

1

RPG ツールの逆襲

まさかの RPG ツール 2000

49 (しく)

1 RPG ツールの限界に挑みたくなった

プログラミングコンテストという大会がある

我が東大マイコンクラブにはこの大会にでて優秀な成績を残す人間が幾人もいる。私 49 もいつの日か痛いチーム名*1で出場してその名を日本、いや世界に轟かせてやろうと気が向いたときに大会の過去問を眺めていた。プログラミングコンテストとは簡単に言えば『二点の座標を与えた時にその距離を計算するアルゴリズムを書け』といった感じの問題をひたすらに解く大会である。

しかし、ある日。私はふと、思ってしまった。

今の私を見たら昔の私はどう思うだろう？

プログラミングを習い始めた中学生の時のことを振り返る。当時は淡い夢でしかなかった東大合格を果たし、大好きなパソコンと物理と二次元（座標系的な意味で）に存分に勤しむ姿はそれなりに昔の私の望みをかなえてくれるかもしれない・・・

「いや、そうではない」誕生日のプレゼントをもらった時の事を思い出したとき、私は目が覚めた。

あの時貰った PRG ツールにどれだけ感動したことが

それに引き換え、今の私は真っ黒なコンソール相手に何をしている！

ゲームの素晴らしいところは見て楽しめることだ。音楽があり、画像があり、アニメがあり、ドラマがある。その楽しみに憧れてプログラミングを始めたのではないか。そう思ったとき、私はおもむろに 8 年前の誕生日に貰った RPG ツール 2000 を起動した。

今こそ、RPG ツールの力を見せ付けるときだ！！

2 今回の戦いの概要

私の閃きが如何に素晴らしいものかを理解してもらおうと、私は部会で心の中を先輩にぶつけてみた。すると

「えー、マジ RPG ツール 2000 ！？ キモ～イ」

「RPG ツール 2000 が許されるのは小学生までだよね！」

*1 カードキャプター綾崎 (AYASAKI The CardCaptor) を予定。

みたいな内容が帰ってきた。これに酷く傷ついた私は

「ならば、三角形の面積を求めてみせましょう。しかも、座標系で与えられているような、ゆとり version*²じゃない奴を。三辺の長さから面積を求めてみせましょう！」とコナミコマンドを 100 連発しながらピアノニッシモで叫んだ。

こうして、敵が決まったのでプロジェクト名「UTMC」、ゲーム名「三角形の面積を求める」を作成。ただ作るだけでは無芸の極みなのでキャラクターを置いてみた。

3 戦略の解説

実は三角形の三辺の長さ (a,b,c) を与えられた時、その面積を求める公式はヘロンの公式として存在する。

その式は

$$S = \sqrt{s(s-a)(s-b)(s-c)} \quad s = \frac{1}{2}(a+b+c) \text{ である。}$$

式があるなら簡単じゃないかと思った君は甘い。RPG ツクールにはこれらの式の使用を阻むような恐るべき敵が潜んでいるのである。

3.1 桁落ち

RPG ツクールの変数は 6 桁である。そして、三辺の長さを代入するために使う『数値入力イベント』は 6 桁の数字まで対応している。つまり、単純な足し算でさえ、入れられる二つの変数を 999999 などと入れられることを考えると変数の分割が必要になってくるのである。具体的には

「変数 A に値 444999 が代入される」 「変数 A を 444 と 999 に分割する」

「変数 B に値 999444 が代入される」 「変数 B を 999 と 444 に分割する」

「999+444 を計算する」 「1000 で除算して繰り上がりを作成して、剰余を求める」 「繰り上がった分を考慮に入れて 1+444+999 を計算する」

といった手順を踏まなければならないのである。今回は掛け算を用いるので変数は基本的に 3 桁で区切る羽目になった。

3.2 ルートの計算

ルート計算という機能は RPG ツクールには存在しない。仕方がないので自力で作る必要がある。今回は初心者でも割合簡単に扱える二分法を使うことにした。例えば $\sqrt{2}$ を求める時には

