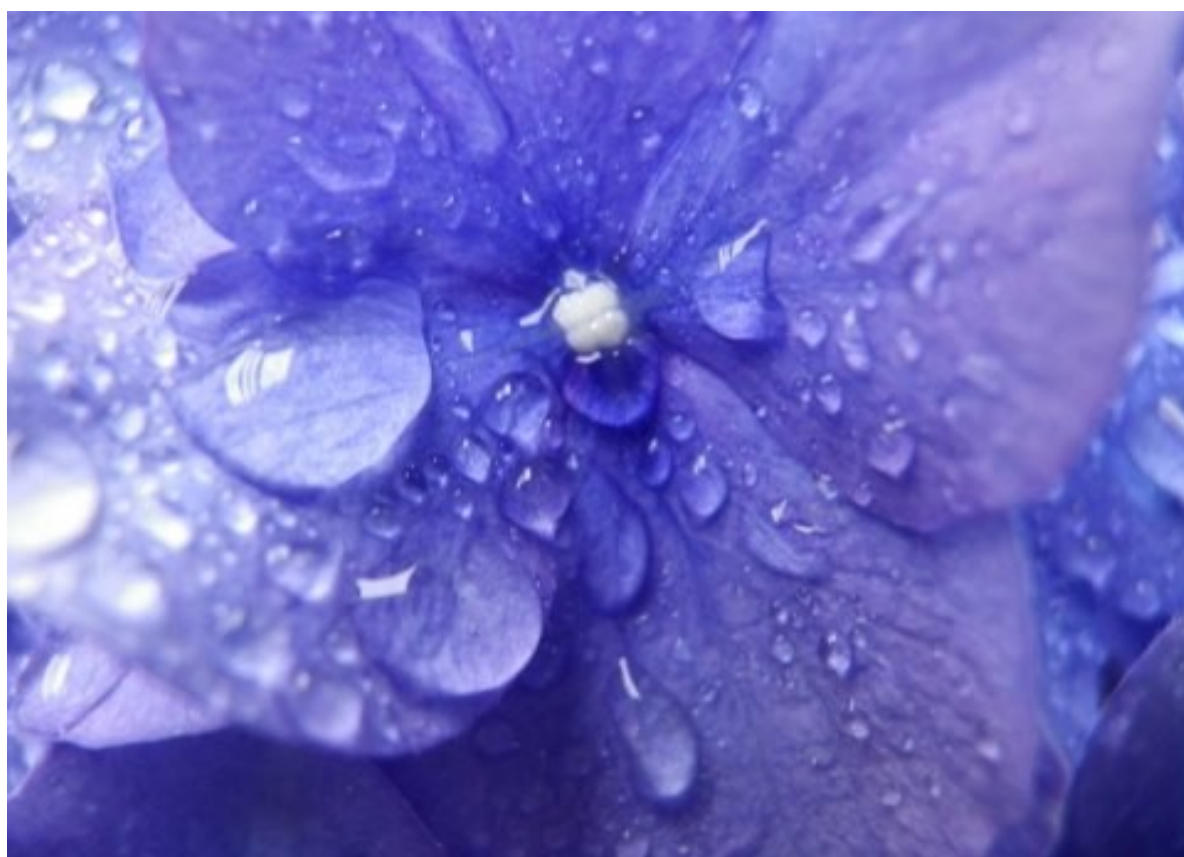


TSNET



TSC

ISSN 1884-2798

目次

巻頭言	jscripiter	...	3
たったひとつの冴えたやりかた	Y さ	...	4
Python の文法 第 5 回	機械伯爵	...	2
よしおさんとロボ太 最終回	海鳥	...	29
エアチェック・ポッドキャスティング	jscripiter	...	31
編集後記	jscripiter	...	37

!

! jscripiter

! 2 11 5 29

!

! 1c "

巻頭言

jscrip^{ter}

近 e er eibel 著 青木靖訳 C ders a rk
mall alk
めぐ 探求 (press 0 ms a 0)
ず L ピ ド チ 言
感 実行
部 確 ま (4 8 ペ ジ)
再入門 ツ
海鳥 順序 整 み 行 思 懐 思 出 出
ボ太 転載 後 転載 快諾 ず ざ ま 思
終 転載 私 せ 制 ず
了承 海鳥 感謝
s scrip Rub
出力 i u es R 出力
み込ませ 組み
i d uc へ音声
転送 速 効率
私 以 ジオ アチ ッ サ バ
i u es dcas アチ ッ
言 便利 使
ディア ア み込ませ i u es 同 能 使
(: 0 7 0)

wri e b

.

指 “チッ 効 ” コ ピュ 思 チ
改訂版 ま
ゆ く二 ゼ 和 有限 確定 完全情報> (※) み
? 悩み ち よ
ろ ろ 残 思 ま

※二 ゼ 和有限確定完全情報 ?
詳 web 検索 d(x_x)\ば
... 常 ざ 簡 言 ば 二 行 勝
他 敗

. 環境

確認
G U wk . .8, mawk . . MBC R 7
(i X DO 窓)行 ま
ち み シ
C U : C re Du 600 / . 6GHz
: GB (C - 00 DDR D-R M)
... ショボ ま ノ C
(;^^ゞ

. コ ピュ 思 チ 木 生 み

二 交互 “手” 打 局面 行 局面 打
“手” ま
コ ピュ 手 選ばせ ば良 よ
?
乱暴 番 始 (?) 局面 可能 手 ず 乱数 使 選択
ま ...
※effec .awk 使 fu c i r d() 定義 残
由 ま d(^^;
般 手順 ま
)コ ピュ 手番 局面
.)“手” 試 局面
局面 評価=評価点 算出
.)試 “手” 打 局面 戻
.)可能 “手”全 .)～.) 繰 返

)全 可能 “手” 試 番高 評価点 局面 “手” 選択
 ま
 基 全 可能性 系統 バッ ッキ 手 用 ま
 .) 局面 評価 ?
 局面 “手” 試 局面
 → 局面 手番 レ ヤ (相手) 番高 評価 “手”
 打ち 局面 生
 → 局面 手番 レ ヤ () 番高 評価 “手”
 打ち 局面 ...
 繰返 局面 (双 打 手 無) 終局面 レ ヤ 評価点
 算出 ま
 局面ま 遡 局面 評価点 ま
 番高 評価点 算出 テッ 行 実現 ま
 . .) 以 局面 局面(終局面) 達 合
 評価点 算出 呼び 値 渡
 . .) ば) ~) 同様 局面 手番 レ ヤ
 可能 “手” 全 後 局面 生 番高 評価点
 算出 後 局面 生 評価 . .), . .) 再帰 行
 実現
 局面 開 結 ま 枝葉 木 木 呼びま
 ま 木 生 打 手 定 み 呼びま (間
 キ パ レ ヤ 同 ま ね)
 ち み “チ ” “将棋”, “オセ ” 局面数 膨 終盤 除
 程度 間 終了 手数 限定 み 工夫 模
 様
 ... ち 優勢 不利 評価点 木 『途 』 算出
 難 ま
 余談 着手 評価 み ぶ 番困難 “ 碁”
 近 途 評価 諦め (^ ; 余 C U パワ 駆使 候補手 打 “終局”ま
 対局 シ ユレ ショ 結 勝率 高 着手 選ぶ ゆ テ
 カ 用 ア ゴ ズ 登
 まあア カ ズ番組 間 チャ ピオ 争 勝利 ちゃ コ ピュ シ テ
 登 ねえ

