忻州师范学院

C语言课程设计报告

题目：俄罗斯方块

专业：

班级：

姓名：

日期：

|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | **姓名** | **分工比例** | **总成绩** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. 设计目的

该俄罗斯方块程序的设计目的在于提供一个在控制台运行的经典俄罗斯方块游戏。通过实现方块随机生成、自动下落、玩家控制、得分计算和游戏结束判定等功能，为用户提供娱乐和逻辑思维训练。

1. 功能描述

**功能描述**

1. **方块生成与旋转**：
   * 随机生成7种方块（I、O、T、S、Z、L、J）及其对应旋转形态。
   * 方块的旋转和移动操作支持合法性检查，避免越界或重叠。
2. **游戏界面**：
   * 使用ASCII字符在控制台绘制游戏界面，包括游戏区域、边框、分数统计区域和“下一方块”展示区域。
   * 通过设置颜色区分不同类型的方块。
3. **玩家控制**：
   * 玩家通过键盘控制方块下移、左右移动及旋转。
   * 游戏支持暂停和重新开始。
4. **得分与记录**：
   * 当一行被完全填满时，清除该行并增加分数。
   * 最高得分保存在本地文件中，并在游戏开始时加载。
5. **游戏结束判定**：
   * 当方块堆叠到达顶部时，判定游戏结束，并允许玩家重新开始或退出。

。

1. 总体设计

程序采用模块化设计，主要功能分为以下模块：

1. **主程序模块**：
   * main：初始化全局变量，加载最高分，启动游戏界面和逻辑。
   * StartGame：核心游戏循环，负责控制方块的生成、移动、旋转、得分计算以及游戏结束判定。
2. **界面管理模块**：
   * InitInterface：绘制游戏初始界面，包括边框、提示信息和得分区域。
   * HideCursor 和 CursorJump：隐藏光标及控制光标位置，用于实现动态界面更新。
3. **方块管理模块**：
   * InitBlockInfo：初始化7种方块及其旋转形态。
   * DrawBlock 和 DrawSpace：负责在指定位置绘制或清除方块。
   * IsLegal：判断方块在某位置的合法性。
4. **得分与存档模块**：
   * JudeFunc：检测是否得分及游戏是否结束，更新分数或重新排列方块。
   * ReadGrade 和 WriteGrade：从文件读取最高分并保存新的最高分。
5. **用户交互模块**：
   * 键盘操作部分，通过检测键盘输入实现玩家对方块的控制（方向键移动、空格旋转、暂停、重新开始等）。

游戏流程图如下：



1. 程序实现
2. 程序代码

import matplotlib.pyplot as plt

import matplotlib.patches as patches

import matplotlib

matplotlib.use('TkAgg') # 更改 Matplotlib 后端

from matplotlib import rcParams

import matplotlib.pyplot as plt

# 设置中文字体

rcParams['font.sans-serif'] = ['SimHei']

rcParams['axes.unicode\_minus'] = False

# Initialize the figure

fig, ax = plt.subplots(figsize=(10, 15))

ax.set\_xlim(0, 10)

ax.set\_ylim(0, 15)

ax.axis("off")

# Helper function to draw rectangles with text

def draw\_box(x, y, width, height, text, fontsize=10):

rect = patches.Rectangle((x, y), width, height, edgecolor="black", facecolor="lightgrey")

ax.add\_patch(rect)

ax.text(x + width/2, y + height/2, text, fontsize=fontsize, ha="center", va="center")

# Draw the main flowchart boxes

draw\_box(3.5, 14, 3, 1, "启动程序", 12)

draw\_box(3.5, 12.5, 3, 1, "初始化全局变量", 10)

draw\_box(3.5, 11, 3, 1, "读取最高分", 10)

draw\_box(3.5, 9.5, 3, 1, "初始化界面", 10)

draw\_box(3.5, 8, 3, 1, "初始化方块信息", 10)

draw\_box(3.5, 6.5, 3, 1, "主游戏循环", 12)

draw\_box(0.5, 5, 3, 1, "生成新方块", 10)

draw\_box(6.5, 5, 3, 1, "玩家键盘控制", 10)

draw\_box(3.5, 5, 3, 1, "方块下落逻辑", 10)

draw\_box(3.5, 3.5, 3, 1, "合法性检查", 10)

draw\_box(3.5, 2, 3, 1, "得分与结束判定", 10)

draw\_box(3.5, 0.5, 3, 1, "保存最高分\n退出或重开", 10)

