GameSalad クリエーター

(Version 0.9.91)

Written by Jack Reed s
Layout by Anne Austin

目次

クリエーター使用ガイド ····· 3
はじめに ····· 3
システム要件 ····· 3
イントロダクション ····· 3
GameSalad ダッシュボード····· 3
シーンタブ ····· 4
シーン編集 ····· 5
アクター編集····· 8
ゲームの作成 ····· 10
シーン ····· 17
シーン属性 ····· 18
アクター ····· 19 **エラー! ブックマークが定義されていません。**
一般的属性 ····· 19 **エラー! ブックマークが定義されていません。**
図形- 関連属性 ····· 20
運動- 関連属性 ····· 20
物理的- 関連属性 (非運動) ····· **エラー! ブックマークが定義されていません。** 1
動作 ····· 22
属性 ····· 24
テーブル ····· 26
テーブルの作成 ····· 26
テーブルの編集····· 26
テーブルでデータを使用する ····· 26
特別なテーブル関数 ····· 27

GameSalad での公開 ····· 27

ヒントとベストプラクティス ····· 28

FAQ(よくある質問) ····· 28

トラブルシューティング····· 29

更なる支援 ····· 30

用語説明 ····· 30

クリエーター 使用ガイド

はじめに

システム要件
Mac OS X Snow Leopard (10.6.0) or later

イントロダクション

GameSalad クリエーターを使用すれば、コードを 1 行も書かなくても、ロジックとアセットを使用して、オリジナルのゲームやアプリケーションを作成することができます。
ここで「**ロジック**」とは、プロジェクトの動作を定義するルール、動作、属性の組み合わせを指し、「**アセット**」とは、プロジェクトにインポートした画像や音声のことを指します。

> ヒント: 本書を通して、太字になった新しい単語が数多く出てきます。その定義については、39 ページを参照してください。

これはクリエーター独自のデザイン、高度なロジックとアセットを組み合わせて高品質の製品を作り出す事により可能になりました。私たちにとって、「ロジック」とは、連帯してプロジェクトの動き方を定義する「**ルール**」、「**動作**」、および「**属性**」の組み合わせのことであり、「アセット」とはプロジェクトに読み込む画像やサウンドのことです。

> ヒント: 本マニュアルでは、たくさんの新規用語を太字で示して紹介します。それらの定義については 43 ページを参照してください。

このあと数ページにわたって、最初にすべきこと、および重要な背景情報をいくつか説明して、ご自身の作品を作り始められるようにサポートしていきます。まず、GameSalad クリエーターをダウンロードして下さい。

Windows 用クリエーターをダウンロードするには、http://gamesalad.com/get-windows-creator to get the most recent version にアクセスしてください。その後、インストールすれば準備は完了です!

> ヒント: GameSalad のアカウントがなくてもクリエーターをダウンロードして使うことはできますが、公開するにはログインする必要があります。まだアカウントを作っていない方は、是非、この機会に登録を完了して下さい。

GameSalad ダッシュボード

クリエーターを起動して最初に表示される画面がダッシュボードです。
この画面では、ご自身のアカウントにログインし、それまでに公開したプロジェクトの一覧を見たり、サンプルプロジェクトのテンプレートを検証したり、自分だけのプロジェクトを作成することができます。

クリエーターのオプションに慣れてきたら、「新規」タブからプロジェクトテンプレートを開いて見てみましょう。まずは「My Great Project」をダブルクリックし、新しいプロジェクトを始めてみましょう。

これであなたもクリエーターのユーザーです。最初の画面は以下のようになります。

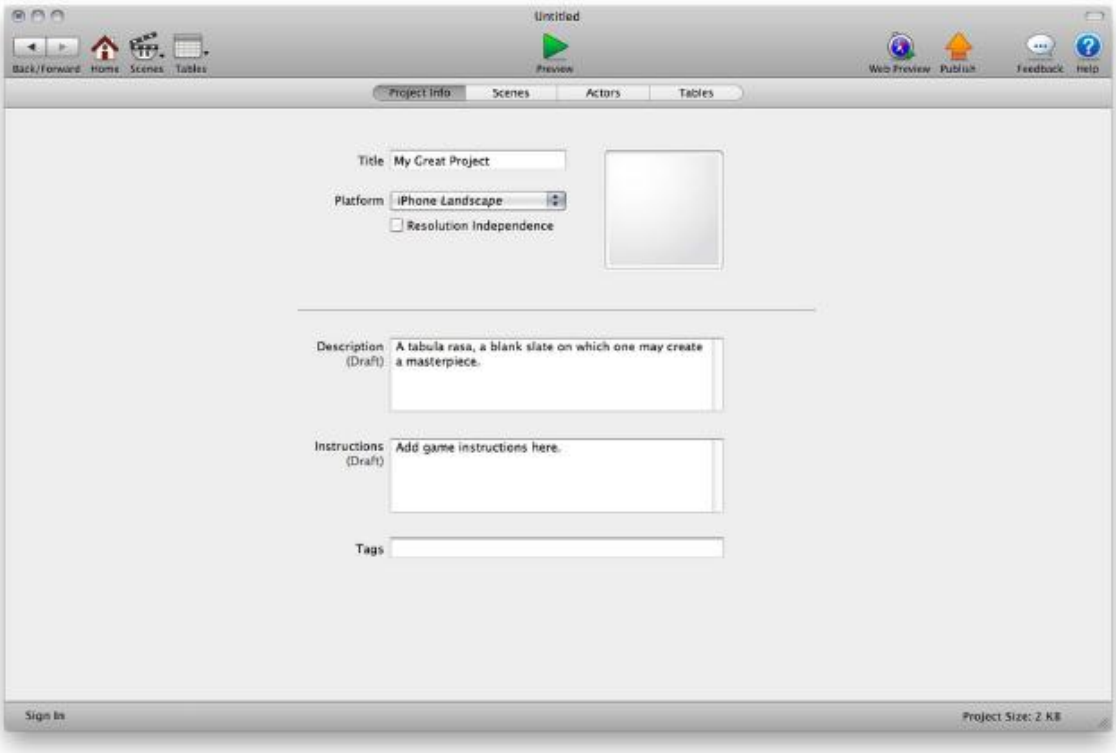

Image 1: A new blank project

心配は不要です。今のところ、テキストフィールドは全て「そのまま」にしておいて構いません。ここで決めなければならないのは、どのプラットフォーム向けのアプリを開発するのかということです。
これは後から変更することもできますが、この設定はプロジェクトの初期解像度に影響します。
開発の対象とするデバイスを選択しておけば、長期的に見て、時間と手間が省けます。

ここでは、プラットフォームはデフォルトの「iPhone (横画面)　」にしておき、クリエーターの上部の緑のプレビューボタンのすぐ下にあるシーンタブに移りましょう。

シーンタブ

今はデフォルトのシーン属性がいくつか存在するだけの、空っぽのシーンが一つあるだけです (詳細は後述)。
左下部にある「+」ボタンをクリックすることで、さらにシーンを作成することができます。また、既存のシーンの名前をクリックすれば、名前を変更する事も可能です (初期シーンは以下のスクリーンショットの通りです)。

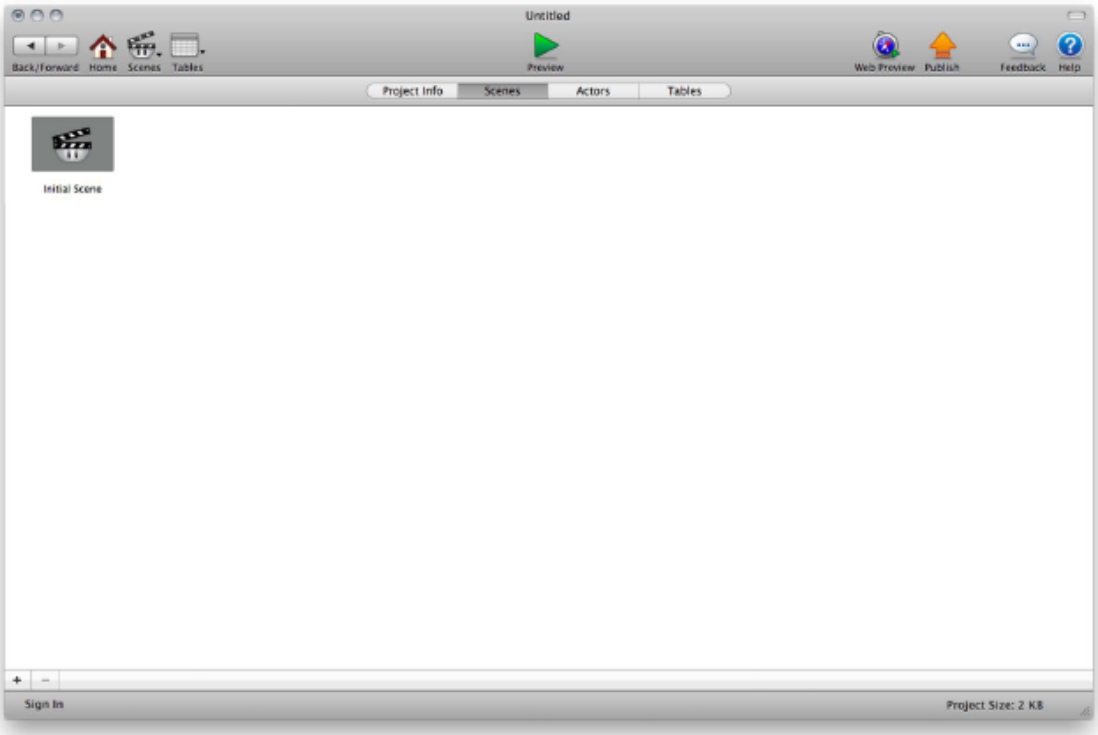

Image 2: A listing of all of the scenes in your project

シーンを利用することで、主にプロジェクトを分割することができます。それぞれのシーンに対しては、シーン属性 (サイズ、**重力**、**X/Y ラップ**) やアクターの配置を含む個別の設定をすることができます。
GameSalad では、シーンの変更は自動的には行われませんが、動作によって行われます (「シーンの変更」と呼ばれます)。
動作については、後の手順で再度解説します。

ヒント: 新しいシーンを追加するときに考えなければならない 2 つの主要因は「次のシーンがロードされている間、プレーヤーが待たされることがあっても問題ないのか」ということと「アクターと動作を追加し続けることで現在のシーンが重くなり、パフォーマンスが低下する可能性があっても大丈夫なのか」ということです。

シーンの編集

「初期シーン」をダブルクリックしてシーンの編集を見てみましょう。
ここでは新しいオプションや知識を一度にお伝えしましたが、慌てないでください。
基本はとても簡単です。そして、それこそがここでお伝えしたいことです。

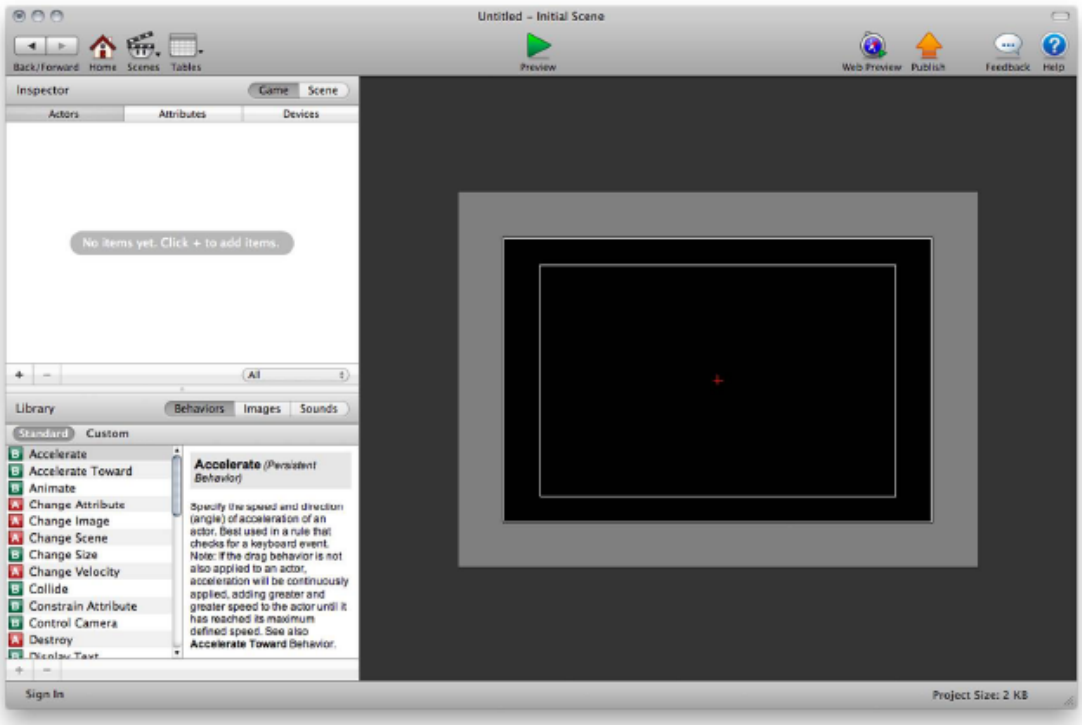

Image 3: The Scene Editor

左下部にある動作ライブラリには、標準的な動作、カスタム動作、そして場合によっては Pro またはバージョン固有の**動作**が含まれます。
他にも、画像や音声のアセットにアクセスするための「イメージ」及び「サウンド」という名称のタブがあります。
画像と音声のライブラリは最初は空ですが、適切なファイル形式であれば、簡単にクリエーターにインポートすることができます (繰り返しになりますが、「+」ボタンをクリックすることで行えます)。

> ヒント: 画像ファイルは PNG 形式でなければなりませんが、他の形式のファイルは自動的に変換されます。使用可能な音声ファイルの形式は、.ogg、.m4a、.wav です。

左上部にはゲームインスペクターがあり、デフォルトではアクターのタブに設定されています。
アクターは、どのプロジェクトにおいても主なエージェントとなり、各アクターは作成中のプロジェクトのアクターターブで見ることができます (このシーンでは使われていません)。
ここでは、新しいアクターを作成したり、既にあるアクターを編集したりすることもできます。

