

La relazione deve contenere:

- le principali scelte di progetto **项目的主要选择** (strutture dati principali, algoritmi fondamentali e motivazioni alla base **主要数据结构, 基本算法, 基本原理**);
- la strutturazione del codice **代码结构** (logica della divisione su pi' u file, librerie etc.);
- la struttura dei programmi sviluppati; **开发程序的结构**
- la struttura e la logica dei programmi di test; **测试程序的结构和逻辑**
- istruzioni su come compilare/eseguire ed utilizzare il codice **如何执行代码的说明**

程序由两个主要部分组成:

1. 服务器程序(server)
2. 客户端程序(client)

服务器程序(server):

1. 在启动时,服务器从命令行参数中读取服务器名称、端口号、可选的矩阵文件名、游戏时长和随机数种子。
2. 服务器生成一个 4x4 的字母矩阵,并开始监听指定的端口,等待客户端连接。
3. 当客户端连接时,服务器为每个客户端创建一个单独的线程进行处理。
4. 客户端注册时,服务器检查用户名是否唯一,并将其添加到玩家列表中。
5. 服务器向客户端发送当前的字母矩阵和剩余游戏时间。
6. 客户端可以提交单词,服务器验证单词是否合法,并根据单词长度计算得分。
7. 服务器将每个客户端的得分存储在一个共享队列中。
8. 一个额外的"评分"线程从队列中读取得分,计算最终排名并确定获胜者。
9. 服务器将最终结果发送给所有连接的客户端。
10. 在两场游戏之间,服务器进入一分钟的休息时间。
11. 服务器在收到 Ctrl+C 信号时终止。

客户端程序(client)

1. 客户端在启动时从命令行参数中读取服务器名称和端口号。
2. 客户端进入交互模式,等待用户输入命令。
3. 用户可以使用以下命令:
  - "registra utente <名称>": 向服务器注册用户。
  - "matrice": 请求服务器发送当前的字母矩阵和剩余游戏时间。
  - "p <单词>": 向服务器提交一个单词。
  - "fine": 退出游戏。
4. 客户端根据服务器的响应,向用户显示相应的信息。
5. 游戏结束时,客户端显示最终排名和得分。
6. 客户端在用户输入"fine"命令或连接意外中断时退出。

特点:

采用客户端-服务器架构,服务器负责管理游戏逻辑,客户端负责与用户交互。  
使用多线程处理多个客户端连接。

1.genera una matrice

生成的全部是小写字母，以便在文件查找











