

浙江大学



项目名称:	单词连连看
姓名:	朱生龙
学号:	3100101158
指导教师:	楼学庆

一、需求分析：

单词连连看是在一般的连连看的规则上添加一些规则而形成的游戏。一般的连连看要求，如果两张图片能消去，当且仅当这两张图片匹配，且可以被不超过三条折线连接，且在这条路径上没有其他图片。单词连连看与一般的连连看的区别是，两张匹配的图片不是一样的，而是汉语和英语相匹配的图片。如下图中，图 1 是两个可以消去的图片，图 2 是两个不能消去的图片。

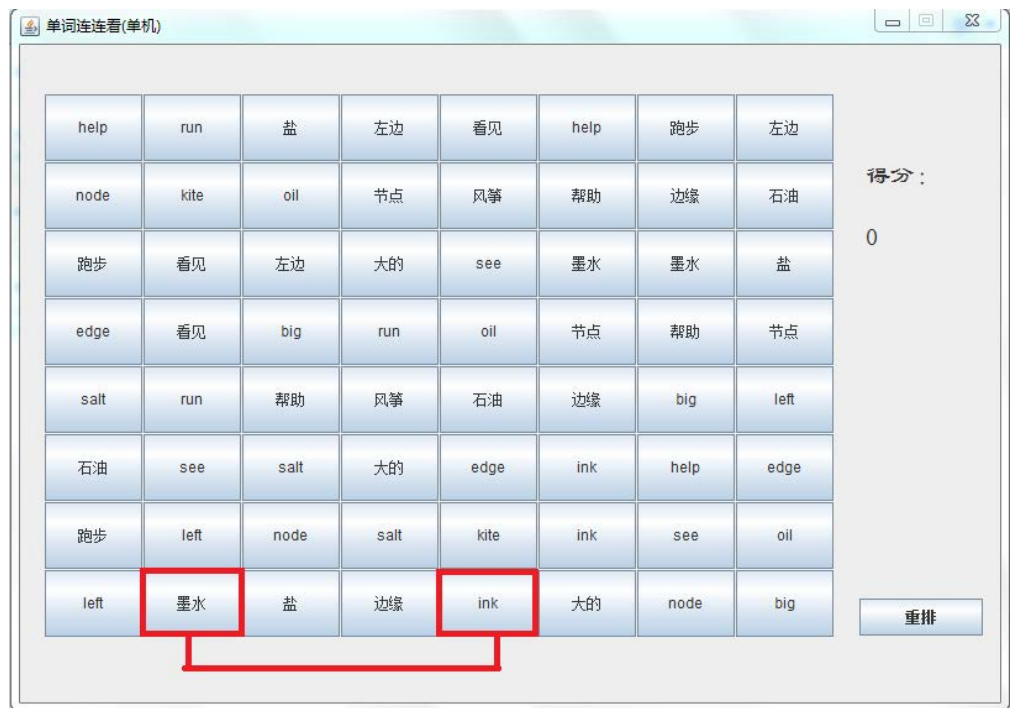


图 1：一对儿可以消去的图片

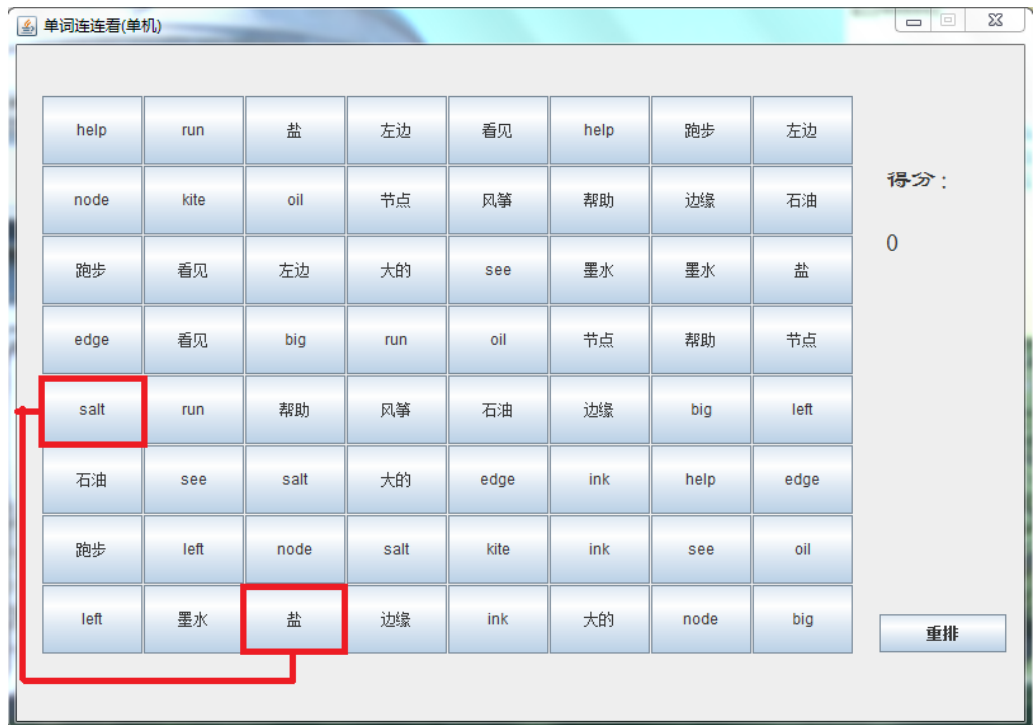
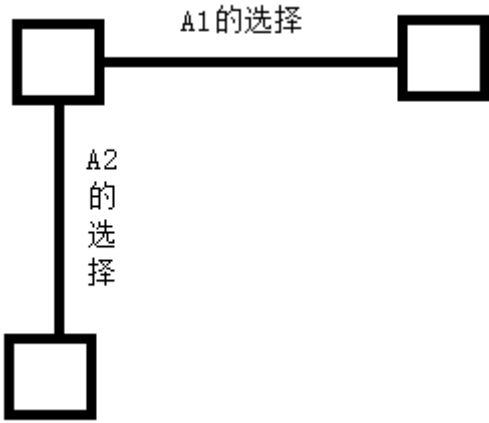