プロトタイプアクターがシーンにドラッグされると、**インスタンスアクター**が作成されます。これは基本的にはプロトタイプアクターのコピーです。
新しいインスタンスアクターはプロトタイプアクターの全ロジックと、ほとんどの属性を持っています。
インスタンスアクターは元のプロトタイプに「ロック」されています。インスタンスアクターがロックされている時には、プロトタイプアクターへの全ての変更 (若干の例外を除く) は、各インスタンスアクターにも影響します。

このようにして、同一のプロトタイプと関連づけられたインスタンスアクターをいくつも作成することができます。
時間をかけて複数のアクターを一つ一つ全く同じように編集するのではなく、たった一つのプロトタイプを編集するだけでいいのです。

ヒント：プロトタイプアクターの編集中には、タイトルバーに (プロトタイプ) という表示が出ます。同様に、インスタンスアクターの編集中には、「プロトタイプに戻る」ボタンが表示されます。このボタンは、インスタンスアクターがロックされている時にはグレーアウトされており、このボタンを押すと、ロックされていないインスタンスアクターは再度ロックされ、プロトタイプアクターから変更した属性及びロジックは取り消されます。

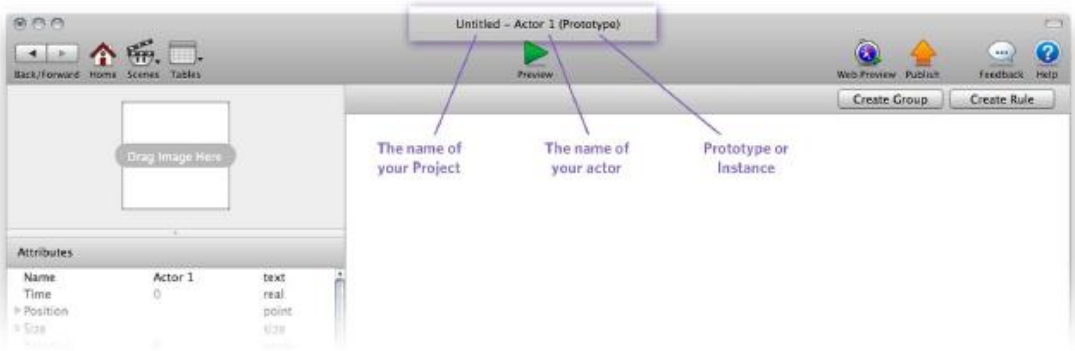

Image 4: Breaking down the title bar

シーンエディタの中で最も馴染みのないものが属性タブです。属性はクリエーターの最も強力な機能で、基本的には値を入れるプレースホルダーです。属性はシーン属性、ゲーム属性、アクター属性 (セルフ属性ともいいます) の 3 つに分類されます。
これらはクリエーターの別々の箇所に存在しますが、ここでは特に、**アクタータブ**のすぐ右にあるゲーム属性について解説します。

Image 5: Locating the Game Attributes tab

属性タブ表示中に「+」ボタンをクリックすると、新しいゲーム属性のタイプの選択画面になります。ブール値、テキスト、整数、実数、角度、指数から選択します (それぞれの違いについては、31 ページの属性の節で解説しています)。
この中には、よく使用するものもそうでないものもありますが、**アクター**がこれらの値を参照、変更したり、これらの値に反応したりできるので、それぞれ非常に有効です。

属性は、プロジェクトの変数です。プロジェクト内のアクターは、いつ新しい敵を登場させるか、主人公がどれだけのライフや弾薬を持っているのかといったことを判断し、ゲームオーバーの表示を出し、あるいはハイスコアを更新したことをプレーヤーに知らせます。

次に、初めての**アクター**を作成し、ルールと動作というクリエーターの機能を学んでみましょう。

アクタータブまで戻り、「+」ボタンをクリックして新しい**プロトタイプアクター**を作成し、新しいアクター上でダブルクリックしてください。

ヒント :この時点では、どのシーンにも、新しく作成したアクターは存在しません。もちろん、厳密に言えば、プロトタイプアクターがシーンの中に存在するというのは不可能です (シーンの中に存在するアクターは全てインスタンスアクターであるため) が、ここでは、プロトタイプから作成されたインスタンスアクターについての話をしています。ロックされた新しいインスタンスを作成するためには、プロトタイプアクターをシーンにドラッグするという作業が必要になります。

アクターの編集

クリエーターの主な作業のほとんどが、ここアクターの編集で行われます。ここからはプロジェクトのオペレーションを決定するルールと動作を設定することができます。

先ほど作成したアクターにはデフォルトの属性をいくつか持っていますが、ルールと動作はまだありません。右上部 (青いヘルプボタンのすぐ下) にある「ルール作成」ボタンをクリックし、ルールを作成してみましょう。

これで、新しいルールが出来上がりました。これがあなたのゲームの特徴となります。動作を追加し始める前に、ルールを作成する各コンポーネントを精査してください。条件部分 (ドロップダウンメニューの一番上) で、いつそのルールを適用するのかを決定します。

ルールが適用される可能性があるのは、ユーザーによるインプット (キーボード入力、マウスクリック、タッチスクリーン入力、加速度変化) があった場合、イベント (アクター同士の衝突、ゲーム内またはデバイス内である時間になる) が起こったとき、ゲームまたはアクター属性が特定の値になった (または特定の値でなくなった) 時です。

ルールが適用されると、ルールは動作部分 (グレーの部分) に記述されている「適用時に何をするのか」を参照し、リストアップされている順に全ての動作の実行を試みます。

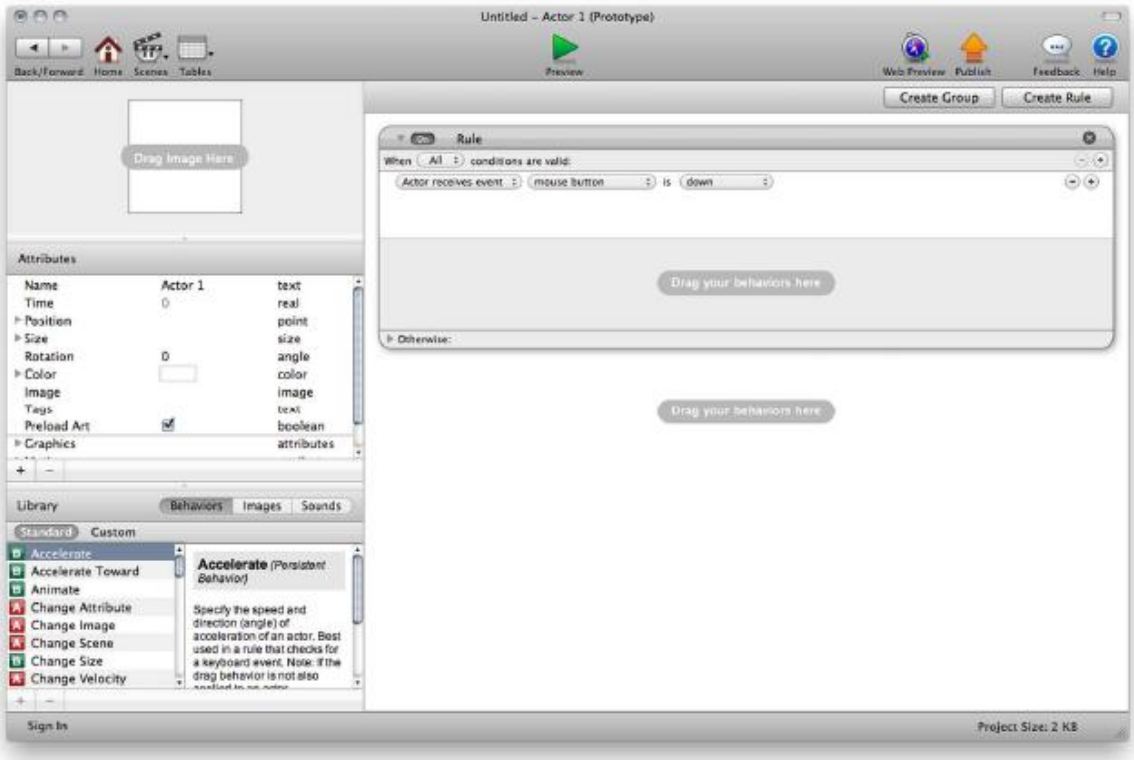

Image 6: Adding a Rule to an Actor

ヒント: **ルール**と**動作**は上から下へと順に実行されるということを認識しておくことが重要です。このことは、もしシーン変更の動作を 1 つのルール内に記述したとすると、2 つ目のシーン変更の動作は (ルール自体は有効で、適切に適用されたとしても) 決して実行されないということを意味します。

ルールの 3 つ目および最後の部分は、「その他手段」で、デフォルトでは折りたたまれ、隠れています。この部分は、「ルールが適用されない場合、どうすればいいのか」をルールに伝えるものです。
これを適切に使用するには注意が必要なので、もっと快適にクリエーターを使用できるようになるまでは、差し当たり最初の 2 つの機能を使用することに集中してください。

ここまでくれば、可能性は無限大です。ルールを使用すれば、あなたの作成するゲームにいくらでも修正を加えることができます。**アニメーション**を表示させたり、**アクター**をシーン内で動かしたり、音楽や音声を再生したり、新しいアクターを登場させたり、プレーヤーの視点を変更したり、いくつもの属性値を変更したりできるのです。

そして、これらの属性が他のルールが適用される場面を作ることもあり、そうこうしているうちにゲームが出来上がってしまうのです。　ここまでで、GameSalad クリエーターの基本的な機能を確認できたので、次は簡単なプロジェクトを作成する練習をしましょう。

ヒント: 解説を読むだけではなく、ここまでで実際に GameSalad クリエーターを使ってみるに十分な知識が身についています。もちろん、続きを読むのを諦めさせようとしているわけではありません。まだまだ学ぶことはありますので、頑張って欲しいと思います！

ゲームの作成

本書で触れていないメニュー、オプション、特徴がたくさんありますが、リストや固有名詞で頭をいっぱいにするよりも、新しいゲームを実際に作成してみましょう。既に用意されているテンプレートを開いてみましょう。ここでは画像や音声のアセットが利用できます。

既に用意されているテンプレートを開いてみましょう。ここでは画像や音声のアセットが利用できます。　シーンタブとアクタータブをクリックすれば、このテンプレートが既に作業されたものだとわかると思います。　アクターはそのままにしておき (このチュートリアルで後ほど使用するため)、シーンについては、両方とも、選択後に、左下部の「-」 (マイナス) ボタンをクリックし、削除してしまいます。

ヒント：テンプレートのオリジナルバージョンは、いつでもダッシュボードから呼び出すことができますので、プロジェクトに独自の変更を加えることについての心配は不要です。

では、「+」ボタンをクリックし、出発点となる新しいシーンを作成しましょう。デフォルトのシーン名 (シーン 1) をクリックし、このシーンで何が起こるのかを思い出せるように、「空中戦闘シーン」 (かぎ括弧は不要) というシーン名にしてみましょう。

シーンをダブルクリックしてシーンの編集に入ると、以下のように表示されます。

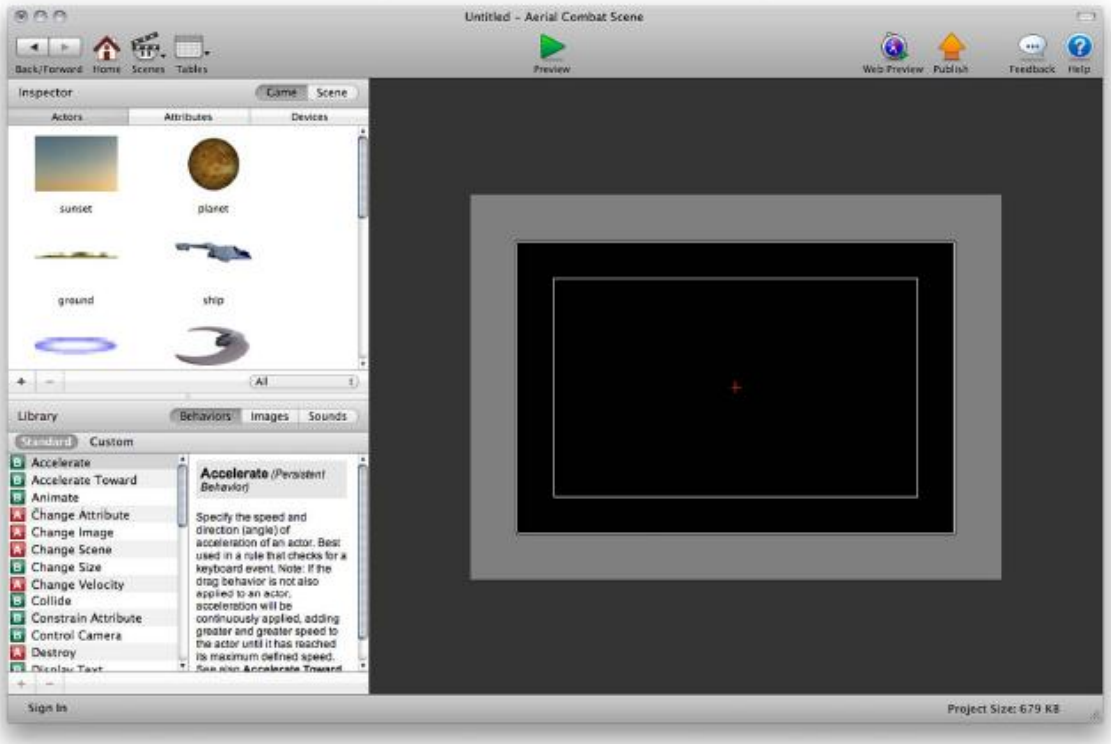

Image 7: Aerial Combat scene

繰り返しになりますが、アクタータブに既にあるアクターは削除しないでおきましょう。ただ、今は使用しません。ここでは「+」ボタンをクリックして、新しいアクターを作成してください。

「アクター1」がアクタータブのリストの一番下に表示されていることがわかります。

「アクター1」をクリックしてアクター名を「プレーヤー戦艦」に変更し、白い四角を画面左側のリストから画面右側のシーンへとドラッグしてください。

まだ見栄えがよくないですが、もう数分もすれば花火が見られますよ。
より正確には、戦艦による空中戦闘ゲーム内でミサイルが発射されるわけですが、もし望むなら、その通りに花火を加えることもできるのです。

