ゼロからゲームプログラミング2

平山尚

まずは先週もらったものを見た

- ざっとどんなのがあったか紹介

字下げの問題

字下げがいいかげんだと、読みにくくなる。

字下げ

```
while ( x < 10 ){
    while( y < 10 ){
        y = y + 1;
    x = x + 1;
}
}
```

xの加算が内側whileにあることに一瞬気づかない。

字下げ

- 字下げは正確にやった方が後々楽。
- スペース3,4個か、タブキー1回を単位にするのがお勧め。(私はタブ派)

- 字下げが全くない人もいる！
  - 頭の中混乱しないか？

間を埋める方式1

- 古いx,yを取っておいて、その中点に点を打つやり方

<長所>

・簡単。高速。

<短所>

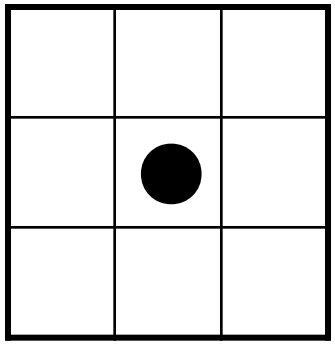
・マウスが速いと切れる。

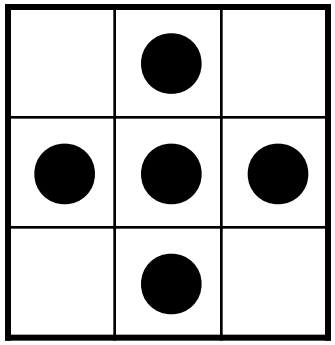
間を埋める

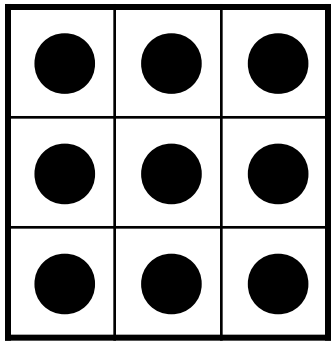
- マウスボタンを離して、また押した時に変なときに点が出るよ？
  - 離した状態でマウスを遠くに動かせばそうなる。
  - 押された瞬間は間に点を描く処理をやめよう。
    - 押された瞬間かどうかを判定する方法は？

消しゴム機能

- 「右クリックだと黒で塗る」というのがあった。
  - でも、消しゴムの太さが小さいと消しにくい
  - 太さをつけるといいかも。
    - その点だけでなく、周りの8点も塗るとか。







太線

- 太い線を引いてるのもあった。
  - でも範囲チェックが不十分！
  - (x+1,y+1)に塗るならそっちも範囲チェック！

色と太さを変える

- 画面外にはみ出したら変わるのもあった。
  - 実験中？
- キーボードで変えるのもあった。
  - 3つのキーで色を混ぜられる。
  - 左が赤で上が青で、両方押すと紫とか。
    - 石津さん

画面クリア機能

- 右クリックで画面が真っ黒になるのもあった。

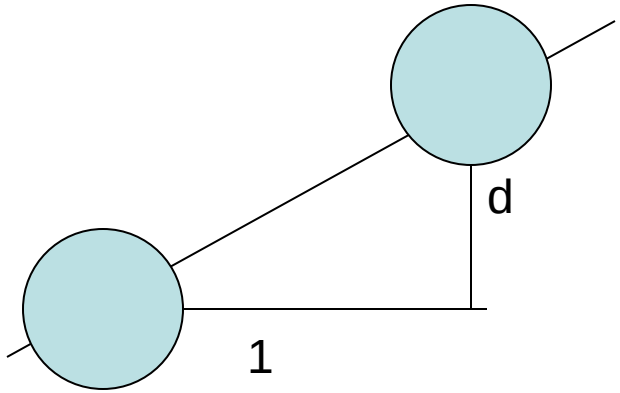
変数名

- i = 10000;と描いてからi[0]とかi[1]とか使ってる
  - 短く書けていい。ただ、プログラムが大きくなったらもうちょっと名前考えような。iの1文字でそれがなんだかわかる人は多くない。

