

情報文化演習
重定如彦先生

2011年1月12日

卒業制作論文
携帯電話の待ち受けFLASH
(待ち受けペット)

法政大学国際文化学部国際文化学科

07G0418

畑部朝子

-目次-

1.	制作背景	3
1-1	はじめに.....	3
1-2	制作動機.....	3
1-3	先行例.....	4
2.	開発環境	7
2-1	Flash Lite	8
2-2	ActionScript2.0.....	8
3.	作品の解説	9
3-1	概要	9
3-2	特徴	10
3-3	ペットのムービークリップの構造	10
3-4	制作手順.....	11
4.	考察	23
4-1	感想	23
4-2	反省	23
4-3	課題点.....	24
5.	参考文献、URL	26

1. 制作背景

1-1 はじめに

この論文では、私の卒業制作のテーマである「携帯電話の待ち受けFLASH」というアプリケーションについて論じていく。

現在、世の中には携帯電話用だけでも様々なアプリが存在する。フリーソフトとして手軽に入手可能な物や、販売されているものなど違いはあれども、携帯電話を使用するにあたって便利な機能を増やしてくれるものである。しかし一部のアプリは使用する携帯のキャリアによって使用不可能なものがある。そこで私はキャリアに左右されず、FLASHに対応しているものならどの携帯電話でも使用可能なアプリを開発することにした。プログラミング初心者の自分の実現力に限界はあるが、既存のアプリを参考に、オリジナルの作品を作ろうと考えた。

また、ここでは筆者のキャリア以外の携帯電話という言葉を使用するが、それは私の使用している携帯電話の会社であるauとは違うものを指すことにする。

1-2 制作動機

携帯電話市場のシェア争いは近年に始まったことではないが、最近では各メーカーによるスマートフォンの発表などを始めとし、ユーザーの選択肢はますます広がってきた。

株式会社stratationの調査によると携帯電話を選ぶ基準として最も多いのが、「デザイン・形状（88.2%）」で、次いで「色（81.3%）」となり、機能（45.2%）、メーカー（40.7%）、性能（38.2%）よりビジュアルを重視することが分かった。

(http://mmd.up-date.ne.jp/news/detail.php?news_id=83 2007年2月現在)しかし同文章から使用料金の高さや反応スピードの遅さ、バッテリーの持続性の短さなど、見た目以外への不満の声も多く、自分の携帯電話に100%満足しているユーザーは少ないと言える。私自身、自分の携帯電話に対する不満は多々あるが、その中でも自分のキャリアに対応していないサービスやアプリケーションなどに対する不満に自分で実現可能な改善策があるのではないかと考えた。要するに、**自作で携帯電話のキャリアに左右されないアプリを作成**すれば良いのである。

また、私が卒業制作の内容を決める条件として、

- (1) 最後まで完成させる為に楽しみながら制作に当たれること
- (2) 他人にも実際に使いたいと思ってもらえる物を制作すること
- (3) プログラムとアニメーションを駆使してFlashやシステムの知識と高度な表現力を

身につけること

を考えた。

上記で述べた自分ができる携帯電話への改善と制作への前提条件、さらに自分の能力、性格を考慮した結果、携帯電話のアプリという概念に至った。そして、実際に筆者のキャリア以外の携帯電話の待ち受け画面でペットが飼えるサービスを目にし、それを参考に待ち受け画面で遊ぶ育成ゲームを制作することにした。この作品にすれば、手本があるので自分の能力に見合った制作が可能となり、結果的にFlashやプログラミングの勉強に繋がる。また、webページの様にユーザーが受身となるコンテンツではなく、ゲームという遊び手のアクションに反応するものを制作することによって、他人にも使用したいと思わせるものが制作できるのではないかと考えた。

尚、このアプリの利点は

- (1) Flashに対応した携帯電話であればキャリアに左右されずに遊べること
- (2) 待ち受け画面上でペットを育成する為、特別にアプリを起動する時間を省けるので手軽に遊べること
- (3) 携帯電話という常に持ち歩いている物をデバイスにすることによって、いつでも使用可能であり、ゲームを続けていきやすいこと

の3点である。

1-3 先行例

ここでは先行例として携帯電話をプラットフォームとしている育成アプリに類似したものを2点紹介する。

マチキャラ

月額210円でドコモが提供しているサービス機能の一つである。携帯電話を便利で使いやすいように自分好みにカスタマイズするコンテンツの一つであり、待ち受け画面やメール画面上に自由に動き回ることができるキャラクターを設定することができる。キャラはメール受信や不在着信などを知らせてくれる。さらに、ケータイが「執事」や「コンシェルジュ」のように、ユーザーの生活をサポートするサービスである。i コンシェル対応機種の場合は、設定したキャラクターが情報を伝えてくれる。その情報は例えば鉄道の運行状況や気象情報、イベント情報など様々である。その他、スポーツの試合日程やアーティストの活動情報など、ユーザーの気になる予定をまとめてダウンロードし、スケジュール帳に書き込み最新の情報に更新するなどのスケジュール管理機能やイベントスケジュールや画像をメンバー同士で共有できる機能などがある。さらにマチキャラアップデート対応マチキャラを設定している場合は、マチキャラが自動的に成長したり変身したりする。当

然ながらドコモのサービスであるので、残念なことにドコモユーザー以外は使用できない。
尚このサービスは私がこの作品を制作しようと思ったきっかけである。



(NTT ドコモのホームページより転載。マチキャラが着信をお知らせしている。

<http://www.nttdocomo.co.jp/service/customize/machichara/#notice01>)

🚩 Dogz 3D

【概要】 携帯電話用のペット育成ゲーム

【開発元】 ゲームロフト株式会社

【課金体系】 従量 315 円

【容量】 961K

【通信機能】 なし

【対応機種】 softbank:3G 端末、ドコモ:FOMA903i 以上、703i、au:BREW2.1 以上

携帯電話のアプリであり、犬を育成して遊ぶゲーム。最初に飼う犬種を選択する。

撫でたり、エサを与えたり、散歩に連れていったりなど世話をする必要はあるが、ボタンを押して自動で終わるものではなく、プレイヤーは携帯電話を操作して直接の操作で世話をを行う。世話を怠るとペットの機嫌が悪くなったり、清潔度が下がったりするのでマメにアプリを立ち上げておく必要がある。ミニゲームや家の庭だけでなく、プールや海など様々な散歩先の発生など幅広い遊び方ができる。また、フル3Dの為、視点を自由に変更して犬の様々な表情を見ることができるのも特徴の一つである。

