

# プログラミング基礎 第8回 (練習問題解答例付き)

藤江 真也  
2021年6月11日

## 課題の再提出について

- 再提出の×切は manaba の提出画面には表示されませんので気をつけてください
- 6月11日現在, 第6回課題の再提出期間です (6月15日23時59分×切)
- 今後の再提出の×切については随時コースニュースでお知らせします

## 準 備

- ファイル(0611.tgz)をダウンロード

```
$ wget http://sites.fujielab.org/ip/files/0611.tgz
```

- ダウンロードしたファイルを展開

```
$ tar zxvf 0611.tgz
```

- 展開されたディレクトリに移動

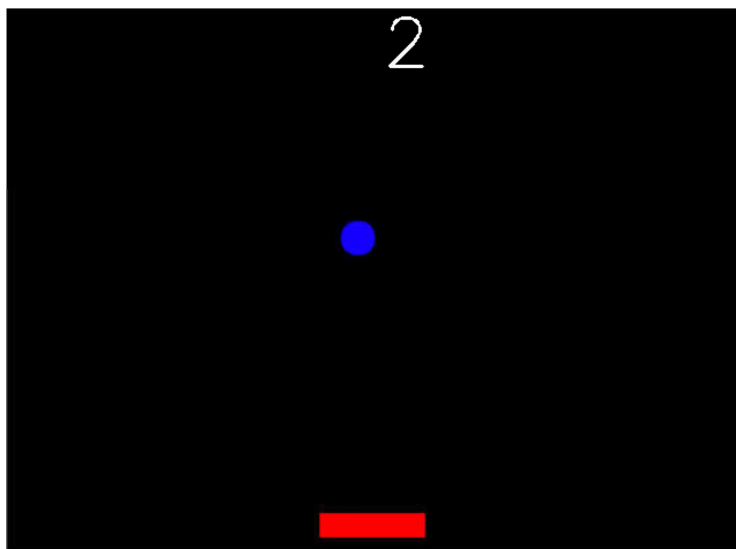
```
$ cd 0611
```

- 01ball.cなどのファイルがあることを確認

```
$ ls
```

## drawlibでゲームを作ろう

## 完成イメージ(lifting)



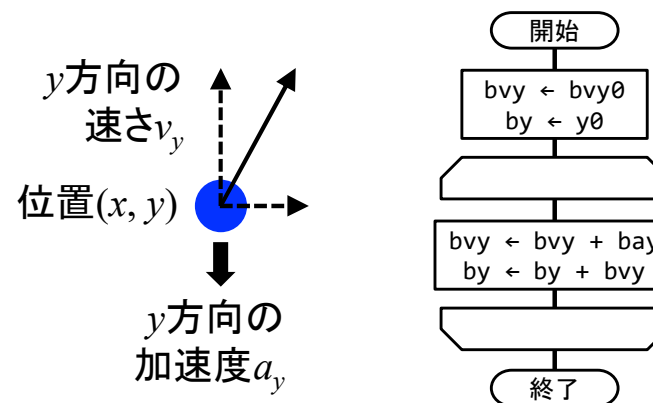
## ステップ① 放物線状に動くボール

## 01ball.c

- 実行をすると、以下の入力が入力がうながされる
  - bay: ボールの加速度
  - bvy0: ボールの初速
- bayは適当に、bvy0に-1を入れてみよう
  - 右上に向かってゆっくり動くはず
  - これを現実世界でボールを上を投げたかのように放物線を描かせたい

## ボールの動き

- プログラムの中の時間は1ステップごとに進む
  - 速度( $v_y$ ) → 1ステップあたりの位置(by)の変化量(bvy)
  - 加速度( $a_y$ ) → 1ステップあたりの速度(bvy)の変化量(bay)



yは下向きが正  
であることに注意

## 練習問題 ①

- (1) 01ball.c を変更して, ボールの速度 (bvy) が毎ステップ加速度 (bay) にしたがって更新されるようにしよう
- (2) 01ball.c を実行して, 適度にぴよんぴよん動くボールの初速 (bvy0), 加速度 (bay) を見極めよう
- (3) ボールは画面の右で跳ね返るが, 左側では跳ね返らず消えてしまう. 左側でも跳ね返るようにしよう