图 2 两张不能消去的图片

该游戏还分为单机版和联机版。单机版的界面如图 1 和图 2 所示，当客户端无法连接到服务器时（比如连接请求被服务器拒绝），该程序自动进入单机版。联机版为 4 人对战游戏，当有 4 个客户端连接到服务器时，开始游戏。游戏中，4 个人分为两队 A 和 B，属于 A 队的两个人控制的界面相同，属于 B 队的两个人控制的界面也相同。由于这种情况下会出现下面的情况：



其中 A1 与 A2 是 A 队的两个人，他们同时选择了不同的单词对，而且这两对单独来看都可以消去，但是最多只能消去一对，在这种情况下，就需要经过服务器进行选择，如果服务器选择了 A2 的那一对，就拒绝 A1 的选择。同时更新 A1 和 A2 的界面。

对于游戏计分规则。单机版的连连看，每消去一对单词得 2 分，每重排一次扣 5 分。没赢一局直接进入下一局，分数累加。联机版的连连看，与单机版的一样，每消去一对单词得 2 分，每重排一次扣 5 分，与单机版的区别是，联机版的要分胜负，A 队与 B 队任意一队结束，游戏均结束，先结束的队的总分加 10 分，然后与另一队的分数作比较，分数比较高的获胜。如果 A 队先结束，但是加了 10 分之后，还与 B 队的分数相等，那么就判 B 队获胜。

二、使用说明

将“服务器.jar”与“data.mdb”放在同一个目录下，“单词连连看.jar”、“data.mdb”、“ip”三个文件放到同一个目录下。

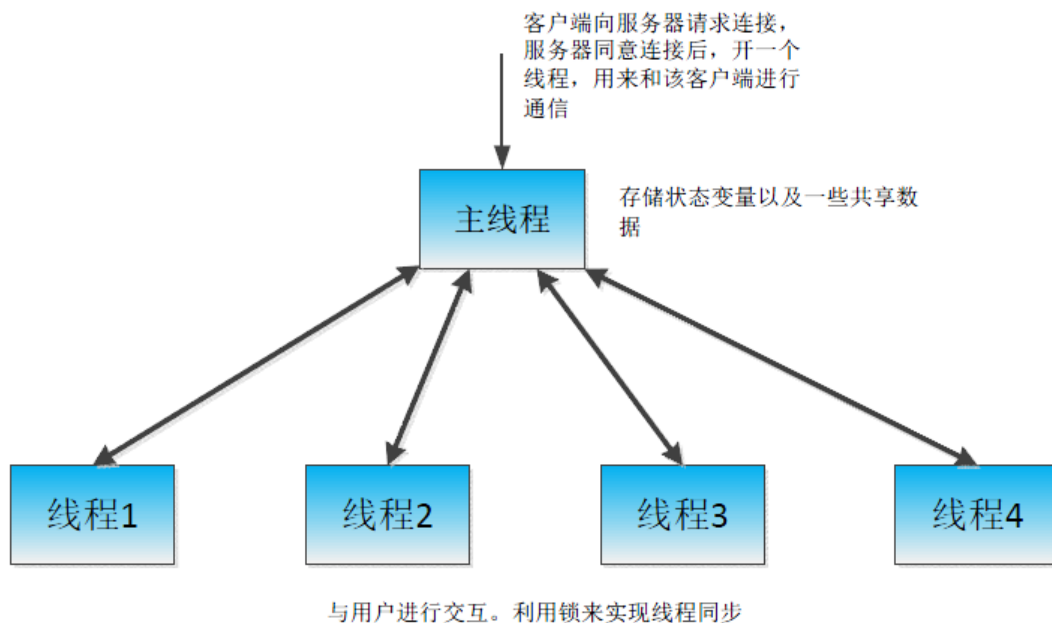
玩单机版时，直接双击“单词连连看.jar”即可。

玩联机版时，先打开“服务器.jar”，如果有人已经打开服务器，则略过此步骤。然后更改“ip”文件中的 ip 地址，格式是点分十进制。最后打开“单词连连看.jar”等待，直到有四个人连接即可开始游戏。

三、程序实现

整个程序分为两个部分：服务器和客户端。服务器与客户端的协议图（状态图）见附录。

3.1 服务器



3.1.1 主线程（class Main）:

（1）主要数据：

```
final static int totalThread = 4; //最大线程数
final static int tail = totalThread+1; //链表尾
final static int head = 0; //链表头
public OneClient[] clients;//客户端链表
public int[][] map1;//map of group1
public int[][] map2;//map of group2
public byte[] mapBytes;//send to the clients
public int state = 0;//服务器的状态
```

