nombreux titres ne s'adressent qu'aux moins de sept ans. Là encore, vous disposez a priori de toutes les connaissances requises pour envisager de tels projets : ils s'articulent autour d'une interface colorée et prennent en charge les gestes de l'utilisateur. À ce titre, la série des *Talking Larry* est un modèle du genre : les graphismes sont mignons et l'interface se contente d'une série de boutons qui déclenchent des sons ou des réactions du personnage (voir Figure 11.8). Inspirez-vous-en !

### Figure 11.8

*Le jeu Talking Larry bénéficie d'une réalisation soignée et plait en quelques instants à tous les enfants. C'est un vrai plaisir de les voir dialoguer avec le personnage virtuel !*



Sitôt les premiers contours de votre projet arrêté, vous devez définir le cadre de votre histoire et de ses personnages (ou éléments) qui seront manipulés par l'utilisateur. À ce stade, ne bridez pas votre imagination et ne pensez même pas aux possibilités techniques de l'iPhone et de l'iPad, ni aux éventuels frameworks que vous devrez importer dans votre projet. Il est primordial de préserver un regard simple et intuitif sur l'ensemble de votre projet. En fonction de la catégorie de jeu que vous avez choisie, cette phase du développement implique des actions différentes : dans le cadre d'un puzzle, par exemple, vous identifierez les principaux challenges

que vous présenterez au joueur, tandis que vous étudierez le dialogue homme-machine avec un jeu d'aventure. Faites preuve de bon sens et mettez-vous dans la peau d'un joueur, sans limiter vos idées. Si vous avez un certain talent artistique, n'hésitez pas à esquisser sur papier les bribes de l'interface ou à préparer des croquis de tous vos éléments graphiques.

## Le prototypage avec GameSalad

Après cette phase initiale de brainstorming, vous devez vous retrousser les manches et rallumer votre ordinateur : votre projet est sur le point de commencer. Mais par où commencer précisément ? Il y a encore quelques mois, le développement d'un jeu vidéo nécessitait une immense rigueur : vous deviez développer chaque aspect du jeu de manière coordonnée, en envisageant le moteur physique ou la gestion des collisions par exemple tout en créant les classes de base des menus et des interactions avec le joueur. Cachets d'aspirine de rigueur ! Apparue au début de l'année 2011, ce service gratuit développé par un éditeur indépendant s'articule autour d'un véritable logiciel à installer sous Mac OS X. Vous créez ainsi votre jeu de toutes pièces en tirant partie d'une bibliothèque impressionnante d'éléments et de comportements, tout en profitant de modèles prêts à l'emploi qui miment les grands classiques de l'App Store (shoot-em-up, clones de *Doodle Jump* ou *Angry Birds*, gestion des collisions...). Vous avez ensuite la possibilité de tester directement le résultat sur votre propre iPhone ou iPad, ou de le diffuser directement sur l'App Store ou le site web de GameSalad. Si les vieux de la vieille voient d'un mauvais œil ce type d'outils qui simplifie le développement d'un jeu de grande envergure, d'autres studios ont bien compris son intérêt : il sert à définir le canevas de base et à s'assurer de la cohérence de tous les éléments de votre projet. On parle alors de *prototypage*.

Ouvrez votre navigateur web et rendez-vous à l'adresse [www.gamesalad.com](http://www.gamesalad.com). Au centre de l'écran, cliquez sur le bouton Download GameSalad Creator et procédez au téléchargement (voir Figure 11.9). Vous devez disposer de Mac OS X 10.6 pour profiter de cet outil gratuit (28,4 Mo). Son installation n'appelle pas de commentaires particuliers : effectuez un double-clic sur le fichier .dmg, puis déplacez GameSalad parmi vos applications.

Au premier lancement de l'application, vous devez vous créer un compte d'utilisateur. Cliquez sur le bouton Login puis sur le lien Create a profile. Saisissez alors votre adresse e-mail, votre identifiant et votre mot de passe avant de presser le bouton Create Profile. De retour dans l'écran d'accueil de GameSalad, indiquez ces précieux sésames puis authentifiez-vous. Cliquez ensuite sur le bouton New, sur le volet gauche de la fenêtre, afin d'entamer un nouveau projet (voir Figure 11.10).

Outre la première icône vide, qui permet de créer un nouveau projet de toutes pièces, vous découvrez ici une série de modèles développés par d'autres programmeurs. Repérez en priorité les titres destinés à l'iPhone et procédez éventuellement à une recherche, à travers le champ en haut à gauche. Dans le cadre d'un premier exemple, nous allons développer un authentique *shoot-em-up* dans la veine de *Galaga* ou *Space Invaders* : sélectionnez "Basic Shoot Em Up" dans la liste, puis cliquez sur le bouton Edit in GameSalad Creator.