$\sqrt{2}$ は 1 以上 2 以下である

$(1.5)^2$ は 2 より大きい $\sqrt{2}$ は 1 以上 1.5 以下

$(1.25)^2$ は 2 より小さい $\sqrt{2}$ は 1.25 以上 1.5 以下

といった感じの手順をとる。これを繰り返すことで、平方根に近い値を素早く見つけること*³が可能になる。しかし、先ほど述べたとおり掛け算を扱う必要があるため簡単に計算できるわけがない。

*² 一つの点を原点にスライドすれば $\frac{1}{2}|x_1y_2 - x_2y_1|$ で求められるので超ゆとり。

*³ 風漬して探すと死ぬほど時間が掛かる。こういったアルゴリズムの考え方は一学期必修の情報などで習う。

4 作戦 1 : 変数の分割

まずは、三辺の長さを計算可能にするために分割する。ソースコードは以下のとおりである。

```

◆文章：まこと：
：：：三角形の辺の長さを代入してね♪
◆数値入力処理：6桁, [0001:辺A]
◆文章：まこと：
：：：もう一つの辺の長さを代入してね♪
◆数値入力処理：6桁, [0002:辺B]
◆文章：まこと：
：：：最後の辺の長さを代入してね♪
◆数値入力処理：6桁, [0003:辺C]
◆文章：【確認】
：：：%V[1],%V[2],%V[3]
◆変数の操作：[0004:辺A@1000の位以降]代入, 変数[0001]の値
◆変数の操作：[0004:辺A@1000の位以降]除算, 1000
◆変数の操作：[0005:辺B@1000の位以降]代入, 変数[0002]の値
◆変数の操作：[0005:辺B@1000の位以降]除算, 1000
◆変数の操作：[0006:辺C@1000の位以降]代入, 変数[0003]の値
◆変数の操作：[0006:辺C@1000の位以降]除算, 1000
◆変数の操作：[0001:辺A]剰余, 1000
◆変数の操作：[0002:辺B]剰余, 1000
◆変数の操作：[0003:辺C]剰余, 1000
◆文章：【確認】
：：：%V[1],%V[2],%V[3]
◆スイッチの操作：[0001:まこと完了]をONにする
◆スイッチの操作：[0002:せかい完了]をOFFにする
◆

```

まこと せかい ことのは のルートで面積が求められるように設定

5 作戦 2 : 平方根の作成

$S = \sqrt{s(s-a)(s-b)(s-c)}$ $s = \frac{1}{2}(a+b+c)$ という計算を行うに当たり、 $s(s-a)(s-b)(s-c)$ を求めてから平方根を求めるという方法を用いると極めて面倒である。概算すればわかるが、 $s(s-a)(s-b)(s-c)$ は三辺の長さを 99999 などになれば 25 桁になってくれるので、単純に考えれば 9 個の変数が必要になってくるのである。後に示す二分法による近似を見れば更に納得がいくだろうが、この計算のしんどさは並大抵なものではない。

というわけで、正確さは落ちるが、 $s, s-a, s-b, s-c$ をの平方根をそれぞれ求めてからその積を求めることにする。

s と \sqrt{s} を求めるためのせかいのソースコードを作成する

```

◆文章：せかい：
：：今回の注文は？
◆選択肢の表示：へロンの公式準備/ばふばふして欲しい/中z(ry)
：：[へロンの公式準備]の場合
◆条件分岐：スイッチ[0001:まこと完了]がON
◆変数の操作：[0007:へロンの数]代入，変数[0001]の値
◆変数の操作：[0007:へロンの数]加算，変数[0002]の値
◆変数の操作：[0007:へロンの数]加算，変数[0003]の値
◆変数の操作：[0008:へロンの数@1000以降]代入，変数[0004]の値
◆変数の操作：[0008:へロンの数@1000以降]加算，変数[0005]の値
◆変数の操作：[0008:へロンの数@1000以降]加算，変数[0006]の値
◆変数の操作：[0009:条件分岐用]代入，変数[0008]の値
◆変数の操作：[0009:条件分岐用]剰余，2
◆変数の操作：[0008:へロンの数@1000以降]除算，2
◆変数の操作：[0007:へロンの数]除算，2
◆条件分岐：変数[0009:条件分岐用]が1
◆変数の操作：[0007:へロンの数]加算，500
◆
：分岐終了
◆変数の操作：[0010:繰り上がり用]代入，変数[0007]の値
◆変数の操作：[0010:繰り上がり用]除算，1000
◆変数の操作：[0008:へロンの数@1000以降]加算，変数[0010]の値

◆変数の操作：[0007:へロンの数]剰余，1000
◆変数の操作：[0011:ルートしたい値]代入，変数[0007]の値
◆変数の操作：[0012:ルートしたい値@1000]代入，変数[0008]の値
◆イベントの呼び出し：ルート計算
◆変数の操作：[0029:s]代入，変数[0017]の値
◆変数の操作：[0030:s@1000]代入，変数[0018]の値
◆スイッチの操作：[0002:せかい完了]をONにする
◆
：それ以外の場合
◆文章：まだ準備中
◆
：分岐終了
◆
：[ばふばふして欲しい]の場合
◆画面のシェイク：強さ9，速さ5，1.0秒(W)
◆
：[中z(ry)の場合
◆文章：ごめんね
：
：
：
：
：
◆ゲームオーバー
◆
：分岐終了
◆

