

## L0 - amgame 实验报告

计算机科学与技术系 171860508 张天昀

### 游戏内容

经典游戏 - 俄罗斯方块

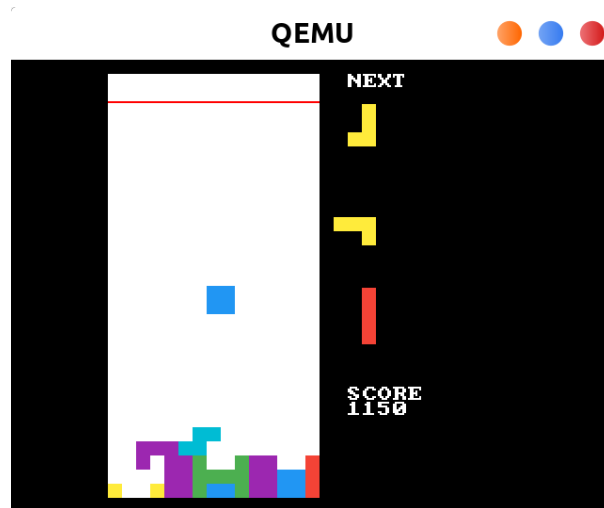


图 1: 游戏运行截图

游戏方式：

- AD/左右键：水平移动
- W/上键：旋转
- S/下键：下落
- 空格：HOLD（切换）

计分方式：

- 自然下落或用户操作：-1 分
- 消除一（至四行）：500 ~ 2000 分
- 不支持对 T-spin 等高级操作的额外赋分

### 扫操作

本次游戏的实现大量的借鉴了 ICS-PA 的骚数组。

tetris.h 中定义了一个结构体 keyCodeMapping，定义如下

```
struct KeyCodeMapping {  
    int code;
```

```
const char* name;
struct Tetromino (*func) (struct Tetromino, bool);
bool param;
};
```

由于 C 没有成员函数这个概念，只能使用函数指针来替代成员函数。这样在 `playTetris` 这个函数里只需要用一个循环来比对 `code`，然后执行对应的 `func`，把返回的新俄罗斯方块的状态保存好就可以啦！

后来发现这个方法可以直接复用来实现自动下落这个过程，只需要将传给 `playTetris` 的按键设置为 `_KEY_NONE`，然后在数组中添加一项，对应的函数指针为 `fallTetromino(.force=false)` 就可以用一个函数处理所有的俄罗斯方块移动操作了。

而上述的 `func` 总共有 4 种，分别是：

```
struct Tetromino moveTetromino(struct Tetromino, bool);
struct Tetromino fallTetromino(struct Tetromino, bool);
struct Tetromino spinTetromino(struct Tetromino, bool);
struct Tetromino swapTetromino(struct Tetromino, bool);
```

这些处理函数的第二个参数分别有不同的意义，`move` 中表示向左向右，`fall` 中表示自动或用户强制下落，`spin` 表示顺时针或逆时针，`swap` 中保持函数格式一致，并没有什么用。

## 超级 bug

一开始刚刚实现好“消除一或多行”的 `clearTetrominos` 函数时，一旦执行消除整个游戏就会崩掉然后出现 QEMU 重启的界面，随后游戏重新开始。

通过使用 `gdb` 调查，发现在执行这个函数后出现问题的地方是主循环的 `state.nextFrame += 1000 / state.FPS`；处报 `SIGFPE` 错误，肯定是从执行这个函数到返回最顶层的二三十行里哪里内存泄露了。然后一直找啊找啊找啊找啊大概找了两个多小时才发现自己在 `memset` 中使用的 `sizeof` 后面的类型写错了。应该写 `int` 类型的，写了 `struct Tetromino`，清空的内存超级加倍，直接爆炸。

(结论：眼瞎写程序有害健康。)

## 小 bug (feature?)

自己测试游戏的时候发现竖杠 (I-type) 和田 (O-type) 总是不出现，游戏体验极差。研究代码后发现自己是对所有的方块形状进行随机，方块共有 19 种，竖杠对应 2 种，而田型方块只有一种（因为怎么旋转都是他自己），所以随机数很难抽到这两种。

于是修改代码，增加了“类别 (`tetrominoGenres`)”这个数组，每次先对 `genre` 进行随机，抽取一种，然后在同 `genre` 中随机抽取一个 `type`。稀有俄罗斯方块也能够多次出现，游戏体验极佳。