ここには、インスタンスアクターがシーンに配置され、単独で存在しているだけで、まだ同じ名前のプロトタイプアクターにロックされています。

ヒント: もう一つ「プレーヤー戦艦」アクターをシーンに配置した場合、2 つのアクターは同じ名前を持ち、シーン上での位置以外、基本的にあらゆる点で全く同一です。それぞれの X 座標、Y 座標は、各アクターに最初から含まれる 2 つのセルフ属性で特定されます。各インスタンスアクターは独立してそれぞれのセルフ属性を保有しているので、セルフ属性は、ゲーム属性とは異なります。つまり、もし敵艦の「現在の状態」というセルフ属性があった場合、新しく登場する各敵艦は、それぞれ独自のセルフ属性値を持つということになります。もしこれを、1 つのゲーム属性で行おうとすれば、1 隻の敵艦にダメージを与えれば、全ての敵艦にダメージを与えたことになってしまいますが、セルフ属性を使えば、各敵艦の状態は個別に管理されるのです。

シーンのインスタンスアクター (現時点ではただの白い箱ですが) をクリックすると、以下のような画面になります。この個別のアクターエディタ画面について強調しておくべきことがいくつかあります：

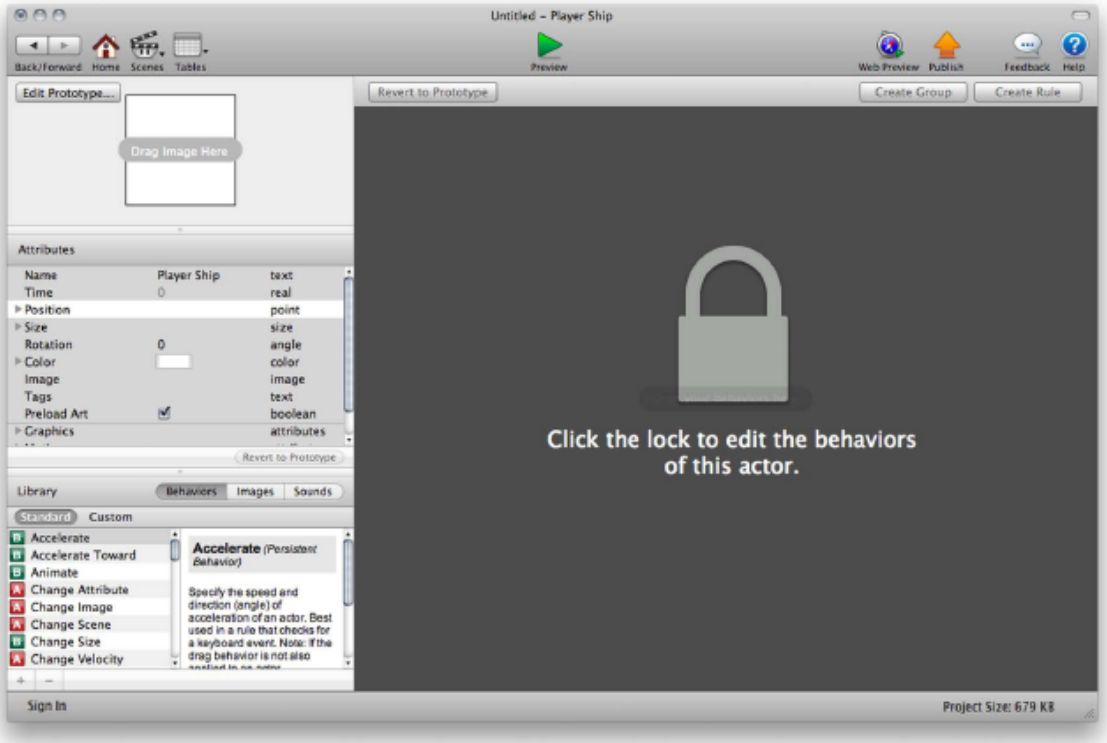

Image 8: A locked instance actor

ロックされたインスタンスアクターについては、本書で数回触れています (ロジックと属性が、作成元のプロトタイプアクターと同じになるアクター) が、ここで扱うのは、ロックされたインスタンスであることを知らせる「強力」な」タイプのアクター編集内通知です。

ロックをクリックすると、このインスタンスに変更を加えることができるようになり、その変更はプロトタイプアクターには影響を与えません。
同様に、ロックを解除した場合、プロトタイプを変更しても、インスタンスアクターには影響を与えなくなります。
インスタンスアクターを再度ロックするには、プレビューの下の「プロトタイプに戻る」を選択してください。

ヒント: ロックされたアクターはほとんど点で作成元のプロトタイプと完全に同一ですが、全く同じというわけではありません。例えば、各インスタンスアクターは、個別の幅と高さの属性 (サイズ) や、X 座標と Y 座標の属性を持ちます。インスタンスアクターのこれらの値を変更すると、プロトタイプとインスタンスのロックは解除されます。

ここでは、プレーヤー戦艦インスタンスアクターはロックされたままにしておき、左上部の「プロトタイプを編集」を選択してください。ここでは、タイトルバーが「タイトルなし - プレーヤー戦艦 (プロトタイプ) 」となっていることに注意してください。

まもなく、このアクターにルールと動作を追加し、ゲームにどのように動作するかを指示することになります。
しかし、その前に、このアクターを白い箱のままにしておかず、戦艦画像のアセットに変えましょう。左下部の「イメージ」タブを開き、「戦艦」の PNG ファイルを左上部の白い箱の上にドラッグ&ドロップしてください 。

これでずっとよくなりました！ですが、この幅 100、高さ 100 のサイズでは自慢の戦艦が少し押しつぶされたように見えてしまいます。属性ウィンドウで、この属性を 75x50 に変更してください。幅と高さのオプションは「サイズ」のドロップダウンリストの下にあります。このサイズ調整は、プロトタイプアクターと、以後このプロトタイプから作成されるインスタンスアクターのみに影響を与えます。
後ほど、既に存在する「プレーヤー戦艦」インスタンスアクターを開き、同じようにサイズ調整をしてみてくだい。

このプロトタイプにルールと動作を追加するにはまず、何をさせたいのかを決めなければなりません。
この戦艦はプレーヤーを表しますので、プレーヤーが戦艦の動作を制御し、武器も発射できるようにしなければならないでしょう。これに加えて危険要素を導入し、プレーヤー戦艦がダメージを受けすぎた場合、破壊されてしまうというようにしましょう。

まずは動きに集中し、またシンプルにする為に加速時計やタッチスクリーン入力ではなくキーボード入力を考慮したルールを作ります。

ヒント: タッチスクリーンデバイスでの動作、ユーザーインターフェースを設定する例としては、GameSalad ランチャーの「公式クロスプラットフォームコントローラーテンプレート」を参照してください。

[ルールの作成] ボタンを使用して、新しいルールを作成します。次に、[移動] 動作を最下部にドラッグします。いくつか簡単な調整を行うと、次のようになります：

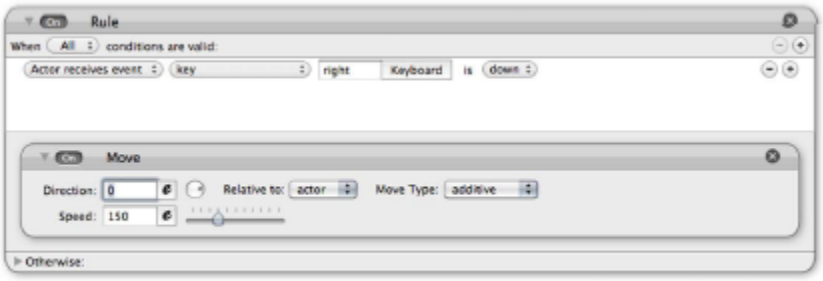

*Image 9: Setting up a **rule** for moving right*

キーボードの右矢印を押すと、'プレーヤーシップ' アクターが現在より右の位置に移動します。

他の 3 つの方向を手動で作成するのではなく、このルールを選択して、Alt キーを押しながら、下の空白部分にドラッグしてください。

これで、選択したロジック (および、その中に含まれるすべてのロジック) が正確にコピーされます。
後は、この新しいロジックに対し、条件となるキーと移動方向を編集すれば完了です。
[プレビュー] を使用して、テストしてみましょう!

キーボードを利用してあなたの船を動かせる一方、プレーヤーが見えない領域に飛ばすことも可能です。

プロトタイプアクターである“船”（私達が作った“プレーヤーシップ”ではない）を開く場合、元のテンプレートの中でこの問題がどのように対処されているか見ることができます：。

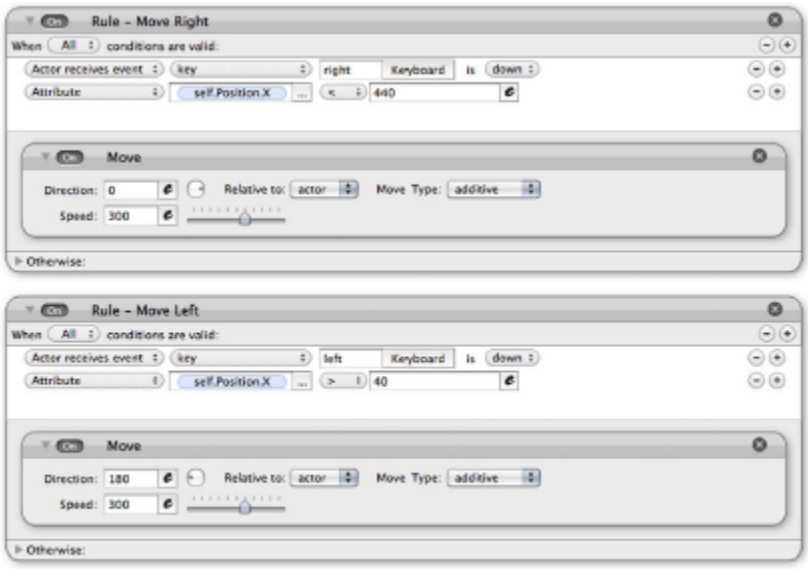

Image 10: Keeping the ship inside the viewable area

オリジナルの“船”アクターには動き動作が引き金となる様に右矢印キーが押されたのと同時に実現されるもう 1 つの条件があります。(“次の全てが実行される時”ルールの一番上にある必要条件に注意してください)
原則として、他の 3 つの移動方法のルールを定義したら、'機体' アクターはゾーン内の表示領域の外へは移動できなくなります。

このように、自分のアクターには自由にルールを追加してください。ただ、ここであまり深く考え込むことはありません。このチュートリアルは、手始めに体験してもらうことが目的です。ロジック構成の例をできるだけたくさん示そうとしているわけではありません。

ヒント: テンプレートのルールは“ルール-右に動く”とされていますが、我々が作成したルールは単に“ルール”と名付けています。ルールもしくは行動のタイトルバーをダブルクリックすれば名前を変更することができます。適切な名前を使うことであなたもしくはあなたの共同作成者は、すぐにその部分のロジックが何を管理もしくは影響するものであるかを知る事が出来るでしょう。これにより過去に行った仕事を振り返る時に多大なる時間を節約することができます。

ちょうどいいタイミングなので、ここでこれまでの作業を振り返り、他にどんな作業が残っているかをもう一度確認してください。(まだプロジェクトを保存していない方がいるようでしたら、こまめに保存することもお忘れなく)。
これは、サンプルのプロジェクトなので説明を略しますが、他にも基本的な要素をいくつか追加することが必要になります。たとえば、背景、弾を撃つ機能、弾を撃つ相手となる (そして、こちらに弾を撃ってくる) 敵、勝ちの条件、敗けの条件などです。

次の手順では、新しいプロトタイプのアクターを作成します (シーンの編集に戻る必要があります。最上部の [シーン] をクリックして、[シーン 1] を選択します)。新しいアクターの名前は、「ボス機」としてください。このアクターをシーンにドラッグする前に、[アクター] タブに表示されているこのアクターの空白部分に、画像ライブラリから「ボス」画像をドラッグします。シーンに画像をドラッグしたら、シーンの表示領域のすぐ外側に 'ボス機' アクターを移動します。これは、シーンが始まってから、表示領域にアクターが飛んでくるようにするためです。「空中戦シーン」は、次のように表示されるはずです ([プレビュー] は使用していません) :

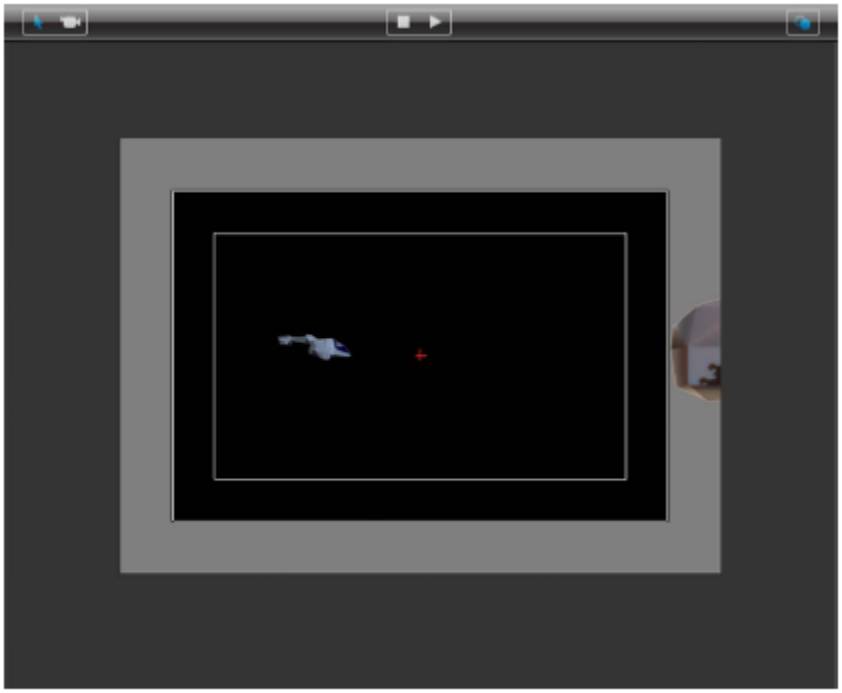

Image 11: The Boss Ship lurks just off-screen

新しく作成した 2 つのアクターのライフを指定するため、「自機のライフ」と「ボス機のライフ」という名前で、新しい整数値のゲーム属性を 2 つ作成し、既定値を 100 に設定します。シーンの編集を使用している今のうちに、'夕日' アクターをシーンにドラッグしておきましょう。これは、背景として使用します。まずいことに、'夕日' アクターが '自機' インスタンスアクターを隠していますが、これは簡単に直すことができます。新しく作成した '夕日' インスタンスアクターをシーン内で右クリックして、[背面に移動] オプションを選択します。

