

第五次作业

一、符合测试相关问题解答

1. 简述什么是符合测试

符合测试是核与粒子探测电子学中典型的时间关联事件筛选方法。系统为多路探测器信号设定一个极短的固定时间区间，称为符合时间窗。当且仅当所有探测通道的脉冲信号在同一时间窗内同时到达时，符合电路输出一次有效计数；若任意一路信号缺失，则不产生有效输出。该技术主要用于筛选由同一物理事件产生的关联信号，剔除随机噪声。

根据应用场景不同，符合测试分为三类：

- (1) 瞬时符合：多路信号在时间窗内同步到达，适用于粒子穿过多层探测器、级联 γ 衰变等同步物理过程检测，是时间分辨率测量的核心工作模式；
- (2) 延迟符合：人为对单路信号施加固定时间延迟，通过测量两路信号时间差，实现粒子半衰期、脉冲时间特性等参数测量；
- (3) 反符合：逻辑与常规符合相反，仅单通道有信号、其余通道无信号时输出有效计数，主要用于剔除宇宙射线、环境散射等本底噪声。

2. 开展符合测试的意义

在粒子探测系统中，环境放射性、宇宙射线、电路噪声会产生大量无关联随机脉冲，严重干扰有效物理信号采集，开展符合测试的核心意义如下：

- (1) 抑制随机本底，提纯有效物理事例。随机噪声脉冲在窄时间窗内同步出现的概率极低，通过时间关联筛选，可最大程度过滤无效噪声，仅保留同一粒子触发多路探测器的真实关联事件，大幅提升系统信噪比；
- (2) 实现物理过程与粒子类型鉴别。不同粒子、核衰变过程的信号关联特征存在差异，符合筛选可精准区分带电粒子、 γ 光子等不同事例，支撑核衰变规律、粒子径迹等物理研究；
- (3) 支撑探测器核心参数测量。探测器时间分辨率、符合分辨时间、探测效率等关键参数，无法通过单路脉冲测算，必须依托同源符合事件的统计分布求解；

(4) 精简有效数据，降低设备负载。从硬件层面提前过滤海量无效噪声事件，减少后端数据采集、存储与运算压力，提升系统运行效率。

3. 测量时间分辨率必须使用符合信号的原因

探测器时间分辨率定义：同一粒子触发两层探测器时，两路输出脉冲的时间差分布的高斯半高宽或标准差，用于表征探测器分辨粒子到达时刻的精准能力。其测量必须依托符合信号，原因如下：

(1) 仅符合事件的时间抖动为系统固有误差。随机噪声脉冲的时间差无任何物理意义，只有符合筛选出的同源粒子信号，其时间涨落仅由探测器电离统计、电子学系统固有抖动导致，是测算时间分辨率的有效数据；

(2) 仅符合事例可构建精准的时间差谱。测量通过时间幅度转换器将两路脉冲时间差转化为电压幅度，再由多道分析仪统计形成时间差谱。若无符合筛选，随机噪声会铺满整个时间轴，导致谱线完全失真；

(3) 保障参数拟合结果准确。对符合事件形成的高斯谱线进行拟合，可得到真实的时间分辨率参数。混入随机本底会导致谱线展宽，测算结果偏大，完全失去工程与物理参考价值。

二、朗道效应及能量沉积不均的时间误差消除

1. 朗道效应物理定义

高能带电粒子穿过探测器灵敏介质时，会通过电离作用损失能量。根据贝特-布洛赫公式，粒子平均能损固定，但单次电离过程中，产生的电子-空穴对数量存在非高斯随机统计涨落，该现象即为朗道效应，对应的分布为朗道分布。

朗道分布具有明显特征：峰值对应粒子最概然能损，右侧存在显著的高能长拖尾。这导致同一能量、同一类型的入射粒子，在探测器内的电离能量沉积幅度差异极大。在双层探测器测量场景中，上下两层介质的能量沉积涨落相互独立，最终造成两路探测脉冲幅度不一致。

2. 能量沉积不均引发的时间误差机制

常规脉冲计时采用前沿甄别方式，以固定电压阈值作为信号触发计时点。受朗道涨落影响，脉冲幅度随机波动，会产生典型的时间游走误差：

(1) 高幅度脉冲上升沿陡峭，可快速跨过触发阈值，系统计时偏早；

(2) 低幅度脉冲上升沿平缓，跨过阈值的时间延后，系统计时偏晚；

该误差与脉冲幅度强相关，会直接展宽探测器时间差谱，严重劣化时间分辨率测量精度。

3. 能量沉积不均导致时间误差的消除方法

针对朗道涨落带来的幅度差异与时间游走误差，行业主流修正与消除方案分为硬件核心方案、软件修正方案及辅助优化方案三类：

(1) 硬件核心方案：恒比甄别器（CFD）

该方案是消除时间游走的标准工程手段，彻底摆脱固定电压阈值的局限。其原理为：将输入脉冲分为两路，一路做固定延迟处理，一路做固定比例衰减处理，通过两路信号差分运算，以差分波形过零点作为统一计时触发点。触发时刻对应脉冲峰值的固定比例，与脉冲绝对幅度无关，可从硬件层面完全消除朗道涨落带来的时间游走误差，广泛应用于闪烁体、硅探测器的高精度时间测量。

