

Snake_game/ |—— frontend/ | |—— main_menu.py # 主界面逻辑 | |—— settings.py # 设置面板逻辑 | |—— game_ui.py # 游戏界面逻辑 |—— backend/ | |—— arduino/ | |—— hardware.ino # Arduino代码 | |—— config.h # Arduino配置文件 |—— python/ | |—— logic.py # 贪吃蛇逻辑函数 | |—— main.py # 主函数, 负责调用UI和硬件通信, 运行游戏 | |—— utils.py # 工具函数 (如音效控制、摇杆输入处理等) |—— config/ | |—— config.yaml # 设置的配置文件 |—— game_config.json # 游戏配置文件 (如难度、速度等) |—— README.md # 项目说明文档 |—— requirements.txt # Python依赖包

程序逻辑

以下是需要重点看的内容 main函数首先打开UI。点击Game start后，游戏会被分为一个一个时间周期。每个时钟周期开始时，硬件先发送摇杆数据，一旦检测到硬件发送数据，main立马通知logic干活。干完活后，logic把该给的数据给硬件，然后硬件结束工作后等待时钟周期结束。

前端

主菜单

主界面需要的外部变量有：None

设置面板界面

设置界面需要的都是静态变量：config里的内容

游戏界面

接受的动态变量：游戏情况矩阵

后端

Python

logic.py

需要接收main传来的动态变量：

- 是否按下了设置键
- 摇杆操作：上下左右 需要往外输入的动态变量：
- 一个游戏情况矩阵，game_ui和SSD1306去读
- 当前分数（用来输出到1638）

main.py

- 每个时钟周期交换软件、硬件部分传来的数据
- 向硬件、软件传输“时钟周期开始”的指令，作为一个clock
 - 先软件，再硬件

Arduino

以下是另一个开发者需要负责的内容

接收数据

接受main函数传来的数据。你需要想一下这个程序需要什么数据。 目前我给的参考数据有：

- 周期开始信号。一旦受到这个信号，程序就开始运行。
- 蜂鸣器是否需要响
- 1638在这个时钟周期的数
- 游戏情况矩阵

发送数据

传输给main函数需要的数据。目前我需要的有：

- 摇杆的数据

arduino部分的逻辑 每个时钟周期开始时，你需要等待摇杆给你传来信号，取第一个信号。“上”表示向上走，“下”表示向下走，以此类推。接到信号后，请传递一个b_rocker(bool)给main函数，这样main函数可以通知软件处理了。软件处理好后，会给main函数发送他该发的东西，然后你收到以后直接更新硬件状态，比如1638啊之类的就好。