私個人的には待ち受け画面上で遊ぶことができればさらに遊びやすくなるのではないかと思う。



(ゲームロフトホームページ <http://www.gameloftjapan.com/>より)

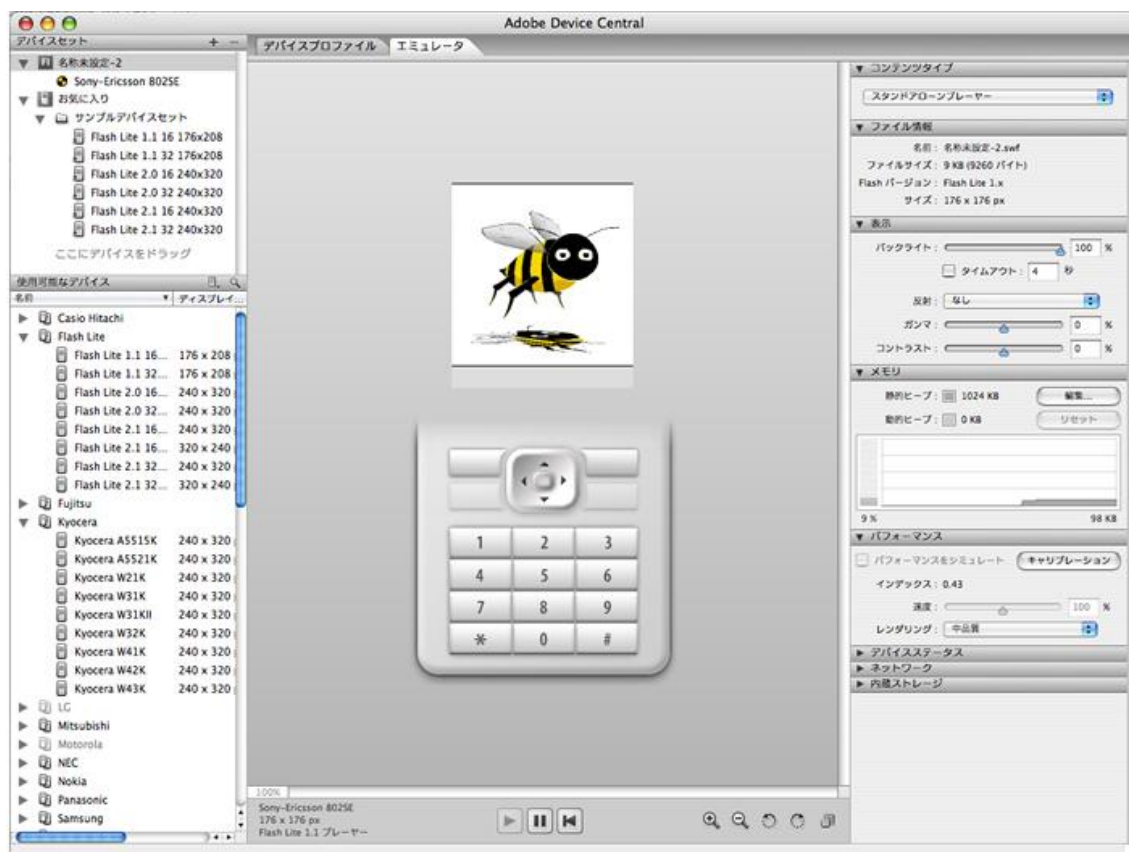
2. 開発環境

このアプリはFlash Lite 2.0に対応した携帯電話で使用することを前提とする。これにより、NTT DoCoMo、au、ソフトバンクモバイル、Willcomの4キャリアで対応できる。実際に使用したツールはAdobeのFlash CS4で、プログラミング言語はActionScript 2.0である。Adobe FlashにはDevice Centralが内蔵されており、これを使用するとコンピューターのデスクトップ上で開発途中のアプリのシミュレーションが可能になる。

ソフト：Adobe Flash CS4

OS：Windows XP Home Edition

メモリ：367KB



デバイスセントラル (Adobe Flash CS3)

画面左側からFlashを再生する携帯電話の機種を選択し、再生ボタンや一時停止ボタンで作成したファイルがどのように見えるかシミュレーションできる。

2-1 Flash Lite

携帯端末向けのFlashプレーヤーであり、Flash Lite2.0ではActionScript 2.0をサポートしている。元々Web用に開発されたFlashプレーヤーを軽量化し、一部の機能を限定したものである。ActionScriptの最新版はActionScript 3であるため、ActionScript 2の主な用途はFlash Lite 2対応携帯端末向けとなる。

2-2 ActionScript2.0

ActionScriptはFlashに搭載されているコンテンツの再生などを制御するプログラミング言語である。これをFlashの作業スペースであるステージ上に描画した※1ベクター画像に組み込むことにより、コンテンツにより複雑なアニメーションや機能を付加することが可能になり、自由な表現ができる。また、ActionScript2.0はFlash 7、Flash Lite2以降で実行可能なオブジェクト指向のスクリプト言語であり、ActionScript2.0以前のバージョンには無かった機能もいくつか存在する。例えば、XMLデータの読み込み、表示する度に初期化されていたコンテンツのデータの維持、外部画像(透明度が設定されたGIFやPNG、JPEG)の読み込みなどである。ActionScript1.0と比較して最も大きな違いは、※2クラスをより意識した開発ができることである。

※1ベクター画像

ベクター画像とは、点の集まりではなく、直線や曲線を数学的に表すベクタデータを用いて描画した画像のことである。因みに点の集まりによって描画された画像やフォントはラスタグラフィックスと呼ばれる。ラスタグラフィックスは拡大すると輪郭がギザギザになり、画像の粗さがハッキリしてしまうが、ベクター画像は輪郭がはっきりとしている。その為、図形やフォントの描画などの表示に向いている。

