



产品智能化设计与 制作-红温乱斗

小组成员：杨亚澄 陈继威 梁佳栋 魏雨 曾建涛

指导老师：王凯

日期：2025.1.8

更多详情：

https://github.com/fredcoffe/emotion_game



目录 :

01

前期构思

02

设计过程

03

制作过程

04

成果展示



01

前期构思



小组讨论

我们四位组员分别提出自己的想法，由杨亚澄组长带领我们一起分别对四个人的想法进行分析、判断，看哪个想法更有趣、新颖、艺术表现力强、跟课程结合起来好实现，最终得到最终的选题。

1.曾建涛：制作一个欠揍玩偶，通过arduino控制受击反应。

01

2.陈继威：编程猫冒险岛中，玩家需运用arduino编程技巧解谜，引领猫咪探索神秘岛屿。

02

4.魏雨：制作一个地铁里两个小人互相战斗的场景，通过感受玩家的操作情绪，arduino反馈在场景中展示各种效果。 **(最终确定方案)**

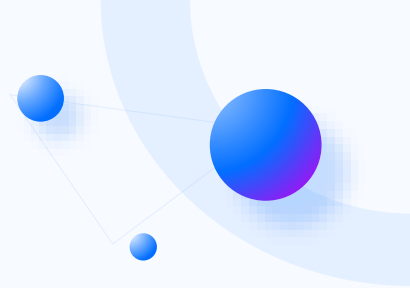
04

3.梁佳栋：构建一个迷宫，通过arduino控制机关，玩家控制小人走出迷宫。

03

头脑风暴

设计思路



我们以人们打游戏时会有激动的情绪为思路，想制作一款具有情绪交互方式的游戏。制作的游戏特点是能捕捉玩家的情绪，玩家玩游戏时越激动，则打出的伤害就越高。

美术风格：

结合组内成员的涂鸦特长，我们选择了“纽约地铁涂鸦”这一带有感性、随意的文化背景，以凸显出游戏注重感性情绪的特点。

产品制作：

