



GameSalad クリエーター

(Windows Version 0.10.0)

Written by Jack Reed
Layout by Anne Austin

目次

Windows 用クリエイター使用ガイド	3
はじめに	3
システム要件	3
イントロダクション.....	3
最初に	4
ライブラリ	4
バックステージ	6
ステージ	8
ゲームの作成	9
シーン	21
シーン属性	21
アクター	23
一般的属性	23
図形- 関連属性	23
運動- 関連属性	24
物理的- 関連属性 (非運動)	24
動作	25
属性	28
テーブル	29
テーブルの作成	29
テーブルの編集.....	29
テーブルでデータを使用する	29
特別なテーブル関数	29
GameSalad での公開	30
ヒントとベストプラクティス	36
FAQ(よくある質問)	37
トラブルシューティング.....	38
更なる支援	39
用語説明	39

Windows 用クリエイター使用ガイド

はじめに

システム要件

Windows 7 or Windows Vista

イントロダクション

GameSalad クリエーターは 2D オブジェクト指向ツールであり、完全なオリジナルのゲームを制作する事が可能なツールです。コードを 1 行も入力することなく、ドラッグ&ドロップのインターフェイスを使い、iOS、Android、HTML5、さらには Mac プラットフォームのゲームアプリケーションを作成ことができます。

ヒント: 現在、Windows 用クリエイターは Android と HTML5 での公開をサポートし、Mac 用クリエイターは iOS、Android、HTML5、および Mac プラットフォームをサポートしています。Android での公開には Windows および Mac 用の GameSalad 有償版(プロ)のアカウントが 1 つ必要です。

これはクリエイ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 39 ページを参照してください。

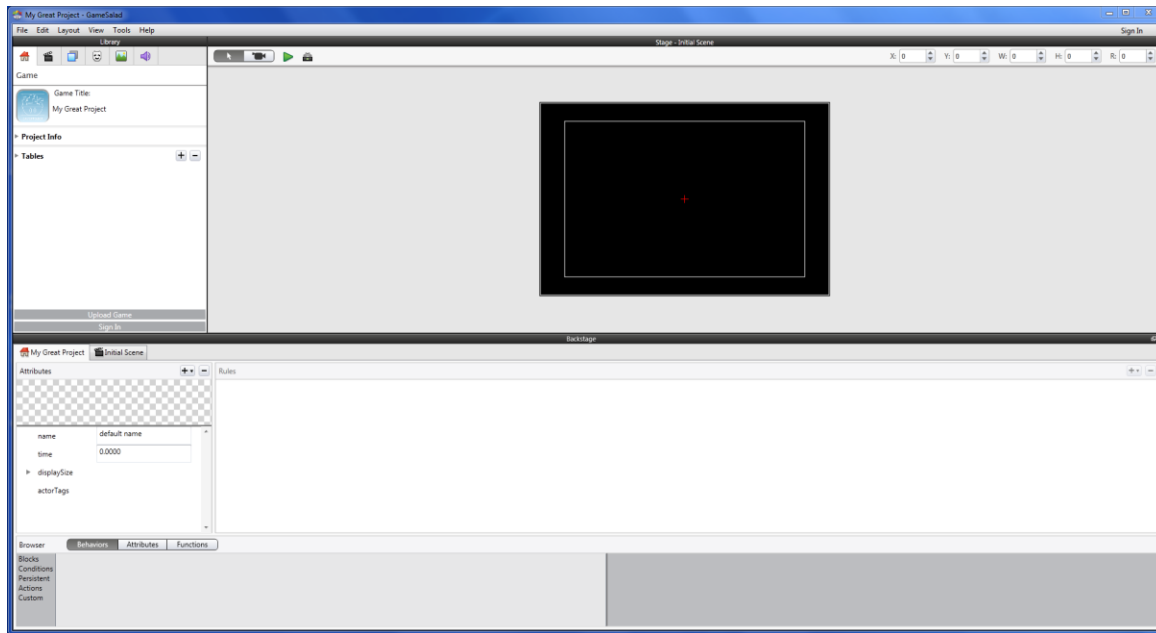
このあと数ページにわたって、最初にすべきこと、および重要な背景情報をいくつか説明して、ご自身の作品を作り始められるようにサポートしていきます。まず、GameSalad クリエーターをダウンロードして下さい。

クリエイターをダウンロードするには、gamesalad.com/creator にアクセスし、ダウンロードボタンをクリックして、最新バージョンを入手してください。
そして、これをアプリケーションフォルダにインストールすれば準備完了です。

ヒント: GameSalad のアカウントがなくてもクリエイ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。

最初に

GameSalad クリエーターを使うのが初めてでも、Mac 用クリエイターの熟練ユーザーであっても、Windows 用クリエイターを実行するとすぐに次の画面が表示されます。そこには数字ボタン、フィールド、タブがあります。

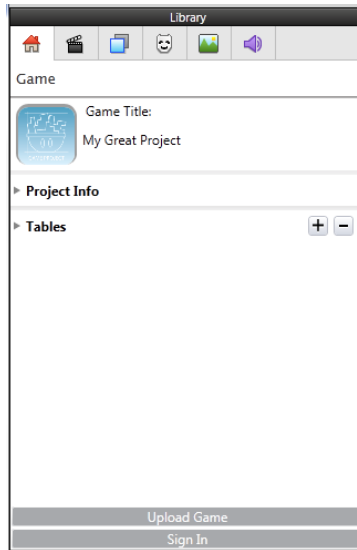


最初は覚えるべきことがたくさんあるように思えますが、心配しないで下さい。表示されているものはすべてより簡単で、スムーズなゲームの作成の為に用意されたものであり、このマニュアルを読めば、本ツールを使い始めるのに必要なことはすべて学ぶ事が出来ます。さっそく新しいプロジェクト（ここでは未公開のゲームを指しています）を開始しましょう。新しいプロジェクトは、クリエイターの起動時に自動的に立ち上がります。

ライブラリ

これで正式にクリエイターを利用することが出来るようになりました。まず、クリエイターのユーザーインターフェイスが3つの主要なセクション、[ライブラリ]、[ステージ]、および [バックステージ] に分かれていることに気付くでしょう。

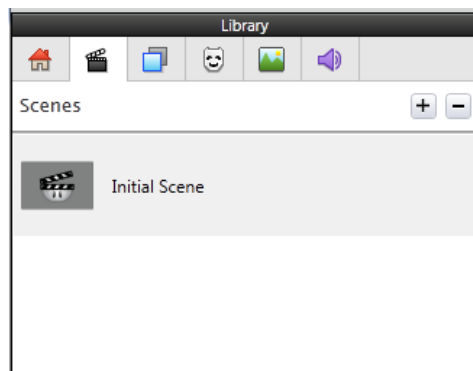
まずは、[ライブラリ] に注目してみましょう。[ライブラリ] はウィンドウの左上隅にあります。[ライブラリ] パネルは、別々に置かれたゲームのさまざまなキーコンポーネントを整理した状態で保つのに役立つと同時に、アクセスが容易なタブです。今はまったく新しいプロジェクトを表示しているので、これらのタブはほとんど空ですが、作業を進めるにつれてすぐにコンテンツが増えていくでしょう。



[ライブラリ] パネルに含まれるサブタブには、[ゲーム]、[シーン]、[レイヤー]、[アクター]、[画像]、[オーディオ]があります。前に述べたように、これらのタブにはプロジェクトの重要なコンポーネントの多くが入っています。たとえば、[ゲーム] タブでプロジェクトにタイトルをつけたり、[画像] タブの[+]アイコンを使って画像ファイルをプロジェクトにインポートしたりできます。

ヒント: 画像ファイルの形式は、.png、.jpg、.jpeg、.gif、.tiff、.wdp、または.bmp が利用可能です。許容可能なサウンド形式は、サウンドには.ogg、音楽には.m4a が含まれます。

[シーン] タブを選択すると、「初期シーン」というタイトル (ダブルクリックして編集可能)のシーンが 1 つ表示されます。これは、後で詳しく説明するクリエーターの [バックステージ] セクションにあるデフォルトのシーン属性によって定義されます。画像やサウンドのインポートと同様、[+]アイコンを押して追加のシーンを作成できます。



シーンはプロジェクトを分割する主要な方法で、たいてい「レベル」や「ステージ」として考えられたり使われたりします。シーン属性 (サイズ、重力、X/Y ラップなど) やアクターの配置を含め、シーンごとにユニークな構成を持たせる事が可能です。

GameSalad ではシーン間の切り替えは自動的に起こるものではありませんが、代わりに「動作」 (正確には「シーンの変更」) によって引き起こされます。「動作」の内容、働き、配置場所は後で説明しますが、今はこのまま[ライブラリ] パネルの[アクター] タブをクリックしてみましょう。

ヒント: 新しいシーンを追加する時に重視する2つの主要な要素は、「次のシーンの読み込み中にプレイヤーに少し間があってもよいか」、および「アクターや動作を追加し続けて現在のシーンが重くなり、潜在的なパフォーマンスが低下しても支障はないか」という事です。

すべてのプロジェクトにおいて、アクターは重要な動作主です。それは、ゲームの主人公から、その主人公が通過するプラットフォーム、大冒険の中で放つ数え切れないほどの発砲まで、すべてが該当します。アクターは、ゲームのジャンルやニーズに合わせて役割を果たすように仕立てることが出来ます。すべての敵アクターを生み出す、不動不可視のアクターを作る事も可能です。

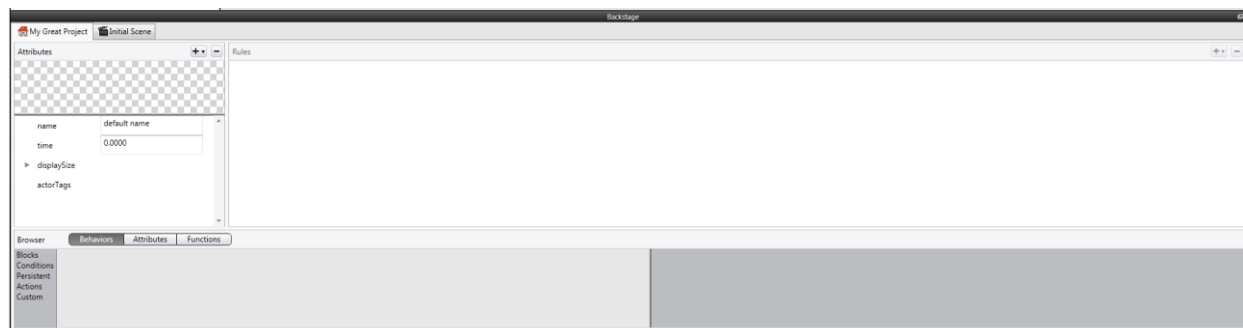
プロジェクトに現在存在するアクターはすべて[アクター] タブに表示されます (特定のシーンで使われるアクターだけではありません)。ここでは新しいプロトタイプアクターを作ったり、既存のアクターを削除したりすることも可能です。

プロトタイプアクターがシーンにドラッグされると (クリエーターの [ステージ] セクションで表示)、基本的にプロトタイプアクターのコピーであるインスタンスアクターが作成されます。プロトタイプとインスタンスの関係について説明すると、インスタンスアクターには作成元のプロトタイプアクターのすべてのロジックとほとんどの属性設定が含まれていて、最初は作成元のプロトタイプに「ロック」されて始まります。インスタンスアクターはロックされた状態でありながら、プロトタイプアクターに加えた変更は (いくつかの例外を除いて) 関係するすべてのインスタンスアクターに反映されます。