間の埋め方2

- (x0,y0)から(x1,y1)まで線を引く
  - xを+1する度に、傾き分だけ足せばいい、と考えたのがあった。

```
d = ( y1-y0)/(x1-x0) //傾き
while(){
    点打つ(x, y );
    y = y + d;
    x = x + 1;
}
```



間の埋め方2

- もうちょっとだったな。
  - Sunabaは整数しかない。
  - 傾きが0.9とかだったら0になっちゃうよ？
  - 垂直な線だと傾き無限大だよ？
  - 傾き大きいとやっぱり途切れるよ？
    - 尾畑さん

繰り返し多すぎ

```
//白
memory[60000 +memory[50000]+(memory[50001]*200)] = 999999;
memory[60000 +memory[50000]+(memory[50001]*200)+1] = 999999;
memory[60000 +memory[50000]+(memory[50001]*200)+2] = 999999;
memory[60000 +memory[50000]+(1+memory[50001]*200)] = 999999;
memory[60000 +memory[50000]+(2+memory[50001]*200)] = 999999;
memory[60000 +memory[50000]+(1+memory[50001]*200)+1] = 999999;
memory[60000 +memory[50000]+(2+memory[50001]*200)+2] = 999999;
```

変数使おうぜ。

```
x = memory[ 50000 ];
y = memory[ 50001 ];
画面 = 60000;
白 = 999999;
画面[ x + ( y * 200 ) ] = 白;
画面[ x + ( y * 200 ) + 1 ] = 白;
画面[ x + ( y * 200 ) + 2 ] = 白;
画面[ x + ( 1 + ( y* 200 ) )] = 白;
画面[ x + ( 2 + ( y * 200 ) )] = 白;
画面[ x + ( 1 + y * 200 ) + 1 ] = 白;
画面[ x + ( 2 + y * 200 ) + 2 ] = 白;
```

すっきり！こうやって直すとバグが良くわかるよね？

- -たぶん((y + 1) * 200)？括弧の位置。
- -足したところも範囲チェックしないと駄目よ。

関数使おうぜ。

```
点打つ( x, y, color ){
  if ( ( x > -1 ) * ( x < 200 ) * ( y > -1 ) * ( y < 150 ) ){
    memory[ 60000 + x + ( y * 200 ) ] = color;
  }
}
```

この関数作っておけば範囲チェック漏れもないし、括弧の付け間違いもなくなるし、短くもなるぞ。

→コンピュータは繰り返しを楽にやるための機械。

同じことを何回か書いたと思ったら絶対いい手がある。

範囲チェック

- だいたい、というかほぼ全員が範囲チェックが不完全。マウス位置だけでやっちゃ駄目。

パレット。

- 国松さん、おもろい。
  - だが、まだまだ。
  - パレットも塗れちゃうよ？
  - 範囲チェックはかなりできてるが、下が不十分

ライブラリ整備

- ライブラリをえらく整備してる人もいた。
  - 好きな数値を表示できたり、余りを出せたり。
- 実によろしいが、本体もがんばろうな。
  - ライン描画は「傾きが小数」問題ではまってる。
  - 柴原さん

変わった範囲チェック

```
if(a>90000){a=89999;}
if(b>90000){b=89999;}
if(c>90000){c=89999;}
if(d>90000){d=89999;}
```

無理やり番号を範囲内にしてるが、
これ、いつも右下に点が出ちまうぞ？

-あと、4つに分割されてるが、何か意図が？

短く書く工夫

```
if(x==0){if(y<2){col=white;}}//白選択
if(x==1){if(y<2){col=white;}}//白選択
if(x==2){if(y<2){col=red;}}//赤選択
if(x==3){if(y<2){col=red;}}//赤選択
if(x==4){if(y<2){col=green;}}//緑選択
if(x==5){if(y<2){col=green;}}//緑選択
if(x==6){if(y<2){col=blue;}}//青選択
if(x==7){if(y<2){col=blue;}}//青選択
```

Yの条件一緒じゃん。

```
if ( y < 2 ){
   if(x==0){ col=white;}//白選択
   if(x==1){ col=white;}//白選択
   if(x==2){col=red;}//赤選択
   if(x==3){col=red;}//赤選択
   ...
}
```