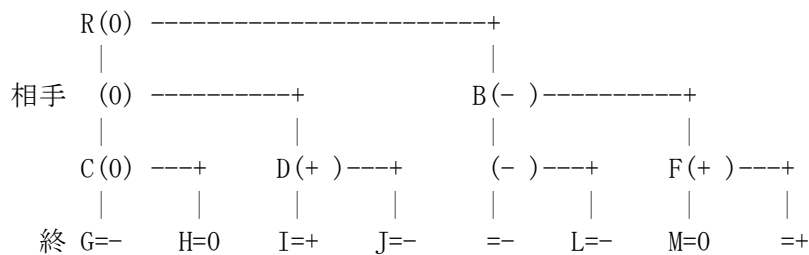
4. 局面 評価 ニ ッ 手

双 打 手 無 終局面 レ ヤ 評価点
 勝ち=+ , 引 =0, 負 =-
 シ

勝ち 少 引 手= 値 選ぶ
 逆 相手 勝ち 少 引 手= 小値 選ぶ
 想定 双 打 手 探索 定 ア ゴ ズ ニ ッ 手 呼びま
 局面 評価点 相手 選択 小評価点 定 ま

局面 R , B 局面 ま

相手 局面 C, D B , F ま
局面 C~F G~ 終局面 ま



例 ば C 局面 G=相手 勝ち=- , H=引 =0 合 値 選択
C 評価点 0 ま
同様 評価点 C, D 小 0 ま
局面 R 評価点 0 R 手 選 ば良
ま

ニ ッ 手 擬似 ジッ ま

```

-----
v=-searc ( 手番); # 現在 局面 評価
:

fu c i searc (w , eval,v)
{
  if(w 手番 終局面 ) re ur w 手番 評価点 算出 返却;
  現在 局面 状況 退避
  eval=-999; # 仮 -∞
  f r(w 手番 全 打 手 ){
    手 打ち 局面 生
    v=-searc (w 手番 相手); # v 評価点 セッ
    局面 退避 状況 戻
    if(v>eval) eval=v; # 評価点 更
  }
  re ur eval; # 評価点 返却
}

```

```

-----
      ジッ 相手 評価点 評価点 同
  勝ち=+ , 引 =0, 負 =-
      算出 符 反転 セ
  v=-searc (w 手番 相手, lev-)
      相手 手番 小 評価点 算出 ま
      ジッ シ ま
  ※ち み ネガ ッ ( ega-Max) 呼ぶ
  
```

・ $\alpha \beta$ 枝刈

ニ ッ 手 全 終局面 評価点 算出 ま

局面 勝 込み 無 』 局面 局
 面 『 ませ 石 数 差 終 勝負 ま 合 別
 ※ "オセ" 石 石 残 勝 ま 合 別
 負 石 石 残 勝 ま 合 別

局面 = 木 枝 狩 枝刈 呼びま
 程 例 C 評価点 0 ま

相手 (?) -----+
 |
 C(0) ----+ D(+ 以)-- / ---+
 | |
 終 G=- H=0 I=+ J ?

D 評価点 算出 め まず I 評価点 +
 手番 値 選ぶ J D 評価点 + 以 定 ま

点 C 評価点(0) < D 評価点(+ 以) 確定 評価点 0 確定 ま
 J 評価点 枝刈 β 枝刈 (β cu) 呼びま ま 合 C 評価
 点 0 " β 値" 呼びま

評価点 0 ま B 評価点 算出 ま

R(?) -----+
 |
 相手 (0) B(- 以)-- / ---+
 |
 (-)----+ F ?
 | |
 終 =- L=-

B 評価点 算出 め まず 評価点 -
 相手 手番 小値 選ぶ F B 評価点 - 以 定 ま

点 評価点(0) > B 評価点(- 以) 確定 R 評価点 0 確定 ま
 F 評価点 枝刈 α 枝刈 (α cu) 呼びま ま 合 評価
 点 0 " α 値" 呼びま

α β 枝刈 擬似 ジッ ま

 v=-searc (手番, -999, 999); # 現在 局面 評価
 # α 値=仮 $-\infty$, β 値=仮 $+\infty$
 :

```
fu c i searc (w ,alp a,be a, v)
{
  if(w 手番 終局面 ) re ur w 手番 評価点 算出 返却;
  現在 局面 状況 退避
```

```

f r(w 手番 全 打 手 ) {
  手 打ち 局面 生
  v=-searc (w 手番 相手, -be a, -alp a); # v 評価点 セッ
                                #  $\alpha$  値=-be a,  $\beta$  値=-alp a
  局面 退避 状況 戻
  if(v>alp a) alp a=v; # 評価点 更
  if(alp a>=be a) re ur alp a; # 枝刈 = 評価点 返却
}
re ur alp a; # 評価点 返却
}

```

```

-----
      ジッ 相手 手番 小 評価点 算出 際 符 反転 セ
      v=-searc (w 手番 相手, -be a, -alp a);
      ジッ シ ま
※ち み ネガア ア(ega- $\alpha$ ) 呼ぶ

```

実際 ニ ッ 手 版 α β 枝刈 版 比較

[#0]

```

( ) ( ) : COM
III I .
.
.
I I .
( ) ( ) : YOU
[L] [R]

```

状態 COM 手 め
ニ ッ 手 版: 評価点算出 4 8 90 , 約 0 秒
 α β 枝刈 版: 評価点算出 9 , 約 秒
結 ま

枝刈 効 ね
参 ま α β 枝刈 版

[#0]

```

( ) ( ) : COM
I I .
.
.
I I .
( ) ( ) : YOU
[L] [R]

```

状態 COM 手 め セ
評価点算出 67 74 , 約 60 秒
結 ま

. コ ピュ 思 チ 改訂版 effec .awk

遊び

参

版 更点 C U 思 チ 選択

begi er レ

版 思 チ

exper レ α β 枝刈 ジッ 採用 ま

思 間

B GI ブ ッ RCH R =4;

手目 探索 行

整 ま

ち み

手

exper レ

後手 C U 勝

ま ...

(;^^ゞ

6. 他

effec .awk ソ

著 権

ま

転載 再配布 改造 消去 由

(ま 素晴 改造 加 後 C 宛 ませ)

ま ソ 使用 損 生 損 対 切 責任 負

ねま

7. effec .awk

effec .awk 以 示 ま

[effect2.awk]

```
## Even-Odd Effect rev.2 written by Y さ
```

```
function rnd(N){ return int(N * rand()); } ## 乱数
```

```
BEGIN{
```

```
    srand(); ## 乱数の初期化
```

```
    # ゲームモード決定
```

```
    gm=GAMEMODE;
```

```
    if(gm!="0" && gm!="1" && gm!="2"){
```

```
        print " <Solitaire> - 0"
```

```
        print " <Match-up>"
```

```
        print " first move - 1"
```

```
        print " 2'nd - 2"
```

```
        printf("\nWhich?[0-2]> ");
```

```
        do{
```

```

    gm=""; getline gm; gm=tolower(gm);
}while(gm!="0" && gm!="1" && gm!="2");
}
GAMEMODE=gm+0;
print "¥n";

# CPU レベル決定
SEARCHSTART=4;
gm=CPULEVEL;
if (GAMEMODE!=0 && gm!="0" && gm!="1") {
    print " <CPU LEVEL>"
    print "    beginner - 0"
    print "    expert   - 1"
    printf("¥nWhich?[0-1]> ");
    do{
        gm=""; getline gm; gm=tolower(gm);
    }while(gm!="0" && gm!="1");
    CPULEVEL=gm+0;
    print "¥n";
}

## Turn;      ## 順番 0:人, 1:COM
Turn=(GAMEMODE!=1)?(0):(-1);  #ゲーム開始時に+1
Times=0; #回数
## Sc[2];      ## 得点 0=人, 1=COM
Sc[0]=Sc[1]=0;
## Bit[9];     ## 指 [人<1:左, 3:右>, COM<9:左, 7:右>]
Bit[1]=Bit[9]=0;
Bit[3]=Bit[7]=1;
## 場所判別 他
pNo[1, 1]=7;  pNo[1, 0]=9;
pNo[0, 0]=1;  pNo[0, 1]=3;
cnvPos[10]=pNo[0, 1];  cnvPos[11]=pNo[1, 1];  cnvPos[12]=pNo[1, 0];
cnvPos[30]=pNo[0, 0];  cnvPos[31]=pNo[1, 0];  cnvPos[32]=pNo[1, 1];
cnvPos[70]=pNo[1, 0];  cnvPos[71]=pNo[0, 0];  cnvPos[72]=pNo[0, 1];
cnvPos[90]=pNo[1, 1];  cnvPos[91]=pNo[0, 1];  cnvPos[92]=pNo[0, 0];
mv["LR"]=mv[12]=10;  mv["LU"]=mv[14]=11;  mv["LX"]=mv[15]=12;
mv["RL"]=mv[32]=30;  mv["RU"]=mv[36]=31;  mv["RX"]=mv[35]=32;
mv[78]=70;  mv[74]=71;  mv[75]=72;
mv[98]=90;  mv[96]=91;  mv[95]=92;

## ゲームメイン
for (;){
    #状況表示
    ++Times;
    dispHands(0);
    #GAME OVER / 1回休み判定
    if(gameOver(Turn)>0) exit;
    if (GAMEMODE!=0) if(++Turn>1) Turn=0;

