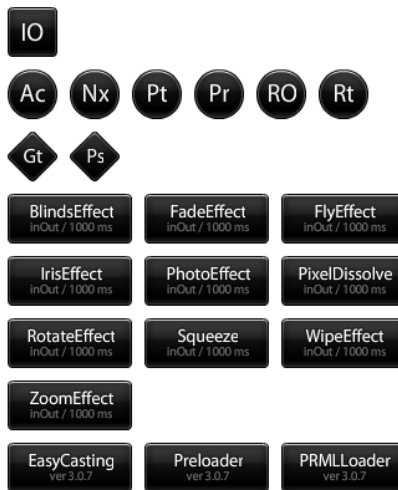


Flash 開発の新機軸「ActionScript ライブラリ」

SERIES #07

Progression コンポーネントスタイル編: ノンスクリプトでフルFlashサイトを構築

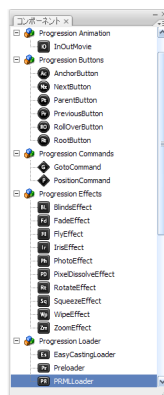
Progression は、HTML サイトが持つサイト基本機能を実装したフルFlash サイトを手軽に構築するためのフレームワークです。その利用方法には「コンポーネント」「タイムライン」「クラス」の3つのスタイルが用意されています。なかでもプログラミングに馴染みのない人でも利用できるのがコンポーネントスタイルで、ActionScript を一切書かずに、マウス操作だけでフルFlash サイトを構築できます。今回はコンポーネントスタイルを取り上げ、各コンポーネントの機能とそれを使ったサイト構築方法を詳しく解説しましょう。



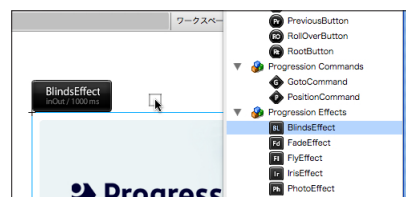
必要な作業はシンプルなマウス操作だけ

ActionScript に不慣れなデザイナーやアニメーターでもフルFlash サイトの構築が可能になるプラットフォーム、それがProgressionの「コンポーネントスタイル」です。このスタイルでは、ActionScript を使用せずにシンプルなマウス操作だけでFlash サイトの制作作業を行います。

Progression をインストールすると、Flash のコンポーネントパネルに、右ページの表にある22種類のProgression 独自コンポーネントが追加されます。MovieClip シンボルにこれらのコンポーネントをドラッグ&ドロップするだけで各種機能を実装でき、下図のようなページ構造と基本機能を持ったフルFlash サイトを簡単に構築できるのです。



Flash のコンポーネントパネルに Progression の独自コンポーネントが追加されます。詳細は右ページを参照ください



コンポーネントをドラッグ&ドロップするだけでサイトを構築できます

ディープリンク: 各ページにURLを割り振り、直接アクセス可能



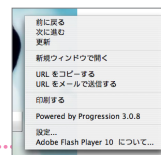
メニューボタン: 各ページにジャンプするためのボタン



ページ遷移時のエフェクト: 右ページにある10種類のエフェクト(Progression Effects)を自由に設定可能



コンテキストメニュー: 右クリックのメニューに「戻る」「進む」などのブラウジング操作を自動追加



ローディングの演出: 任意のフレームアニメーションを用意して、それをローディング演出として使用可能



□ Progressin コンポーネントリファレンス

Progression Animation	
InOutMovie 	このコンポーネントを内包するシンボルが画面に表示される、または画面から削除される際の効果を、フレームアニメーションで設定可能にする。
Progression Commands	
GotoCommand 	Progressionの動作中に、このコンポーネントが設置されているフレームを表示した際に、指定した移動先シーンに移動する。
PositionCommand 	このコンポーネントを内包するシンボルの画面上の座標を変更する。
Progression Loader	
EasyCastingLoader 	コンポーネントスタイルで開発する際に、外部XMLファイルを読み込んでProgressionを起動させる処理を実行する。このコンポーネントはFlashのステージ上(ドキュメントルート)以外に設置することはできません。
Preloader 	タイムライン・アニメーションを使用したプログレスバー表示を設定する。このコンポーネントはFlashのステージ上(ドキュメントルート)以外に設置することはできません。
PRMLLoader 	PRML形式のXMLファイルを読み込んでProgressionを起動する。このコンポーネントはFlashのステージ上(ドキュメントルート)以外に設置することはできません。

