すうけん

数学研究部 平成二十年度活動報告

-日次-

@ 目次 ••• 1 @ RPGゲームの紹介 ••• 2 中等部3年 劉 嘉叡 ••• 3 @ ~ CangeableCitv~ 中等部3年 齊藤 百 @ もじ達の戦い ••• 5 中等部1年 佐藤邑星 村田拓也 長井童介 @ ぱくぱくゲーム ••• 6 中等部1年 新田 泰大 ••• 7 ② キャラクタを動かす方法について 中等部1年 佐藤邑星·村田拓也·長井章介·新田泰大 @「宿題お助け君」の作成 ••• 11 顧問 堀井雅司

@編集後記

··· 21

はじめに

本日は数学研究部の展示においでいただき有り難うございました。昨年度から活動内容の発表の場としてこの小冊子「すうけん」を発行することになりました。数学研究部は現在はプログラミングを中心に活動しています。プログラムといってもほとんどがゲームを作ることになるのですが、ゲーム制作を通して、数学的な考え方やコンピュータサイエンスの入り口を学び、そして何よりも、「ものを作る楽しさ」を感じてくれたらと考えています。

今年はRPG,シューティングゲーム,アクションゲームと,この春からプログラムを 勉強し始めたにしてはいろいろの作品ができました。まだまだ不十分なところもある のですが、自分のアイデアが「モノ」になり、思い通りに動くと嬉しいもので、みんな 楽しんで作っていました。中1の部員たちが、初めてゲームを作るにあたって学習 したことをまとめています。内容をご一読いただければ幸いです。

(顧問 堀井雅司)

~~questions~~

11月3日(月) 劉 嘉叡

本作はRPG2003を使用して作られたものです。

く内容>

法王として権力を握ったYと名乗る者のやり方に疑問を持った、 主人公のザックが仲間と共にYの試練(クイズ)を受けるゲームです。 しかし、最後はやはり力のあるものが勝つのです。 Yは強めに設定されていますが、クイズに答えるとレベルがあがったり するのでがんばったらクリアできるかも? 隠しキャラクターがあります。話かけるのはセーブしてからをお勧めしま す。

<操作方法>

移動 : カーソルキー 決定(話しかける) : Zキー キャンセル : Xキー ステータス : Xキー



<感想>

RPGツクール2003で作りました。RPGツクールを使うのは初めてだったので、スイッチとかイベントなどの設定がうまくいかず結構苦労しました

~ChangeableCitv~

2008/11/3

1. ゲーム制作趣旨

一人の王子が、自分の正義の心にしたがって悪を倒します。 正義の心は時に命取りになりうるということを、 このゲームを通してわかってもらいたいと思って作り ました。



別に正義の心を持つな、と言っているのではないのです。 人生では時に適応力や妥協する力が必要となります。 このゲームを通して適応力、妥協する力を 鍛えてもらいたいと思います。

2. 操作方法

キーボード 方向キー:移動 Zキー :決定、話しかけ Xキー :キャンセル ステータス :フルスクリーン F4 モードとウィンドウ モードとを変更 F12 :リセット



3. ゲームの制作と感想

RPGツクール2003というソフトを使ってこのゲームを作りました。 RPGツクールというのは、誰でもそれなりに普通のゲームが 作れるゲーム制作ソフトです。

しかし、初めての人では非常に単純なゲームしか作れません。 自分はRPGツクールの良さを最大限に引き出すことで なかなか納得できる出来になりました。 無料で配布というのは極めて遺憾であります。



大阪星光学院中学校三年A組 齊藤 亘



①キャラクターの説明 これは主人公です。走ったり 跳んだりすると文字が変わり ます。やってみてください。



これは敵です。当たり判定(弾があたるところ)は文 字のところにあります。レベルが上がると、敵の表示 が変わります。がんばって隠しステージを見つけてく

ださい。

②遊び方

主人公が敵を全部倒したらゲームクリア、敵にやられるとゲームオーバーです。

負ける時

- [1] 敵にぶつかった時
- 〔2〕落とし穴に落ちた時
- 〔3〕敵の弾にライフ分当たった時

③キーボードの説明

Zキー : ジャンプ	Xキー:弾の発射
→キー : 右に移動	←キー:左に移動

④その他説明

ゲームクリアをしたら、その時に「文字」が表示されます。そ れをタイトル画面で押してください。難しくなります。

ゲーム本体製作	佐藤邑星	村田拓也	長井章介
説明書製作	佐藤邑星	村田拓也	
責任者	佐藤邑星	村田拓也	長井章介
キャラクター製作	佐藤邑星	村田拓也	
発行所	大阪星光学院	数学研究部	

