

Flash、Perl を用いた
自己サイト構築

山本昌弘、重定如彦ゼミナール

4-J 01G0908

窪田一平

目次

1. はじめに

はじめに

卒業研究の目的

2. 使用言語について

Flash MX Action Script

Perl(CGI)

(必要な環境)

(必要なソフトウェア)

3. サイトの概要、仕様

HP の形態

プログラム内容

4. 考察

選択について

基礎の大切さ

何通りもある中で

今後の課題

5. 謝辞

6. 参考文献

1. はじめに

はじめに

現在の社会は、まぎれもなくコンピュータ社会である。日本中の企業を見渡す事は、不可能であるが、ほとんどの企業がコンピュータを使用し、企業活動を行っているだろう。1990年代にITバブルという時代が到来したが、今ではコンピュータと社会は切っても切れない関係に成長した。

企業活動以外を見ても、徐々にコンピュータが浸透してきている。周囲の人間のほとんどが携帯電話を持っている時代、それも小学生の層まで広がっている事実を見ても、このように考える事は容易な事だろう。しかも、五年前までは、小学生が携帯電話を持っている事が普通だという事は考えられなかった。このような事からも、ITの浸透するスピードが以上に速いという事が分かる。

多くの家庭は家にコンピュータを保持している。家に一台というだけでなく、一人一台という家庭も珍しくはない。特に、私が所属している国際文化学部ともなれば、コンピュータが必須の学部という事もあるので、コンピュータを持っていない学生がいるのかどうか分からない程である。実際、私はコンピュータを持っていない友人を知らない。また、私の周囲に関しては、コンピュータの使用頻度が非常に高く、毎日のようにコンピュータを開いている場合が多い。

では、なぜこれほどまでに多くの人がコンピュータを保持しているのであろうか。その理由は、コンピュータを持っていないと生活に影響が出るような状態に社会がさせているからだと考えられる。今では、買い物から調べ物までコンピュータがあれば何でも出来る時代になっている。以前、私たちは宿題が出ると図書館に行って調べ物をしたものだが、現代の学生はインターネットによって、それを行う事が出来る。これだけでは、影響が出るとまでは言えないかもしれないが、次にあげる例は、完全に人生に影響を与える事例である。

人生に影響を与えるというのは、私が昨年度経験した就職活動である。インターネットがなければエントリーすら出来ないという現状を考えると、コンピュータを持っている事が普通だという時代になったという事が、感じ取る事が出来るであろう。

卒業研究の目的

今回 Flash を用いた自己サイトの構築をしようと考えるにあたっては、さまざまな紆余曲折があった。

実際、卒業研究が始まった頃、PHP を使用したシフト管理システムを構築しようとしていた。三年生の後期から就職活動を通して、四年生の春先まではシフト管理システムを構築していたのではないだろうか。

しかしながら、四年生の春先になると自分の卒業研究に疑問を感じ始めていた。自分はなぜこのようなシステムを作成しているのだろうか。もちろん、公にプレゼンテーションをする中では、自分が働いている中で、このようなシステムがあれば、非常に便利だという理由は存在していた。しかし、本当に自分で考え抜いて出した結論ではなかった。三年生の時点で、先輩の後姿を見て、このようなシステムを作ればいいのではないか。何か流れに身を任せたような自分が存在していた。使用言語に関しても、何も分からないまま先生や先輩の助言のままに Perl を使用する事に決めた。

三年生の十月から四年生の六月までは、就職活動によって完全にプログラミングから遠ざかった時間を過ごした。ゼミを休む事はなかったが、ゼミの時間中も先輩から就職活動の情報を聞いたり、コンピュータを使用し、情報を捜したりして、集中するという言葉からは到底近づけない状態でした。

しかし、就職活動は卒業研究にとって決して無駄なものではなかった。なぜシフト管理システムを構築しようと思っているのか。なぜ Perl を使用しているのか。悶々としながら次は i アプリへ。そして、最後に Flash へと考えはまとまっていた。この背景には、就職活動を経験する事によって、自分自身の思考をどれだけ深いものに出来るか、貫く事、計画を立てる事がどれだけ大切なもののなかという事を学ぶ事が出来たという事が大きく関わっている。

このような紆余曲折を経て、卒業研究に自分自身の課題を解決するという事を重ねて、Flash を用いた自己サイトの構築という結論に至った。ホームページというのはいろいろな人に見てもらう為には、更新をする事が不可欠である。いつ見ても同じであれば、一回見ただけで終わりという事になってしまう。皆の興味をひくような内容を常に発信していれば、あのずっとそのホームページに訪れるという事が日常茶飯事となってくる。卒業後、皆と会う機会が減少する事は明らかな事実なので、ホームページをハブとして繋がっていら

れればいいなという点もある。今まで、こまめに何かを続けるという事が出来なかつた自分。この点を改善するという自分自身の目標を考え、自己サイトの構築という結論を出した。

2. 使用言語

Flash MX Action Script

Flash の原型は、1996 年 Future Splash Animator という名前で発売されたソフトウェアである。このソフトがその後 Flash に名前を変えて、現在では Web アニメーション作成ソフトの定番となった。

Web アニメーション作成ソフトの定番と書いたが、Flash ではプログラム制御をほとんど使用しないアニメーション機能のみを使用するという形もある。ゲーム作成に非常に高い能力を持っていると考えられているのは、このアニメーション機能の部分があるからではないだろうか。

もう一点、ゲーム作成機能として非常に高い能力だと考えられている点は、ユーザーを選ばないという点である。例えば、有名なゲームソフトであるエニックス社のドラゴンクエストの最新バージョンであれば、PlayStation2 という機械本体を購入しなければ遊ぶ事が出来ないし、同様に、同社の他製品、他社のさまざまな製品に関しては、ファミリーコンピュータ、Game Cube、任天堂 DS などの本体を購入しなければ遊ぶ事が出来ない。それぞれなかなか値段もはるので、一般家庭であれば全ての機械本体をそろえる事は容易な事ではないと考えられる。

しかし、Flash に関しては Web ブラウザを持っているコンピュータ環境さえあれば誰でもお金をかけずに楽しむ事が出来る。Flash を使用して何かゲームを作るという場合は、有料の Flash ソフトを用意するしかないが、Flash で出来たゲームを楽しむ事に関しては、Flash Player というプラグインソフトをマクロメディア社の Web サイトから無料でダウンロードする事が出来るので、気軽に楽しむ事が出来ます。

Flash に本格的なスクリプト機能が搭載されたのは 1999 年リリースのバージョン 4 であり、Flash5、Flash MX のバージョンアップにより、より充実されたスクリプト機能が搭載され、複雑なゲームの作成も可能なソフトウェアになった。

今回私が使用しているのは、最新バージョンの Flash MX であり、Flash MX Action Script と多少のアニメーション機能を活用し、自己サイト構築に取り組んだ。

正直、Flash を使用したのは今回が初めてなので、Flash4、Flash5、Flash MX というそ

それぞれのバージョンの違いがどれほどのものなのかという事は分からぬのが残念な事であるが、Flash MX では初心者でもゲーム作成に取り組む事が出来るという事は間違ひはない。