## 練習問題 ① 解答例

- (1) 01ball.c を変更して, ボールの速度 (bvy) が毎ステップ加速度 (bay) にしたがって更新されるようにしよう

```
/* --- メインループ --- */
while(1) {
    /* ボールの移動処理 */
    bvy += bay;
    by += bvy; /* Y座標にY方向の速度を加える */
    bx += bvx; /* X座標にX方向の速度を加える */
}
```

## 練習問題 ① 解答例

- (2) 01ball.c を実行して, 適度にぴよんぴよん動くボールの初速 (bvy0), 加速度 (bay) を見極めよう

- bvy0が-10の場合, bayは0.12くらいで画面いっぱい飛び跳ねます

- 【参考】途中の計算は省略しますが, 画面いっぱいに飛び跳ねさせたいときの bay と bvy0 の関係式は以下の通りになります.

$$\text{bay} = (\text{bvy0} * \text{bvy0}) / (800 - \text{bvy0});$$

## 練習問題 ① 解答例

- (3) ボールは画面の右で跳ね返るが, 左側では跳ね返らず消えてしまう. 左側でも跳ね返るようにしよう

```
by += bvy; /* Y座標にY方向の速度を加える */
bx += bvx; /* X座標にX方向の速度を加える */

/* ボールの境界処理(右) */
if (bx + br > DL_WIDTH) {
    bvx *= -1.0; /* 速度の正負を反転させる */
    bx = DL_WIDTH - br; /* 位置の微調整 */
}

/* ボールの境界処理(左) */
if (bx - br < 0) {
    bvx *= -1.0;
    bx = 0 + br;
}

/* 下限判定 */
if (bvy + br > bary) {
```

## ステップ② 操作可能なラケット(バー)

## 02bar.c

- 実行をすると, 赤いバーが表示される
- Fキーで左に, Jキーで右に移動する
  - 画面の端に来て外に出てしまう

## 練習問題②

- (1) 02bar.c を変更して, バーが画面外に出ないようにしよう
- (2) 02bar.c を変更して, キーが押された時にバーが移動する量を変えてみよう. 元は一度にバーの幅だけ移動する.
- (3) 【発展問題】速度制御(バーにも速度を設定し, キーを押すことでバーの移動速度が変わる)にしてみよう

## 練習問題② 解答

- (1) 02bar.c を変更して, バーが画面外に出ないようにしよう

```
/* Fキーが押されたらバーのX座標 */
if (k == 'f') {
    barx -= barw / 2;
}
}

/* 右側の処理 */
if (barx + barw > DL_WIDTH) {
    barx = DL_WIDTH - barw;
}
/* 左側の処理 */
if (barx < 0) {
    barx = 0;
}

/* 描画処理 */
dl_ston();
```

## 練習問題② 解答

- (2) 02bar.c を変更して、キーが押された時にバーが移動する量を変えてみよう。元は一度にバーの幅だけ移動する。

```
/* 入力キーの処理 */
while (dl_get_event(&t, &k, &x, &y)) {
    if (t == DL_EVENT_KEY) {
        /* Jキーが押されたらバーのX座標を右にずらす */
        if (k == 'j') {
            barx += barw / 2;
        }
        /* Fキーが押されたらバーのX座標を左にずらす */
        if (k == 'f') {
            barx -= barw / 2;
        }
    }
}
```

右辺の数値が一度に左右に動く量なのでこれを適当に変える。  
この解答は元の2分の1にしている。

## ステップ③ スコアの表示

## 03score.c

- 実行はできるが何も表示されない

## 練習問題③

- (1) 03score.c を変更して、正しくスコアが表示されるようにしよう
  - dl\_text関数を利用する
  - pingpong.cを参考にするとよい
- (2) スコアが 500 点を超えたらループを抜けるようにしてみよう
- (3) 【発展問題】表示されるスコアの色を、スコアの値とともに変化するようにしてみよう

## 練習問題 ③ 解答

- (1) 03score.c を変更して, 正しくスコアが表示されるようにしよう

```
/* 描画処理 */  
dl_stop();  
dl_clear(DL_C("black"));  
sprintf(sscore, "%5d", score);  
dl_text(sscore, sx, sy, 2.0, DL_C("white"), 2);  
dl_resume();  
dl_wait(wait_time);
```

## 練習問題 ③ 解答

- (2) スコアが 500 点を超えたらループを抜けるようにしてみよう

```
while(1) {  
    score += 1;  
  
    if (score > 500) {  
        break;  
    }  
  
    /* 描画処理 */  
    dl_stop();
```