※これは、大槻有一郎さんの「14歳からはじめるC言語わくわくゲームプ ログラミング教室 Visual Studio 2008 編」を参考に製作しました。

[©] パクパクゲーム

2008.11.3 新田 泰大

1. ゲーム内容

このゲームはパッ●マン(ナ●コ)のパクリゲームです。ここから、 ルールと詳しい説明をしていきます。

★キャラ等の紹介

・・・自キャラです。カーソルキーで操作します。

 ・・・・・ボールです。ゲームクリア・ゲームオーバーには無関係で 点数が高くなります。

● ・・・・鍵です。とれば鍵で開く扉 が開いて通れるようにな ります。

_____・・・・・レンガです。ふつうの壁です。

・・・・・敵です。ぶつかるとゲームオーバーです。鍵をとると追ってくるスピードがはやくなります。

← ・・・・これを取るとゲームクリアとなります。

★あそびかた

まず、メニュー画面でレベルを設定して下さい。レベル1ならZ、レ ベル2ならX、レベル3ならC、をおして下さい。すると「ボールの数設 定」と表示されます。上下キーで選択、スペースキーで決定です。ゲー ムが開始されます。ゲームオーバー・ゲームクリアの画面でスペース キーを押すとタイトルに戻ります。

キャラクタを動かす方法について

「もじ達の戦い」と「パクパクゲーム」を作る過程で学習したことから、キャラクタの移動についてをまとめました。これは、大槻有一郎さんの「14歳からはじめるC言語わくわくゲームプログラミング教室 Visual Studio 2008 編」を参考に製作しました。

中1 佐藤邑星·村田拓也·長井章介·新田泰大

1. 自キャラの表示

プログラム内にある、「LoadGraph」という関数で画像をメモ リに読み込み、「DrawGraph」という関数で画像を表示します。 読み込んだ時に、 ghandle のような整数型の変数に画像の番号が入る ので、描画する時はその番号を使います。(下のプログラム参照)

2. 画像を動かす

画像を動かすというのは、もとの画像の消し、次の画像を描く ことを繰り返すだけですが、そのままでは、画像がちらつきます。 そこで、オモテ画面とウラ画面の2つの画面を用意して、オモテ 画面を表示している間に、ウラ画面を消して画像をすべて書き込 み終わってから、オモテ画面と入れ替えるようにします。そのス ピードがとても速い(1秒間に約60~100回の書きかえをす る)ので、瞬時に画像が変化し、画像が動いているように見える のです。具体的には、

```
#include "DxLib.h"
int WINAPI WinMain (HINSTANCE hI, HINSTANCE hP, LPSTR lpC, int nC) {
 ChangeWindowMode (TRUE):
 if(DxLib Init() == -1) return -1; //DxLib 使用開始
 int ghandle=LoadGraph( "C: Y YChara.bmp") ;
 int x=0:
 SetDrawScreen( DX SCREEN BACK );
                                // ウラ画面に描画する
 while (ProcessMessage () = = 0 && CheckHitKey (KEY INPUT ESCAPE) = = 0 ) {
                               // 何もない時はループし続ける
                               // ウラ画面の消夫
     ClsDrawScreen():
     if (x \le 320) = x + 2;
                              // キャラクタの位置を2だけ右にずらす
     DrawGraph(x, 224, ghandle, TRUE); // キャラクタ画像の表示
     ScreenFlip():
                               // オモテ画面とウラ画面の入れ替え
 }
```

//DxLib 使用終了

DxLib_End(); return 0;

}

という風にします。これで、キャラクターは2ドットずつ右に動 き続けるのです。

WINAPI WinMain(HINSTANCE hI, ・・・は決まり文句と思ってく ださい。

ProcessMessage() = = 0 & CheckHitKey(KEY_INPUT_ESCAPE) = 0 というのは、「Windows からのメッセージも無く、ESC キーも押 されていない」という意味です。