我们选择木板与彩卡纸、油漆笔的方式进行制作，用到手绘、激光切割等工艺，可快速制作出原型。

技术思路：

在首轮头脑风暴后，我们最终使用薄膜压力传感器进行情绪捕捉，玩家握住贴着压力传感器的操纵杆，当玩家情绪激动时，会不自觉握紧，此时压力传感器便能采集玩家的情绪变化。

02

设计过程



地铁环境推敲过程

风格上选取涂鸦元素。



参考图片

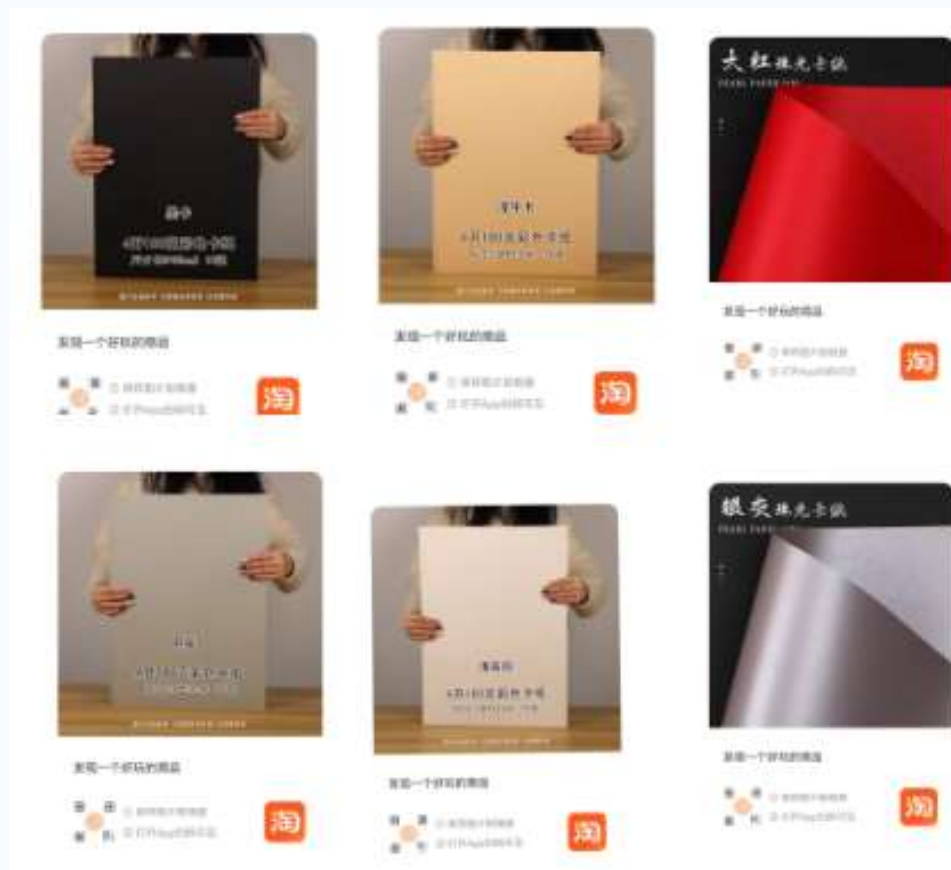


procreate手绘效果图

地铁环境材料选择

表面材料选择：有色卡纸

结构材料选择：木板



地铁场景尺寸图

地铁窗外背景参考

背景参考左图效果，通过车窗透出后面风景，增加层次效果。



窗外背景效果参考