こうして、1つのプロトタイプアクターに関連付けた、たくさんのインスタンスアクターを作ることができます。これにより、時間を掛けてすべてのインスタンスアクターを個々に手動で変更する事なく、1つのプロトタイプアクターを編集するだけで大量の時間を節約することができます。

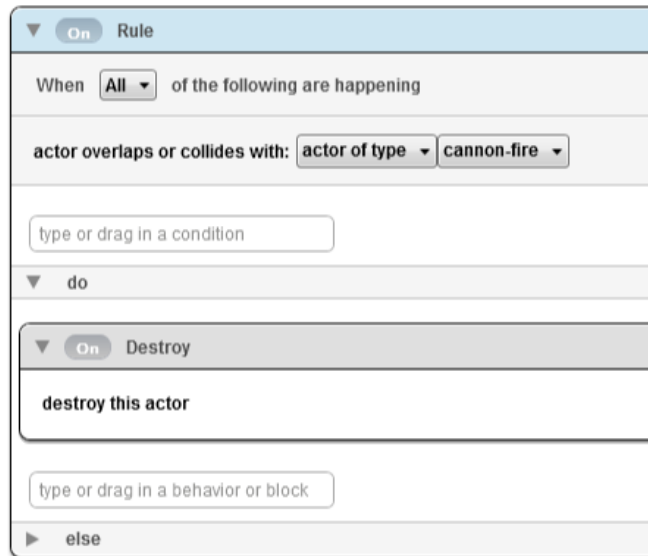
バックステージ (裏面)

[バックステージ] では、ゲーム属性、シーン属性、アクター構成 (ロジックおよび自己属性)、および設定したすべての**テーブル**のコンテンツにアクセスできます。



ここでのロジックとは、多くの場合「ルール」と「動作」を指しています。「ルール」とは、端的に言えば、定められた条件が真実または有効であるかどうかを考える直接的な **if/then** 文であり、この答えに基づき 1 つまたは連続した事前に定義されたアクション (「動作」として知られる) を行うものです。

具体的に考えるために、アクターが大砲の炎に重なるまたは衝突すると自己破壊するという「ルール」を想像してください。

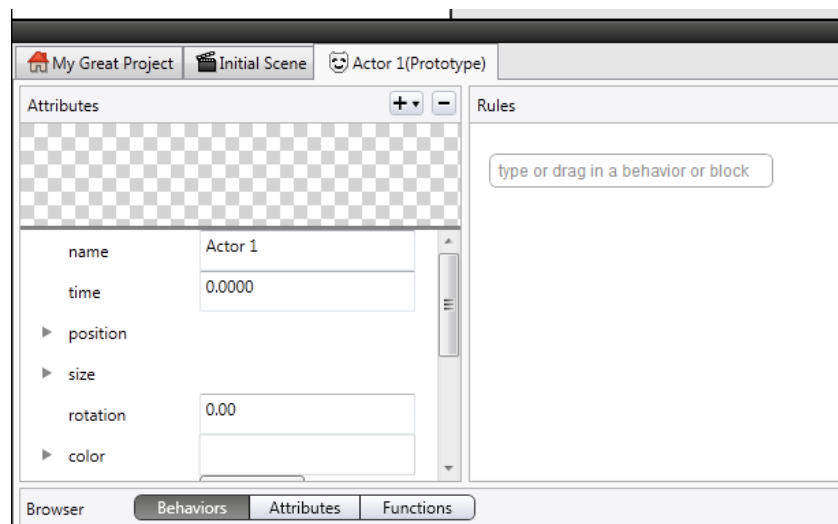


アクターが特定のタスクを達成するように設定したり、オブジェクトを設計するための特定の手順は本書の「ゲームの作成」セクションで説明しますが、上記ではクリエイターによってコードを学んだり入力したりすることなくアイディアを活かす方法の一般概念が分かります。

新しい (すなわち、ほとんど空の) プロジェクトを表示しているとき、[バックステージ] で現在利用可能なタブは [お気に入りプロジェクト] と [最初のシーン] の2つです。[ライブラリ] パネルを使って、[アクター] タブをクリックし、[+] ボタンを押して新しいアクターを作ります。

ヒント: シーンにはインスタンスアクターしか存在しません。例えば、プロトタイプアクターをシーンにドラッグするとそのアクターの新しいインスタンスができます。現在選択しているシーンのすべてのインスタンスアクターの一覧は、[レイヤー] タブに表示されており、[アクター] タブに一覧化されているのはプロトタイプアクターです。

「アクター1」という名前の新しいアクターが、サブヘッダー [すべて] に表示されます。「アクター1」をクリックすると [アクター1 (プロトタイプ)] という名前の新しいタブが自動的に表示され、[バックステージ] の焦点となります。



ヒント:[バックステージ] タブの名前に (プロトタイプ) があると、プロトタイプアクターを編集しているのだとわかります。同様に、(インスタンス) があると、インスタンスアクターを編集しているのだとわかります。ロックされていないインスタンスアクターを編集している間、[プロトタイプに戻す] ボタンを押してプロトタイプアクターと異なるロジックをすべて元に戻すと、これにより、インスタンスアクターはプロトタイプに再ロックされます。

空の[ルール]セクションから分かりますとおり、この新しく作られたアクターはロジックをもたず、動作方法を設定されるのを待っています。[バックステージ] の [ブラウザ]、[属性]、[ルール]セクションを使って設定を行います。

[属性] パネルでは、現在選択したタブの核となる構成設定の一部にアクセスできます。[初期シーン] が選択されていた場合、[アクター1(プロトタイプ)] というタブをクリックすると、異なる属性が一覧表示され、これらの属性に対する異なる値が表示されます。これらの両方のタブには、「名前」と「サイズ」の属性がありますが、[初期シーン] と違って [アクター1(プロトタイプ)] には [グラフィック] カテゴリもあります。これは、アクターとシーンが互いに独立していて一意の属性が反映されている為です。

属性はクリエイターの重要なアスペクトの 1 つであり、基本的に値のプレースホルダーです。これらは、シーン属性、ゲーム属性、およびアクター属性 (自己属性としても知られる) の 3 つの主要なカテゴリに分けられます。

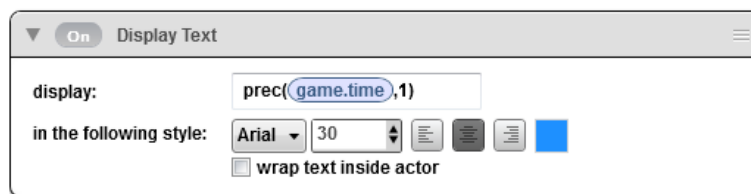
[属性] パネルの [+] アイコンを押すと、現在選択している [バックステージ] タブの新しい属性のタイプを選ぶように求められます。それらは、**ブール値**、**テキスト**、**整数**、**リアル**、**アングル**です (これらの違いは 28 ページの「属性」セクションで説明します)。これらのタイプのいくつかを他のタイプよりも頻繁に使っていると、アクターはその値を参照、変更、反応できるため、各タイプは非常に有効です。

属性とは、プロジェクト内のアクターが新しい敵を出すタイミングや、主人公が残した生命力や銃弾の量を知る方法であり、ゲームがゲームオーバー画面を出せるようにし、プレイヤーに新しいハイスコアの獲得を知らせるなど、多くのことを行います。

後でこれらの内容を詳しく説明しますが、今は [バックステージ] の [ブラウザ] セクションに進みます。[ブラウザ] セクションは、[動作]、[属性]、および [機能] の 3 つのタブに分かれます。

[動作] はすでに簡単に説明してあるので (後でさらに詳しく説明します)、[属性] タブをクリックしてみましょう。名前が示すように、[属性] タブは [属性] パネルに直接つながっています。今後、ゲーム、シーン、またはアクターの属性を参照する必要があるロジックを作ることが多くあります。このような場合、単純に、参照先を追加する「ルール」または「動作」のフィールドをクリックして、[属性] タブで関連する属性を選択します。

機能は、ゲームの裏側で行われている計算が期待に沿って作動していることを確認するために有力なツールです。例えば、機能「prec」(precision (正確) の略) を使うと、値の算出が可能な小数位の最大数を決めることができます。



このスクリーンショットには 3 つすべてのタブの要素が含まれています。「動作」(表示テキスト)、属性参照 (game.time)、機能 (prec または precision) があります。このアクターがシーンに置かれ、そのシーンが [再生] でレビューされるとき、プロジェクトが実行されている時間を小数 10 の位まで表示します。

掘り下げられる他の可能性は測り知れませんが、クリエイターの 3 つ目の主要なユーザーインターフェイス [ステージ] に進みましょう。

ステージ

[ステージ] では、新しく作ったアクターを置いて、生命を与えそれぞれが相互活動できるようにします。緑色の [再生] ボタンを使ってここからプロジェクトをプレビューして、迅速な反復作業をすることもできます。すなわち、変化を加え、即座にプロジェクトへのその変化の影響を確認でき、プロジェクト生成の情報をリアルタイムで得る事が可能です。

ヒント: 特に重いシーンでの作業中や低い解像度のモニターを使用中の場合は次のようにします。[バックステージ切替え] ボタンを使えば、一時的に [ステージ] でさらなる作業ができるようスペースを確保出来ます。

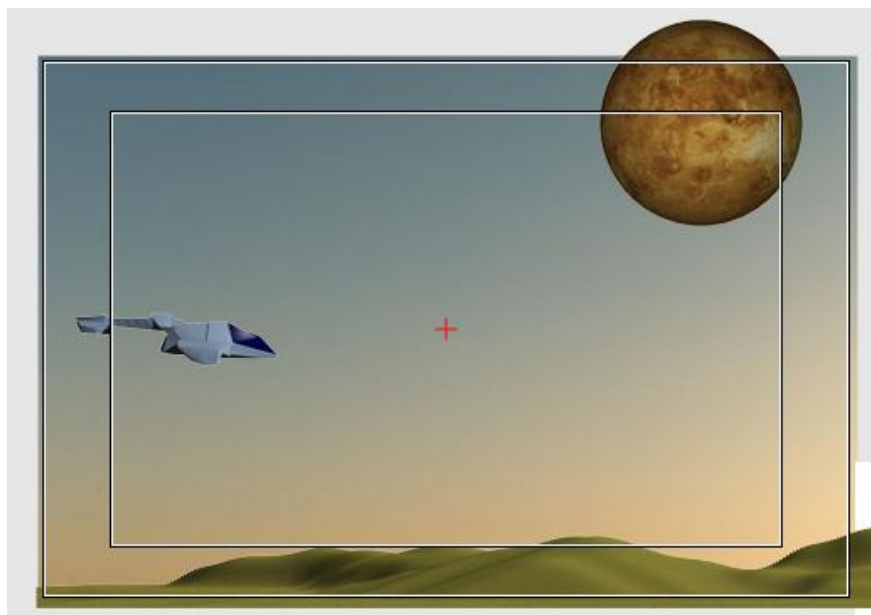
GameSalad クリエーターの基本的なアスペクトが構築されたので、次は単純なプロジェクトの作り方に進みます。

ヒント: 読むのではなく作業をすることで学ぶ場合は、この時点で、GameSalad クリエーターを始めるための情報を十分に紹介出来ました。もちろん、もう読み進めるのを止める事をお勧めしているわけではありません。この後も、知っておくべき多くの情報を紹介しておりますので、是非続けてお読みください。

ゲームの作成

まだご紹介出来ていないメニュー、オプション、機能はたくさんありますが、クリエイターについてすべてを覚えようとするよりは、今この場でゲームを作りましょう。始めはアートやサウンドのアセットを使えるようにテンプレートプロジェクトを使います。

