

情報処理技法 (マルチメディアと表現)2

第8回
Flashでボタン作成
人間科学科コミュニケーション専攻
白銀 純子

Copyright (C) Junko Shiragane, Tokyo Woman's Christian University 2015. All rights reserved.

第8回の内容

- Flashでボタン作成

Copyright (C) Junko Shiragane, Tokyo Woman's Christian University 2015. All rights reserved.

Flashでボタン作成

Copyright (C) Junko Shiragane, Tokyo Woman's Christian University 2015. All rights reserved.

Flashでボタン

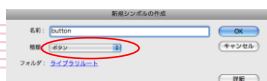
- 作成手順
 - ボタンシンボル(ボタンの形)を作成
 - ボタンシンボルをキャンバス上に配置(配置されたものがボタンになる)
 - ボタンが押された時の処理を定義
- ボタンで使う絵
 - アップ: 通常のボタンの状態
 - オーバー: マウスポインタがボタンの上に来た状態
 - ダウン: マウスがボタンの上で押された状態(押したままで離さない状態)

※「ヒット」は気にしないで良い

Copyright (C) Junko Shiragane, Tokyo Woman's Christian University 2015. All rights reserved.

ボタンシンボル作成(1)

1. 「挿入」→「新規シンボル」を選択
2. 「名前」の欄にボタンシンボルの名前を入力、
「種類」の欄で「ボタン」を選択



3. 「アップ」の状態の絵を作成
 - 2.によりタイムラインが、「アップ」・「オーバー」などのボタンモードに

Copyright (C) Junko Shiragane, Tokyo Woman's Christian University 2015. All rights reserved.

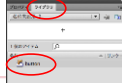
ボタンシンボル作成(2)

4. タイムラインの「オーバー」の下を選択し、その後マウス長押し→
「キーフレームの挿入」
 - 「オーバー」の絵を描画可能に
5. 「オーバー」の絵を作成
 - 「ダウン」の絵も同様にして作成
6. メニューバーの「編集」→「ドキュメントの編集」を選択
 - タイムラインがもとの形式に

Copyright (C) Junko Shiragane, Tokyo Woman's Christian University 2015. All rights reserved.

ボタンシンボル作成(3)

7. 周辺にあるウィンドウの「ライブラリ」タブをクリックすると、作成したボタンシンボルが存在



8. ボタンシンボルをキャンバス上にドラッグ&ドロップ

9. 「制御」→「再生」で動作確認

- or メニューバーの「ファイル」→「書き出し」→「ムービーの書き出し」でファイルに保存し、Safariで開く

ここで作成したものは、あくまで「ボタンシンボル」(ボタンの形)
「ボタンシンボル」を使って、実際のボタンを作成

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

ボタンの使用方法の基本

1. ボタンシンボルをキャンバス上に配置
2. ボタンシンボルを選択し、「プロパティ」ウィンドウの「インスタンス名」の欄にボタンの名前を入力
 - 半角英数で
3. レイヤを1つ作成し、ボタンシンボルの上に画面表示用の名前を配置
4. レイヤを1つ作成し、「ウィンドウ」→「アクション」で表示されるウィンドウ内にアクションスクリプトの内容を記述
 - アクションスクリプト: Flashで様々な処理をさせるためのプログラム
 - 2. でつけた名前を、アクションスクリプトでのボタンの名前として使用

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

他のページに移動

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

ページ移動の設定(1)

- 「ボタンシンボル」をキャンバス上に配置すると、ボタンを押したときの処理の設定開始

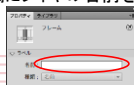
1. キャンバス上に配置したボタンを選択し、周辺にあるウィンドウの「プロパティ」タブをクリック
2. 最上部にボタンの名前を入力



Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

ページ移動の設定(2)

1. 新しいレイヤを作成し、その新しいレイヤを選択
2. プロパティウィンドウの「名前」の欄にレイヤの名前を入力(半角英数で)



3. 新しいレイヤの1つ目のフレームを選択



ここをクリック

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

ページ移動の設定(3)

4. 「ウィンドウ」→「アクション」で、表示されたウィンドウにボタンの動作を記述
 - 大文字・小文字、記号など、スペルミスしないよう注意すること

2. の手順でつけたボタンの名前

```
this.{{X}}.addEventListener(MouseEvent.CLICK, goToPage);
function goToPage(e:MouseEvent):void {
    var url:URLRequest = new URLRequest("{{Y}}");
    navigateToURL(url);
}
```

設定したいURL

※HTMLファイルに埋め込んだとき、Webに公開していない状態(「file://」のURL)で確認すると検索できないが、Webに公開した状態(「http://」のURL)では検索可能

