

プログラミング基礎 第4回 レポート 解答例

提出期限: 2021 年 5 月 18 日 (火) 23:59

提出場所: manaba の指定場所に添付ファイルで提出する

注意: 解答には, ソースコードをどのように改変したか, コンパイルやプログラム実行時のエラーや警告, スクリーンショットなどを含めるなど, わかりやすくなる工夫であれば自由に含めてよい. ページ数もいくら増やしてもよい.

番号 21C1XXX

氏名 藤江 真也

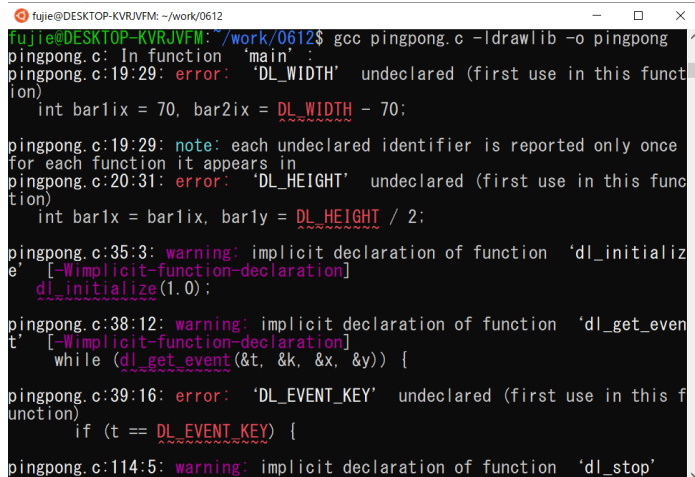
課題 講義内で説明した `pingpong.c` について, 以下の問いに答えよ.

- (1) プログラム内で 3 種類のヘッダーを読み込んでいる(`#include` で始まる行がそれに該当する). このうちの一つのヘッダーの読み込みをやめたとき(行ごと削除する), コンパイルするとどのようなエラーが発生するか答えよ. また, そのときのエラーから, 読み込みをやめたヘッダーが担う役割を考え, 答えよ.

冒頭の行

`#include <drawlib.h>`

を削除して、コンパイルをした。その結果、下記のようなエラーが表示され、コンパイルに失敗した。



```
fujie@DESKTOP-KVRJVF: ~/work/0612$ gcc pingpong.c -ldrawlib -o pingpong
pingpong.c: In function 'main':
pingpong.c:19:29: error: 'DL_WIDTH' undeclared (first use in this function)
    int barlix = 70, bar2ix = DL_WIDTH - 70;
                              ^
pingpong.c:19:29: note: each undeclared identifier is reported only once
for each function it appears in
pingpong.c:20:31: error: 'DL_HEIGHT' undeclared (first use in this function)
    int barlx = barlix, barly = DL_HEIGHT / 2;
                               ^
pingpong.c:35:3: warning: implicit declaration of function 'dl_initialize' [-Wimplicit-function-declaration]
    dl_initialize(1, 0);
    ^
pingpong.c:38:12: warning: implicit declaration of function 'dl_get_event' [-Wimplicit-function-declaration]
    while (dl_get_event(&t, &k, &x, &y)) {
           ^
pingpong.c:39:16: error: 'DL_EVENT_KEY' undeclared (first use in this function)
    if (t == DL_EVENT_KEY) {
               ^
pingpong.c:114:5: warning: implicit declaration of function 'dl_stop'
```

エラーを抜粋すると、

pingpong.c:19:29: error: 'DL_WIDTH' undeclared (first use in this function)

と

pingpong.c:35:3: warning: implicit declaration of function 'dl_initialize' [-Wimplicit-function-declaration]

の 2 種類があった。前者は、`DL_WIDTH` が未宣言 (undeclared) であり、この関数で初めて使われて

いる (first used in this function) というエラー (error) である。後者は、`dl_initialize` という関数 (function) の暗黙の宣言 (implicit declaration) という警告 (warning) である。

いずれも、変数や関数の宣言に関するエラーや警告なので、`drawlib.h` を読み込むことで、これらを宣言せずに正常に使うことが可能になると考えられる。

- (2) 変数 `bar1□`, `bar2□` という変数が多数ある (□は任意の文字列)。これらのうち、それぞれ 1 つずつ取り上げ、変数宣言時に与えている初期値を変更し、プログラムを実行せよ。変更前と変更後の実行時の動作の違いから、取り上げた変数の意味を考え、答えよ。