クックブックに移動して「Basic Shoot Em Up」テンプレートをダウンロードします。それをクリエイターで開いたときに表示される指示に従います。

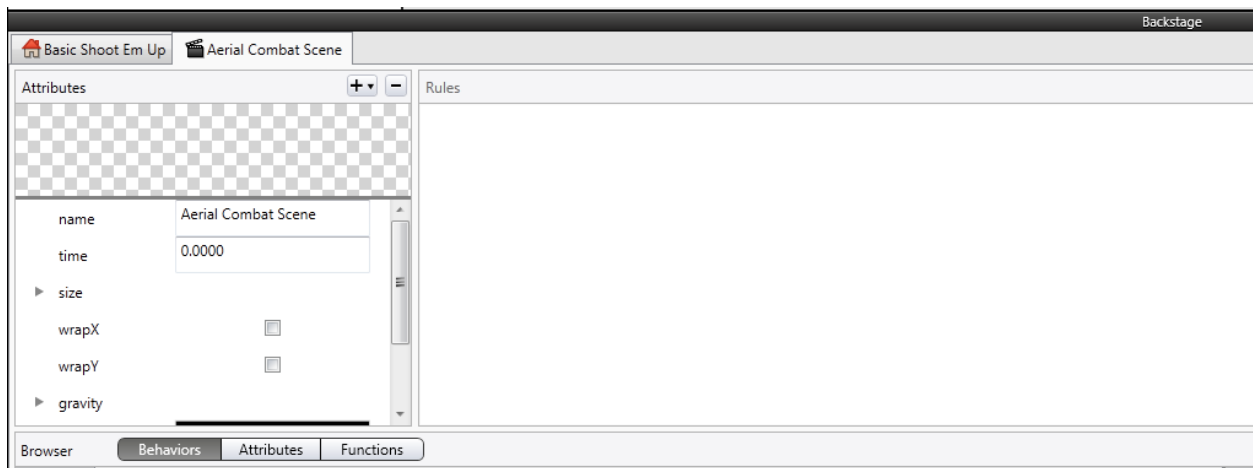


アクターはそのまま残して (このチュートリアルの後半で利用します)、[ライブラリ] パネルの [シーン] タブで「最初のシーン」というタイトルのシーンを選んで削除し、[-] アイコンを押します。

ヒント: テンプレートの元のバージョンをいつでも再ダウンロード可能ですので、これらのプロジェクトに独自の変更を加えても心配ありません。トライとミスを繰り返すことは素晴らしい学び方なのです。とはいえ、複数のファイルに頻繁に保存を繰り返して下さい。ユーザーからよく報告される問題は、1つのファイルに保存している場合のトラブルです。

[+] をクリックしてステージングポイントとなる新しいシーンを作ります。デフォルトの名前 (シーン 1) をダブルクリックし、このシーンに最適な名前をつけます。「Aerial Combat Scene」(負の引用)。また、新しいシーンを後でゲームを試すときに正しく再生されるよう、シーン一覧の最上部にドラッグしておきます。

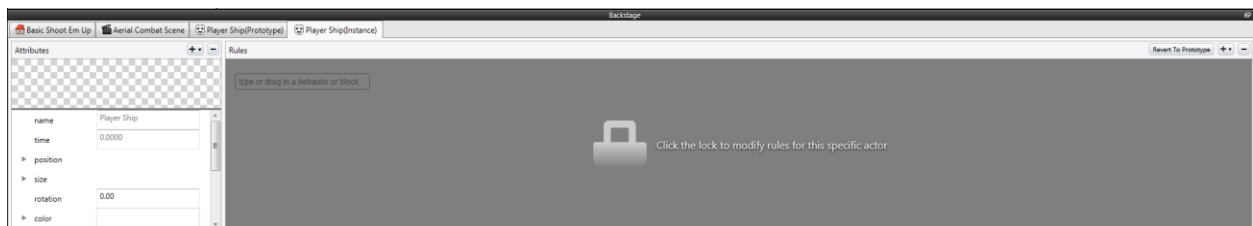
一覧の中のシーンをクリックし、[バックステージ] での焦点にして表示します。



[ライブラリ] パネルの [アクター] タブをクリックします。ここでも、[すべて] カテゴリに隠れている事前に作成されたアクターを削除してはいけません。ただし、しばらくこれらのアクターは使いません。代わりに [+] をクリックして新しいアクターを作ります。[アクター] タブの一覧の一番下に「アクター1」があります。

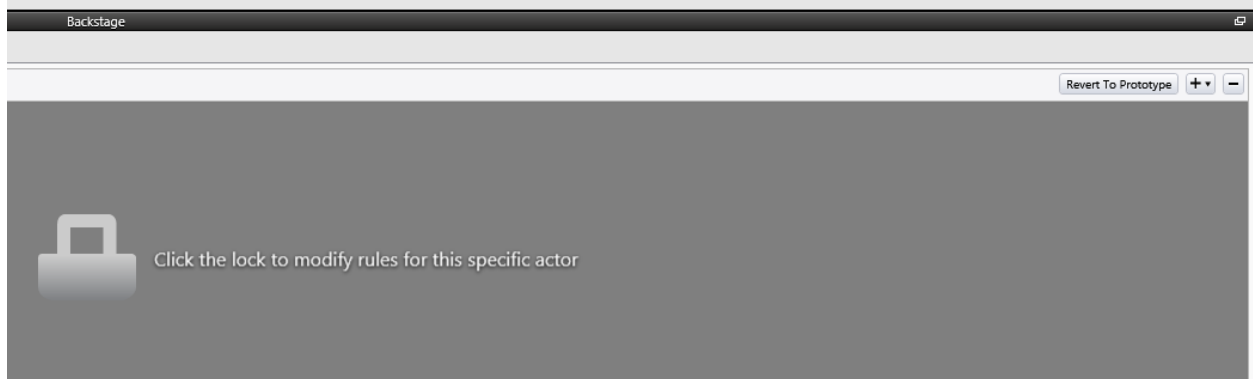
「アクター1」をダブルクリックして「プレイヤーシップ」に名前を変えます。それを画面左側の一覧から [ステージ] のシーンにドラッグします。

これは、現時点ではあまりピンときませんが、数分すると新しいゲームの核となるコンポーネントに変わることを認識出来るでしょう。今表示されているのはシーンにドロップされた1つのインスタンスアクターで、そのインスタンスアクターは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：二つ目の“プレイヤーシップ”アクターを場面に投入した場合、それらは同じ名前を持ち本質的に彼らのシーン内でのポジション以外はどのような角度からも全く同一のものにあります。この X と Y のポジションは、全てのアクターに初期設定で含まれている二つのセルフ属性によってトラックされます。セルフ属性はゲーム属性からは異なるものでありその中で各インスタンスアクターは自主的に自身のセルフ属性をトラックします。これは仮に敵の船を象徴するプロトタイプアクターに“ライフ状態”というセルフ属性を追加した場合、そのプロトタイプをベースとして発生した新しいインスタンスアクターはそれ自身のセルフ属性の値をトラックし続けます。私達がこれをゲーム属性と一緒にに行なおうと試みた場合、1つの敵を傷つけると全ての敵に影響します。セルフ属性を使用することで各敵船のライフは個別にトラックされます。これにより他に触れることなく1つの船を激しく傷つけることが可能です。

バックステージパネルの中のこの特別なアクターのタブには確認しておくべきいくつかのアイテムがあります。ロックされたインスタンスアクターに関してはこのマニュアルの中で既に何度か言及しています（そのロジック及び属性がそれを作ったプロトタイプアクターによって与えられたアクター）が、ここで我々が、まだプロトタイプにロックされているインスタンスアクターを見ている事を知らせてくれる見逃すことの出来ない通知を確認する事ができます。



ロックの表示をクリックすることで私達はこのインスタンスアクターがプロトタイプアクターの影響をそれ以上受けない様に変更することが出来ます。同様に、アンロックされている間はプロトタイプに対する変更はこれ以降インスタンスアクターに影響しません。インスタンスアクターを再ロックするには「プロトタイプに戻す」オプションを選択してください。

ヒント：ロックされたアクター達は多くの場合そのプロトタイプの複製ですが、彼らが完全に同一ではないことを覚えておきましょう。例えば各インスタンスアクターはサイズの幅と高さの属性同様、特有の X 及び Y ポジション属性を持つことができます。インスタンスアクターのこれら及び同類の属性値を変更することでプロトタイプとインスタンス間のロックを壊す事はありません。

ここではこのプレイヤーシップインスタンスアクターをロックしたままにしましょう。そしてその代わりにプロトタイプタブの「プレイヤーシップ(プロトタイプ)」をクリックしましょう。ロックの表示がないことを確認してください。これがプロトタイプにロックされた単なるインスタンスアクターだからです。逆ではありえません。

ほんの少しの間、このアクターにルールと行動パターンを追加しましょう。これはゲームにどう作動するかを知る為の大きなステップです。しかしまずはこのアクターを一般的な白い箱からシップアートアセットに変更しましょう。ライブラリの中の「イメージ」タブを開き“全て”カテゴリーを拡大してください。そして“船”というタイトルの PNG ファイルをステージの中のプレイヤーシップを表す白い箱にドラッグして離してください。

良くなってきましたね！しかしながら、現在の幅 100、高さ 100 という寸法は船を少し押しつぶした様に見せてしまいます。バックステージの属性パネルにある「サイズ」カテゴリーを拡大することでこの寸法を調整する属性を見つけましょう。そして幅を 75、高さを 50 に変更してください。

もしシーンの中で船のイメージのサイズに変更が見られないようであれば、あなたはこの調整をインスタンスアクターではなくプロトタイプアクターに行ってしまった可能性があります。プロトタイプアクターをアップデートすると、今後このプロトタイプアクターから作るインスタンスアクター全てにこのサイズが適用されることになります。しかしながら既存の全てのインスタンスアクターについては手作業によって変更しなければなりません。これはセルフ属性サイズがプロトタイプとインスタンス間のロックによって影響されないからです。

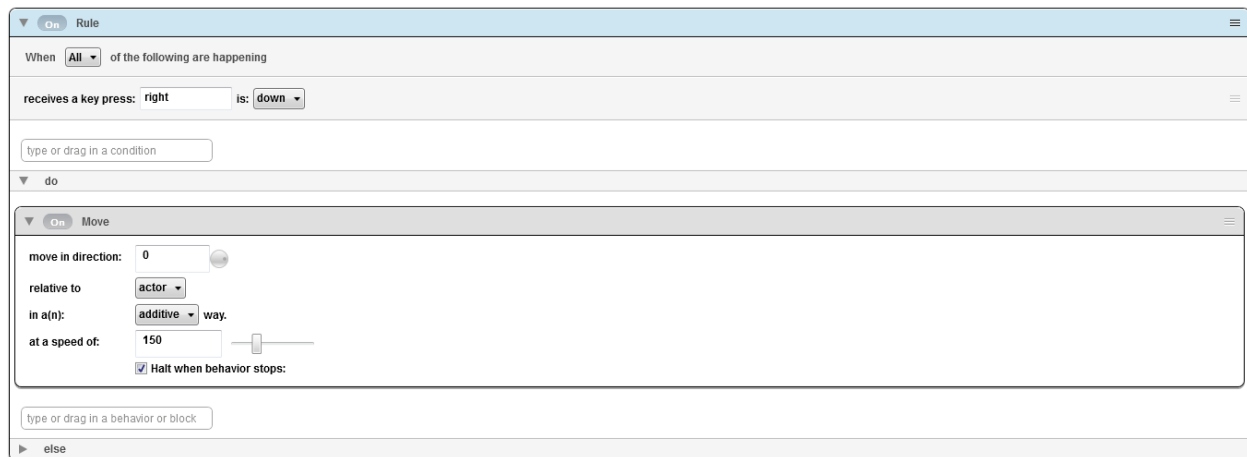


このプロトタイプにルールと行動パターンを加えるのと同様、まず何をしたいかを決定する必要があります。この船が重要な主役としてのプレイヤーとなっている為、プレイヤーに、船の動きをコントロールしたり、兵器を船から降ろしたりすることができます。更に危険な要素を登場させましょう。そうすることにより多大なダメージを受けた場合にプレイヤーの船がもしかすると破壊されうる状況を作ります。