`clients` 是与客户端通信的线程的主要信息，是用数组模拟的双向链表。因为 4 个线程要很快地获取其在链表中的信息，所以，为了能让某个线程能够方便地获取它在链表中的数据，就自己维护了这个双向链表。`OneClient` 是一个自定义类型，用来存储一个节点的数据。

`map1` 和 `map2` 是两个队的单词连连看显示在界面上的数据。0 表示那个位置没有图片，整数和负数分别表示那个位置有英语或汉语。

`mapBytes` 是将 `map1` 或 `map2` 转换成字节数组之后，用来传递到客户端的数据。

（2）主要函数

函数原型： **public void reorder(int[][] map)**

函数功能： 将 `map` 中的数据重排。这个函数仅在“重排”按钮被单击后调用。

函数原型： **public void toBytes(int[][] map)**

函数功能： 将 `map` 中的数据转成字节数组

函数原型： **public void generateMap()**

函数功能： 生成连连看的数据

函数原型： **public static void main(String[] args)**

函数功能： 初始化数据；创建套接字，监听连接请求；当有连接请求时，创建新线程

3.1.2 子线程（class Client）

（1）主要数据

Main server;//主线程对象，用来直接获取主线程的数据。

int[][] map;//对于当前 player 的连连看主界面数据。

int n;//拐点数；客户端请求消去一对图片，向服务器发送 n，表示连接这对图片需要拐几次弯

int[] I;//要消去的图片及拐点的行号

int[] J;//要消去的图片及拐点的列号

DataStream inputFromClient;//接收数据的变量

有了 n，I，J 三个变量，服务器便可以再 $O(N)$ 的时间复杂度内判断，客户端请求的一对图片是否能消去。

（2）主要函数

函数原型： **public void parse(String str)**

函数功能： 根据附录中的协议图（状态图）解析接收到的数据

函数原型： **public void sendStart()**

函数功能： 通知所有 players 游戏开始

函数原型： **public void sendReorder()**

函数功能： 向请求重排的 player 及其队友发送重排后的结果；并更新他们的分数

函数原型： **public boolean rowHasPic(int row, int start, int over)**

函数功能： 判断某一行中的两个位置之间是否有图片，如果有，则返回 true；否则返回 false

函数原型： **public boolean colHasPic(int col, int start, int over)**

函数功能： 判断某一列中的两个位置之间是否有图片，如果有，则返回 true；否则返回 false

函数原型： **public void sendFinish()**

函数功能： 通知所有 players 游戏结束，并告诉他们胜负结果

函数原型： **public void sendPairOK()**

函数功能： 通知 player 及其队友可以消去那一对图片；并更新分数

函数原型： **public void acceptPair()**

函数功能： 可以消去一对图片后，调用 sendPairOK()或 sendFinish()

函数原型： **public void sendNoTurningPoint()**

函数功能： 如果没有拐点，则调用这个函数判断两张图片是否能消去

函数原型： **public void sendOneTurningPoint()**

函数功能： 如果只有一个拐点，则调用这个函数判断两张图片是否能消去

函数原型： **public void sendTwoTurningPoints()**

函数功能： 如果有两个拐点，则调用这个函数判断两张图片是否能消去

函数原型： **public void run()**

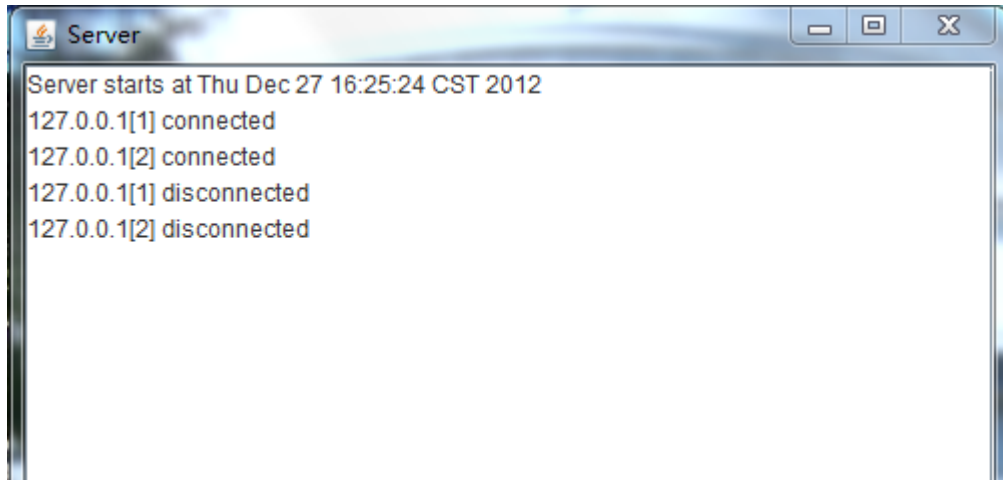
函数功能： 重写的线程函数中的 run()方法。用来实现同客户端的通信。具体见附录中的状态图。