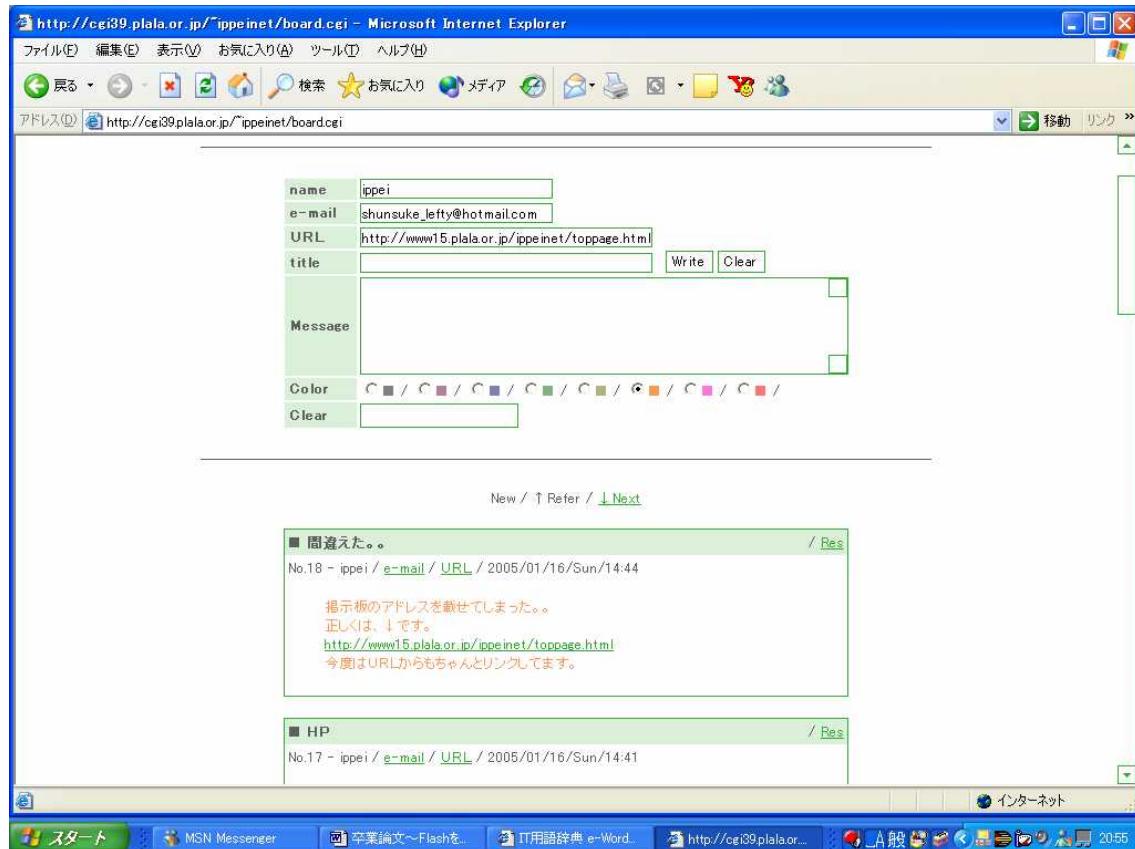
Perl、PHP、C 言語など、さまざまな言語にはさまざまな言語の特徴があるが、今まで学習した Visual Basic などの考え方と基本的なところは変わらないので、四年生の後期から本格的に Flash を始めたという事ではあるが、基礎的な部分は以前からの知識もしっかりと役に立っている。しかし、Flash は少し独特な部分があり、Flash 自体のアニメーションの作り方や、プログラムを書く場所など使い方を理解する事に多少時間を費やさなければならぬというマイナス点も見逃す事は出来ない。実際、私も一ヶ月近くは Flash 自体の使い方を学習するという事に時間を取られてしまった。

Perl(CGI)

Perl という言語は、Practical Extraction and Report Language の略であり、Larry Wall 氏が開発したプログラミング言語である。テキストの抽出や検索、レポート作成に向いている言語だと考えられており、表記法としては C 言語に似ているだろう。インタプリタ型であるため、プログラムを作成したら、すぐに実行する事が出来、コンパイルなどの処理を省く事が出来るという特徴を持っている。多機能な事で知られており、CGI の開発にはよく使われる。

今回の自己サイト構築の中で、Perl を使用するという構想は当初はなかった。しかし、サイトの内容を考えていくにつれて、CGI を使用したコンテンツを盛り込みたいという考えが浮かんできた為、CGI の中で使用頻度が高い Perl という言語を選択し、使用した。具体的なコンテンツとしては、ゼミの掲示板でのみ使用されている。

ゼミ掲示板



(必要な環境)

FLASH + CGI を使ったホームページを構築する為に必要な環境は、基本的には FLASH のみの場合とほぼ同じである。しかし、唯一違う点はその WWW サーバーで CGI を利用する事が出来る必要があるという点があるという事だ。よって、必要な環境は以下の二つである。

制作を行う為のパソコン

自作の CGI を利用出来る WWW サーバー

特に注意が必要なのが である。プロバイダのホームページを閲覧してみると、ほとんどの場合、CGI に関して記述してあるので、ホームページをチェックして見る必要がある。では、どのような場合に問題が生じるのか。それは、以下の点である。

プロバイダの中には CGI を利用出来ないものも存在する。

CGI が利用出来ても、プロバイダが用意した CGI しか利用出来ない場合がある。

必要な環境が揃わなければ、どんなに素晴らしいものを構築出来たとしても、公開する事が出来ないので、一番初めにクリアしなければならない問題がこれらの事だろう。

(必要なソフトウェア)

では、FLASH + CGI を利用したコンテンツを制作して公開する為にはどのようなソフトウェアが必要だろうか。それは、以下のソフトウェアである。

Macromedia Flash MX(or Flash 5)

Flash のコンテンツを制作する為には必要不可欠なものがこれである。必要なソフトウェアのうち、唯一フリーソフトではないのがこの Flash MX である。ちなみに、Flash を閲覧する為に必要な Flash Player はフリーソフトとなっているので、もし、Flash MX(Flash 5)を持っていない人がいたとしても、ホームページなどを閲覧する事は、Flash Player さえ準備すれば可能である。

テキストエディタ

CGI プログラムはテキストベースで記述する事からこのテキストエディタが必要になってくる。Windows に付属しているメモ帳や、Macintosh に付属している SimpleText で十分編集は可能である。しかし、もっと使い勝手の良いテキストエディタも存在しているので、チェックしてみるのも良いかもしれない。

FTP ソフト

FTP ソフトとは、パソコン上にあるファイルをサーバー上へコピーして、アップロードする為のソフトである。また、アップロードした CGI プログラムやデータを CGI として動かす為に、パーミッション (ファイル権限) の設定も行う。

3. サイトの仕様、概要

ホームページの形態

top page

* トップページ。以下の のコンテンツとリンクで繋がっている。 *

Diary

* 日記。 *

Link

* 友人や大学関係のサイト *

friends

univ.

Soccer(link)

* 趣味であるサッカー関係のリンク先 *

snow stage

* 雪が降るようなアニメーションをプログラムしている。 *

Shunsuke Nakamura Official Web Site

Hidetoshi Nakata Official Web Site

Junichi Inamoto Official Web Site

Shinji Ono Official Web Site

Photo

* 写真にプラスして、その写真に関する文章を加えている *

Board1

* ゼミ、留学先、クラスの掲示板とリンク *

Board2

Board3

Game

* Action Script を使用したゲーム。 *

Mogura (もぐら叩き)

Shooting akiba ver. (シューティングゲーム アキババージョン)