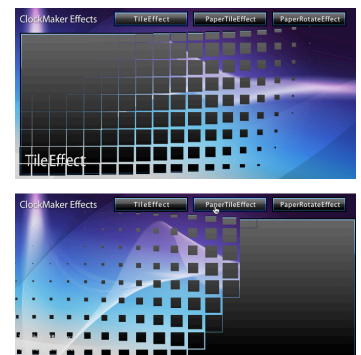
Progression Buttons	
AnchorButton 	このコンポーネントを内包するシンボルをボタン化し、クリック時に指定した外部リンク先に移動する。
NextButton 	このコンポーネントを内包するシンボルをボタン化し、クリック時に現在シーンの次に位置するシーンに移動する。
ParentButton 	このコンポーネントを内包するシンボルをボタン化し、クリック時に現在シーンの親に位置するシーンに移動する。
PreviousButton 	このコンポーネントを内包するシンボルをボタン化し、クリック時に現在シーンの前に位置するシーンに移動する。
RollOverButton 	このコンポーネントを内包するシンボルをボタン化し、クリック時に指定した移動先シーンに移動する。
RootButton 	このコンポーネントを内包するシンボルをボタン化し、クリック時にルートシーンに移動する。

Progression Effects*		
BlindsEffect 	徐々に表示される矩形、または消えていく矩形を使用したエフェクトを適用する。	 
FadeEffect 	アルファ効果によるフェードイン/フェードアウトするエフェクトを適用する。	 
FlyEffect 	指定した方向からMovieClipオブジェクトをスライドインするエフェクトを適用する。	 
IrisEffect 	正方形のシェイプ、または円のシェイプがズームインまたはズームアウトするアニメーション化されたマスクエフェクトを適用する。	 
PhotoEffect 	写真のフラッシュのようなエフェクトを適用する。	 
PixelDissolveEffect 	チェッカーボードのパターンでランダムに表示される矩形、または消える矩形を使用したエフェクトを適用する。	 
RotateEffect 	回転エフェクトを適用する。	 
SqueezeEffect 	水平または垂直に拡大/縮小するエフェクトを適用する。	 
WipeEffect 	水平方向に移動するシェイプのアニメーション化されたマスクを使用したエフェクトを適用する。	 
ZoomEffect 	縦横比を維持しながら拡大/縮小するエフェクトを適用する。	 

※ Progression Effects コンポーネントのエフェクト効果は、コンポーネントを内包するシンボルが画面に表示される際、または画面から削除される際に適用されます。

拡張トランジション 「ClockMaker Effects」

Yasuさんのサイト「ClockMaker」では、3DライブラリのPapervision3Dを使ったProgression拡張トランジション「ClockMaker Effects」が公開されています。Progression標準搭載コンポーネント同様にドラッグ&ドロップするだけで、画面がタイル上に分割されて表示される3Dエフェクトを追加できます。ただし、標準搭載コンポーネントとは違い、使用する際にインスタンス名を設定する必要があります。インストールやパラメータの設定方法については、同サイトを見て下さい。
<http://clockmaker.jp/project/flash-effects/>



TUTORIAL コンポーネントでフルFlashサイトを構築する

それでは、コンポーネントスタイルでP122に掲載しているサイトを制作してみましょう。3ページで構成されたサイトです。なお、ここではProgressionのバージョン3をベースに解説しています。バージョン3を利用するにはFlash CS3以降が必要です。



View & Download

サンプルデータをダウンロードする際には、目次に掲載しているIDとパスワードが必要です。



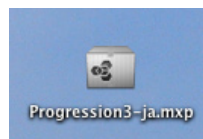
<http://book.mycom.co.jp/wd/>

STEP 01 Progression をインストールする

ProgressionはFlashの拡張機能として組み込まれ、そのパッケージは公式サイト (<http://progression.jp/>) から入手できます。インストールに成功すると、コンポーネントのほか、Flashのメニューの「ウインドウ」その他のパネルに「Progressionプロジェクト」と「Progressionシーンエディタ」が追加されます。



