

基礎プログラミングおよび演習 レポート # 15

2120029, 氏名 政野玄空 (ペア:i2020002, 伊東剛・i2120003, 石川ひろき)

2022年2月13日

1 構想・計画・設計

まずなにを作りたいかを相談してボールが転がる表現が例題から簡易に作れると考え, ボールを使うものの中からボーリングの動画を作ることにした. 別案ではビリヤード等もあったがボールのぶつかったあとの演算に自信がなかったので却下した.

題が決まったのでこれを達成するために例題の中では足りないものを話し合った. ボーリングのピンが足りないという結論になり, img.c にボーリングのピンの実装が必要という結論になった. 石川さんがボーリングのピンの描写の実装を担当することとなり, 伊藤さんは img.h の実装, 自分は main.c の実装と分担することになった. まず石川さんが実装をし, その後に伊藤さん, 次に自分という順番で実装することに決めた.

2 プログラムコード

```
img.h
#define WIDTH 300
#define HEIGHT 200
struct color { unsigned char r, g, b; };
void img_clear(void);
void img_write(void);
void img_putpixel(struct color c, int x, int y);
void img_fillcircle(struct color c, double x, double y, double r);
void img_fillpin(struct color c, double x, double y, double r);
```

```
img.c
#include <stdio.h>
#include <stdlib.h>
```

```

#include "img.h"

static unsigned char buf[HEIGHT][WIDTH][3];
static int filecnt = 0;
static char fname[100];

void img_clear(void) {
    int i, j;
    for(j = 0; j < HEIGHT; ++j) {
        for(i = 0; i < WIDTH; ++i) {
            buf[j][i][0] = buf[j][i][1] = buf[j][i][2] = 255;
        }
    }
}

void img_write(void) {
    sprintf(fname, "img%04d.ppm", ++filecnt);
    FILE *f = fopen(fname, "wb");
    if(f == NULL) { fprintf(stderr, "can't open %s\n", fname); exit(1); }
    fprintf(f, "P6\n%d %d\n255\n", WIDTH, HEIGHT);
    fwrite(buf, sizeof(buf), 1, f);
    fclose(f);
}

void img_putpixel(struct color c, int x, int y) {
    if(x < 0 || x >= WIDTH || y < 0 || y >= HEIGHT) { return; }
    buf[HEIGHT-y-1][x][0] = c.r;
    buf[HEIGHT-y-1][x][1] = c.g;
    buf[HEIGHT-y-1][x][2] = c.b;
}

void img_fillcircle(struct color c, double x, double y, double r) {
    int imin = (int)(x - r - 1), imax = (int)(x + r + 1);
    int jmin = (int)(y - r - 1), jmax = (int)(y + r + 1);
    int i, j;
    for(j = jmin; j <= jmax; ++j) {
        for(i = imin; i <= imax; ++i) {
            if((x-i)*(x-i) + (y-j)*(y-j) <= r*r) { img_putpixel(c, i, j); }
        }
    }
}

```

```

    }
}

void img_fillpin(struct color c, double x, double y, double r) {
    double xr1 = r; double yr1 = r*2.2; double xr2 = r*6/10; double yr2 = r*9/10;
    double y2 = y + yr2 + yr1;
    int imin1 = (int)(x - xr1 - 1), imax1 = (int)(x + xr1 + 1);
    int jmin1 = (int)(y - yr1 - 1), jmax1 = (int)(y + yr1 + 1);
    int imin2 = (int)(x - xr2 - 1), imax2 = (int)(x + xr2 + 1);
    int jmin2 = (int)(y2 - yr2 - 1), jmax2 = (int)(y2 + yr2 + 1);
    int i, j;
    for(j = jmin1; j <= jmax1; ++j) {
        for(i = imin1; i <= imax1; ++i) {
            if((x-i)*(x-i)/(xr1*xr1) + (y-j)*(y-j)/(yr1*yr1) <= 1) { img_putpixel(c, i, j); }
        }
    }
    for(j = jmin2; j <= jmax2; ++j) {
        for(i = imin2; i <= imax2; ++i) {
            if((x-i)*(x-i)/(xr2*xr2) + (y2-j)*(y2-j)/(yr2*yr2) <= 1) { img_putpixel(c, i, j); }
        }
    }
}