3.1.3 数据库访问（class Access）

由于服务器对数据库的操作仅限于计算数据库中有多少组数据，而且这个类和客户端里

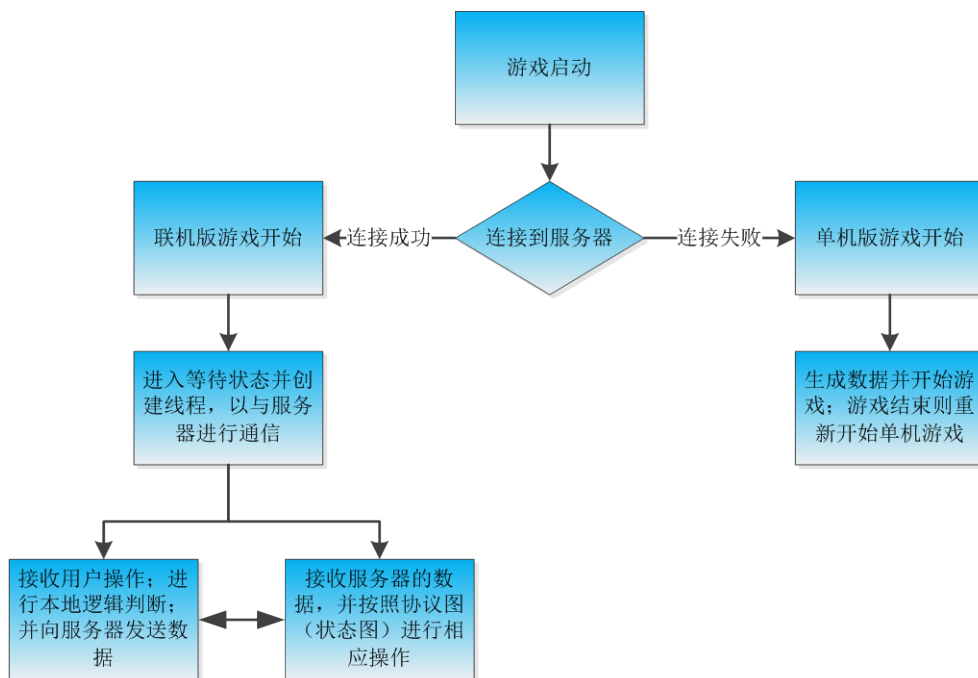
的 `class Access` 是一样的，所以这里就介绍这个类了。

3.1.4 最终的服务器



3.2 客户端

客户端分为单机版和联机版，当客户端向服务器发送的连接请求失败（服务器未开启或者被服务器拒绝）时，自动进入单机版；否则进入联机版。



3.2.1 class Main

(1) 主要数据

```
final int N = 8;//rows and cols
public Map<Button, Integer> btnMap;//显示在游戏界面的图片，为简单起见，用按钮实现
public int[][] map;//游戏界面的数据； 0 表示无障碍物。大于 0 表示是英语；小于 0 表示是汉语
public int lasti, lastj;//上次选中的图片的位置
```

```

public int cornerNum;//拐点数
public int cornerI, cornerJ, cornerII, cornerJJ;//所有的拐点
public boolean online;//true 表示是联机版； false 表示是单机版

```

(2) 主要函数

函数原型: **public void init()**

函数功能: 初始化数据, 以及连接到服务器

函数原型: **public Main()**

函数功能: 构造函数, 初始化游戏界面

函数原型: **public void generateData(int[][] data)**

函数功能: 用于单机版游戏生成数据

函数原型: **public void parseMap(byte[] bytes)**

函数功能: 解析服务器接收到的单词连连看的数据

函数原型: **public void initBtnMap()**

函数功能: 从数据库中读取数据到相应的按钮上, 从而显示在游戏界面上

函数原型: **public static void main(String[] args)**

函数功能: 开始游戏。并根据 **online** 的值来决定进入单机游戏还是联机游戏

3.2.2 class Reorder

class Main 的嵌套类。用来相应“重排”消息。



3.2.3 class MapButton

(1) 主要数据

```

public int i;
public int j;
Main tMain;

```

(i, j) 是这个按钮 (图片) 的位置, **tMain** 是 **Main** 的一个对象, 这个变量用来直接获取主类中的数据。

(2) 主要函数

函数原型: **public MapButton(Main tMain, String text)**

public MapButton(Main tMain, int i, int j, String text)

函数功能: 构造函数, 其中 **text** 是要显示到按钮 (图片) 上的内容

函数原型: **public boolean rowHasPic(int row, int start, int over)**

public boolean colHasPic(int col, int start, int over)