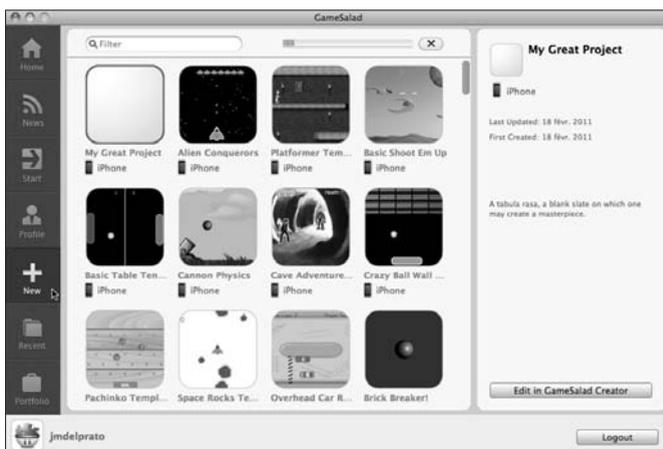
Figure 11.9

Téléchargez *GameSalad Creator*, l'outil de prototypage de jeux vidéo pour iOS.



Figure 11.10

L'écran des nouveaux projets est constellé de modèles prêts à l'emploi, qui constituent d'excellentes bases pour votre développement.



Vous découvrez alors l'interface de création de GameSalad Creator. Après avoir passé en revue la description du jeu, cliquez directement sur le bouton Preview pour l'essayer. À la manière du Simulateur iOS, le shoot-em-up se lance dans sa propre fenêtre : vous manipulez le vaisseau spatial avec le pavé fléché et vous lancez des missiles en pressant la touche Entrée (voir Figure 11.11). Naturellement, tous ces contrôles seront remplacés par une manipulation tactile si vous exportez ce projet vers l'iPhone, comme nous allons le voir.

Cliquez sur la flèche Back, en haut à gauche, afin de retourner à l'interface de développement. GameSalad Creator identifie les *scènes* (vues de votre application, des écrans de menu à l'interface du jeu) et les *acteurs* (tous les éléments graphiques) de votre projet à travers deux onglets. Dans le cadre de ce jeu, vous repérez ainsi deux vues et une dizaine de sprites prédéfinis (vaisseaux spatiaux, rayons laser, ennemis, éléments du décor, etc.). Effectuez un double-clic sur la scène principale afin de l'ouvrir dans l'éditeur (voir Figure 11.12). Tous les acteurs impliqués dans cette scène figurent dans la bibliothèque, sur le volet gauche de l'interface.

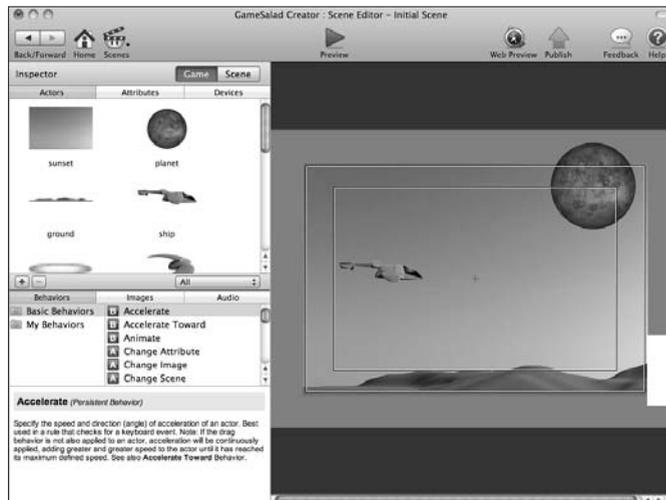
**Figure 11.11**

*Testez la réactivité de votre jeu et assurez-vous du moindre détail de son comportement.*



**Figure 11.12**

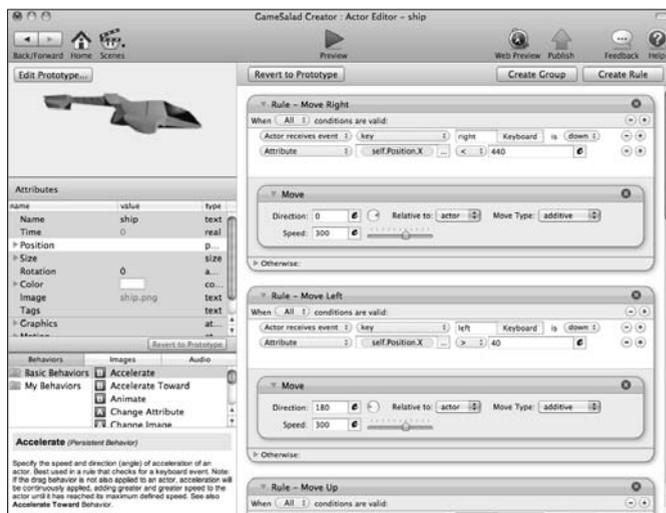
*GameSalad Creator permet de développer un jeu sans connaissance technique, par simples glisser-déposer d'éléments de la bibliothèque.*



