实验报告

171860578 张梓悦

实验进度:

我完成了实验所要求的所有内容。

功能介绍:

需要使用上下左右键操作小白点,使其顺利到达红点的位置。其中蓝点和绿点随机移动, 若触碰到这两个点则游戏失败。

```
File Ed
Key pr
Key pr
Key pr
Key pr
You wi
Key pr
qemu-s
zhangz
Buildi
                                                                            ziyue/oslab/os-workbench/amg
ame/..
Buildi
Buildi
Buildi
Creati
Buildi
  LD -
  CREA
                                                                           uild/amgame-x86-qemu
#@qemu
wÄRNING: Image format was not specified for 'build/amgame-x86-gemu' and probing
guessed raw.
Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.

Specify the 'raw' format explicitly to remove the restrictions.
```

代码框架设计:

```
int next_frame = 0;
while (1) {
    while (uptime() < next_frame) ; // 等待一帧的到来
    read_keyy();
    move();
    if(px == rx && py == ry) {
        printf("You win!\n");
        reset();
}
else if((px == x1 && py == y1) || (px == x2 && py == y2)) {
        printf("You lost!\n");
        reset();
}
//printf("1fps\n");
next_frame += 1000 / FPS; // 计算下一帧的时间</pre>
```

在一个 while 循环中一帧一帧地接收键盘信息并移动小格点。由于本次实验是一个熟悉库函数的实验,故基本使用了全部库函数用以测试库函数的正确性。 其中使用 strcmp 来判断键盘信息:

```
if(strcmp(key_names[event.keycode], "UP") == 0) {
    if(py > 0)
        py--;
        clean();
        display();
}
else if(strcmp(key_names[event.keycode], "LEFT") == 0) {
    if(px > 0)
        px--;
        clean();
        display();
}
else if(strcmp(key_names[event.keycode], "RIGHT") == 0) {
    if(px < mw - 1)
        px++;
        clean();
        display();
}</pre>
```

遇到的 buu:

由于 git 没有添加库函数文件,导致回滚的时候 klib 直接变成了原始的那种直接 return0 的状态,小游戏直接挂掉,找了半天才发现是这个问题。