※2クラス

クラスとはオブジェクト指向において、データとその操作手順であるメソッドをまとめたものである。オブジェクトとはある機能を持つプログラムを部品の様にあらかじめ纏めたもので、それらを組み合わせてプログラムを作ることができる。つまりクラスはオブジェクトを表現するプログラム自体、あるいは同じ特性を持つオブジェクトの集合に名前を付けたものである。例えば、ある会社で働いている人をオブジェクトと考え、その会社で働いている人の集合を「従業員」というクラスにすることができる。また、クラスを作る上で重要なのは、誰でも使えるということである。

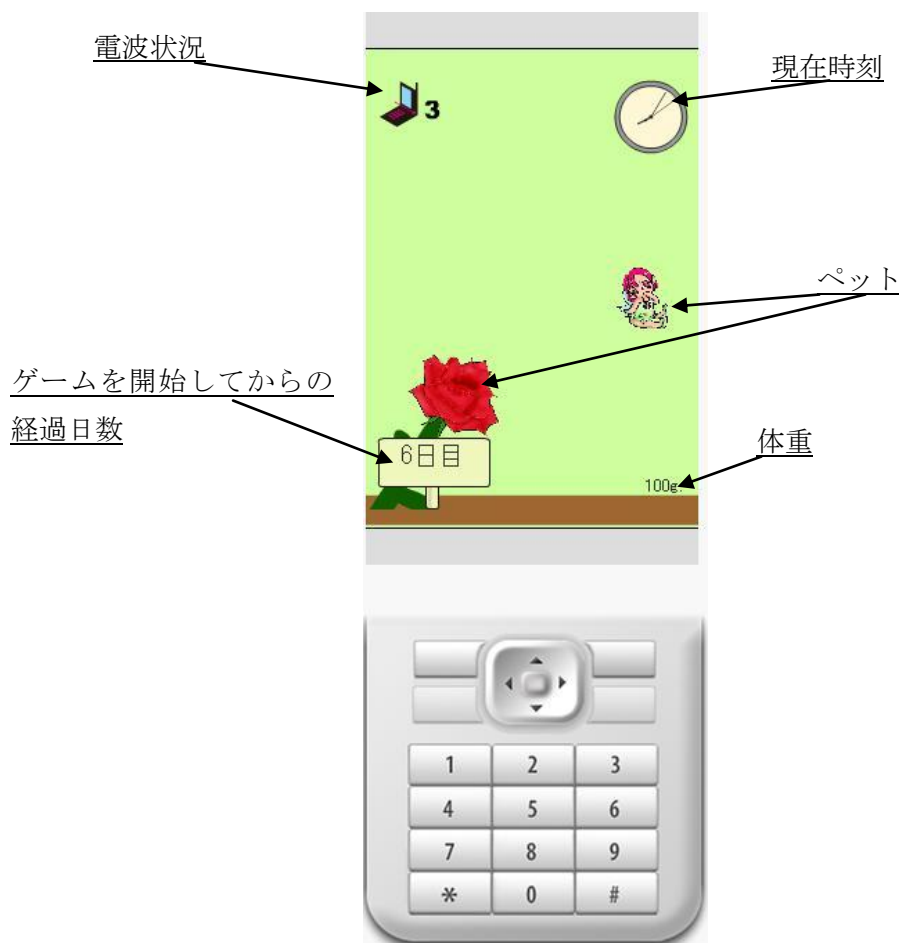
3. 作品の解説

3-1 概要

このFlashは携帯電話の待ち受け画面でペット（花の妖精）を飼う育成ゲームである。とはいっても、この作品は0から私が開発したものではなく、Flashの技術を教える学校を運営する株式会社クスール (<http://www.cshool.jp/>) が作成した「携帯ペット」を手本に作成した。

完成したswfファイルを携帯の待ち受け画像に設定した時点でゲーム開始となる。時間の経過に伴い、花からペットが生まれ、成長していく。また、ペットは17日目以降成長しないので、姿は変化せずに経過日数のみが経過する仕様となっている。

メイン画面



3-2 特徴

- (1) ユーザの操作によってペットの体重が変わり、それに伴い、ペットの表示サイズが変わる。現段階では体重が増えると縦横のサイズが体重の分だけ大きくなる。
- (2) 実際に体重を増やす方法だが、待ち受け画面で使用する場合、Flash側では携帯電話の全てのキー操作を取得することができない。よって、どのようにしてエサを与えるかということが問題であるが、この作品では充電器を使用するという行為でエサをあげるという行為を代用することにした。つまり、携帯電話の充電をするとペットの体重が増加し、それに伴い、ペットの表示サイズが拡大されるという仕組みである。尚、エサを与えることができるのは朝7時～9時の2時間である。
- (3) ゲーム開始7日目以降は体重によって成長過程に変化があらわれ、ユーザーの好みによってペットの姿をある程度カスタマイズできる楽しみがある。例えば、成長途中で体重が軽すぎると成長途中からペットが不健康そうな外見になったり、逆に体重が増えすぎると最終形態で豊満な体型になったりする。
- (4) 待ち受け画面上に設定されたペットの成長を見ているだけでは面白みに欠けるので、携帯電話の電波状況の強弱によってペットが異なる動きをする。圏外の場合と電波が1本、2本、3本の時に少しずつ違うアニメーションを流すという仕組みである。