最终背景参考图片

场景搭建推敲过程

场景搭建效果图，由前到后做了多个层次。



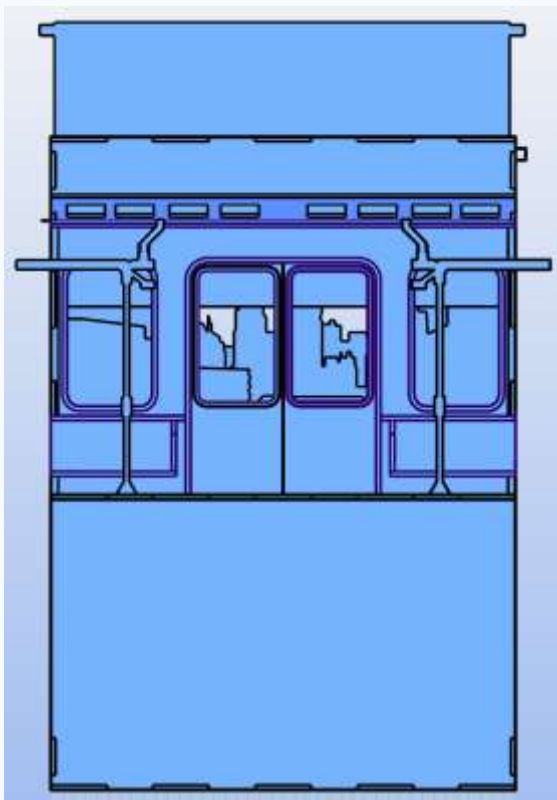
参考图片



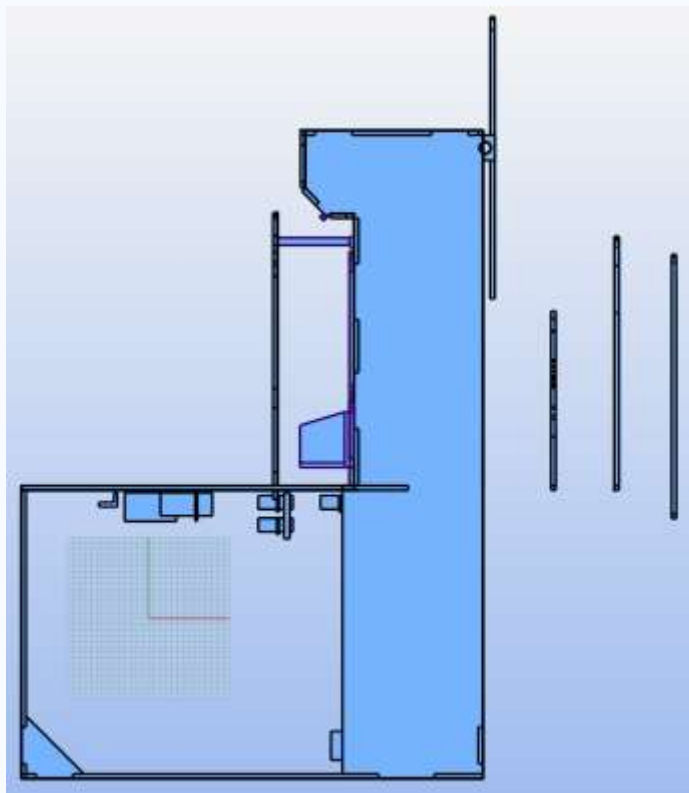
procreate手绘场景效果图

建模

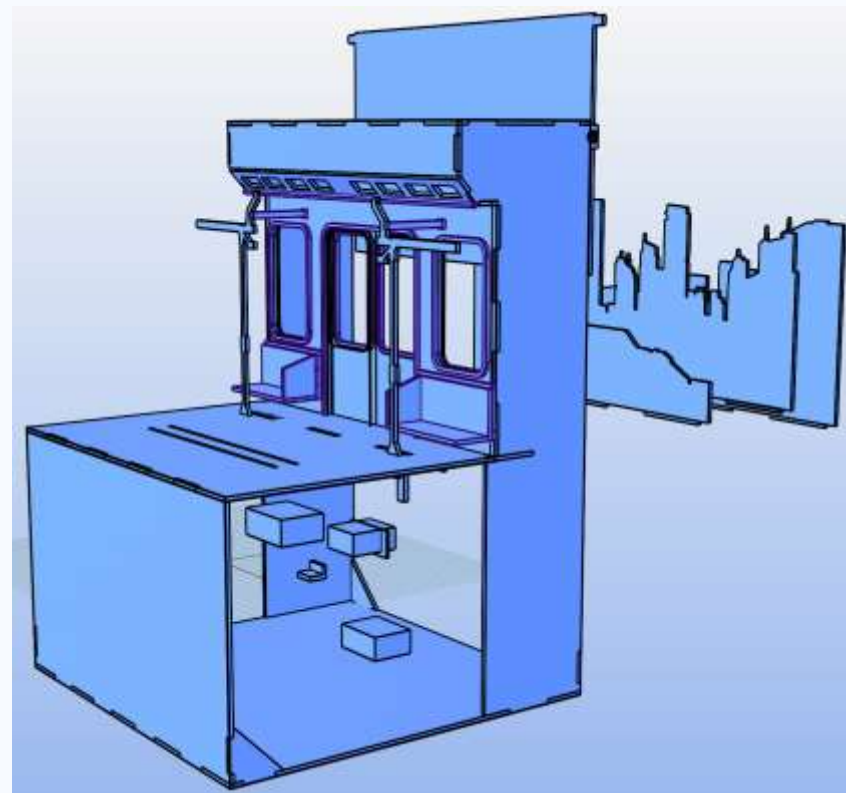
整个环境建模图



前视图



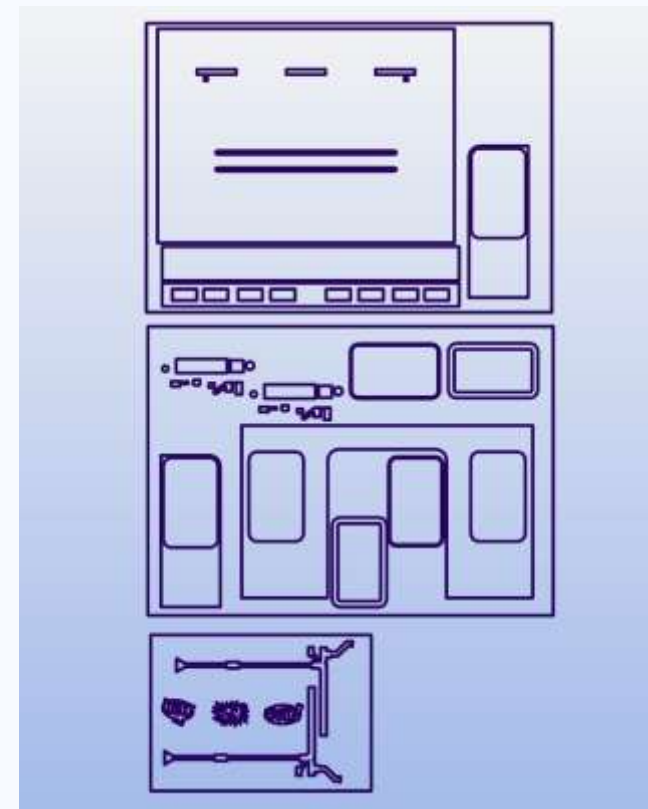
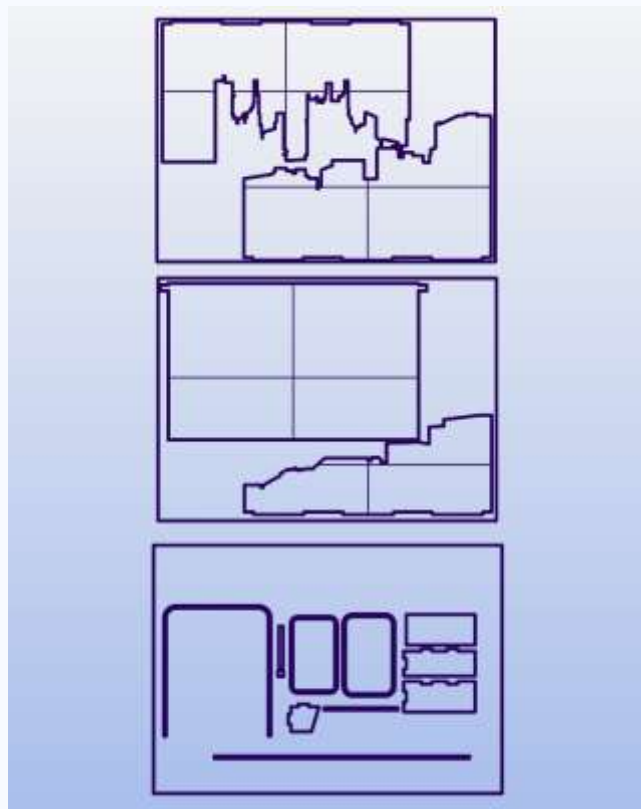
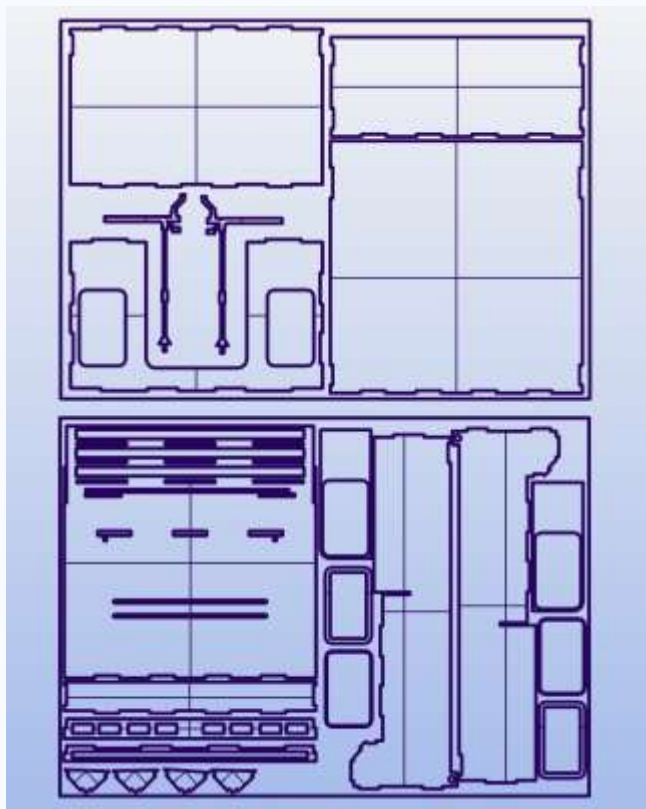
侧视图