ここからは、既に存在するアクターを再作成するのではなく、これまでに作成済みのアクターを使って効率よく作業を進めます。ただし、ロジックには少し手を加えます。'ミサイル' を開き、ルール条件を調整して、自機にヒットするとアクターが適切に爆発するようにします。これには、[機体] を [自機] に変更します。'レーザー' アクターも同様に変更します。一番下のルールまでスクロールして、[タグの付いたアクター] を [指定した種類のアクター] に変更し、アクターを [ボス機] に設定します。

'自機' アクターと 'ボス機' アクターを機能させるため、次のスクリーンショットのように、もう何点かロジックを設定します。

まず、'自機' について見ていきましょう：

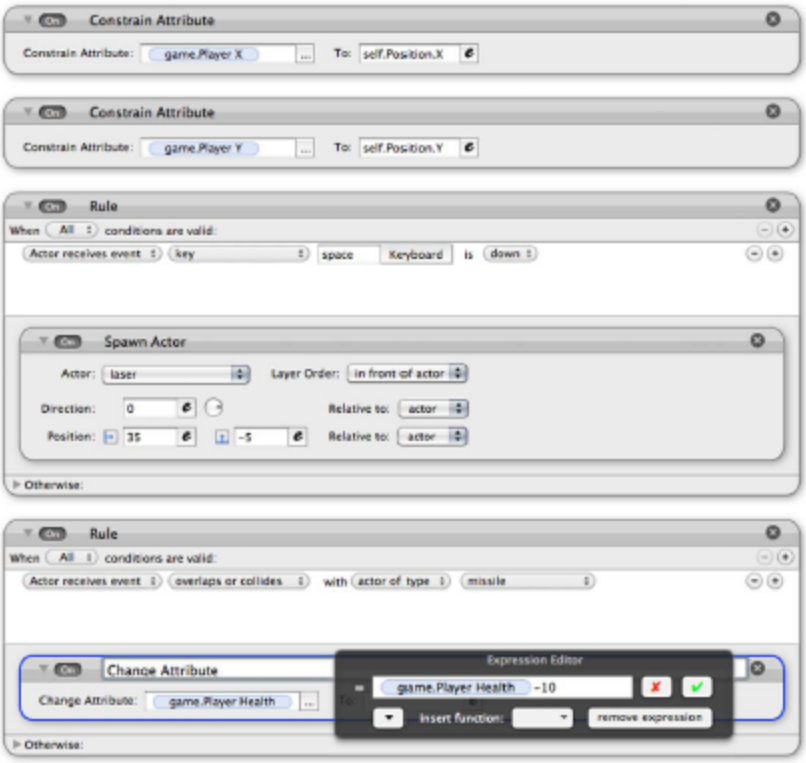

*Image 12: Player Ship **Logic***

簡単に説明すると、2 つの [属性管理] 動作を使用して、'自機' アクターの位置が、2 つのゲーム属性としてコンスタントに報告されるようにします。これらの属性はいずれも「実数」型の属性です。なお、この属性はこのテンプレートの一部として組み込まれているので、作成する必要はありません。これで他のアクターは、これらのゲーム属性を使用して、それに反応できるようになりました。たとえば、ボス機が発射する熱探知ミサイルなどに使用できます。

ヒント: アクターは、自分の自己属性を参照することはできますが、他のアクターの自己属性を参照することはできません。この問題に対処するには、ゲーム属性を作成し、対象となるアクターで [属性の変更] 動作または [属性管理] 動作を使用して、ゲーム属性が更新されるようにします。参照する側のアクターは、この新しく作成されたゲーム属性をチェックして、反応するようにします。

また、スペースバーを押すと 'レーザー' アクターを表示して、自機のレーザーが発生するというルールを作成します。 - シンプルですね!

ここで使いたいアイテムは、数式編集です。このエディターは、“e” アイコンが表示されるどのフィールドでも使用できます。この編集では、数式や関数を使用でき、可能性が大きく広がります。今回は非常に単純な方法で、'自機' にミサイルがヒットするたびに、'自機のライフ' 属性を 10 ずつ減らしていきます。もちろん、自機のライフがなくなったら、自機は破壊されます。

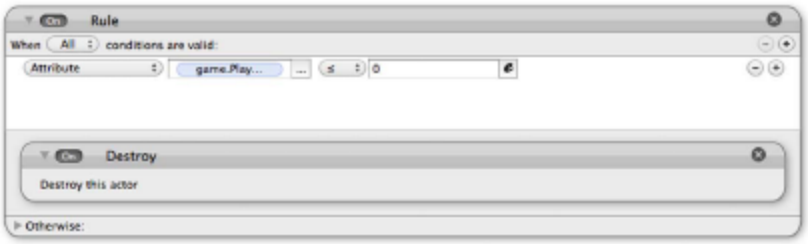

Image 13: Player ship logic for destruction

ここからはボスシップについてです：

私達のゲームでは“ボスシップ”は一秒に一度ミサイルを発射することでプレーヤーの行動から独立して行動します。動き動作はタイマー動作とペアになってやがて終結に至る前に“ボスシップ“がゆっくり場面に登場することを可能にします。

“プレーヤーシップ”がミサイルの衝突によってダメージを受けた場合と同様に“ボスシップ“もレーサーの衝突によってダメージを受けます。これはやがては“大爆発“アクターを発生させた直後に破壊されることにつながります。（アニメ化動作とプレイサウンド動作の両方を持っています）

また、'小爆発' アクターも用意されていて、自機が破壊される直前に [アクターの出現] 動作として、自機のロジックに追加することもできます。

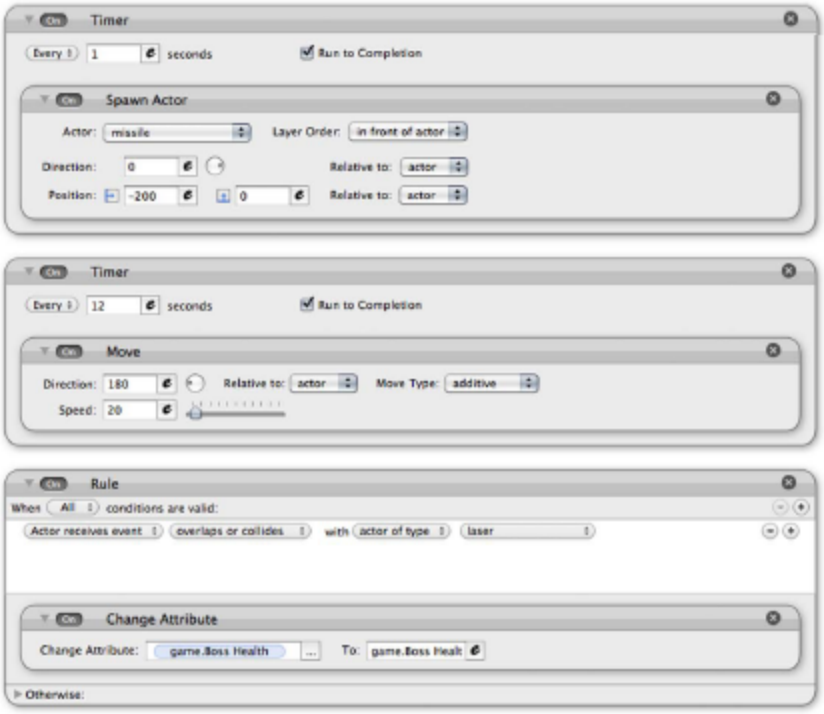

Image 14: Boss Ship Logic

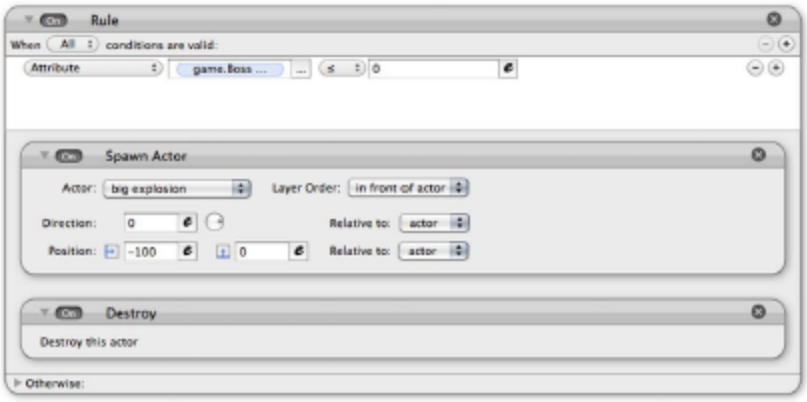

Image 15: Additional Boss Ship Logic

上記はあなたがシーンを作成する際に取ると思われるステップのアイデアを提供しています。ステージ内のプレイボタンをクリックして私達がどのようなものを作ったかご覧ください。ここから、パワーアップ、各種敵のタイプ、代替えとなる爆弾システム、スコアトラッキング、難易度を調整する場面等を追加することによって深さとまた繰り返しプレイを追加することができます。

今あなたは GameSalad クリエーターの原理についてより理解を深めていますね。さあ、あなた自身のアイデアと新企画をいくつか試してみましょう。あなたにどんなアイデアが浮かんでくるか見てみましょう。そしてインスピレーションを得る為に他のテンプレートのチェックも忘れないでください。

ここまで読んでもらって、あらためて感謝です!

ヒント: この開発者向けチュートリアルの残りの部分では、GameSalad Creator の機能に関するより詳細な情報を紹介しています。ゆくゆくはこのドキュメント全体に目を通してもらいたいのですが、最初から最後まで順番に通して読むというのではなく、現在の自分の作業に対応するところだけを拾い読みするような形で読んでもらえればよいと思います。また、ここから先の情報は、cookbook.gamesalad.com の GameSalad Cookbook で最新の内容が入手できることにもご注意ください。

シーン

シーンは、ゲームの構成要素です。シーンは、ゲームのオブジェクトを含んでおり、異なるセクションを整理する為の不可欠な方法を提供します。例えば、あなたはゲーム用の最初のメニュー場面、また個々のレベルの場面、ゲームの終了場面等を作成するでしょう。区分内でゲームをデザインしたり構築したりする為に、シーンを使用します。

ゲームの中でシーンはアクターを入れる入れ物です。各場面をいくつものレイヤーに分けることができます。（Photoshop や Illustrator 等の多くのデザインプログラムと同様）レイヤーはゲームに更に可視出来る構成レベルを提供します。そしてシーン内でオブジェクトをグループにまとめることが可能になり、他のレイヤー/オブジェクトの前面や背面にアレンジすることができます。例えば、あるレイヤーはバックグラウンドを含んでおり、他のレイヤーはあなたのラベル全てを含んでいる。そして、さらに他のレイヤーはプレーヤーが相互作用するアクターの為のものかもしれません。ゲーム内のシーンを表示または編集するには、ナビゲーションメニューの [シーン] ボタンを選択します。

新しいシーンを追加するには、[ホーム] ボタンを押し、[シーン] タブのインターフェイス左下部にある [+] ボタンを押します。既存のシーンをコピーするには、オプションキー (または Alt キー) を押しながら、シーンリストの所定の位置に既存のシーンのコピーをドラッグします。シーンを削除するには、そのシーンをクリックして、Delete キーを押すか、インターフェイス左下部の [-] ボタンをクリックします。

シーン属性

以下の属性は各シーンを定義し、また修正することが可能です。

名前　-　あなたのゲームの様々なシーンを参照する為の分類方法です。

時間 - シーンがアクティブだった秒数。この属性は読み込まれ、ルール/動作に組み込まれる可能性がありますが、設定されるものではありません。

サイズ - 現在のシーンのピクセル寸法 （幅と高さ）。

ラップ x (折り返し) - 有効にすると、シーンの左側から出たアクターはシーンの右側から再び入ります。(その逆もまた同様です。) 無効にすると、アクターは (設定した動作を通じて) そのアクターが破壊されない限り画面外を無限に移動し続けます。

ラップ y (折り返し) - “ラップ x”と類似していますが、この機能はアクターの上下の動きになります。有効にすると、場面の上部から出たアクターは場面の下部から再び入ります。(そして逆もまた同じです。)

重力 - シーン内の重力の大きさ。初期値は 0 です。100 から 1000 の間の値を使用すると、ほぼ 「普通の」重力となります。10,000 を大幅に超える値はどんな値も使用しないで下さい。 重力はX軸方向と Y 軸方向のどちらにも作用します、負の値を入力するとアイテムは反対方向に動きます。重力は、シーン内のすべての移動可能なオブジェクトに影響を与えます。

カラー - シーンの背景色で、赤、緑、青、そして 0 から 1 の間のアルファ値を通じて表現されます。これらの個々の値のいずれかを編集したり、カラーピッカーからカラーを選択することができます。

カメラ - 以下のサブカテゴリーを伴う複合属性：

原点 - シーンに対するカメラの左下隅の x 軸と y 軸の開始位置。

サイズ - ゲームをプレイする際に表示される場面の幅と高さを設定します。これらの値は、プロジェクトエディタで選択された解像度に応じて設定されるので、手動で変更することは出来ません。シーンのサイズを調整するには、「ホーム」 ボタンをクリックして 「プロジェクト インフォ (プロジェクト情報) 」タブを選択し、ドロップダウンメニューの 「プラットフォーム」 を使用します。これらのオプションのいずれかを選択すると、自動的にそのプラットフォームの画面サイズに合わせて、ゲーム内シーンのすべてのサイズを変更します。

追跡エリア -シーン内にいるアクターに「カメラ制御」動作を与え、そのアクターがシーンを進む様子をカメラで確実に追いかけるようにします。この追跡エリアでは (可能な限り) カメラをコントロールする動作を伴うアクターの位置に基づいて、シーンのスクロールを開始する際の境界 (幅と高さ) が設定されます。追跡エリアではカメラが強制的に場面の端を越えて移動するよう設定されていない限り、カメラはこの動作を伴うアクターを逃しません。

