

[バンタン体験授業(2019年05月版)]

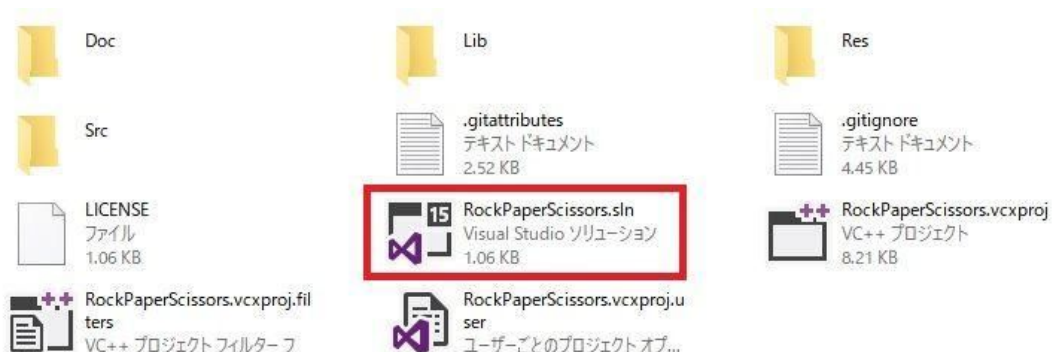
じゃんけんゲームを作ってみよう

1 Visual Studioを起動する

1.1 SLNファイルを開く

このテキストでは、「C++(シー・プラス・プラス)」というプログラミングを使って、簡単なコンピューター・ゲームを作っていきます。
また、ゲームを作るために「Visual Studio(ビジュアル・スタジオ)」というツールを使います。

さっそくVisual Studioを起動しましょう。
デスクトップの「RockPaperScissors(ロック・ペーパー・シザーズ)」というフォルダをダブルクリックして開いてください。その中に
「RockPaperScissors.sln(ロック・ペーパー・シザーズ・エス・エル・エヌ)」というファイルがありますので、このファイルをダブルクリックしてください。
するとVisual Studioが起動します。



1.2 プログラムを実行する

Visual Studioを起動したら、もうプログラムを実行する準備はできています。
プログラムを実行するには、上の方にある「ローカルWindowsデバッガー(ローカル・ウィンドウズ・デバッガー)」というボタンをクリックします。



しばらく待つと、プログラムが実行されてウィンドウが表示されます。



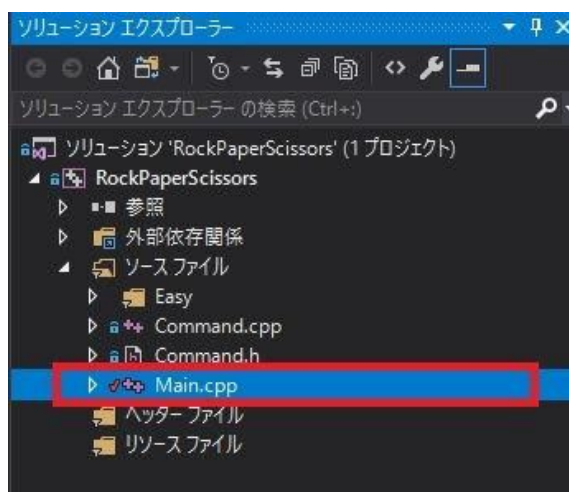
どうやら、じゃんけんが行われて勝利したようですね。
よく分かりませんが、とりあえず指示に従って何かキーボードのキーを押してみま
しょう。そうすると、ウィンドウが閉じると思います。

2 プログラムを見てみよう

2.1 Main.cppを開く

さきほど実行したプログラムはどのような仕組みになっているのでしょうか。
それを調べるためには、「Main.cpp(メイン・シー・ピー・ピー)」というファイル
を開かなくてはなりません。

Visual Studioの右側に「ソリュー
ション エクスプローラー」というウィ
ンドウがあると思います。その中に
「ソース ファイル」というフォルダア
イコンがあり、さらにその下に「
Main.cpp」と書かれた行があるのが分
かるでしょうか。この行をダブルクリッ
クするとMain.cppを開くことができま
す。



2.2 C++言語

Main.cppは「C++(シー・プラス・プラス)」という言語で書かれています。わたし
たちが日本語を話すように、世界では国によってさまざまな言語が使われていますよ
ね。それと同じように、コンピューターの世界にもさまざまな言語が存在するので
す。コンピューターのプログラムは「プログラミング言語」と呼ばれるもので書かれ
ます。