# Add arrows

arrowprops = dict(arrowstyle="->", lw=1.5, color="black")

ax.annotate("", xy=(5, 13.9), xytext=(5, 12.6), arrowprops=arrowprops)

ax.annotate("", xy=(5, 12.4), xytext=(5, 11.1), arrowprops=arrowprops)

ax.annotate("", xy=(5, 10.9), xytext=(5, 9.6), arrowprops=arrowprops)

ax.annotate("", xy=(5, 9.4), xytext=(5, 8.1), arrowprops=arrowprops)

ax.annotate("", xy=(5, 7.9), xytext=(5, 6.6), arrowprops=arrowprops)

ax.annotate("", xy=(5, 6.4), xytext=(5, 5.1), arrowprops=arrowprops)

ax.annotate("", xy=(2, 4.9), xytext=(2, 5.6), arrowprops=arrowprops)

ax.annotate("", xy=(8, 4.9), xytext=(8, 5.6), arrowprops=arrowprops)

ax.annotate("", xy=(5, 4.9), xytext=(5, 5.6), arrowprops=arrowprops)

ax.annotate("", xy=(5, 3.4), xytext=(5, 4.9), arrowprops=arrowprops)

ax.annotate("", xy=(5, 1.9), xytext=(5, 3.4), arrowprops=arrowprops)

ax.annotate("", xy=(5, 0.4), xytext=(5, 1.9), arrowprops=arrowprops)

# Connect sub-boxes to main logic

ax.annotate("", xy=(3, 6.4), xytext=(1.5, 5.9), arrowprops=arrowprops)

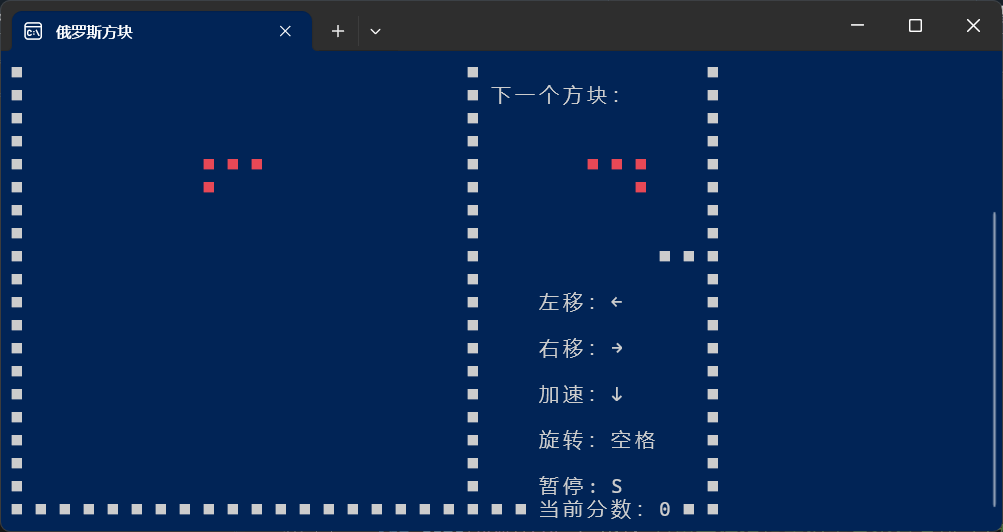
ax.annotate("", xy=(7, 6.4), xytext=(8.5, 5.9), arrowprops=arrowprops)