全部見てみて

- 字下げいいかげんなの多い。
  - 右に行き過ぎて書きにくいような状況になるのは整理できてないからだ。変数の使い方。
- 変数うまく使おう。
- プログラムの文字数が多いほど大抵は駄目。
  - シンプルに書く方法を模索しよう。
  - 長くなると自分でもよくわからなくなってくる。

で、どうやって線引くの？

- 線の引き方。
- まず、線を引く部分を関数に分けるとしよう。

線を引く

- drawLine(x0, x1, y0, y1, color)
- みたいな形になるだろな。
- (x0,y0)から(x1,y1)に線を引く。

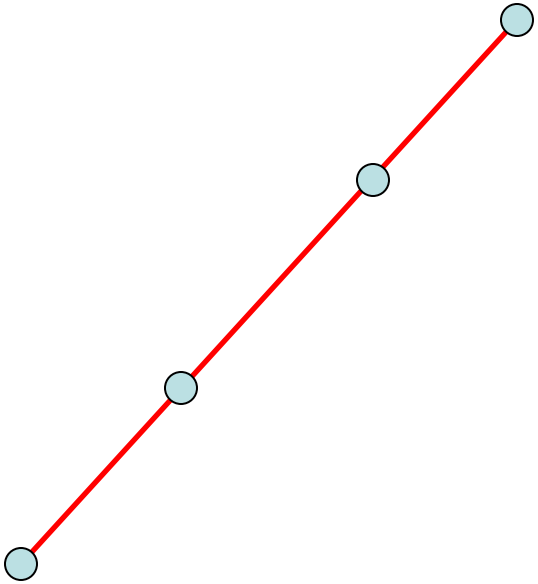
線を引く（１）

- 中点を置く。
  - これは簡単。

点打つ(x0, y0, 色);
点打つ((x0+x1)/2, (y0+y1)/2, 色);
点打つ(x1, y1, 色);

で終わり。ただし、これでは早く動かすと
つながらない。

線を引く(2)



- 点増やせばつながるんじゃね？
  - 3分割してみる。

点打つ(x0, y0, 色);

点打つ((x0+(x1*2))/3, (y0+(y1*2))/3, 色);

点打つ(((x0*2)+x1))/3, ((y0*2)+y1)/3, 色);

点打つ(x1, y1, 色);

  - x0を66%、x1を33%、みたいに混ぜている。
- まだ切れる。じゃあもっと割ればいい。

線を引く(3)

while使うよ。

```
a = 0;
while ( a < 10 ){
  x = ( x0 * ( 10 - a ) ) + ( x1 * a );
  y = ( y0 * ( 10 - a ) ) + ( y1 * a );
 点打つ( x/10, y/10, 色 );
 a = a + 1;
}
```

線を引く(4)

- この方法について整理
- 長所
  - 簡単で短く書ける
  - 計算量一定
- 短所
  - 繰り返しが少ないと切れる。多いと無駄が多い。

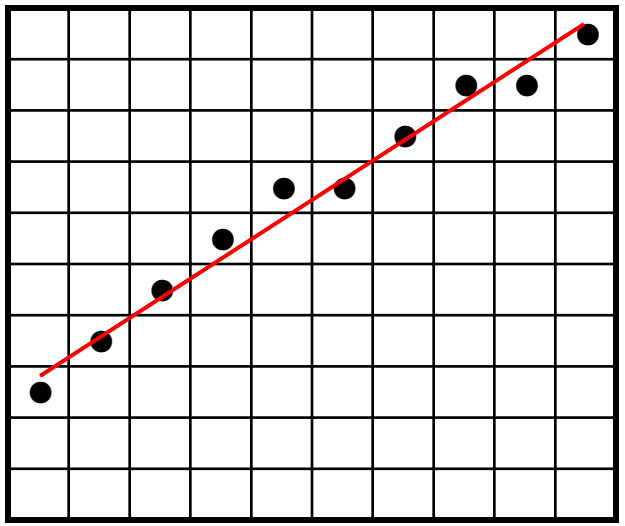
線を引く(5)