```

```

    setSkip();

    do{
        #移動指定 -> 影響判定
        sw=(Turn==0)?(man()): (com());
        ev=canMove(sw);
        if(Turn==0 && ev<0){
            printf(" No effect, retry."); pushEnter();
            dispHands(0);
        }
    }while(ev<0);
    #変化
    change(sw, ev);
}
exit;
}
END{
    print "¥n";
    if(finish(0)) printf("      Congratulations !! (SCORE:%05d0)¥n", Sc[0]);
    else{
        printf("      ***** GAME OVER *****¥n");
        printf("      ... YOU LOSE (SCORE:%05d0)¥n", Sc[0]);
    }
    print "";
}
function pushEnter( tmp){ printf("<PUSH ENTER>"); getline tmp; }
function yorn( yn)
{
    do{ yn=""; getline yn; yn=toupper(yn); }while(yn!="Y" && yn!="N");
    return yn;
}

##
## 状況表示
##
function dispHands(sw, d, ev, i)
{
    delete d;
    delete ev;
    makeDispTbl(sw, d, ev);

    print "";
    for(i=((GAMEMODE==0)?(6):(0)); i<=8; ++i) print d[i];
    print "";
}
function makeDispTbl(sw, d, ev, s0, s1)
{
    #0...5...0...5...0...5...0...
    d[0]=" <s1> .<s2> <s3> . <s4>";
    d[1]="      :      : ";
}

```

```

d[2]="<s5> : zzzzz xxxxx : COM [cccc0]";
d[3]="      . zzzzz xxxxx .";
d[4]="      .      .";
d[5]=d[4];
d[6]="      . @@@@ &&&& .";
d[7]="<s0> : @@@@ &&&& : YOU [yyyy0]";
d[8]="      [L]      [R]      ";

if (sw>=90) {      s0=      mkBitStr (Bit[9],1);
                  s1=      mkBitStr (Bit[9],0);
} else if (sw>=70) { s0=rev(mkBitStr (Bit[7],1));
                  s1=rev(mkBitStr (Bit[7],0));
} else if (sw>=30) { s0=      mkBitStr (Bit[3],0);
                  s1=      mkBitStr (Bit[3],1);
} else if (sw>=10) { s0=rev(mkBitStr (Bit[1],0));
                  s1=rev(mkBitStr (Bit[1],1));
}

#0...5...0...5...0...5...0...
if (sw==90) {      d[2]="<s5> : zzzzz" s0 "      : COM [cccc0]";
                  d[3]="      . zzzzz" s1 "      .";
}
if (sw==70) {      d[2]="<s5> :      " s0 "xxxxx : COM [cccc0]";
                  d[3]="      .      " s1 "xxxxx .";
}
if (sw==30) {      d[6]="      . @@@@" s0 "      .";
                  d[7]="<s0> : @@@@" s1 "      : YOU [yyyy0]";
}
if (sw==10) {      d[6]="      .      " s0 "&&&& .";
                  d[7]="<s0> :      " s1 "&&&& : YOU [yyyy0]";
}

#0...5...0...5...0...
if (sw==31 || sw==91) { d[4]="      .      " s0 "      .";
                      d[5]="      .      " s1 "      ."; }
if (sw==32 || sw==92) { d[4]="      .      " s0 "      .";
                      d[5]="      .      " s1 "      ."; }
if (sw==12 || sw==72) { d[4]="      .      " s0 "      .";
                      d[5]="      .      " s1 "      ."; }
if (sw==11 || sw==71) { d[4]="      .      " s0 "      .";
                      d[5]="      .      " s1 "      ."; }

#0...
cnvHandStr (7, "zzzzz", sw, 1, d, 3, 2);      cnvHandStr (9, "xxxxx", sw, 0, d, 3, 2);
cnvHandStr (1, "@@@@@", sw, 1, d, 6, 7);      cnvHandStr (3, "&&&&", sw, 0, d, 6, 7);

setAllEval (ev);
# YOU
sub (/s0/, ev[0], d[7]);
sub (/yyyy/, sprintf ("%04d", Sc[0]), d[7]);
# COM
sub (/s1/, ev[1], d[0]);
sub (/s2/, ev[2], d[0]);
sub (/s3/, ev[3], d[0]);
sub (/s4/, ev[4], d[0]);
sub (/s5/, ev[5], d[2]);

```

```

    sub(/cccc/, sprintf("%04d", Sc[1]), d[2]);
}
function cnvHandStr(hand, regexp, dispSw, revSw, dTbl, ln1, ln2, s0, s1)
{
    #0...
    if(unpackUpper(dispSw)==hand && unpackLower(dispSw)!=0) s0=s1=" ";
    else{
        s0=mkBitStr(Bit[hand], 0);
        s1=mkBitStr(Bit[hand], 1);
        if(revSw){ s0=rev(s0);
                  s1=rev(s1); }
    }
    sub(regexp, s0, dTbl[ln1]);
    sub(regexp, s1, dTbl[ln2]);
}
function mkBitStr(n, pos, s, i)
{
    if(pos==0){
        for(i=1; i<=n; ++i) s=s "1";
        return substr(" " s " ", 1, 5);
    }
    if(pos==1)
        return (n==5)?(" - 5 ") : (sprintf("( %d )", n));
}
function rev(src, dst, len)
{
    for(len=length(src); len>0; --len) dst=dst substr(src, len, 1);
    if(substr(dst, 1, 1)=="") dst="(" substr(dst, 2);
    if(substr(dst, 5, 1)=="") dst=substr(dst, 1, 4) ")";
    return dst;
}
function setAllEval(ev )
{
    delete ev;
    ev[0]=evenOddStr(Bit[1], Bit[3]);
    ev[1]=evenOddStr(Bit[3], Bit[7]);
    ev[2]=evenOddStr(Bit[1], Bit[7]);
    ev[3]=evenOddStr(Bit[3], Bit[9]);
    ev[4]=evenOddStr(Bit[1], Bit[9]);
    ev[5]=evenOddStr(Bit[7], Bit[9]);
}
function evenOdd(b1, b2){ return ((b1+b2)%2); }
function evenOddStr(b1, b2){ return ((evenOdd(b1, b2))?("od")?("ev")); }

##
## 影響判定
##
function canMove(sw){ return canMoveFromTo(unpackUpper(sw), cnvPos[sw], Turn); }
function canMoveFromTo(from, to, who)

