

1 一般说明

¹我们实施了一个名为 "Boggle" 的项目，它允许一组用户玩**改进版**的在线游戏 "the Boggler"。游戏包括使用放置在 16 个字母的正方形矩阵（4x4）上的字母组成单词。每个单词必须至少由 4 个字符组成，并且只能使用矩阵中的字母组成一次。所使用的方框必须是连续的，并且可以横向或纵向组成，但在我们的版本中不能斜向组成。同一个单词不能多次通过一个方框。

A	T	L	C
I	O	Qu	A
D	V	E	S
I	S	B	I

表 1: 字符矩阵示例

¹https://it.wikipedia.org/wiki/Il_Paroliere

例如，在表 1 所示的矩阵中，除其他外，以下词语都是合法的 CASE、CASE、DRINK、VOTE，而单词：DRINK、VOTE、FLY 等则没有。*提示：要确定一个词的词性，除了要看它在生成的矩阵中是否实际存在外，还可以使用文件 dictionary.txt 中包含的字典。*

2 游戏结构

游戏采用客户端-服务器方案。客户端和服务端使用定义明确的通信协议交换信息，如下图所示。

2.1 通信协议

服务器和客户端通过 INET 套接字进行交互。客户端连接到与其交换信息的服务器。服务器在同一连接上进行回复。请求和回复都包括 3 个字段：类型、长度和日期。类型字段是一个字符（8 位），包含所发送信息的类型，可以有如下值：

```
#define MSG_OK 'K'
#define MSG_ERR
'E'
#define MSG_REGISTER_USER 'R' #define MSG_REGISTER_USER 'R'.
#define MSG_MATRIC 'M'
#define MSG_TEMPO_PARTY 'T'
#define MSG_TEMPO_ATTESA 'A'
#define MSG_PAROLA 'W'
#define MSG_PUNTS_FINAL 'F'
#define MSG_PUNTS_PAROLA 'P'
```

长度字段是一个无符号 int，表示日期字段中重要数据项的数量。如果日期字段不重要，长度字段的值为 0。日期字段是一个字符串，包含实际的回复信息。对于每条报文，只需发送报文的重要字节，而无需发送固定长度的缓冲区。

2.2 服务器

服务器流程包括以下活动：

- 启动时，服务器会生成一个 4x4 矩阵，并在作为命令行参数指定的端口上打开一个 INET 插口，然后在该端口上等待。
- 服务器必须能够随机生成数组，或者从作为命令行参数传递给服务器的输入文件中依次读取数组。该文件每行包含 16 个不同的字母，每个字母之间用

空格。唯一的例外是一对字母 Qu，它必须被视为一个字符。举例来说，将表 1 所示的矩阵视为读取下一行（以 \n 结尾）的结果。

A T L C I O Qu A D V E S I S B I

- 当收到客户端的连接请求时，服务器会生成一个专门管理该客户端的线程。该线程一直处于活动状态，直到被管理的客户端断开连接。
- 连接后，客户端注册其名称，该名称必须是唯一的。客户端使用的报文类型是 MSG_REGISTER_USER。如果该名称尚未被使用，服务器将回复 MSG_OK；如果已被使用，服务器将回复 MSG_ERR。如果该名称已被使用，客户端在收到 MSG_ERR 错误消息后，必须提出一个不同的名称。该过程重复进行，直到客户提出一个唯一的名称为止。
- 然后，客户端被添加到玩家中，玩家由服务器使用学生选择的适当数据结构进行维护。服务器会告诉客户端角色的大小（使用 MSG_MATRICE 消息）和当前游戏的剩余时间（使用 MSG_TEMPO_PARTITA 消息）或下一局游戏开始前的剩余时间（使用 MSG_TEMPO_RESTANTE 消息）。在这两种情况下，时间都以秒为单位。
- 每个客户端可以一次提出一个单词（使用 MSG_PAROLA 消息）。服务器在检查单词的正确性后，会给客户端分配一个积分，积分等于单词中的字母数（一对 "Qu" 为 1）。如果单词正确，服务器将回复 MSG_PPOINTS 消息，说明分配给该单词的分数；如果单词不正确，服务器将回复 MSG_ERR。只有当客户首次提出一个之前未由同一客户在同一游戏中提出的单词时，才会获得该分数。因此，同一客户端在同一游戏中重复出现的词语不会增加得分。在这种情况下，服务器会向客户端发送 MSG_PPOINTS 消息，表示 0 分。
- 每场比赛都有一个持续时间，由服务器启动时从命令行传递的特定参数指定。两场不同的比赛之间必须有一分钟的暂停时间。在暂停期间，服务器会响应客户端的某些请求，但不会启动新游戏。特别是，服务器会对以下信息做出积极响应（不会返回错误信息）：MSG_REGISTER_USER、MSG_MATRICE（在这种情况下仅决定 MSG_TEMPO_ATTENTION 的通信）、MSG_FINAL_POINTS。
- 游戏结束时，每个线程（管理客户端）都会向不同线程共享的队列发送一条