回転 - 自動回転に基づいて変化するカメラの回転。これは、手動では変更できません。

自動回転 - プレーヤーが行うデバイス (スマートフォン) の回転に応じて場面が回転します。例えば、プレーヤーが自分のデバイス (スマートフォン) を上下逆さまに回転する場合、そのゲームも縦方向 (もしくは横方向) に上下逆さまに自動回転し、正しく見えるように調整して欲しいとあなたは思うかもしれません。対象的に、迷路もしくはティルトゲームを作成している場合は、プレーヤーが迷路の中のボール (もしくは任意のオブジェクト) をゴールに導こうと装置 (スマートフォン) を傾けるのに応じて場面が回転して欲しいとは思わないでしょう。

属性は、任意の時点で容易かつ迅速に変更されたオブジェクトに関連する値（数値またはテキスト）です。例えば、シーンにおけるカメラの位置決めは「カメラ原点」 属性を変更することによって変更することができ、一方、場面の背景色は 「カラー」 属性で変更可能です。
どのシーンが現在アクティブなのかには、いくつかの重要な動作がゲーム内で影響します。これらの動作の中で、主要なものは「シーンの変更」動作です。

この動作のドロップダウンメニューからゲーム内の任意の場面を選択します。 一般的に、アクターがゴールに到達するもしくは一定のスコアに到達するといった場面の変更のためのある種の前提条件を示しているルールでの動作を設定します。この動作が起動すると、それによりゲームの焦点が選択された場面に変更され、あなたがちょうど居た場面がリセットされます。「シーンの変更」動作を使用して前の場面に戻ることもできます。

ゲームを一時停止する動作は似たように機能しますが、重要な違いが数か所あります。現在の場面をリセットする代わりに、現在の場面の中のすべてを単に一時停止して、選択した場面を開きます。

ゲームを一時停止する動作が選択されている場面で一時停止を解除する動作を使用して元の場面に戻ります。通常、ユーザーはこの動作を (一時停止の解除もしくはゲームの再開) ボタンが押されたもしくは何か類似のことを示すルールで起動させたくなるでしょう。

最後に、「シーンのリセット」 と 「ゲームのリセット」という 動作があります。前者により現在の場面やその場面の属性が単にリセットされ、また、後者によりそのゲームのすべての場面と属性がリセットされます。
カメラ制御を意識することもまた有用です。カメラ画像をクリックすることによりカメラゾーンを調整できます。カメラゾーンは、カメラの動きを引き起こす前にアクターが持つ動きの許容量です。言い換えれば、カメラの動きをコントロールするアクターが居ると想定します。一般的には、アクターが場面中を移動するとカメラが後に続きます。しかし、アクターが振り向いて反対方向に向かう場合、カメラは少ない移動距離でアクターの後に続きたいと考えます。この距離が動きの許容量であり、それはカメラゾーンでコントロールされます。ハンドルを使用してこの許容量を減らしたり増加させたりし、基本的にカメラをコントロールしているアクターはカメラゾーンに突入してカメラがアクターと一緒に移動するまで中央の黒いエリアを自由に動くことができます。場面の編集モードに戻るには、カメラ画像の左側にある矢印をクリックするだけです。

アクター

アクターはプレーヤーがコントロールしているキャラクター、またはあなたのプレーヤーが話しかける、衝突する、飛び越える、もしくはゲームプレイの間に一般的にやりとりする周囲のオブジェクト/キャラクターになることができます。

ゲームデザイナーはしばしば異なるシーンを横断してゲームに必要となるアクターのセットを作成することにより工程に着手します。 これらはプレーヤーがシーン内でやりとりする実際のアクター/キャラクターの「プロトタイプ」(もしくはモデル) として知られています。例えば、パックマンにおいて、あなたは「幽霊」 のプロトタイプを作成し、いくつかの幽霊の視覚属性やその幽霊がゲームでどのように動作するかを指定するかもしれません。各シーン/ゲームのレベルに現れるそれぞれの幽霊はその幽霊のプロトタイプに基づく特定のインスタンス(実体) です (そしてそれゆえそのプロトタイプのすべての属性や動作を継承します) 。その幽霊のプロトタイプに変更を加える場合、そのゲーム内のすべてのロックされた幽霊のインスタンスも変更されます。

インスタンスアクターはシーンという建物のブロック (構成単位) のようなものです。インスタンスアクターを作成するには、場面の中にプロトタイプをドラッグするだけです。その場面内でこの特定のアクターに加えたいかなる変更もそのインスタンスアクターに影響を与えるのみで、プロトタイプ自身には影響は与えません。

一般的属性

名前 – ゲーム内のアクターを呼ぶ際の記述方法

時間 - アクターがシーン内でアクティブだったもしくは「生きて」いた秒数。この属性は読み込まれてルールや動作に組み込まれる可能性がありますが、変更はされません。

位置 - 現在のシーン内のアクターの x 軸と y 軸の位置。この特性は場面のインスタンス アクターに関連していますが、プロトタイプ アクターには関連していません。

サイズ - 現在のアクターのピクセル寸法（幅と高さ）。

回転 – シーン内でアクターが最初に現れる角度と回転を特定します。例えば、その値が 90 の場合、そのときそのアクターは反時計回りに 90 度回転して現れます。

カラー - アクターの背景色で、赤、緑、青、そして 0 から 1 の間のアルファ値を通じて表現されます。これらの個々の値のいずれかを編集したり、カラーピッカーからカラーを選択することができます。

画像 - （もしあれば、） アクター用に表示される画像。これは手動で編集できませんが、ステージ内のアクター インスタンス上のイラスト/スプライトから画像をドラッグすることにより修正することができます。

タグ - そのアクターに与えられたタグ。

図形 - 関連属性

可視化 - アクターがプレーヤーから目で見えるか否かを設定します。チェックが入れられていれば、アクターは見られることになります。この属性はゲームプレイ中には変更できません。そのアクター表示する必要がなくなった場合は、この属性のチェックを外すことにより、ゲームのパフォーマンスが向上します。

ブレンドモード - アクターの図形がその背景に対してどのように描かれるかを決定します。

通常 - 背景の上にアクターを重ねて表示します。 これが初期設定のブレンドモードです。判別がつかない時はこれを使用します。

不透明 - 背景と差し替えて透明度を無視しながらアクターの画素を正確にコピーします。不透明モードは最も手軽なブレンドモードですので、それを特に大きな背景画像用に使用するとあなたのゲームのパフォーマンスを向上させるかもしれません。

加算性 - 画像の各画素の明度をその背後にある画像の明度に追加します。これには強烈な光沢効果があり、一般的に放射光源 (火、火花、爆発、レーザーなど)を描くのに使用されます。

スクリーン - 加算性に類似。 アクターと背景色を加える代わりに、スクリーンはアクターの色を背景色の逆数に追加します。この結果、画像の「露出オーバー」少なく現実的な増光効果をもたらします。

相乗 - 背景とアクターの画素の明度を掛け合わせます。これは背景画像に色合いを付けたり暗くしたりし、グラデーションオーバーレイ、着色ガラスの窓、影効果、もしくはマスキングに役立ちます。透明度と相乗を組み合わせて使用すると奇妙な結果をもたらすこととなり、通常は避ける方が良いでしょう。

水平方向のラップ (折り返し)/垂直方向のラップ (折り返し)、それには 3 つのオプションがあります：

引き伸ばす - 画像は引き伸ばされもしくは圧縮されアクターの境界線に収まります。

固定 - アクターはファイルのサイズ/解像度でその画像を表示します。その画像はアクターの中心に置かれ、アクターは画像の見栄えを変更することなくその画像の解像度を小さくしたり大きくしたりすることができます。

タイル - アクターがその解像度よりも大きい場合、その画像は繰り返されます。

運動- 関連属性

直線速度 - X 軸方向や Y 軸方向で指定した特定の方向におけるアクターのスピードを設定します。

角速度 - アクターが回転する速度。 0 より大きい値は反時計回りの回転を引き起こします。負の値は時計回りの回転を引き起こします。

最高速度 - 「最高速度」が選択された場合にアクターが出せる最高速度。

最高速度の設定 - チェックを入れると、そのアクターは最高速度に設定されたスピードしか出せなくなります。そうでなければ、そのアクターはスピードを加速し続けることになります。

物理的-関連属性 (非運動)

密度 - アクターの重さ。値が大きいほど、そのオブジェクトはより密度の低くいアクターによって動かすことが難しくなります。0 値は、アクターは静止状態にはありますが、他のアクターへ影響を及ぼすことは可能です。密度は任意の正の実数に設定できます。(つまり、密度 >= 0)

摩擦 - アクターが別のアクターに接触する度にどれだけ遅くなるかを特定します。 0 は滑らかであり、数字が大きくなるほどアクターは速度がすぐに遅くなります。 摩擦は任意の生の実数に設定できます。(つまり、密度 >= 0)

跳ね返り - アクターの弾性を表す数。0 は弾性無しです。跳ね返りは 0 と 2 の間の任意の実数に設定できます。

回転の固定 - 場面でアクターが他のアクターと衝突する場合にそのアクターが回転するか否かを指定します。

可動性 - これが他のアクターとの衝突や重力を含め、そのアクターが動かせるかどうかを決定します。

衝突形状 - アクターは長方形もしくは円形の衝突領域持つことができます。円形衝突領域の場合、アクターの寸法に完全に収まる最大の円形です。

抵抗 - これがアクターに直線抵抗を与えます。アクターの運動を修正する他の力 (運動動作) が何も無い場合、それによりアクターの動きは徐々に遅くなります。

回転に対する抵抗 - アクターの運動を修正する他の力 (回転動作) が何も無い場合、それによりアクターの回転は徐々に遅くなります。

新しいアクターの属性は、[属性] パネルの左下部にある [+] または [-] ボタンを使用して、作成または削除できます。新しく作成した属性は、属性リストの下部付近にハイライト表示されます。これらの属性には、ルールエディターからアクセスできます。特に注記されていない限り、[属性の変更] 動作を使用して変更できます。

前述したように、プロトタイプアクターを変更すると、このアクターに対応するすべてのロックされたインスタンスに影響を与えます。 アクター (つまり、シーンに配置されているアクター) をダブルクリックすることによって特定のインスタンスアクターをカスタマイズすることができます。あなたがインスタンスアクターを選択すると、あなたが誤ってそのインスタンスアクターのプロトタイプとは別のインスタンスアクターのロジックを変更してしまわないように、まずルール 編集が最初にロックされます。しかしながら、そのロックアイコンをクリックするだけでそのアクターのロックを解除して編集可能にすることができます。ルール 編集の上部にあるプロトタイプに戻るボタンを押してプロトタイプのルールに戻ることができます。あなたがインスタンスアクターのロックを解除すると、そのプロトタイプアクターにどんな変更を加えてもその特定のインスタンスには影響を与えません。インスタンスアクターの属性は、[属性]パネルのロックを解除せずに編集できます。各属性を元に戻すには、属性を選択してから、[属性] パネルの上部にある [プロトタイプに戻す] ボタンを押します。

アクターはさまざまな方法で追加できます。まず、シーンの中にアクターを配置することができます。これは単純に、シーン編集で所定のシーンを読み込み、アクターを好みの位置にドラッグするだけです。次に、アクターを「シーンの外」に配置して、ゲームプレイ中の任意の時点で、シーンに登場させることができます。シーンのサイズ (シーン編集でシーンを表示し、「シーン」を切り替えて [属性] タブを参照) が、カメラサイズ (選択したプラットフォームによって決定されます) より大きい場合、シーン編集には、カメラの範囲外の領域が表示されます。
ここにアクターを配置し、さまざまな動きの動作を使用してシーン内に移動させます。具体的な方法については、このドキュメントの後半で紹介します。最後に、[アクターの配置] 動作を使用して、アクターを出現させることができます。この動作を使用すると、所定の場所と速度で、アクターをシーンに配置することができます。

動作

動作はアクター達にどうやってお互いやりとりするかを割り当て、そして姿を変化させることができる行動です。動作はゲームにロジックを加えて (アクターがオブジェクトに衝突する時のような) いくつかのイベントが発生する際には何が起こるかをコントロールし、アクターに (そのアクターの速度を変化させるというような) 何かの行動を取らせたり、もしくはサイズ、画像、色、もしくは透明性といったようなアクターの姿を変化させるために使用されます。

動作には、3 つの種類があります。

コンテナ - ある動作を、特定の条件でのみ発生させる場合に使用します。
たとえば、特定のアクターをクリックした場合にプロンプトを表示させるといったルールを指定できます。これは、動作名の横に表示される [G] で示されます。コンテナの別の形態として、**タイマー**があります。タイマー内に動作を配置すると、その動作を発生させる頻度や待ち時間を制御できます。

保存 - 継続的にアクターに働きかけて(その条件が真ではないルールに配置されない限り) 行動させ続ける動作。これらは動作名の隣の[B]により表示されます。

動作 - 一度発生し、その条件が偽になり、そして再度真になるルールが配置される場合にのみ何度も現れる。これらは動作名の隣の[A]により表示されます。

36 の標準動作がありますので、次のリストを参照してください。これらの動作は、その他の用語と同様に用語説明に掲載されています：cookbook.gamesalad.com/definitions

加速 - アクターの速度と向きが設定できます。抵抗の属性または重力がアクターに適用されていない場合は、アクターは設定最大限まで加速し続けます。「加速方向」を参照してください。

加速方向 - アクターの加速度とターゲット位置が設定できます。編集機能を使うと、静止または動いているターゲットの位置が設定できます。抵抗の属性または重力がアクターに適用されていない場合は、アクターは設定最大限まで加速し続けますその他の動作または属性を使ってアクターの速度を落とさなければ、アクターは軌道に沿って加速し続け、ターゲット位置を通過します。「加速」動作を参照してください。

アニメーション -プロジェクトライブラリーから一連のイメージをこの動作にドラッグ&ドロップします。追加後は、イメージを録画したり追加のイメージを挿入できます。この動作のコントロールでは、アニメーションのフレームレートが設定できます。最終フレームでアニメーションがループまたは停止、またはアニメーションが開始する直前のイメージに戻ったりする場合でも、フレームレートは 30 フレーム/秒まで設定できます。