```

```

{
  # 初手
  if(Times<=2)
    if(other(who,to)) return -1;
  # 奇数
  if(evenOdd(Bit[from],Bit[to])) return ((Bit[from]<5)?(1):(-1));
  # 偶数
  if((Bit[to]<1) || (Bit[from]>=4)) return -2;
  return 2;
}
function other(who,to, oth){
  oth=another(who);
  if(to==pNo[oth,0] || to==pNo[oth,1]) return 1;
  return 0;
}
function another(who){ return ((who==0)?(1):(0)); }
function total(who){ return (Bit[pNo[who,0]] + Bit[pNo[who,1]]); }
function finish(who){ return (total(who)==10); }
function noEffectLR(l,r){ return (Bit[l]==4 && Bit[r]==4); }
function noEffect(who){ return noEffectLR(pNo[who,0],pNo[who,1]); }
function gameOver(who)
{
  if(finish(who)) return 3;
  if(finish(another(who))) return 2;
  if(noEffect(0) && (GAMEMODE==0 || noEffect(1))) return 1;
  return 0;
}
function setSkip()
{
  if(isSkip(Turn)){ Turn=another(Turn); printf(" Can't effect."); pushEnter(); }
}
function isSkip(who)
{
  return ((noEffect(who) && evalOddEven(another(who))==4)?(1):(0))
}
function evalOddEven(who){ return evalOddEvenLR(pNo[who,0],pNo[who,1]); }
function evalOddEvenLR(l,r, minOE, maxOE)
{
  minOE=Bit[getMinPos(l,r)]%2;
  maxOE=Bit[getMaxPos(l,r)]%2;
  if(minOE==1 && maxOE==0) return 1; # 奇 < 偶
  if(minOE==0 && maxOE==1) return 2; # 偶 < 奇
  if(minOE==1 && maxOE==1) return 3; # 奇 <= 奇
  if(minOE==0 && maxOE==0) return 4; # 偶 <= 偶
  return 0; # err
}
function getMinPos(l,r){ return ((Bit[l]<Bit[r])?(l):(r)); }
function getMaxPos(l,r){ return ((getMinPos(l,r)==l)?(r):(l)); }
function change(sw,ev, from,to)

```

```

{
    from=unpackUpper (sw);
    to=cnvPos[sw];
    countUp (ev, from, to);

    #得点加算
    ++Sc[Turn];
    if (ev==2 && other (Turn, to)) Sc[Turn] += 2;
    if (finish (Turn))
        Sc[Turn] += 50;
    else if (Bit[pNo[Turn, 0]] == 5 || Bit[pNo[Turn, 1]] == 5)
        Sc[Turn] += 5;
}

function countUp (ev, from, to)
{
    # 奇数
    if (ev == 1) {
        if (Bit[from] < 5) ++Bit[from];
    }
    # 偶数
    if (ev == 2) {
        if (Bit[to] >= 1 && Bit[from] < 4) {
            --Bit[to];
            Bit[from] += 2;
        }
    }
}

function cnvMoveNo (from, to, dir, n)
{
    n = from * 10;
    for (dir = 0; dir <= 2; ++dir)
        if (cnvPos[n + dir] == to) return n + dir;
    return -1; # err
}

function find (bit, who, dir)
{
    for (dir = 0; dir <= 1; ++dir)
        if (Bit[pNo[who, dir]] == bit) return pNo[who, dir];
    return -1; # not found
}

function findOEPos (dir, tgtOE, who, list, p, st, iim)
{
    delete list;
    list[0] = getMinPos (pNo[who, 0], pNo[who, 1]);
    list[1] = findOtherPos (list[0], pNo[who, 0], pNo[who, 1]);
    st = (dir == 1) ? (0) : (1);
    lim = (dir == 1) ? (2) : (-1);
    for (p = st; p != lim; p += dir)
        if (Bit[list[p]] % 2 == tgtOE) return list[p];
}

```

```

    return -1; # not found
}
function findOtherPos(tgt, l, r)
{
    if(tgt==l) return r;
    if(tgt==r) return l;
    return -1; # not found
}
function move(from, to) { return moveMain(cnvMoveNo(from, to), from, to); }
function moveBySW(sw) { return moveMain(sw, unpackUpper(sw), cnvPos[sw]); }
function moveMain(sw, from, to)
{
    dispHands(sw);
    printf("[#%02d] ... %s to %s.", Times, dirName(from), dirName(to));
    i[r(sw)]し化(sw, fon move太況e太/found

```



```

## COM main
##
function com()
{
    if(CPULEVEL==0) return beginner();
    return expert();
}
function beginner( ev,minP,maxP,yEv,yMin,yMax,me,you)
{
    me=1; you=0;
    ev=evalOddEven(me);
    minP=getMinPos(pNo[me,0],pNo[me,1]);
    maxP=findOtherPos(minP,pNo[me,0],pNo[me,1]);
    yEv=evalOddEven(you);
    yMin=getMinPos(pNo[you,0],pNo[you,1]);
    yMax=findOtherPos(yMin,pNo[you,0],pNo[you,1]);

    if(yEv!=4)
        if((Bit[minP]==3 && Bit[maxP]==5) || (Bit[minP]==4 && Bit[maxP]==4))
            return move(minP, findOEPos(1,1,you));
    if((Bit[minP]==3 && (Bit[maxP]==4 || Bit[maxP]==3)) && (Bit[yMin]==3 && Bit[yMax]==5))
        return move(minP,yMin);
    if(Bit[minP]==2 && Bit[maxP]==5) {
        if(yEv!=3 && Bit[findOEPos(1,0,you)]>0)
            return move(minP, findOEPos(1,0,you));
        return move(minP,maxP);
    }
    if(Bit[minP]==2 && Bit[maxP]==4 && Bit[yMin]==2 && Bit[yMax]==5)
        return move(minP,maxP);
    if(Bit[minP]==3 && Bit[maxP]==3 && Bit[yMin]==3 && Bit[yMax]==3)
        return move(minP,yMin);
    if(Bit[minP]==1 && Bit[maxP]==1 && Bit[yMin]==1 && Bit[yMax]==1)
        return move(minP,yMin);

    if(Times>2 && ev!=4 && yEv!=4) {
        if(Bit[minP]%2)
            return move(minP,findOEPos(1,1,you));
        if(Bit[maxP]<5 && Bit[maxP]%2)
            return move(maxP,findOEPos(1,1,you));
    }

    if(Bit[yMin]==4 && Bit[yMax]==5)
        if((Bit[minP]==0 || Bit[minP]==2) && (Bit[maxP]==0 || Bit[maxP]==2 || Bit[maxP]==4))
            return move(minP,yMin);

    if(Bit[yMin]==4 && Bit[yMax]==5) {
        from=find(3,1); if(from>0) return move(from,yMax);
        from=find(1,1); if(from>0) return move(from,yMax);
    }
}

```

```

    if(ev==1 || ev==2) {
        from=findOEPos(1, 0, me);
        to=findOtherPos(from, pNo[me, 0], pNo[me, 1]);
    }else{
        from=minP;
        to=maxP;
    }
    return move(from, to);
}
function expert( mv, alpha, beta, cnt, v, p, lev)
{
    #0...5
    printf("%n    ... thinking"); srand(); st=srand();
    mv=0; alpha=-999; beta=999;
    cnt=((Times>SEARCHSTART)?(setApproach(1, 0, (lev=99))):(0));
    if(cnt>0) backUp(lev);
    for(p=0; p<cnt; ++p) {
        tryMove(approach[lev, p]);
        v=-search(0, lev-1, -beta, -alpha);
        restore(lev);
        if(v>alpha) { alpha=v; mv=unpackUpper(approach[lev, p]); }
        if(alpha>=beta) break;
    }
    srand(); ed=srand(); printf(" (%dsec)%n", ed-st);
    if(mv) return moveBySW(mv);
    return beginner();
}
function search(who, lev, alpha, beta, cnt, v, p, oth)
{
    if((v=gameOver(who))>0) return phaseEvaluate(v, lev);
    oth=another(who);
    cnt=setApproach(who, oth, lev);
    if(cnt==0) return search(oth, lev, alpha, beta);
    backUp(lev);
    for(p=0; p<cnt; ++p) {
        tryMove(approach[lev, p]);
        v=-search(oth, lev-1, -beta, -alpha);
        restore(lev);
        if(v>alpha) alpha=v;
        if(alpha>=beta) return alpha;
    }
    return alpha;
}
function phaseEvaluate(v, lev)
{
    if(v==3) return 500+(99-lev); # 手番 who の勝ち
    if(v==2) return -500-lev;     # 手番 another(who) の勝ち = 手番 who の負け
    return 0; # 双方手詰まり = 引分
}