包含客户端得分的消息。

另一个线程（我们将其称为 "记分员"）负责从队列中检索信息，编译游戏排名，并最终选出获胜者。在进行这项工作时，计分器线程必须与其他线程同步，而其他线程在提名获胜者后，会将最终结果发送给各自的客户端。通过消息 MSG_PUNTS_FINAL，向客户端发送最终结果。消息内容为 CSV（逗号分隔值）格式，其中用户名和分数交替出现，以字符"....."结束。

\0.

- 如果新游戏开始时，任何仍处于连接状态的玩家都被视为下一局游戏的参与者。
- 如果客户端因错误而断开连接，或客户端的套接字因故关闭，则必须将玩家从当前游戏中删除，并关闭相应的线程。
- 在启动服务器的 shell 上按下 CTRL-C 时（即向服务器进程发送 SIGINT 信号），服务器就会终止。

语法

必须使用以下命令启动服务器进程：

```
> ./paroliere_srv server_name port_server_name [--matrix  
    data_filename] [--duration_in_minutes] [--seed rnd_seed] [--diz  
    dictionary] [--diz dictionary]
```

在哪里？

- paroliere_srv 是可执行文件的名称；
- server_name 是启动服务器的服务器名称（提示：您可以指定 127.0.0.1 或 localhost，在您的机器上测试该项目）
- server_port 是启动服务器进程的端口号（提示：使用大于 1024 的端口号）
- 在可选参数 --matrices 后面，是要加载矩阵的文件名（否则将使用伪计算器生成）。
- 可选的 --duration 参数允许您用分钟来表示游戏的持续时间。如果没有表示

，默认持续时间为 3 分钟。

- 通过可选的 `--seed` 参数，可以指出用于生成伪随机数的种子。

- 可选参数 `--diz` 允许您指明用于检查从客户端接收的单词词性的词典。字典文件是一个文本文件，每行显示一个单词。每行以一个 () 结束。

2.3 客户

客户流程包括以下活动：

- 启动时，客户端通过命令行从用户那里接收服务器名称和连接端口的指示。
- 客户端启动与用户的交互会话。用户可以指定一些与服务器交互的命令，以便 *注册和游戏*。游戏命令必须允许用户 *获得有关角色矩阵和提议词语的信息*。
 1. 第一步是用户注册，由客户机通过注册用户 (`register user`) 命令进行，用户名不超过 10 个字母数字字符（仅限于意大利语字母表中的数字和字符）。正如在介绍服务器时所提到的，该命令会导致客户端向服务器发送 `MSG_REGISTER_USER` 类型的信息。
 2. 如果该名称未被使用，服务器将返回 `MSG_OK` 类型的信息，否则服务器将返回 `MSG_ERR` 错误信息。只要用户没有完成注册程序，除了退出游戏外，其他所有操作都将被禁止。换句话说，只要用户没有正确注册，除结束外，调用任何其他命令都会导致错误信息。
 3. 游戏运行时，客户端可以开始提出词语。每个单词都使用 `MSG_PAROLA` 消息提出。如果单词是正确的，服务器会以 `MSG_PAROLE` 这样的消息做出回应；如果不正确，服务器会以 `MSG_ERR` 这样的消息做出回应。
 4. 要想知道游戏是否正在运行，以及运行的是哪个矩阵，客户端可以通过调用 `matrix` 命令向服务器发出请求。该命令会导致从客户端向服务器发送一条信息 `MSG_MATRICE`。如果游戏正在进行中，服务器会首先回复一条 `MSG_MATRIX` 消息，指出游戏矩阵，然后回复一条 `MSG_TIME_PLAY` 消息。如果游戏未在运行，服务器将回复一条 `MSG_TEMPO_ATTESA` 消息。
- 每局游戏结束时，服务器会通知客户端获胜者和所有玩家的得分排名。为此，服务器会向客户端发送一条信息

msg_finals_points。客户端会在屏幕上向用户显示收到的结果。正如在服务器演示中已指定的那样，信息内容采用 CSV（逗号分隔值）格式，其中用户名和分数交替出现，以字符 0 结束。