19 行目にある

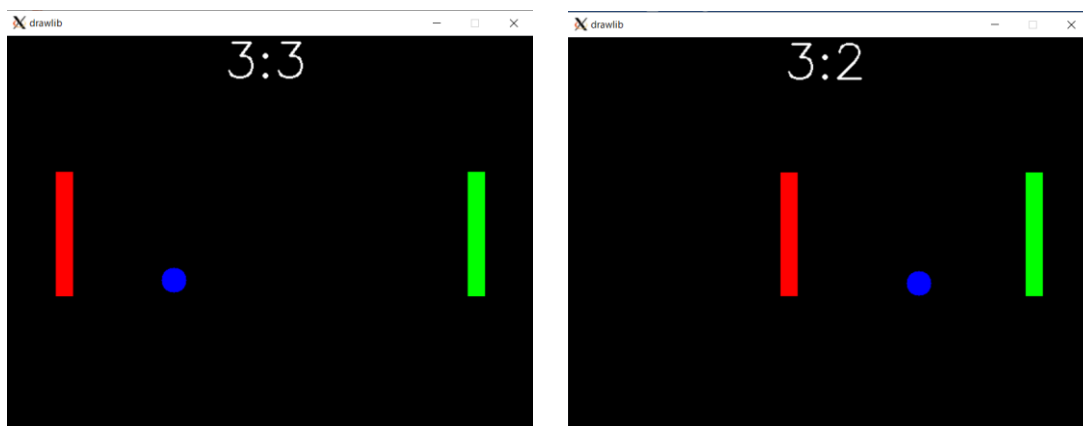
```
int bar1ix = 70, bar2ix = DL_WIDTH - 70;
```

という行を

```
int bar1ix = 270, bar2ix = DL_WIDTH - 70;
```

のように、`int` 型の変数 `bar1ix` の初期値を 70 から 270 に変更して実行した。

下の左が変更前、右が変更後の様子である。



変更前に比べて、左の赤いバーが右にずれている。このことから、`bar1ix` は赤いバーがどれだけ右にずれるか (x 座標がいくつ) ということを表す変数であると考えられる。

同様に、19 行目の

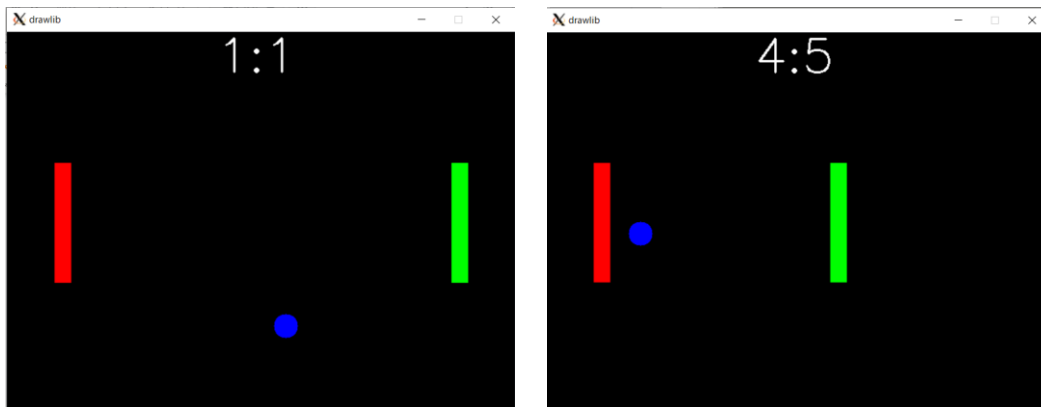
```
int bar1ix = 70, bar2ix = DL_WIDTH - 70;
```

という行を

```
int bar1ix = 70, bar2ix = DL_WIDTH - 270;
```

のように、`int` 型の変数 `bar2ix` の初期値にある 70 を 270 に変更して実行した。

下の左が変更前、右が変更後の様子である。



変更前に比べて、右の緑のバーが左にずれている。このことから、`bar2ix` の初期値に含まれている `70` は、バーがどれだけ左にずれるかを表すと考えられる。`DL_WIDTH` がウィンドウの幅 (x 座標の最大値を表す) と考えれば、`DL_WIDTH - 70` は、ウィンドウの左端から `70` 左にずらしていると考えられる。このことから `bar2ix` 自体は緑のバーの x 座標そのものを表しているとも考えられる。