```

```

function tryMove(tgt, sw)
{
    sw=unpackUpper(tgt);
    countUp(unpackLower(tgt), unpackUpper(sw), cnvPos[sw]);
}

# bkBit[レベル, 1~9]    (レベルはスタック代り)
function backUp(lev)
{
    bkBit[lev, 7]=Bit[7];    bkBit[lev, 9]=Bit[9];
    bkBit[lev, 1]=Bit[1];    bkBit[lev, 3]=Bit[3];
}
function restore(lev)
{
    Bit[7]=bkBit[lev, 7];    Bit[9]=bkBit[lev, 9];
    Bit[1]=bkBit[lev, 1];    Bit[3]=bkBit[lev, 3];
}

# approach[レベル, 添字] (レベルはスタック代り)
function setApproach(who, oth, lev, cnt, sw, p, d, lr, dLast, hLast, ev)
{
    cnt=0;
    dLast=((Bit[pNo[oth, 0]]==Bit[pNo[oth, 1]])?(1):(2));
    hLast=((Bit[pNo[who, 0]]==Bit[pNo[who, 1]])?(0):(1));
    for(d=0; d<=dLast; ++d)
        for(lr=0; lr<=hLast; ++lr){
            sw=pack(pNo[who, lr], d);
            ev=canMoveFromTo(pNo[who, lr], cnvPos[sw]);
            if(ev>0) approach[lev, cnt++]=pack(sw, ev);
        }
    return cnt;
}
function pack(upper, lower){ return (upper*10+lower); }
function unpackUpper(src){ return int(src/10); }
function unpackLower(src){ return (src%10); }

```

(: 0 7)

Python

TSNET スクリプト通信版 草稿 第5回

6-2-2-2-3-2 専用の順序列型

収容 容 ま 順序列型 コレ ショ オブジ 字列 扱 's r'

バ 列 扱 'b es' 'b earra' ま

歴史 ば 以 バ 列 字列 扱 ま

U ic de 字 扱 U ic de 字列型 登 CII バ 列 旧

字列型 U ic de U ic de 字列 扱 区 ま

デ 構造 基 類 不 現在 字列 扱

バ 列 扱 形 類

始め 類 頂 ば良 思 ま 旧

愛用 特 部 慎重 確認

6-2-2-2-3-2-1 文字列(str)

■クラス名 : 'str'

字列 's r'

's r' 型オブジ (以後 字列) 字 専用 扱 不 型 順序列型コレ ショ

コレ ショ 字列 特徴 不 1 字 字列

字 1 字取 出 1 字 字列 扱 関数(' rd' 関数) ま

字 ず 字列 ま

```
>>> T
(10, 20, 30) # 【例えばタプルで……】
>>> T[1]
20 # 【要素は整数】
>>> S
'あいう' # 【しかし文字列は……】
>>> S[1]
'い' # 【要素も文字列】
```

字列オブジ 不 汎用順序列オブジ 'uple' ソッド 演算子

漏 無 使用 ま

```
>>> S = "あいうえお abcde"
>>> S + "xxx"
'あいうえお abcdexxx'
>>> S * 2
'あいうえお abcde あいうえお abcde'
>>> S[2:8]
'うえお abc'
```

字列 不 型 コレ ショ 字列 直接操 別 字列

ませ

- 同 効 得 め 以 ち 取 ま
 1. 文字列 換 操 'j i' ソッド ぎ直 文字列
 再 入
 2. 文字列 換 ソッド 得 文字列 再 入
 ば 数 "abcde" "abXde" 換 感

```
>>> S = "abcde"
>>> SL = list(S)          # 【リストに変換】
>>> SL
['a', 'b', 'c', 'd', 'e']
>>> SL[2] = 'X'
>>> SL
['a', 'b', 'X', 'd', 'e']
>>> S = ''.join(SL)       # 【文字列に連結】
>>> S
'abXde'
>>> S = "abcde"
>>> S = S.replace('c', 'X') # 【入れ替えメソッド】
>>> S
'abXde'
```

文字列オブジ 字 加工 文字列 返 ソッド ま

※ただし、Pythonは「文法として」 表 (regular expression) ポートしていま ん 表
 Pythonで 用 場 は、 の're'モ ール インポートして 用して ださい

【 】
 Python2 の「文字列」は「文字」「イット」という記 で っていました、Python3
 の「文字列」は、文字通り「文字」で Python2 で文字列型 イット ード に 用していた
 方は、Python3では後の'bytes'型 用して ださい
 また、一 ット 算 であ '%'も、 時点の最 であ 'Python 3.2.1'ではまだ 用で ようで
 、 に され と 定されてい ので、より 用 に んだ'format'メソッド 用 ように
 されていま っ て では、一 ット 算 .についての は し、 しい であ
 「メソッド 用いた 一 ット」のみ とにしま

6-2-2-2-3-2-1-1 一 ット用メソッド

オ ッ 関 ソッド 扱 ま
 オ ッ 式 及び 式 従 章 組み 指 ま
 以 '%'演算子 行 操 ソッド 現力 強力 ま
 確 ま
 強力 ぎ 全部 度
 オ ッ ? ソッド 紹介
 実際 式 基礎 説 思 ま
 詳細 説 思 ま

■ 一 ット用メソッド

: S0.format(arg1[, arg2, ...]) S1
 動 : "S0"に かれた に っ て、"arg1" の引数 した文字列"S1" キー ード引数も 用

 : S0.format map(mapping) S1
 動 : 'format'の引数 エクトでまと たもの け ー ン(ver.3.2より)

も も ー ットって の

※ 項目 文字列 オ ッ 行 説 不

文字列 オ ッ 構 基礎 み

文字列 オ ッ 例

☆ イ

氏名 ()

年齢 ()

住 ()

電話 ()

例 実際 () 様 よ 氏名 年齢 項目 字

化 ませ

印刷 各 配 空 埋め 使 よ

化 部 空 指定 容

'f rma ' ソッド 式 当 文字列 与 章 完 言 ま ソッド

容 直接指定 合 与 デ 加工 容 出

合 ま

基礎 直接指定 説 ま

十 便利 まず ち 使 み

詳細 加工 ……桁数 揃 整形 デ 整数化

説 ま

ろ 例 試 みま よ

```
>>> S=""☆プロ イール
... 名 {0}
... {1}
... 所 {2}
... {3} ""
>>> sk = S.format('機械伯爵','れた','時間 ','いよ')
>>> print(sk)
プロ イール
名 機械伯爵
    れた
所 時間
    いよ
```

6-2-2-2-3-2-1-1-1 文字列 ー ット 文 編

'f rma ' ソッド 式 文字列 '{ }'(ブレ) ま 部 引数 指定 部

入 替 ま

'{ }' 数字 入 ま

数字 合 引数 '{0}' 順 '{ }','{ }'.. ま

合 キ ワ ド引数 指定 ま

字 文字列 条件 数 同

まず 例 みま よ

```
>>> "{0}-{1}-{2}".format('a','b','c')
'a-b-c'
>>> "{0}-{1}-{0}".format('a','b','c')
```

```
'a-b-a'
>>> "{one}-{two}-{three}".format(three='a',two='b',one='c')
'c-b-a'
>>> "{one}-{one}-{two}".format(three='a',two='b',one='c')
'c-c-b'
```