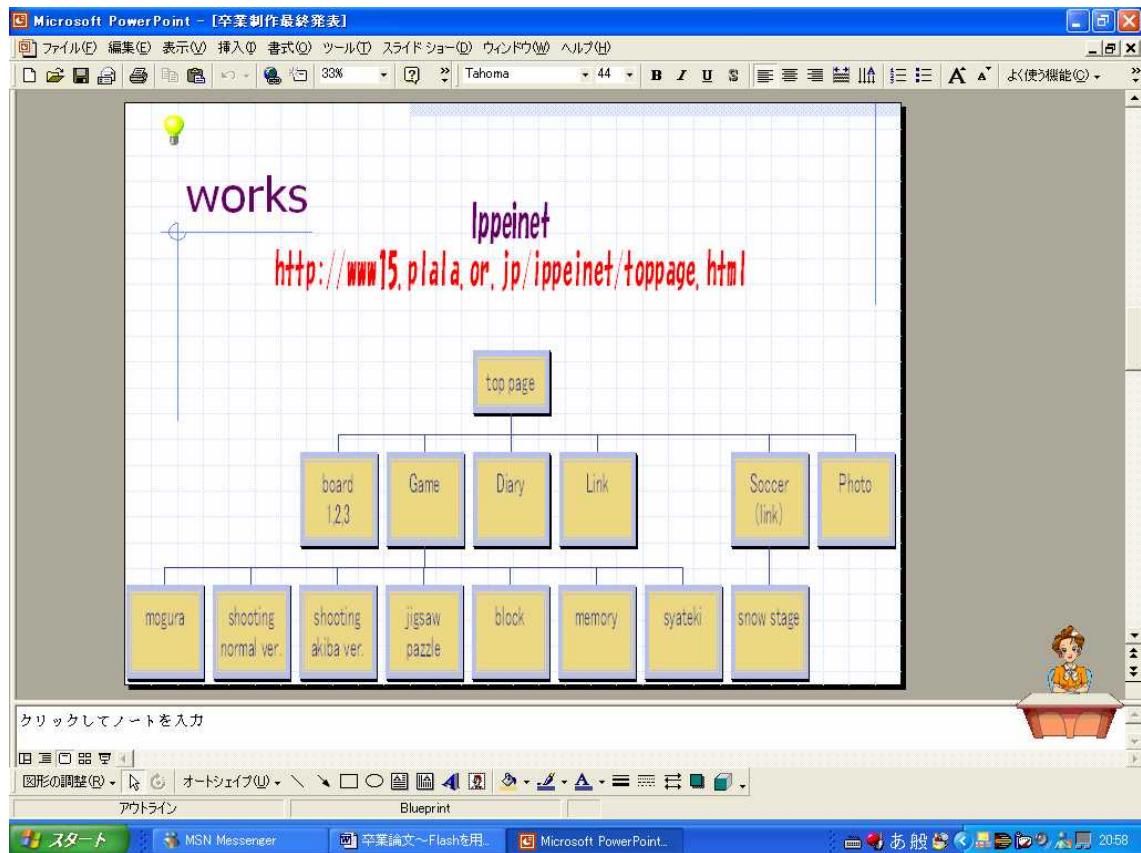
Shooting normal ver. (シューティングゲーム ノーマルバージョン)

Jigsaw puzzle (ジグソーパズル)

Block (ブロック崩し)

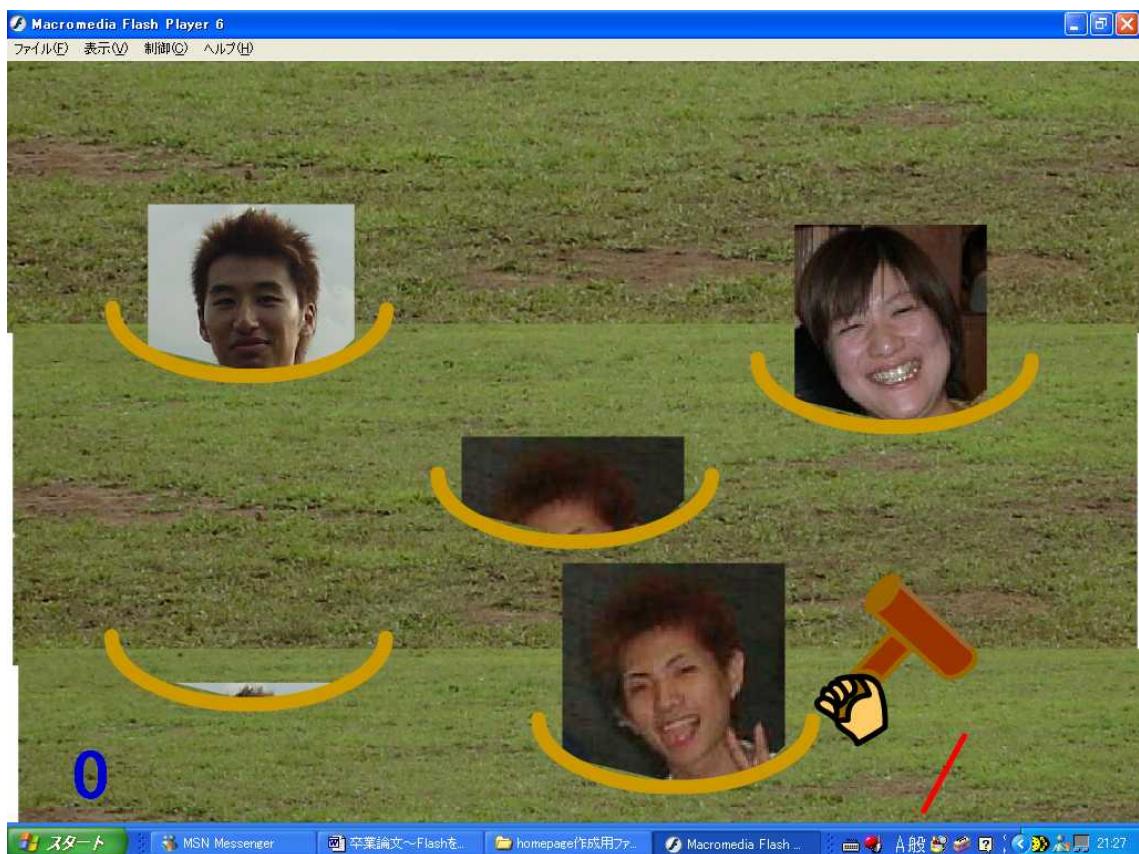
Memory (神経衰弱)

Syateki (射的)



プログラム内容

もぐら叩き



もぐら叩きのプログラムの一部

```
this.stop();
//ストップさせる
Mouse.show();
//マウスを表示させる
score = 0;
//スコア
function addScore() {
    score += 10;
}
//addScore というファンクションを設定
```

```
//スコアを10点加える
function addScore10 {
    score += 15;
}

function addScore20 {
    score += 30;
}

function addScore30 {
    score += 5;
}

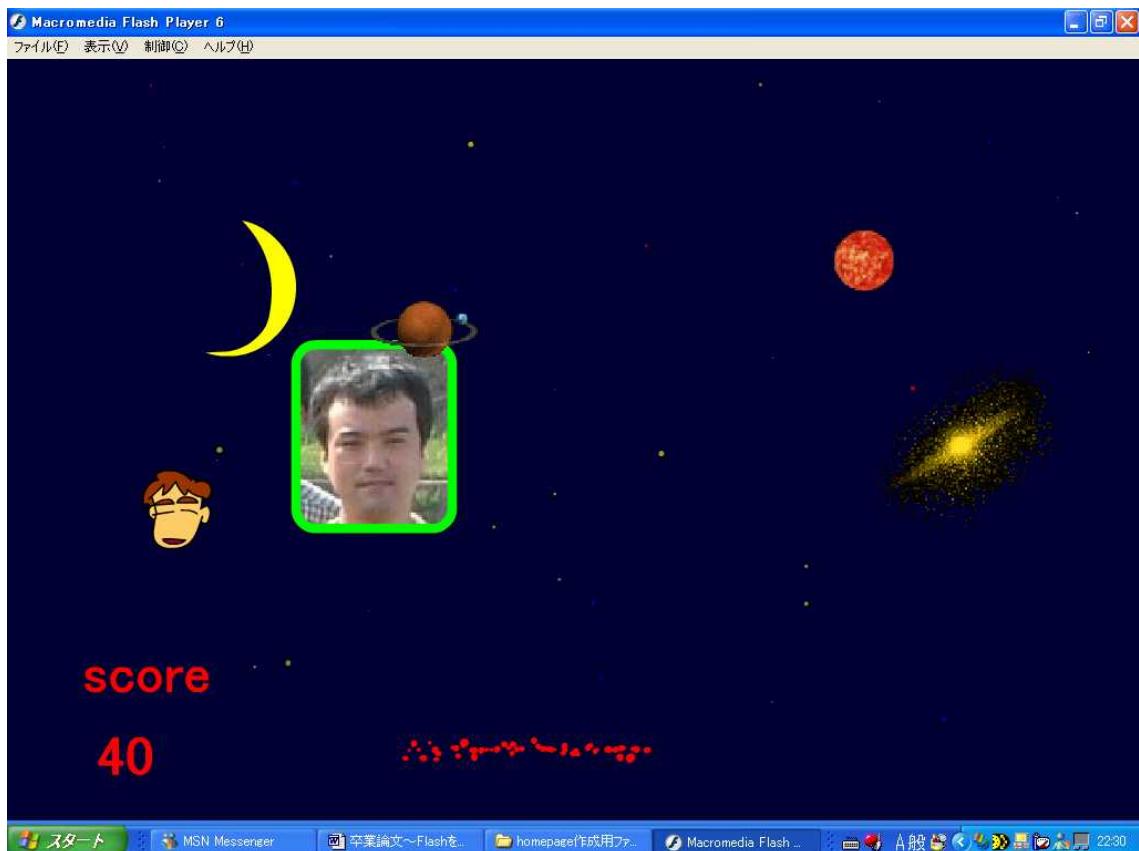
function addScore40 {
    score += 1;
}

function decScore() {
    score -= 100;
}

//decScore というファンクションを設定 スコアを10点マイナス
```

