

# L0：运行在硬件上的小游戏

2020.3.5  
181240035  
刘春旭

## 实验进度：

- 完成了klib中的大部分函数（除 `malloc()` 和 `free()`），对 `vsprintf()` 进行了完善；
- 完成了小游戏；

——于是，他毫不客气地拿走了送来的分

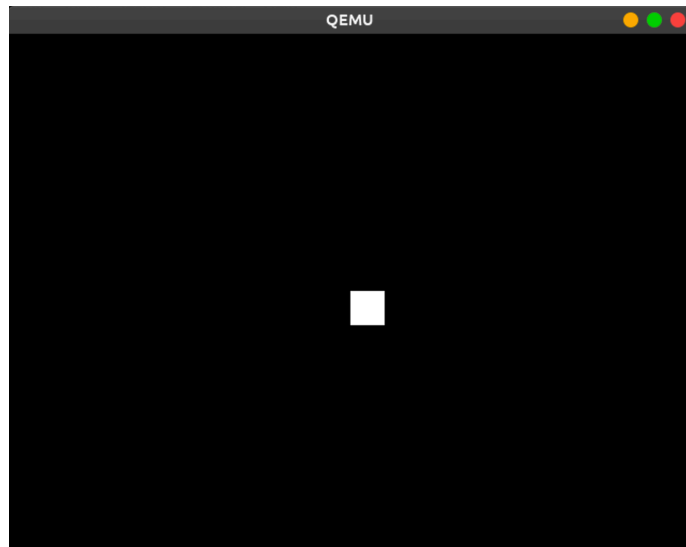
## 1. 代码框架：

从PA中的 `vga_test` 学习了代码的写法，和网站上给出的伪代码思路基本一致。

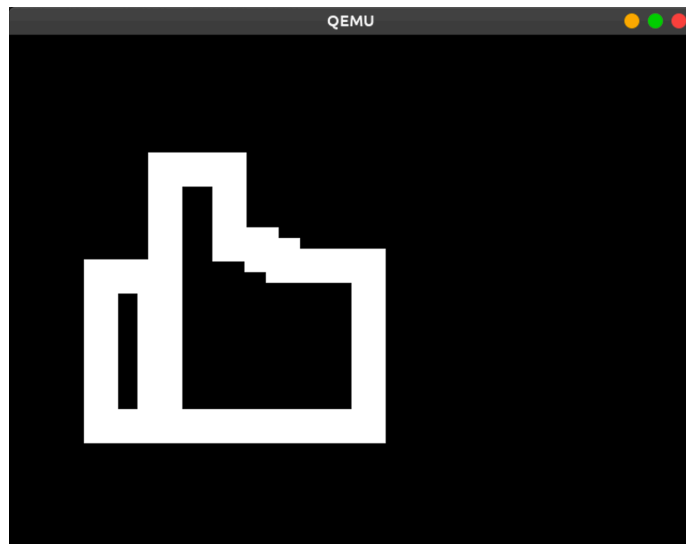
```
while (1) {
    unsigned long upt = uptime(); //得到当前时间
    key = read_key(); // 读取键码
    if (key != _KEY_NONE)
    {
        if(key == _KEY_ESCAPE) _halt(1); // 判断ESC（每个循环1次）
        if (upt - last > 1000 / FPS) { //更新屏幕信息（每秒FPS次）
            update(key); //update_game_process
            redraw(); //redraw_window
            last = upt; // 更新上一帧的时间
        }
        //省略FPS信息显示
    }
}
```

## 2. 游戏内容：画图小游戏

开始时界面为（白色方块为画笔）：



可以使用方向键操作画笔上下左右移动，形成有趣的图案（点赞）：



当画笔再次回到初始位置时，进行清屏操作，回到图一的样子。

### 3. 游戏进程更新

```
void update(int key){
    switch (key)
    {
        case _KEY_UP:    y = y - v; break;
        case _KEY_DOWN:  y = y + v; break;
        case _KEY_LEFT:  x = x - v; break;
        case _KEY_RIGHT: x = x + v; break;
        default:         x = x; y = y; break;
    }
    check(); //判断x, y坐标是否回到原点
}
```

主要的困难（？）可能是看懂AM中提供的API，理解所有的API是干什么的，AM是如何与设备寄存器沟通的(`_io_write()`、`_io_read()`)等即可。