(2) 软件修正方案：数字化离线校正

适用于全波形数字化采集系统。首先通过标准放射源标定探测器的“脉冲幅度-时间偏移”校正曲线；再采集每一路符合事件的脉冲峰值幅度与原始时间戳；最后依据标定曲线对原始计时数据进行补偿修正，抵消幅度相关的时间偏移，优化时间差谱质量。

(3) 辅助优化方案：阈值事例筛选

设置最小电离电荷阈值，剔除朗道分布中极低能量沉积对应的小幅值脉冲。此类脉冲时间游走误差最大，剔除后可压缩谱线拖尾、减少异常数据，辅助提升测量精度，仅作为补充优化手段。

(4) 结构优化方案：双端信号读出平均

在单层探测器两端同时采集信号并独立计时，取两端计时平均值抵消单侧能量沉积涨落带来的时间偏移，通常与恒比甄别器搭配使用，进一步提升系统测量稳定性与精度。

```
41 def calc_score(cfg: dict, remaining: int) -> int:
42     """根据剩余次数计算本局得分"""
43     return cfg["score_base"] + remaining * 15 + random.randint(0, 9)
44
45
46 def play_round(diff_key: str, highscores: dict) -> int:
47     """进行一次游戏，返回本局得分（未通过返回 0）"""
48     cfg = DIFFICULTIES[diff_key]
49     low, high = cfg["range"]
50     target = random.randint(low, high)
51     max_g = cfg["max_guesses"]
52
53     print(f"\n{'!' * 40}")
54     print(f" 难度: {cfg['name']} | 数字范围: {low}~{high} | 最大次数: {max_g}")
55     print(f" 当前最高分: {highscores.get(diff_key, 0)}")
56     print(f"{'!' * 40}")
57
58     for attempt in range(1, max_g + 1):
59         raw = input(f"\n[{attempt}/{max_g}] 请输入你猜的数字: ").strip()
60
61         if raw.lower() in ("q", "quit", "exit"):
62             print("> 已退出本局")
63             return 0
64
65         if not raw.isdigit():
66             print("X 请输入有效整数")
67             continue
68
69         guess = int(raw)
70         if guess < low or guess > high:
71             print(f"X 数字必须在 {low}~{high} 之间")
72             continue
73
74         if guess == target:
75             remaining = max_g - attempt
76             score = calc_score(cfg, remaining)
77             print(f"\n🎉 恭喜! 答案就是 {target}! 剩余次数 {remaining}, 得分 {score}")
78             return score
79         elif guess < target:
80             print("↑ 小了, 再大一点")
81     else:
```

```
1 """
2 猜数字小游戏
3 - 三档难度
4 - 最高分本地存档 (JSON)
5 - 重新开始菜单
6 """
7
8 import json
9 import random
10 import os
11
12 HIGH_SCORE_FILE = os.path.join(os.path.dirname(__file__), "guess_highscores.json")
13
14 DIFFICULTIES = {
15     "1": {"name": "简单", "range": (1, 50), "max_guesses": 10, "score_base": 100},
16     "2": {"name": "普通", "range": (1, 100), "max_guesses": 7, "score_base": 200},
17     "3": {"name": "困难", "range": (1, 200), "max_guesses": 5, "score_base": 400},
18 }
19
20
21 def load_highscores() -> dict:
22     """从本地 JSON 读取最高分，结构: {"1": 0, "2": 0, "3": 0}"""
23     if os.path.exists(HIGH_SCORE_FILE):
24         try:
25             with open(HIGH_SCORE_FILE, encoding="utf-8") as f:
26                 return json.load(f)
27         except (json.JSONDecodeError, OSError):
28             pass
29     return {"1": 0, "2": 0, "3": 0}
30
31
32 def save_highscores(scores: dict) -> None:
33     """将最高分写回本地 JSON"""
34     try:
35         with open(HIGH_SCORE_FILE, "w", encoding="utf-8") as f:
36             json.dump(scores, f, ensure_ascii=False, indent=2)
37     except OSError as e:
38         print(f"⚠️存档写入失败: {e}")
39
40
```

```
1 """
2 猜数字小游戏
3  - 三档难度
4  - 最高分本地存档 (JSON)
5  - 重新开始菜单
6 """
7
8 import json
9 import random
10 import os
11
12 HIGH_SCORE_FILE = os.path.join(os.path.dirname(__file__), "guess_highscores.json")
13
14 DIFFICULTIES = {
15     "1": {"name": "简单", "range": (1, 50), "max_guesses": 10, "score_base": 100},
16     "2": {"name": "普通", "range": (1, 100), "max_guesses": 7, "score_base": 200},
17     "3": {"name": "困难", "range": (1, 200), "max_guesses": 5, "score_base": 400},
18 }
19
20
21 def load_highscores() -> dict:
22     """从本地 JSON 读取最高分, 结构: {"1": 0, "2": 0, "3": 0}"""
23     if os.path.exists(HIGH_SCORE_FILE):
