

# OS2018 Lab0 实验报告

姓名：陈劭源

学号：161240004

April 6, 2018

## 1 玩法说明

运行 `make run`，进入游戏。注意，工程的 `Makefile` 文件已经被修改，请运行修改过后的 `Makefile` 文件。

进入 `qemu` 后，首先运行自检程序，并在终端中打印当前机器的基本情况、设备信息 and 时间信息。随后，播放 `Splash` 界面（可按 `return` 键跳过）。按 `C` 键进入游戏。

左右键控制当前方块的位置，上键顺时针旋转当前方块，下键逆时针旋转当前方块，空格键加速下落。消除单行可获得 1 分，单次消除 2 行获得 3 分，单次消除 3 行获得 5 分，单次消除 4 行可获得 8 分。其他规则和经典的俄罗斯方块游戏基本一致。由于 `Wall kick` 系统尚未实现，诸如 `L-spin` 等骚操作暂时无法使用...

## 2 游戏截图



Figure 1: Splash 界面

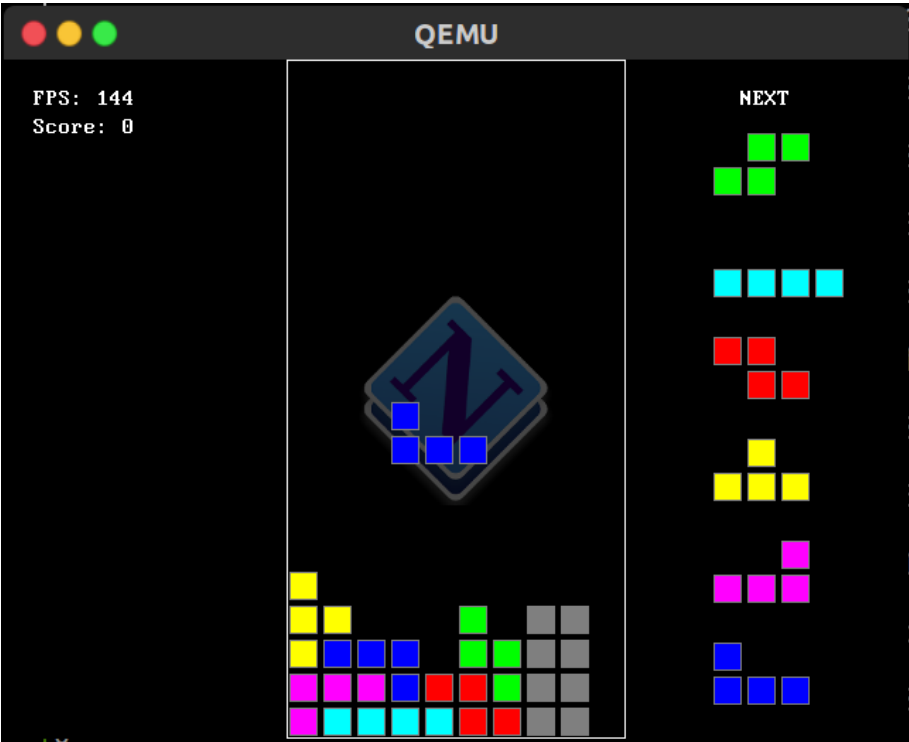


Figure 2: 游戏界面

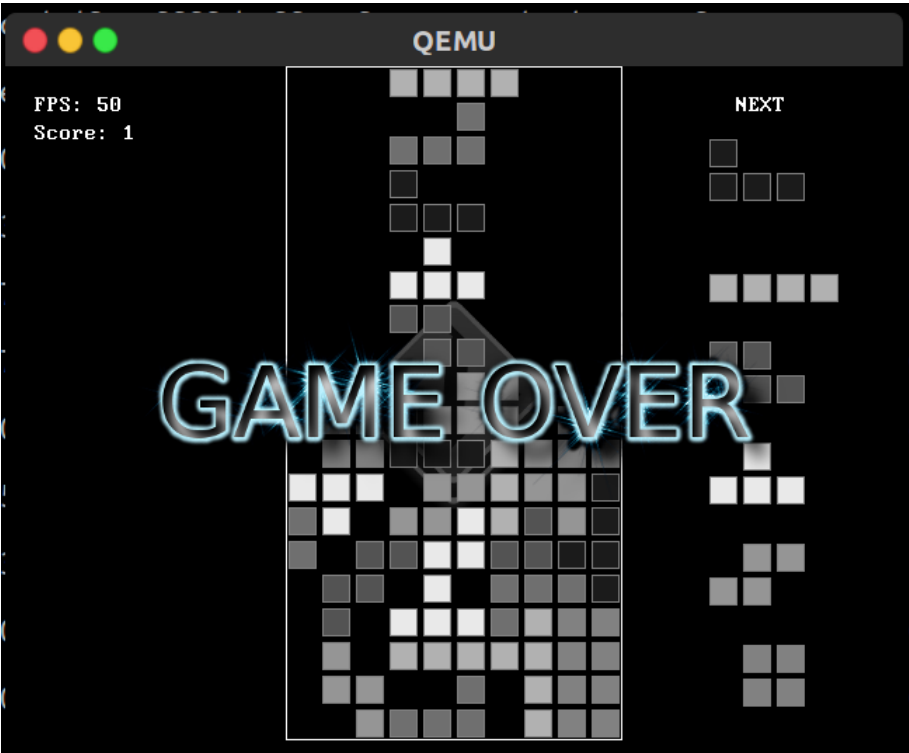


Figure 3: Gameover 界面

### 3 主要技术说明

#### 3.1 图片和文字

游戏中的图片素材，除 ProjectN 的 logo 外，均使用 GIMP 工具制作。

使用 GIMP 的 export 功能，可将图片导出为 C 源代码格式，然后逐点绘制在屏幕上即可。

文字使用点阵字模。每个 ASCII 文字的形狀以 16\*8 点阵的形式存储在程序中，绘制时根据字模数据逐点绘制即可。

相关图形库函数：gDrawString, gDrawStringA (font.c), gDrawImage\* (image.c)

#### 3.2 Splash 界面的渐变效果

绘制图片时，可以附加一个透明度参数  $\alpha$ ，使得绘制出的颜色满足下式

$$\begin{cases} r = r_f \alpha + r_b (1 - \alpha) \\ g = g_f \alpha + g_b (1 - \alpha) \\ b = b_f \alpha + b_b (1 - \alpha) \end{cases} \quad (1)$$

其中  $r_f, g_f, b_f$  为图像颜色的三个分量， $r_b, g_b, b_b$  为背景色的三个分量。

在播放渐变动画时，只需要将  $\alpha$  从 0 逐渐调整为 1，就可实现渐变效果。

相关图形库函数：gChannelBlend, gColorBlend (graphics.h), gDrawImageA\* (image.c)

#### 3.3 Gameover 界面的变灰效果

在绘制图片时，还可附加一个颜色矩阵  $M$ 。 $M$  是  $3 \times 3$  的矩阵，用于对颜色的三个分量进行线性变换：

$$\begin{cases} r' = rM_{11} + gM_{12} + bM_{13} \\ g' = rM_{21} + gM_{22} + bM_{23} \\ b' = rM_{31} + gM_{32} + bM_{33} \end{cases} \quad (2)$$

其中， $M_I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  表示恒等变换， $M_G = \begin{bmatrix} 0.33 & 0.59 & 0.11 \\ 0.33 & 0.59 & 0.11 \\ 0.33 & 0.59 & 0.11 \end{bmatrix}$  表示灰度变换（即将彩色图片变换为黑白图片）。

这样，只需设置一个混合因子  $\lambda$ ，并且使用  $\lambda M_G + (1 - \lambda) M_I$  绘制图片，同时将  $\lambda$  从 0 逐渐调整为 1，即可实现逐渐变灰的效果。

相关图形库函数：gColorMatrixTransform (graphics.h), gDrawImageM (image.c)

### 4 代码结构

```