それでは、Main.cppを見ていきましょう。しかし、その前にひとつ注意点を伝えて
おきます。プログラムでは見間違いやすい文字を使います。以下の文字は特に見間違
えやすいので気をつけて読んでください。

- 1(いち)、i(小文字のアイ)、I(大文字のアイ)、l(小文字のエル)、|(垂直線)
- 0(数字のゼロ)、o(小文字のオー)、O(大文字のオー)

他にも、アルファベットの大文字小文字は見間違いやすいものです。注意して読むよ
うにしてください。なお、このテキストで使われているのはほとんどが小文字です。

2.3 インクルード指令

```
1 #include "Command.h"
```

「#include」(シャープ・インクルード)は「インクルード指令」といって、プログラムで使う部品を選択します。「Command.h」(コマンド・ドット・エイチ)というのが部品ファイルの名前です。Command.hには、この「じゃんけんゲーム」で使われているさまざまな部品が含まれています。

2.4 イント・メイン

```
3 // ここからプログラムの実行が開始される
4 int main()
5 {
```

プログラムは「int main」(イント・メイン)で始まります。mainのすぐ後ろには「(」(丸かっこ)と「)」(閉じ丸かっこ)があり、その次の行の「{」(波かっこ)から、ファイルの一番下の「}」(閉じ波かっこ)までが実行されるプログラムの内容になっています。つまり、この波かっこの内側に書いてあるものが「じゃんけんゲームのプログラム」ということです。プログラムは上の行から順番に実行されます。

3行目の緑色の部分は「コメント」といいます。コメントは人間がプログラムを読みやすくするためのものです。コンピューターはコメントを無視します。コメントは「//」(スラッシュ・スラッシュ)ではじまり、改行で終わります。

2.5 プログラムの初期化と終了

```
6 // プログラムの初期化处理
7 initialize("じゃんけんゲーム");
```

「initialize」(イニシャライズ)は、Command.hに含まれている部品をセットアップして、プログラムで使える状態にします。丸かっこと閉じ丸かっこの内側に書かれているのは、プログラムを実行して表示されたウィンドウの上枠に表示された文章です。C++では、「"」(ダブル・クォーテーション)という記号で囲むことで、この範囲はプログラムではなくて文章ですよ、ということを示します。

```
37 // プログラムの終了処理
38 finalize();
```

プログラムの最後にある「finalize」(ファイナライズ)は、initializeでセットアップした部品を解体し、プログラムを安全に終了できるようにします。

2.6 画像を表示する

```
9 // 背景を表示
10 set_image(No_0, 400, 300, "bg_paper.jpg");
```

「set_image」(セット・イメージ)は画面に画像を表示します。丸括弧の内側には「,(カンマ)」で区切られた4つのパラメーターがあります。

最初の「No_0」は画像の管理番号です。その次の「400」と「300」は画像を表示する位置で、ひとつめがウィンドウの左端からのピクセル数、ふたつめが上端からのピクセル数です。ウィンドウの大きさは横が800、縦が600です。つまり、ウィンドウの中心に画像を表示しようとしているのですね。

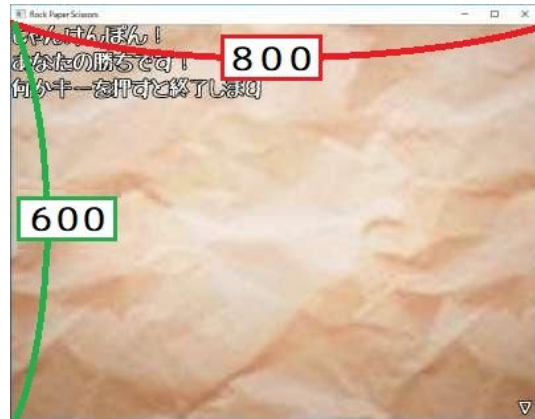
最後にある「"背景_紙.jpg"」は表示する画像ファイルの名前です。C++のファイル名は文章と同じ扱いなので、「"」で囲みます。

ゲームの雰囲気を変えるために、背景を変えてみましょう。10行目のファイル名を、次のように書き換えてください。

```
9 // 背景を表示
10 set_image(No_0, 400, 300, "bg_brick.jpg");
```

