目录

[一、 引言 2](#_Toc74777706)

[1.1标识 2](#_Toc74777707)

[1.2系统概述 2](#_Toc74777708)

[1.3文档概述 2](#_Toc74777709)

[1.4功能需求 3](#_Toc74777710)

[1.5基线 3](#_Toc74777711)

[二、引用文件 3](#_Toc74777712)

[三、详细系统设计 3](#_Toc74777713)

[四、数据描述 4](#_Toc74777714)

[五、程序设计说明 4](#_Toc74777715)

[5.1 程序描述 5](#_Toc74777716)

[5.2 功能 6](#_Toc74777717)

[5.3 性能 6](#_Toc74777718)

[5.4输入项 6](#_Toc74777719)

[5.5 输出项 6](#_Toc74777720)

[5.6 算法 7](#_Toc74777721)

[5.7 流程逻辑 7](#_Toc74777722)

[六、接口 7](#_Toc74777723)

[七、总结 8](#_Toc74777724)

斗地主系统详细设计报告

# 引言

网上斗地主系统是为了满足广大玩家的游戏需求，能够打破传统斗地主形式的空间限制，满足玩家在不同地理位置能够进行斗地主游戏的需求，实现邀请玩家进入房间以及踢出玩家的功能，并对玩家的的个人信息包括胜负次数，积分等一些信息的维护。

## 1.1标识

本系统只适用于windows操作系统安装及使用。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文件状态：  [ ]草稿  [ √ ]正式发布  [ ]正在修改 | 报告编号 | 斗地主系统详细说明书 | | |
| 当前版本 | V1.0 | | |
| 编写人 | pog小队 | 编写日期 | 2021-6-10 |
| 审批人 |  | 审批日期 |  |
| 保密级别 | 公开 | | |

## 1.2系统概述

系统希望通过网络，为传统的斗地主游戏打造一个新的网络平台。本系统希望通过计算机平台对游戏的操作环节进行管理，为玩家打造自动化操作管理的同时保证游戏的公平性，使得玩家可以在游戏娱乐时不必对发牌、抢地主、积分计算等操作负责。同时通过网络平台的形式使得传统斗地主的地域限制被打破，玩家可随时随地满足自己的精神娱乐需求。

编写斗地主系统详细设计说明书的目的在于，从详细设计的角度明确斗地主系统项目的业务品种、功能范围，使系统开发人员和产品管理人员明确产品功能，可以有针对性的进行系统开发、测试、验收等各方面的工作。

读者：项目经理、概要设计人员，编码人员，测试人员。

## 1.3文档概述

斗地主系统由本机对本地玩家的信息进行存储，个人信息包括胜负场次，积分等一些信息，该信息可提供其他同玩玩家进行查看。实现了邀请玩家进入房间与踢出玩家的功能。比较真实的还原了现实打牌，提供叫地主，抢地主的功能，比较真实的实现了现实打牌中的规则。实现账户切换的功能，允许一台机子存在多个账户，在登录状态下可以进行账户的切换。实现玩家之间的信息交流服务，该信息交流主要限制于固定的话语交流与表情包，该功能具有一定的限制性，但一方面增加了游戏的趣味性。游戏平台提供一定简单的玩家界面，提供一定的视觉效果与牌游环境氛围。主要提供用户之间的对战，对于人机对战根据实际情况进行实现，该功能根据每一阶段的进行状态以决定是否要提供。