函数功能: 与服务器里面的那两个函数相同, 用来判断某行或某列中两张图片之间是否有图片

函数原型: `public boolean checkMatched()`

函数功能: 检查两张图片是否匹配, 如果匹配, 且是单机游戏, 则消去 (隐藏按钮); 如果匹配但是是联机游戏, 则向服务器发送数据校验请求

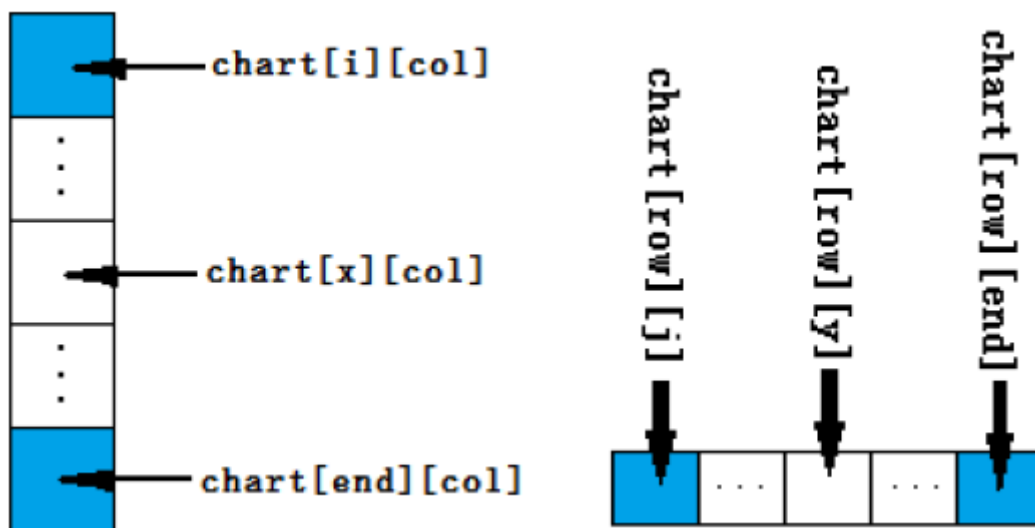
函数原型: `public void actionPerformed(ActionEvent e)`