パッケージは、「MXP」「JSFL」「SWC」の3つの形式を用意しています。通常は、MXP形式を利用して下さい



ダウンロードしたMXP形式ファイルをダブルクリックすると、Adobe Extension Managerが起動してインストールが開始されます



Progressionシーンエディタ。Flashのシーン、つまりサイトのページ構成を定義します

Progressionプロジェクト。制作スタイルの選択、コンテンツサイズなどの基本設定を行います

STEP 02 プロジェクトのセットアップ

Progressionを使ったプロジェクトの作業を開始するには、まず「Progressionプロジェクト」パネルを開きます。今回はコンポーネントスタイルで作成するので、種類メニューから「コンポーネント(EasyCasting)」を選び、その下にタブで基本設定を行います。

設定後、「新しく作成する」ボタンをクリックすると、右図のようにプロジェクトに必要なファイル群が自動生成されます。これらのファイルのフォルダ構成やファイル名は変更しないでください。



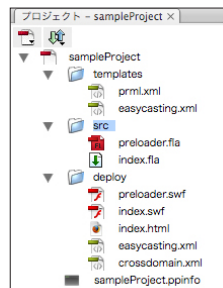
プロジェクトの名前は、書き出されるフォルダ名やFlashを埋め込むHTMLのtitle要素に使用されます



「Flash」タブでは、フレームレートやウィンドウモードを設定します



「一般」タブでは、Flashサイトの幅や高さ、背景色などを設定します



各種設定後、「新しく作成する」ボタンをクリックするとプロジェクトに必要なファイル群が自動生成されます。さまざまなファイルがありますが、Progressionの動作・実装に必要なものなので、そのままにしておいてください

STEP
03

シーン（ページ）構造を定義する

プロジェクトのセットアップが完了したら、「Progressionシーンエディタ」を使って、シーン（ページ）構造を定義します。初期状態ではルートシーンのみが設定されています。ルートシーンとはサイトにおけるトップページに相当するものです。ルートシーン以下に必要なだけ子シーンを設定して、サイト構造を定義します。



子シーンを作成するには、親シーンとなる項目（この場合はルートシーン）のアイコンをクリックしてメニューを開き、「ネストされたシーンの挿入」を選びます



作成された子シーンでは、アイコンの右側のテキストエリアを編集することでシーン名を設定することができます。また、「タイトル」欄も同様に編集でき、この内容はHTMLのtitle要素や、Webブラウザにもありますが、履歴やお気に入りの名前にも使用されます

STEP
04

シーンにキャストを挿入する

シーン構造はあくまで“器”だけを用意した状態なので、実際に画面は何も変わりません。画面を作成するにはキャストを設定する必要があります。各アイコンをクリックして、メニューから「シーンにキャストの挿入」を選び、キャスト名を設定します。「x」と「y」は、ステージの左上を基準（x:0, y:0）として、座標指定で配置したい場合に設定します。「index」は、一つのシーンに複数キャストした場合に、インデックス位置（深度）を指定するためのプロパティです。数字が大きいほど手前にキャストされます。



各アイコンをクリックして、メニューから「シーンにキャストの挿入」を選びます



すべてのシーンにキャスト名 (IndexPage、GalleryPage、ContactPage) を設定します

STEP
05

キャスト設定をFlaファイルに反映する

キャスト設定が完了したら、その内容をFlaファイルに反映させます。パネル上部の「書き出し」メニューから「キャストをシンボルとして書き出す」を選びます。すると、各キャストに対応するMovieClipシンボルが、flaファイルのライブラリに追加されます。このMovieClipシンボルが、各ページのコンテンツとなります。ただし、現段階では空っぽの状態です。次のステップでコンテンツを作成します。

以上でシーン構造の作成は完了です。パネル上部の「ファイル」メニューから「保存」を選び、プロジェクト内のdeployフォルダ内にあるeasycasting.xmlを上書きします。シーン定義やキャスト挿入など、シーンエディタ上で行った編集作業内容を保存するには、easycasting.xmlを上書き保存する必要がありますので注意して下さい。