属性の変更 - ゲーム、シーンまたはアクター属性を設定、変更、インクリメントができます。属性によって定義された数値、色、サイズ、動作、加速度やその他の値が変更できます。

イメージの変更 - アクターに使用中のイメージを、他のイメージと交換する事ができます。新しいイメージをこの動作にドラッグするか、ドロップダウンメニューから新しいイメージを設定してください。

シーンの変更 - この動作では、使用中のシーンを停止して即座に設定されたシーンに移動します。特定の条件に見合ったゲームのクレジットシーン、メニューシーン、または新しいゲームレベルにこの動作を入れると効果的です。

サイズの変更 - アクターの縮尺を変更します。 (縮小するにはマイナス数値を使用します。) 注意: アクターの元のサイズは、衝突ボリュームによって定義されます。元のサイズを変更するには、「属性の変更」または 「改値」 を使用してください。サイズの変更にかける時間はタイマーで設定してください。

速度の変更 - アクターの動作の向きを一定の速度に設定できます。向きを設定すると、ドラッグ、重力またはその他の動作などの影響がアクターに適用され始めます。

衝突 - アクターまたはアクターグループの衝突を設定します。アクターグループはプロジェクト編集のタグから作成できます。

属性管理 – 属性間の値を継続的に更新しながら合わせます。2 つのオブジェクトの動作をシンクさせたり、アクターの動作をマウスやタッチの動作と連携させたりできます。

カメラ制御 - アクターに適用するとシーンのカメラがスクロールしてフォローします。カメラの追跡範囲はステージのカメラ編集モードで変更できます。 1 つのシーンにつき 1 インスタンスアクターにのみこの動作を使用できます。スクロールできないレイヤーのアクターはこのカメラ制御機能を使用できません。

削除 - アクターをシーンから即座に削除します。

テキストの表示 - ボックスに入力されたテキストを表示します。色、配置、フォント、折り返しやテキストのサイズなどのコントロールも設定できます。折り返しはアクター内に文字列をすべて表示する際に使用します。

グループ - 動作やルールを簡単にグループにまとめられます。「グループの作成」 ボタンを選択してグループを作成してください。

改値 - 設定された時間内で属性の既存値を新規の値に変更します。改値では設定された時間内に継続的な変更レートを使用します。ゲーム属性の急な変更または緩やかな変更に影響し、その他の動作によって遮断または停止することはありません。

属性読込み - 保存属性動作を使用してカスタムキーネームで保存された値を起動します。キーは特定の属性の保存場所を示しますお好きなキーネームで保存し、情報を起動させる場合は同じキーを使用してください。

移動 - アクターまたはシーンを設定速度で特定の向きに動かします。累積動作では、アクターに設定された複数の動作がスタックするかまたは同時に適用し始めます。スタックされた動作は最新の動作のみがアクターの動きをコントロールします。

移動先 - アクターまたはシーンに連携した XY 座標値まで移動します。コントロールしている条件が無効な場合や 「最後まで実行」 が選択されていない場合を除き、この動作は設定された座標で停止します。コントロールしている条件が無効な場合は、この動作はただちに停止します。

注意: この機能で、ルール、動作、グループ、またはその他のゲーム要因についてのリファレンスを記録することができます。これらは、ゲーム中に表示やまたは操作に影響を与えることはありません。

パーティクル - アクターに設定されたパーティクル数を生成します。色、サイズ、寿命、速度、イメージなどのオプションが含まれます。

ゲームの一時停止 - 既存のシーンを一時的に停止し、動作で選択されたシーンを既存のシーン上に表示します。一時停止を解除すると、元のシーンに戻ります。

BGM 一時停止 - 音楽が再生されている場合は、トラックを一時的に停止します。「BGM 再生」を使用してトラックを再生します。

BGM 再生 - 選択された音楽ファイルの再生を開始します。「ループ」 を選択すると、選択された音楽がすべて再生し終わった後にもう一度再生します。

サウンドの再生 - 選択されたサウンドファイルの再生を開始します。「ループ」 を選択すると、選択されたサウンドファイルが再生し終わった後にもう一度再生します。「最後まで実行」 が選択されている場合は、サウンドが再生し終わるまでその他の動作は遮断されます。「ポジショナルサウンド」 と 「速度変更」は、サウンド効果をコントロールするアクターがシーンを移動するに当たり、サウンドの音量とピッチを調節します。

複製 – 属性値を基に、アクターを複製します。シーンに追加するアクターを新規に生成する作業を省きます。残りのライフ数を表示する際によく使用されます。

ゲームのリセット - すべてのシーンを含むゲームをリセットします。これによりすべての属性値が元の状態に戻ります。「属性保存」を使用して保存されたキーは削除されません。

シーンのリセット - 既存のシーンとアクションをリセットします。一時停止中に表示されたシーンでリセットされた場合、そのシーンはリセットされません。

回転 - エクスプレッションエディターで設定された速度でアクターが時計回りまたは時計と反対周りに回転します「回転角度」 と 「回転位置」の動作は似ていますがまったく異なるタスクです。

回転角度 - 設定された角度までアクターが時計回りまたは時計と反対周りに回転して止まります。「設定値で停止」 が選択されていない場合は、「回転」同様の動作をします。

回転方向 - 設定された XY 座標値までアクターが時計周りまたは時計と反対周りに回転して止まります。「設定値で停止」 が選択されていない場合は、「回転」 動作と同様の動作をします。「角度のオフセット」で特定の XY 座標値からの角度を設定します。

ルール - 条件や条件グループを作成して設定された動作を実行前に確認します。条件にはプレーヤーのインプット (マウスのクリック、タッチ、キー) や属性値も含まれます。ルールには 「その他手段」 設定も含まれます。ルールの条件が真 (True) を返さない場合に発生する動作もここに含まれます。

属性保存 - カスタムキーネームの属性値を保存します。お好きなキーネームを使用してください。「属性読込み」にキーネームを入力すると保存値を引き出します。以前に使用されたキーネームで新規の属性値を保存すると、既存データが上書きされてしまいます。ゲームまたは端末のスイッチが切られていても 「属性保存」 で保存された値はアクセス可能です。

アクターの配置 - シーンのアクターインスタンスを作成します。アクターまたはシーンに連携した向きや位置を設定します。同じシーン内のどこででもアクターを追加して生成することができます。新規に生成されたアクターは連携した動作を即座に開始します。

BGM の停止 - 既存の音楽トラックを停止します。「BGM 一時停止」とは異なり、この動作ではトラックがリセットされ、この後に 「BGM 再生」が実行された際は、トラックの頭から音楽を再生します。

タイマー - 動作やルールの間隔を設定します。すべてのタイマー値は秒単位です。「オンタイマー」 は設定された時間の後に動作やルールが実行されます。「頻度」 は設定された時間の間隔ごとに動作が繰り返し実行されます。「期間」 は設定された時間内で動作が実行されます。

ゲームの一時停止解除 - 「ゲームの一時停止」が実行されて一時停止シーンが表示された場合、一時停止シーンを削除して元のシーンに戻します。

属性

GameSalad を利用する上で、属性は重要なアスペクトとなります。属性は、値の保存先または変数です。さまざまなシーンで使用される数値やテキスト値を保存します。大砲の発射角度やキャラクター名の表示、そしてゲームの得点を記録したりと、どんな目的にも簡単に使用できます。属性を正しく使用することで、GameSalad をより効果的にご利用いただけます。

値を記録する場面では (角度やテキストを含め) 属性を作成してください。このコンセプトが難しいとお考えの場合は、まずは数値の属性から始めてみてください。例えば、レーシングゲームを作成している場合。車の速度の属性を作成すると、加速、ブレーキ、衝突などいろいろな場面での速度変更が簡単に実行できます。これと同様に、トラックを一周する時間や得点などの属性も作成することができます。

GameSalad では、膨大な数の属性、アクター、シーン、ゲーム情報を提供しています。シーンサイズ、アクターの動作、ゲーム時間などもこれらの事前設定された属性でコントロールされています。デベロッパー向けとして、カスタ

ム属性の作成オプションも提供しています。まずは既存の属性について説明します。カスタムア属性の使用方法については追って説明します。

属性は、ゲーム、シーン、そしてアクターに使用します。ゲーム属性には、すべてのプロトタイプアクターやインスタンスアクターでもアクセスでき、どのシーンでも閲覧できるようになっています。シーン属性には、特定のシーンでのみアクセスできますが、そのシーンのすべてのインスタンスアクターは、これらのシーン属性を個々に閲覧、使用、編集することができます。アクター属性は、インスタンスアクターのみアクセスできますが、どのプロトタイプアクターでも定義そして使用することができ、プロトタイプのすべてのインスタンスに事前に定義付けされた動作を持つことができます。

属性には 6 種類あります：

ブール値 - 真または偽 (True/False) の値です。ドアやボタンの場合、「真」 の値を開く/オン、「偽」 の値を閉じる/オフに設定して使用します。すべてのアクターは、「物理的動作」 設定で少なくとも 1 つのブール値属性によりアクターが動作するかしないかをデフォルト設定されています。

テキスト - 文字や数字の値。「テキスト表示」 と一緒に使用して、シーンのタイトルやアクターの名前などを表示します。

整数 - 15、0、42、5801 など、すべての自然数。整数属性は最も役立つ属性の 1 つです。ゲームの得点、生成単位の設定、シーンにある得点オブジェクトの数などを記録する際に使用します。

実数 - 3.582、0.882、-2.5 などの小数点値が使用できます。整数も使用できます。実数属性は、タイマーのカウントや、流体の流出量、ヒット数やダメージ数など端数を使用する際に用います。

角度 - 0 から 359 度までの値が設定できます。この角度内での少数点値も使用できます。「大砲物理」 テンプレートにあるような大砲の角度など、回転したり角度が必要なオブジェクトに使用します。

インデックス - 0、1、58、2804 などの正の数値に使用します。インデックス属性は整数属性と共通の作用を多くありますが、負の数値を保存できません。例えば、4 から 6 を引いた値は、−2 ではなく 0 を返します。これらの属性は、GameSalad テーブルや配列など高度な機能を使用する際に非常に役立ちます。

ゲーム内の既存の属性は、シーン編集 (インスペクターでは、[ゲーム] または [シーン] タブを選択し、下部の [属性] タブ)、またはアクター編集で表示できます (左側にリストが表示されます)。これらの既存の属性には、ゲーム、シーン、アクターに関するデータ (サイズ、場所、色など) が記録されます。一方、カスタム属性には、そのゲーム固有のデータが含まれます。たとえば、レースカーの速度や、パズル内にブロックがどれだけ残っているか、などです。属性リストの下部にある [+] ボタンをクリックし、必要に応じて名前を変更します。

この属性の値にアクター動作からアクセスするので、覚え易い名前に書き換えると次回から簡単にアクセスできます。例: 「レースカー速度」 や 「残りピース数」 など。アクターの位置を記録したい場合は属性から実行でします。その場合は属性の名前を 「プレーヤーX 」 や 「プレーヤーY」 にしてプレーヤーの XY の座標値を記録します。

作成した属性を参照するには、任意の動作で数式編集(“e”) または属性ブラウザー (“...”) を使用します。

画像とサウンド

まず、画像を使って、簡単にアクターを作成する方法を見ていきましょう。GameSalad クリエーターを開き、プロジェクトを開きます。次に、最上部の選択バーから [アクター] を選択します。特定の画像を使ってアクターを作成する場合は、その画像をこのボックスにドラッグします。アートワークをプロジェクトに追加するのに必要な作業はこれだけです。その画像を使ったアクターが自動的に作成されます!

また、プロジェクトの画像ライブラリに画像をインポートすることもできます。後からアクターに画像を割り当てる場合や、単にゲームの背景として画像を使う場合に便利です。画像ライブラリに画像を追加するには、最上部の選択バーで [シーン] を選択します。シーンを開くには、ダブルクリックします。シーン編集の左下部にライブラリが表示されます。プロジェクト内ですべての画像を表示するには、[画像] を選択します。プロジェクトに新しい画像を

追加するには、その画像をこのセクションにドラッグします。ここから、シーンやアニメーションシークエンスに画像をドラッグしたり、アクター編集の [イメージの変更] 動作からアクセスしたりできます。また、アクターボックスに画像をドラッグして、その画像を使った新しいアクターを作成することもできます。

必要に応じて、画像ライブラリの [+] 記号を選択し、画像を指定して [開く] を選択すると、自分の画像をプロジェクトに追加できます。.png 画像ファイルなら、どのような画像でも GameSalad にインポートできます。

原則として、iOS では自動的に、次の組み合わせのピクセル比に最も近いサイズに画像が拡大されます。

16x16
32x32
64x64
128x128
256x256
512x512
1028x1028

たとえば、画像が 50x30 ピクセルの場合、64x64 の画像と同じメモリが使用されます。サイズが 1028 ピクセルより大きな画像は、自動的に 1028 ピクセルに縮小されます。つまり、130x20 ピクセルの画像の場合、いずれかのピクセル数を 2 ピクセル減らさないと、256x256 の画像と同じメモリが使用されることになります。

サウンドも画像と同じように追加できます! ライブラリで [サウンド] を選択し、ファイルをドラッグするか、[+] 記号を選択して自分のサウンドファイルを指定し、アップロードします。

サウンドのボリュームは、ゲーム内で簡単に変更できます。[サウンドの再生] 動作で、作成した実数値の属性にボリュームを設定します。その後、[属性の変更] 動作を使用して、新しい属性を 0〜1 の間の任意の値に変更します。

テーブル

テーブルの作成

プロジェクト編集のページを表示します ([ホーム] ボタンをクリックします)。
“テーブル”タブをクリックします。
画面の左下部にある [+] 記号をクリックします。

テーブルの編集

作成したテーブルをダブルクリックします。[行] と [列] という表示の右側にある編集可能なボックスに、数値が表示されています。ボックスをクリックして値を変更し、必要な行数と列数を入力します。列の型を変更するには、ドロップダウンメニューの矢印をクリックして、設定する型をクリックします。あるデータ型を後から変更する場合、GameSalad は、その列のすべてのセルに対し、新しい属性のデータ型への変換を試行します。セルに情報を追加するには、列/属性の型が選択されていること、および利用する行数が設定されていることを確認してください。後はセルをクリックして、入力を開始するだけです。