À l'instar d'Interface Builder, GameSalad Creator vous laisse manipuler les éléments par glisser-déposer. Vous retrouvez ainsi le décor principal du jeu (un fond lunaire et un astre en arrière-plan) ainsi que son personnage central (un vaisseau spatial). Vous pouvez librement cliquer sur chaque élément ou ajouter un nouvel élément graphique. À titre d'exemple, déposez une seconde instance de la planète en arrière-plan, puis effectuez un clic droit sur cet objet et choisissez Send To Back. Vous personnalisez ainsi l'interface du jeu. Dans le même ordre d'idée, vous ajoutez n'importe quel élément graphique supplémentaire en cliquant sur le bouton +, en bas du volet gauche de la fenêtre à la section Inspector. Reportez-vous ensuite à la section Behaviors (comportements). Vous découvrez des dizaines d'effets prédéfinis, typiques des jeux vidéo : accélération, gestion des collisions, affichage d'un message texte, déplacement, mise en pause du jeu, rotation, apparition d'un nouvel élément graphique... Les possibilités sont quasi-infinies ! Effectuez un double-clic sur le vaisseau spatial afin de vérifier tous les comportements qui lui sont associés (voir Figure 11.13).

Figure 11.13

Tous les déplacements du vaisseau spatial dérivent de comportements prédéfinis par GameSalad Creator.



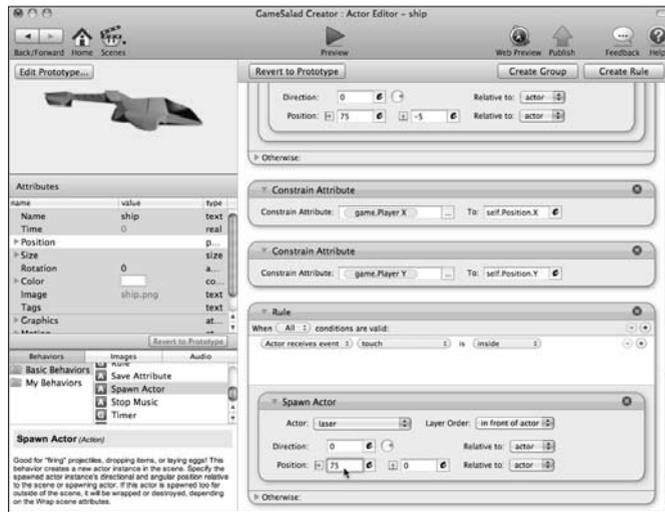
Sur le modèle de l'utilitaire Automator livré avec Mac OS X, GameSalad Creator dresse la liste de toutes les actions associées à votre sprite. Déplacements dans toutes les directions, gestion des tirs et apparition de nouveaux ennemis face au joueur : ce sont les trois comportements-types qui ont été personnalisés. Comme vous le constatez, les règles sont relativement simples et s'écrivent schématiquement de la manière suivante :

- Si l'acteur concerné reçoit l'événement  $x$  avec telle condition complémentaire, alors on déclenche le comportement  $y$  avec des attributs spécifiques.
- Plus concrètement : si le vaisseau spatial reçoit l'événement "flèche vers la gauche du clavier pressée" et que son attribut `self.Position.X` est supérieur à 40 pixels, alors on le déplace vers la gauche à une vitesse de 300.
- Pour le lancement d'un tir : si l'on presse la barre d'espace, alors on crée un nouvel acteur (spawn actor) "laser" à 75 pixels en face du vaisseau spatial.
- Pour l'apparition d'un ennemi : on crée un timer toutes les 0,1 seconde qui fait apparaître un acteur de type "ennemi" à 75 pixels du joueur.

Créez à votre tour une nouvelle règle, en adaptant les tirs du vaisseau spatial à l'écran tactile de l'iPhone. Cliquez sur le bouton Create Rule, en haut à droite, puis déroulez le menu face à "Actor receives event" et choisissez "touch is inside". Dans le volet gauche, à la section Behaviors, sélectionnez le comportement "Spawn Actor" puis déplacez-le dans la zone "Drag your behaviors here", sous votre règle. Choisissez alors l'acteur "laser" et faites-le apparaître face au joueur (in front of actor) à 75 pixels (voir Figure 11.14). Votre règle est aussitôt appliquée : il vous suffit de cliquer sur le bouton Preview pour tester le nouveau comportement que vous avez défini.