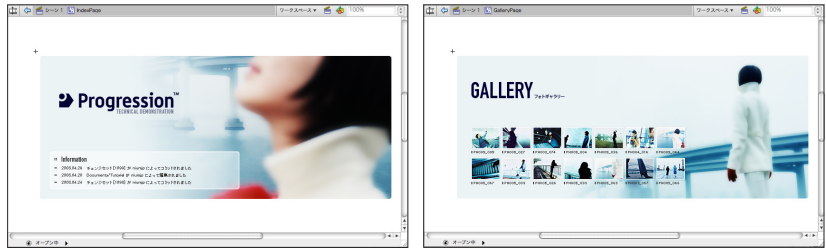
キャストをシンボルとして書き出すと、Flaファイルのライブラリパネルにキャスト名のMovieClipシンボル (IndexPage、GalleryPage、ContactPage) が追加されます。これが各ページのコンテンツを格納するシンボルとなります



キャスト設定をflaファイルに反映したら、保存して、プロジェクト内のdeployフォルダ内にあるeasycasting.xmlを上書きします

STEP 06 ページのコンテンツを作成する

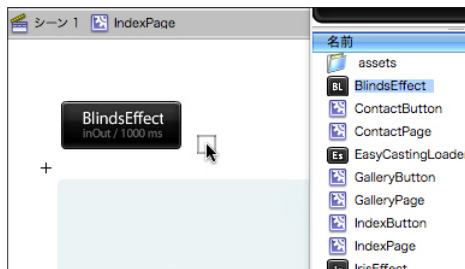
STEP05 で書き出した MovieClip シンボル (IndexPage、GalleryPage、ContactPage) は空っぽの状態です。そのシンボル内にコンテンツを追加してページを作成します。今回はテキストや写真だけのページコンテンツですが、その他にも各種シンボルやビデオなど、好きなコンテンツを追加することができます。



各 MovieClip シンボルに好きなコンテンツを追加してページを作成します

STEP 07 ページを表示する際のエフェクトを設定する

各ページが完成したら、ページを表示する、あるいは画面から削除する際のエフェクトを設定します。コンポーネントパネルから、好きなエフェクトのコンポーネントをステージにドラッグ&ドロップするだけです。エフェクト系コンポーネントは複数と同時に設定することも可能です。複数のエフェクトを設定する場合は、「TransitionGenerator」拡張機能を使うと、その効果を視覚的に確認できて便利です (次ページコラム参照)。



コンポーネントをドラッグ&ドロップする位置はどこでも構いません。設置したコンポーネントは、パブリッシュ後は自動的に (0, 0) に移動した上で非表示になります。エフェクトを変更したい場合は、設置したコンポーネントを選択して削除し、新しいコンポーネントを追加して下さい

STEP 08 メニューボタンを作成する

ページが完成したら、あとはメニューボタンです。必要な数だけボタン用の MovieClip シンボルを作成し (IndexButton、GalleryButton、ContactButton)、ボタン用のグラフィックを配置します。

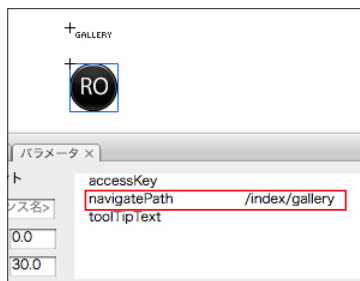


好きなボタングラフィックを配置してください

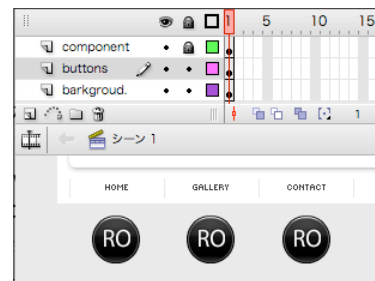
STEP 09 ボタン機能を追加する

次にボタン用の MovieClip シンボルに、コンポーネントパネルにある RollOverButton コンポーネントをドラッグ&ドロップします。そして、コンポーネントを選択した状態でパラメータパネルを開き、「navigatePath」欄に移動先を示すパスを入力します。なお、「accessKey」欄はキーボード操作、「toolTipText」欄はロールオーバー時のヒント表示に使用します。

すべてのボタンで移動先を設定した後、メインタイムラインの buttons レイヤーを選び、ステージのメニューの位置に各ボタンのシンボルを配置します。



「navigatePath」欄のパスは、スラッシュで始まり、シーン名をスラッシュ区切りでつなげたものです。たとえば、Gallery ページに移動させる場合は、「/index/gallery」となります。最後がスラッシュで終わらないように注意してください