函数功能: 用来相应该按钮被单击的事件, 判断两张图片是否匹配

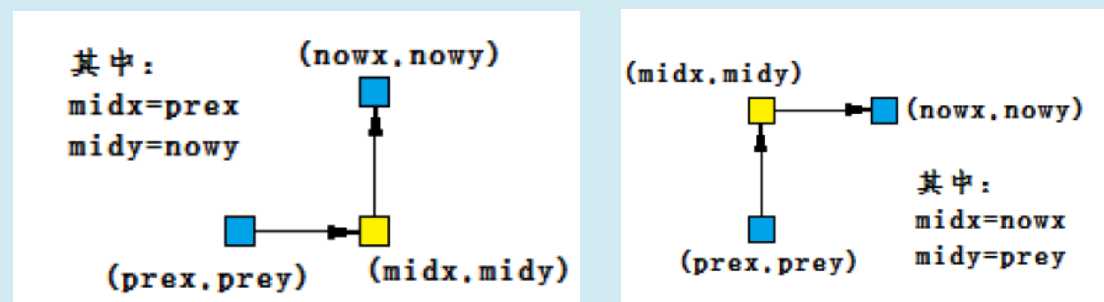
(3) 判断两张图片是否匹配的算法实现

Case 1: 如果两张图片在同一行或同一列

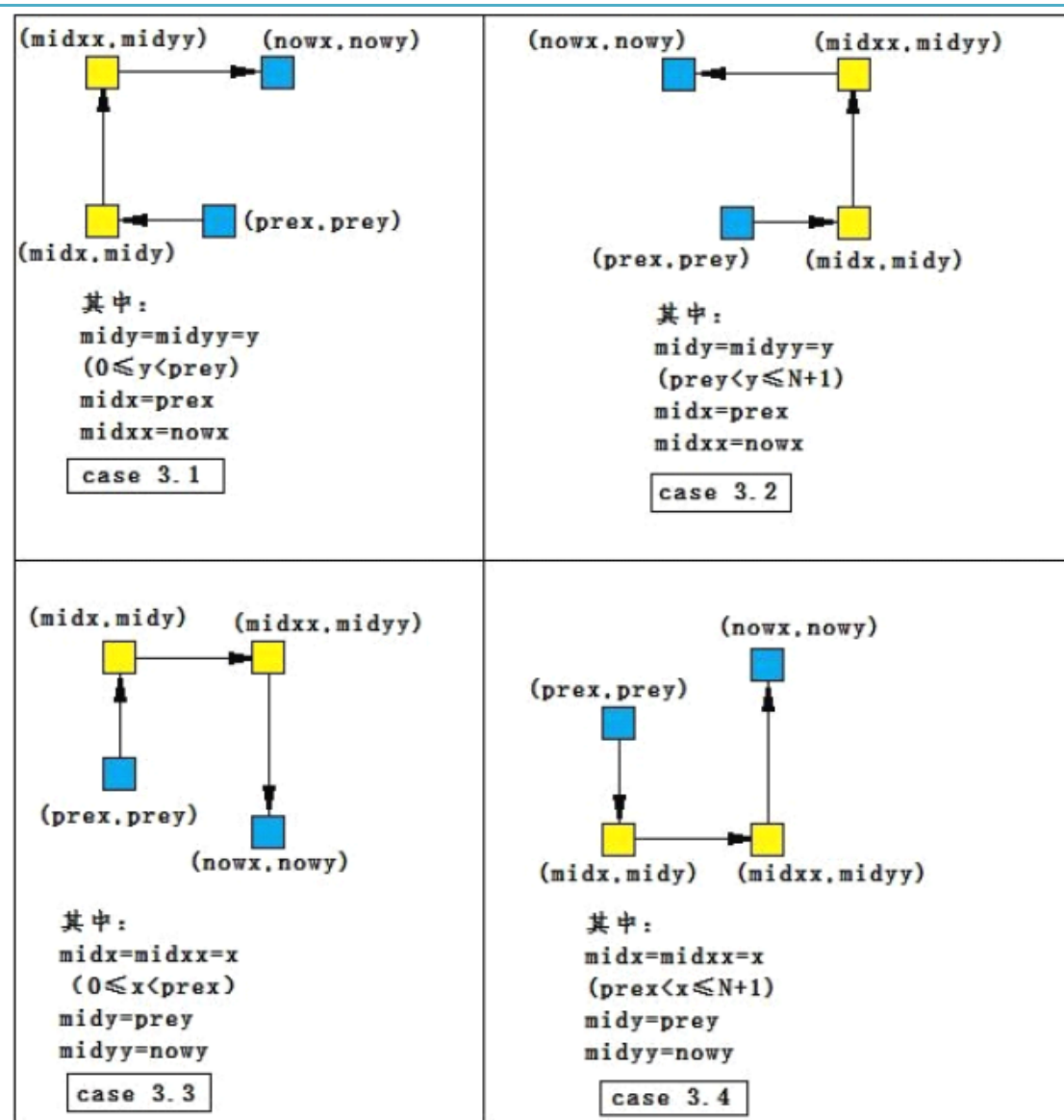
直接调用 `rowHasPic()` 或 `colHasPic` 进行判断



Case 2: 如果两张图片不在同一行也不在同一列
则判断只有一个拐点的情况



Case 3: 如果以上两种情况都没有找到可行解, 则判断有两个拐点的情况



3.2.4 class Client

(1) 主要数据

```
public Main tMain;//直接获取主类中的数据
public int state;//游戏当前状态
public int[] I;//接收到的数据解析后的行号
public int[] J;//接收到的数据解析后的列号
```

(2) 主要函数

函数原型: **public Client(Main tMain)**

函数功能: 构造函数, 初始化数据

函数原型: **public void parse(String str)**

函数功能: 解析服务器端发送的数据

函数原型: **public void run()**

函数功能: 线程的运行函数, 接收到服务器数据后, 根据当前客户端所处的状态执行相应的动作

3.2.5 class Access

(1) 主要数据

String url;//本地数据库的路径

Connection conn;//连接数据库后获得的句柄

Statement stmt;//SQL 查询语句

(2) 主要函数

函数原型: **public Access()**

函数功能: 初始化数据, 并连接数据库

函数原型: **public String getItem(int num)**

函数功能: 从数据库中取出一个英语或汉语

函数原型: **public void closeAccess()**

函数功能: 关闭数据库

函数原型: **public int getTotal()**

函数功能: 获取数据库中数据的个数

3.2.6 最终的客户端

单机版



单词连连看(单机)

trip	旅行	bag	hour	包	地带	zone	坦克
唱歌	包	院子	bag	案例	小时	sing	help
坦克	help	年	旅行	年	脖子	旅行	院子
zone	案例	小时	neck	bag	year	yard	trip
帮助	hour	帮助	neck	case	唱歌	案例	year
case	脖子	院子	tank	包	year	坦克	case
tank	年	脖子	唱歌	帮助	sing	sing	help
hour	地带	tank	yard	yard	neck	trip	小时

得分：

0

重排

单词连连看(单机)