24         try:
25             with open(HIGH_SCORE_FILE, encoding="utf-8") as f:
26                 return json.load(f)
27         except (json.JSONDecodeError, OSError):
28             pass
29     return {"1": 0, "2": 0, "3": 0}
30
31
32 def save_highscores(scores: dict) -> None:
33     """将最高分写回本地 JSON"""
34     try:
35         with open(HIGH_SCORE_FILE, "w", encoding="utf-8") as f:
36             json.dump(scores, f, ensure_ascii=False, indent=2)
37     except OSError as e:
38         print(f"▲存档写入失败: {e}")
39
40
```

```
whoami
Thought for 2s (ctrl+o to expand)
● PowerShell(whoami)
  └─ laptop-u89fdfhf\86131
Thought for 1s (ctrl+o to expand)
● You're logged in as laptop-u89fdfhf\86131 – a local Windows user account on this machine.
* Cogitated for 5s
/cd D:\code\game
└─ Couldn't find a directory at D:\code\game.
写Python控制台猜数字小游戏, 包含三档难度、最高分本地存档、重新开始菜单, 输出完整可运行代码
● Thinking for 1s... (ctrl+o to expand)
  └─ The user wants me to create a Python console-based number guessing game with: 1. Three difficulty levels 2. Local high score storage 3. Restart menu 4. Complete, runnable code Let me write this as a complete Python script.
Create file
D:\code\guess_number.py
1 """
2 猜数字小游戏
3  - 三档难度
4  - 最高分本地存档 (JSON)
5  - 重新开始菜单
6 """
7
8 import json
9 import random
10 import os
11
12 HIGH_SCORE_FILE = os.path.join(os.path.dirname(__file__), "guess_highscores.json")
13
14 DIFFICULTIES = {
```



```
开发Python猜数字控制台 x + v
PowerShell(whoami)
└─ laptop-u89fdhfh\86131

Thought for 1s (ctrl+o to expand)

• You're logged in as laptop-u89fdhfh\86131 - a local Windows user account on this machine.

* Cogitated for 5s

/cd D:\code\game
└─ Couldn't find a directory at D:\code\game.

写Python控制台猜数字小游戏，包含三档难度、最高分本地存档、重新开始菜单，输出完整可运行代码

Thought for 1s (ctrl+o to expand)

• Write(D:\code\guess_number.py)
└─ Wrote 165 lines to D:\code\guess_number.py
  1 """
  2 猜数字小游戏
  3 - 三档难度
  4 - 最高分本地存档 (JSON)
  5 - 重新开始菜单
  6 """
  7
  8 import json
  9 import random
 10 import os
  ... +155 lines (ctrl+o to expand)

Thought for 1s (ctrl+o to expand)

• PowerShell(python D:\code\guess_number.py)
└─ Error: Exit code 1
=====
Traceback (most recent call last):
  File "D:\code\guess_number.py", line 165, in <module>
    main()
  File "D:\code\guess_number.py", line 105, in main
```

```
开发Python猜数字控制台 x + v

Thought for 2s (ctrl+o to expand)

• gbk 编码不支持 emoji，替换为纯 ASCII 风格：

• Update(D:\code\guess_number.py)
└─ Added 9 lines
  8 import json
  9 import random
 10 import os
 11 +import sys
 12
 13 +# 确保控制台能正确显示中文
 14 +if sys.stdout.encoding and sys.stdout.encoding.upper() != "UTF-8":
 15 +    try:
 16 +        sys.stdout.reconfigure(encoding="utf-8")
 17 +    except AttributeError:
 18 +        import io
 19 +        sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding="utf-8")
 20 +
 21 HIGH_SCORE_FILE = os.path.join(os.path.dirname(__file__), "guess_highscores.json")
 22
 23 DIFFICULTIES = {

Thought for 1s (ctrl+o to expand)

• PowerShell(python D:\code\guess_number.py)
└─ Running in the background (↓ to manage)
└─ (timeout 15s)

Thought for 1s (ctrl+o to expand)

• Stop Task
└─ python D:\code\guess_number.py... · stopped

Thought for 1s, read 1 file (ctrl+o to expand)

• 启动正常 (未崩溃)。让我确认最终代码完整无误：

Thought for 1s, read 1 file (ctrl+o to expand)
```

```
开发Python猜数字控制台 × +
● 启动正常 (未崩溃)。让我确认最终代码完整无误:
Thought for 1s, read 1 file (ctrl+o to expand)
● 已生成完整可运行的猜数字游戏: D:\code\guess_number.py
运行方式: python D:\code\guess_number.py
---
功能一览