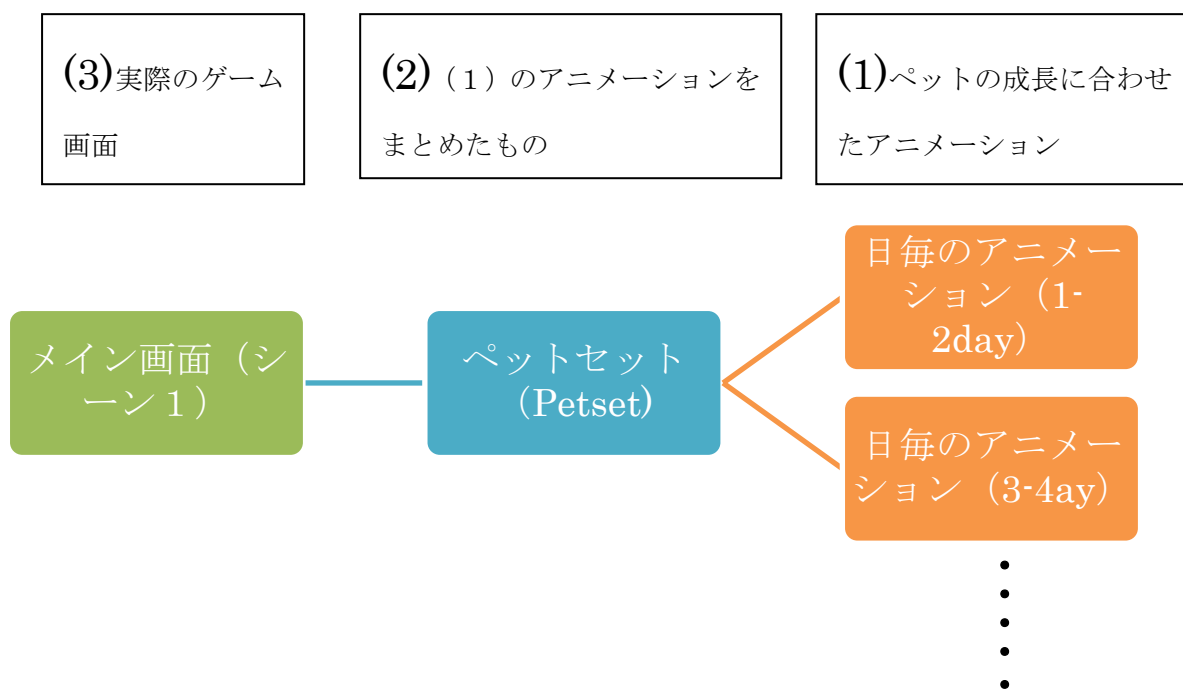
3-3 ペットのムービークリップの構造

この作品のペットのムービークリップは大きく分けて3つの階層から構成されている。ムービークリップとは、アニメーションのついたシンボルのようなもので、flashの中に組み込むことによってより豊かな表現が可能になる、部品の様なものである。本作品はFlashで作成するアニメーションの中にこのムービークリップというアニメーションを入れ子の様に組み込んで作成する階層構造となっている。

まず、ペットの各成長段階によって変化するアニメーションの階層。次に成長段階別のアニメーションを集合させ、時系列順にまとめたペットセットのアニメーションの階層。そして時計や背景、ペットセットを表示する為のメイン画面の階層である。全て一つの階層に纏めることも可能だが、その際、ペットのアニメーションを管理するフレームだけが異常に長くなり、作業がし辛くなる上に、修正する度に他のフレームにも影響が出るので、

一つ一つの部品を分けて制作していく。

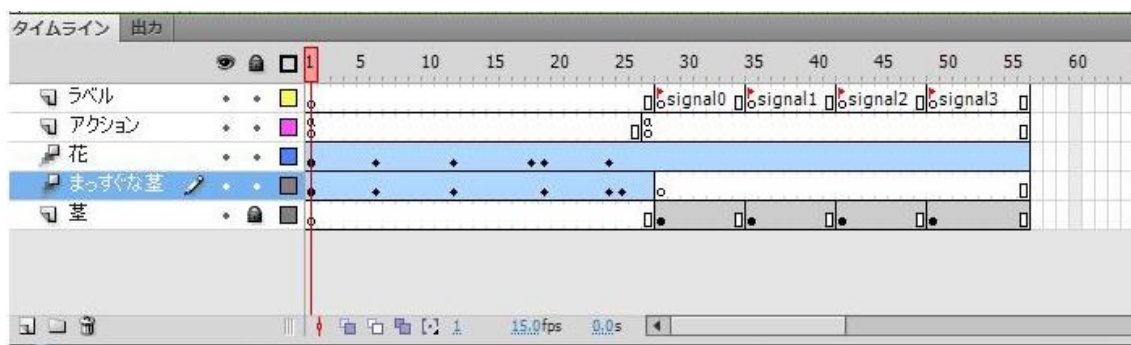
図で表すと以下のようになっている。



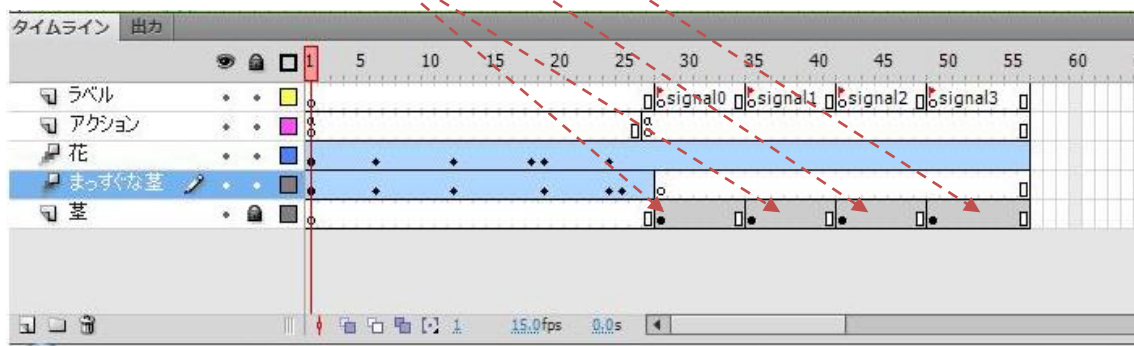
3-4 制作手順

(1) 日毎のアニメーション (1-2 day、3-4 day、5-6 day、… 15-17 dayのような最下層のアニメーション)

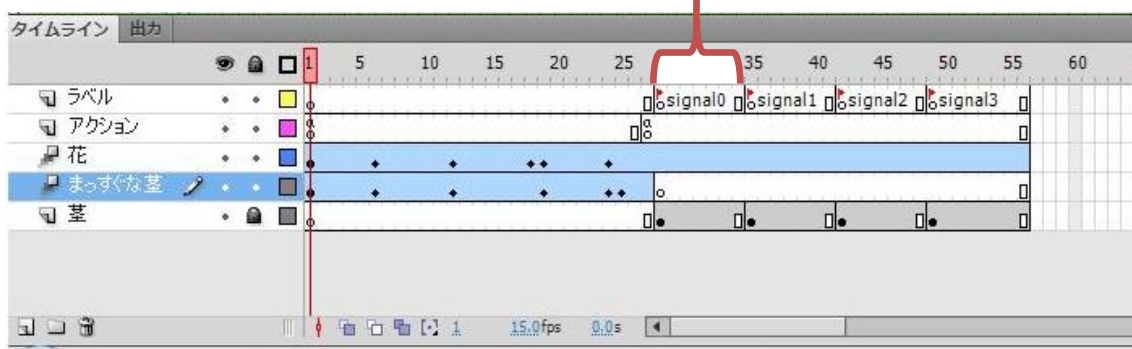
1. ペットの成長段階に合わせた画像を描画し (使用するソフトは何でも良い)、Flashに読み込む。新規でレイヤーを作成し、ペットレイヤー (画像では「花」、「まっすぐな茎」「茎」というレイヤーに相当する) など分かりやすい名前をつける。