注意: 行数と列数の最大値を超えた位置にある行や列に入力することはできません。

.csv ファイルから GameSalad にインポートするには、情報のインポート先になる表を指定し、その表を開きます。右上角にある [CSV のインポート] ボタンをクリックし、インポートする CSV を選択して、[開く] をクリックします。

インポートしたいデータのタイプを選択し、[インポート] をクリックしてください。

テーブルでデータを使用する

データを参照したいアクターを表示し、それに 表現の編集 を利用する動作やルールを与えてください。 表現の編集を開くには [e] をクリックしてください。[関数を挿入する] ドロップダウン メニューをクリックしてください。

関数 [TableCellValue] をクリックすると、関数 tableCellValue(table,row,col) 内に情報が入力できるようになります。ゲーム属性ブラウザ に移動して、どのテーブルを最初の [テーブル] 座標で使用するかを指定してください。行と列の座標を指定してください。それらは追加関数や属性参照を使うことで修正またはランダムに変更できることを覚えておいてください。

特別なテーブル関数

テーブルの情報を集めることのできる 2 つの特別な関数があります。“TableCol-Count” は指定した [テーブル] フィールドの最大列数を取得するために使われます。“TableRow-Cout”は指定した[テーブル]フィールドの最大行数を取得するために使われます。

これらの 2 つの関数はテーブルからランダムにデータを抜き出すのに有用で、値の範囲を高度にカスタマイズ可能にします。

GameSalad での公開

この文書は、公開の手順よりも、GameSalad クリエーターそのものについて説明することがねらいです。ここでは、公開する段階になった際の参考に、公開に必要なものを簡単な一覧にまとめました。詳細なチュートリアルが必要な場合は、GameSalad Web サイトを参照してください

> ヒント: スケジュールされた期日よりも前に、プロジェクトの公開をテストすることは重要です。 これはあなたのプロジェクトをアドホック テストする機会と、遭遇するかもしれない公開を阻害するあらゆる問題に対処するための時間をあなたに与えます。

Apple 向けの公開の為のクリエーター要求事項

ブロードバンドインターネット接続
512x512 PNG アイコンイメージ
Xcode 4.2
有効なプロビジョニングプロファイルとキーチェーン証明書
適切な解像度の、1 つ以上の PNG 形式スクリーンショット画像 (Creator でプレビュー中に [スクリーンショット] オプションで取得可能)
PNG 形式のスプラッシュ画面 (オプション)

Android 向けの公開の為のクリエーター要求事項

ブロードバンドインターネット接続
有償版(Pro)メンバーシップ
512x512 PNG アイコンイメージ
有効なキーストア
キー署名ツール (Keytool、Jarsigner、Zipalign) のインストール
適切な解像度の、1 つ以上の PNG 形式スクリーンショット画像 (Creator でプレビュー中に [スクリーンショット] オプションで取得可能)
PNG 形式のスプラッシュ画面 (オプション)

ヒントとベストプラクティス

- 早めにそして頻繁に保存を行ってください! さらに、必ず何らかのバージョン管理 (あなたのプロジェクトの複数のファイルを保存するもの) を使用し、そしてプロジェクトのコピーを別々の記憶装置 (例えばフラッシュ ドライブや 2 台目のコンピューター) に保存するようにしてください。これは、ハードウェアの故障、データの破損、またはあなたのプロジェクトが意図せず変更されること対しての備えになります。もし最悪の事態が起きてあなたのプロジェクトがもう開けなくなってしまったときでも、過去のある時点で公開に成功していたら、あなたは GameSalad サーバーからのそのバージョンを取り戻すことができます。 この回復プロセスについては、このマニュアルの FAQ セクションに詳細な記載がありますのでご確認下さい。

- メモと詳細なドキュメンテーションを保管しておいてください! しばしば見落とされがちなことですが、これは特にあなたのプロジェクトが大量の複雑な依存関係を持ったアクターと属性を扱っているなら、尚更大切な事です。

イライラしないでください! 誰でもときどき壁にあたります。容易な解決方法があるようには見えない問題に直面したら、複数の角度からそれに取り組んでください。

FAQ (よくある質問)

Q: GameSalad を使用するのに Apple デベロッパ証明書は必要ですか?
A: GameSalad クリエーター自体を使用するのにデベロッパ証明書は必要ありませんが、プロジェクトを公開したり Apple デバイスでテストすることが必要な場合、および Apple App Store にアプリを公開する場合は、証明書が必要です。

Q: Apple デベロッパ証明書とプロビジョニングプロファイルは、どのように設定するのですか?
A: 残念なことに、この手順は複雑で、Apple の管理下にある事柄なうえ、それぞれのユーザー契約書で規定される内容であることから、Apple の公式文書に示された手順に従って設定することをお勧めします。Apple の公式文書は、https://developer.apple.com にあります。

Q: GameSalad クリエーター を使って作成した自分のプロジェクトの権利を、私は所有していますか?
A:はい!利用規約によって定義されるあなたのすべての権利と義務を、精読してください。GameSalad のウェブサイト、クリエーター、またはそれと連権したサービスを閲覧・利用することによって、あなたは gamesalad. com/terms で述べられている利用規約に同意するものとします。

Q: GameSalad 有償版(pro)メンバーシップを購読する利点は何ですか?
A: プロの開発者はプロジェクトをマネタイズするための革新的なメソッドや、さらに進歩的な動作 (例えば ゲームセンターリーダーボードと Open URL) を利用する権利があり、Android 向けに公開でき、そしてカスタマー サービスとのやりとりでもより早い返事が期待できます。これらの利点の完全な詳細は、gamesalad.com/menbership/pricing にあります。

Q: Pro のメンバーになってから、Pro 動作にはどのようにアクセスするのですか?
A: Pro メンバーシップにサインアップしたら、いったんログアウトして、再度 GameSalad クリエーターにログインし直してください。Pro 動作は、アクター編集の左下角にある [Pro] タブ([標準] タブと [カスタム] タブの右側) から利用できます。このオプションが表示されない場合は、数時間待ってから、やり直してください。それでもまだ Pro 動作を利用できない場合は、arcade.gamesalad.com/feedback から GameSalad カスタマサポートにサポートを要請してください。なお、e-check で支払いを行った場合、決済に時間がかかり、アクセスが許可されるまで数日間かかる場合があることにご注意ください。
Q: 自分のプロジェクトを クリエーター プレビューモードで実行すると他のデバイス上で実行したときと違うのはなぜですか?

A: 各々の環境は、ユニークなハードウェア構成とソフトウェア構成を持ちます(PC、Mac、iOS、または Android デバイス)。それはあなたのプロジェクトがどのように機能するかに影響を及ぼします。プロジェクトを、公開するつもりのすべてのデバイスでアドホックテストするのをお勧めします。これが最も正確なテストの形式だからです。

Q: 自分のコードをプロジェクトに追加して、まだ利用できない機能を追加したり、機能をカスタマイズしたりできますか?
A: GameSalad の基本方針は「みんなの為のゲーム開発」です。そのため GameSalad 内部では、ユーザーがコーディングせず利用できる機能を開発することに力を注いでいます。そのうえで、GameSalad はユーザーからのフィードバック、アイディア、機能のリクエストに、常に耳を傾けています。ユーザーからのコメントは、スケジューリングに大きく反映されています。ご意見をお聞かせくださる際には、forums.gamesalad.com/categories/new-features をご利用ください。GameSalad ソフトウェアで許可されている利用方法の詳細については、gamesalad.com の利用規約 (Terms of Service) を参照してください。

Q: 以前に公開したプロジェクトファイルを復旧するにはどうすればよいですか?
A: GameSalad Launcher を開き、左下部にある [ポートフォリオ] タブを選択します。Launcher と公開サーバーとの通信が開始され、以前に公開された内容のリストが表示されます。(この通信プロセスが完了するには、この画面を表示した状態で、最大 15 分かかる場合があります。ご了承ください)。復旧したいプロジェクトを選択して、[GameSalad クリエーターで編集] をクリックします。

トラブルシューティング

問題: 数式編集で、[属性の保存/読み込み] 動作や [アプリ内購入] 動作など、動作の一部に問題が発生します。
考えられる解決法: 既定のテキストフィールドと数式編集では、必要とされる形式が異なっています。数式編集は、この編集による追加機能が必要な場合にのみ使用することをお勧めします。この編集は、ユーザーが文字列を入力することを想定しておらず、文字列によっては引用符の使用が必要になります。

問題: ルールのトリガーに一度は成功しますが、以降はトリガーされません。
可能性のある解決方法: このような状況では、ルールをおとり戦術として考えることが役立ちます。一度トリガーされたら、それらは クリエーターによって再び準備をするように「指示される」必要があります。これは、ルールの状態がもはや有効でなくなるとすぐに起こります。例えば、整数の属性値が 1、3 または 5 であるときにトリガーされるルールなら、その属性値が 2 または 4 のとき、ルールは条件がもう有効でないと気付き、再びトリガーするための準備を自ら行うでしょう。しかし、属性値の 1 から 3 への (2 をスキップした) 直接のジャンプなら、ルールの知る限り、それはすでに意図されたようにトリガーが成功しており、再び準備がなされるまでそれ以上の動作はしないでしょう。このような状況に遭遇しないようにルールの条件を調整するか、またはいったんルールがトリガーされたら、それ自身が自動的に準備をするようなロジックを作成することをお勧めします。

問題: プロジェクトを公開しようとすると、エラーが表示されます。
考えられる解決法: 次のいずれかの問題が原因である可能性があります。公開元のコンピューターの構成が不適切、ファイアウォール/ウイルス対策ソフトウェアによる干渉、公開サーバーの混雑。GameSalad カスタマーサポート (arcade.gamesalad.com/feedback) にお問い合わせください。その際、公開使用した GameSalad クリエーターのバージョン、エラーに表示されるテキスト、公開の対象となるプラットフォームなど、詳細をお知らせください

問題: 自分のプロジェクトのパフォーマンスを向上できそうな方法は何かありますか?
可能性のある解決方法: パフォーマンスは、使用するデバイスのハードウェア構成、プロジェクトと一緒に動作している他のすべてのソフトウェア、プロジェクトのサイズ、およびプロジェクトのロジック構成の影響を受けます。ユーザーにとって、可能な限りスムーズに動作し、そしてさらに最適化の進んだ、裏で安定動作するゲームを開発できる環境かは大変重要な事柄です。さらに、以下の様な対策により、可能なかぎりロジックを合理化することをお勧めします：

- 同時にアクティブになっているアクターの数を減らす
- タイマーと制約属性動作の使用を少なくする
- 他のロジックと意図せずに矛盾しているかもしれないロジックがないか確認する

更なる支援

もっと様々な情報を知りたい方は、GameSalad の公式ウェブサイトに、ユーザーのためのリソースを継続的に追加しています。クックブック(cookbook.gamesalad.com) のアップデートはもちろん、フォーラムの Salad Solution セクション (forums.gamesalad. com/categories/Salad-Solutions) にも新しい情報が随時投稿されますのでご確認下さい。

技術的な支援が必要な際は、arcade.gamesalad.com/feedback のコンタクトフォームを使って、GameSalad のカスタマーサービスチームにお気軽にご連絡ください。

用語説明

加速度計–デバイスを動かした時の方向や程度を判断します。ユーザーがデバイスを傾けたり回転させた時の測定に便利です。

加速（動作）– S アクターの加速スピードと方向を特定します。ドラッグするか、優先された他の動きが無い限り、アクターは加速し続けます。

加速方向（動作）–アクターが加速する特定の位置を指定します。

動作–動きや画像、色やサイズ、キーボード入力など、常に起こるもののことは該当しません。ある規則によって定められている場合を指します。

アクター–ゲーム上にある全てのモノは（見える、見えないを問わず）アクターと呼びます。それぞれのアクターは、他のアクターや、ゲームのプレーヤーにどう影響するかを定める「動作」の通りに作用します。

アクターモード - シーン編集の既定のモードです。シーン内で、アクターの配置、移動、回転、サイズ変更を行うことができます。

アクタータグ–アクターを種類分けするのに使用します。プロジェクト編集者はアクタータグを追加したり削除したりすることができます。
アルファカラー–アクターの透過性を調整します。ゼロから一までの数値に定める事ができ、ゼロは完全に透明あることを、一は全く透けて見えないことを表します。

アニメーション（動作）–一連の画像を高速で続けて表示することにより、アニメーションを作成します。

属性–ゲームや、画像、アクターの様々な要素を定める数値や文字列のデータです。

abs（関数）–ある数値の絶対値を示します。つまり、負の数は正の数表記となります。例: abs(-5.23)=5.23

acos（関数）–アークコサイン（逆コサイン）関数。入力値 X はー 1 から 1 までの値をとり、算出される値は、それぞれ 180 から 0 までとなります。それ以外の値が入力されると、無効という計算結果となります。

asin(関数) –アークサイン（逆サイン）関数。入力値 X は-1 から 1 までの値をとり、算出される値はそれぞれ－90 から 90 までです。それ以外の値が入力されると、無効という計算結果となります。

atan（関数） –アークタンジェント（逆タンジェント）関数。計算結果はー90 から 90 までの値をとります。
動作ライブラリ–アクターに適用することができる動作のリストです。

動作–アクターに適用される動作や状態のことです。アクターの動き、表示、音声、行動を変えることが可能です。

カメラモード–画面編集中に選択できるモードの一つで、カメラの動きの幅や感度を設定することができます。

属性の変更（動作）–ゲームやシーン、アクターの属性を設定、変更します。例えば、スコアが記録されるようにしたり、特定のアクターに対しポイントを加える（あるいは減らす）など。また、プレーヤーやアクターのライフを減らすことも可能です。

イメージの変更（動作）–アクターの画像を変更します。ダメージや、その他の変化を表すの使われ、複数のアクターを作らなくてすむ為、便利です。

シーンの変更（動作）–特定のシーンへ進みます。次のレベルや、クレジット画面、メニュー画面に遷移するのに使われます。

サイズの変更（動作）–アクターを拡大、縮小させます。（縮小させるにはマイナスの値を入力）タイマーコンテナを挿入する事により、どのくらいの時間で拡大、縮小するかの調整が可能になります。

速度の変更（動作）–他のアクターやシーンとの関係で、動きが変化するようにします。

衝突（動作）–タグと共に使用し、基本のアクターがどのアクター、あるいはどの種類のアクターとなら、ぶつかった時に跳ね返るのを設定します。

衝突形態–他の物体がその物体にぶつかった時に、円形の（丸みのある）ものとぶつかったようになるのか、四角い（長方形の）ものとぶつかったようになるのか定めます。

属性管理（動作）–継続的に属性を更新する。例えば、アクターを常にマウスの位置にあるようにすることができる。基本的に、二つの属性を統合させることになる。

カメラ制御（動作）–常に画面上にあるよう、カメラがアクターを追いかけます。

ceil（関数）–切り上げで、整数の値をとる関数。例）ceil(0.3095)=1、 ceil(9.2850)=10、 ceil(-3.5)= -3

cos（関数）– 三角関数のなかの、余弦を示す関数です。cos(0)=1 となります。sin θ と cos θ について、詳細は次を参照下さい：http://en.wikipedia.org/wiki/Trigonometric_functions.