```

模块	说明
三档难度	简单 1~50/10次 / 普通 1~100/7次 / 困难 1~200/5次
计分系统	基础分 + 剩余次数×15 + 随机0~9, 剩余越多分越高
最高分存档	自动读写同目录 guess_highscores.json, 支持查看 & 重置
重新开始菜单	每局结束后可选「y 换难度继续」或回车回主菜单
中途退出	猜数字时可输入 q / quit / exit 提前结束本局
编码兼容	自动处理 Windows GBK 终端, 避免中文/符号乱码

```
操作流程
主菜单 → 1)开始游戏 → 选难度(1-3) → 猜数字(↑↓提示)
          → 结算 → 再来一局? → 换难度/回主菜单
          → 或直接选 2)查看最高分 / 3)重置最高分 / 0)退出
* Brewed for 44s
※ recap: You're building a guessing number game in Python with difficulty levels, high score storage, and restart menu.
The code is complete at D:\code\guess_number.py - just run 'python D:\code\guess_number.py' to play it. (disable
recaps in /config)
/cd D:\code\game
└─ Couldn't find a directory at D:\code\game.
```

```
开发Python猜数字控制台 × +
主菜单 → 1)开始游戏 → 选难度(1-3) → 猜数字(↑↓提示)
          → 结算 → 再来一局? → 换难度/回主菜单
          → 或直接选 2)查看最高分 / 3)重置最高分 / 0)退出
* Brewed for 44s
※ recap: You're building a guessing number game in Python with difficulty levels, high score storage, and restart menu.
The code is complete at D:\code\guess_number.py - just run 'python D:\code\guess_number.py' to play it. (disable
recaps in /config)
/cd D:\code\game
└─ Couldn't find a directory at D:\code\game.
/reload-skills
└─ Reloaded skills: 13 skills available (no changes)
python guess_game.py
Thought for 1s (ctrl+o to expand)
● 文件名不对—我上次创建的是 D:\code\guess_number.py, 试试这个:
python D:\code\guess_number.py
如果你想用 guess_game.py 这个名字, 我帮你拷贝一份:
● PowerShell(Copy-Item D:\code\guess_number.py D:\code\guess_game.py)
└─ (No output)
Thought for 1s (ctrl+o to expand)
● 文件名不对—我上次创建的是 D:\code\guess_number.py, 试试这个:
python D:\code\guess_number.py
如果你想用 guess_game.py 这个名字, 我帮你拷贝一份:
● PowerShell(Copy-Item D:\code\guess_number.py D:\code\guess_game.py)
└─ (No output)
```

```
python D:\code\guess_number.py

如果你想用 guess_game.py 这个名字，我帮你拷贝一份：

• PowerShell(Copy-Item D:\code\guess_number.py D:\code\guess_game.py)
  └ (No output)

Thought for 1s (ctrl+o to expand)

• 已复制为 D:\code\guess_game.py，现在两个名字都能用了：

python D:\code\guess_game.py

* Sautéed for 4s

python D:\code\guess_number.py

Thought for 2s (ctrl+o to expand)

• PowerShell(python D:\code\guess_number.py)
  └ Running in the background (↓ to manage)
  └ (timeout 30s)

Thought for 1s (ctrl+o to expand)

• 这是交互式游戏，后台运行无法接收输入。请在你自己的终端里直接运行：

python D:\code\guess_number.py

或者双击 D:\code\guess_number.py 文件（如果 Python 关联了 .py 文件）。

* Crunched for 35s · 1 shell still running

※ recap: The goal was to create a number guessing game. I've written and saved `D:\code\guess_number.py` (and a copy as `guess_game.py`). To play it, run `python D:\code\guess_game.py` in your own terminal (it needs interactive input). (disable recaps in /config)

> |

1 shell · ← for agents · ↓ to manage
```

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS C:\Users\86131> cd D:\code
PS D:\code> python guess_number.py

=====
🎮 猜数字
v1.0
=====
1) 开始游戏
2) 查看最高分
3) 重置最高分
0) 退出
=====
请选择 [0-3]: 2

-----
🏆 最高分榜
-----
简单      0
普通      0
困难      0
-----

按 Enter 返回菜单 ...

=====
🎮 猜数字
v1.0
=====
1) 开始游戏
2) 查看最高分
3) 重置最高分
0) 退出
=====
请选择 [0-3]: 1

— 难度选择 —
```

```
Windows PowerShell
3) 重置最高分
0) 退出
=====
请选择 [0-3]: 1

— 难度选择 —
1) 简单      1~50  10 次
2) 普通      1~100 7 次
3) 困难      1~200 5 次
请选择难度 [1-3] (q 返回菜单): 2

=====
难度: 普通 | 数字范围: 1~100 | 最大次数: 7
当前最高分: 0
=====

[1/7] 请输入你猜的数字: 6
↑ 小了, 再大一点

[2/7] 请输入你猜的数字: 4
↑ 小了, 再大一点

[3/7] 请输入你猜的数字: 5
↑ 小了, 再大一点

[4/7] 请输入你猜的数字: 5
↑ 小了, 再大一点

[5/7] 请输入你猜的数字: 6
↑ 小了, 再大一点

[6/7] 请输入你猜的数字: 7
↑ 小了, 再大一点

[7/7] 请输入你猜的数字: 7
↑ 小了, 再大一点

💀 次数用尽! 答案是 78

— 再来一局? —
输入 y 重新选难度, 其它键返回菜单: y
```

```
Windows PowerShell

— 再来一局? —
输入 y 重新选难度, 其它键返回菜单: y

=====
🎮 猜数字
v1.0
=====
1) 开始游戏
2) 查看最高分
3) 重置最高分
0) 退出
=====
请选择 [0-3]: 3
✅ 最高分已重置

按 Enter 返回菜单 ...

=====
🎮 猜数字
v1.0
=====
1) 开始游戏
2) 查看最高分
3) 重置最高分
0) 退出
=====
请选择 [0-3]: 1

— 难度选择 —
1) 简单      1~50  10 次
2) 普通      1~100 7 次
3) 困难      1~200 5 次
请选择难度 [1-3] (q 返回菜单): 3

=====
难度: 困难 | 数字范围: 1~200 | 最大次数: 5
当前最高分: 0
=====

[1/5] 请输入你猜的数字: 4
```

```
Windows PowerShell
— 难度选择 —
1) 简单    1~50  10 次
2) 普通    1~100 7 次
3) 困难    1~200 5 次
请选择难度 [1-3] (q 返回菜单): 3