日本語を入力するには、キーボード左上にある「半角/全角」(はんかく/ぜんかく)キーを押して「全角モード」にします。日本語の入力を漢字に変換するには、キーボード中央下にある細長い「スペースバー」を押します。押すたびに変換候補が切り替わります。キーボード右側の「Enter(エンター)」キーを押すと、表示された漢字が決定されます。入力が終わったら、もう一度「半角/全角キー」を押すと「半角モード」に戻ります。C++のプログラムを書くときは、日本語を入力するとき以外は常に「半角モード」を使います。

文章を書き換えたら「ローカルWindowsデバグガー」ボタンをクリックして実行してください。
レンガの背景が表示されていれば成功です。



2.7 文章を表示する

```
12 // 文章を表示
13 printf("じゃんけんぽん!");
```

「printf」(プリント・エフ)を使うと、画面に文章を表示できます。丸括弧の内側に書かれた文章が画面に表示されます。

2.8 数値に名前を付ける

```
15 // じゃんけんの手を定義する
16 int gu = 0;
17 int choki = 1;
18 int pa = 2;
```

16~18行目は、じゃんけんの手に数値を割り当てています。じゃんけんでは「グー」、「チョキ」、「パー」の3種類の手で勝敗を決めます。ところが、コンピューターというのは数字の扱いは得意なのですが、文字の並びを扱うのはあまり得意ではありません。そこで、それぞれの手に番号を割り当ててあげます。

番号を使うとコンピューターは助かりますが、今度は人間のほうが理解しにくくなってしまいます。人間もコンピューターも理解しやすいプログラムが書けるように、C++言語には「数値に名前を付ける機能」が用意されています。数値に名前をつけるには

```
int 名前 = 数値;
```

と書きます。この式は「これから整数に名前を付けますよ」という意味です。「int」(イント)は「integer(インテジャー、「整数」という意味)」を省略したもので、「名前 = 数値」の部分で数値に名前を付けています。最後の「;(セミコロン)」は行の終わりを示します。C++言語では、行の終わりにセミコロンを付ける決まりになっています。

名前を付けたことで、コンピューターはプログラムにguと書かれていたら、それを数値の0として認識するようになります。同様にchokiと書けば1、paと書けば2と認識します。

2.9 数値に名前をつける(その2)

```
20 // プレイヤーの手を決める
21 int player_hand = gu;
22
23 // コンピューターの手を決める
24 int cpu_hand = choki;
```

21行目と24行目は、それぞれプレイヤーの手とコンピューターの手に名前を付けています。

「数値」の部分には、16～18行目で付けた名前を使っています。

2.10 一定時間待つ

```
26 // 1秒待つ
27 wait(1);
```

「wait」(ウェイト)を使うと、指定した秒数だけプログラムの実行を停止できます。

2.11 条件によって違うことをする

```
29 // 勝敗を判定する
30 if (player_hand == gu && cpu_hand == choki) {
31     printf("あなたの勝ちです!");
32 }
```

条件に応じて違うことをするには「if」(イフ、「もしも～ならば」という意味)を使います。ifに続く「()」の内側に条件を書きます。条件が成立したときだけ、そのあとの「{ }」の内側にあるプログラムが実行されます。

数値が同じかどうかを調べるには「==」(イコール・イコール)記号を使います。2つ以上の条件を調べるには、条件のあいだに「&&」(アンド・アンド)記号を書きます。つまり、このifの条件は、

プレイヤーの手がグー のとき	かつ	コンピューターの手がチョキ のとき
player_hand == gu	&&	cpu_hand == choki

という意味になります。

それでは、ちょっと出す手を変えてみましょう。
プレイヤーの手とコンピューターの手を、次のように書き換えてください。

```
20 // プレイヤーの手を決める
21 int player_hand = choki;
22
23 // コンピューターの手を決める
24 int cpu_hand = pa;
```

手を書き換えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。

...おや？
勝敗が表示されなくなっていましたね。

勝敗が表示されなくなった理由は、30行目のifに書かれた条件にあります。現在の条件は「プレイヤーの手がグー、かつ、コンピューターの手がチョキの場合」なので、それ以外の手を指定しても、条件を満たせないのです。