透视图

激光切割图

用于激光切割机将各部分切割下来



其他

购买素材及贴纸设计



竹节人素材



自己设计与制作的贴纸素材

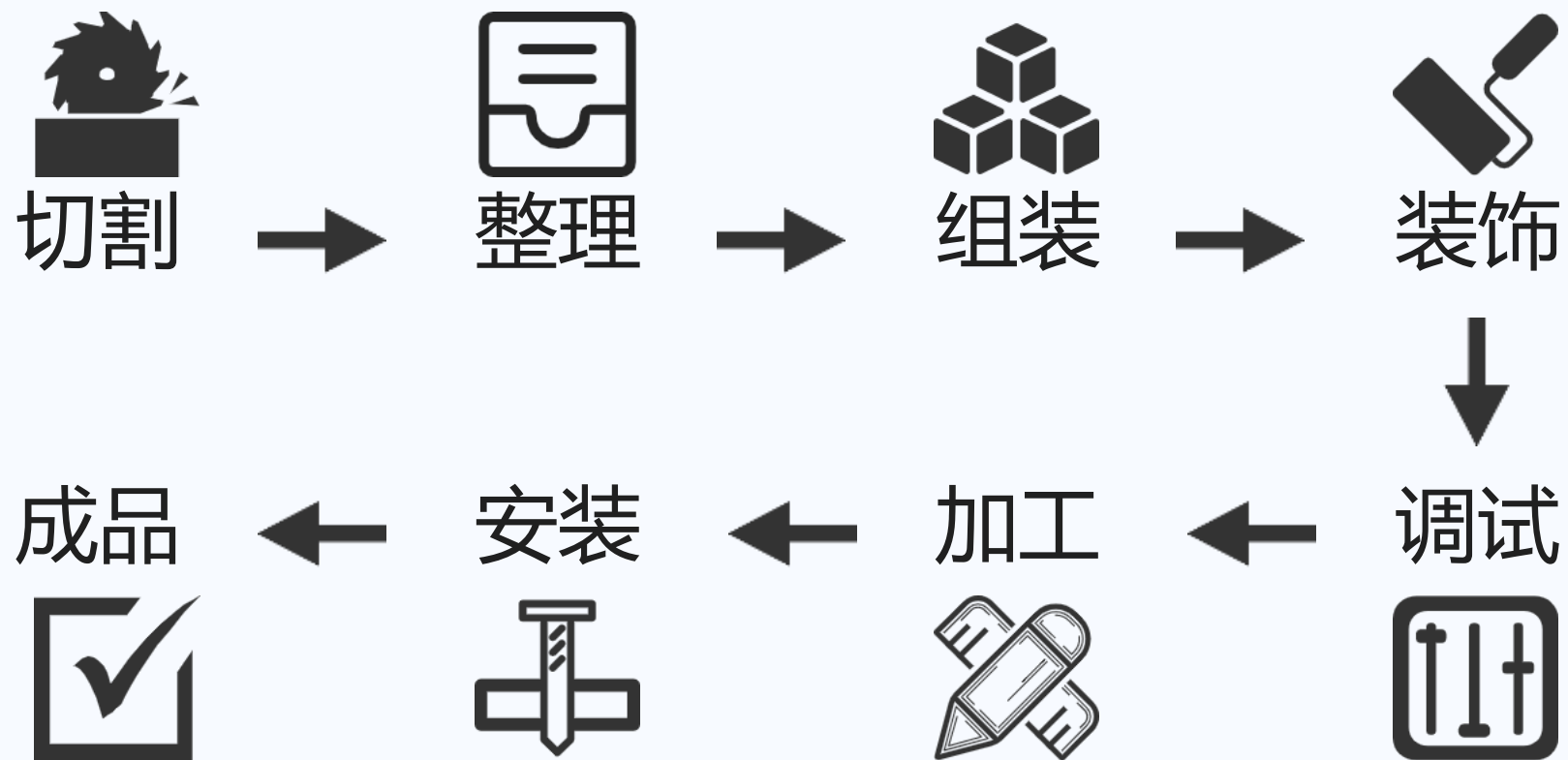


03

制作流程

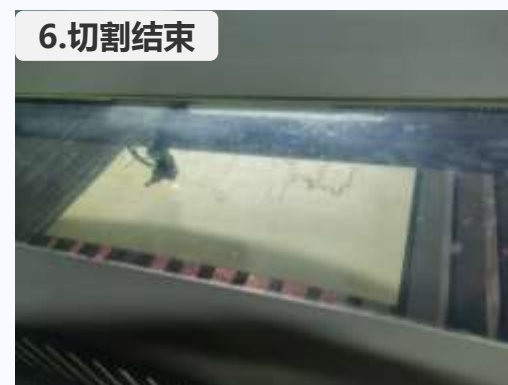


流程



切割

将图纸导入激光切割机对木板、卡纸进行切割



切割过程



[点击观看视频](#)