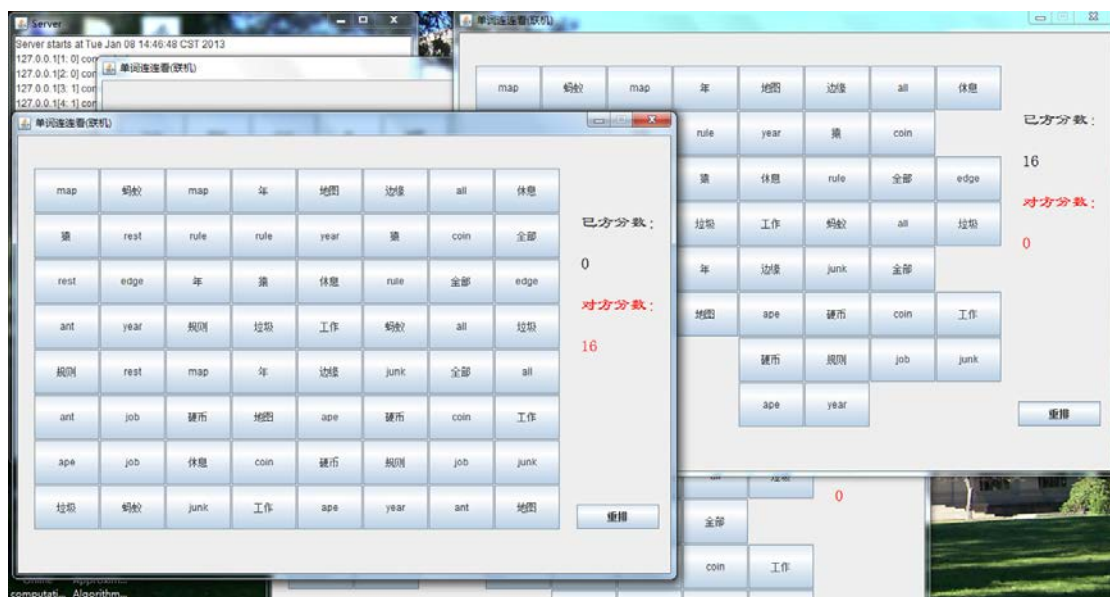
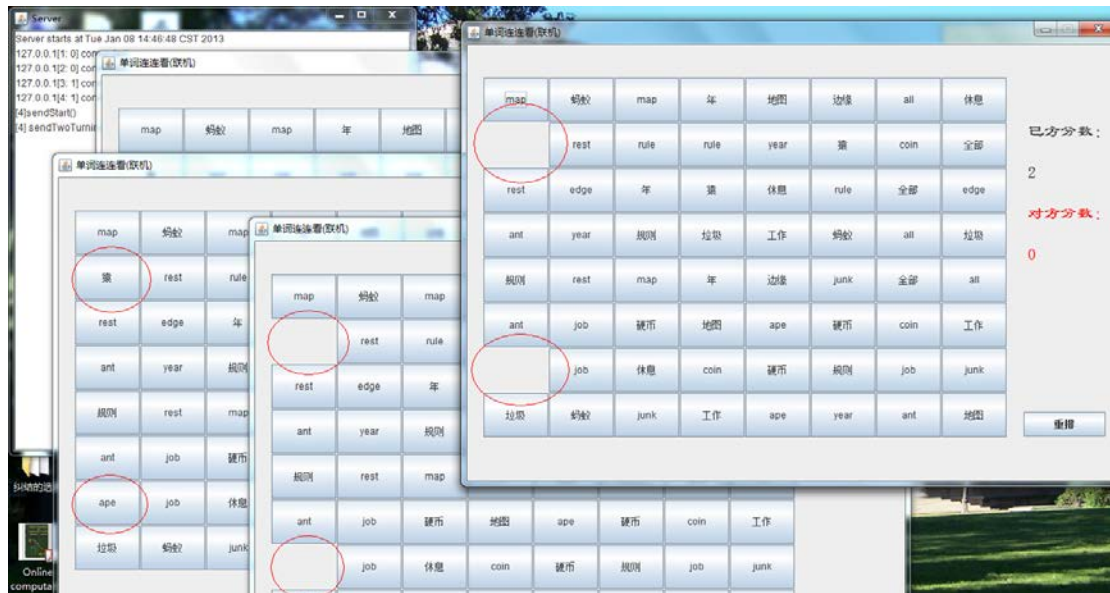
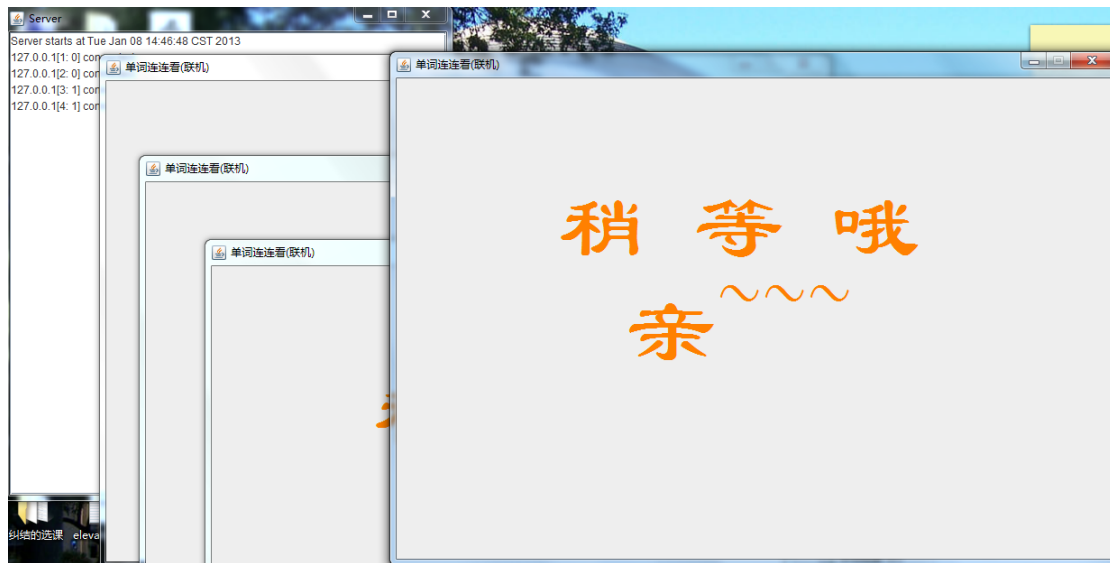
							坦克
唱歌		院子		案例		sing	help
		年	旅行	年	脖子	旅行	院子
			neck	bag	year	yard	trip
帮助				case	唱歌	案例	year
		院子	tank	包	year	坦克	case
	年	脖子	唱歌	帮助	sing	sing	help
		tank	yard	yard	neck	trip	