=====
难度: 困难 | 数字范围: 1~200 | 最大次数: 5
当前最高分: 0
=====

[1/5] 请输入你猜的数字: 4
↑ 小了, 再大一点

[2/5] 请输入你猜的数字: 5
↑ 小了, 再大一点

[3/5] 请输入你猜的数字: 5
↑ 小了, 再大一点

[4/5] 请输入你猜的数字: 6
↑ 小了, 再大一点

[5/5] 请输入你猜的数字: 6
↑ 小了, 再大一点

💀 次数用尽! 答案是 29

— 再来一局? —
输入 y 重新选难度, 其它键返回菜单: y

=====
🎮 猜数字
v1.0
=====
1) 开始游戏
2) 查看最高分
3) 重置最高分
0) 退出
=====
```

```
Windows PowerShell

=====
🎮 猜数字
v1.0
=====
1) 开始游戏
2) 查看最高分
3) 重置最高分
0) 退出
=====
请选择 [0-3]: 1

— 难度选择 —
1) 简单    1~50  10 次
2) 普通    1~100 7 次
3) 困难    1~200 5 次
请选择难度 [1-3] (q 返回菜单): 3

=====
难度: 困难 | 数字范围: 1~200 | 最大次数: 5
当前最高分: 0
=====

[1/5] 请输入你猜的数字: 67
↓ 大了, 再小一点

[2/5] 请输入你猜的数字: 45
↑ 小了, 再大一点

[3/5] 请输入你猜的数字: 54
↑ 小了, 再大一点

[4/5] 请输入你猜的数字: 60
↑ 小了, 再大一点

[5/5] 请输入你猜的数字: 63
↓ 大了, 再小一点


💀 次数用尽! 答案是 62


— 再来一局? —
输入 y 重新选难度, 其它键返回菜单:
```

```
Windows PowerShell
版权所有 (C) Microsoft Corporation. 保留所有权利。

安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows

PS C:\Users\86131> cd D:\code
PS D:\code> claude
Claude Code v2.1.177

Welcome back!

Claude Code v2.1.177

Welcome back!


deepseek-chat · API Usage Billing
D:\code

Tips for getting started
Run /init to create a CLAUDE.md file with instructions for Claude

What's new
Check the Claude Code changelog for updates

在 D:\code 目录生成纯控制台 Python 贪吃蛇小游戏, 无需 Pygame 图形库, WASD 控制方向, 吃到食物加分, 撞墙/撞到自身游戏结束, 支持重新开始、退出, 代码兼容 Windows 终端中文不乱码, 生成完整 snake_game.py 文件

• Thinking for 1m 55s, reading 1 file... (ctrl+o to expand)
  File doesn't exist yet. Let me write it now.

Create file
snake_game.py

1 #!/usr/bin/env python
2 #-*- coding: utf-8 -*-
3 """
4 纯控制台贪吃蛇小游戏
5 =====
6 WASD 控制方向 | 吃到食物加分 | 撞墙/撞自身游戏结束
7 R 重新开始 | Q 退出
8 兼容 Windows 终端中文不乱码
9 """
10
11 import os
12 import sys
13 import random
14 import time
15 import msvcrt
16 import ctypes
17 from ctypes import windypes
18
19 # =====
20 # Windows Console API 封装
21 # =====
22
23 STD_OUTPUT_HANDLE = -11
24 kernel32 = ctypes.windll.kernel32
25
26
27 class _COORD(ctypes.Structure):
28     _fields_ = [("X", ctypes.c_short), ("Y", ctypes.c_short)]
29
30
31 class _SMALL_RECT(ctypes.Structure):
32     _fields_ = [
33         ("Left", ctypes.c_short),
34         ("Top", ctypes.c_short),
35         ("Right", ctypes.c_short),
36         ("Bottom", ctypes.c_short),
37     ]
38
39
40 class _CONSOLE_SCREEN_BUFFER_INFO(ctypes.Structure):
41     _fields_ = [
```

```
1 #!/usr/bin/env python
2 #-*- coding: utf-8 -*-
3 """
4 纯控制台贪吃蛇小游戏
5 =====
6 WASD 控制方向 | 吃到食物加分 | 撞墙/撞自身游戏结束
7 R 重新开始 | Q 退出
8 兼容 Windows 终端中文不乱码
9 """
10
11 import os
12 import sys
13 import random
14 import time
15 import msvcrt
16 import ctypes
17 from ctypes import windypes
18
19 # =====
20 # Windows Console API 封装
21 # =====
22
23 STD_OUTPUT_HANDLE = -11
24 kernel32 = ctypes.windll.kernel32
25
26
27 class _COORD(ctypes.Structure):
28     _fields_ = [("X", ctypes.c_short), ("Y", ctypes.c_short)]
29
30
31 class _SMALL_RECT(ctypes.Structure):
32     _fields_ = [
33         ("Left", ctypes.c_short),
34         ("Top", ctypes.c_short),
35         ("Right", ctypes.c_short),
36         ("Bottom", ctypes.c_short),
37     ]
38
39
40 class _CONSOLE_SCREEN_BUFFER_INFO(ctypes.Structure):
41     _fields_ = [
```