整理

整理并修正切割好的物体



拿取切割出来的木板、卡纸



对木板、卡纸进行整理修正



整理好的木板及卡纸（部分图片）

组装

先将木板拼装在一起然后贴上卡纸。



组装中



组装完成



初步贴纸

装饰

用贴纸装饰及马克笔涂绘，营造涂鸦元素。



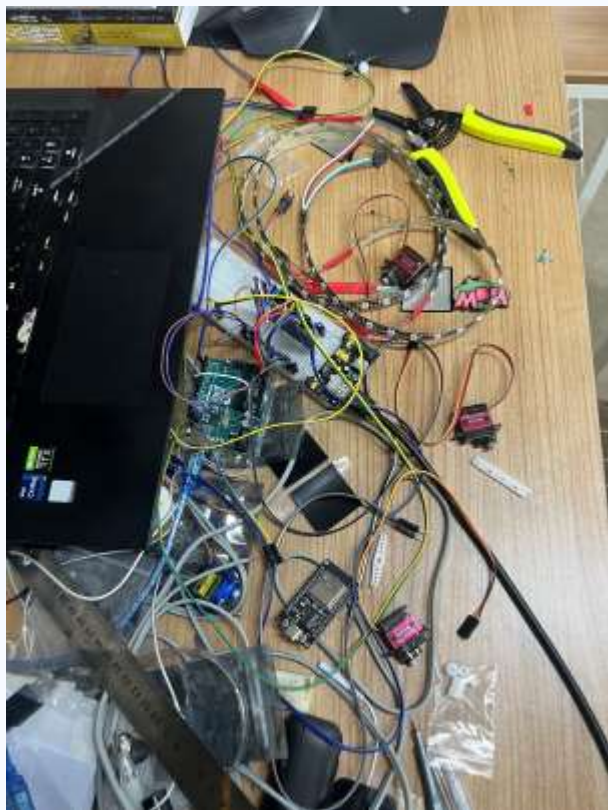
贴纸装饰马克笔涂绘



装饰完成效果

调试

整理所使用到的元器件并调试



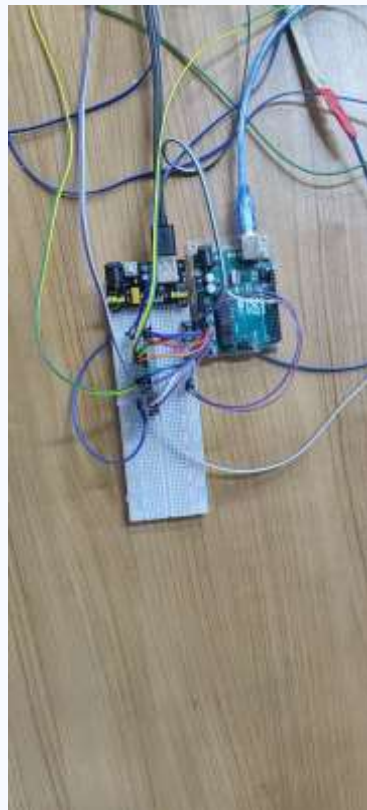
使用到的元器件



调试元器件

加工

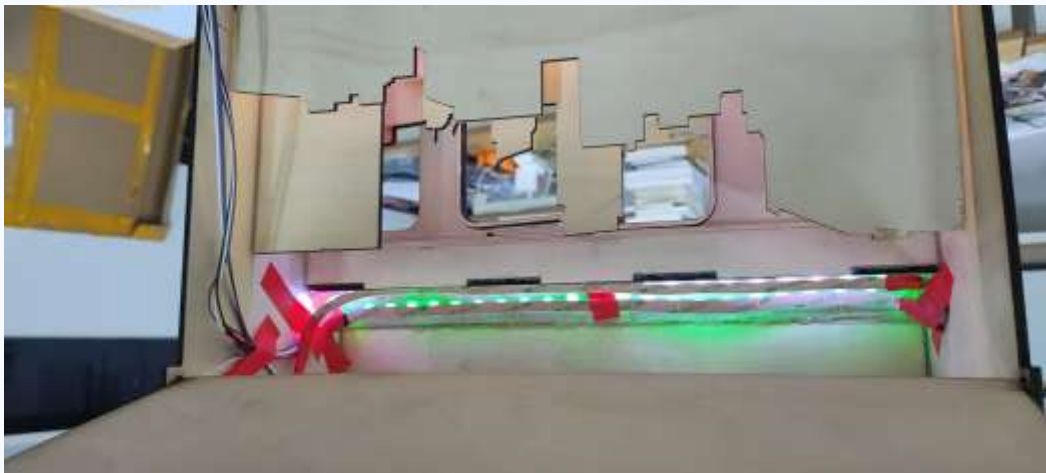
加工完成的元器件、电路、马达、压力传感器等



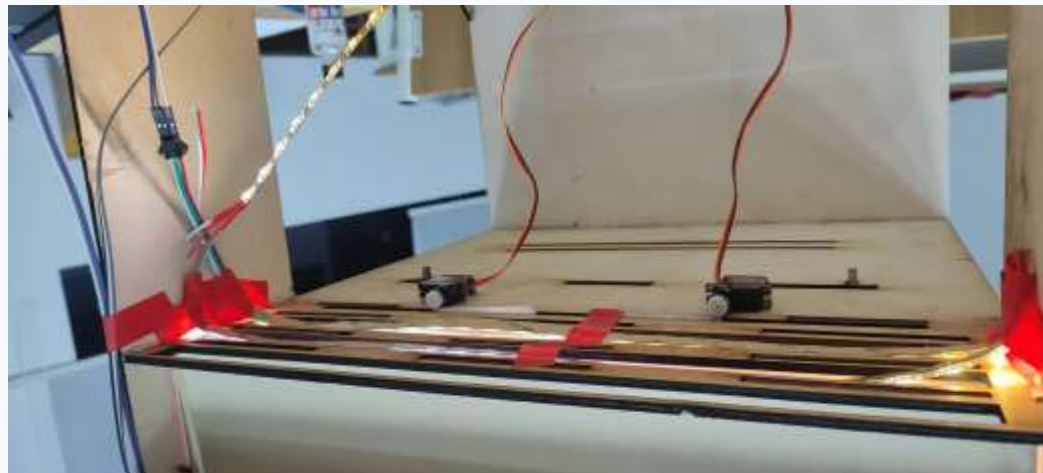
用到的各部分元器件

安装

安装电路、灯光、齿轮、马达等元器件



安装电路、灯光



安装齿轮、马达

04

成果展示



成品展示

场景展示



成品展示

战斗展示



成品展示

技术展示





附

代码解释

更多详情：

https://github.com/fredcoffe/emotion_game