引数 キ ワ ド引数 順番 ま ませ 2 以 引用 引数 使用
引数 構 ませ

```
>>> "{key}-{0}-{1}".format('a','b', key='c')
'c-a-b'
```

位置引数 キ ワ ド引数 混合 合 関数 引数 於 キ ワ ド引数 常 引数 後
有効 式 適用 順序 当 関係 ませ
'f rma _map' ソッド 使 想像 思 ま

```
>>> d = {'a':'AAA', 'b':'BBB', 'c':'CCC'}
>>> "{c}-{b}-{a}".format(**d)
'CCC-BBB-AAA'
>>> "{c}-{b}-{a}".format_map(d)
'CCC-BBB-AAA'
```

※ エクトについては後の「型 エクト」の で し ま

引数 字列 現 (珍) 全 オブジ 取 ま
数字 まま ば's r' オブジ へ ちゃ

```
>>> '{0}-{1}-{2}'.format(3.14,(lambda x:x), str)
"3.14-<function <lambda> at 0x00C71390>-<class 'str'>"
```

カ オブジ ' __s r__' ソッド ' __repr__' ソッド 実装 合 指定
字 現 字列 現 ま

```
>>> class X:
...     def __str__(self):return 'xxx'
...
>>> x = X()
>>> x
<__main__.X object at 0x00C72050>
>>> print(x)
xxx
>>> "{0}-{1}-{2}".format(x,x,x)
'xxx-xxx-xxx'
```

式 換 柔軟 例 ばコレ ショ 型 引数 ツ
可能

```
>>> "{0}".format("abc")
'abc'
>>> "{0[1]}".format("abc") #【アイテム 】
'b'
>>> "{0[a]}".format({'a':100,'b':200})
'100'
>>> class C:pass
...
```

July 2011

TSNET スクリプト通信 4.1

```
>>> o = C()
>>> o.x = 'xxx'
>>> "{0.x}".format(o)          # 【      乚…】
'xxx'
```

ブレ 2連 ケ (オ ツ 除外) ま

```
>>> "{ {, {0}, } }".format(100)
'{ ,100, }'
>>> "{ {, {0}, } }".format(100)
```

利用 式 式 (オ ツ) ま

```
>>> "{0[{0}]}" .format(1) .format('abcde')
'b'
```

 $\{ \}$


```
'7/2'
```

‘a’ 換 例 語 使用 ば若干 易 ませ

```
>>> j = "あいうえお"
>>> j
'あいうえお'
>>> print(j)
あいうえお
>>> "{0!r}".format(j)
"'あいうえお'"
>>> "{0!s}".format(j)
'あいうえお'
>>> "{0!a}".format(j)
"'\u3042\u3044\u3046\u3048\u304a'"
```

● 式設定

主 数値 示 関係 式設定 以 順番 設定 ま

{ イ ド名 : (詰め物) 位置 符 数 示 数字 ゼ 示幅 3桁
区切 コ 精度 数値型 }

式設定 説 ま

まず 示幅 0以外 正 整数 (数) 低 示幅 ま
ま 数値 桁 ろ ぺ 確保
例 みま よ

```
>>> "{0}|".format(123.45)
'|123.45|'
>>> "-{0:10}-".format(123.45)
'|        123.45|'
```

ぺ 関

位置 左揃 (‘<’) 右揃 (‘>’) イ ド名 ぐ横 (詰め物) 位置
種類 指定 ま 央揃 (‘^’) 符 数字 間 開 (‘=’) 4

```
>>> "{0:10}|".format(-123.45)
'|        -123.45|'
>>> "{0:<10}|".format(-123.45)
'| -123.45    |'
>>> "{0:>10}|".format(-123.45)
'|        -123.45|'
>>> "{0:^10}|".format(-123.45)
'| -123.45    |'
>>> "{0:=10}|".format(-123.45)
'| -        123.45|'
```

位置揃 指定

開 ぺ

字 置 換

ま

```
>>> "{0:a<10}|".format(-123.45)
'| -123.45aaa|'
>>> "{0:@<10}|".format(-123.45)
'| -123.45@@@|'
>>> "{0:<^10}|".format(-123.45)
```

```
'|<-123.45<<|'
```

※最後の例のように、指定の記号も用で、ま、しいのでおでまん
のルトは右えで、右えでも詰指定場は'>'の指定要にりま

符 '+' , '-' , ' ' (空白) 指定 ま
'+' 指定 ず ナ 示 ま
'-' 指定 負 数 ナ ま (デ オ)
' ' 指定 正 数 ペ 示 ま

```
>>> "|{0}|".format(100)
'|100|'
>>> "|{0:~}|".format(100)
'|+100|'
>>> "|{0:-}|".format(100)
'|100|'
>>> "|{0: }|".format(100)
'| 100|'
```

数字 ゼ 字 '0' 詰め物

```
>>> "|{0:10}|".format(100)
'|          100|'
>>> "|{0:010}|".format(100)
'|0000000100|'
>>> "|{0:^010}|".format(100)
'|0001000000|'
>>> "|{0:<010}|".format(100)
'|1000000000|'
```

2 例 位置 緒 使 妙 感 注 ま よ
(使 別)

3桁区切 コ ' ' 字 桁数 数字 示 3桁毎 コ 示 ま

```
>>> "|{0}|".format(1000000)
'|1000000|'
>>> "|{0:,}|".format(1000000)
'|1,000,000|'
```

精度 数値 示桁数 指定 ま
精度 指定 合 浮 小数点数('f' a ') 使用 (整数 含め 精度
設定 合 数字型 'f' 及び 'F' オ ショ 使用)
精度 '.' 後 整数 示 ま

```
>>> "{0}".format(123.45)
'123.45'
>>> "{0:.4}".format(123.45)
'123.5'
>>> "{0:.4}".format(123.44)
'123.4'
>>> "{0:.3}".format(123.44)
'1.23e+02'
```

数字 切 捨 四捨五入 示 ま
後 小数点部 無 め 科学計算 概数 示 式 換 ま
数値以外 指定 合 精度 示 ペ ま

```
>>> "{0:.3}".format("abcde")
'abc'
```

数値型 数値 換 様 示型 式
ち'b','d',' ','x','X' 引数 整数 合 み有効
'd' 普 十 示 (使用 ませ)
'b' 整数 二 ま
' ' 整数 八 ま
'x' 'X' 整数 十六 ま
二 八 十六 合 数 位置 '#' 入 '0x' 付
ま ('X' ア ア ッ 部 字 ま)

```
>>> "{0}".format(200)
'200'
>>> "{0:b}".format(200)
'11001000'
>>> "{0:#b}".format(200)
'0b11001000'
>>> "{0:o}".format(200)
'310'
>>> "{0:#o}".format(200)
'0o310'
>>> "{0:x}".format(200)
'c8'
>>> "{0:#x}".format(200)
'0xc8'
>>> "{0:X}".format(200)
'C8'
>>> "{0:#X}".format(200)
'0XC8'
```

'e' ' ' 強制 科学計算用概数 示 ま ('e' ' ' 違 位 示 字/小 字
示 違)
'f' 'F' 固定小数点数 示 ま (精度 指定 シ デ オ 従 ま
)

'e',' ','f','F' 精度 指定 合 有効数字 桁数 小数点以 桁数 精度
ま 十 注

'g' 'G' 科学計算 示 固定小数 示 桁数 ま
ま 桁数 科学計算 示 少 ば固定小数点 示 普 示 採用 ま
('G' 科学計算 示 ' ' 採用)
'g' デ オ
精度 有効数字 桁数 指 ま

' ' 'g' ぼ同 (環境設定 限)

```
>>> "{0:.4}".format(123.56)
'123.6'
>>> "{0:.4f}".format(123.56)
```

```
'123.5600'
>>> "{0:.4F}".format(123.56)
'123.5600'
>>> "{0:.4e}".format(123.56)
'1.2356e+02'
>>> "{0:.4E}".format(123.56)
'1.2356E+02'
>>> "{0:.4g}".format(123.56)
'123.6'
>>> "{0:.4G}".format(123.56)
'123.6'
```