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

ページ移動の設定(4)

5. 「**チェック**」ボタンを押すと、エラーがあるかどうかが表示
 - タイムラインが表示されているところに新しいタブが追加され、表示される
 - エラーがあれば修正する(エラーの先頭にある数字が、エラーが存在する行番号なので、その行をよく見ること)
6. 「**制御**」→「**ムービープレビュー**」→「**プレビュー**」で動作確認
 - or メニューバーの「**ファイル**」→「**書き出し**」→「**ムービーの書き出し**」でファイルに保存し、Safariで開く
 - or メニューバーの「**制御**」→「**ムービープレビュー**」→「**Device Central**」を選択する
 - 「Device Central」だと、他の方法で確認できないものも確認できることがある

※動作確認時にも、エラーが出ることもあるので、その場合はまた修正すること

画像の切り替え

画像切り替えの設定(1)

1. 1フレームごとに1枚切り替わるアニメーションを用意
 - a. キャンバスに絵を書く or 絵を貼り付ける
 - 写真などを貼り付ける場合には、そのファイルをFinderからライブラリにドラッグ&ドロップ

画像のファイルを
ドラッグ&ドロップ



- ライブラリから、画像ファイルをキャンバスにドラッグ&ドロップ
- b. タイムラインで、a. の次のフレームでマウス長押し→「**キーフレームの挿入**」
- c. a. の次のフレームの絵を書く or 絵を貼り付ける

必要なだけ繰り返す

画像切り替えの設定(2)

3. 新しいレイヤを1つ作成する
4. 3.のレイヤにボタンシンボルを1つ配置する
5. 4.のボタンをクリックし、「**プロパティ**」欄の「**名前**」の欄に「**nextBut**」(すべて半角)と入力する
 - 1つ次の画像を表示するためのボタン
6. 新しいレイヤを1つ作成する
7. 6.のレイヤにボタンシンボルを1つ配置する
8. 7.のボタンをクリックし、「**プロパティ**」欄の「**名前**」の欄に「**backBut**」(すべて半角)と入力する
 - 1つ前の画像を表示するためのボタン

画像切り替えの設定(3)

9. 新しいレイヤを1つ作成する
10. 9.のレイヤのタイムラインの1フレーム目をクリックし、「**ウィンドウ**」→「**アクション**」を選択する
11. 10.のウィンドウに、ボタンの動作を記述する

```
stop();
this.nextBut.addEventListener(MouseEvent.CLICK, showNext);
this.backBut.addEventListener(MouseEvent.CLICK, showPrev);
function showNext(e:MouseEvent):void {
    this.nextFrame();
}
function showPrev(e:MouseEvent):void {
    this.prevFrame();
}
```

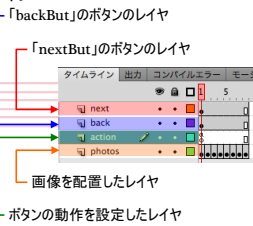
画像切り替えの設定(4)

12. 「**チェック**」ボタンを押すと、エラーがあるかどうかが表示
 - タイムラインが表示されているところに新しいタブが追加され、表示される
 - エラーがあれば修正する(エラーの先頭にある数字が、エラーが存在する行番号なので、その行をよく見ること)
13. 「**制御**」→「**ムービープレビュー**」→「**プレビュー**」で動作確認
 - or メニューバーの「**ファイル**」→「**書き出し**」→「**ムービーの書き出し**」でファイルに保存し、Safariで開く
 - or メニューバーの「**制御**」→「**ムービープレビュー**」→「**Device Central**」を選択する
 - 「Device Central」だと、他の方法で確認できないものも確認できることがある

※動作確認時にも、エラーが出ることもあるので、その場合はまた修正すること

画像切り替えの設定(5)

■ 設定後のタイムライン



アニメーションの切り替え

アニメーションの切り替え(1)

1. アニメーションのFlashのファイルを複数用意
 - 「.swf」の形式でファイルに保存しておく
2. 新しいキャンバスを作成
 - アニメーションのキャンバスよりも大きなキャンバスにすること
3. アニメーションを表示させる領域を、四角形で描画
4. 3. で描画した四角形を選択し、「修正」→「シンボルに変換」
5. 表示されたウィンドウで「ムービークリップ」を選択し、名前を入力
6. 四角形を選択し、「プロパティ」ウィンドウの「インスタンス名」の欄に「animeDisplay」と入力

アニメーションの切り替え(2)

