

[D4SH] - 2022 NJU CPL Mid-Term Project

author: [Neonscape\(github\)](#)

1.工具链

- **IDE:** Visual Studio 2022 Community
- **操作系统:** Windows 11 22H2
- **编译器:** GNU C Compiler 12.1.0 (*msys version*)
- **图形库:** SDL2 (with extensions: *SDL2_image*, *SDL2_ttf*, *SDL2_mixer*)
- **图像处理工具:** SAI2 & Adobe Photoshop CC 2018

2.环境要求

编译环境要求:

- **操作系统:** Windows 10 / 11
- **编译器:** GNU C Compiler 12.1.0
- **依赖库:** SDL2 (and extensions) correctly installed
- **编译指令:**

```
gcc -I <SDL2_include_path> common.c init.c main.c mouse_widgets.c draw.c  
input.c -o D4SH.exe -lSDL2 -lSDL2_mixer -lSDL2_ttf -lSDL2_image -mwindows
```

运行环境要求:

- **操作系统:** Windows 10 / 11

3.关于游戏

i. 键位

使用 **[W]** 和 **[S]** 来 **跳起** 和 **滑铲**。（考虑到部分键盘没有方向键的可能）

使用 **[P]** 来暂停游戏。

使用 **[Esc]** 来退出游戏。

关于道具：大约每2,000 *Ticks*会刷新一个道具，使用道具会加分且使得玩家无敌一段时间。（这个真的
有做!!!）

ii. 设计思路

游戏的核心机制在于 **图层** 和 **计时器** 两个系统。

- **图层(Layer)** 分为 *动画图层(AnimatorLayer)* 和 *实体图层(Entity2DLayer)* 两种, 并由一个 **图层管理器(LayerManager)** 来统一管理各个图层中的对象。这样保证了游戏运行过程中各元素的显示顺序不会错乱, 也使得添加、删除、更改元素的操作变得更灵活, 更简单。
- **计时器(Timer)** 是基于 *游戏刻(Tick)* 设计的定时任务系统, 由单独的 **计时器管理器(TimeManager)** 实现管理和更新。一个 **Timer** 可以实现某个函数的周期性调用 (如随机绘图), 也可以实现某个函数的定时调用 (如取消玩家的无敌状态)。
- **动画类(Animator)** 是通过将其自身 **实体** 的材质依次替换为其所含 **材质列表(Animator.texList)** 的各个材质而实现的。 (*其实这个思路感觉也没有多难想?*)
- **按钮类(Button)** 实质上是一类特殊的 **实体(Entity2D)**, 由一个 **按钮池(ButtonPool)** 单独管理和渲染。判断按钮是否按下, 只需要在 **对应鼠标按键按下** 的时候, 检测鼠标位置 **是否在按钮区域中** 即可。
- 游戏速度随时间增加, 只需要让新生成的实体速度乘一个随时间增大的小数即可。这里我使用的是exp函数 (也就是说会涨的越来越快)。

iii. 游戏资源的来源

[!] 所有游戏中使用到的图片资源均为我自己绘制。

[i] 游戏中使用到的字体为开源字体 **得意黑(SmileySans-Oblique.ttf)**。

[i] 游戏使用的背景音乐: **夜光を呼ぶ - 幸祐** (没想到吧, 我是koko厨哒)

4. 一些说明 和 碎碎念

- 游戏码量共计 **1778** 行, 共 **87** 个函数。(并不推荐一行行看, **有些地方可能写的很烂**, 看了会高血压; **但是我可以保证其原创性。**)
- 游戏的分辨率为**固定的** **1600 * 900**, 无法在运行时改动 (也**非常不推荐改动**, 我使用的并非是可以无损放缩的矢量图形)。
- 游戏的帧率、分辨率、障碍物生成因子等数值均可以在 **defs.h** 文件中调整, 但我并不**对调整效果做保证**。
- 游戏的背景音乐可以将自己的mp3文件重命名为 **bgm.mp3** 后放入music文件夹中来替换。
- 本来想做下午——傍晚——午夜三个场景的, 但是做到一半发现, 以我的画工, 做了不如不做的好 ((((谁来教我画画啊啊啊啊))
- 在代码文件开头加上 **#define LOG_ON** 可以让游戏在运行过程中生成**debug.txt**文件, 里面可能会有一些有用的信息。
- 我个人认为这次 (有图形界面的) 项目的难点在于 **如何写出一个清晰合理的程序结构**。我帮不少同学看过程序, 大多数程序的问题都是其混乱不堪的结构 (甚至有人使用了**goto!!!**)
- 我个人觉得这次的项目还是**蛮有意思的** (当然, *我没写计算器*)。所以说, 谢谢你, 撒酱 (什)