```

$X=1000a+b$ として・・・繰り上がりも繰り下がりも考慮に入れなければいけない。

こうして、目出度くへロンの公式用の変数が整った。ちなみに、平方根部分は関数っぽくしたほうが読みやすくなるので、コモンイベントを作成してそれを呼び出すことにした。ソースコードは以下のとおりである。

```

◆条件分岐：変数[0012:ルートしたい値@1000]が0より小さい
◆文章：三角形にあらず。
◆ゲームオーバー
◆
:分岐終了
◆条件分岐：変数[0011:ルートしたい値]が0より小さい
◆文章：1000以下の値が負。
◆ゲームオーバー
◆
:分岐終了
◆条件分岐：変数[0012:ルートしたい値@1000]が0
◆条件分岐：変数[0011:ルートしたい値]が0以下
◆文章：三角形にあらず。
◆ゲームオーバー
◆
:分岐終了
◆
:分岐終了
◆変数の操作：[0021:繰り返し回数]代入, 0
◆変数の操作：[0013:AN]代入, 変数[0011]の値
◆変数の操作：[0014:AN@1000]代入, 変数[0012]の値
◆変数の操作：[0015:AN+1]代入, 0
◆変数の操作：[0016:AN+1@1000]代入, 0

```

数列の初期値を入力して、此処から先をひたすら繰り返す。割り算をするときは繰り返し下がり方を考慮に入れないといけないので面倒である。だが、1000以上と1000未満の分割をしっかりとやらないと何がなにだかわからなくなるので諦めて作る。

```

◆注釈：-----此処をしばらく繰り返す-----
◆繰り返し処理
◆条件分岐：変数[0021:繰り返し回数]が21以上
◆繰り返し処理の中断
◆
:分岐終了
◆変数の操作：[0017:AN+2]代入, 変数[0015]の値
◆変数の操作：[0017:AN+2]加算, 変数[0013]の値
◆変数の操作：[0018:AN+2@1000]代入, 変数[0014]の値
◆変数の操作：[0018:AN+2@1000]加算, 変数[0016]の値
◆変数の操作：[0009:条件分岐用]代入, 変数[0018]の値
◆変数の操作：[0009:条件分岐用]剰余, 2
◆変数の操作：[0018:AN+2@1000]除算, 2
◆変数の操作：[0017:AN+2]除算, 2
◆条件分岐：変数[0009:条件分岐用]が1
◆変数の操作：[0017:AN+2]加算, 500
◆
:分岐終了
◆変数の操作：[0010:繰り返し上がり用]代入, 変数[0017]の値
◆変数の操作：[0010:繰り返し上がり用]除算, 1000
◆変数の操作：[0018:AN+2@1000]加算, 変数[0010]の値
◆変数の操作：[0017:AN+2]剰余, 1000