メインタイムラインの buttons レイヤーを選び、メニューの位置にボタンシンボルを配置します

STEP
10

プリローダーを作成する

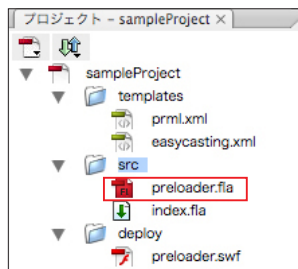
最後にswfファイルの読み込み状況を表示するプリローダーを作成しましょう。前項まではindex.fliaにすべてのコンテンツを作成していましたが、Progressionの仕様上、プリローダーのコンテンツは別ファイルに作成します。プロジェクトを作成した際に自動生成されたpreloader.fliaを使います。

preloader.fliaを開くと、タイムラインにはレイヤー、フレーム、ラベルなどがすでに設定されている状態となっています。labelレイヤーには「load」「complete」「error」の3つのラベルが設定されています。ファイルの読み込み状態に応じて、自動的にloadラベルからcompleteラベルの間をアニメーション処理するよう

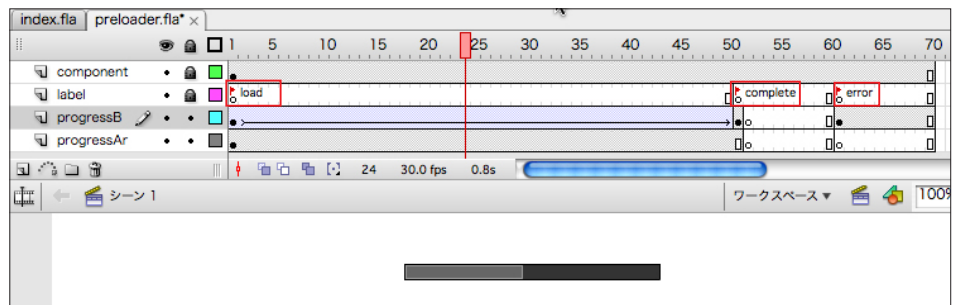
になっています。また、読み込み時に何か問題が発生した場合にはerrorラベルのコンテンツが表示されます。

今回はよくあるバータイプのプリローダーを作成します。loadラベルのフレームには「読み込み0%」の状態を、completeラベルのフレームには「読み込み100%」の状態をグラフィックで作成してトゥイーンを設定するだけです。あとは、errorラベルのフレームに読み込みに失敗した場合の画面を作成すれば完成です。

以上で、コンポーネントスタイルによるサイト作成作業は完了です。



必ず、自動生成されたpreloader.fliaを使用してください



loadラベルのフレームには「読み込み0%」の状態を、completeラベルのフレームには「読み込み100%」の状態をグラフィックで作成してトゥイーンを設定します。errorラベルのフレームには、読み込み失敗時の画面を作成します

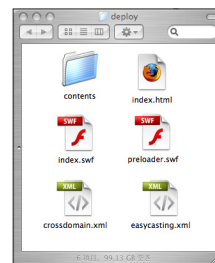
STEP
11

プロジェクトを公開する

制作作業は完了したので、あとは公開するだけです。まず、index.fliaとpreloader.fliaの両方のファイルにおいて、「Progressionプロジェクト」パネルからバブリッシュを行います。あとは、プロジェクト内のdeployフォルダの中身一式をそのままサーバにアップロードすれば公開作業は完了です。



「Progressionプロジェクト」パネルからバブリッシュしてください



プロジェクト内のdeployフォルダの中にあるすべてのファイルをサーバにアップロードします

column

複合エフェクトの効果を視覚的に確認できる「TransitionGenerator」

1つのページに複数のエフェクト系コンポーネントを設定したい人もいます。その場合に便利なのが、あつのすけ (flabaka) さんが開発した「TransitionGenerator」拡張機能です。インストールすると、FlashのメニューにTransitionGeneratorパネルが追加され、コンポーネント設置前にエフェクト効果をパネル上で確認することができます。

<http://progression.jp/ja/download/extension/>



好きなエフェクト名をチェックして、「プレビューする」ボタンをクリックすると、黒い矩形部分でその複合エフェクトが再現されます



キャストしたシーンのMovieClipシンボルを開き、「ステージに配置する」ボタンをクリックすれば、選択したコンポーネントが自動で設置されます。なお、エフェクトのActionScriptコードもパネル上に生成されるので、クラススタイルで制作する人にもおすすめです