```

animate.c

```
#include "img.h"
```

```

int main(void) {
    struct color pin1 = { 219, 0, 32 };
    struct color pin2 = { 255, 15, 51 };
    struct color pin3 = { 255, 71, 99 };
    struct color pin4 = { 255, 117, 137 };
    struct color pin5 = { 255, 168, 181 };
    struct color ball = { 0, 0, 0 };
    int i;

    for(i = 0; i < 40; ++i) {
        img_clear();
        if (i < 28) {

```

```

//4 列目
img_fillpin(pin4,120,135,9);
img_fillpin(pin4,140,135,9);
img_fillpin(pin4,160,135,9);
img_fillpin(pin4,180,135,9);
}
if (i < 24) {
//3 列目
img_fillpin(pin3,130,130,10);
img_fillpin(pin3,150,130,10);
img_fillpin(pin3,170,130,10);
}
if (i < 20){
//2 列目
img_fillpin(pin2,140,125,11);
img_fillpin(pin2,160,125,11);
}
if (i < 16){
//1 列目
img_fillpin(pin1,150,120,12);
}
img_fillcircle(ball, 150, 30+i*5, 40-i);
img_write();
}

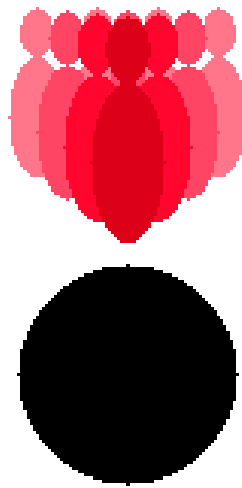
return 0;
}

```

3 プログラムの説明

img_clear でイメージの消去,img_write でイメージの描写,img_putpixel は今回使っていない.img_fillcircle で円の描写,img_fillpin でボーリングのピンの描写をしている.

4 生成された動画の説明



ボーリングの様子. ボールが転がってすべてのピンを倒している.

5 開発過程の説明

2022年2月5日に石川さんの作業が完了したという報告があったので自分も実装の準備をし始めた.`img.h`は待たなくてもできるので2022年2月6日に完成させてそのまま動作確認をした. ボーリングのピンを同じ色にして生成するとピンに見えないという問題があったのでグラデーションをつけることにした. もしくは`img.c`に手をいれて輪郭を生成するという方法もあったが, あえて分担通りにした. まずボールの転がる様子だけを実装し, 次にピンを後ろから順番に追加していき, 最後に倒れる表現として前から順番に描写を消していった. 完成したものを共有し自分の作業は終わった.

6 考察

ピンの倒れる描写は更に工夫できる余地があったと思う. 実際に倒れる様子をコマ数をふやしてピンがランダムで倒れる様子を描写できればさらに現実に近いものになっただろう.

また今の動画ではピンとボールの様子が噛み合わないところがあるのでボールの様子もピンに当たってからの減速を表現すればよかったと思った.

7 アンケート

7.1 Q1：うまく分担して課題プログラムを開発できましたか。

内容が少なかったので分担するのが大変だった。

7.2 Q2:複数で分担する際に注意すべきことは何だと思いましたか。

コミュニケーションが重要だと思った。

7.3 Q3：ここまでの科目全体を通して、学べたこと、学びたかったけど学べなかったことは何ですか。その他感想や、この科目の今後改善した方がよいこと、今後も維持したことがよいことの指摘もどうぞ。

仕事でもたまに使うことがあったが ImageMagick がとても便利だと学んだ。