- 当用户指定结束命令时，客户端会断开与服务器的连接，会话结束。在这种情况下，用户名必须从系统中删除。即使客户端意外或主动断开连接，也必须从系统中删除用户名。

客户端命令语法

客户启动：

```
> ./parolier_cl server_name port_server
```

在哪里？

- word_cl 是可执行文件的名称；
- server_name 是要连接的服务器名称；
- port_server 是要连接的端口号；

启动客户端的结果是

[提示词作者]-->

它提供了一组与服务器交互的命令。更具体地说

help → 命令用于在屏幕上显示可用的客户端命令及其语法。该命令不会向服务器发送任何信息。

register-user username → 用于注册新用户的命令。为此，客户端向服务器发送 MSG_REGISTER_USER 类型的消息。如果给定的用户名不唯一，服务器将回复一条错误信息 MSG_ERR。否则，服务器将回复一条确认信息 MSG_OK。

matrix → 命令用于向服务器进程请求当前矩阵。如果游戏正在运行，服务器会回复一个由 16 个字母组成的矩阵列表，以及以秒为单位的剩余游戏时间。如果游戏未运

行，服务器将以秒为单位回复新游戏开始前的剩余时间。与此命令相关的信息类型是 MSG_MATRICE。

p 给定单词 → 命令用于向服务器进程提交单词。如果该词是正确的，且用户在同一场游戏中尚未提出过，服务器将以该词的指定分数作出回应。如果该单词正确，但之前已被提出，服务器将以零分回应。如果单词不正确，服务器将以错误信息回复。

end → 命令，用于退出游戏。

3 未通过足够数量的内部测试者的增补件

除上述内容外，未通过足够数量的进行中测试的学生还需要开发一些额外的功能。本节将介绍这些功能，并要求引入一些额外的信息类型。

```
#define MSG_CANCEL_USER 'D'
#define MSG_LOGIN_USER 'L'
#define MSG_POST_BACHECA 'H'
#define MSG_SHOW_BACHECA 'S'
```

- 在 断开连接的情况下，不能自动从服务器上删除用户，而必须通过客户端提供的适当命令 `delete_user` 来处理。该命令必须导致发送 `MSG_DELETE_USER` 消息。
- 必须能用以前注册但尚未删除的用户名重新登录游戏。为此，客户端必须提供 `login_user` 命令，并发送 `MSG_LOGIN_USER` 消息
- 在日志文件中记录用户注册和删除过程、每个用户发送的文字、每场比赛的得分。学生可自行决定使用何种格式。
- 在一定时间内（以分钟为单位，可通过 `--disconnect-after-time-in-minutes` 作为命令行参数指定给服务器）未与服务器通信的客户端会被弹出当前游戏，不再被视为在进行游戏。断开连接后，相关线程必须终止，玩家也必须在当前游戏中注销注册。但用户名不会从服务器上删除。

- 服务器上有一个留言板。客户端可以使用特殊的 msg 消息文本命令（会导致发送 MSG_POST_BACHECA 消息）在上面发布消息（每条消息最多 128 个字符）。留言板只存储最近发送的 8 条留言。客户端可以通过 show-msg 命令（决定发送 MSG_SHOW_BACHECA 消息）查看这些消息的内容。这些命令也可在一个游戏和下一个游戏之间的暂停期间使用。对于 MSG_POST_BACHECA 类型的报文，服务器会根据操作结果以确定（MSG_OK）或错误（MSG_ERR）类型的报文做出响应。另一方面，服务器响应 MSG_SHOW_BACHECA 类型的报文，报文内容为 CSV（逗号分隔值）格式，其中用户名和发送的单条报文交替出现。同样，要使用的数据结构和管理留言板的方法由学生自己决定。

4 报告：

项目文件包括对代码的评论和一份简短的报告（最多 8 页），其目的是描述已完成工作的整体结构。该文件中的规范不得重复。报告必须包括

- 主要设计选择（主要数据结构、基本算法和基本原理）；
- 代码结构（将逻辑划分为多个文件、库等）；
- 所制定计划的结构；
- 测试程序的结构和逻辑；
- 关于如何填写/执行和使用代码的说明 报告必须是 PDF

格式。

尾注

- 使用的数据结构以及服务器和客户端进程的并发结构由学生自行决定。
- 所有练习必须独立完成，不得有任何合作，否则将取消考试。

- 开发的代码必须使用符合 POSIX 标准的 C 语言。建议在实验室机器 `laboratorio2.di.unipi.it` 上进行测试，以确保符合 POSIX 标准。

- 为便于交付，请在考试结束后创建一个压缩文件，文件名为 `project-lab2-estate24-<NUM_MATRICOLA>.zip`（请注明您的学号，而不是 `<NUM_MATRICOLA>`）。
- 解压压缩包后，应该可以通过 `make`（默认目标）编译代码。压缩包中还应包含报告。

5 交付方式

项目必须在截止日期前通过教室上交。交稿后还需通过电子邮件发送通知给两位老师

。