密度 – アクターの重さを示します。値の大きい(重い)アクターは、それより小さい値のアクターによっては動かされにくくなります。値がゼロだとその物体は動かなくなりますが、画面上の他のアクターに影響を与えます。密度は正の実数なら、どの値にも設定可能です。（すなわち、密度＞＝０）

削除（動作）– アクターをシーンから消去します。ブロック崩しやマリオのブロックのような、削除が可能なものに対して適用されます。ある規則に基づいて使われるのがよいでしょう。例: ブロック崩しでは、ボールとの接触で消える。

デバイス - [デバイス] ペイン (シーン編集の [ゲーム] タブ内) では、デバイス (マウス、タッチ、加速度センサー、画面、オーディオなど) に関連するさまざまな属性を変更できます。

テキスト表示（動作）– ゲーム内で表示される文字の色、サイズ、フォント、その他の設定を変えることができます。「アルファ」をゼロに変えると（属性リストの「カラー」の項目内）、テキストを除くアクターの全部分を見えなくすることが可能です。

編集 - このソフトウェアには、3 つの編集が用意されています。プロジェクト編集は、ゲーム全体の詳細を編集するために使用します。

exp（関数） – 指数関数 e^x において、e は約 2.71828182818。詳細は、次を参照下さい：
http://en.wikipedia.org/wiki/Exponential_function.

ローテーション設定 –設定により、画面上で他のアクターと衝突する場合には、ローテーションを止める事が出来ます。このボックスからチェックを外すと、アクターは他のアクターと衝突する場合でも、通常どおりローテーションします。

摩擦力 – 対象物が、他の物と接する時、この値を増やすと対象物の動きが遅くなり、ゼロに設定すると、摩擦が無くなります。（つまり遅くならない！）

関数 – 式編集機能で、サイン・コサインや log 計算など、様々な計算式が利用できます。

floor（関数） –切り捨てで、整数になります。例: floor(1.5)= 1、 floor(9.2850)= 9、floor(-3.5)= -4
重力 – E 各画面について X と Y の重力属性を設定することができます。実数の値をとり、移動するアクター全てに影響します。

グループ（動作）–規則や作用をグループごとにまとめることができます。

GameSalad クリエーター –コードの要らない、世界一のゲーム作成ソフトです！

画像ライブラリ – GameSalad に内蔵されている png ファイルの保管場所です。

インスタンス（アクター） – 基本のアクターの行動や属性、機能に変更を加えて作られたユニークなアクターの例。

改値（動作） – 一定の時間で属性（位置、値）が A から B になるよう設定が可能です。例: 100 から 1 （カウントダウン）になるようにしたり、X 地点から Y 地点へ移動・変化（あらかじめプログラムされた動きとして）させることができます。
iOS – iPhone や iPad、iPod Touch に使用されているアップル社の OS（オペレーティング・システム）。

In（関数）– 自然対数。自然対数は、約 2.71828182818 である底 e についての対数です。詳細は、次を参照下さい：
http://en.wikipedia.org/wiki/Natural_logarithm.

キーボード入力– キーボードで入力されたテキストを属性情報として保存することができます。

Kiip – ゲームの中で何かを達成したプレーヤーに対し、現実世界におけるご褒美が与えられるシステムです。「プロ」のアカウント保有者は Kiip のご褒美システムをゲームの中に組み込み、プレーヤーが現実世界におけるご褒美を受け取った場合には、その利益の分配を受けることができます。
レイヤー– アクターが互いに干渉するのを防ぎます。これにより、プレーヤーのアクターが背景やスコア、その他の物にぶつからないようにすることができます。

属性読込（動作）– 任意のキーについて、設定されていた値を常時保管のストレージから読み込みます。特定の入力、あるいは現象があった場合に、属性を変化させることが可能です。

ロジック –「規則」や「作用」が組み合わされて出来たプロジェクトが、どのように作動するかを定める場合において、その関係性を示す言葉です。

バグ修正通知記録（動作）– バグ修正ウィンドウに通知を記録します。「通知」はテストのためにゲームを進行させる際、イベントフラグを立てる為、エラーをチェックしたり、テキスト入力する為の属性値とみなすことが可能です。

log 10（関数） – 底 10 についての対数。例: log10(10)=1, log10(100)=2. 詳細は次を参照下さい：
http://en.wikipedia.org/wiki/Logarithm.

最高速度– この属性で、「進行」や「重力」の設定を通じ、アクターが到達できる最高速度を設定します。「速度の変更」や「改値」など、特定の「動作」においては、最高速度を超える場合がある。

可動性 – アクターが他のアクターとぶつかる場合、そのアクターはそのまま進むことができるのか、出来ないのか、個別に定めることができます。（進むようにするには、ボックスにチェックを入れて下さい）

移動（動作） – アクターや画面について、ある方向へ進むよう定めることができます。移動は他の「規則」や物体によって止められない限り、継続します。

移動先（動作） – ある特定の XY 座標上の地点に移動するよう設定ができます。アクターは到達すると止まるか、再び「移動先」で書き換えられる前の動きを再現します。

magnitude（関数） –原点（0,0）からある地点に直線で引かれた線分の長さを測ることができます。また差し引きすることで、ある二点間の長さを測ることも可能です。例: magnitude(x-x',y-y') あるアクターが（100,240）地点にあり、別のアクターが（25,30）地点にあるとすると、magnitude(25-100,30-240)でその距離を知ることができます。

max(関数） – 二つの、あるいは変動する値のうち、より高い方の値を示します。新しいスコアがそれまでのスコアより高いかどうかといったような、比較をする際に、非常に役立つでしょう。例: max(12,35)=35

min(関数）– 二つの、あるいは変動する値のうち、より低い方の値を示します。例: min(12,35)=12
備考（動作）– 自身や他の制作者のために、特定の作用や規則、アクター、グループについての覚書を残すことができます。ある事柄について、どうしてその様になっているのかを説明する際に役立て下さい。

URL 公開（プロ限定） – プロ会員に限り、特定の URL を記載し、特定の反応（ボタンをクリックまたは押すようにするのがおススメ）があった場合に、開かれるようにすることができます。URL は利用者の初期設定のブラウザで開かれます。

端末向き–アップル社製の端末（iPhone、iTouch、iPod）について、ゲームは縦方向（ポートレイト）に対応するのか、横方向（ランドスケープ）に対応するのかを設定します。

その他手段–規則を補完する要素です。元の規則が有効でない場合には、この項目で設定された作用が起こります。

パーティクル–ある定められた手順でアクターから出てくる小さなオブジェクトです。詳しくは「パーティクル（動作）」を参照下さい。

パーティクル（動作）–爆発を引き起こす！放射状、あるいは噴射状になるなど、イメージや色、継続時間その他の値についての設定が可能です。

BGM 一時停止（動作）–ある現象が起きた時に、その時点で流れていた音楽がとまるように設定します。

プラットフォーム–ゲームが稼働するデバイスあるいは記憶場所のこと。現在利用可能なプラットフォームには、iOS、Mac のデスクトップ、Android、HTML5 などがあります。

BGM 再生（動作）–音楽ファイルをスタートさせます。一度のみの再生、繰り返し再生ともに可能です。

サウンド再生（動作）–サウンドファイルをスタートさせます。一度のみの再生、繰り返し再生ともに可能です。

プレビュー–その場で、ゲームがどのように表示され、稼働するか確認することができます。GameSalad 内またはブラウザのウィンドウ上で見ることが可能です。アクターに適用した規則や動作が問題なく働くか確かめるのに使われます。
プロジェクト・サイズ–現時点で必要となるメモリーの容量は GameSalad Creator の右下の隅に表示されており、推奨される GameSalad アーケードゲームの最大値は２０MB です。

プロトタイプ(アクター) –プロトタイプアクターには、作用全般が適用可能です。ただし、参考例のうちのいくつかの要素（場所を作り出すことなど）は適用されません。

公開– GameSalad サーバー上にゲームをアップロードすることです。二進法に変換され、直接アップルや GameSalad アーケード、またはアンドロイド・ストアに投稿することが可能です。

ゲームの停止（動作）–シーンにおける全ての動作が停止し、別のシーンを開くことができます。ゲーム中、メニューを開く場合、もしくは単に動作を停止させる場合に使われます。元の画面に戻るには、停止解除を使用して下さい。

プレイヘブン–創造的で新しい広告ネットワークで、ゲームが始まる前にお薦めのアプリを紹介します。「プロ」アカウントのユーザーは、宣伝を入れるか入れないか選択することができ、また自身のゲームの中の広告について、成功報酬を得ることができます。

padInt(関数）–整数を任意の桁数で表示します。例えば、padInt(32,5) は「00032」と表示されます。ただし、常に、ある数 X の値を変えないために必要な最小限の桁数以上で表示され、padInt(38025,2) は「38025」と表示されます。

padReal（関数）–フローティング・ポイント（浮動小数点）について、切り上げた数値で、そのままの数値で表示します。例えば、 padReal(9.1234,15,6) は 9.1234 と少なくと合計１５桁まで表示される（小数を含む）。小数点以下は、最大６桁まで表示されます。すなわち「00000009.123400」となります。

pow（関数）– X を Y 乗した数値を表示します。例: pow(2,3)=2*2*2=8

prec（関数）–フローティング・ポイント（浮動小数点）について、小数点以下の桁数を任意に定めて表示することができまる。例えば、(1234.234,2)は「1234.23」と表示されます。

複製（動作）–属性や数値に基づいて、アクターの複製を行います。ライフがあといくつあるかを表示したり、あるアイテムの複製を作る場合などに使われます。

ゲームのリセット（動作）–全ての画面を初期状態に戻し、ゲームをリセットします。（いつ、どのようにして起こるのか規則を定めると良いでしょう。）

シーンのリセット（動作）–現在のシーンについて、画面上の全てのアクターを初期状態にし、画面をリセットします。（いつ、どのようにして起こるのか規則を定めると良いでしょう。）

独立解像度–端末の許容量に合わせて、画質を上げたり、下げたりすることが可能です。

反発力–アクターをバウンスさせます。0 はバウンスなし。2 はスーパーボール並みのバウンスです。バウンスの大きさは 0 から 2 の数値で設定します。

回転（動作）–単独で、右回り、あるいは左回りに回転します。また、何かキーボードのキーが押された場合や、衝突があった場合など、特定のことが起きた場合にのみアイテムが回るよう設定することも可能です。

回転角度（動作）–他のアクターや場面によってさだめられた方向へ回る

回転方向（動作）–特定の XY 座標上の地点や、他のアクターとの関係で定められる特定の位置方向へ回転します。

ルール–条件、あるいは様々な条件の組み合わせを定め、その条件が定められたときに、アクターが特定の動作行います。

ルール（動作）–条件、あるいは様々な条件の組み合わせを定め、プレーヤーの入力や属性の変更について適用します。

random（関数）–初めの整数、その次の整数と同じ、またはそれらの間の整数がランダムに選ばれます。例えば random(1,5) では、1、2、3、4、5 のいずれかが選ばれます。

属性保存（動作）–特定の値を名前を付けて永続記憶として保存することができます。ゲームをセーブすることが可能です。（セーブしたデータの読み出し方については、「属性読込」を参照してください）

シーン属性–アクターやゲーム全体ではなく、特にシーンについて定める属性。名前、時間、サイズ、重力、カラー、カメラ、自動再生が含まれます。

シーン編集は、アクターの配置、各種属性の変更、タグの設定などを行って、シーンを作成するために使用します。

アクターの配置（動作）–新しいアクターインスタンスをシーン内に作り出すことを指します。何かを発射させる場合やアイテムを残す場合などに使用します。

BGM 停止（作用）–音楽ファイルの再生を止めます。

sin（関数）–サイン関数。コサイン関数と類似していますが、波形パターンは 1/4 相殺されています。詳細は、次を参照下さい：http://en.wikipedia.org/wiki/Trigonometric_functions.

sqrt（関数）–ある値の平方根を導きます。マイナスの数値を入力すると、無効となります。

タイマー（動作）–「動作」や「規則」を一定の間隔で作動するようにできます。間隔は、秒単位で設定でき、何秒おき、何秒間というような設定が可能です。

tableCellValue（関数）–選択されたテーブルのセルの値を、行と列を指定して表示します。テーブルには１から順に番号が振られ、その行と列の名前はこの関数における入力値とすることも可能です。例えば、tableCellValue(game.Data,1,15) とすると、テーブル「データ」内の位１行１５列にその値が出力されます。

tableColCount （関数） – 選択された表の列数を表示します。

tableRowCount（関数） –選択された表の行数を表示します。

tan（関数）–タンジェント関数。詳細は次を参照下さい。：http://en.wikipedia.org/ wiki/Trigonometric_functions.

ゲームの一時停止解除 – ポーズ画面を取消、ポーズ状態だったシーンを再び動かします。

vectorToAngle（関数） –基点（0.0）から XY 座標上の特定の地点への角度を示します。例えば、vectorToAngle(100,200)=63.435。差し引きした角度を算出することも可能です。例えば、vectorToAngle(x-x',y-y'). VectorToAngle(100-200,150-250) とすると、点（ 100,150）と点（200,250）の間の角度が示されます。すなわち、１35 度ですね。