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: タッチスクリーン機器用のムーブメントユーザーインターフェースの設定方法のいくつかの例として GameSalad クックブックの「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。

今、「プレイヤーシップ(プロトタイプ)」タブにいることを確認し、「ブラウザ」パネルの中の「ブロック」カテゴリからドラッグするか、“+”アイコンをクリックし、ドロップダウンメニューから「ルール」を選択することで新しいルールを作ってください。新しいルールを作成した後、キー 動作 (コンディションにあります)を“いつ”セクションにドラッグし動作を“実行”セクションに入れます。いくつかの簡単な調整の後、以下のように、見えるようになります:



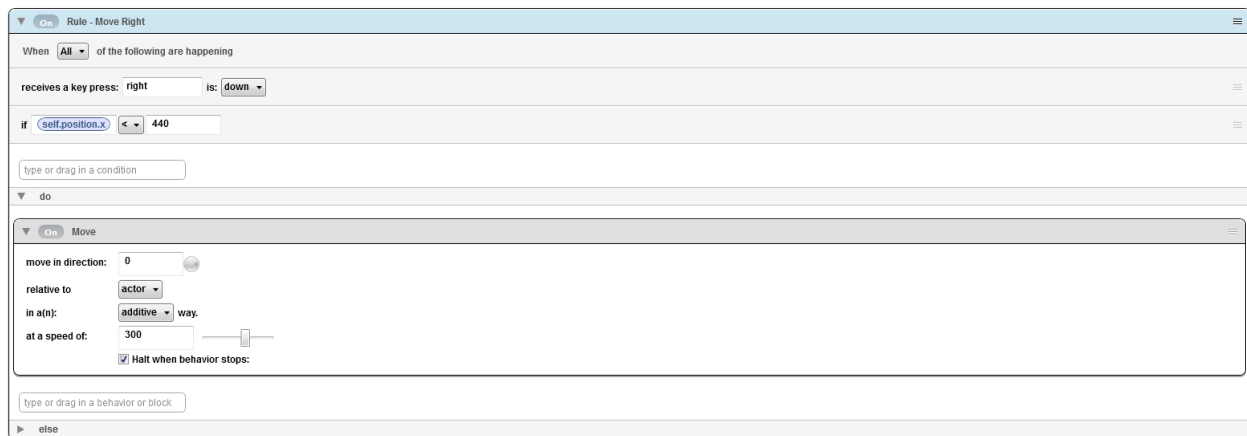
ヒント: 次回ルールを作成する際、オートコンプリート機能を利用してください。例えば“いつ”セクションの中に“キー”と入力し、エンターを押します。そして“実行”の下に“動き”とタイプしてください。これによりあなたは多くの時間を節約することができます。

キーボードの右矢印が押されるたびに“プレイヤーシップ”アクターが今いる位置から右に動くようになります。他の3方向への動きを手動で作るより、ルールを選択しコピー/ペーストを使ってそれら3つのコピーを作成してください。

これにより選択されたロジックと全く同じ複製が作成されます。（それに含まれている他のロジックも同様）さあ、この新しいロジックに対するコンディションキー及び行動の方向を編集すれば全て準備完了です。ステージパネル内の“プレイ”ボタンを押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、プレイ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“プレイヤーシップ”ではありません）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することができます：



ヒント: もしあなたがロジックの山が見られずに困っていたら、バックステージタイトルバーを使ってバックステージをアンドッキングしてみてください。再ドックする場合には単にもう一度  ボタンを押してください。

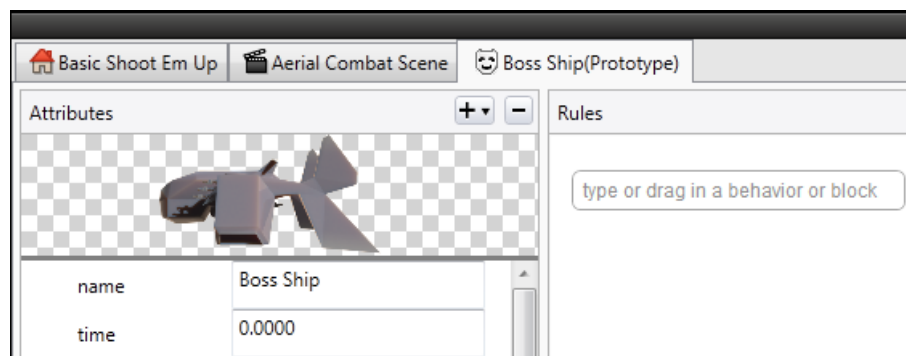
オリジナルの“船”アクターには動き動作が引き金となる様に右矢印キーが押されたのと同時に実現されるもう1つの条件があります。（“次の全てが実行される時”ルールが一番上にある必要条件に注意してください）本質的に他の3つの動きルールの文脈に取り込まれる時、それはアクターである“船”がその領域の目視できる部分の外にいないということです。

良ければ、自由にこのルールをあなたのアクターに補足してください。ただしあまり心配しすぎないで下さいね。このマニュアルの目的はあなたにクリエイターのコアコンセプトを提供することである為、最高のやり方を追求しすぎない様にしています。

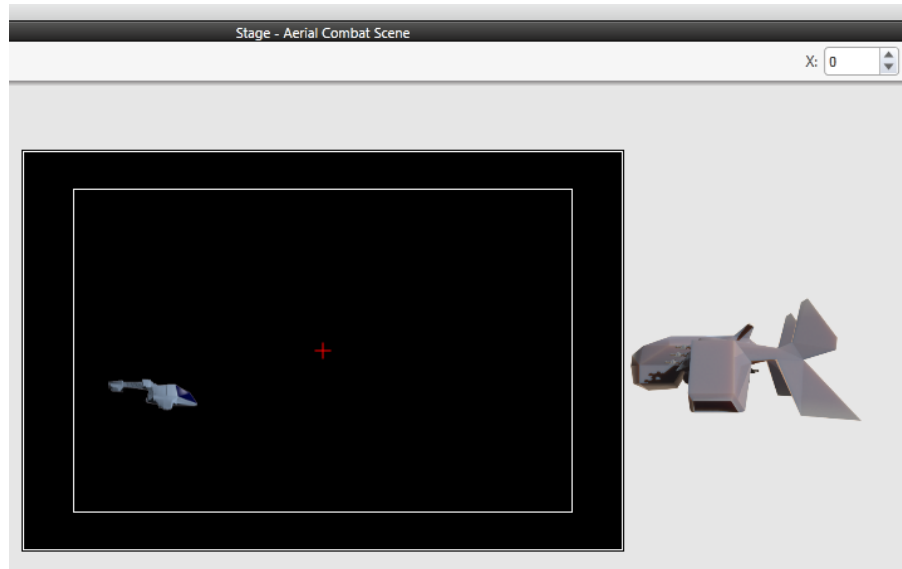
ヒント: テンプレートのルールは“ルール-右に動く”とされていますが、我々が作成したルールは単に“ルール”と名付けています。ルールもしくは行動のタイトルバーをダブルクリックすれば名前を変更することができます。適切な名前を使うことであなたもしくはあなたの共同作成者は、すぐにその部分のロジックが何を管理もしくは影響するものであるかを知る事が出来るでしょう。これにより過去に行った仕事を振り返る時に多大なる時間を節約することができます。

ここで、一度立ち止まって、どのような作業が残っているか確認してみましょう。（また、まだ行っていないようであれば時々セーブを行ってください）これは例としてのプロジェクトである為、私達はそれほど多くの時間をここに費やすことはしません。しかしながら、いくつかの基本的な追加は含めています：バックグラウンド、射撃する能力、射撃的となる敵（そしてまたあなたに対して撃ち返す）、勝つ条件、そして負ける条件。

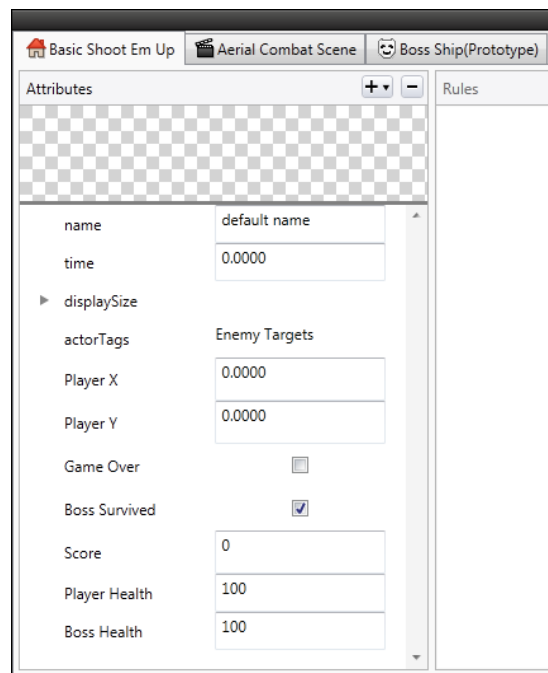
この方針の次のステップはライブラリーパネルを使用して新プロトタイプアクター作ることです。これを“ボスシップ”と名付けます。このアクターをシーンに登場させる前にイメージ タブから“ボス”のイメージを属性パネルのプロトタイプアクターのイメージセクションへドラッグします。（灰色と白のチェックのエリアです）更にプロトタイプのサイズ属性を幅 200、高さ 100 に変更してください。



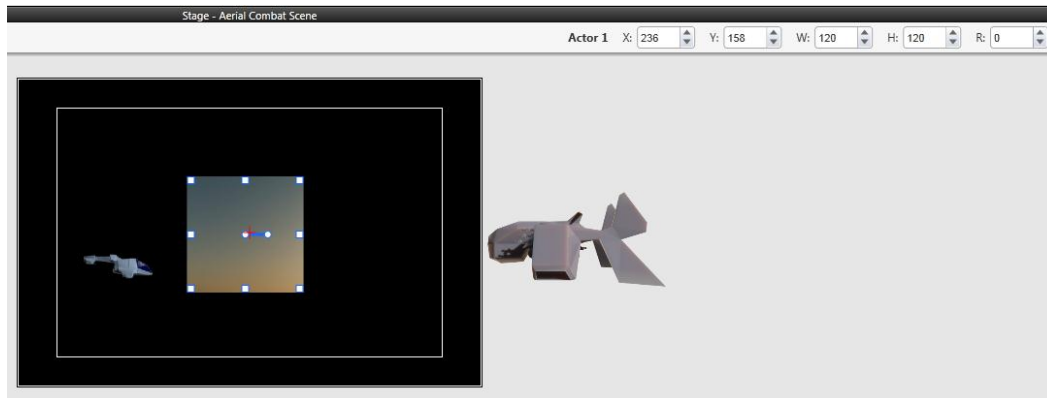
さあ、シーンに置いてみましょう。そして“ボスシップ”アクターを動かせばそれはもうシーンの目に見える部分の外側にいます。更なるドラマチックな効果の為に、シーンが始まればこれを目に見える区域に飛び込ませることができます。あなたの“空中戦”用のステージは現時点でこのように見えるはずで：