```
生成控制台贪吃蛇小游戏 x + v
38
39
40 class _CONSOLE_SCREEN_BUFFER_INFO(ctypes.Structure):
41     _fields_ = [
42         ("dwSize", _COORD),
43         ("dwCursorPosition", _COORD),
44         ("wAttributes", ctypes.c_ushort),
45         ("srWindow", _SMALL_RECT),
46         ("dwMaximumWindowSize", _COORD),
47     ]
48
49
50 def gotoxy(x: int, y: int):
51     """移动控制台光标到 (x, y)"""
52     kernel32.SetConsoleCursorPosition(
53         kernel32.GetStdHandle(STD_OUTPUT_HANDLE), x | (y << 16)
54     )
55
56
57 def get_console_size() -> tuple:
58     """返回控制台窗口大小 (列数, 行数)"""
59     csbi = _CONSOLE_SCREEN_BUFFER_INFO()
60     kernel32.GetConsoleScreenBufferInfo(
61         kernel32.GetStdHandle(STD_OUTPUT_HANDLE), ctypes.byref(csbi)
62     )
63     w = csbi.srWindow.Right - csbi.srWindow.Left + 1
64     h = csbi.srWindow.Bottom - csbi.srWindow.Top + 1
65     return w, h
66
67
68 def hide_cursor():
69     """隐藏控制台光标"""
70     ci = (ctypes.c_ushort * 2)(100, 0)
71     kernel32.SetConsoleCursorInfo(
72         kernel32.GetStdHandle(STD_OUTPUT_HANDLE), ctypes.byref(ci)
73     )
74
75
76 def show_cursor():
77     """恢复显示控制台光标"""
78     ci = (ctypes.c_ushort * 2)(100, 1)
79
27° 20:42 2026/6/13
```

```
生成控制台贪吃蛇小游戏 x + v
78     ci = (ctypes.c_ushort * 2)(100, 1)
79     kernel32.SetConsoleCursorInfo(
80         kernel32.GetStdHandle(STD_OUTPUT_HANDLE), ctypes.byref(ci)
81     )
82
83
84 def set_console_title(title: str):
85     """设置控制台标题"""
86     kernel32.SetConsoleTitleW(title)
87
88
89 # =====
90 # 方向常量
91 # =====
92
93 DIR_UP = (0, -1)
94 DIR_DOWN = (0, 1)
95 DIR_LEFT = (-1, 0)
96 DIR_RIGHT = (1, 0)
97
98 KEY_DIR_MAP = {
99     b"w": DIR_UP,
100    b"W": DIR_UP,
101    b"s": DIR_DOWN,
102    b"S": DIR_DOWN,
103    b"a": DIR_LEFT,
104    b"A": DIR_LEFT,
105    b"d": DIR_RIGHT,
106    b"D": DIR_RIGHT,
107 }
108
109 OPPOSITE_DIR = {
110     DIR_UP: DIR_DOWN,
111     DIR_DOWN: DIR_UP,
112     DIR_LEFT: DIR_RIGHT,
113     DIR_RIGHT: DIR_LEFT,
114 }
115
116
117 # =====
118 # 贪吃蛇游戏类
119
27° 20:42 2026/6/13
```

```
119 # -----
120
121 class SnakeGame:
122     """贪吃蛇游戏核心逻辑与渲染"""
123
124     # 渲染字符
125     CHAR_WALL = "#"
126     CHAR_HEAD = "O"
127     CHAR_BODY = "o"
128     CHAR_FOOD = "@"
129     CHAR_EMPTY = " "
130
131     def __init__(self, width: int = 20, height: int = 20):
132         self.width = width
133         self.height = height
134         self.reset()
135
136     # ----- 状态管理 -----
137
138     def reset(self):
139         """重置游戏到初始状态"""
140         cx, cy = self.width // 2, self.height // 2
141         # 蛇身坐标列表, 索引 0 为头部, 随后依次为身体、尾部
142         self.snake = [(cx - i, cy) for i in range(3)]
143         self.direction = DIR_RIGHT # 当前实际方向
144         self.next_dir = DIR_RIGHT # 输入缓冲方向
145         self.score = 0
146         self.game_over = False
147         self.won = False
148         self.speed = 0.15 # 帧间隔 (秒)
149         self.food = None
150         self._spawn_food()
151
152     def _spawn_food(self):
153         """在空白位置随机生成食物"""
154         occupied = set(self.snake)
155         empty = [
156             (x, y)
157             for x in range(self.width)
158             for y in range(self.height)
159             if (x, y) not in occupied
160         ]
161         if empty:
162             self.food = random.choice(empty)
163         else:
164             # 蛇已填满整个区域 -> 胜利
165             self.won = True
166             self.game_over = True
167
168     def change_direction(self, direction):
169         """设置下一个方向 (防止原地掉头)"""
170         if direction != OPPOSITE_DIR.get(self.direction):
171             self.next_dir = direction
172
173     # ----- 核心逻辑 -----
174
175     def step(self) -> bool:
176         """
177         推进一帧游戏逻辑。
178
179         返回 True 表示游戏继续;
180         返回 False 表示游戏结束 (撞墙 / 撞自身 / 胜利)。
181         """
182         if self.game_over:
183             return False
184
185         self.direction = self.next_dir
186
187         # 计算新头部位置
188         hx, hy = self.snake[0]
189         dx, dy = self.direction
190         new_head = (hx + dx, hy + dy)
191         nx, ny = new_head
```