├── .git                                # git数据
├── ...
├── am                                  # 抽象机相关文件
├──   ├── mbr                          # x86的主引导扇区(用于加载内核)
├──   └── am-x86-qemu.a                # AM API在x86上的实现
├── build                              # 编译结果目录
├──   ├── ...
├── include                            # 头文件
├──   ├── am.h                        # AM API
├──   ├── amdev.h                     # AM设备API
├──   ├── arch.h                      # x86体系结构相关的声明
├──   ├── game.h                      # 游戏相关的函数
├──   ├── graphics.h                  # 图形库
├──   └── resource.h                   # 资源列表

```

|                  |                |
|------------------|----------------|
| └─ test.h        | # 自检程序相关函数     |
| └─ libc          | # 标准C库的部分实现    |
| └─ include       | # 头文件          |
| └─ assert.h      |                |
| └─ ctype.h       |                |
| └─ stdio.h       |                |
| └─ stdlib.h      |                |
| └─ string.h      |                |
| └─ time.h        |                |
| └─ src           | # C库函数的实现      |
| └─ ...           |                |
| └─ resource      | # 资源文件         |
| └─ ...           |                |
| └─ src           | # 源代码          |
| └─ game          | # 游戏代码         |
| └─ game.c        | # 全局相关         |
| └─ gameover.c    | # Gameover界面   |
| └─ splash.c      | # Splash界面     |
| └─ tetris.c      | # 游戏的主体部分      |
| └─ graphics      | # 图形库的实现       |
| └─ draw.c        | # 绘制图形         |
| └─ font.c        | # 绘制文字         |
| └─ graphics.c    | # 图形库全局相关      |
| └─ image.c       | # 绘制图片         |
| └─ keyboard.c    | # 键盘事件处理       |
| └─ main.c        | # 主程序          |
| └─ test.c        | # 自检程序         |
| └─ .gitignore    | # .gitignore文件 |
| └─ git-commit.sh | # git追踪用脚本     |
| └─ report.pdf    | # 本实验报告        |
| └─ Makefile      | # Makefile文件   |