- 線の長さに比例した数の点しか打たず、絶対に切れないようにする方法もある。

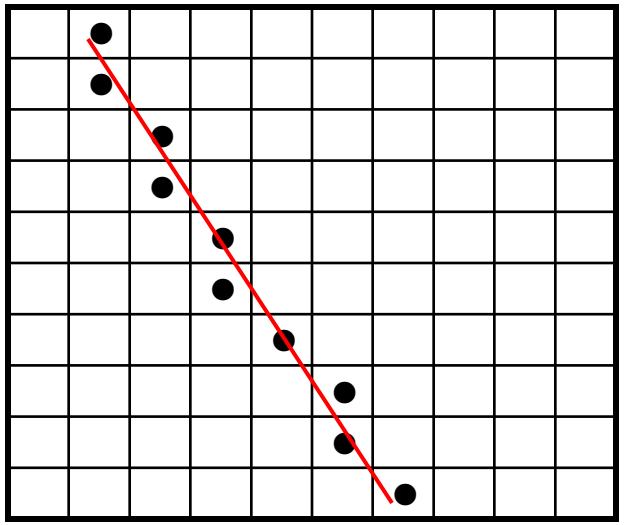
- でも、ややこしいよ？

線を引く(6)

```
abs( x ){  if ( x < 0 ){ x = -x } return x; }
drawLine( x0, y0, x1, y1, color ){
        //x,yの長さを測る
        xl = x1 - x0;                yl = y1 - y0;
        xlAbs = abs( xl );           ylAbs = abs( yl );
        //長い方でループ
        if ( xlAbs > ylAbs ){ //横長
                x = x0;
                dx = 1;
                if ( x0 > x1 ){ dx = -1; }
                while ( x != x1 ){
                        y = y0 + ( ( yl * ( x - x0 ) ) / xl );
                        drawDot( x, y, color );
                        x = x + dx;
                }
        }
        if ( ( xlAbs > ylAbs ) == 0 ){ //縦長
                y = y0;
                dy = 1;
                if ( y0 > y1 ){ dy = -1; }
                while ( y != y1 ){
                        x = x0 + ( ( xl * ( y - y0 ) ) / yl );
                        drawDot( x, y, color );
                        y = y + dy;
                }
        }
}
```



横長(xで反復)



縦長(yで反復)

線を引く(7)

- 結構注意点多い。
  - x0>x1,y0>y1の時の注意。
  - y0 + (x-x0)*(y1-y0)/(x1-x0)での計算
    - 傾き(y1-y0)/(x1-x0)を先に計算しておくと小数を持てないために全然違う結果になるぞ。
    - 割り算は最後な。(4/5=0を忘れずに！)
  - sample/simpleDrawingInterpolation参照のこと。
- こんなに面倒くさいんだし、別に10回とか100回点を打つ奴でいいと思います。

じゃあ、準備運動終わり。

- ゲームを作ろう。

- 次回前半が締め切り。後半は発表その他。
- 今日も提出してもらいますが、動かなくていいです。何をしようとしてるかわかればOK。
- 動いた人には発表してもらおっかな。

ゲームって言われても…

- 自分のレベルに応じて、課題を考えてください。
  - 何か自分で考えたゲームを作れそうなら最高。
  - テトリスいけそうな人はテトリス。
  - テトリスがきついなら3マスに減らしたテトリスとか、2マスに減らしたり、とりあえず背景だけ出したり。
- 一応、テトリスを作るつもりでがんばって。

その他、できそうなもの

- インベーダーは結構楽に作れます。
- オセロとか作ってもいいよ(思考が大変だが)
- 弾幕シューティングもたぶん行ける。
- RPGは難しいだろうなあ。
- スーパーマリオも、面が適当でよければ。

終わり

- バグ報告、質問はhirasho0@gmail.com

- 最新のSunabaPlayerは
- http://www.page.sannet.ne.jp/hirasho/
- .msiはインストーラ。簡単おすすめ。
- .zipはSunabaPlayerのソースコード。上級者向け。