他の条件でも勝敗を表示するには、ifを増やして対応できる条件を増やすのが簡単です。次のように30～32行目にプログラムを書き加えてください。

```
29 // 勝敗を判定する
30 if (player_hand == gu && cpu_hand == choki) {
31     printf("あなたの勝ちです！");
32 }
33 if (player_hand == choki && cpu_hand == pa) {
34     printf("あなたの勝ちです！");
35 }
```

書き加えたら「ローカルWindowsデバッガー」をクリックして実行しましょう。

「あなたの勝ちです！」と表示されたら成功です。



3 勝敗条件を追加しよう

3.1 パーVSグーの条件を加える

グーVSチョキ、チョキVSパーの次は、パーVSグーのifを書き加えましょう。
まずプレイヤーとコンピューターの手を書き換えてください。

```
20 // プレイヤーの手を決める
21 int player_hand = pa;
22
23 // コンピューターの手を決める
24 int cpu_hand = gu;
```

そして、パーVSグーのifを書き加えてください。

```
33 if (player_hand == choki && cpu_hand == pa) {
34     printf("あなたの勝ちです！");
35 }
36 if (player_hand == pa && cpu_hand == gu) {
37     printf("あなたの勝ちです！");
38 }
39
40 // 何かキーが押されるまで待つ
```

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。

「あなたの勝ちです！」と表示されたら成功です。

3.2 コンピューターが勝利する条件を加える

これで、プレイヤーが勝つ条件を全て加えました。しかし、このままではコンピューターが勝ったとしても何も表示されません。ちょっと可哀想なので、コンピューターが勝つ条件も加えましょう。

まず、プレイヤーとコンピューターの手を、以下のように書き換えてください。

```
20 // プレイヤーの手を決める
21 int player_hand = gu;
22
23 // コンピューターの手を決める
24 int cpu_hand = pa;
```


次に、プレイヤーが勝つ条件のifの下に、コンピューターが勝つ条件のifを書き加えてください。

```
36  if (player_hand == pa && cpu_hand == gu) {
37      printf("あなたの勝ちです！");
38  }
39  if (player_hand == gu && cpu_hand == pa) {
40      printf("あなたの負けです");
41  }
42
43  // 何かキーが押されるまで待つ
```

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。
「あなたの負けです。」と表示されたら成功です。

<練習1>

グーVSパーのifの下に、チョキVSグーと、パーVSチョキのifを書き加えましょう。

3.3 あいこの条件を加える

じゃんけんでは、双方が同じ手を出した場合は「あいこ」になります。
だから、あいこの条件も付け加えましょう。まずはあいこになるように手を変えてください。

```
20  // プレイヤーの手を決める
21  int player_hand = gu;
22
23  // コンピューターの手を決める
24  int cpu_hand = gu;
```

そして、あいこのifを書き加えてください。

```
45  if (player_hand == pa && cpu_hand == choki) {
46      printf("あなたの負けです");
47  }
48  if (player_hand == cpu_hand) {
49      printf("あいこです");
50  }
51
52  // 何かキーが押されるまで待つ
```

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。
「あいこです。」と表示されたら成功です。

4 手を選ぶようにしよう

4.1 プレイヤーの手を選ぶ

プログラムに手を書いているので、出す手を変えるにはプログラムを書き換えなくてはなりません。

これは面倒なので、実行するときに手を選ぶようにしましょう。

プレイヤーの手を決めるプログラムを、次のように書き換えてください。

```
20 // プレイヤーの手を決める
21 int player_hand = select(3, "グー", "チョキ", "パー");
22
23 // コンピューターの手を決める
24 int cpu_hand = choki;
```

「select」(セレクト)を使うと、選択肢を表示してプレイヤーに選んでもらうことができます。

最初の数字が選択肢の数、そのあとの3つが選択肢として表示される文章です。

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。

グー、チョキ、パーの選択肢が表示されたら成功です。

上キーと下キーで好きな手を選びます。Enterキーを押すと決定されます。

4.2 コンピューターの手を選ぶ

今度はコンピューターの手が毎回変化するようにしましょう。

```
20 // プレイヤーの手を決める
21 int player_hand = select(3, "グー", "チョキ", "パー");
22
23 // コンピューターの手を決める
24 int cpu_hand = random(0, 2);
```