それぞれのもぐらに対して得点をバラバラに設定する為に、function を最初に設定し、それぞれのもぐらに対してハンマーで叩かれた時に、function を呼び出すように設定し、得点の設定としている。

シューティング



シューティングのプログラムの一部

//上の図中央、縁で囲まれた中ボス出現のプログラム

//中ボスは、一発の弾丸で倒す事が出来ないように、同じ位置に15匹出現させる事によって、15発命中させなければ倒せないように設定している。

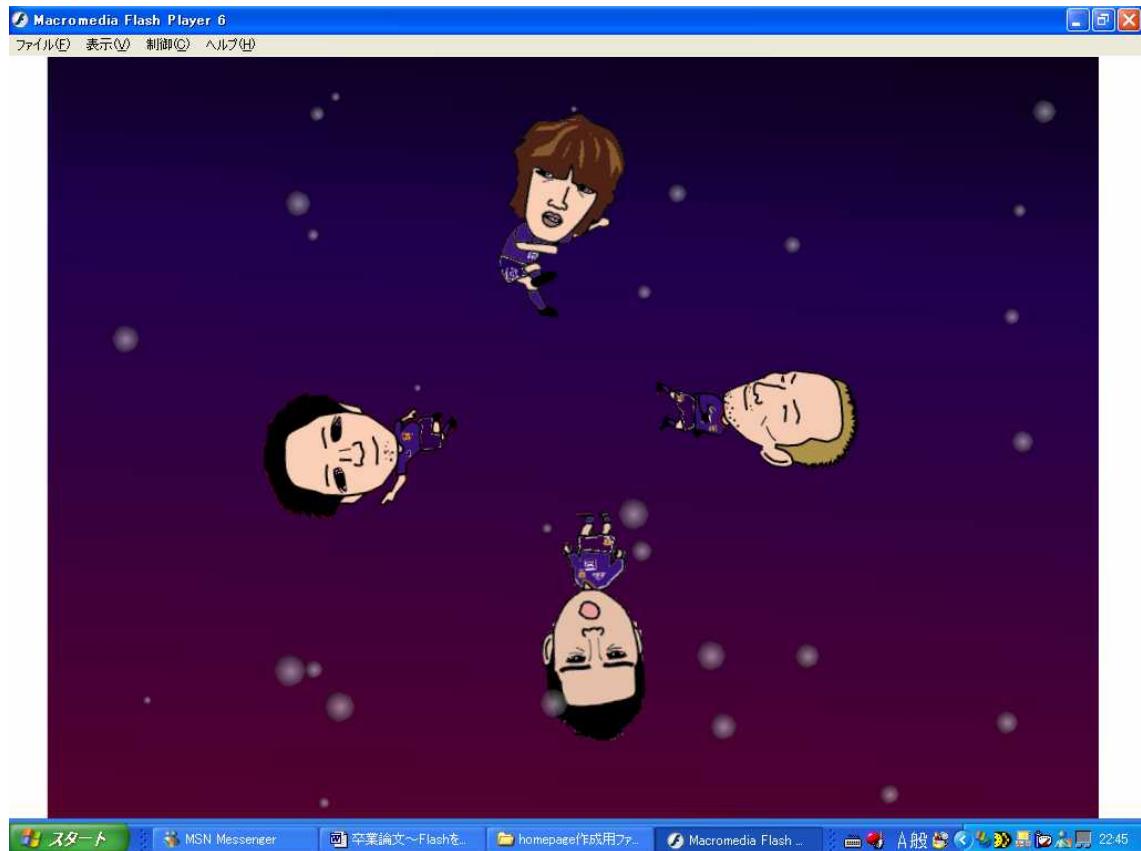
//for文を使用し、中ボス出現を15回繰り返している。

```
for(i=1; i<=15; i++) {  
    this.smallbosssyutsugen1(i);  
}  
//flash 独特のプログラム  
//敵と弾丸を hitTest というプログラムで判定するのだが、その為に attachMovie  
という方法で敵を出現させている。
```

```
//中ボス 1 出現
function smallbosssyutsugen1(i) {
    //smallboss1 というインスタンス名を一度 smallbossderu1 という変数に
    //一度入れる。
    _root.attachMovie("smallboss1","smallbossderu1", ++n);
    newBullet = eval(_root.smallbossderu1);
    // x 座標を指定
    // y 座標を指定
    newBullet._x = 4200;
    newBullet._y = 200;
    //smallboss1 というインスタンス名に戻す
    newBullet._name = "smallboss1"+i;
}
```

