

randomize

ゲームを作る際に、ランダムな数値が必要なときもあると思います。ここではその方法を説明します。

ランダムな数を使う場合に、必要なことは次のふたつです。

① : cpp 内に

```
randomize();
```

と記述すること

これでランダムな数字を使うことを宣言します。

② : ランダムな数値を決定したいところで (例として変数 a にランダムな値を代入する)

```
a = random(入りたい値の最大値);
```

と記述すること

例

```
a = random(6);
```

この場合注意してほしいことは a に入る値が 1~6 ではなく 0~5 が入るということです。

なので、a に入る値を 1~6 にしたい場合には `random(6)` のあとに `+ 1` という文を書き足してください。

これでランダムな数を作ることができるようになりました。

ですが、この方法では短時間で連続使用すると同じ数字が連続して出やすくなる、という欠点があります。

それを回避する方法はいくつかあるのですがここではそのうちの一つを紹介します。

ここで紹介するのは for 文を使った方法で、同じ数字が連続して出ないようにするためのものです。

使用するコンポーネントは Label と Button を 1 つずつです

まずヘッダーで int 型の変数を二つ用意します。

ここでは a と b とします。

<ヘッダー>

```
private:           //ユーザー宣言
    int a, b;
```

次に、cpp での記述は以下のようになります。

<cpp>

```
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    randomize();           //←でランダムな数を使うことを宣言

    Label1->Caption = a;

    Button1->Caption = “ランダム” ;
}
```

```
//-----  
void __fastcall TForm1::Button3Click(TObject *Sender)  
{  
    b = a;                //変数 b に変数 a を代入  
  
    for (;;) {  
        a = random(6)+1;  
        if(a != b)      ← ここで変数 b と変数 a  
                           ← 同じだったときに  
                           ← もう一度ランダムに  
                           ← 変数 a を選ばせる。  
        break;  
    }  
  
    Label2->Caption = a;  
}  
//-----
```

これで同じ数字が連続して出てくることがなくなりました。
これ以外にもいくつか方法があるので、気になった人は周りの上回生に聞いてください。

ここに書いてあるのは基本的な使い方ですが、これを応用して面白いゲームを作ってくれることを期待しています。
頑張ってください。