Time Bridge 设计文档1.0

黄国正

概论：

Time Bridge 是一款面向桥牌初学者，主打时间回溯等教学功能的桥牌软件，相信能给桥牌新用户带来舒适快捷的桥牌入门及训练服务。

Time Bridge 1.0 基本功能接口及界面设计

牌局界面：

电脑玩家2（North）

13张牌显示（显示手牌数）

玩家2提示框：

显示“pass”或叫牌类型，仅在叫牌时显示

当前出牌

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1♧ | 1◇ | 1♡ | 1♠ | 1NT |
| 2♧ | 2◇ | 2♡ | 2♠ | 2NT |
| 3♧ | 3◇ | 3♡ | 3♠ | 3NT |
| 4♧ | 4◇ | 4♡ | 4♠ | 4NT  当前出牌 |
| 5♧ | 5◇ | 5♡ | 5♠ | 5NT |
| 6♧ | 6◇ | 6♡ | 6♠ | 6NT |
| 7♧ | 7◇ | 7♡ | 7♠ | 7NT |
| (如果要叫牌，点击相应位置，系统自动将之前位置的区域变灰并设定为无效。如果要Pass,点击Pass)  Pass | | | | |

电脑玩家1

(East)

电脑玩家3

(West)

玩家1提示框

玩家3提示框

当前出牌

当前出牌

用户提示框

用户：(South)

13张牌显示

注：此叫牌区域在开局后变为：

契约： 由 叫出

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 轮次 | N出牌 | W出牌 | S出牌 | E出牌 | 获胜方 |  |
| 1 |  |  |  |  |  | 回溯 |
| 2 |  |  |  |  |  | 回溯 |
| 3 |  |  |  |  |  | 回溯 |
| 4 |  |  |  |  |  | 回溯 |
| 5 |  |  |  |  |  | 回溯 |
| 6 |  |  |  |  |  | 回溯 |
| 7 |  |  |  |  |  | 回溯 |
| 8 |  |  |  |  |  | 回溯 |
| 9 |  |  |  |  |  | 回溯 |
| 10 |  |  |  |  |  | 回溯 |
| 11 |  |  |  |  |  | 回溯 |
| 12 |  |  |  |  |  | 回溯 |
| 13 |  |  |  |  |  | 回溯 |
| 总胜场 |  |  |  |  |  |  |
| 总分（该行仅在13轮结束后显示） |  |  |  |  |  | 复盘  新的一 局 |

用户可以在进行游戏时点击任意一行右侧的回溯按钮进行回溯。回溯后所有状态将恢复到该轮次的上一轮的状态。

13轮均结束后，总分表格显示，玩家可以点击复盘（游戏以同一发牌重新开始）或新的一局开始新的一轮游戏

基本功能接口：

全局变量Now表示当前回合玩家

牌类：Card

//int type 表示花色

//int number 表示大小

玩家类：User

//编号

int Number 记录玩家位置

//牌组数组

梅花Club[14];

方块Diamond[14];

红心Heart[14];

黑桃Spade[14];

//是否定约

玩家一方定约后，修改玩家状态为定约，玩家代替队友出牌（队友的Number改为玩家的Number）

//每次操作都必须判断是否是当前回合。（Now==Number）

//叫牌阶段函数（仅当前叫牌阶段可用）

点击相应叫牌区域后，修改用户状态栏的叫牌信息

Call(鼠标位置);

Pass();选择pass时执行

//出牌阶段函数（仅当前出牌阶段可用）

//突出牌 当玩家鼠标指向自己（或己方明手）的一张牌时，将该牌从牌组里突出，并将该牌状态改为highlight；鼠标移走后highlight状态清零。

Highlight(鼠标位置);

//出牌 Highlight后的牌被点击，判断是否可出后打出

Play(鼠标位置){执行highlight判断函数；执行可出判断函数；打出};

//回溯

牌局中每一轮结束后，玩家均可点击中央回溯按钮进行回溯操作。所有状态回到回溯的上一回合

Return (turn a);回溯第a轮(回到第a轮开始状态)

//复盘

整局结束后，玩家可点击复盘按钮回到发牌后未叫牌的阶段

Reset();

//新的一局

所有信息清零，新的牌局函数执行

New();

AI:玩家子类

//选择当前叫牌

ChooseCall(); 自动执行叫牌函数并叫牌

//选择当前出牌

ChoosePlay(); 自动执行出牌函数，打出该牌

牌局绘制：

//绘制背景

Draw (background.jpg);

//发牌

Distribute();

//进行完发牌函数后，每个玩家的牌组数组更新

//绘制牌

Draw (User a);

将User a 的所有牌组数组依顺序绘制出来

//绘制叫牌表格;

Calltable.Draw();

//每次有新的叫牌输入，重新绘制表格，重新绘制用户叫牌信息

Calltable.Draw();

Draw(User.info);

//叫牌结束后，绘制出牌表格，覆盖叫牌表格

Playtable.Draw();

//每次有用户出牌，刷新当前出牌信息；刷新每个玩家手牌

Draw(User.card)

Draw(User.hand);

//每轮结束后，刷新出牌表格

Playtable.Draw();

//回溯或复盘时，刷新手牌和出牌表格

Draw(User.hand);

Playtable.Draw();

//13轮结束后，计算总分并绘制表格

Playtable.Draw();

//新一轮游戏全部清零

New();

Calltable类

//Draw() 绘制叫牌表格并显示信息

//int progress 表示当前叫牌进度。低于进度的叫牌无效

//Pass() 向牌局报告pass操作

Playtable类

//Draw() 绘制出牌表格并显示信息

注：AI、牌局、Gui为实现功能可添加自己的新函数（比大小等）和新属性。最后要以批注形式反馈到设计表格中。

接口共用的函数在添加前必须先向其他分支声明公共命名和方法。