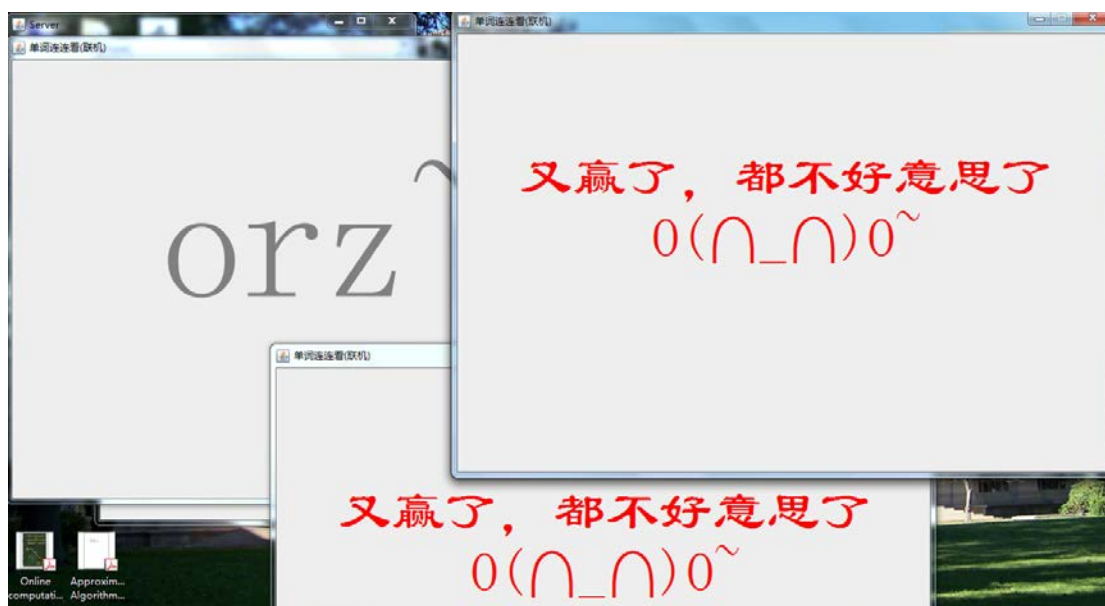
得分：

24

重排

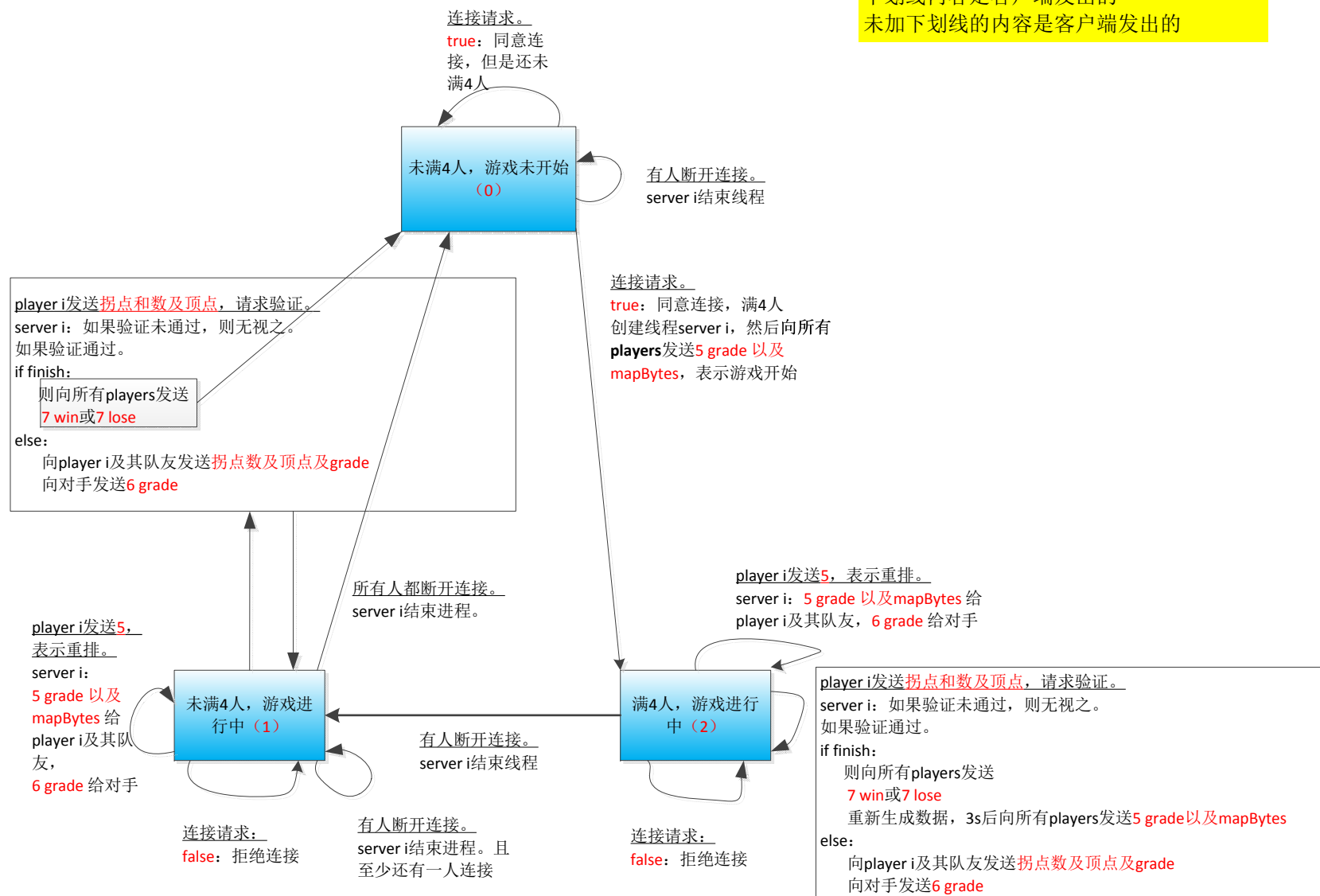
联机版：





服务器 (如果player i发消息, 且没

有断开，服务器不会改变状态)



符号说明：
 红色字均表示传输的数据
 下划线内容是客户端发出的
 未加下划线的内容是客户端发出的

客户端 (有可能主动发送

消息，也有可能先接收消息，
然后做出相应状态转移)