```

ここまでが数列作成のアルゴリズム。三角形の条件を満たさない辺の長さは此処で弾かれる。

次に、二乗した時の値を比較するアルゴリズムを作る。s が取りうる値の制限を駆使して出来るだけ楽をできるようにする。

```

◆変数の操作：[0019:AN+2の2乗]代入，変数[0017]の値
◆変数の操作：[0019:AN+2の2乗]乗算，変数[0017]の値
◆変数の操作：[0020:AN+2の2乗@1000]代入，変数[0018]の値
◆変数の操作：[0020:AN+2の2乗@1000]乗算，変数[0018]の値
◆変数の操作：[0020:AN+2の2乗@1000]乗算，1000
◆変数の操作：[0010:繰り上がり用]代入，変数[0019]の値
◆変数の操作：[0010:繰り上がり用]除算，1000
◆変数の操作：[0020:AN+2の2乗@1000]加算，変数[0010]の値
◆変数の操作：[0019:AN+2の2乗]剰余，1000
◆変数の操作：[0010:繰り上がり用]代入，変数[0017]の値
◆変数の操作：[0010:繰り上がり用]乗算，変数[0018]の値
◆変数の操作：[0010:繰り上がり用]乗算，2
◆変数の操作：[0020:AN+2の2乗@1000]加算，変数[0010]の値
◆条件分岐：変数[0020:AN+2の2乗@1000]がV[0012]より大きい
◆変数の操作：[0013:AN]代入，変数[0017]の値
◆変数の操作：[0014:AN@1000]代入，変数[0018]の値
◆変数の操作：[0021:繰り返し回数]加算，1
◆
：分岐終了
◆条件分岐：変数[0020:AN+2の2乗@1000]がV[0012]より小さい
◆変数の操作：[0015:AN+1]代入，変数[0017]の値
◆変数の操作：[0016:AN+1@1000]代入，変数[0018]の値
◆変数の操作：[0021:繰り返し回数]加算，1
◆
：分岐終了

```

厳密な分岐をするために条件分岐機能は充実している。流石だぜ、RPG ツクール！と、無駄にテンションを上昇させて作成続投。

```

◆条件分岐：変数[0020:AN+2の2乗@1000]がV[0012]と同値
◆条件分岐：変数[0019:AN+2の2乗]がV[0011]より大きい
◆変数の操作：[0013:AN]代入，変数[0017]の値
◆変数の操作：[0014:AN@1000]代入，変数[0018]の値
◆変数の操作：[0021:繰り返し回数]加算，1
◆
：分岐終了
◆条件分岐：変数[0019:AN+2の2乗]がV[0011]より小さい
◆変数の操作：[0015:AN+1]代入，変数[0017]の値
◆変数の操作：[0016:AN+1@1000]代入，変数[0018]の値
◆変数の操作：[0021:繰り返し回数]加算，1
◆
：分岐終了
◆条件分岐：変数[0019:AN+2の2乗]がV[0011]と同値
◆繰り返し処理の中断
◆
：分岐終了
◆
：分岐終了
◆
：以上繰り返し
◆

```

二乗された時の値を比べて、二分法を成立させる。繰返す回数は 21 回あれば十分*4である。

6 作戦 3：面積の計算

与えられた四つの値を掛け算すれば完成だが、例の如く計算がかなり面倒である。ことのはのソースコード*5は以下のとおりである。

*4 \sqrt{N} を求めたいなら $\log_2 N$ 回繰返せばよい。

*5 ばふばふの画面シェイクの速さが素早くなっているが気にしない。

```

◆文章：ことのは：
：           今回の注文は？
◆選択肢の表示：三角形の面積を求める/ばふばふして欲しいVポートに：
： [三角形の面積を求める]の場合
◆条件分岐：スイッチ[0002:せかい完了]がON
◆注釈：-----s-aを求める-----
◆変数の操作：[0012:ルートしたい値@1000]代入，変数[0008]の値
◆変数の操作：[0012:ルートしたい値@1000]減算，変数[0004]の値
◆条件分岐：変数[0007:ヘロンの数]がV[0001]より小さい
◆変数の操作：[0012:ルートしたい値@1000]減算，1
◆変数の操作：[0011:ルートしたい値]代入，1000
◆変数の操作：[0011:ルートしたい値]加算，変数[0007]の値
◆変数の操作：[0011:ルートしたい値]減算，変数[0001]の値
◆
： それ以外の場合
◆変数の操作：[0011:ルートしたい値]代入，変数[0007]の値
◆変数の操作：[0011:ルートしたい値]減算，変数[0001]の値
◆
： 分岐終了
◆イベントの呼び出し：ルート計算
◆変数の操作：[0023:s-a]代入，変数[0017]の値
◆変数の操作：[0024:s-a@1000]代入，変数[0018]の値
◆注釈：-----s-bを求める-----