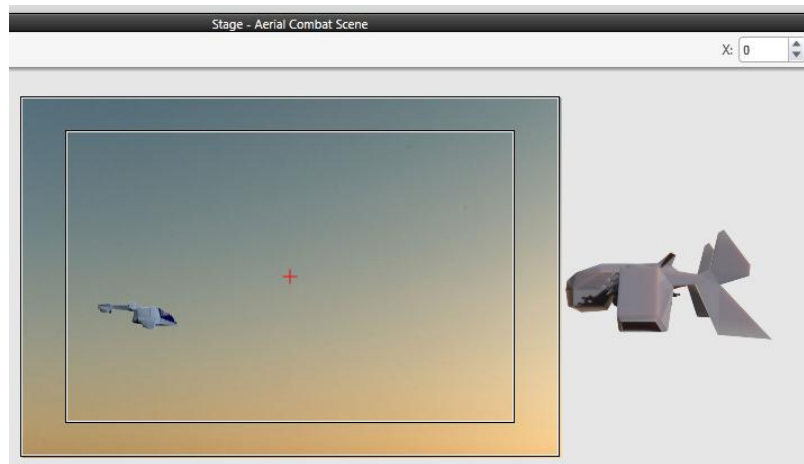
これら 2 つのアクターにライフ状態をトラックする能力を与える為に“Basic Shoot Em Up”タブが選択されている間に二つの新しい `integer`(整数)ゲーム属性を属性パネルの中に作成しましょう。そしてそれらを“プレイヤーのライフ”及び“ボスのライフ”と名付けましょう。彼らの初期設定値を 100 にセットします。



次はバックグラウンドを追加します。“サンセット” と名付けられているイメージを直接シーンにドラッグしてください。これは実際には新しいアクター（これは“背景”と名前を変更するべきものです）を作成する為のショートカットです。私たちはまた上右端のオプションを使い、直接ステージからこの新しいアクターのサイズを編集することで他のショートカットを使用します。



選択したサンセットアクターの“W:”（幅）フィールドを 480 に、“h:”（高さ）フィールドを 320 に変更します。残念ながら現在、“背景”アクターは“プレイヤーシップ”アクターを隠しています。これは単にステージ上の“背景”アクターを右クリックし、“後ろに送る”オプションを選択するだけで簡単に直すことができます。



ここからは既に存在するロジックをまた作成するよりも利便を考えいくつか先に作成したアクターを利用します。しかしながら我々のニーズに沿うものにする為にいくつかの微調整を行います。“ミサイル”アクターを開き、衝突/破壊ルールを調整します。これにより“船”を“プレイヤーシップ”と交換することでアクターは適切にそれがプレイヤーにあたった時に爆発する様になります。同じような変更を“レーザー”アクターに行います。一番下のルールまでスクロールダウンし、“アクター(タグ付き)”を“タイプのアクター”に変更しアクターを“ボスシップ”にセットします。

まだいくつか“プレイヤーシップ”及び“ボスシップ”アクターに設置するロジックの要素があります。これは下のスクリーンショット上で見る事ができます。まず、“プレイヤーシップ”についてみてみましょう：要約すると二つの制約属性行動を利用し“プレイヤーシップ”アクターは二つのゲーム属性を経由して常にその位置をゲームに報告します。これらの属性は両方とも“本当の”属性でありこれらが既にこの特定のテンプレートの一部である為、これらを作成する必要はありません。現在、他のアクター達はこれら二つのゲーム属性の値を利用したり反映したりすることができます。例としては“ボスシップ”が発砲する熱追尾式ミサイル等です。

ヒント: アクターがシーンの目に見える部分から永久に出てしまった場合（ミサイル等、素早く逃げるのに成功した敵）には、必ずそれらのアクターを破壊する特定の行動を設定してください。そうでなければあなたのゲームはパフォーマンスが落ちることになります。更にシーン自体の境界線を超えすぎて動いたアクターは自動的に破壊されます。

▼ On Constrain Attribute

constrain:

to:

▼ On Constrain Attribute

constrain:

to:

▼ On Rule

When of the following are happening

receives a key press: is:

▼ do

▼ On Spawn Actor

spawn actor:

in the direction: 

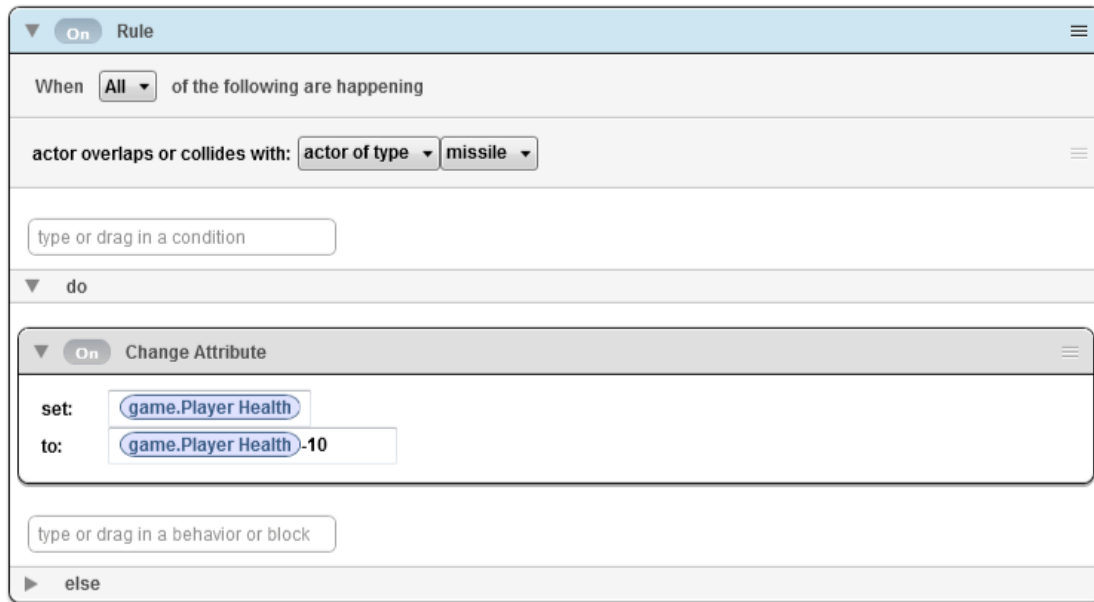
relative to:

from position: ,

relative to:

place:

▶ else

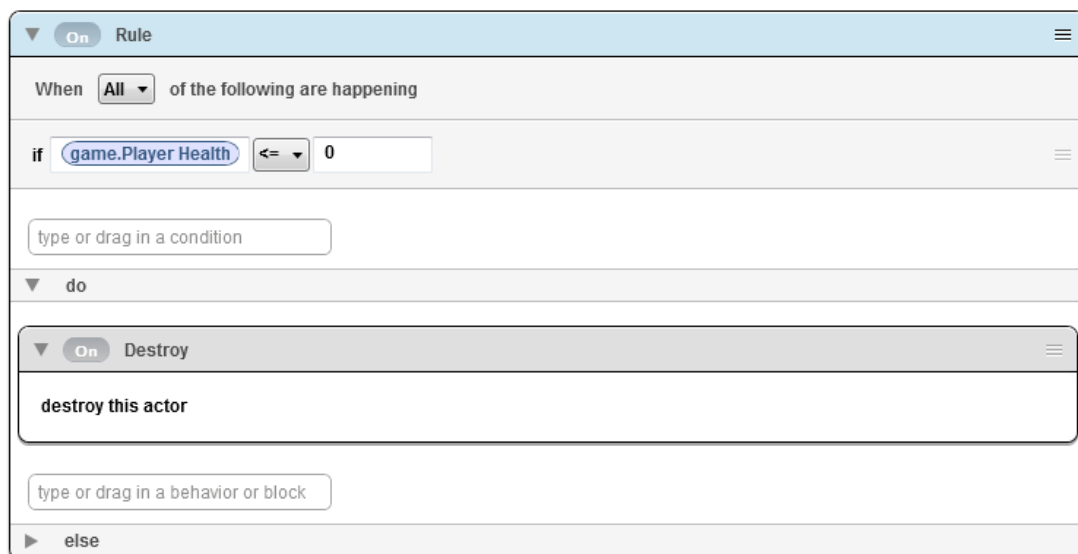


ヒント: アクターが彼ら自身のセルフ属性を参照できる一方、他のアクターのセルフ属性を参照することはできません。

この解決方法はゲーム属性をひとつ作成しレポーティングアクターによって変更属性 もしくは制限属性動作をアップデートすることです。そして参照アクターにチェックさせることにより、作成したゲーム属性が機能するようになります。

更にスペースバーを押すとレーザーアクターを発生させることによりプレイヤーのレーザーが発砲するというルールがあります。 簡単ですね。

ここで、特に付記したいある特定のアイテムがあります。それはゲームデザインに数えきれない可能性を開く数学の方程式と機能です。今回の状況ではとても単刀直入な方程式を使用し“プレイヤーシップ”がミサイルに当たる度に“プレイヤーのライフ”ゲーム属性を 10 下げる様にしました。そしてもちろんプレイヤーのライフがなくなった時点で彼らの船は破壊されます。



ここからはボスシップについてです：

我々のゲームでは“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イヤーの行動から独立して動きます。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。

“プレイヤーシップ”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”もレーザ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。(アニメーション動作とプレイサウンド動作の両方を備えています)

ヒント: 重要なことはルール及び動作はトップダウン命令で実行されます。もし同じルールの中に、二つの場面動作の変更を入れた場合に二つ目の場面動作の変更はルール自体が有効で適切にトリガを掛けられていても、実行されないということです。

もしあなたがこの効果をボスに付けたい場合は、プレイヤーシップが壊される前に戻って“小さな爆発”アクターを発生させる動作をプレイヤーシップのロジックに追加することができます。

The image displays two screenshots of a game engine's behavior editor interface. The top screenshot shows a 'Timer' block set to 'Every 1 seconds' with the 'Run to Completion' checkbox checked. Below it is a 'Spawn Actor' block configured to spawn a 'missile' actor in the direction of 0, relative to the 'actor' at a position of -50, 0, and in the place 'in front of actor'. The bottom screenshot shows another 'Timer' block set to 'For 10 seconds' with 'Run to Completion' unchecked. Below it is a 'Move' block configured to move in the direction of 180, relative to the 'actor' in an 'additive' way, at a speed of 20, with the 'Halt when behavior stops' checkbox checked.