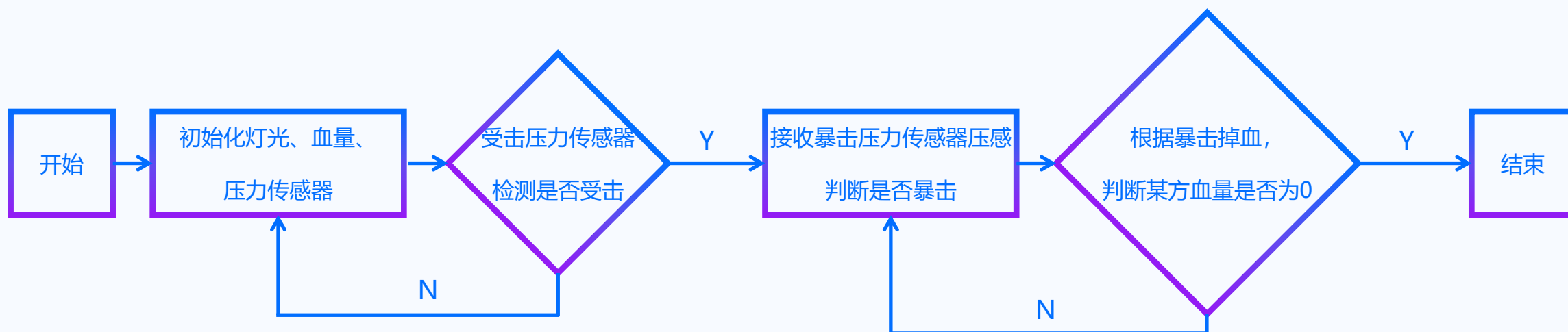
运行逻辑

代码中主要特点为检测玩家情绪，我使用压力传感器检测，以暴击的形式来表现玩家的情绪如何影响游戏的运行

暴击压力传感器的压感小于6000时，为普通攻击，掉一格血（在模型中三个灯珠为一格血，玩家有四格血）

暴击压力传感器的压感大于6000时小于10000时，为暴击，掉两格血

暴击压力传感器的压感等于10000时，为大暴击，掉三格血



关键代码展示

更多详情:

https://github.com/fredcoffe/emotion_game

人物初始化

```
// 人物1
int lost1 = MAX_HEALTH - characters[0].health;
for (int i = 0; i < lost1; i++) {
    for (int j = 0; j < 3; j++) {
        int idx = (MAX_HEALTH - 1 - i) * 3 + j; // 11->0
        if (idx >= 0 && idx < 12) {
            leds[idx] = CRGB::Red;
        }
    }
}
```

