

ゼロからゲームプログラミング2

平山 尚

Sunaba更新

- 新しい実行ファイルはbin/SunabaIDE.exe

先週もらったものを見た

- 変数普及してないなあ。
- 部品に関してはほぼ皆無。
- 同じことを何度も書くことを悪いと思っていない。
- 一人ずつ1時間とか教えられればなあ！
 - 仕方ないので、出来る範囲でやります。

去年よりヤバイ。

- 顕著にヤバイ。
 - 単純に皆さんプログラミング不慣れ？
 - Sunabaの文法がCやJAVAと似ていないことが原因？
- 「覚えたことを使う」は期待してない
 - 「その場でルールを把握して応用する」力を期待して敢えて文法を独自にしたんだが...

今週の課題

- 最低限使えるようになってもらう機能
 - 変数
 - 部品(関数)
 - この二つが使えてないと70点あげない。
 - 何かしら「ゲーム」と言い張れるものを作っていないと80点あげない(途中でもいいし、つまらなくてもいい)

今週の課題

1. 左クリックでお絵かきするソフト
 2. 左クリックを2回すると一回目が左上、二回目が右下になるように長方形を描くソフト。
 3. 四角形は右クリックに割り振って、1,2のソフトを一つにしてほしい。
- ここを最低限としたい。
 - 変数と部品は使うこと。

以下に気をつけて

- 「memory」を極力減らそう。
 - 変数を使え。
- 同じことを何度も繰り返すな。
 - 部品を使え！
 - 範囲チェック付きで、好きなx,yに点を打つ部品
 - 左上と右下をもらって四角形を描く部品
- 似たようなものが何度も出てくるプログラムは高い評価をしない。

60000+(y*100)+xの意味は？

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79

- 1回10x10だと思って考えてみて？
 - (2,5)は52、(8,2)は28だよね...
 - こうする式は(y*10)+xじゃない？

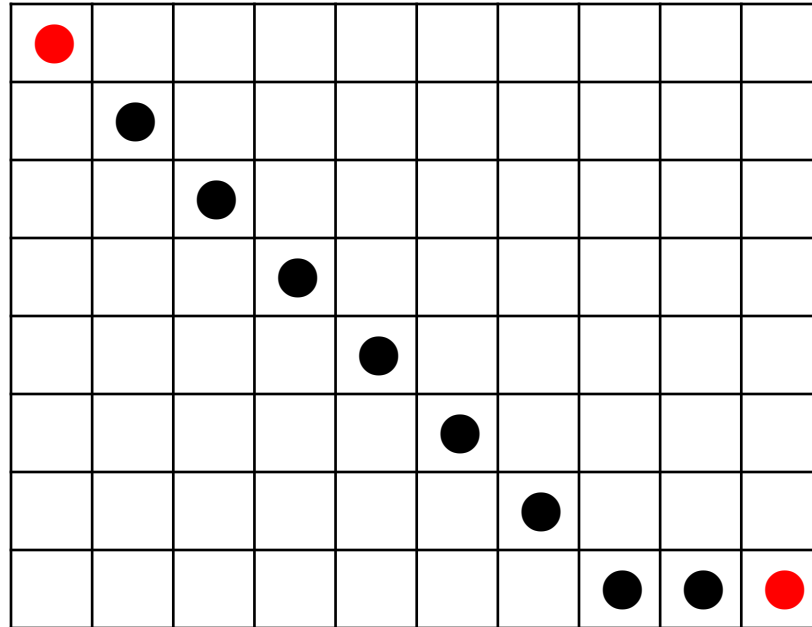
線をつなぐ方法

- 前に打った点との間を埋める。
 - 中点だけでも結構埋まる。
 - 10個くらい埋めれば十分じゃない？
 - どれだけ離れてもつながるようにできればなおよし。

補間

- (x_0, y_0) と (x_1, y_1) の中点は？
 - $((x_0+x_1)/2, (y_0+y_1)/2)$
- 間に2点入れるなら？
 - $((2x_0+x_1)/3, (2y_0+y_1)/3)$
 - $((x_0+2x_1)/3, (y_0+2y_1)/3)$
- これを一般化すると、
 - $(x_0*t+x_1*(1-t), y_0*t+y_1*(1-t))$ をいくつかの t で計算する。

こんな例があった。



- いけるだけ45度で行って、残りは水平か垂直でつなぐ。
 - まあ悪い考えじゃないが、20度とか30度も欲しいよな。

中点の話

- マウスボタンを離して、また押した時に変な所に点が出るよ？
 - 離れ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？

出力値がない部品

- もうちょっと詳しく。

//部品作成部

点を打つ(x, y, 色)

```
memory[60000+(y*100)+x] <- 色
```

//部品使用部

点を打つ(100,50,990000)

出力値がある部品

```
余りを出す( a, b ){  
    return a - ( ( a / b ) * b );  
}
```

余り = 余りを出す(10, 4); //変数「余り」が2になる。

- 「return 値」で計算結果を返す。
 - returnがなければ0。
 - 「余りを出す(10,4)」は変数や値を書くところにはどこでも書ける。

出力値で置き換える

- 余り = 余りを出す(余りを出す(15, 9), 余りを出す(23, 5));
 - 余り = 余りを出す(6, 3);
 - 余り = 0;
-
- 以上のように自動で置き換えられていく。
 - 部品の真髄は、「一旦書いてしまえば中身のことを忘れても使える」ことにある。
 - 余りを出す計算がどんなものかわからなくても、余りを出すことさえ知っていれば使える

高速モード

- 何もしないと点を一個打つ度に1/60秒止まります。
 - 画面に点が表示されるのを待つので。
- これでは困る！遅すぎる！という場合には...

高速モード

```
memory[50015] <- 1 //画面更新を待たなくなるスイッチ  
while 1 //一番外側のwhile  
  ... (いろいろな処理) ...  
  memory[50016] <- 1 //画面更新を待つ
```

- 50016番に1を入れたタイミングでマウスやキーボードを見に行き、50000-50010あたりの値を変えるので、注意。
 - キーを押されても、そこで50016番に1を入れないとそれが見えない。

サンプルその他

- `doc/lecture/2012`以下にこのパワポ
- `doc/lecture/2012/sample`以下にサンプル
 - `tetrisSimple`: 単純なテトリス
 - `tetrisAdvanced`: だいたい作りこんだテトリス
 - `drawingWithInterpolation`: お絵かきのお手本
- `doc/book/book.pdf`: 出版予定の説明本
 - ただし書き始めたばかり。独習用。

で、ゲーム

- 本題はゲームを作ること。

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。

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、ステージが適当でよければ。

終わり

- バグ報告、質問は
 - hirasho0@gmail.com
 - Twitter: hirasho
- 最新のSunabaは以下
 - <http://hirasho.myhome.cx/>
 - 書籍原稿の最新版もここに。