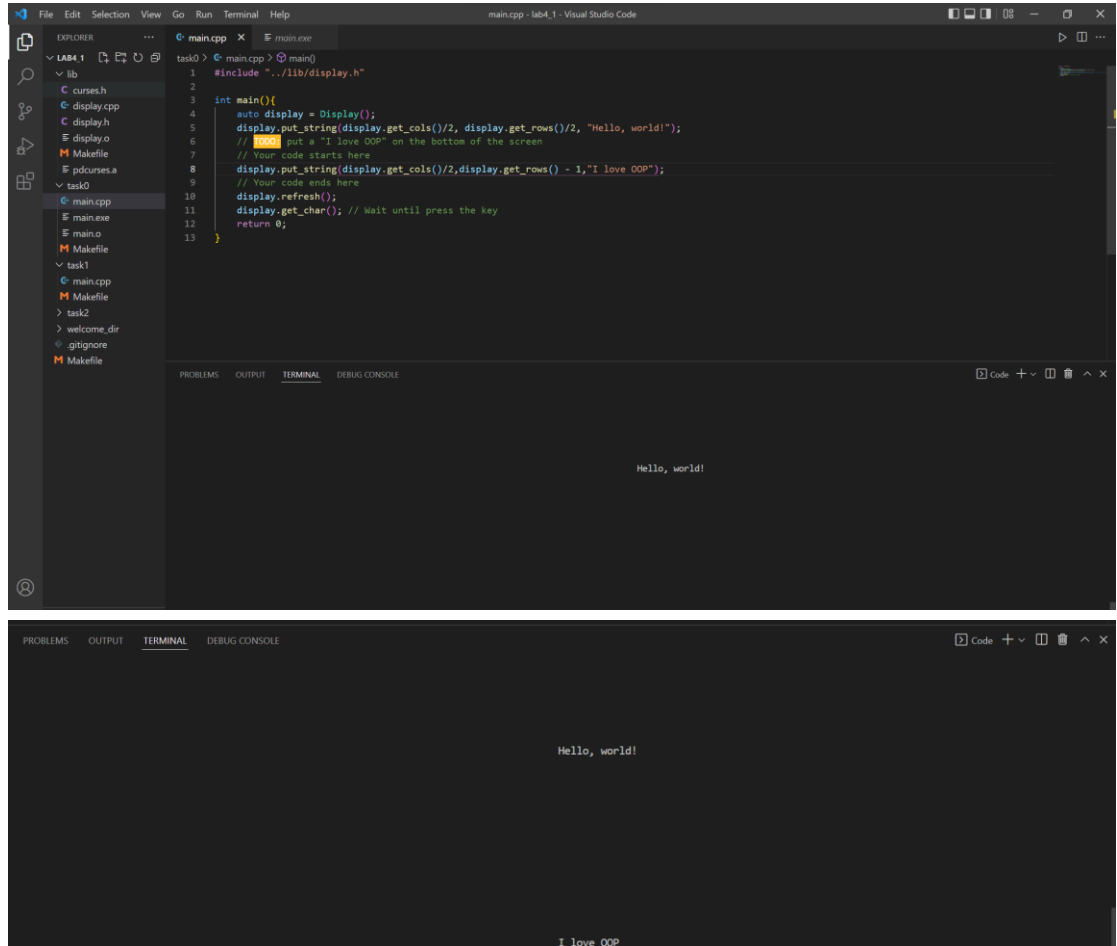


敖伟智 21307130326

实验报告(Lab4_1):

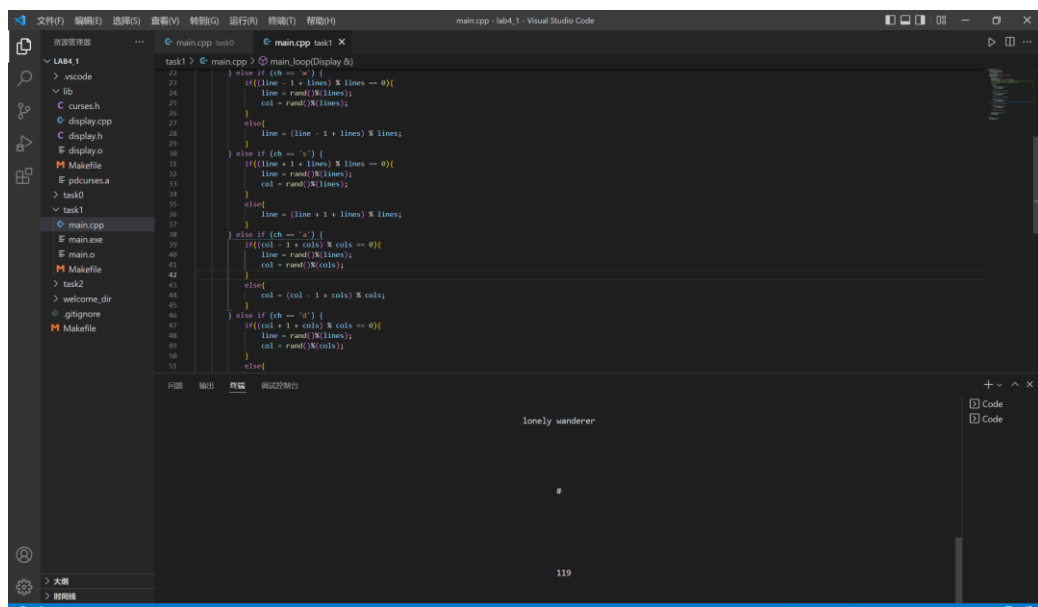
Task0:

1) 运行截图



Task1:

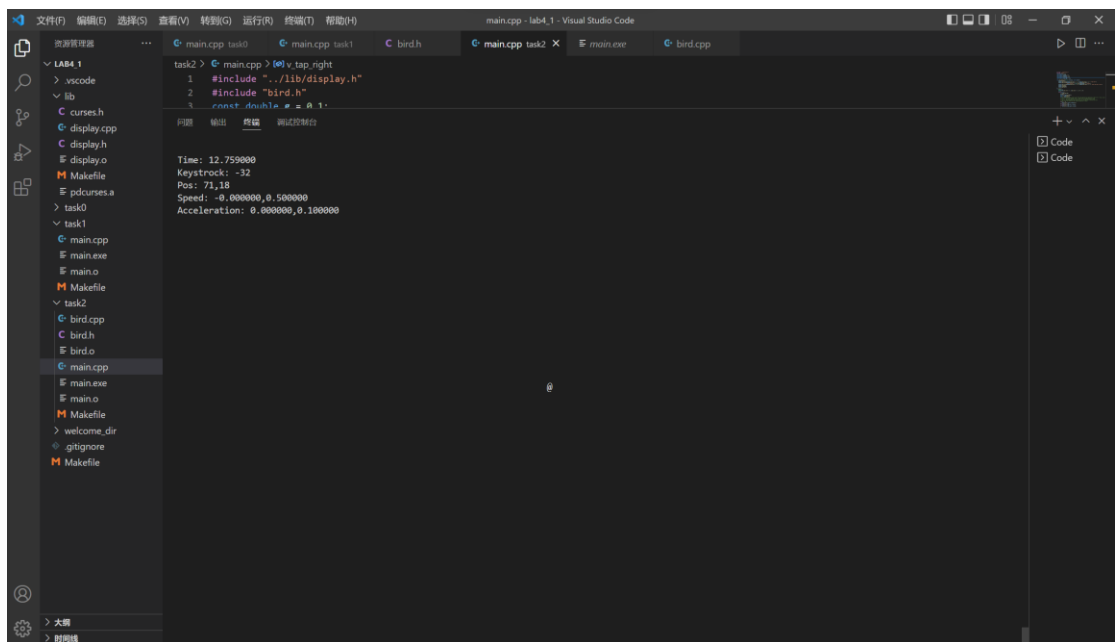
1) 运行截图





Task2:

1) 运行截图



2) 运行逻辑

```
Display d;
Bird bird(d.get_cols() / 2, d.get_rows() / 2, 0, 0, 0, 0);
int c;
```

1.

先创建一个幕布作为图形界面的区域，创建一个鸟的对象 bird 和一个记录按键输入的 int 类型变量 c。

2.

```
while (true) {
```

(a)进入循环，每个循环就是游戏中的一帧，在每一个循环（每一帧）中会在幕布上打印

```
d.put_string(0, 0, "Time: " + std::to_string(d.time() / 1000.0));
d.put_string(0, 1, "Keystroke: " + std::to_string(c));
d.put_string(0, 2, "Pos: " + std::to_string(pos.first) + ", " + std::to_string(pos.second));
d.put_string(0, 3, "Speed: " + std::to_string(v.first) + ", " + std::to_string(v.second));
d.put_string(0, 4, "Acceleration: " + std::to_string(a.first) + ", " + std::to_string(a.second));

d.put_string(pos.first, pos.second, "@");
```

作为游戏的视觉画面

其中 Time,Keystroke,Pos,Speed,Acceleration 会打印在固定位置：左上角

而鸟@会打印在 Bird 中储存的位置(x,y)，在连续的打印中(约每秒 20 帧)会形成连续的轨迹。

(b)在循环中的另一部分

```
// ... else { continue; }
if(c == ' '){
    bird.set_v(v.first,v_tap_up*1.0);
}else if(c == 'z'){
    bird.set_v(v_tap_left,v.second);
}else if(c == 'x'){
    bird.set_v(v_tap_right,v.second);
}
if(v.first > 0){
    bird.set_a(a_tap_right,g);
    if(v.first + a_tap_right < 0){
        bird.set_a(0.0,g);
    }
}else if(v.first < 0){
    bird.set_a(a_tap_left,g);
    if(v.first + a_tap_left > 0){
        bird.set_a(0.0,g);
    }
}else{
    bird.set_a(0.0,g);
}
```

用于通过键盘输入的指令来更新鸟的加速度和速度, 作为下一帧鸟的位置的计算依据

(c)游戏结束的判断

```
if (pos.second <= 0 || pos.second >= d.get_rows() - 1 || pos.first <= 0 || pos.first >= d.get_cols() - 1) {
    show_game_over(d, d.time() / 1000.0);
    break;
}
```

用鸟的位置与幕布的边框位置进行比较, 如果鸟的位置超出了幕布的范围即跳出循环并调用 show_game_over 函数

```
void show_game_over(Display &display, double second) {
    display.clear();
    display.put_string(display.get_cols() / 2 - 7, display.get_rows() / 2 - 1, "Game Over");
    display.put_string(display.get_cols() / 2 - 14, display.get_rows() / 2,
        "Bird survives " + std::to_string(second) + " seconds");
    display.put_string(display.get_cols() / 2 - 13, display.get_rows() / 2 + 1, "Press any key to exit");
    display.refresh();
    display.sleep(200);
    display.get_char();
}
```

显示游戏结束时的相关数据, 并 break 跳出循环, 结束游戏。

(d)最后调用 tick 函数

```
void Bird::tick(){
    vx = vx + ax;
    vy = vy + ay;
    double delta_x = vx + 0.5*ax;
    double delta_y = vy + 0.5*ay;
    x = x + delta_x;
    y = y + delta_y;
}
```

用于更新鸟的位置(x,y)信息 (利用在循环中更新的加速度与速度信息)