暴击压感判断

```
// 处理命中
void handleHit(int characterIndex, unsigned long currentTime) {
    Character &ch = characters[characterIndex];

    bool hitStatus = isHit(ch.hitPin);
    long critVal = getCritPressure(ch.critPin);

    if ((currentTime - ch.lastHitTime) < HIT_COOLDOWN) {
        return;
    }

    if (hitStatus && !ch.hitDetected) {
        int damage = 0;
        String desc;

        if (critVal < HIT_THRESHOLDS_NORMAL) {
            damage = 1;
            desc = "普通攻击";
        } else if (critVal < HIT_THRESHOLDS_CRIT) {
            damage = 2;
            desc = "暴击";
        } else {
            damage = 3;
            desc = "大暴击";
        }
    }
}
```

背景led颜色, 使用FastLed库来管理led

```
// 初始化背景LED (简单 Noise+火焰调色板)
void initializeBackgroundLEDs() {
    static uint8_t noiseSeed = 0;
    for (int i = BG_START_INDEX; i < BG_END_INDEX; i++) {
        uint8_t noise = inoise8((i - BG_START_INDEX) * 10, noiseSeed);
        // 采用火焰调色板
        CRGB color = ColorFromPalette(FireColors_p, noise, 255, LINEARBLEND);
        leds[i] = color;
    }
    noiseSeed++;
    FastLED.show();
}
```

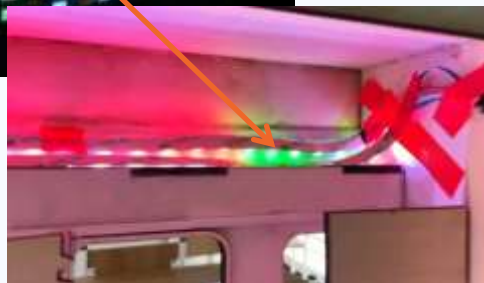
掉血时闪烁逻辑, 掉几格血闪烁几格血

```
void flickerDeductLEDs(int charIndex, int damage) {
    // 人物1 => 索引0~11(从右往左)
    // 人物2 => 索引12~23(从左往右)
    // damage格 => each格3LED
    // 快速闪2次
    for (int flick = 0; flick < 2; flick++) {
        // 闪烁(白色)
        if (charIndex == 0) {
            int lost = MAX_HEALTH - characters[0].health;
            // 需要扣掉 "damage" 格 => lastX
            // 右->左 => from (lost) up to (lost + damage - 1)
            for (int i = 0; i < damage; i++) {
                int offset = lost + i;
                for (int j = 0; j < 3; j++) {
                    int idx = (MAX_HEALTH - 1 - offset) * 3 + j;
                    if (idx >= 0 && idx < 12) {
                        leds[idx] = CRGB::White;
                    }
                }
            }
        } else {
            int lost = MAX_HEALTH - characters[1].health;
            // 左->右 => from (lost) up to (lost + damage - 1)
            for (int i = 0; i < damage; i++) {
                int offset = lost + i;
                for (int j = 0; j < 3; j++) {
                    int idx = 12 + offset * 3 + j;
                    if (idx >= 12 && idx < 24) {
                        leds[idx] = CRGB::White;
                    }
                }
            }
        }
        FastLED.show();
        delay(80);

        // 恢复原色
        updateLEDDisplay();
        FastLED.show();
        delay(80);
    }
}
```


关键硬件展示

led灯带, 此项目使用了118个灯珠



arduino uno



压力传感器模块, 用于把薄膜压力传感器的电压转换成压力值g



RP-C5-LT薄膜压力传感器, 用来测量是否受击



FSR406, 用来检测暴击压力





谢谢老师!

小组成员：杨亚澄 陈继威 梁佳栋 魏雨 曾建涛

指导老师：王凯

日期：2025.1.8