6. ボタンシンボルを、アニメーションの個数分キャンバスに配置
7. 6. で配置したボタンを1つずつ選択し、「プロパティ」ウィンドウの「インスタンス名」の欄に名前を設定
 - 半角英数で、それぞれ違った名前にすること
8. レイヤを1つ作成し、6. で配置したボタンの上に画面表示用の名前を配置
9. レイヤを1つ作成し、新しいレイヤの1フレーム目を選択
10. 「ウィンドウ」→「アクション」で表示されたウィンドウに、アクションスクリプトの内容を記述

アニメーションの切り替え(3)

■ アクションスクリプトの内容

```
var animeLoader = new Loader();
anime.x = animeDisplay.x;
anime.y = animeDisplay.y;
removeChild(animeDisplay);

function animeLoad(e) {
    if(anime.root) {
        removeChild(anime);
        removeChild(animeDisplay);
    }
    e.target.content.stop();
}

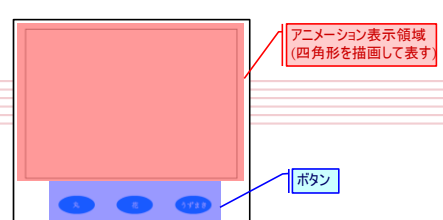
function animeStart(e) {
    addChild(anime);
    anime.mask = animeDisplay;
    e.target.content.play();
}

var animeListDictionary = new Dictionary();
animeList[1つのボタンの名前] = アニメーションFlashのファイル名1;
animeList[2つのボタンの名前] = アニメーションFlashのファイル名2;
animeList[3つのボタンの名前] = アニメーションFlashのファイル名3;

function loadContent(e) {
    if(animeList[e.target]) {
        anime.load(new URLRequest(animeList[e.target]));
    }
    anime.contentLoaderInfo.addEventListener(Event.INIT, animeLoad);
    anime.contentLoaderInfo.addEventListener(Event.COMPLETE, animeStart);
    addEventListener(MouseEvent.CLICK, loadContent);
}
```

アニメーションの切り替え(4)

■ 1.～10. まで行った後のキャンバスの状態



アニメーションの切り替え(5)

- 「**チェック**」ボタンを押すと、エラーがあるかどうかが表示
 - タイムラインが表示されているところに新しいタブが追加され、表示される
 - エラーがあれば修正する(エラーの先頭にある数字が、エラーが存在する行番号なので、その行をよく見る)
- 一旦「.fla」形式でファイルに保存し、「**制御**」→「**ムービープレビュー**」で動作確認
 - アニメーションのFlashのファイルと同じフォルダに保存すること
 - 動作確認時にも、エラーが出ることもあるので、その場合はまた修正すること

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

検索ボックス

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

検索ボックス(1)

- キーワードを入力し、「**検索**」ボタンを押すと、Googleでの検索結果を表示するFlash
- 検索キーワードの入力フィールドと検索ボタンを配置するレイヤを1つ作成
- 「**ウィンドウ**」→「**コンポーネント**」を選択
- 表示されたウィンドウで、「**User Interface**」をクリックし、選択枝から「**TextInput**」をキャンバス上にドラッグ&ドロップ
- 入力フィールドの隣にボタンを配置し、ボタンに画面表示用の名前を配置
- ボタンを選択し「**プロパティ**」ウィンドウの「**インスタンス名**」の欄に「**searchText**」と入力
 - 半角英数で入力すること

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

検索ボックス(2)

- レイヤを1つ作成し、1つ目のフレームを選択
- 「**ウィンドウ**」→「**アクション**」で表示されたウィンドウに、アクションスクリプトの内容を記述

```
this.searchBut.addListener(MouseEvent.CLICK, searchGoogle);

function searchGoogle(e:MouseEvent):void {
    var googleSiteString = "http://www.google.co.jp/search?q=" + this.searchText.text;
    var urlURLRequest = new URLRequest(googleSite);
    navigateToURL(url);
}
```

※HTMLファイルに埋め込んだとき、Webに公開していない状態(「file://～」のURL)で確認すると検索できないが、Webに公開した状態(「http://～」のURL)では検索可能

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

FlashのWebページへの掲載

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.

FlashのWebページへの掲載

- 以下のタグをHTMLファイルに書く
 - HTMLファイルとFlashのファイルは同じフォルダ内に置く
 - Flashのファイルは「.swf」のファイルを使う

```
<object width="横幅の大きさ" height="縦幅の大きさ">
  <param name="movie" value="Flashファイル名">
  <param name="quality" value="high">
  <embed src="Flashファイル名" quality="high"
    width="横幅の大きさ" height="縦幅の大きさ" type="application/x-shockwave-flash">
  </embed>
</object>
```

Copyright (C) Junko Shimogawa, Tokyo Woman's Christian University 2015. All rights reserved.