一度、int ghandle=LoadGraph("C: Y Chara.bmp");で読み込 まれた画像は、 ghandle という変数に入れられた番号で呼び出す 事になります。 (ハンドルといいます)

実際には、いろいろと処理があるので、 ProcessMessage の while 文の中はもっと複雑になっています。次のページの図にある ように、タイトル表示やゲームオーバー表示もこの while 文の中に入っ ていて、ESC キーを押しても、×をクリックしても終了できるよう になっています。しかし、どの処理でも基本は、

- 1. ウラ画面を消去
- 2. 表示したいことをウラ画面に描く
- 3. ウラ画面とオモテ画面を切り替える

としています。アニメーションをしたいときは、表示する画像を 順に替えていくだけで、走っているかのように見えます。

止 打 走 走 走 走 走 走 走 武 恐 死



3. キャラクターのジャンプ

ジャンプは重力の影響をプログラムしています。地面を蹴った ときの最初の速度 JUMPMOVEC を単位時間ごとに重力分 GRAVITY ずつ減らしていくと、重力に引かれ、徐々に落下する ように見えるのです。

具体的には、

```
① const Vector RIGHTMOVEC = {120, 0};
const Vector LEFTMOVEC = {-120, 0};
const Vector JUMPMOVEC = {0, -180};
const Vector GRAVITY = { 0, 300}; // 重力加速度
```

②//ジャンプ処理 if (key & PAD_INPUT_A) { // Zキーを押している間、 curvec1 = Addvector(curvec1, JUMPMOVEC); } // 上に移動する (中略) //ジャンプ中 g_jikivector = AddVectorInFrameTime(g_jikivector, GRAVITY); //GRAVITYの大きさをFrameTimeで調整して加える if(g_jikivector.y > 400) g_jikivector.y = 400; (後略)

これらの処理は1回の書き換えごとに実行されるので、単位時 間ごとにg_jikivector(キャラクタの上への速度)が少しずつ減って いくことになります。

4. 矢印キーでキャラクタを動かす

キャラクタの座標が pack.x、pack.y に入ってい	いるとき、
プログラム	解說
<pre>int key = GetJoypadInputState(DX_INPUT_KEY_PAD1);</pre>	//キーを読み取れ。
if(key & PAD_INPUT_UP) {	//もし↑をおしていたら{
pack.y -= 4;	// y座標4減らせ。
}	//}
if(key & PAD_INPUT_DOWN) {	// もし↓をおしていたら{
pack.y += 4;	// y座標4増やせ。
}	// }
if(key & PAD_INPUT_LEFT) {	// もし←をおしていたら{
pack.x $-= 4;$	// x座標4減らせ。
}	// }
if(key & PAD_INPUT_RIGHT) {	// もし→をおしていたら{
pack.y $+=4;$	// x座標4増やせ。
}	// }
DrawGraph(pack.x, pack.y, ghandle, TRUE);	//自キャラの画像を(x,y)座標 //に表示

「宿題お助け君」の作成

数学科 堀井雅司

Microsoft からフリーの開発環境として Visual C++ 2008 Express Edition が 提供されていて、インターネットから自由にダウンロードして使うことができます。 しかし、C++は敷居が高く、ちょうど良い教材が少ないように思います。以下は、 クラブの中学1年生を対象に試作した C++プログラム講座のメモです。

1. 目標=連立方程式を解くプログラムを作る

なんと言っても、数学研究部なのだから、数学に関した題材を使いましょう。 仕様:2元1次方程式を解く。ただし、係数はすべて整数とす る。右のような連立方程式を解きたい。答えは必要なら分 数で表示する。

小数で表示しても宿題には使えませんよね:-)。

2. VC の起動

スタートメニューから, Visual C++ 2008 Express Edition を起動します。



	(17)後 へんから 「すうまからな」 Cell 「すうまからな」 Cell アイルロー 既得のロードからプロジェクトを行 レisual C+++ Express Edition	анна окра 2008	
Victary Correct State Corr REERAZY(AD) Corr N1 2001 € SCART© The	2019年7日 - 2019年7日 - 2019年7日 - 2019年7日 - 1日日日 - 1日日日日日日 - 1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日	Service of the servic	and site of the second se
	2++ 起動	画面	