# Display the chart

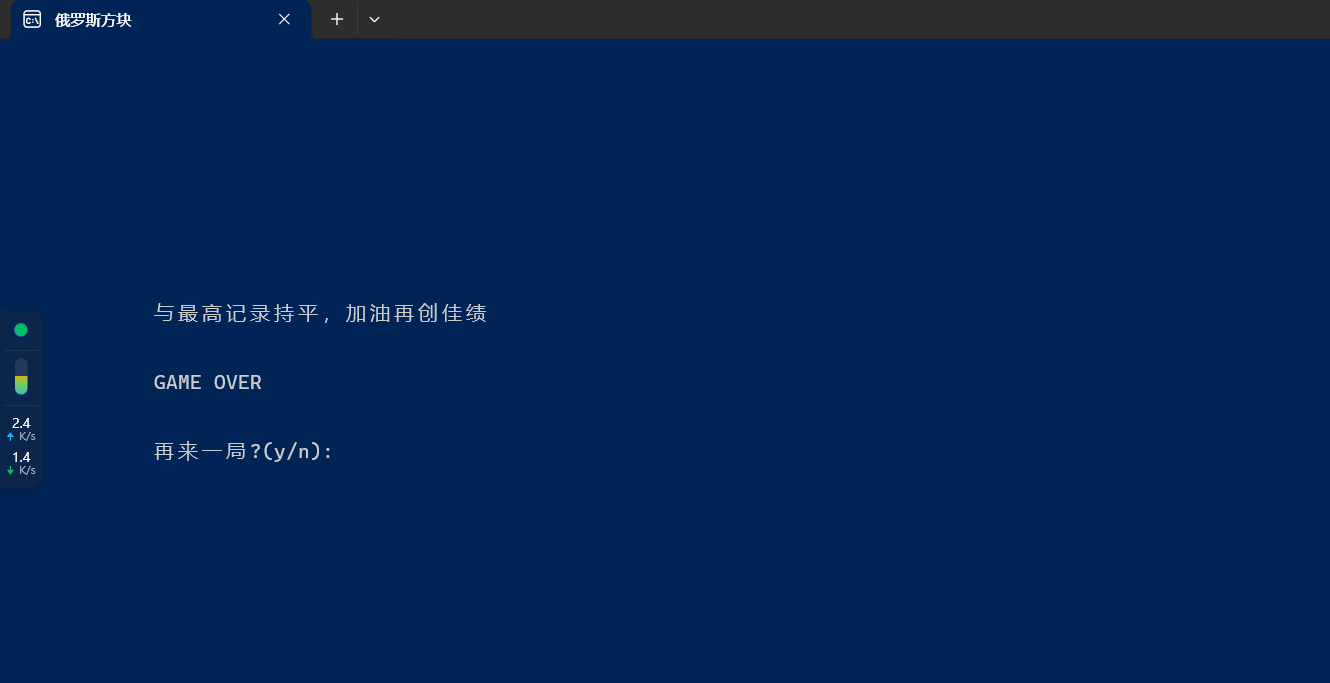
plt.show()

1. 运行结果

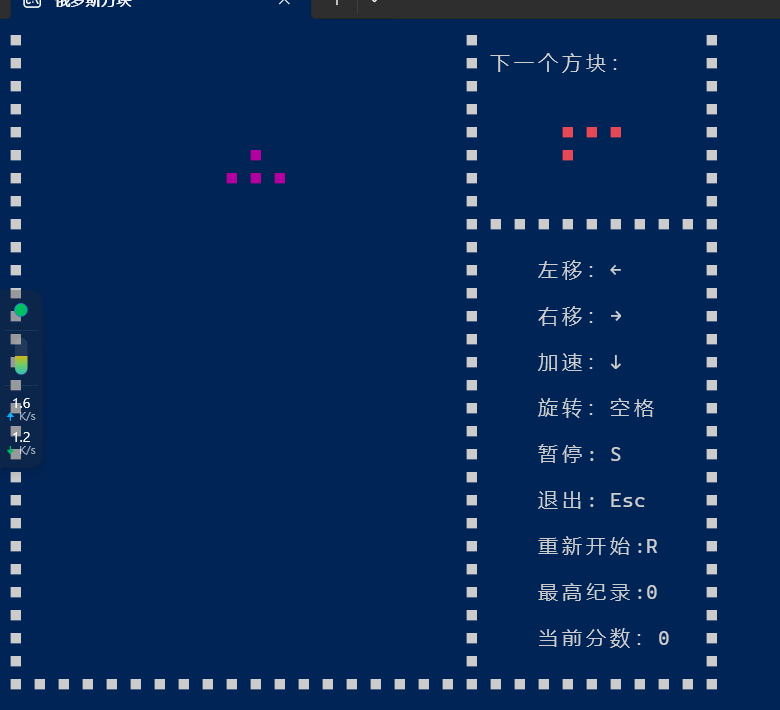
开始界面：



结束界面：



重新开始界面：



游戏中界面：