```
151
152     def _spawn_food(self):
153         """在空白位置随机生成食物"""
154         occupied = set(self.snake)
155         empty = [
156             (x, y)
157             for x in range(self.width)
158             for y in range(self.height)
159             if (x, y) not in occupied
160         ]
161         if empty:
162             self.food = random.choice(empty)
163         else:
164             # 蛇已填满整个区域 -> 胜利
165             self.won = True
166             self.game_over = True
167
168     def change_direction(self, direction):
169         """设置下一个方向 (防止原地掉头)"""
170         if direction != OPPOSITE_DIR.get(self.direction):
171             self.next_dir = direction
172
173     # ----- 核心逻辑 -----
174
175     def step(self) -> bool:
176         """
177         推进一帧游戏逻辑。
178
179         返回 True 表示游戏继续;
180         返回 False 表示游戏结束 (撞墙 / 撞自身 / 胜利)。
181         """
182         if self.game_over:
183             return False
184
185         self.direction = self.next_dir
186
187         # 计算新头部位置
188         hx, hy = self.snake[0]
189         dx, dy = self.direction
190         new_head = (hx + dx, hy + dy)
191         nx, ny = new_head
```

```
182     if self.game_over:
183         return False
184
185     self.direction = self.next_dir
186
187     # 计算新头部位置
188     hx, hy = self.snake[0]
189     dx, dy = self.direction
190     new_head = (hx + dx, hy + dy)
191     nx, ny = new_head
192
193     # — 撞墙检测 —
194     if nx < 0 or nx >= self.width or ny < 0 or ny >= self.height:
195         self.game_over = True
196         return False
197
198     eating = new_head == self.food
199
200     # — 撞自身检测 —
201     if eating:
202         # 吃食物 → 尾部不动 → 不能与任何身体部分重叠
203         if new_head in self.snake:
204             self.game_over = True
205             return False
206     else:
207         # 不吃食物 → 尾部即将移走 → 允许与尾部重叠
208         if new_head in self.snake[:-1]:
209             self.game_over = True
210             return False
211
212     # — 移动 —
213     self.snake.insert(0, new_head)
214
215     if eating:
216         self.score += 1
217         self._spawn_food()
218         # 随着分数增加略微提速
219         self.speed = max(0.05, self.speed * 0.97)
220     else:
221         self.snake.pop()
222
```

```
224
225 # — 渲染 —
226
227 def render(self, ox: int, oy: int):
228     """在控制台 (ox, oy) 处绘制整个游戏画面"""
229     lines = []
230     W, H = self.CHAR_WALL, self.CHAR_HEAD, self.CHAR_BODY
231     F, E = self.CHAR_FOOD, self.CHAR_EMPTY
232
233     # 上边界
234     lines.append(W * (self.width + 2))
235
236     # 游戏区域
237     head_pos = self.snake[0]
238     body_set = set(self.snake[1:])
239     for y in range(self.height):
240         row = [W]
241         for x in range(self.width):
242             p = (x, y)
243             if p == head_pos:
244                 row.append(H)
245             elif p in body_set:
246                 row.append(self.CHAR_BODY)
247             elif p == self.food:
248                 row.append(F)
249             else:
250                 row.append(E)
251         row.append(W)
252         lines.append(" ".join(row))
253
254     # 下边界
255     lines.append(W * (self.width + 2))
256
257     # 输出
258     for i, line in enumerate(lines):
259         gotoxy(ox, oy + i)
260         sys.stdout.write(line)
261         sys.stdout.flush()
262
263 def render_status(self, ox: int, oy: int):
264     """绘制分数和操作提示"""
```

```
261     sys.stdout.flush()
262
263 def render_status(self, ox: int, oy: int):
264     """绘制分数和操作提示"""
265     gotoxy(ox, oy)
266     sys.stdout.write(f"得分: {self.score}    ")
267     gotoxy(ox, oy + 1)
268     sys.stdout.write("W 上 | S 下 | A 左 | D 右    ")
269     gotoxy(ox, oy + 2)
270     sys.stdout.write("按 R 重新开始 | Q 退出    ")
271     sys.stdout.flush()
272
273 def render_game_over(self, ox: int, oy: int):
274     """在画面中央绘制游戏结束信息"""
275     bh = self.height + 2
276     cx = ox + self.width // 2 + 1
277     cy = oy + bh // 2
278
279     if self.won:
280         title = "=== 恭喜通关! ==="
281     else:
282         title = "=== 游戏结束! ==="
283
284     score_text = f"最终得分: {self.score}"
285     hint = "按 R 重新开始 | 按 Q 退出"
286
287     # 清除可能残留的文字
288     for dy in range(-2, 4):
289         gotoxy(cx - 18, cy + dy)
290         sys.stdout.write(" " * 36)
291
292     gotoxy(cx - len(title) // 2, cy - 1)
293     sys.stdout.write(title)
294     gotoxy(cx - len(score_text) // 2, cy + 1)
295     sys.stdout.write(score_text)
296     gotoxy(cx - len(hint) // 2, cy + 2)
297     sys.stdout.write(hint)
298     sys.stdout.flush()
299
300
301 # =====
```