- 読みこんだ素材をメニューバーの変換からシンボルに変換し、タイムラインを選択してモーショントゥインの挿入などを行い、好きなアニメーションを付ける。携帯電話はメモリサイズが制限されている為、なるべく小さいサイズのアニメーションにすると良い。
- 2で作成したアニメーションのタイムラインの終わりに、作業ステージ上にペットの画像がない状態にする為に空白のフレーム（時間の流れを制御するコマ。映画フィルムの1コマ1コマに相当する。15フレームで1秒を表わしたり、24フレームで1秒を表わしたりする）を作成する。これによって最初のアニメーションが終了する。その後ろに電波が0本の時のアニメーションを作成する。同様に、その後ろに空白のキーフレームを挿入し、電波1本、2本、3本の時のアニメーションを作成する。k



- ラベルレイヤーを作成し、その下にあるペットレイヤーの電波0本のアニメーション開始フレームから終了フレーム部分に相当するラベルレイヤーの場所に「signal0」という名前を付ける。これをラベルを付けるという。これにより、電波が0本しか立っていない時（圏外）に再生するターゲット（アニメーション）を指定できる。



- ラベルレイヤーで電波1本と2本と3本のアニメーションを制御するフレームにも同様にラベルを付ける。(signal1,signal2,signal3など) 後程、このアニメーションを呼び出す設定をする。
- アニメーションを繰り返し再生する為、アクションレイヤーを作成し、actionパネル

を開いて以下のActionscriptを記述する。

- ```
① stop();
② _root.initView();
③ fscommand2("StopVibrate");
```

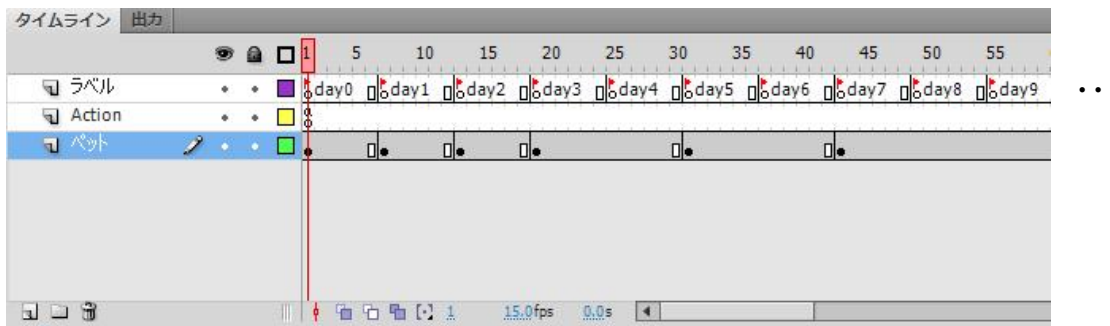
- ①ここでアニメーションの再生を止める
- ②アニメーションが終わり次第この関数（init関数。手順（3）で説明する）を呼ぶことで、電波状況などを反映した壁紙になる
- ③携帯電話のバイブレータを停止する。

7. day 1 など分かりやすい名前を付け、ムービーして保存する。
8. この手順を成長段階分と体重による分岐分繰り返す。

## (2) ペットセット

1. daysレイヤーとペットレイヤーを作成する。

daysレイヤー上のタイムラインを0日目～17日目まで区切り、「day0」など分かりやすいラベル名を付ける。

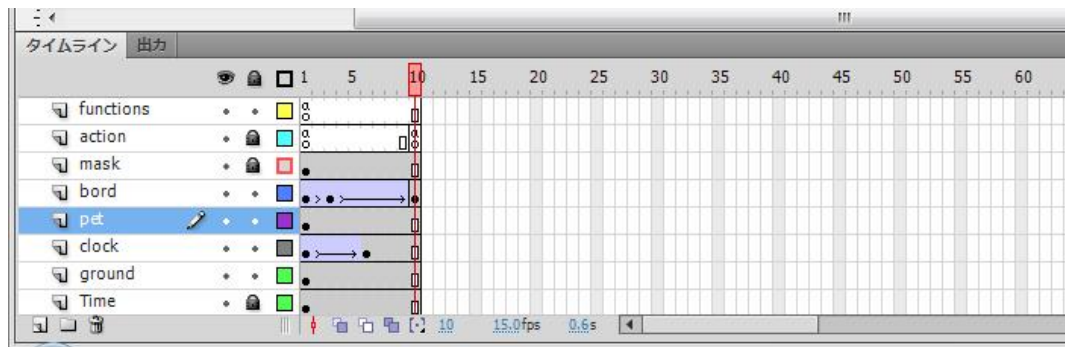


2. (1)で作成した成長段階毎のペットのムービークリップをdaysレイヤーの区切りに合わせてペットレイヤーに配置する。
3. 同様に痩せた場合と太った場合の配置も行う。
4. 全て配置し終わったらムービーとして出力する。これで「ペット」としての部品は完成した。

### (3) メイン画面

ここからは出来た部品をメイン画面に配置し、それを上手く動かす為のプログラミングを行っていく。

1. 一番上にaction script記述用のactionレイヤーを作成し、その下に背景、時計など他に必要なコンテンツの分だけレイヤーを作成する。
2. 作成したレイヤーに時計などの部品を配置する。尚、ペットレイヤーには(2)で作成したペットセットのムービークリップを配置する。