「random」(ランダム)は整数を無作為に選択します。

ひとつめの数字が選択される範囲の最小値、ふたつめの数字が最大値です。

今回の場合は最小値が0、最大値が2なので、0、1、2のいずれかの数値が無作為に選ばれて、コンピューターの手になるわけです。

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。

選択肢を決定したとき、実行するたびに勝敗が変化したら成功です。

5 手を表示しよう

5.1 プレイヤーの手を表示する

じゃんけんの結果は表示されますが、実際に何を出したのかは分かりません。インチキされてるのかもしれませんがね。そこで、出した手を表示することにしてしまおう。
コンピューターの手を決めるプログラムの下に、次のプログラムを書き加えてください。

```
23 // コンピューターの手を決める
24 int cpu_hand = random(0, 2);
25
26 xyprintf(100, 260, "[あなたの手]");
27 if (player_hand == gu) {
28     set_image(No_1, 200, 400, "gu.png");
29 }
30 if (player_hand == choki) {
31     set_image(No_1, 200, 400, "choki.png");
32 }
33 if (player_hand == pa) {
34     set_image(No_1, 200, 400, "pa.png");
35 }
36
37 // 1秒待つ
38 wait(1);
```

26行目のxyprintf(エックス・ワイ・プリント・エフ)は、最初の2つの数値で表示位置を自由に決められるバージョンです(注: xyprintfはこのテキストのために作成したもので、一般のC++言語には存在しません)。

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。選択肢を決定したとき、手に対応する画像が表示されたら成功です。

<練習2>

プレイヤーの手を表示するプログラムの下に、コンピューターの手を表示するifを書き加えましょう。画像番号はNo_2を使ってください。

5.2 音声を鳴らそう

じゃんけんする時に小鼓(こつづみ)の音を鳴らしてみましょう。
音声を鳴らすには「play_sound」(プレイ・サウンド)を使います。プレイヤーとCPUの手を表示するプログラムの下に、次のプログラムを書き加えてください。

```
43  if (cpu_hand == pa) {  
44      set_image(No_2, 600, 400, "パー.png");  
45  }  
46  
47  play_sound("kotudumi.mp3");  
48  
49  // 1秒待つ  
50  wait(1);
```

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行...の前に、音量をチェックしましょう。音量は右下のスピーカーアイコンから変更できます。20くらいがいいと思います。大丈夫そうならボタンをクリックして実行してください。選択肢を決定したとき、小鼓の(ような)音声が再生されたら成功です。

5.3 BGMを鳴らそう

じゃんけんを始める前にBGMを鳴らしてみましょう。
BGMを鳴らすには「play_bgm」(プレイ・ビー・ジー・エム)を使います。
初期化を行うプログラムの下に、次のプログラムを書き加えてください。

```
6  // プログラムの初期化処理  
7  initialize("じゃんけんゲーム");  
8  
9  play_bgm("bgm_normal.mp3")  
10  
11 // 背景を表示  
12 set_image(No_0, 400, 300, "bg_brick.jpg");
```

書き加えたら「ローカルWindowsデバッガー」ボタンをクリックして実行してください。BGMが再生されたら成功です。

5.4 何度でも遊べるようにしよう

一度実行すれば何度でも遊べるようにしましょう。

プログラムを繰り返し実行するには「for」(フォー)を使います。繰り返したい範囲の手前に「for (;;) {」と書き、最後に「}」を書くと、その範囲にあるプログラムは、何度でも繰り返し実行されるようになります。

じゃんけんゲームのほぼ全体が繰り返されるように、play_bgmのすぐ下に「for (;;) {」を、finalizeのすぐ上に「}」を書き加えてください。

```
9   play_bgm("bgm_normal.mp3");
10
11  for (;;) {
12      // 背景を表示
13      set_image(No_0, 400, 300, "bg_brick.jpg");
```

...

```
74      // 何かキーが押されるまで待つ
75      printf("何かキーを押すと勝負を続けます");
76      wait_any_key();
77  }
78
79      // プログラムの終了処理
80      finalize();
```

書き加えたら実行してください。

何度でもじゃんけんができるようになっていれば成功です。

6 完成！

おめでとう！

これでじゃんけんゲームのテキストは終了です。

いくつかの練習問題を用意しましたが、あとはあなたの好きなように改造してってください。お疲れ様でした。

<練習3>

勝敗によって異なる音声を再生してみましょう。
音声はResというフォルダに入っています。

<練習4>

勝敗を文章ではなく画像で表示してみましょう。
画像はResというフォルダに入っています。

<練習5>

あなたの好きなようにゲームを改造してください。
Resフォルダの画像や音声は自由に使って構いません。