「新しいプロジェクト」のダイアログが開くので、

[プロジェクトの種類(P)]を

「CLR」

[テンプレート]を

[Windows フォームアフ^{*}リケーション] として,下の[プロジェクト名]にこ れから作るプログラムの名前を 入れます。ここでは、

shukudai

としておきます。(図2)

このとき,自分の「マイドキュメント」の中に,「Visual Studio 2008」というフォルダが

9 編集(2) 表示(2) 9-80 90円(2) - 29 編集(2) 大田(9-10-2) - 29 編集(2) 大田(9-10-2)

Visual C++ 200

1052 54750 1050 CLR 705 マイ アンプレート

図2:新しいソリューションをつくるウィザード

作られ、その中の「Projects」の中 の「shukudai」フォルダにプログラ ムが作られます。あとで「どこにあ るのだ?」と探さなくてもよいよう に、覚えておきましょう。OKボタ ンをクリックすると、フォームが1 つ出来上がります。最初の「ソ リューション」の誕生です。(図3)



3. コントロールを配置する

フォームの上にボタンなどのコントロールを配置して、入出力表示画面を作ります。 右端に、縦にある「ツールボックス」というタブにマウスカーソルをあわせるとメ ニューが出てきます。

「コモンコントロール」の中の「テキストボックス」をクリックし、「Form1」の上で、マウスの左ボタンを押しながら適当な大きさの四角形を作ります。これを繰り返し、図4のように、8個のテキストボックスを配置します。1つのテキストボックスを選択し、

コピー&ペーストで複製することもできます。これは、連立方程式の係数を入力す る場所になります。

配置が終われば、各テキストボックスに名前を付けます。上の6つは連立方程式の係数を入れるところなので、

「RHTB1(連立方程式 TextBox 1)」 下の2つは答えを入れるところなので, 「RHKTB1(RH 答 TextBox1)」のように 決めます。名前の入力はテキストボッ クスを選択し,右クリックー「プロパ ティ」で図5のようなプロパティーウィン ドウが開くので,その中の「デザイン」



にある「(Name)」の右に新しい名前を入れます。プロパ ティーを abc 順に並べているときは、一番上にくるので注 意しましょう。(名前は半角文字で入力し、大文字小文字 を区別します)(図4)

答えが分数になることも考えて、さらに TextBox を追加し ます。分数の線はラベル(Label)コントロールを使います。 また、答えのところには、「x=」のようなラベルをおきましょ う。方程式の形も、「□x+ □y= □」のような形にしましょう。 さらに、連立方程式は「{」でくくりましょう。

ボタン(Button)コントロールをひとつおき、プロパティー Windowで「Text」プロパティーを「解く!」とし、最後に、 フォームのバー(青いところ)をクリックして、フォームを選 択し、フォーム(Form1)の Text プロパティーを「宿題お助 け君」とします。(図6)これでフォームは完成。後はプログ ラムを組むことになります。