On Rule

When All of the following are happening

actor overlaps or collides with: actor of type laser

type or drag in a condition

do

On Change Attribute

set: game.Boss Health

to: game.Boss Health -10

type or drag in a behavior or block

else

On Rule

When All of the following are happening

if game.Boss Health < 0

type or drag in a condition

do

On Spawn Actor

spawn actor: big explosion

in the direction: 0

relative to: actor

from position: -50, 0

relative to: actor

place: in front of actor

On Destroy

destroy this actor

else

上記はあなたがシーンを作成する際に取られ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。

この時点では空が限界です。ルールによってゲームにいくらでも調整をかけることが可能です。発射するよう設定されたアニメーション、場面内を動き回る様に命令されたアクター、流される音楽及び音、発生させられた追加のアクター達、プレイヤーの視点は変える事ができました属性のいかなる値も変更が可能です。するとこれらの属性が、あなたが気づくより前に、他のルールのきっかけになったりする場合があります。さて、ゲームが出来上がりました。

ヒント: クリエイターガイドのリマインダーは更に **GameSalad** クリエイターの特徴及び機能についての更に詳細な情報を提供しています。

このドキュメント全てを読むことをお勧めする一方、最初から最後まで読まなければいけないと感じるのではなく、是非、自由にあなたが今現在取り組んでいることが書かれている箇所に飛んで読んでみてください。

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イ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。

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、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。

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。対症的に、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いとなるでしょう。

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

アクターはプレイ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イヤーが話しかける、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な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。これらはプレイ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、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。可視不要なアクターを持つ理由は、音や音楽を演奏するためのロジックを含む場合のみです。

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 (+) と (-) ボタンを使用して作成したり取り除いたりできます。新しく作成された属性は属性リストの下部付近に強調表示されます。これらの属性はバックステージブラウザの「属性」タブからアクセスでき、特に断りのない限り、属性を変更する動作を通じて変更できます。

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。

シーンにアクターを追加する方法はいくつか別にあります。最初に、アクターをそのシーンで起動します。これを行うには、目的のシーンをアップロードして、そこにあなたが追加したいアクターをドラッグするだけです。第二に、アクターを「シーン外」で起動する可能性があります。ゲームのどこかの時点でそのシーンを入れることができます。シーンのサイズが (選択したプラットフォームによって決まる) カメラサイズより大きい場合、カメラの範囲外にシーンの領域が存在することに気付くことになります。ここでアクター達を配置し、様々な動きの動作を使用してそのアクター達をそのシーンに入れることができます。その動作は後述します。最後に、アクター起動動作を使用してアクターを「起動」します。これにより、シーンにおいてアクターを指定した場所と速度で配置することになります。

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 4 つのタイプがあります：

ブロック - その主目的と他の行動を含んでいるかを検討している動作。

条件 - 主に、キーボードを押したり、接触したり、衝突検出したり、そしてマウスをクリックしたりするというような入力をします。

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。サイズの変更にかか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エディター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。

属性読み込み - 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。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。条件にはプレイヤーの入力 (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。「属性読み込み」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спект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、属性パネル (バックステージ内) にあります。これらの属性内には、ゲーム、シーン、アクターのサイズ、位置、色などの情報が保存されています。カスタム属性には、レーシングカーの速度やパズルの残りピース数など、特有のデータが保存されます。属性リストの上の「+」ボタンをクリックして名前を変更することができます。

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「プレイヤーX」や「プレイヤーY」にしてプレイヤーのXYの座標値を記録します。

テーブル

テーブルの作成

ライブラリーパネルから「ゲーム」タブを選択し、「テーブル」カテゴリ内にある「+」ボタンをクリックして新規の表を作成してください。

テーブルの編集

「ゲーム」タブ内にある作成した表をクリックすると、新規のタブがバックステージに表示されます。

[行] と [列] の文字の左側には、編集可能なフィールドに入った数値があります。値を変更するにはボックスをクリックして、持たせたい行と列の数を入力してください。テーブルの各列には、属性タイプを選択できるドロップダウンメニューがあります。列のタイプを変更するには、ドロップダウンメニューの矢印をクリックして、指定したい属性を選択します。あるデータのタイプから開始して、それを後で変更した場合は、GameSalad はその列のすべてのセルを新しい属性データのタイプに変更しようとしています。セルに情報を追加するために、列/属性タイプを選択したこと、ならびに利用したい行数を設定したことを確認してください。単純にセルをクリックして、入力を始めてください。

GameSalad に.csv ファイルをインポートするには、データをインポートしたいテーブルまで進み、開いてください。ウィンドウの右側にある、反対方向を指す 2 本の矢印のアイコンをクリックしてください。インポートしたい CSV を選択し、[開く] をクリックしてください。

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-Count” 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

あなたが最新の素晴らしいプロジェクトに最後の仕上げを加えてゲームを世界中にシェアする準備がちょうどできている場合でも、まだ鋭意開発中であり **Android** デバイスでアドホックテストを行いたい場合でも、最初に **GameSalad** クリエーターであなたの作品を公開する必要があります。

ヒント: スケジュールされた期日より前に、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。

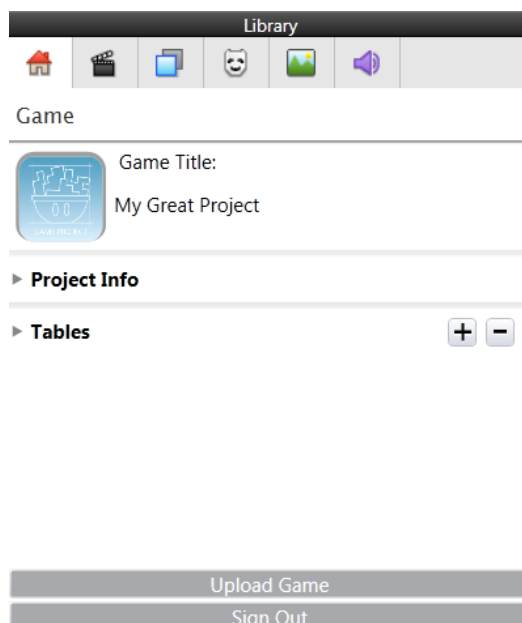
ほとんどのアプリストアはあなたのアプリケーションをリリースするまでに少なくとも 2 週間の時間を要するので、それに応じてプランを立ててください。

Mac 版のクリエイターをよく知っている方々は、このプロセスが重要な部分で異なっている点に注意してください。従って、このセクションを注意深く読み通すことをお勧めします。

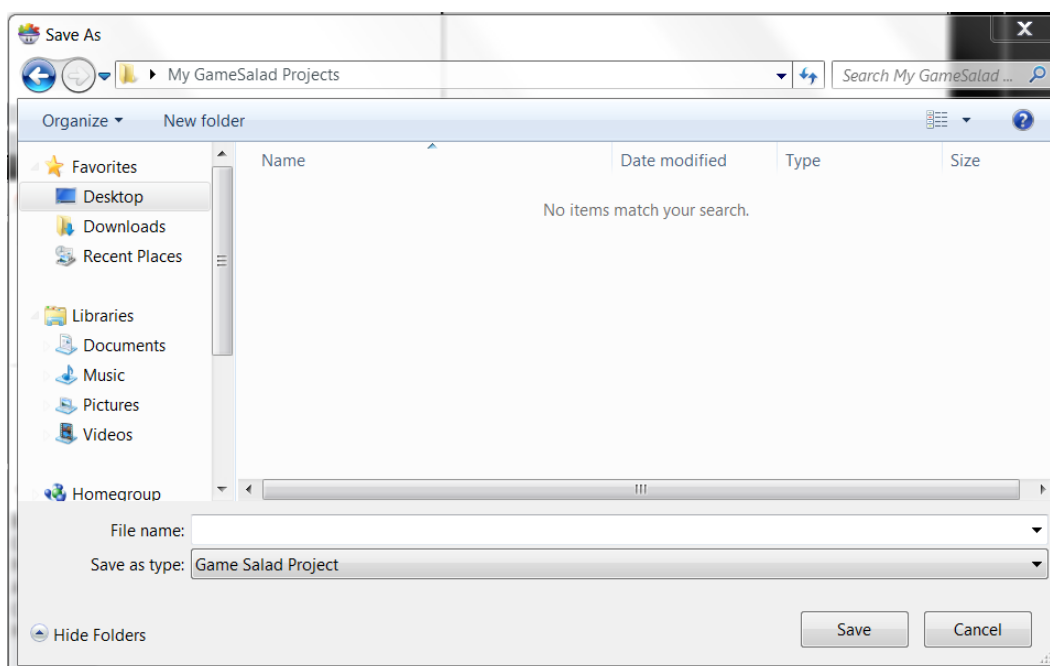
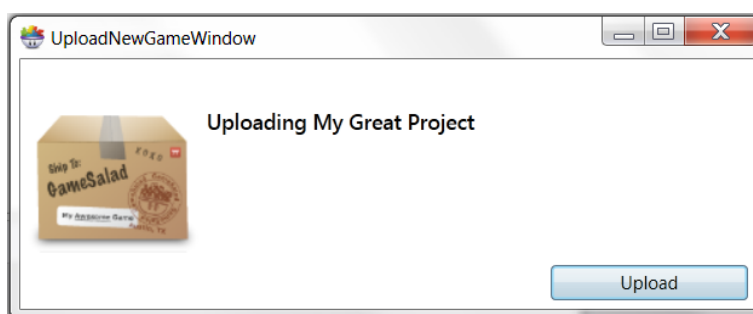
もしあなたが、新登場した **PC** 版の **GameSalad** を使っている数多くのユーザーのうちの一人であれば……ようこそ! これから、**GameSalad** アーケード を通してあなたのプロジェクトを **HTML5** 向けに公開する手順を、一つずつ説明します。すべてのユーザーが使用可能な **HTML5** 公開に加えて、有償版(**Pro**)ユーザーはテスト用とディストリビューション用の両方で **Android** デバイス向けの公開オプションも使用できます。**Apple** は署名プロセスに **Mac** プラットフォームを要求するので、**iOS** 向けに公開するには **Mac** クリエーターを利用する必要があります。

いったん公開する準備ができたなら (あなたのゲームが完成しているかまだ開発中かによらず)、最初のステップはあなたが取り組んできたプロジェクトファイルを **GameSalad** クリエーターで開くことです。

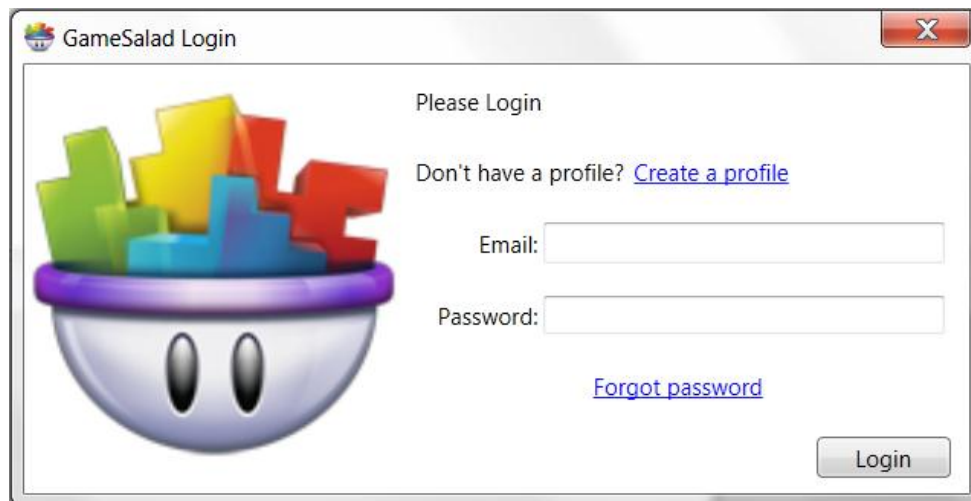
クリエイターのライブラリパネルの [ゲーム] タブにある、[サインイン/サインアウト] ボタンの上部に、[ゲームをアップロードする] ボタンの形式で公開用のオプションがあります。.