//この一連の作業で中ボスを出現させている。

スノーステージ



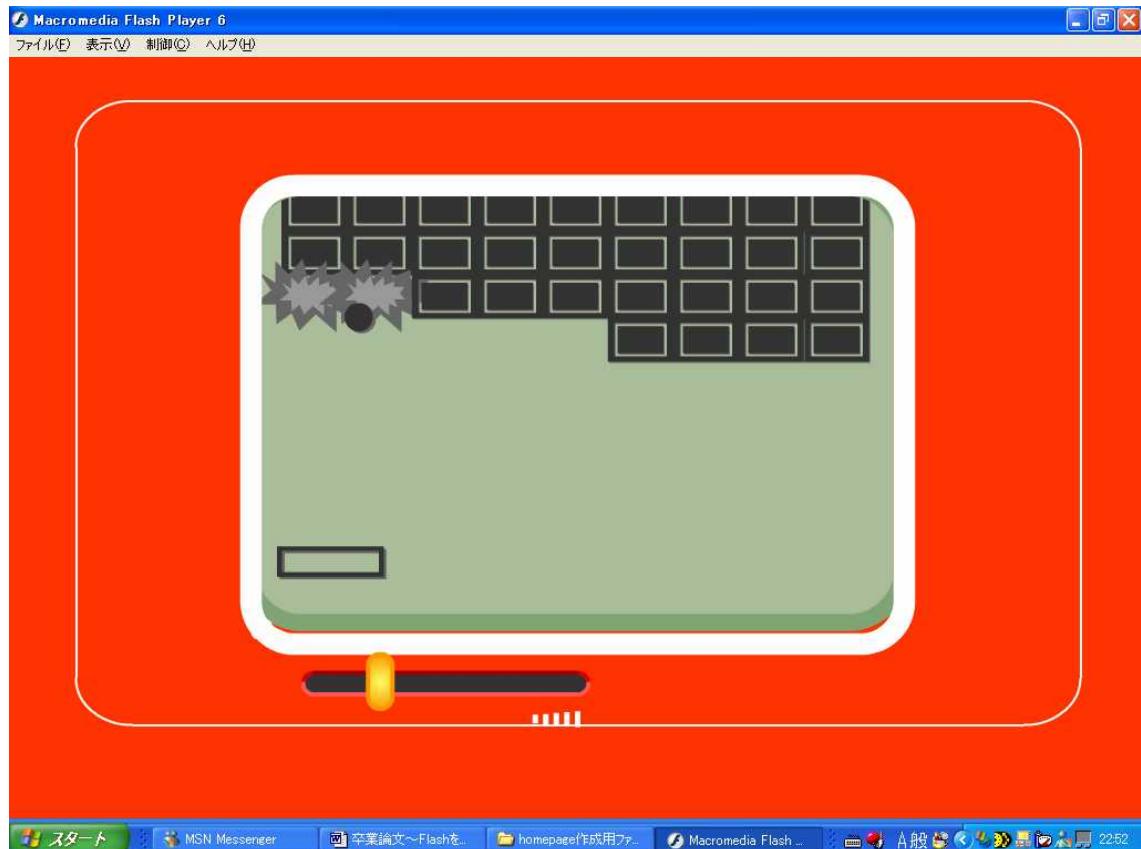
スノーステージのプログラムの一部

雪が降っているようなアニメーション。雪の大きさを手前ほど大きく、奥ほど小さく、また、スピードは手前ほど速く、奥ほど遅くする事によって、立体感を出している。

```
init();
function init() {
    //擬似的な深度
    myDepth = ++_global.depth;
    //遠くほど小さく、手前ほど大きくする
    this._width = 3+0.5*myDepth;
    this._height = 3+0.5*myDepth;
    //半透明
```

```
this._alpha = 50;  
}  
//雪が降るアニメーション  
this.onEnterFrame = function() {  
    // 横揺れ  
    this._x += (1-Math.random()*3);  
    // 落下  
    this._y += 5+0.5*myDepth;  
    // ループ  
    if (this._y>Stage.height) {  
        this._x = Math.random()*Stage.width;  
        this._y = 0;  
    }  
};
```

ブロック崩し



ブロック崩しのプログラムの一部

```
//ボールがラケットに当たったときの定義
function hitRacket() {
    //もし、ボールがラケットに当たったら、
    if (this.hitTest(_root.racket)) {
        //ラケットに当たった時のサウンドを生成
        //racketSound.start()というfunctionは、別の場所で設定している。
        racketSound.start();
        //ボールの進行方向を下から上へ
        moveY = -moveY;
        if (moveY>=0) {
            moveY = -moveY;
```

```
        }
        moveX += addX;
        if (moveX>10) {
            moveX = 10;
        } else if (moveX<-10) {
            moveX = -10;
        }
    }
}

//ボールがブロックに当たったときの定義
function hitBlock() {
    //ボールがブロックに当たった時、ラケットに当たった時と同様にボールの進行方
    //向を逆にする。
    moveY = -moveY;
    if (moveY<=0) {
        moveY = -moveY;
    }
}
```

フォト



フォトのプログラムの一部

このフォトに関するプログラムは、Flash の特徴を活かしたものである。

Flash は、アニメーションを中心として作成する事も出来ると前述したが、アニメーションは、レイヤーというコマ送りの手法で作成する。

今回のフォトに関しては、このコマ送りを利用した非常にシンプルな作りとなっており、ひとコマに対して、写真と文章をひとつずつ貼付したものを、プログラムによって、コマを送ったり、もどしたりという動作をさせるという手法によって、構築されている。

```
//これは、ボタンにプログラムをしたものである。  
//ボタンを離した時  
//ルート（基本の画面）にある変数 pic というムービーを次のコマに送る。  
//同様に picsentence というムービーを次のコマに送る。
```

```
on(release) {  
    _root.pic.nextFrame();  
    _root.picsentence.nextFrame();  
}
```

クッキー



クッキーのプログラムの一部

```
//画面を開くたびに vvisitCount という変数を 1 ずつ増やす
```

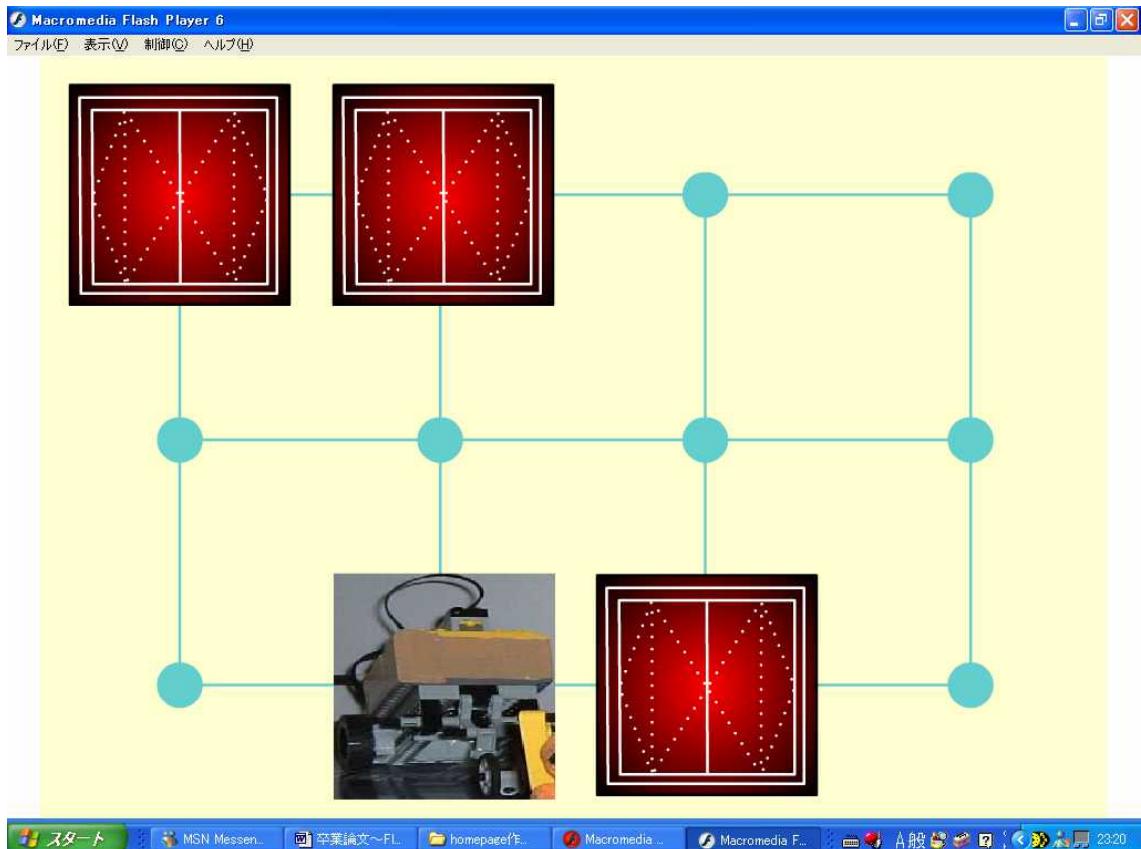
```
//考え方としては、CGI プログラムなどを使用するわけではなく、その画面を開いたコンピュータのローカルフォルダにユーザー名と訪問数を記録させるという仕組みである。
```

```
//よって、初めての訪問時にユーザー名を入れると、次回以降の訪問時にはユーザー名と訪問数が呼び出される。
```

```
my_so.data.vvisitCount++;
_root.uuserName.text = my_so.data.uuserName;
```

```
_root.vvisitCount.text = my_so.data.vvisitCount;
```