3. ペットの成長を記録する為の Shared Object の設定を行う。

Shared Object とは Flash Player がユーザーのマシン（ローカル）にデータを保存するものである。これによって複数のムービークリップやオブジェクト間でデータを共有できる。Shared Object の用途には、web ページでユーザーが入力した情報を記憶したり、web 上でゲームを公開したとして、そのゲームのスコア（点数）を保存して、再び表示したりと様々なものがある。ここでは待ち受け画面上にこのムービーを設定した日時や体重を保存する為に使用する。

```
① function init(){
② my_so = SharedObject.getLocal("grow");
③ SharedObject.addListener("grow", onCheckDate);
④ _quality = "HIGH";
⑤ petSet_mc.setMask(mask_mc);
}
```

- ① **init**関数の定義をする。
- ② **SharedObject**を使用するには、初期化しなければならない為、この初期化のスク립トを入れる。「**grow**」という名前の**SharedObject**を**getLocal**メソッドを使って新しく作成する。書き方は「任意の変数名 = **SharedObject.getLocal**("好きな名前");」この場合、変数名(**my\_so**)というのはセーブデータに該当するSQLファイルのことを指す。これにより、クライアントのコンピューター側に**grow.sol**というセーブデータが作成される。
- ③ **onCheckDate**をイベントリスナーに登録するの為のスク립ト。イベントリスナーとは何らかのイベントが発生した場合に呼ばれるメソッドを持つクラスである。「アニメーションが再生された」「マウスがクリックされた」といった「イベント」が発生する場合、そのイベントに対応する処理をするためのメソッドを用意する必要がある。そのメソッドを持つクラスが「イベントリスナー」である。**SharedObject**からデータがダウンロードされ、準備ができるとイベント（次の動作が起こるきっかけ）が発生する。（ ）内は引数であり、前者は**SharedObject**の名前、後者はイベントを受け取るハンドラ名である。
- ④ ファイルが重い場合、携帯端末にかかる負荷が大きすぎて画像が粗くなる場合があるので、パブリッシュ（Flashドキュメントから再生用のムービーなどを作成すること）の際、画像の表示レベルを高で設定する。必要なければ省略可。
- ⑤ 作成した**petSet**のムービークリップを読み込み、マスクをかける。マスクとはマスクレイヤーの下にあるオブジェクトをマスクレイヤーにある塗りの部分だけ表示

させる機能である。例えばカーテンを上げていきだんだんと人が現れるようにしたり、暗闇の中へスポットライトを当て中にあるものを照らしたりする事もできる。ここではペットの塗りのある部分だけを表示し、ペットセットの背景（何もない部分）を非表示にする。

4. 日付を確認するプログラム（onCheckDate関数）の設定をする。このプログラムの目的はSharedObjectを使用して携帯端末にプレイ時間や日付のデータを保存することである。

```
//日付チェック
function onCheckDate(){
 ① if(my_so.data.startDate == undefined){
 ② my_so.data.startDate = new Date().getTime();
 ③ my_so.data.growingPoints = 100;
 ④ my_so.flush();
 }

 ⑤ startDate = my_so.data.startDate;
 ⑥ today = new Date().getTime();
 ⑦ period = new Date (today - startDate);
 ⑧ totalDays = Math.floor(Math.floor(period.getTime()/1000)/24/60/60);
 ⑨ selectGrowingPet(totalDays);
}
```

- ① 初めてゲームを開始したかどうかをチェックし、初めてなら②～④のプログラムを実行する。
- ② 初期値設定 開始日の入力
- ③ 体重に関するパラメータを初期設定（ここでは100g）
- ④ flushメソッドを使用してSharedObjectへデータを書き込む
- ⑤ ペットの育成開始日を取得
- ⑥ 今日の日付を取得
- ⑦ 今日から開始日を引き算して得られたトータル時間をDateオブジェクトに代入
- ⑧ ペットを育て始めてから何日経過したかのミリ秒を元に日付を計算。計算結果



の値をtotalDaysに切り捨てて代入する。

⑨ 取得したトータル日数に合わせたペットのアニメーションを選択する。

5. ペットの成長に合わせて（1）で作成した特定のムービークリップの再生をする設定をする。

```
① function selectGrowingPet(num:Number){
② if (num >= 7 && weight < 110) {
③ if(num<17){
④ petSet_mc.gotoAndStop("dayyase"+num);
⑤ }
⑥ else {
⑦ petSet_mc.gotoAndStop("dayyase17");
⑧ }
⑨ }
⑩ else if(num >=15 && weight > 130){
⑪ petSet_mc.gotoAndStop("daykoe17");
⑫ }
⑬ else if(num > 17){
⑭ petSet_mc.gotoAndStop("day17");
⑮ }
⑯ else{
⑰ petSet_mc.gotoAndStop("day"+num);
⑱ }
⑲ board_mc.days_txt.text = 1 + num + "日目";
⑳ getFeed();
㉑ }
```

① ペットの成長を表示する関数を定義する。

② もし日付が7日以降で体重が110以下で

③ かつ17日未満なら④を実行する。

④ petSetのムービークリップのdayyaseフレーム（体重が軽い場合のアニメーションを制御するフレーム）に飛んで再生して止める。numとはゲームのプレイ日数

であるが、今後はdayyaseにnumをプラスしたフレームに飛んで再生する。

- ⑤ それ以外（17日以降で110グラム以下）ならpetSetのムービークリップのdayyase17フレームに飛んでアニメーションを再生して止める。
- ⑥ あるいは日付が15日以上で体重が130以上なら
- ⑦ petSetのムービークリップのdaykoe17フレーム（体重が重い場合のアニメーションを制御するフレーム）に飛んで再生して止める。17日以降は変化しない。
- ⑧ あるいは日付が17日以上（で体重が110グラム～130グラム）なら⑨を実行する。
- ⑨ petSetのムービークリップのday17フレームに飛んで再生して、止める
- ⑩ それ以外なら
- ⑪ petSetのムービークリップのdayフレームに日付を足したフレームに飛んで再生して、止める
- ⑫ boardのムービークリップのテキストボックスに育て始めてからの日数を表示
- ⑬ getFeed関数（手順9で定義する。エサをあげる為の関数）の実行