bun1,bun2	(Label コントロール)
-----------	----------------

プロパティ	値
(Name)	bun1
AutoSize	False
Size Width	60
Size Height	3
BackColor	Black

(Name) 以外は bun2 も同じです。



4. プログラミング

ボタンをクリックすると答えが出るようにしましょう。プログラムを入力する Window は, button1(「解く!」のボタン)をダブルクリックすると現れます。

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {

となっているところにカーソルが来ているはずです。

private: System::Void button1_Click(..... の行の右端で ENTER キーを入力し, 次のページのプログラムを入力します。 プログラム中,「hyoujix()」「hyoujiy()」 「GCD()」「invisible()」という関数が出 てきますが,後で定義します。何をす る関数か,何となく名前から判ってもら えるのではないでしょうか。



```
private: System::Void button1 Click(System::Object^.....
      // 変数の定義
      long aa. bb. cc. dd. ee. ff. adbc. edbf. afce; // ax+by=e
                                          // cx+dy=f
      long x1, x2, y1, y2, gcd; //x=x1/x2 y=y1/y2, 最大公約数
      int sx=1:
                            // x.v の符号
      int sv=1:
      // テキストボックスから数値を読み込む
      aa = Convert::ToInt32(RHTB1->Text);
      bb = Convert::ToInt32(RHTB2->Text);
      cc = Convert: ToInt32(RHTB4->Text):
      dd = Convert::ToInt32(RHTB5->Text):
      ee = Convert::ToInt32(RHTB3->Text):
      ff = Convert::ToInt32(RHTB6->Text);
      //整数以外ならエラーになる
      adbc = aa * dd - bb * cc; // edbf
                                          afce
      edbf = ee * dd - bb * ff; // x=---- y=--
      afce = aa * ff - cc * ee; // adbc adbc である
      invisible(): // 答えのTextBoxをすべて消す
      if(adbc!=0) { // ad-bc ≠0 のとき
          x1=edbf; x2=adbc; // 分子と分母をセット
          v1=afce; v2=adbc;
          if(x1*x2<0) { sx=-1; }; // xの符号をとる
          x1=Math::Abs(x1); x2=Math::Abs(x2);// 絶対値
          if(y1*y2<0) { sy=-1; }; // yの符号をとる
          y1=Math::Abs(y1); y2=Math::Abs(y2);
                                  // GCDを求める
          gcd = GCD(x1, x2);
          if (gcd!=0) {x1=x1/gcd; x2=x2/gcd; };// 約分する
          gcd = GCD(v1, v2);
                                // GCDを求める
          if (gcd!=0) {y1=y1/gcd; y2=y2/gcd; };// 約分する
                                 // xを表示する
          hvouiix(x1*sx.x2);
          hyoujiy (y1*sy, y2); // yを表示する
      }
プログラム 1-1:
```

```
else if((edbf!=0)) (afce!=0)) ( // adbc==0のとき
          RHKTB1->Visible=1;
          RHKTB1->Text="解なし";
                                  // x が解なしなら
          RHKTB2->Visible=1:
                                  // v も解なし
          RHKTB3->Text="解なし";
                    // adbc==0 edbf==0 afce==0のとき
      }else{
          if((aa==0)&&(cc==0)){ // 普通なら「不定」
             if((bb==0)&&(dd==0)) { // だが、特殊な係数
                 RHKTB1->Visible=1: // の時は解があった
                 RHKTB2->Visible=1; // り. 解なしになる
                 if((ee==0) &&(ff==0)) {
                    RHKTB1->Text="不定";//係数がすべて0
                    RHKTB2->Text="不定";//のとき
                 }else{
                    RHKTB1->Text="解なし";//a,b,c,dが0
                    RHKTB2->Text="macle";//だがe.f≠0
                                        //のとき
             }else{
                 if(bb!=0) { // x の係数が0でyの
                    y2=bb;y1=ee; // 係数が≠0なら,
                 }else if(dd!=0) { // xは不定でyは
                    v2=dd;v1=ff; // 定まる
                 }
                 RHKTB1->Visible =1:
                 RHKTB1->Text="不定";
                 if(y1*y2<0) sy=-1;
                 v1=Math::Abs(v1);
                 y2=Math::Abs(y2);
                 gcd=GCD(y1, y2);
                 hyoujiy(y1/gcd*sy, y2/gcd);
             ł
          }else{ // a ≠0 またはc ≠0
             if((bb==0)&&(dd==0)) { // v の係数が0で
                 if(aa!=0){ // x の係数が≠0
                    x2=aa;x1=ee; // のとき
                 }else if(cc!=0) {
                    x^2 = cc: x^1 = ff:
プログラム 1ー2:
```



「hyoujix()」「hyoujiy()」「GCD()」「invisible()」の定義は,

#pragma endregion と

private: System::Void button1_Click(System::Object^.....

との間に入れます。(下図)

```
#pragma endregion
( ここに hyoujix( ) や hyoujiy( ) を入れる )
private: System::Void button1_Click(System::Object<sup>^</sup>.....
( プログラム1ー1~3のコードが書かれている )
```