"%" 字 百 率 ま (精度 小数点以 桁数 指 ま)

```
>>> "{0:.2%}".format(0.5)
'50.00%'
```



式設定 引数 指定 可能

```
>>> "{0:{1}}".format(100, "#b")
'0b1100100'
>>> "{0:{1[1]}}".format(100, ("#b", "#o"))
'0o144'
```

オブジ 特別 式 使用 ま
例 付 刻 扱 オブジ 特別 式設定 適用例

```
>>> import datetime
>>> d = datetime.datetime(2011, 7, 31)
>>> "{:%Y/%m/%d}".format(d)
```

' 0 /07/ '

(: 0 7)

ボ太

海鳥

「温・特技」



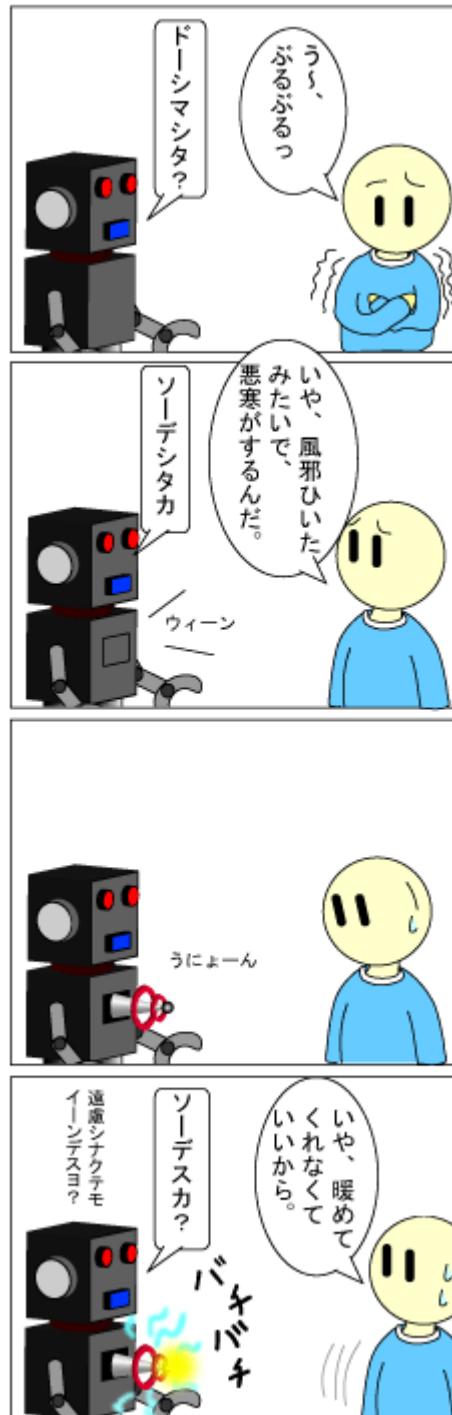
いや、便利だろうけれども。

(転 載 : 0 7 0)

ボ太

海鳥

「暖・特技」



心も体もすぐポカポカ！（たぶん80度くらいに）

(: 0 7 0)

アチ ッ ポッドキャ ティ

- エアチェック iTunes の Podcast に -

jscripiter

1. iTunes の 場と Podcasting

pple I(pple C mpu er 976) 数 パ ソナ コ ピュ ティ
 s - C 言葉 ティ ブ ジョブズ 身 唱
 突入 既 pple 社名 C mpu er 言葉 007 消 アッ
 s - C ち C(Mac) アッ 素 過ぎ 既定路線 突 走

歴史 ば ソ M -D0 / i d ws IBM C 990 コ ピュ ティ
 衆化 間違 980 半 C GUI 取 込 駆 割
 アッ ティ ブ ジョブズ 離 脇 ぎ

ジョブズ 戻 紀 入 アッ i d 音楽 レ ヤ i u es ア
 ケ ショ 生み出 00 i u es Music re 音楽 オ 販売 始め
 i u es i d ws サボ アッ 快 撃 始ま 007 i e 0 0
 i ad 登 せ 驚異 遂

アッ Mac C カ 音楽 ディア レ ヤ カ
 音楽 ビデオ オ 販売業 i e ケ カ i e
 ケ デジカ Face ime ビデオ 話 シ eb ブ ウザ
 i d ビデオ再生 シ シ 電子 籍
 pple V ir la シ ま i ad s - C カ

i e 音楽 i u es re ウ ド ア pp re 簡 ウ
 ド C/Mac i u es コ テ ツ 共有 バッ アッ
 在 ディア レ ヤ コ ュニケ ショ ツ 売 ず
 著 購入 i d uc () ケ 能
 GB 9,800 円 低価格 シ

アッ 製品 核 i u es 過言 004
 i d & i u es (オ ジャパ) 00 i d & i u es H C (オ
 ジャパ) 売 著 00 i d mi i 購入 晴 アッ ュ ザ

実 00 ODC I G H C (オ ジャパ) 籍 売
 i u es へ dcas ポ 項目 ジョブズ DC 00 キ ノ 話
 i u es dcas i g 対 dcas 当 楽曲 配 使
 可能性 思 著 権 め 音楽サ ショ
 ビュ ニュ ジオ番組 配 使

2. ラ エアチェック - -

デ ッ 二 (. Ma 0 0) U B接 FM/ M
 ジオ アチ ッ 音 ア 存在 曜工 - ジオサ

July 2011

TSNET スクリプト通信 4.1

バ デ ツ サ バ (カ サ バ) 音 ア アッ
ぞ 再生 CGI

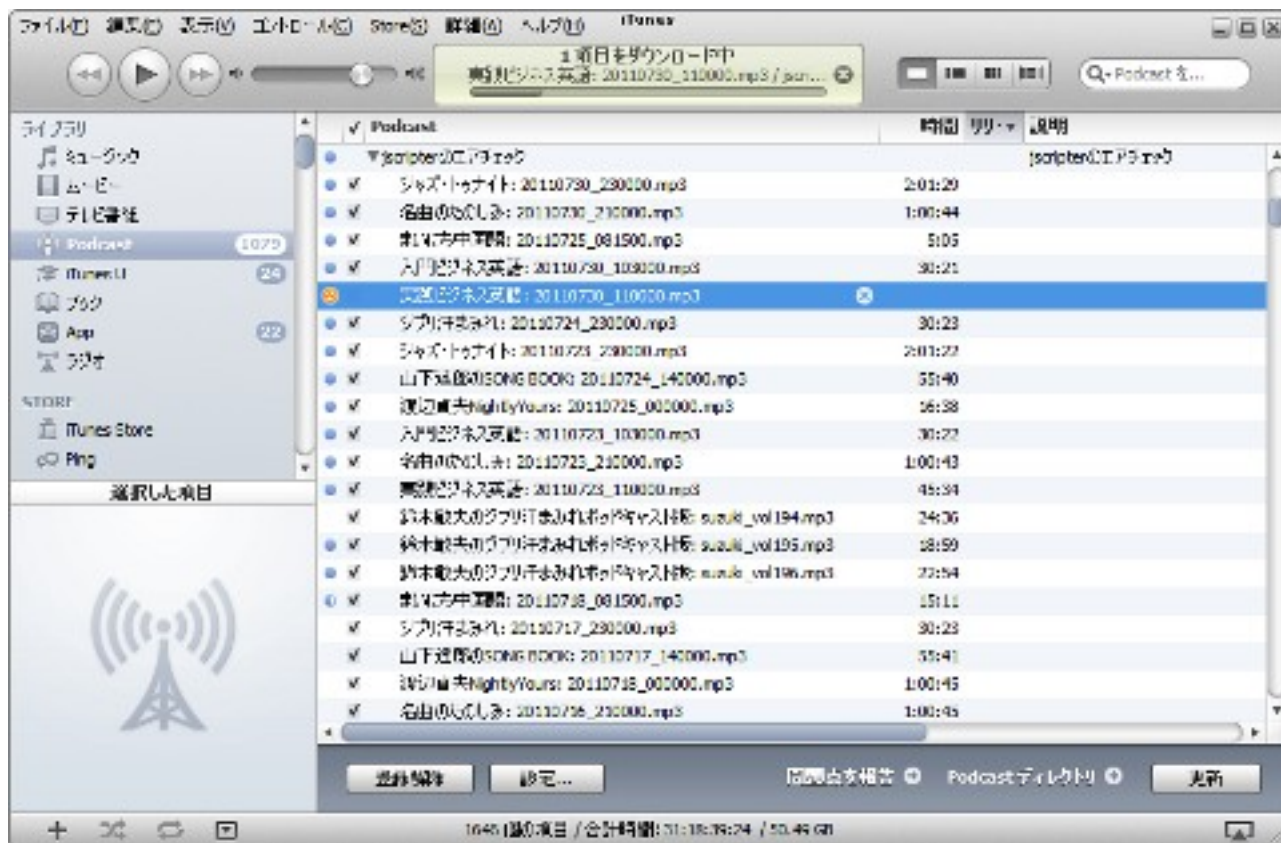
用 i u es radi recm.pl CGI出力 i u es dcas 用 R .0 出力 更
i u es み込ませ み 実験報告 サバ 設定 実行環境
曜工 - ジオサバ 参照