つまり、やせた場合は7日以降、太った場合は15日以降に外見が変化し、17日以降は成長が止まる。このペットの成長に合わせた分岐を表でまとめると以下のようなになる

| ゲームのプレイ日数 | 体重           | 再生するアニメーション                                            |
|-----------|--------------|--------------------------------------------------------|
| 1～6日目     | 条件なし         | day1～day6フレーム                                          |
| 7日目以降     | 110g未満       | dayyaseフレーム～（痩せた妖精のアニメーション）以降、yaseとついたフレームを再生する（痩せルート） |
| 7日目～15日目  | 110g以上       | day7フレーム～day15フレーム～                                    |
| 15日目以降    | 130g以上       | daykoeフレーム～（肥えた妖精のアニメーション）以降、koeとついたフレームを再生する（肥えルート）   |
| 17日目以降    | 111g以上130g未満 | day17フレームを再生。それ以降進まない                                  |

## 6. 時計の計算（setTime0関数）

ここからはfscommand2を使用してプログラムを書く。

fscommand2とは携帯端末対応用にflash Liteを使用する際のActionScriptの関数である。

fscommand2 コマンドを使用し、flash lite プレーヤーを通して以下のような情報を携帯端末から取得したり制御することが可能である。

- 現在の年月日時の情報取得
- バッテリー残量状況の取得
- 電波受信状態の取得
- バイブレータ機能制御 etc...

この手順ではメイン画面にアナログ時計を作成する。上記で説明したfscommand2を使用して時間、分、秒を取得し、取得した値をhours,minutes,secondsのそれぞれの変数に割り当てる。そして作成した時計の針に割り当てる。

アナログ時計を作るプログラム。ただし、待ち受け画面上に時計が不要な場合は省略可能。

```
function setTime(){
① var hours:Number = fscommand2("GetTimeHours");
② var minutes:Number = fscommand2("GetTimeMinutes");
③ var seconds:Number = fscommand2("GetTimeSeconds");/
④ clock_mc.hoursHand_mc._rotation = hours*30+(minutes/2);
⑤ clock_mc.minHand_mc._rotation = minutes*6+(seconds/10);
⑥ clock_mc.secHand_mc._rotation = seconds*6;
```

- ① 時間を取得。値をhoursに代入。
- ② 分を取得。値をminutesに代入。
- ③ 秒を取得。値をsecondsに代入。
- ④ 時間の針（短針）を一分毎に回す計算。時計の針は360度の盤面を回転するので、一時間当たりの短針の回転角度は、取得した時間に360を12（時間）で割った数値（ $360 \div 12 = 30$ ）を掛けたものになる。しかし、これだけでは1時間に毎にしか短針が移動しない。なので、360を12で割った1時間ごとの角度にプラスして、現在の分数（minutes）分も動かすようにする。短針は一時間当たりの回転角度である30度を60分かけて回転するので、 $30 \div 60 = 0.5$ となり、これが短針が一分当たりに回転する角度である。従って、短針が1時間で回転する角度に1分で回転する角度を足したものが現在の短針の角度となる。
- ⑤ 分の針（長針）が一分毎に回転する角度（ $360 \div 60 = 6$ ）を計算する。60秒で6度なので1秒では0.1度（ $6 \div 60 = 0.1$ ）回転する。
- ⑥ 秒針が一秒ごとに回転する角度を計算し、その分回転させる。（進ませる）

## 7. 曜日の取得計算

この手順では現在の曜日を取得し、表示するプログラムを作成する。曜日を表示したくない時は省略しても良い。

```
function setWeekday(){
① var weekday:Number = fscommand2("GetDateWeekday");
② switch(weekday){
③ case 0:weekday_txt.text = "SUN";break;
 case 1:weekday_txt.text = "MON";break;
 case 2:weekday_txt.text = "TUE";break;
 case 3:weekday_txt.text = "WED";break;
 case 4:weekday_txt.text = "THR";break;
 case 5:weekday_txt.text = "FRI";break;
 case 6:weekday_txt.text = "SAT";break;
 }
}
```

- ① 曜日をweekday変数に入れる。曜日は日曜=0、月曜=1、…のように0から6までの数字で表現される。
- ② 曜日表示の分岐文
- ③ 値が0の場合はweekdayのテキストボックスにSUNという表示をする。  
以下同様。break;で各caseの処理は終了する。

8. 電波状況の取得、アニメーションの設定を行う。

電波状況に合わせたアニメーションを再生する為の設定をする。

```
function setSignal(){
① var maxSignal:Number = fscommand2("GetMaxSignalLevel");
② var currentSignal:Number = fscommand2("GetSignalLevel");
③ var signalLevel:Number = Math.floor(currentSignal/maxSignal*3);
④ signal_txt.text = signalLevel;
⑤ petSet_mc.pet_mc.gotoAndStop("signal"+signalLevel);
}
```

- ①電波状況 MAX を取得。maxSignal 変数に入れる。  
②現在の電波状況を取得。currentSignal 変数に入れる。  
③電波状況を 3 段階+圏外に計算。signalLevel 変数に入れる。  
④電波状況の電波数を表示する。  
⑤電波状況によって再生するアニメーションを変更する。

9. 充電しているかのチェックを行うプログラムの設定を行う。ここでエサを与えることができるようにする。

```
function getFeed(){
① var hours:Number = fscommand2("GetTimeHours");
② var powersource = fscommand2("GetPowerSource");
③ if(hours > 7 && hours < 12 && powersource == 1){
④ my_so.data.growingPoints++;
⑤ my_so.flush();
 }
⑥ var weight = my_so.data.growingPoints;
⑦ petSet_mc._xscale = petSet_mc._yscale = weight;
⑧ weight_txt.text = weight +"g."
}
```

- ①時間を取得し、値を**hours**変数に入れる。
- ②バッテリーを電源としているか、外部電源を使用しているかを調べる。
- ③もし朝7時から9時までの間に充電していたら④と⑤を実行する。
- ④**sharedObject**の体重を表わすプロパティである**growingPoints**に+ 1 する。つまり、ここでペットの体重が 1 グラム増える。
- ⑤**sharedObject**にデータを書き込む。
- ⑥変数**weight**はセーブデータである**SOL**ファイルの体重を表わす**growingPoints**というプロパティである。
- ⑦体重が増えるとそれに合わせてペットの縦横のスケールを変更する。そのことにより、ペットの表示サイズが大きくなる。
- ⑧体重を数字で表示する。後に”g”をつける。