[ゲームをアップロードする] ボタンをクリックすると、アップロードの確認が行われ、その後 **GameSalad** プロジェクトファイルのコピーをあなたのマシンのローカルのどの場所に保存するかを尋ねられます。続けるには、ファイル名を選択して [保存] をクリックします。

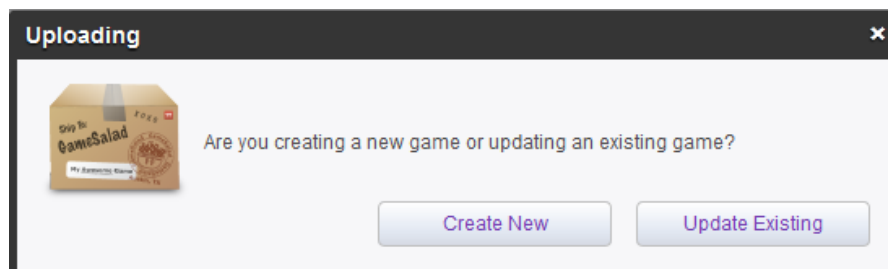


アップロードの際、アカウントにログインしていない場合、公開の要件であるサインインを求められる点に注意してください。もしパスワードを忘れてしまっても、心配しないでください! [パスワードをお忘れの方はこちら] オプションをクリックするだけで、パスワードリセットの手順を案内するウェブサイトが表示されます。



プロジェクトの保存のあと、プロジェクトの GameSalad サーバーへのアップロードが開始します。一旦これが完了したら、デフォルトのブラウザで、GameSalad のウェブサイトで管理される公開手順の第二部が表示されます。もしウェブサイトへのサインインが済んでいなければ、それと同じように求められます。

もしこれが Windows 版の GameSalad クリエーターでの初めての公開であれば、新しいゲームを作るオプションを選択してください。その後は、ゲームの開発が進むごとに新規のエントリーを作成するのではなく、[既存のものをアップデートする] ボタンを使って、プロジェクトファイルの以前の旧式のバージョンと、最新の機能の搭載・改善・バグフィックスがなされたバージョンとを交換することができます。既存のゲームを更新すると、GameSalad サーバーの既存のファイルが上書きされることに留意してください。そのため、ゲームのあらゆるバージョンのローカルコピーを確保しておくようにしてください。



HTML 5 向けの公開

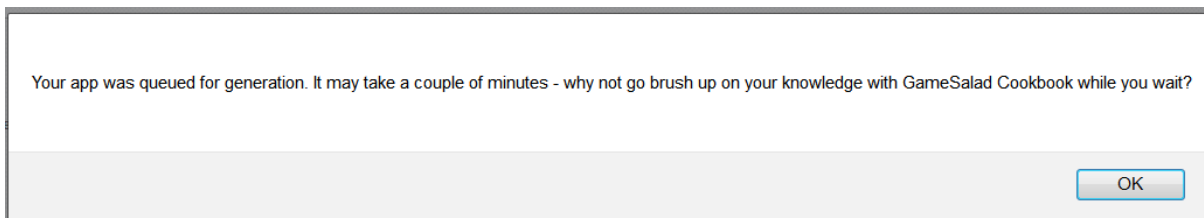
一般情報タブには、プロジェクトについての重要な情報 (タイトル、説明、操作方法、タグ、その他) を表示するだけでなく、スクリーンショットやアイコンをアップロードすることもできるいくつかのフィールドがあります。

プロジェクトに関するこれらの詳細とアセットを提供すると、ウィンドウ上部右隅のプログレス バーが徐々に満たされます。このバーが 78%に達したら、あなたのゲームを GameSalad アーケード上に載せることができます!

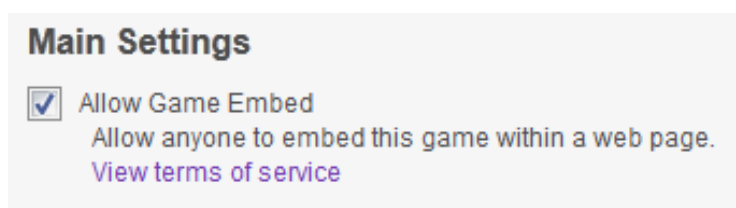


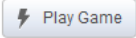
すべてのフォームに入力することをお勧めしますが、実際には次のアイテムを提供するだけでアーケードにゲームを追加することができます：「ゲームタイトル」「アイコン」「スクリーンショット」

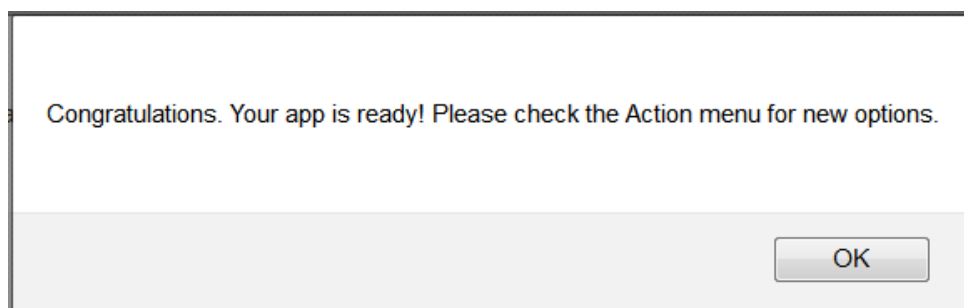
上記のアイテムの入力が済んだら、ページ左側のプラットフォームの下にある **HTML5** タブをクリックしてください。あなたは **78%** の基準値を超えたので、右側にある [公開する]  ボタンが現在操作可能になりました。



おめでとうございます! 数分であなたのゲームは **GameSalad** アーケードで世界的にプレイできるようになります。もう一歩進めて、あなたのゲームをウェブサイトに埋め込みたい、または他者がそうできるようにしたい場合は、メイン設定で [ゲームの埋め込みを許可する] を有効にすれば可能です。



アーケードゲームが動き出したら、あなたはリンクを含んだポップアップ通知とウィンドウ内通知の両方を受け取るでしょう。加えて、[ゲームをプレイする] ボタンがクリックできるようになるでしょう： 



You have published your game! [Play It.](#) 

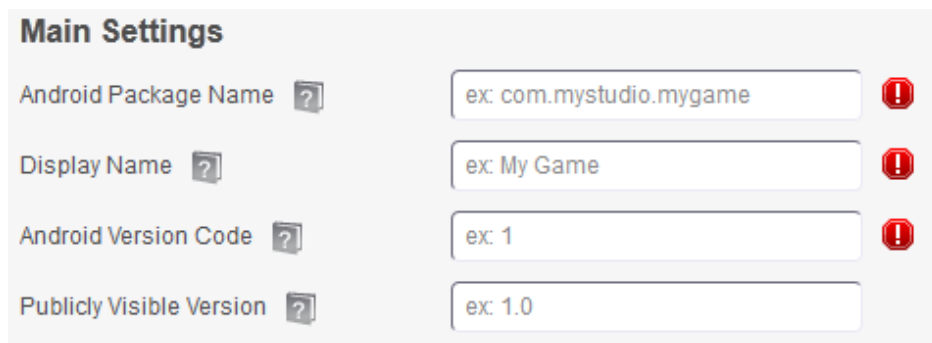
Android 向けの公開

有償版(Pro) ユーザーはさらに **Windows** クリエーターを使って、アプリの販売代理店に提出できる署名されたまたは無署名の .apk を公開し、ハードワークと才能から収益を得ることが可能になります。

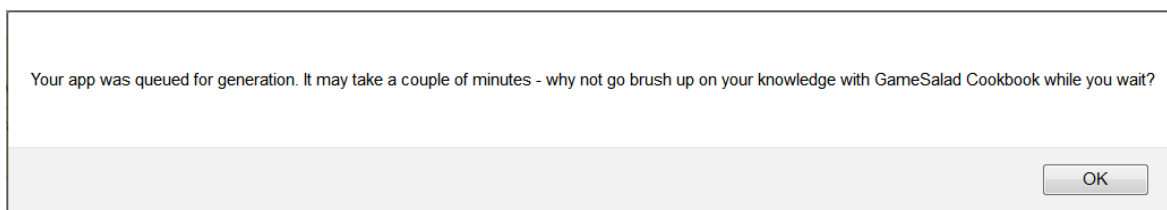
HTML5 向けの公開と違って、**Android** 向け公開は **GameSalad** アーケードで一般公開はされず、ユーザーが選択したアプリケーションディストリビューターでのみ公開されます。

とは言え、プラットフォーム特有のいくつかの構成とセットアップ要件のために、Android 向けの公開は HTML5 向けよりかなり複雑です。この次のセクションであなたが .apk ファイルをうまく生成できるように基本的な説明を行いますので、投げ出さないでください！

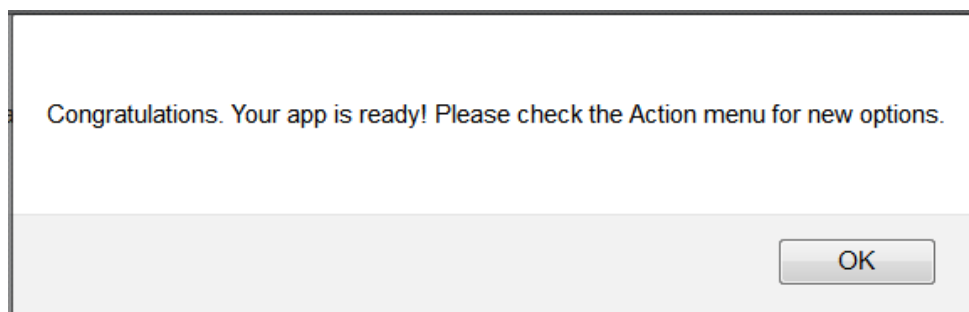
最初に、ブックアイコンにマウス オーバーしたときに表示されるツールチップの書式ガイドラインを使って、[メイン設定] ヘッダの下にある空白のフィールドを入力してください:



これらのフィールドを入力したら、[アプリを生成する] が選択可能になったことが分かるでしょう。数分以内に利用可能になる無署名の .apk の生成をキューするためには、このボタンをクリックしてください。



あなたの無署名の .apk が準備できたら、あなたは署名手順を開始するためのリンクを含んだポップアップ通知とウィンドウ内通知の両方を受け取るでしょう。加えて、[ダウンロード] ボタンと [アプリに署名する] ボタンがクリック出来るようになります。



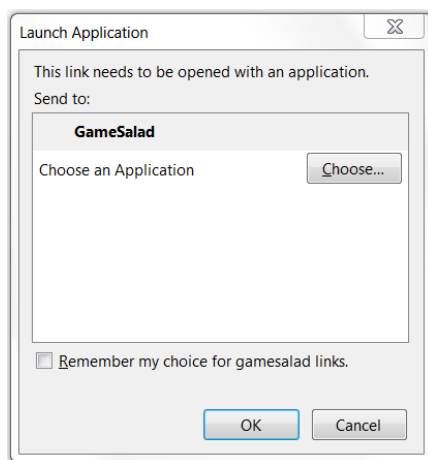
Congratulations. Your app is ready! You asked us to build it on May 17 at 04:19 UTC. To start the app the signing process [click here](#) 

この時点で、[ダウンロード] ボタンを使って無署名の .apk をダウンロードすることができます。ほとんどの場合、デバイス上でアプリケーションをテストしたり、第3者のアプリディストリビューター向けにアプリケーションを提出したりするために、署名された .apk が必要になります。しかし、これがすべてのケースで正しいというわけでは

ありません。例えば、Amazon は最後に彼ら自身でファイルに署名するようにしている為、GameSalad ユーザーには無署名の .apk の提出を義務づけています。