## 1.4功能需求

(1)、用户登录

(2)、用户信息维护

(3)、斗地主游戏

(4)、用户交流

## 1.5基线

1、斗地主系统可行性分析报告

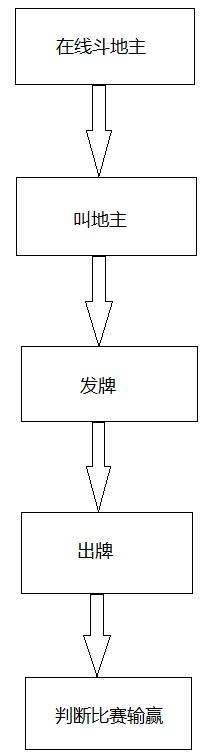
2、斗地主系统需求分析报告

# 二、引用文件

1、中华人民共和国国家标准GB T-8567-2006

# 三、详细系统设计

在线斗地主有运行游戏的一个功能



运行游戏：

（1）叫地主：初始是玩家自己来叫分，然后按照一个方向继续下去，每个人根据自己的分数来叫地主，如果一个玩家叫了3分那么他就是地主，直接开始发牌。如果一个玩家不叫地主就顺着下一个人继续。如果一个人叫了小于3分大于0分的分数，那么下一个人可以继续叫分，但是后续这个玩家叫分的最小值要比上一个人叫的分要大，当然，顺延的时候允许不叫分，这样会继续给下一个人叫分，如果大家都叫完了那么就看分数情况，有叫分的，分数大的是地主，没有一个人叫分那么本次废弃，重新开始叫分。

（2）发牌：洗牌之后，随机抽出三张牌作为地主牌并展示给全体玩家，剩余的牌均匀的分给三个人，最后根据谁是地主来把这三张牌给地主，主界面把该人表示为地主，其余两人表示为农民。

（3）打牌：根据选择的地主确定地主先出牌，每个玩家选择牌（可以一张或者多张）之后点击出牌进行出牌，如果不选择那么只能选择不出牌，之后根据斗地主的规则进行打牌，规矩是牌型要一致，后来者牌面要比前面的大，如果是炸弹那么就可以抵消掉前面的任何牌型，此外后来者出牌需要出比炸弹牌面要大的牌，这里牌面指的是排的字符而不是花色，从小到大是3、4、5、6、7、8、9、10、J、Q、K、A、2、小花、大花，如果牌型符合要求就出牌并给下一个人出牌，否则本次操作无效，重新进行操作。当然每个玩家轮到自己出牌的时候可以选择不出牌，此时把出牌权利给下一个人，如果另外两个人都不出牌那么本次出牌结束，由出牌者重新出牌并继续游戏，牌型也要重置为新出牌者出的牌。

（4）判断输赢：游戏中任何一个玩家出完牌的时刻都要判断是不是出完牌，当一个玩家没有牌那么游戏终止，如果是地主没有牌那么地主赢，否则是农民赢，根据输赢来计算得分，输家减少分数，赢家获得分数。

# 四、数据描述

1、玩家信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 字段名 | 类型 | 宽度 |
| 玩家标志ID | PlayerID | Int | Not null |
| 玩家当前分数 | Score | Int | Not null |
| 玩家身份 | Identity | Nvarchar(40) | Not null |
| 玩家卡牌 | Cards | Int(40) | Not null |

2、卡牌信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 字段名 | 类型 | 宽度 |
| 卡牌ID | CardID | Int | Not null |
| 卡牌类型 | Type | Int | Not null |

# 五、程序设计说明

逐个地给出各个层次中的每个程序的设计考虑。以下给出的提纲是针对一般情况的。对于一个具体的模块，尤其是层次比较低的模块或子程序，其很多条目的内容往往与它所隶属的上一层模块的对应条目的内容相同，在这种情况下，只要简单地说明这一点即可。

## 5.1 程序描述

根据玩家的需要显示游戏初始界面，玩家可以选择地主或者不叫。然后开始游戏内容。玩家在自己的回合将可以选择牌，以及是否可以出牌。在电脑的回合将会进行计算模拟AI输出自己的决策。最后出完牌的人获得胜利。出牌规则参考斗地主。

(1) Cards/CardsGroup 这两个部分负责的是对卡牌进行对象化建模，实现一个卡牌对象。用于描述一组扑克。

(2) Game负责整个游戏的逻辑部分，用来管理游戏当前状态，发牌等行为。

(3) Player 负责的是出牌的合法性检测，以及AI逻辑实现。当player不是真人的时候负责执行AI逻辑。

(4) Scene 用来和主窗口进行交互，负责UI场景的渲染。

(5) Winmain则是主程序，创建窗口管理。

这里放入一部分代码进行一些举例。

void Game::SendCard(void)