- (3) 変数 `b□` (ただし `bar□`, `bar1□`, `bar2□` 以外) という変数がいくつかある。これらのうち、どれか 1 つを取り上げ、変数宣言時に与えている初期値を変更し、プログラムを実行せよ。変更前と変更後の実行時の動作の違いから、取り上げた変数の意味を考え、答えよ。

25 行目の

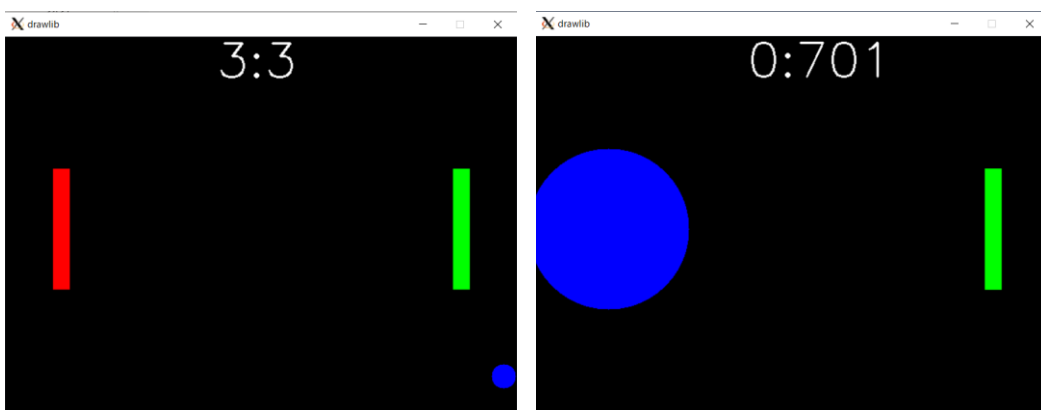
```
int br = 15;
```

を、

```
int br = 100;
```

と変更して実行した。

下の左が変更前、右が変更後の様子である。



変更前に比べて、変更後では青い円が大きい。このことから、変数 `br` は青い円の大きさを表す変数だと考えられる。

なお、その他に異なる点として、青い円が動かない、左の点数が上がり続ける、といった点も挙げられるが、これらは青い円の大きさが変わったことによる副作用だとかんがえられる(青い円の左端が、ウィンドウの左端に接した状態で開始されるため、即座に右の点数が上がるということが繰り返されている)。

※参考

例題 変数 `score1` の初期値を変更し、プログラムを実行した上でこの変数の意味を考え答えよ.

解答例

プログラムの 29 行目にある

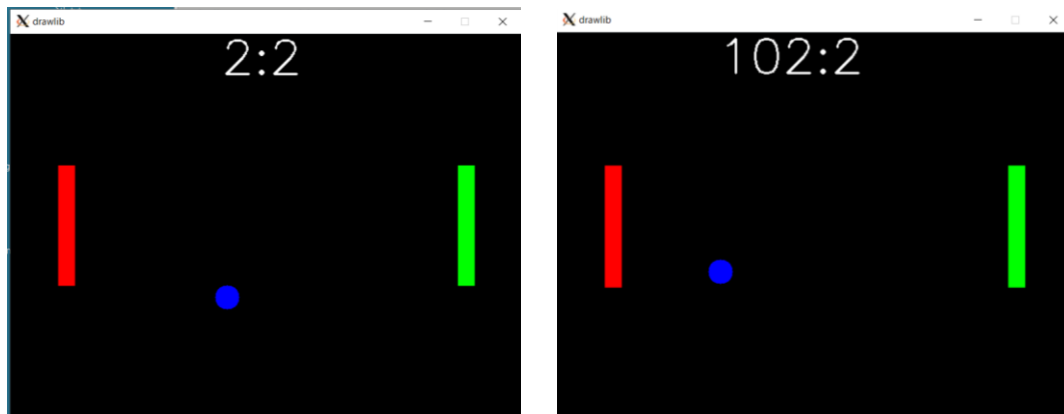
```
int score1 = 0, score2 = 0;
```

という変数宣言文を、

```
int score1 = 100, score2 = 0;
```

と変更して実行した。

下の図の、左側が変更前、右側が変更後のプログラム実行開始直後の様子である。ただし、開始後すぐにスクリーンショットがとれないため、初期値そのものではなく、すこし点数が動いた状態である。



変更前は左側のスコアが 0 から始まっているのに対して、変更後は 100 から始まっている。このことから、変数 `score1` は左側のスコアを表すものと考えられる。