場 | 篤une's通の終

i es

力

項 示 i u e s p d c a s 用 備 R . 0 出力 CGI URL 入力
 0 ボ 押 常 件 ウ ド み込ま 他 入
 手 ボ 押せば ウ ド ウ ド画面 例 掲載



4. radiorecpcast.pl

URL 名 テ 用 短 ろ RSS 音 ア ろ
 iTunes 載 示 曲 長 itunes:duration 載 参
 取り 詳細 コ 照

Podcast 用 RSS 2.0 取り 容易 アッ Podcast
 ペ ジ ち 参 へ 道 URL 紹介
 Podcast 公開 め 詳細 丁寧

アッ - iTunes - Podcast - Podcast
<http://www.apple.com/jp/itunes/podcasts/specs.html>

```
#!/Per15.10/bin/perl.exe
use URI::Escape;
use Encode;
```

```

# ストップ — — の IP アドレス・ポート番 の 定
my server = "xxx.xxx.xxx.xxx:xxxx";# IP adress and port number of your server
# USB ラ の イル イレクトリの 定
my radiorecdir = "E:/RadioREC";# Radio Recording Directory

# イレクトリ名(mp3 の た い イレクトリ要素)の 得
opendir(DIR, radiorecdir);
my @rdirs = grep {!(/[.][.]?¥z|¥.mp3)/i} readdir DIR;
closedir(DIR);

# CGI : RSS 2.0 の の最初の 分
# iTunes 用の RSS タグ 加 定とチャ ル 定 分
# — — にチャ ルの は変 要 あ ( 後、 定)
print <<HEADER;
Content-type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd" version="2.0">

<channel>
  <title>jscripter のエアチェック</title>
  <link>http:// server/cgi-bin/radiorecpodcast.pl</link>
  <description>jscripter のエアチェック</description>
  <language>ja</language>
  <copyright>Copyright(c) 2011 jscripter</copyright>
  <image>
    <url></url>
    <title>jscripter のエアチェック</title>
    <link></link>
  </image>
  <pubDate></pubDate>
  <itunes:subtitle></itunes:subtitle>
  <itunes:author>jscripter</itunes:author>
  <itunes:summary></itunes:summary>
  <itunes:owner>
    <itunes:name>jscripter</itunes:name>
    <itunes:email>jscripter9@gmail.com</itunes:email>
  </itunes:owner>
  <itunes:image href="" />
  <itunes:category text="Music">
    <itunes:category text="Jazz"/>
    <itunes:category text="Classics"/>
    <itunes:category text="Alternatives"/>
  </itunes:category>
  <itunes:category text="Music Media">
    <itunes:category text="FM"/>
    <itunes:category text="AM"/>
    <itunes:category text="Podcast"/>
  </itunes:category>

```

HEADER

```

# USB ラ の イルは 名で 定された イレクトリに され
# イレクトリ名は cp932(SJIS)の文字 ードで み り、utf8にエン ードし、
# さ にURI エス ープ とによって、 イレクトリ URL の スとして
# アク スで ように
# はTSNET スクリプト通信 2.1 (May 2010)、79-80 ー の と
foreach my rdir (@rdirs){
    my encode_rdir = encode("utf8", decode("cp932", rdir));
    my esc_rdir = uri_escape(encode_rdir);
    opendir(DIR, "radiorecdir/rdir");
    my @rmp3s = grep(/¥.mp3/i, readdir(DIR));
    closedir(DIR);
    foreach my rmp3 (@rmp3s){
        my rmp3_url = "http://server/radio/esc_rdir/rmp3";

# CGI : RSS 2.0 の item 要素
# Podcast の番 つ つの 定
        print <<ITEM;
        <item>
        <title> {encode_rdir}: rmp3</title>
        <link> rmp3_url</link>
        <description><![CDATA[]]></description>
        <itunes:author></itunes:author>
        <itunes:subtitle></itunes:subtitle>
        <itunes:summary></itunes:summary>
        <enclosure url=" rmp3_url" length="" type="audio/mpeg" />
        <guid> rmp3_url</guid>
        <pubDate></pubDate>
        <itunes:duration></itunes:duration>
        <itunes:keywords></itunes:keywords>
        </item>

ITEM
    }
}

# CGI : チャ ル要素と RSS 要素の終 タグ
print <<FOOTER;

</channel>
</rss>
FOOTER

```

5. お りに、さ 後の

radiorecpcodast.pl + iTunes + iPod touch

便利

音楽生活

手放せ

July 2011

TSNET スクリプト通信 4.1

Web 情報 radiorecpcast.pl
iTunes デ iTunes RSS 2.0 出力 radiorecm.pl
RSS 生 RSS 2.0 出力 radiorecm.pl
埋め込 デ

音 ア 情報 如 取得 radiorecm.pl
音 ア 対 特定 番組 感想 便利 ろ
音 ア 対 特定 番組 感想 便利 ろ
サ 載 便利 ろ

ちろ 将 セ ティッ ウ ブ化 特定 情報 良
情報 良

(: 2011 7 31)

編集後記

jscrippter

後 荒業 ま 残 安 抱 まま 巻頭言
取 組め
ボ太 全4 . 始ま 4. 無 終
ま 行 継 まず 素直 喜び 7 ボ0
へ 掲載 願 後 在庫(失礼^^;) 肩 凝 ビ
惜 まず 入 残 少 数 絞 ざ 得 出 海鳥 海鳥黙示 サ
ジュア 系 掲載希望 ば 誌面 ゆ 0 小麦粘土 製 ボ太
閉鎖 継 難
取 せ 思 出
0 末 行予定 め 7, 0, , 4 サ 度 数 少
サ ず ば 思 参加 , 8, , 増 サ ぐ ば 思 宣伝 4
サ 維

! 2 11 7 31 \$

スクリプト通信

ISSN 1884-2798 版 :

2011 7 31

4.1.001 版

稿

[TSNETWiki](#) : 「[稿](#)」の ー の と

編集 (稿順)

Y さ saw[a]s7[dot]w [dot]qi [dot] e[dot]jp
機械伯爵 [信]4連fえr冴」:の2ソW]

TSNET スクリプト通信 第4巻第1 (通算第13)
: TSC編集 : 2011 7 31
ISSN: 1884-2798 版 : : 2008 5 7