{

while (cardheap.GetRemain() > 3){

player[0]->AddCard(cardheap.GetCard());

player[1]->AddCard(cardheap.GetCard());

player[2]->AddCard(cardheap.GetCard());

}

for (int i = 0; i < 3; ++i)

landlordcard[i] = cardheap.GetCard();

}

这一部分代码将会给每个玩家进行发牌，获取牌后将其添加到玩家的手牌中去。

void Scene::DrawHumanCards(HDC hdc, int highlight)

{

PatBlt(hdcHumanCards, 0, 0, sceneSize.cx, sceneSize.cy, PATCOPY);

int i = 0, y1,

c = game->player[0]->cards.size(),

x = (650 - cardSize.cx - 22 \* c + 22) / 2,

y = 128 - cardSize.cy - 15;

for (auto rb = game->player[0]->cards.rbegin(); rb != game->player[0]->cards.rend(); ++rb){//显示玩家手牌

if (game->player[0]->selection.cards.find(\*rb) !=

game->player[0]->selection.cards.end())//将已选择的牌上移一段，突出显示

y1 = y - 15;

else

y1 = y;

TransparentBlt(hdcHumanCards, x + 22 \* i, y1, cardSize.cx, cardSize.cy, hdcCards,

cardSize.cx \* \*rb, 0, cardSize.cx, cardSize.cy, RGB(0, 0, 255));

//if (mem == highlight){

// HPEN pen = CreatePen(PS\_SOLID, 3, RGB(0, 255, 255));

// HPEN oldpen = (HPEN)SelectObject(hdcScene, pen);

// SelectObject(hdcScene, GetStockObject(NULL\_BRUSH));

// Rectangle(hdcScene, x + 22 \* i, y1, x + 22 \* i+cardSize.cx, y1+cardSize.cy);

// SelectObject(hdcScene, oldpen);

// DeleteObject(pen);

// }

++i;

}

BitBlt(hdc, 100, 412, 650, 128, hdcHumanCards, 0, 0, SRCCOPY);

}

这一部分代码将会使用GDI绘制玩家的手牌，根据玩家的手牌信息以此将图片呈现上去。

## 5.2 功能

## 5.3 性能

使用WIN32 API提供的GDI进行UI渲染，使用C++作为逻辑开发。能够在系统的支持下实现硬件渲染。

## 5.4输入项

用户的鼠标操作，跟随着游戏界面进行输入。首先是输入叫分或者不叫。然后是选择自己要的出的卡牌，然后再选择出牌。

- 鼠标输入

- 按钮输入

## 5.5 输出项

对于输出项来说，当前游戏的场景就是输出项。对方出的牌，玩家的手牌等信息为输出项。这是一个整个全局状态(手牌，对方手牌，打的牌等)。

- 玩家手牌：渲染输出

- 对方手牌：固定样式输出，避免看到其他人的牌。

- 打出去的牌

## 5.6 算法

无

## 5.7 流程逻辑

开始

打开斗地主客户端

主界面

游戏开始

是否所有玩家都不叫

顺时针下一位玩家叫地主

叫地主并进行叫分

下发地主牌

从地主开始顺时针出牌

是否所有的玩家出完牌

确定获胜者

结束

# 六、接口

1、用户和管理员信息接口图

用户信息

管理员信息

管理

2、玩家和卡牌信息接口图

玩家

卡牌

拥有

# 七、总结

网上斗地主系统能够满足广大玩家的游戏需求，能够打破传统斗地主形式的空间限制，满足玩家在不同地理位置能够进行斗地主游戏的需求。

系统希望通过网络，为传统的斗地主游戏打造一个新的网络平台。本系统希望通过计算机平台对游戏的操作环节进行管理，为玩家打造自动化操作管理的同时保证游戏的公平性，使得玩家可以在游戏娱乐时不必对发牌、抢地主、积分计算等操作负责。同时通过网络平台的形式使得传统斗地主的地域限制被打破，玩家可随时随地满足自己的精神娱乐需求。

斗地主系统是一个比较大的系统，涉及到玩家对战和信息交流等等，在这次设计中，小组完成了信息管理、游戏逻辑等功能，在各方面的技术上更进一步。