メモリー



メモリーのプログラムの一部

//神経衰弱の最も大切な部分である、めくったカードに対する判定に関するプログラム

```
//めくった二枚のカードが同じカードかどうかをカードの名前についている数字で判定している。  
//カードは全部で十二枚。同じカードは一と七、二と八というように差が六となるように設定している。  
//そこで、カードの差が六の時、二枚のカードを消去している。  
//もし、差が六以外の時は、カードをめくる前の状態に戻す。  
//カードがワンペア消えていくたびに、ポイントを一つずつ加えていく。  
//六ペア全てのカードが消える、つまり六ポイントになった時点で、ゲームクリアとなる。
```

```
//2つの駒の数字の差を求める
//ここでは、slice(2)という部分で三文字目(0, 1, )を取り出している。
//これは、カード自体にMC1、MC2...MC12というように名前をつけていたる為、数字だけを取り出して、当たり判定の材料にしている。
```

```
//カードの数字(三番目)を取り出している
```

```
mcNum1 = mcName1.slice(2);
mcNum2 = mcName2.slice(2);
```

```
//最初にめくったカードと二番目にめくったカードの数字の差を出している
```

```
difNum = Math.abs(mcNum1-mcNum2);
pairMC1 = eval(mcName1);
pairMC2 = eval(mcName2);
```

```
//数字の差が6であれば駒を消す
```

```
if (difNum == 6) {
    pairMC1.disappear();
    pairMC2.disappear();
    winPoint++;
}
```

```
//数字の差が6以外であれば駒を戻す
```

```
} else {
    pairMC1.putBack();
    pairMC2.putBack();
}
```

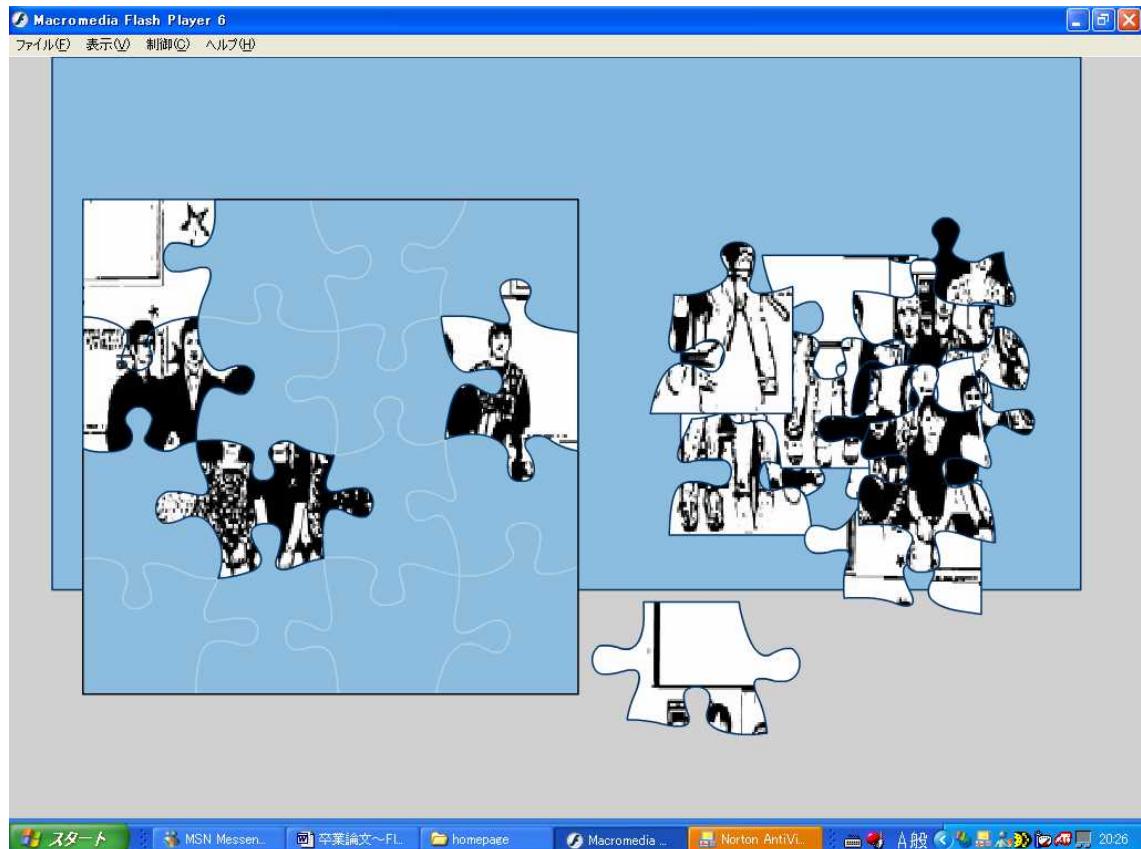
```
//ポイントが6になった時点で、ゲームクリアの画面へ。
```

```
//それ以外の場合は、ゲームに戻る設定。
```

```
if (winPoint == 6) {
    this.gotoAndPlay("win");
```

```
 } else {  
     this.gotoAndPlay("game");  
 }  
 
```

ジグソーパズル



ジグソーパズルのプログラムの一部

//パズルをドラッグし、はめ込む際のプログラム

//基本の考え方は、ピースをドラッグし置いた時、ピースの変数とはめ込む側のパズルの変数とが合った場合に、はめ込まれるという考え方。

//ボタンを押した時（つまりクリックした時）

```
on (press) {
```

```
    //ドラッグを開始する
    this.startDrag(true);
```

```

//重なり順をひとつ上の階層の変数「n」+1にする
this.swapDepths(++_parent.n);

}

//ボタンを離した時(つまりクリックを離した時)
on (release, releaseOutside) {

    //ドラッグを止める
    this.stopDrag();

    //変数 myNum をインスタンス名の2番目以降の文字にする
    myNum = _name.slice(2);

    //変数 lineName を line + 変数 myNum にする
    lineName = "line"+myNum;

    //変数 lineMC をインスタンス_parent.+変数 lineName にする
    lineMC = eval("_parent."+lineName);

    //もし、ドラッグ先が変数 lineMC であるなら
    if (_droptarget == lineMC._target) {

        //X座標を変数 lineMC のX座標と同じにする
        _x = lineMC._x;

        //Y座標も同様に
        _y = lineMC._y;

        //はめ込まれた時に、位置が正しいという事を知らせる為、淵を光らせる
        //ようなアニメーションを動かす。その為のplay。
        this.play();

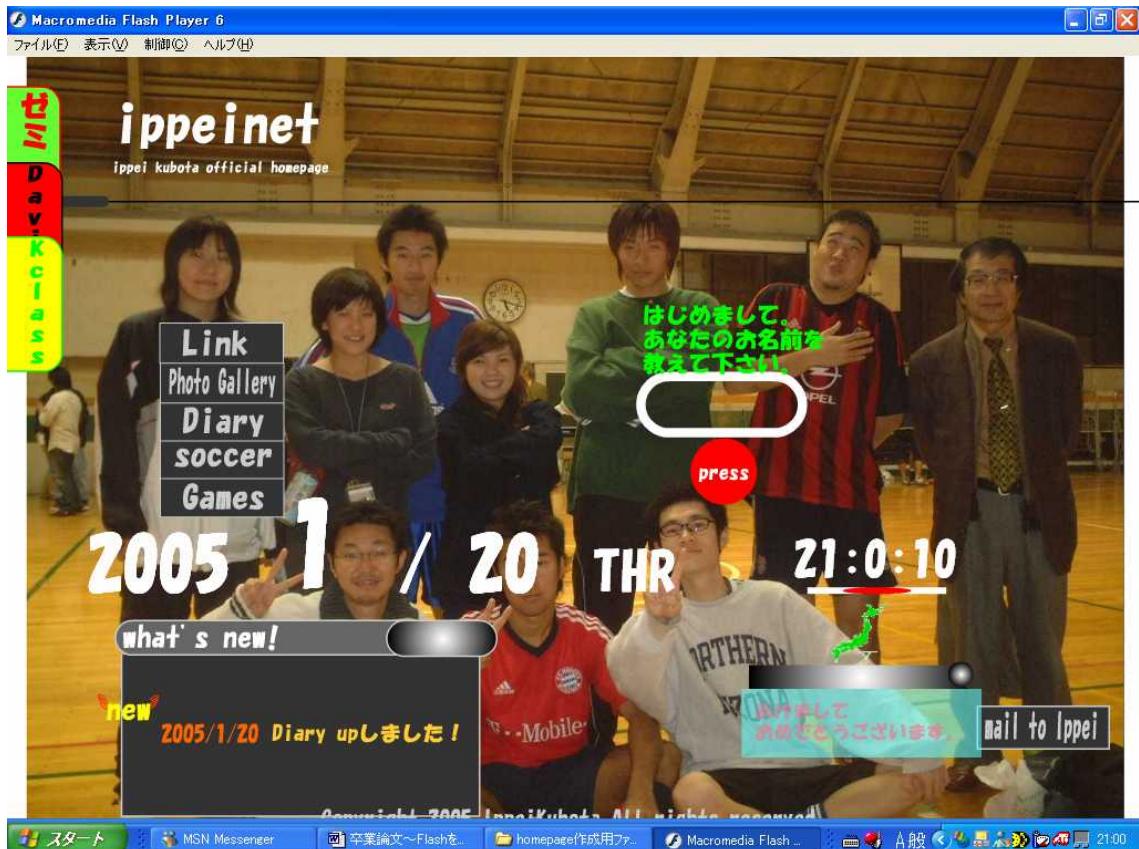
        //_rootという場所に設定したファンクションであるcompPieceを実行する。
        _root.compPiece();
    }
}