```
303 # =====
304
305 def main():
306     """游戏入口函数"""
307
308     # — 控制台初始化 —
309     try:
310         sys.stdout.reconfigure(encoding="utf-8")
311     except AttributeError:
312         pass # Python < 3.7 不支持 reconfigure
313
314     os.system("chcp 65001 > nul 2>&1")
315     set_console_title("贪吃蛇 Snake Game")
316
317     # 启用虚拟终端处理 (让可能用到的 ANSI 序列生效)
318     try:
319         handle = kernel32.GetStdHandle(STD_OUTPUT_HANDLE)
320         mode = wintypes.DWORD()
321         kernel32.GetConsoleMode(handle, ctypes.byref(mode))
322         kernel32.SetConsoleMode(handle, mode.value | 0x0004)
323     except Exception:
324         pass
325
326     hide_cursor()
327
328     # — 终端尺寸检查 —
329     MIN_W, MIN_H = 28, 28
330     term_w, term_h = get_console_size()
331     if term_w < MIN_W or term_h < MIN_H:
332         show_cursor()
333         print(f"控制台窗口太小 ({term_w}x{term_h}), 请调整为至少 {MIN_W}x{MIN_H}")
334         input("按 Enter 键退出...")
335         return
336
337     # — 创建游戏 —
338     game = SnakeGame(20, 20)
339
340     board_w = game.width + 2
341     board_h = game.height + 2
342     ox = max(0, (term_w - board_w) // 2)
343     oy = max(0, (term_h - board_h - 5) // 2)
```

```
349     try:
350         while running:
351             # 输入处理
352             while msvcrt.kbhit():
353                 key = msvcrt.getch()
354
355             # 方向键
356             if key in KEY_DIR_MAP:
357                 game.change_direction(KEY_DIR_MAP[key])
358
359             # 方向键 (扩展码) -> 吃掉 \xe0 前缀
360             elif key == b"\xe0":
361                 msvcrt.getch() # 丢弃第二个字节
362
363             # 重新开始 (仅在 game_over 时有效)
364             elif key in (b"r", b"R") and game.game_over:
365                 game = SnakeGame(20, 20)
366                 os.system("cls")
367
368             # 退出
369             elif key in (b"q", b"Q"):
370                 running = False
371
372             # Ctrl+C
373             elif key == b"\x03":
374                 running = False
375
376         if not running:
377             break
378
379         # 游戏逻辑步进
380         if not game.game_over:
381             game.step()
382             game.render(ox, oy)
383             game.render_status(ox, oy + board_h)
384
385         # 游戏结束 -> 等待玩家选择
386         if game.game_over:
387             game.render(ox, oy)
388             game.render_game_over(ox, oy)
389
```

```
D:\code
```

在D:\code目录生成纯控制台Python贪吃蛇小游戏，无需Pygame图形库，WASD控制方向，吃到食物加分，撞墙/撞自身游戏结束，支持重新开始、退出，代码兼容Windows终端中文不乱码，生成完整snake_game.py文件

Thought for 1m 55s, read 1 file (ctrl+o to expand)

- Write(snake_game.py)
 - └ Wrote 420 lines to snake_game.py
 - 1 #!/usr/bin/env python
 - 2 # -*- coding: utf-8 -*-
 - 3 """
 - 4 纯控制台贪吃蛇小游戏
 - 5 =====
 - 6 WASD 控制方向 | 吃到食物加分 | 撞墙/撞自身游戏结束
 - 7 R 重新开始 | Q 退出
 - 8 兼容 Windows 终端中文不乱码
 - 9 """
 - ... +410 lines (ctrl+o to expand)

Thought for 1s (ctrl+o to expand)

- 文件已生成! 让我快速验证一下语法是否正确:
- PowerShell(cd D:\code; python -c "import py_compile; py_compile.compile('snake_game.py', doraise=True); print('语法检查通过')")
 - └ ♦♦♦♦♦♦

Thought for 1s (ctrl+o to expand)

- 语法检查通过。让我再验证一下模块导入:
- PowerShell(cd D:\code; python -c "import snake_game; print('OK')")
 - └ OK

Thought for 1s (ctrl+o to expand)

- 一切正常! D:\code\snake_game.py 已生成, 可以直接运行。

```
生成控制台贪吃蛇小游戏 x +
| OK
Thought for 1s (ctrl+o to expand)
• 一切正常! D:\code\snafe_game.py 已生成, 可以直接运行。
---
如何使用
cd D:\code
python snake_game.py

游戏说明


| 功能   | 按键                    |
|------|-----------------------|
| 控制方向 | W 上 / A 左 / S 下 / D 右 |
| 重新开始 | R (游戏结束后按)            |
| 退出   | Q 或 Ctrl+C            |



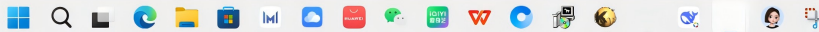
设计要点
- 纯控制台 - 无任何外部依赖 (不依赖 Pygame / colorama 等)
- Windows 原生 API - 通过 ctypes 调用 kernel32 实现光标定位、隐藏、窗口标题设置, 比 ANSI 转义码更可靠
- 中文不乱码 - 启动时自动将控制台代码页切换为 UTF-8 (chcp 65001), 所有中文文本正常显示
- 即时输入 - 使用 msvert.kbhit() 非阻塞读取键盘, 方向操控无延迟
- 防掉头 - 缓冲下一帧方向, 防止一帧内两次输入导致蛇原地掉头自杀
- 渐进加速 - 每吃一个食物帧间隔缩短 3% (最低 0.05s), 挑战性逐步提升
- 胜利判定 - 蛇身填满整个棋盘时判定通关
- 退出恢复 - finally 块确保退出时恢复光标显示和控制台状态

运行后尽量把终端窗口拉大一些 (推荐 ≥28×28), 游戏画面会自动居中显示。

* Crunched for 2m 31s

python D:\code\snafe_game.py
```

27°



20:43
2026/6/13