次のページにソースコードを載せておきます。

```
#pragma endregion
// 宿題お助け君の関数の定義開始
   long GCD(long b, long r) { // b>0, r≧0 の整数のGCDを求める
                         // ユークリッドの互除法
       long a;
       while(r!=0) {
          a=b: b=r:
          r=a % b: // r はa をb で割った余り
                   // r=0 となったらそのときの b がGCDだ
       }
       return b:
   }
   void hyoujix(int x1, int x2) { // x=x1/x2 を分数または
       if (x2==1) {
                              // 整数で表示する
          RHKTB1->Visible=1:
          RHKTB1->Text=Convert::ToString(x1);
       }else{
                              11
                            // 分数のTextBoxを表示
          RHKTB3->Visible=1;
          RHKTB4->Visible=1;
          bun1->Visible=1;
          RHKTB3->Text=Convert::ToString(x1);
          RHKTB4->Text=Convert::ToString(x2);
      }
   }
   void hvouiiv(int v1. int v2) / // v=v1/v2 を分数または
                             // 整数で表示する
       if (y2==1) {
          RHKTB2->Visible=1:
          RHKTB2->Text=Convert::ToString(v1);
       }else{
          RHKTB5->Visible=1; // 分数のTextBoxを表示
          RHKTB6->Visible=1:
          bun2->Visible=1;
          RHKTB5->Text=Convert::ToString(v1);
          RHKTB6->Text=Convert::ToString(y2);
       }
   }
プログラム2ー1:
```



5. 連立方程式を解く

 $\begin{cases} ax+by=e\cdots \\ cx+dy=f\cdots \\ cx+dy=f\cdots \\ cx+dy=f\cdots \\ cx+dy=f\cdots \\ cx+dy=f\cdots \\ cx+dy=fc \\ cx+dy$

よって、

 $ad-bc \neq 0$ obtain, $x = \frac{ed-fc}{ad-bc}$, $y = \frac{af-ec}{ad-bc}$

ad-*bc*=0 のときは

(ア) ed - fc = 0 かつ af - ec = 0 のときは「不定」

(イ) $ed - fc \neq 0$ または $af - ec \neq 0$ のときは「解なし」

となるのですが、(イ)の場合は $a \neq 0$, b = d = 0 のような特殊な場合があり、 場合わけが必要になります。($0 \times d - 2 \times b$ が無意味なものになるからです) プログラム1-2 で場合わけしています。 6. 終わりに

最後までお読みいただき有り難うございました。for 文こそ出てきませんでしたが, ユークリッドの互除法があり,連立方程式の解法があり,数学的な分析の入った題 材として使えるのではないかと考えています。CDにソースコードとソリューションを 入れておきますので,いろいろ試してみて下さい。(宿題お助け君2も入れておきま した)

見栄えを良くするためのアイデアをいくつか挙げておきます。

①最初, すべての TextBox の Visible プロパティを False にしておく。

- ②分数になっている TextBox を左に移動させ, RHKTB1, RHKTB2 に重ねておく と、答えが分数の時にも自然な表示になる。
- ③テキストボックスの TextAlign プロパティを Center にする。数字がセンタリング されて見やすい。

今回は割愛しましたが、C++では「有理数型」を作ることができます。

x,y,zを有理数型として,x.bunshi=1; x.bunbo=6; y.bunshi=1; y.bunbo=3; (つまり, x=1/6, y=1/3) として, z=x+y; とすると, z=1/2 となる便利な機能です。こ れを利用してプログラムを書き直すことも楽しいのではないかと考えています。

最後までご覧いただきありがとうございました

編集後記

来年は再びC言語に戻ろうかと思います。できればアクションゲームを作り たいと思います。 劉 嘉叡

スクールフェアが終わったら、本格的にC言語を勉強します。 将来のためにもなるかもしれないので頑張ります。 齊藤 亘

中一になって、初めてゲームを作りました。まだ簡単なゲームしか 作ることが出来ませんが、よろしくお願いします。 佐藤 邑星

C言語はとても難しかったですが、なんとかゲームを作ることが出来てよかったです。 村田 拓也

あまり手伝えることはなかったですが、C 言語を少しでも学ぶことが出来 たので良かったです。 長井 章介

はじめは、敵が規則正しく動いていて、壁をすりぬけてしまい、簡単すぎま した。そこを試行錯誤して敵が追ってくるようにしました。これが一 番難しかった。 新田 泰大

新人6人でしたが、多くの方のお世話になりなんとかSFを迎える ことができました。勉強もせずにプログラムに熱中する部員もいた ことと思いますが、我慢し暖かく見守っていただいた保護者の方々、 先生方に感謝いたします。また、いつも検閲でご迷惑をおかけして いる梅津先生、有難うございました。 堀井 雅司