以上でこの作品は完成である。

## 4. 考察

### 4-1 感想

制作していく内に新しいアイデアが浮かんだり、携帯電話用であるが故にメモリの制限が厳しくアニメーションを作り直したりというハプニングがあった。制作に取り掛かる前に大まかな制作スケジュールを立てておき、デザインの骨組みを明確にしておくべきだったと反省している。特に苦戦したのはメモリ制限のトラブルで、メモリの容量が不足したせいで折角用意した画像やアニメーションが正しく表示されないことが多々あった。この問題に対しては各画像のサイズを小さくして結果的にファイルサイズを小さくしたり、アニメーションのボリュームを抑えたり、一つのムービーファイルをできるだけ少数の素材で作成したりすることで何とか解決することができた。このようなトラブルが続いた為、当初予定していたところまで作成できなかったことは残念である。しかし大学での作業と家でもできる作業を分担し、作品の構想を常に考え空き時間を有効に使ったことで一応、当初の目的であった「携帯電話のキャリアに左右されずに遊べるゲームを作成すること」が達成でき、満足している。

制作途中では見本があるだけに、見よう見まねでプログラムを書き、制作しながら少しずつ理解していただけないか心配であった。しかし今回作品についての論文を書くことで改めて理解したプログラミングの考え方も多々あり、Flashへの理解と興味もさらに深まった。作品作りが楽しめたことも、ゲームを形にする為の大きなモチベーションの一つとなった。

今後は応用テクニックを身につけ、さらにオリジナリティに溢れるモノへとブラッシュアップしていきたい。

### 4-2 反省点

一番の心残りは時間配分が上手くいかず、このゲームを設計通りに仕上げられなかったことである。上記でも述べたが、作りたいものの構想案だけでなく、最初に進行スケジュールを立て、具体的にどのような機能をつけ、どのようなデザインにするかというところまでしっかり決めておくべきであった。当初は自分でアニメーションを用意する予定もなく、考えていたデザインも現時点のものとは異なっていた。制作していく上で新しいアイデアが思い浮かぶのは誰にでもあることだとは思いますが、私の場合、思いついたことをその場でぶっつけ本番で実践してみたりと、寄り道が多かった。まずは最初にデザインしたものをしっかり形にすることから優先すべきだったのではないかと思う。最初に完成後の形をしっかり決めておき、できるだけ決めた軸からブレないようにするのが限られた時間

内でモノ作りを完成させるコツであると思う。さらに、**actionsript**を使用してこのような大きな作品制作をしたことが初めてだったこともあり、手本にした作品の構造やプログラムについて理解するのに予想以上に時間がかかった。自分で制作しながら少しずつ理解していったが、自分が作成しようとしているものについてどの程度実現可能なのか、どこまで理解しているのか、理解する時間を作るとしたらどれくらいの時間を割けるのかをしっかりと把握しておく必要があると感じた。

また、もっとオリジナル性のある機能を付けたいと思う。その際には使用するプレイヤーのことを考えたものを追加したい。もし自分がこのゲームの遊び手だったらどんな機能やデザインがあるとさらに楽しめるかを考えていきたい。これについては次の節で具体的に述べようと思う。

#### 4-3 課題点

作品制作においてゴールというものは存在せず、どこかで妥協しなければ制作活動は永遠に続くものであるが、ここでは私が今後このアプリにどのような機能を付加したいかを挙げていく。

まず、この作品はゲームであるにもかかわらず能動的に遊べる部分が少ないので、

- エサをあげない状態が続くと体重が減る
- ペットの性別を選択できたり、エサをあげる時間を選べるようにする

という機能を付け加えたい。次に、私がこの作品を作ろうと思ったきっかけであるドコモのiコンシェルのように携帯電話で生活を便利にする機能も付けたい。そこで現時点できそうなのはお知らせ機能の追加である。例えば

- メールや電話が来たら知らせるアニメーションを付ける
- バッテリーが減ってきたら知らせるアニメーションを付ける
- 電源のon/off時にアニメーションを付ける
- スケジュール帳と連動して、予定時刻になったら画面に予定を表示して知らせるアニメーションを付ける

のようなことである。

さらに、ゲームをより楽しく遊ぶ為にペットの動きや成長にもバリエーションを増やしたい。例えば

- 充電時にエサを食べているようなアニメーションを付ける
- 老体まで成長させる
- 病気になる
- プレイヤーとペットとの親密度を表わす点数を付け、それによって異なるアニメーションを付けたり、プロフィールを参照してプレイヤーの誕生日にはお祝いしてくれる機能をつける



などである。現時点だけでもこれだけ追加したい機能があり、今後はこれらを実装し、ソフトの機能を充実させていきたい。

## 5. 参考文献、URL

### 書籍

- ✚ 『Flash For Mobile』  
著者：クスール（松村慎+尾崎俊介+杉本篤史+岡崎大典）  
発行人：篠原 淳  
発行・販売：株式会社ワークスコーポレーション  
出版年月日：2008年12月4日

### Webページ

- ✚ 株式会社クスール  
<http://www.cshool.jp/>  
(2010/9アクセス)
- ✚ Forum.Nokia  
<http://www.forum.nokia.com/>  
(2010/12~2011/1アクセス)
- ✚ Flashの枝  
<http://1art.jp/>  
(2010/12~2011/1アクセス)
- ✚ DR.SHADOWrin  
<http://dr.shadowrin.com>  
(2010/12~2011/1アクセス)
- ✚ Adobe Flex 3ヘルプ  
[http://livedocs.adobe.com/flex/3\\_jp/html/help.html](http://livedocs.adobe.com/flex/3_jp/html/help.html)  
(2010/12~2011/1アクセス)
- ✚ @IT情報マネジメント  
<http://www.atmarkit.co.jp/im/>  
(2010/12~2011/1アクセス)

✚ IT用語辞典

<http://e-words.jp/>

(2010/12~2011/1アクセス)

✚ Frogstone フロッグストーン

<http://flash.frogstone.jp>

(2010/12アクセス)

✚ FLASHTAVE

<http://flashrave.org/>

(2010/12~2011/1アクセス)

✚ MMD研究所

[http://mmd.up-date.ne.jp/news/detail.php?news\\_id=83](http://mmd.up-date.ne.jp/news/detail.php?news_id=83)

(2010/12アクセス)

✚ ゲームロフト研究所

<http://www.gameloftjapan.com/>

(2010/12アクセス)