```

☆以下、しばらく省略☆

```

◆注釈：-----ここからs(s-a)の掛け算-----
◆変数の操作：[0031:s(s-a)]代入，変数[0029]の値
◆変数の操作：[0031:s(s-a)]乗算，変数[0023]の値
◆変数の操作：[0032:s(s-a)@1000]代入，変数[0030]の値
◆変数の操作：[0032:s(s-a)@1000]乗算，変数[0024]の値
◆変数の操作：[0032:s(s-a)@1000]乗算，1000
◆変数の操作：[0010:繰り上がり用]代入，変数[0031]の値
◆変数の操作：[0010:繰り上がり用]除算，1000
◆変数の操作：[0032:s(s-a)@1000]加算，変数[0010]の値
◆変数の操作：[0031:s(s-a)]剰余，1000
◆変数の操作：[0010:繰り上がり用]代入，変数[0023]の値
◆変数の操作：[0010:繰り上がり用]乗算，変数[0030]の値
◆変数の操作：[0032:s(s-a)@1000]加算，変数[0010]の値
◆変数の操作：[0010:繰り上がり用]代入，変数[0024]の値
◆変数の操作：[0010:繰り上がり用]乗算，変数[0029]の値
◆変数の操作：[0032:s(s-a)@1000]加算，変数[0010]の値
◆注釈：-----ここから(s-b)(s-c)の掛け算-----
◆変数の操作：[0033:(s-b)(s-c)]代入，変数[0025]の値

```

☆以下同様に計算☆

相当めんどくさい。次はいよいよ面積の計算である。

面積の計算は、10桁以上の面積を取りうるので変数が三つ必要である。

```

◆注釈：-----ここから面積計算-----
◆変数の操作：[0035:面積@100万]代入，変数[0032]の値
◆変数の操作：[0035:面積@100万]乗算，変数[0034]の値
◆変数の操作：[0037:面積]代入，変数[0031]の値
◆変数の操作：[0037:面積]乗算，変数[0033]の値
◆変数の操作：[0036:面積@1000]代入，変数[0031]の値
◆変数の操作：[0036:面積@1000]乗算，変数[0034]の値
◆変数の操作：[0010:繰り上がり用]代入，変数[0036]の値
◆変数の操作：[0010:繰り上がり用]除算，1000
◆変数の操作：[0035:面積@100万]加算，変数[0010]の値
◆変数の操作：[0036:面積@1000]剰余，1000
◆変数の操作：[0038:繰り上がり用その2]代入，変数[0032]の値
◆変数の操作：[0038:繰り上がり用その2]乗算，変数[0033]の値
◆変数の操作：[0010:繰り上がり用]代入，変数[0038]の値
◆変数の操作：[0010:繰り上がり用]除算，1000
◆変数の操作：[0035:面積@100万]加算，変数[0010]の値
◆変数の操作：[0038:繰り上がり用その2]剰余，1000
◆変数の操作：[0036:面積@1000]加算，変数[0038]の値
◆変数の操作：[0010:繰り上がり用]代入，変数[0037]の値
◆変数の操作：[0010:繰り上がり用]除算，1000
◆変数の操作：[0036:面積@1000]加算，変数[0010]の値
◆変数の操作：[0037:面積]剰余，1000
◆変数の操作：[0010:繰り上がり用]代入，変数[0036]の値
◆変数の操作：[0010:繰り上がり用]除算，1000
◆変数の操作：[0035:面積@100万]加算，変数[0010]の値
◆変数の操作：[0036:面積@1000]剰余，1000

```

7 孔明の罠

さて、これで目出度く面積が求められた。後は与えられた変数を@百万、@千、@零の順番に並べれば面積が現れ・・・と思った瞬間に戦慄が走った。おぞましい事にも**桁が落ちる**のである。