```

}

}

トップページ



トップページのプログラムの一部

//トップページの時間を取得する為のプログラム

```
//EnterFrame というのは、常に監視し続けるという意味  
//時間を常に監視し続けて、常に新しい情報を得る。  
//常に新しい情報を得る事が出来るので、時間が正しく進んでいく。  
this.onEnterFrame = function() {
```

```
//now_date を new Date という値に入れる  
now_date = new Date();
```

```
//変数 h に表示させるものを取得する。  
//この場合は、何時何分の何時を取得。  
h = now_date.getHours();  
  
//変数 m に表示させる値を取得。  
//この場合は、何分という部分を取得。  
m = now_date.getMinutes();  
  
//変数 s に表示させる値を取得。  
//この場合は、秒を取得。  
s = now_date.getSeconds();  
  
//dispTime という変数に hms の変数を並べ、間にコロン( :)を入れる事によつ  
て、デジタル時計のように表示させている。  
this.dispTime = h + ":" + m + ":" + s ;  
  
//上記と同様の考え方により、変数 mo に何月かという数字を取得する。  
//数字は 0 ~ 11 という数字を取得する事になるので、何月かという表示をさせる  
為には、表示させる前に + 1 をする必要がある。  
//よって、最後に mo + 1 をしている。  
now_month = new Date();  
mo = now_month.getMonth();  
this.dispMonth = mo+1;  
  
//上記同様の考え方。日にちを取得。  
now_day = new Date();  
d = now_day.getDate();  
this.dispDay = "/" +d;  
  
//上記同様の考え方。年を取得している。  
now_year = new Date();  
y = now_year.getFullYear();  
this.dispYear = y;  
  
//上記とは多少異なる考え方になる。  
//曜日を求めているのだが、曜日は数字では表す事は出来ない。
```

```
//よって、配列に SUN(日) から SAT(土) までの表示を入れておいて、その配  
列から曜日を取り出している。  
now_date = new Date();  
youbi = ["SUN", "MON", "TUE", "WED", "THR", "FRI", "SAT"];  
youbiNum = now_date.getDay();  
this.dispYoubi = youbi[youbiNum];  
};  
  
//cokkiee という外部ファイルを呼び出して表示させている。  
loader_mc.loadMovie("cokkiee.swf");
```

4. 考察

選択について

授業も含めると、大学生活の中で四年間プログラミングを経験した。三年生から始まつたゼミ以外のプログラムというのは、授業で経験したほんの少しだけなので、実質一年半程度になるだろう。しかし、一年生から二年生にかけて経験した授業内でのプログラムからもたくさんの事が学べたと今は思っている。

私は、一年生から二年生にかけてのプログラムの授業は本当に分からずに苦労した。C言語を学んだのだが、何を学んだかは全く覚えていないほど、授業の内容を理解していなかった。覚えているのは for 文という言葉くらいだろうか。それも、for 文と言えば、今となっては理解しているが、当時はタイトルとして for 文という言葉を覚えているだけであった。当然まともにプログラムが動いた経験などはなかったし、授業の中でもひとり取り残されている状態だった。

三年生になるとゼミが始まる。そこで、どのようなゼミに所属したいかという事を必死で考えた。先輩の話などもたくさん聞いた。以前からコンピュータの知識があれば、少なくともこれからマイナスになる事はないだろうと考えていたし、学んでみたいという気持ちはあった。しかし、授業の状態が自分にコンピュータを選択させなかった。

しかし、最後には山本昌弘ゼミに辿り着いた。このような私が、今ではプログラミングが楽しいと思えるところまで辿り着き、サンプルを動かすだけではなく、自分で考えてプログラムを組み立てている。

このような経験からプログラミングをする為に最も必要な事は何かを考えると、やはり第一に楽しむ事ではないだろうか。プログラミングに対する意欲があるのとのないとでは、同じ一ヶ月を過ごすとしても、内容の濃さが全く変わってくるのである。

実際私のケースを考えてみると、内容の濃さにどれほどの違いが出ているのかが分かるだろう。三年生の九月から自分の卒業研究を始めたのだが、Flash を始めるまでの四年生の八月まで、就職活動における研究休止状態を考慮に入れても約半年間は研究を続けていたわけだが、全くと言っていいほどシステムが出来上がってない。出来上がってないどころか、サンプルをいじる状態から抜け出せずにいたのだ。

しかし、Flash を始めたのが九月。最終発表までの約四ヶ月間という期間でどんどんと自

分のシステムが完成に近づいていった。

ほぼ三分の二の時間で Flash の場合は完成へと進んで行った。どうしてここまで差が出たのか。それは、自分が感じている限り、明らかに興味、意欲が湧いていたか、いなかったかという違いだろう。シフト管理システムの場合、家でやるという事は全くなかったけれども、Flash の場合は、楽しみながら家でも進んでやった記憶がある。それに加えて、最終発表が終わった今でも、より高度な事を学んでいきたいという気持ちがあり、本屋へ足を運び、この本が欲しいなと選んでいるような状態だ。

この状態を考えると興味関心が湧くようなシステムを選択するという事がどれだけ重要な点なのかという事が納得出来るだろう。プログラムに関しては、一日二日の問題ではなく、数ヶ月という年月を要するのに加えて、少し間違えるだけで動かないという厳しさもあるので、忍耐力も必要になってくる。よって、他ゼミの卒業研究の選択よりも、より慎重な選択が強いられるのではないだろうか。

