

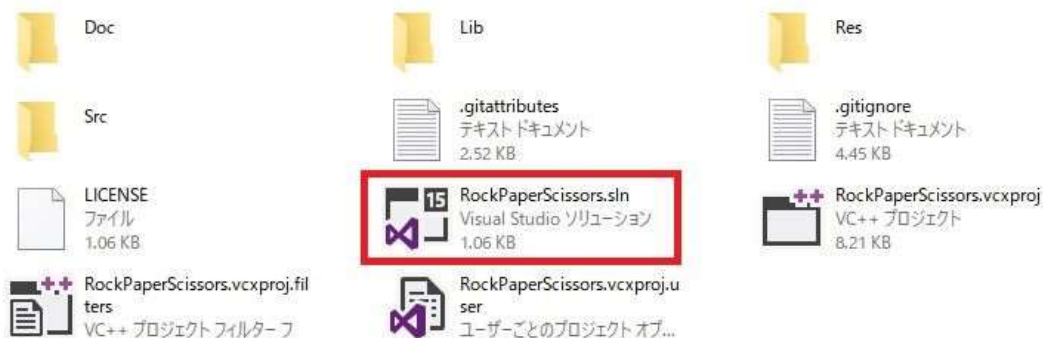
# じゃんけんゲームを作ってみよう

## 1 Visual Studio を起動する

---

### 1.1 SLN ファイルを開く

RockPaperScissors.sln(ろっく・ペー・パー・しざーず・えすえるえぬ)というファイルをダブルクリックしてください。すぐに Visual Studio(びじゅある・すたじお)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」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を押してみましょう。すると、ウィンドウが閉じると思います。

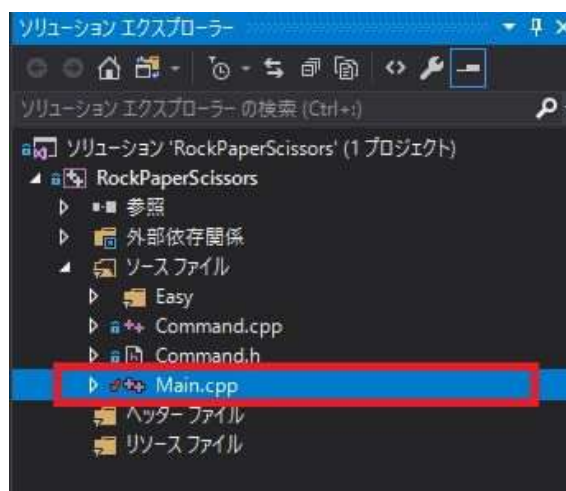


## 2 プログラムを見てみよう

### 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、「Main.cpp(めいん・しーピーピー)」というファイルを開きましょう。

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++」(しー・ぷらす・ぷらす)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。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。コンピューター世界の言語は「プログラミング言語」と呼ばれます。

それでは、Main.cpp を見ていきましょう。

## 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。

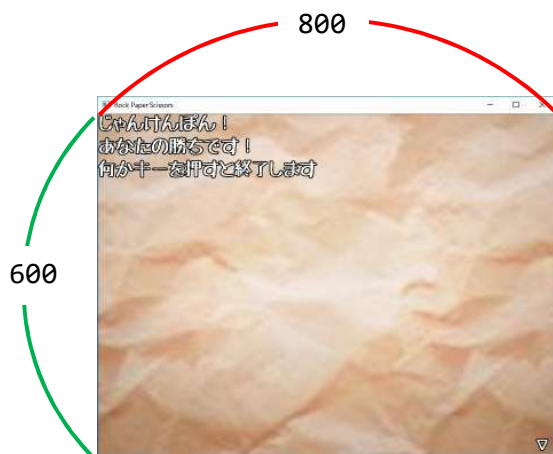
```
37 // プログラムの終了処理
38 finalize();
```

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。

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

```
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 デバッガー」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 set_text("じゃんけんぽん!");
```

「set\_text」(セット・テキスト)は文章を表示します。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、コンピュータはこういった文字の並びを扱うのがあまり得意ではありません。そこで、数値をそれぞれの手に割り当ててあげます。

数値に名前をつけるには「int 名前 = 数値;」と書きます。「int」(イント)は integer(インテジャー、整数)を省略したもので、「これから整数に名前を付けますよ」という意味になります。「名前 = 数値」の部分が数値に名前を付けている部分です。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 として認識することができるようになります。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     set_text("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)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。

ここで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、if の条件を見れば分かります。プレイヤーの手がグー、コンピューターの手がチョキの場合しか条件を満たさないからです。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     set_text("あなたの勝ちです!");
32 }
33 if (player_hand == choki && cpu_hand == pa) {
34     set_text("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」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 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     set_text("あなたの勝ちです!");
35 }
36 if (player_hand == pa && cpu_hand == gu) {
37     set_text("あなたの勝ちです!");
38 }
39
40 // 何かキーが押されるまで待つ
41 set_text("何かキーを押すと終了します");
```

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。  
「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     set_text("あなたの勝ちです!");
38 }
39 if (player_hand == gu && cpu_hand == pa) {
40     set_text("あなたの負けです");
41 }
42
43 // 何かキーが押されるまで待つ
44 set_text("何かキーを押すと終了します");