**Figure 11.14**

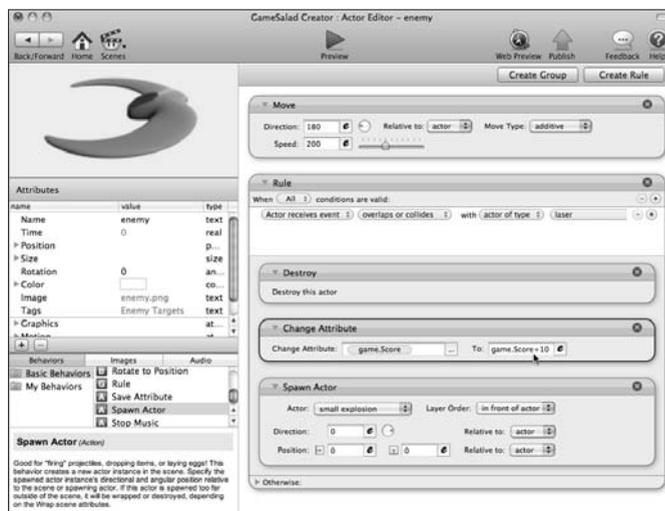
*Vous ajoutez en quelques secondes un nouveau comportement qui enrichit la manière de jouer.*



Complétez ces premiers enseignements par une série d'exercices. Les ennemis, par exemple, n'apparaissent pas dans l'interface principale du jeu. Vous devez effectuer un double-clic sur leur sprite, dans la bibliothèque, afin d'accéder aux règles qui leur sont associées. Vous découvrirez alors qu'elles sont au nombre de deux : un déplacement permanent vers la gauche et une suppression du sprite lorsqu'un élément vient le percuter (overlaps or collides). Dans ce dernier cas de figure, on enchaîne même par l'apparition d'un nouvel acteur, l'explosion, à la place du vaisseau ennemi détruit. Au passage, le joueur récolte 10 points supplémentaires à travers la simple règle "change attribute" (voir Figure 11.15).

**Figure 11.15**

*Il est possible d'enchaîner les règles de manière très intuitive : ici, un vaisseau ennemi détruit disparaît de l'écran et se voit remplacé par une explosion. Le joueur gagne au passage dix points.*

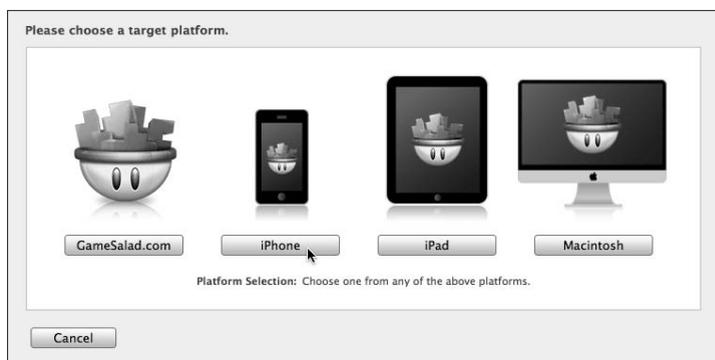


Vous avez compris le principe, envisagez maintenant de tester le résultat sur votre iPhone. Pour cela, cliquez sur le bouton Publish en haut à droite de GameSalad Creator. Sélectionnez ensuite

l'iPhone en guise de cible. Comme vous pouvez le constater, vous pouvez librement décliner votre projet à l'iPad ou au Mac Store (voir Figure 11.16).

**Figure 11.16**

*Grâce à l'outil de publication de GameSalad, vous testez et diffusez simplement votre projet.*



Passez ensuite en revue toutes les étapes de la publication, qui s'apparentent au protocole mis en place par Apple (reportez-vous au Chapitre 12 pour de plus amples détails à ce sujet). Vous devez ainsi rédiger la description de votre jeu et lui associer une icône et des captures d'écran (voir Figure 11.17). En dernier lieu, cliquez sur le bouton Publish, acceptez les conditions d'utilisation et patientez pendant la conversion. Vous obtenez ainsi un fichier .app que vous pouvez directement diffuser à travers iTunes Connect et vous avez également la possibilité de tester le jeu sur votre propre iPhone en cliquant sur le bouton Test. iTunes prend alors le relais et il vous suffit de glisser-déposer votre application vers votre mobile, connecté en USB, pour en profiter sans plus tarder (voir Figure 11.18).

**Figure 11.17**

*Prenez le temps de rédiger la description de votre jeu avant de le publier sur l'App Store.*