というのは、例えば@百万が123，@千が45，@零が678の時、正しくは「123045678」と表示されるべきところを「12345678」となってしまうのである。つまり、値が100以上、100未満10以上、10未満の時に備えて、条件分岐をしなければならないのである。単純に考えて9通りの条件分岐を書かなければならない。なぜなら、RPG ツクールにはString 型の変数がないから・・・

いや、ないことはない。主人公の名前はString 型変数として扱うことが出来るじゃないか！この閃きのもと作られたのが以下のソースコードである。



わたし、負けない！！

```

◆条件分岐：変数[0036:面積@1000]が100より小さい
◆条件分岐：変数[0036:面積@1000]が10より小さい
◆主人公の名前変更：1000の位用の名前を001に変更
◆
：それ以外の場合
◆主人公の名前変更：1000の位用の名前を01に変更
◆
：分岐終了
◆
：それ以外の場合
◆主人公の名前変更：1000の位用の名前を1に変更
◆
：分岐終了
◆条件分岐：変数[0037:面積]が100より小さい
◆条件分岐：変数[0037:面積]が10より小さい
◆主人公の名前変更：0の位用の名前を001に変更
◆
：それ以外の場合
◆主人公の名前変更：0の位用の名前を01に変更
◆
：分岐終了
◆
：それ以外の場合
◆主人公の名前変更：0の位用の名前を1に変更
◆
：分岐終了

```

命名神の怒りに触れる名前のオンパレード

こうして、目出度く文字列保存が出来たので答えを列挙する。条件分岐をしっかりこなさないと「00045678」みたいに表示されて非常にかっこ悪い。まあ、この部分の条件分岐は他のに比べれば楽勝なのだが。

```

◆条件分岐：変数[0035:面積@100万]が0より大きい
◆文章：面積は%V[35]%N[4]%V[36]%N[5]%V[37]だよ。
◆
：それ以外の場合
◆条件分岐：変数[0036:面積@1000]が0より大きい
◆文章：面積は%V[36]%N[5]%V[37]だよ。
◆
：それ以外の場合
◆文章：面積は%V[37]だよ。
◆
：分岐終了
◆
：分岐終了
◆
：それ以外の場合
◆文章：まだ準備中
◆
：分岐終了
◆
：[[ばふばふして欲しい]]の場合
◆画面のシェイク：強さ9，速さ9，1.0秒(W)
◆
：[[ボートに乗りたい]]の場合
◆文章：Nice boat.
◆ゲームオーバー
◆
：分岐終了
◆

```

これぞ、RPG ツールの真骨頂！

こうして、晴れて三角形の面積を求めることが出来た。

8 作成を振り返る

作成したのはいいが、それにしても精度の低い作品となってしまった。一番の敗因は平方根を求める時に、 $s(s-a)(s-b)(s-c)$ の平方根をとらずに、各々の平方根を掛け合わせてしまったことだろう。 $(s-a)$ などが 1 の時は三角形の面積が 0 になってしまうのは夢も希望もへったくれもないといわざるを得ない。小物を相手にしないとえば聞こえがいいが。この作品をよりよい作品に出来ると思った人は更なる挑戦をしてみると良いだろう。答えを得るために敵を倒さないといけないなどのストーリーを加えれば普通のプログラミングコンテストに励む人々を敗者の上から目線で「うわぁ～、この人普通すぎるよぉ」と見下ろすことが可能になるであろう。

9 オマケ：作者紹介

内容の正しさは保障しませんが、気になる方はどうぞどうぞ。

作者ハンドルネーム：49（しく）

中の方は二人いる。ハンドルネーム「4 様」と「999（くくく）」が共同ゲームを作る時に生まれた名前。

所属：理科 類（運がよければ理学部物理学科へ進学）

不可を取った科目：形式言語理論（試験時間を間違えて不可となった）

第二外国語：ひんたば語

将来の夢：3LDK

コミケ出店名：劇団しく*6（くじが通ったら）

憧れる神：ライト < カフカ

*6 崩壊しました。2010/05 追記