基礎の大切さ

では、実際プログラミングを行う際に感じた事を考えていく事にしよう。

プログラムを組み立てる際に、技術的な面から考えて何が一番大切になってくるのだろうか。誰もが思いつかないような奇抜なアイディアを形にする事だろうか。難しいプログラムを一目見て、理解できる事だろうか。これらもシステムを完成させる為には重要な事であるのは確かである。しかし、どんなに難しいプログラムであっても、やはり基本がしっかりした上で、色気を付け加えていくものである。基本が出来ていないプログラムでは、いくら色気を出そうと思っても、良い色気は出てこないだろう。これは、プログラムだけに言える事ではないだろう。サッカーや野球などのスポーツを始め、さまざまな事にあてはまる事実だろう。NY ヤンキースの松井秀喜は、寮の畳がすりきれるほど夜中に素振りを繰り返し、世界の舞台に飛び出していった。イタリア、レッジーナで活躍中のサッカー日本代表、中村俊輔は、日本有数のスタープレーヤーとなった現在でもドクターストップがかかるまで居残り練習を繰り返す。稀にこれらの例に当てはまらない人もいるが、基本的には世界に羽ばたくスタープレーヤーも、基本を大切にし、其の上で、オリジナリティを構築していっている。

Flash に出会ってから、向上心に溢れ、意欲をむき出しにして努力をして来たわけであるが、やはり、このようになったポイントは基礎をしっかり学習したという事実が関わっている。

Flash を始めた時、FLASH Action Script スーパーサンプル集という本を買い、即座にサンプルをプログラムしてみる事から始めてみたのを覚えている。しかし、プログラムはおろか、Flash 自体の使い方すら理解する事が出来ず、早々と挫折しそうになった事を覚えている。そのまま挫折していたらいままでの PHP や i アプリと同様の結果になってしまっていたんだろう。Flash が前例と違っていたのは、ここで改めて基礎をしっかりと学ぼうと考え、超初心者用の Flash 本を買い、復習を始めた事だろう。ただでさえ卒業まで時間がせまっている、他の人より遅く始めているという状況ではあったが、焦らず初心に戻ったのがよかったです。基礎が理解出来ると、後は自分自身の努力次第だと考えていたので、期限までに完成させるという事を忘れず、信念を持って進める事が出来た。

何通りもある中で

プログラムというのは、決まった答えというものが存在しないものだと私は考えている。実行したい事を数字で表す事が重要なプログラミングは、数学のような概念を持つ人も少なくはないだろう。実際、私も以前はそのような固定概念を持っており、数学が嫌いな自分には向いてないし、そんなに難しい事をやるのは不可能だろうと考えていた。しかし、実際は、数学とは違うものだという感想を持っている。

それは、なぜだろうか。数学では、答えがひとつしかない場合がほとんどで、その答えに達する事が出来ない場合は、全て誤答となってしまう。答えがひとつしかないというものは何かと息苦しいものだ。しかし、プログラミングに関しては 100 人のプログラマーがいれば、100 通りの答えが存在し得るのである。もっと大げさに言ってしまえば、一人の持っているプログラム方法がひとつでないと考えると、100 人に対して、200 通りや 300 通りの方法が存在し得るのである。

この幅の広い方法が、自分をプログラミングに対して楽に接する事が出来るようにしてくれたのかもしれない。他人のプログラム内容が自分に合わなかったり、他人の教えに納得が出来なかったりした場合でも、自分の独自の方法を見つけ、築き上げていく事によりプログラミングに関して理解出来なかったという問題が問題ではなくなるのだ。

独自の方法という観点からもう一つ注意しなければいけない事が存在する。それは、自分独自の方法を見つけた時は、必ずと言っていいほど自分の得意な方法のみを使用するケースが多く存在する。すると、自分が使用しないような方法に関しては忘れてしまったりする事が出てくる。よって、自分独自の方法を模索する際は、なるべく多くの方法を使用してプログラミングしていく事も重要な点になってくるだろう。自分が使い続けている方法に関しては忘れる事が少ないだろうし、プログラミングをする際のレパートリーも増えてくるからである。

以上のようなことから、プログラミングをするにあたって、ひとつのプログラムに対して何通りも存在する方法の中から、自分に合ったものを選択していく事が重要になってくる。自分独自で築き上げた方法が、一番理解しやすいと考えられるし、他人の方法を真似るのではなくて、自分自身でプログラムを考える事により、応用にも適用する事が出来るように、自分を成長させる事も出来るからである。自分自身で考えていく事により、自然に理解しながらプログラムを書けるようになるだろうし、基本を理解する事はその組み合わせによって、自然と応用という方法へと導かれていくのである。

しかし、ひとつだけ考えなければいけない問題がある。それは、現在は自分のシステムをひとりで構築しているという点だ。自分自身だけでプログラミングをするのであれば、このように自分独自の方法をどんどん考えて実行していけば良いのであるが、もし、これが大勢でひとつのプログラムを構築するという事になれば話は別だ。文字が一文字違うだけでも動かなくなってしまうプログラムというものに対して、それぞれの人が書いたプログラム同士の互換性というものは非常に重要になってくる。よって、このような場合は、他人とどれだけ協調性を持ったプログラムを書いていけるかという事が一番重要になってくるので、話し合ってプログラム方法を模索しなければならないだろう。

今後の課題

CGI の利用

最終発表が終わり、卒業論文提出が終わっても卒業研究自体を終了させる気は全くない。なぜならば、卒業研究の目的でも前述したが、自己サイト構築の目的のひとつに定期的な更新という目標が存在するからである。

まず、このサイトの主なコンテンツであるゲームに関してであるが、現在の状況では、ただ単に遊んで終わりという状態である。もちろんスコアが出るので自分の中での結果に関して、満足度を向上させていくという事は出来るが、限界がすぐに見えてきてしまう。又、これならばインターネット上で遊ぶ必要はなく、自分のコンピュータ上だけでプレイする事が可能なので、インターネットという利点を利用しきれていない部分がある。

そこで、CGI を利用したスコアランキングなどの機能を追加させる事によってユーザーに飽きさせない工夫を凝らす事が出来るのではないかと考えている。自分だけでは、又はゲームの製作者でも想像もつかないようなゲームの方法や、スコアを創出させる仕組みがそこにはある。

頻繁な更新

情報というものは、どんなに素晴らしい内容のものでも一度見てしまったら価値が下がってしまうものである。これは、当然の事で同じ内容のものを何度も見たいと思わせるものは、特別なものを除き、あまり存在しない。

つまり、こまめに更新をする事が必要になってくる。ホームページという形にした以上、更新という課題はいつまでもついてくる課題であり、放棄出来るものではない。放棄した瞬間にホームページを訪れる人数がどんどん減っていくのは目に見えている。

本卒業研究では、今まで頻繁に更新をするような事が出来なかった自分への課題という意味も込めて、ホームページという形態にこだわった為、更新をする事が出来る部分がたくさん備わっている。Photo、Diary、news など自分の努力次第で毎日でも更新出来るものである。

人は毎日違う経験をして生きている。同じようで違う毎日を過ごしている。全く同じ日というのは一生で一度もないはずだ。これだけの世の中を生きているのだから、更新が出来ないという理由は、ただ一つでさえ存在しない。もし、存在するとしたらハードウェアが壊れるなどの物的的理由くらいだろう。

これからどれだけ更新を続けていけるだろうか。自分自身に問いかけるとともに、自分自身が課した課題の面白さをしみじみと実感している今日この頃である。

5. 謝辞

最後に、本卒業研究、並びに卒業論文作成にあたり、熱心にご指導、ご鞭撻してくださった山本昌弘教授、重定如彦教授をはじめ、同ゼミ生の皆様、その他卒業研究にご協力くださいましたすべての方々に厚く御礼申し上げます。

6. 参考文献 URL

おしえて FLASH MX 2004 Action Script

森 巧尚 + komachan 著

フラッシュ ゲームプログラミング FOR WIN & MAC

インフォメディア 著

macromedia FLASH Action Script スーパーサンプル集

大重 美幸 著

一週間でマスターする CGI for Windows

福島 靖浩 著

Action Script + CGI プログラミング

森 巧尚

<http://www2.netwave.or.jp/~light/>

FLASH 道場

<http://www.vivos.jp/flash/ne/index.html>

初心者の為の動くホームページ

<http://flash.obi.ne.jp/>

FLASH DREAM