## ステップ④ 開始画面と終了画面

## 04start.c

- 実行すると PUSH 'F' to start と表示される
- が, 何を押しても画面は進まない

## 練習問題④

- (1) 04start.c を変更し, F キーが押されたときに正しく開始画面が終了するようにしよう
- (2) ループ終了後(スコアが500を超えた後)に終了画面が出るが, 一瞬で消えてしまう.  
ここでもFキーが押されるまで待つようにしよう

## 練習問題④ 解答

- (1) 04start.c を変更し, F キーが押されたときに正しく開始画面が終了するようにしよう

```
dl_resume();

/* --- Fキーが押されるまで待機 --- */
start = 0;
while(start == 0) {
    /* Fキーが押されたら start を 1 にする */
    /* 入力キーの処理 */
    while (dl_get_event(&t, &k, &x, &y)) {
        if (t == DL_EVENT_KEY) {
            if (k == 'f') {
                start = 1;
            }
        }
    }
}

dl_wait(wait_time);
}
```

## 練習問題④ 解答

- (2) ループ終了後(スコアが500を超えた後)に終了画面が出るが, 一瞬で消えてしまう.  
ここでもFキーが押されるまで待つようにしよう

```
/* --- Fキーが押されるまで待機 --- */
/* 練習問題 開始画面と同じことをする */
start = 0;
while(start == 0) {
    /* Fキーが押されたら start を 1 にする */
    /* 入力キーの処理 */
    while (dl_get_event(&t, &k, &x, &y)) {
        if (t == DL_EVENT_KEY) {
            if (k == 'f') {
                start = 1;
            }
        }
    }
}
dl_wait(wait_time);
}
```

## ステップ⑤ 統合作業

## ステップ ⑤

- 01ball.c ~ 04start.c を組み合わせることで、ゲームらしきものが作れる
  - まずは、開始画面 → 操作可能なバー、ボール、スコアの表示の繰り返し → 終了画面 の流れを作る
- 足りないのは...
  - バーとボールの当たり判定と当たった時の処理
  - ゲームオーバーの判定処理

## 第8回 課題

## 第8回 課題

- ステップ⑤を完了させて、リフティングゲームを完成させてください
- 工夫した点があったら、ソースコードの冒頭の部分にコメントで説明を加えてください
- ソースコードを提出してください
  - ファイル名は 番号.c にしてください  
(例: 21C1981.c)
- 〆切は 6月15日23時59分です

## 補足



## drawlibを使ったゲームプログラムの基本構造

```
#include <drawlib.h>
```

```
int main(void) {
```

変数宣言, 初期化など

```
dl_initialize(1.0);
```

```
while(1) {
```

入力処理(必要な場合)

動き, 当たり, スコアなどの処理

```
dl_stop();
```

描画処理

```
dl_resume();
```

```
dl_wait(0.01);
```

```
return 0;
```

```
}
```

メインループ  
(この部分を何度も繰り返す)

0.01秒程度待つ  
ループを遅く回したければこの  
数字を大きくする

## 入力処理

```
while (dl_get_event(&t, &k, &x, &y)) {
```

```
if (t == DL_EVENT_KEY) {
```

操作に応じた  
変数の値の変更

```
}
```

操作情報が取り出せたら真, 無ければ偽

t には操作の種類が入っているので,  
これで場合分けをする.  
DL\_EVENT\_KEY はキーボードの入力で  
あることを指す.

- DL\_EVENT\_KEY ... いずれかのキーが押下された
- DL\_EVENT\_L\_DOWN ... マウスの左ボタンが押された
- DL\_EVENT\_L\_UP ... マウスの左ボタンが離された
- DL\_EVENT\_R\_DOWN ... マウスの右ボタンが押された
- DL\_EVENT\_R\_UP ... マウスの右ボタンが離された

## キーボード操作の場合

```
while (dl_get_event(&t, &k, &x, &y)) {
```

```
if (t == DL_EVENT_KEY) {
```

```
if (k == 'a') {
```

Aキーが押されたときの処理

```
}
```

```
}
```

```
}
```

k には押されたキーの文字が入って  
いる.  
'a' や 'b' などと比較し, 等しい  
場合にその文字のキーが押された  
ときにしたい処理をする.

## マウスのボタン操作の場合

```
while (dl_get_event(&t, &k, &x, &y)) {
```

```
if (t == DL_EVENT_L_DOWN) {
```

x と y に入っている押された位  
置を使った処理

```
}
```

```
}
```