```

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。

「あなたの負けです。」と表示されたら成功です。

#### <Challenge!>

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     set_text("あなたの負けです");
47 }
48 if (player_hand == cpu_hand) {
49     set_text("あいこです");
50 }
51
52 // 何かキーが押されるまで待つ
53 set_text("何かキーを押すと終了します");

```

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。

「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ないですね。そこで、出した手を表示することにしましょう。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 set_text(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 `set_text` は、文章を表示する位置を指定したいときの書き方です。最初の2つの数値で表示位置を決めます。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、手に対応する画像が表示されたら成功です。

#### <Challenge!>

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。以下の2点に注意してください。

- ・画像番号は「No\_2」を使ってください。
- ・プレイヤーの手と重ならないように、表示する位置を調整して下さい。

### 5.2 音声を鳴らそう

音声を鳴らすには「`play_sound`」(ふれい・さうんど)を使います。

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。

```
43 if (cpu_hand == pa) {
44     set_image(No_2, 600, 400, "pa.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 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)音声再生されたら成功です。

#### <Challenge!>

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。

### 5.3 何度でも遊べるようにする

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 (;;) {」を書き、最後に「}」を書くと、その範囲にあるプログラムが、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、initialize のすぐ下に「for (;;) {」を、finalize のすぐ上に「}」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 for (;;) {
10 // 背景を表示
11 set_image(No_0, 400, 300, "bg_brick.jpg");
```

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

### 5.4 文章と画像を消去する

じゃんけんを繰り返すと、文章がどんどん下に表示されて、最後には画面の外にいつてしまいます。また、じゃんけんの手の画像も、次の手が決定されるまで表示されっぱなしです。こういった状態にならないようにするには、じゃんけんを繰り返す瞬間に文章と画像を消してしまうのが簡単です。文章を消すには reset\_all\_text(りせつと・おーる・てきすと)を使います。そして、画像を消すには reset\_all\_image(りせつと・おーる・いめーじ)を使います。この 2 つを wait\_any\_key の下に書き加えてください。

```

73     set_text("何かキーを押すと勝負を続けます");
74     wait any key();
75     reset_all_text();
76     reset_all_image();
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。

じゃんけんを繰り返したとき、いつも同じ位置に文章が表示されたら成功です。

## 6 勝敗の回数を記録してみよう

### 6.1 勝った回数を記録する

勝ったり負けたりした回数をコンピューターが数えてくれると楽ですね。

勝った回数に名前をつけることでそれが可能になります。initialize の下に、勝った回数を示す win\_count (ういん・かうんと) という名前を追加してください。

```

6 // プログラムの初期化処理
7 initialize("じゃんけんゲーム");
8
9 // 勝った回数を記録する.
10 int win_count = 0;
11
12 for (;;) {
13     // 背景を表示
14     set_image(No_0, 400, 300, "bg_brick.jpg");

```

最初、win\_count は 0 という数値になっています。勝つたびに win\_count が指している数値を増やすことができれば、勝った回数を記録することができます。

勝敗を判定するプログラムに、次のプログラムを書き加えてください。

```

56 // 勝敗を判定する
57 if (player_hand == gu && cpu_hand == choki) {
58     set_text("あなたの勝ちです!");
59     play_sound("win.mp3");
60     win_count = win_count + 1;
61 }

```

今回追加したプログラムの意味は

「現在の win\_count が指している数値に 1 を足す (win\_count + 1)。そして、win\_count がその数値を示すように変更する (win\_count =)。」

というものです。数学ではありえない不思議な式ですが、C++ 言語のイコール記号は「数値に名前をつけるための記号」だということを思い出してください。つまり、「= 記号の右側にある式を計算した結果に、左側にある名前を付けている」わけです。

**<Challenge!>**

チョキ vs パー、パー vs グーのときも、勝った回数が 1 増えるようにしましょう。

## 6.2 勝った回数を表示する

勝った回数を表示するには、文章と同じく `set_text(せつと・てきすと)` を使います。

「じゃんけんぽん！」という文章の手前に、次のプログラムを書き加えてください。

```
16 // 文章を表示
17 set_text("勝った回数: %d", win_count);
18 set_text("じゃんけんぽん!");
```

「%d(ぱーせんと・でいー)」というのは、コンピューターに「ここに数値を表示してください。」と指示するための記号です。表示する数値は、文章のうしろにカンマで区切って指定します。

書き加えたら実行してください。

勝った回数が表示されていたら成功です。

**<Challenge!>**

負けた回数を記録して表示してみましょう。

**<Challenge!>**

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。

**<Challenge!>**

「あいこ」のときは自動的に次のじゃんけんが始まるようにしてみましょう。