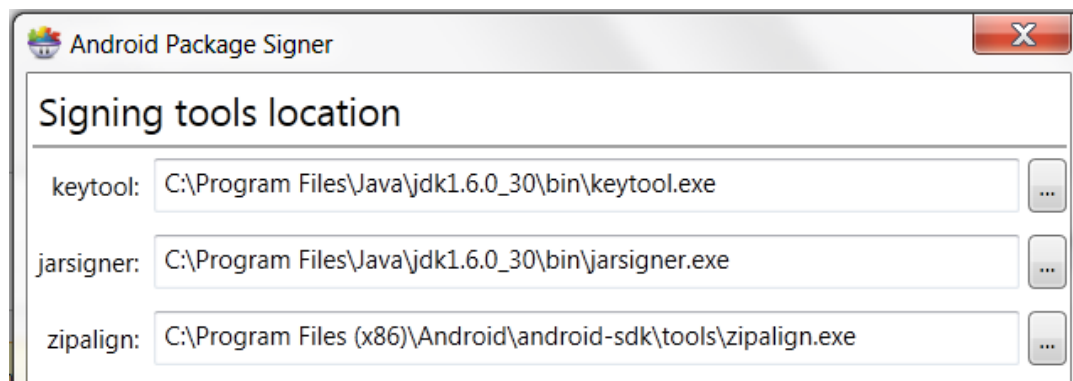
.apk に署名するのは、無署名の .apk を作るよりも複雑な手順なので、ウォークスルーのこの時点でクックブックの「Preparing Your PC for Android Development/Android 向け開発の為の PC セットアップ」(パート 1 とパート 2) で述べられている手順を使って、あなたの PC 環境をセットアップすることをお勧めします。

上記リンクで述べられている手順に従い、GameSalad クリエーターを通じて起動を依頼する [アプリに署名する] をクリックしてください。



この操作で Android Package Signer が表示されますが、それは 4 つの部分に分かれています。[署名ツールの場所] [キーストア] [キー] [署名するパッケージ]

[署名ツールの場所] では、それぞれのフィールドの隣にあるブラウザアイコンを使って、適切な .exe ファイルの場所を決定してください。各々のファイルの特定の位置はコンピューターごとに異なるかもしれませんが、以下の場所で、あなたはこれらのファイルをおそらく見つけられるでしょう。(もちろん、Android 開発向けに PC を準備したあとでになります)。



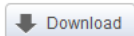
では、あなたの最初の PC キーストアの作成を始めるために、[新規作成...] ボタンをクリックしてください。保存は、見つけやすい (そして忘れづらい) 場所を選んでください。

では次に、すべての残りのフィールドを入力してください! キーストアのパスワードとキーのパスワードは同一でも良いですが、両方とも 6 文字以上の長さが必要です。最後のセクションの (署名するパッケージ) フィールドには、署名されるファイルのアドレスが自動的に入力されているでしょう。[署名する...] ボタンをクリックし、そして署名された.apk を保存するための安全な場所を選択してください。

これであなたは署名された .apk をアプリのディストリビューターに提出する前に、Android デバイスでゲームをテストする準備ができました。この手順を行う方法の詳細については、Prepping Your Device というタイトルの クックブックのエントリーを調べてください。

ダウンロードと削除のオプション

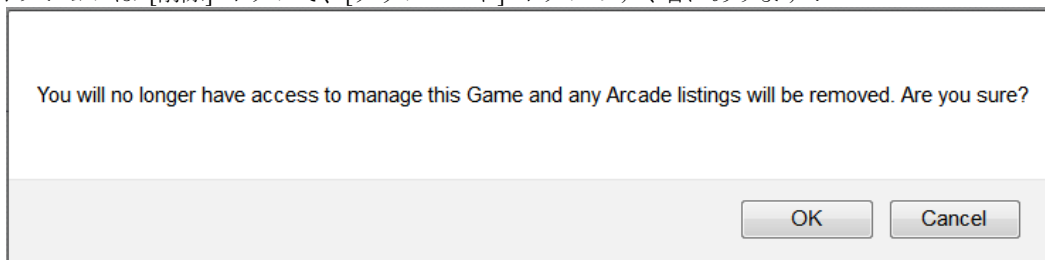
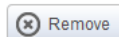
このセクションにて、2 つのオプションを紹介します。最初のひとつは [ダウンロード] ボタンで、一般情報タブのプログレス バーの真下にあります：



もしあなたが複数の PC を所有または使っていたら、この機能を使ってプロジェクトを 1 台のコンピューターでアップロードし、そして別のコンピューターでオリジナルのプロジェクトのコピーをダウンロードすることができます。アップロード手順の最初の部分を行わずにこのページにアクセスするには、ウェブサイトであなただけのアカウントにログインし、そして <http://publish.gamesalad.com/games> に移動してくださいそこから、ダウンロードしたいゲームのアイコンをクリックし、そして [ダウンロード] ボタンをクリックしてください。とても簡単で、便利です！

一般情報タブにあるダウンロードボタンと、Android タブにあるダウンロード ボタン (署名された .apk を Pro ユーザーに提供するもの) は違うボタンであることに注意してください。

第 2 のオプションは [削除] ボタンで、[ダウンロード] ボタンのすぐ右にあります：



このボタンをクリックすると削除確認のプロンプトが表示されます。[OK] をクリックするとアーケードからあなたのプロジェクトを削除し、そしてさらに、あなたのゲームへ張られたすべてのリンク (埋め込まれているかどうかによらず) が切れるでしょう。削除を承認したあとは、もう警告を表示したり決定を変更する事ができなくなりますので、慎重に考えて、プロジェクトの最新のコピーがローカルにバックアップされているかダブルチェックしてください。

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に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ください。
- 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。他の GameSalad ユーザーがあなたの問題の解決方法をすでに発見していないか、フォーラムで探してみることも検討してはいかがでしょうか。もし質問を投稿することに決めたのなら、投稿がロジカルで、状況を包括的に説明したものであれば、より役に立つ反応を得られる可能性が高いことに留意してください。

FAQ (よくある質問)

Q: Windows クリエーター から iOS 向けに公開できますか?

A: アップルは現在、アプリのバイナリが開発者証明書を使用して Mac 上で公開されることを要求しています。現時点ではまだ、iOS プロジェクトは Mac 版 GameSalad クリエーターを通してビルドされる必要があり、そしてあなたの開発者証明書をういたあなたの iTunes Connect デベロッパー アカウントを通して提出されなければなりません。GameSalad では現在、Windows ユーザー向けの代替ソリューションを模索していて、今後数週間~数ヶ月中にさらなる情報を提供出来るでしょう。

Q: Windows クリエーター から Mac クリエーター へプロジェクトを移行するにはどうしたらよいですか?

A: バージョン番号を気にすることなく、この移行を自由に行うことができます。すべての Windows プロジェクトは、Mac で開くことができます。プロジェクトを Windows から Mac へ移行するには:

1. Windows クリエーターであなたのプロジェクトを開いてください
2. ファイルメニューで [エクスポート] オプションを選択してください
3. エクスポートしたプロジェクトを Mac コンピューターに持って行ってください
4. Mac クリエーター でプロジェクトを開いてください

Q: 私はまだ Windows XP を使っています。Windows をアップグレードしなければなりませんか?

A: Windows 版の GameSalad クリエーター は、現在 WindowsXP をサポートしていません。多くの PC ユーザーが XP を好んで現役の OS として使用されていることは、事実ですので現在 GameSalad では、XP 版の提供に取り組んでおります。将来のアップデートでサポートしたいと思っています。

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/membership/pricing にあります。

Q: メンバー購読したのち、Pro の動作(機能)にはどうやってアクセスするのですか?

A: Pro メンバーシップに申し込んだ後に、GameSalad クリエーター からログアウトし、そして再びログインしてください。Pro にて提供される動作はバックステージの [動作] タブ内の、[Pro] セクションの下、ちょうど [アクション] タブと [カスタム] タブの下に位置しています。これらのオプションが表示されていないならば、数時間待って、もう一度お試しください。もしこの時点で Pro の動作がまだ操作不能になっていたら、GameSalad カスタマー サポート arcade.gamesalad.com/feedback に連絡していただくをお勧めします。もしあなたが e-check による支払を選択していたら、伝票のクリアランスが未完了の間、アクセス権が付与されるまで数日かか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: 新機能の追加または既存機能のカスタマイズのために、プロジェクトに私自身のコードを追加して使うことはできますか?

A: GameSalad の哲学は「みんなの為のゲーム開発」です。その為、私達は内部の開発努力をユーザーがコーディングを必要としない機能の提供に集中させています。とは言え、我々は常にユーザーの皆様のフィードバック、アイデア、そして機能のリクエストを開きたいと思っています。これらのコメントは私達のスケジューリングプロセスにおり込むことができますし、そしてユーザー様は GameSalad のフォーラム forums.gamesalad.com/categories/new-features で、考えを私達と共有することができます。どのような使い方が GameSalad ソフトウェアで許諾されるかについての完全な詳細については、gamesalad.com/terms で利用規約を参照してください。

Q: 自分のプロジェクト用のアートとサウンドのアセットを探しています。何かアドバイスはありますか?

A: プロジェクト用のアセットを入手できる場所は (実在の店舗とインターネットサイト両方に) いろいろあります。重要なのは、あなたはそのアセットをプロジェクトに含めるための著作権者の許可などの法的権利を有している、という確信を持つことです。ユーザーの中には自分自身でアセットを作成することを選ぶ人もいる一方、自分のプロジェクト向けに他者に作成を依頼するか、GameSalad マーケットプレースからアセットを直接購入する人もいます。

Q: 以前に公開したプロジェクト ファイルを復旧することはできますか?

A: 私達のウェブサイト <http://publish.gamesalad.com/games> (ログインが必要です) で前に公開されたプロジェクトを見つけてください。編集ボタンをクリックして、[一般情報] タブが選択されているときに [ダウンロード] ボタンを使ってください。これは Windows クリエーターの公開にのみ適用できることに注意してください!

トラブルシューティング

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く、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: 自分のプロジェクトを公開しようとするときエラーになります。

可能性のある解決方法: これは、いくつかの問題のうちいずれか一つに起因することがあります。公開を行うコンピュータの間違った設定から、ファイアウォール/アンチウイルスソフトの干渉、パブリッシュサーバーの混雑等です。GameSalad のカスタマーサポートに連絡して、公開している GameSalad クリエーターのバージョン、特定のエラー テキスト、および公開対象としているプラットフォームのバージョンを含む、できるだけ多くの詳細情報を提供していただくことをお勧めします。

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がありますか?

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
- 既存のアクターをシーンから削除するよりも早く新しいアクターをシーンに登場させていないか確認する
- タイマーと制約属性動作の使用を少なくする
- 1 秒の何分の 1 にセットされている “毎 X 秒” タイマーの使用を避ける

- 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て、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ください。

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、ゲームのプレイ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—画像編集用の初期の画面です。画面の中で、アクターを配置したり、動かしたり、回転させたりサイズの変更が可能です。

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: $\text{abs}(-5.23)=5.23$

acos (関数) —アークコサイン (逆コサイン) 関数。入力値 X は-1 から 1 までの値を取り、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 までの値を取り、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-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、プレイ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) $\text{ceil}(0.3095)=1$ 、 $\text{ceil}(9.2850)=10$ 、 $\text{ceil}(-3.5)=-3$

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。 $\sin \theta$ と $\cos \theta$ 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 - デバイスウィンド（シーン編集内の、ゲームのタブの下）では、マウス、タッチ、加速度センサー、スクリーン、音響装置を含む各種デバイスに関する属性を変更することができます。

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。

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: $\text{floor}(1.5)=1$ 、 $\text{floor}(9.2850)=9$ 、 $\text{floor}(-3.5)=-4$

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)。

ln (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イヤーに対し、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、プレイ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、プレイ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: $\log_{10}(10)=1$ 、 $\log_{10}(100)=2$ 。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ります。

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: $\text{magnitude}(x-x',y-y')$ 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、 $\text{magnitude}(25-100,30-240)$ 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) があつた場合に、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 (ポートレート) 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事が可能です。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 2 0 MB 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un–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 と少なくとも合計 1 5 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す。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を行います。

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、プレイ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できます。(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。

iAd 表示 (プロ限定作用) -アップルの iAd システムで文句や目的が定められた宣伝を表示します。

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 1 5 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。