

**面向对象程序设计**

**中期报告**

**学生姓名： 沈韵沨 学号： 3200104392**

**学生姓名： 王柯棣 学号： 3200105119**

**学生姓名： 魏鼎坤 学号： 3200105652**

**学生姓名： 赵伊蕾 学号： 3200104866**

**2021-2022 春夏学期 2022年05月**

**目录**

[1 IDE、编译器与库 3](#_Toc104129549)

[1.1 IDE介绍 3](#_Toc104129550)

[1.2 编译器介绍 4](#_Toc104129551)

[1.3 库与函数 4](#_Toc104129552)

[1.3.1 库 4](#_Toc104129553)

[1.3.2 函数 5](#_Toc104129554)

[2 项目功能与目标 6](#_Toc104129555)

[2.1 基本功能 6](#_Toc104129556)

[2.1.1 实现美观、交互友好的图形化用户界面 6](#_Toc104129557)

[2.1.2 实现21点与德州扑克两种游戏模式 6](#_Toc104129558)

[2.1.3 支持4名玩家单机参与游戏 6](#_Toc104129559)

[2.2 扩展功能 7](#_Toc104129560)

[2.2.1 支持少于4名玩家参与游戏 7](#_Toc104129561)

[2.2.2 支持多玩家通过局域网联机参与游戏 7](#_Toc104129562)

[2.2.3 实现玩家历史游戏数据的存档与读取 7](#_Toc104129563)

[2.2.4 21点模式：实现对特殊牌型的判定与注码结算 7](#_Toc104129564)

[2.2.5 德州扑克模式：彩池限注与无限注模式 7](#_Toc104129565)

[3 程序设计 8](#_Toc104129566)

[3.1 页面架构 8](#_Toc104129567)

[3.1.1 标题页 8](#_Toc104129568)

[3.1.2 游戏选择页 9](#_Toc104129569)

[3.1.3 21点游戏页 10](#_Toc104129570)

[3.1.4 德州扑克游戏页 11](#_Toc104129571)

[3.2 类的架构 12](#_Toc104129572)

[3.2.1 主窗口 12](#_Toc104129573)

[3.2.2 21点 13](#_Toc104129574)

[3.2.3 德州扑克 14](#_Toc104129575)

[4 项目进度与后续计划 16](#_Toc104129576)

[4.1 项目进度 16](#_Toc104129577)

[4.1.1 总体进度 16](#_Toc104129578)

[4.1.2 前端进度 16](#_Toc104129579)

[4.1.3 后端进度 18](#_Toc104129580)

[4.2 后续计划 18](#_Toc104129581)

[5 成员分工 19](#_Toc104129582)

[附录 20](#_Toc104129583)

# IDE、编译器与库

## IDE介绍

由于本次程序采用Qt框架，经讨论，本组决定选用Qt Creator作为本次Project的IDE，以下是本组选用该IDE的几点理由：

* 支持跨平台开发

作为跨平台的集成开发环境，Qt Creator支持在Windows、Linux和macOS桌面操作系统上运行并创建桌面应用程序，满足了本组成员在Windows与macOS环境下便利的共同开发程序的需求。

* 便利的项目与版本管理

Qt Creator本身集成了大多数现有流行的版本控制系统，包括Git、Subversion、Perforce和Mercurial，便于小组成员通过远程仓库进行协作开发。

* 集成的UI设计

Qt Creator集成了可用于C++小部件构建的或基于现有控件实现动画UI的可视化编辑器。“所见即所得”的开发方式大大便利了游戏界面的开发与调试过程。其中Qt Designer支持以命令性方式，使用带有传统C++ Qt API的屏幕表单快速设计和构建小部件与对话框，为游戏界面提供了更高的灵活性。

* 强大的辅助功能

Qt Creator对于C++语言具有完整的表达式检查、自动补全、指示行间错误及警告、上下文关联、告知函数声明及其调用位置等功能。此外还支持对变量智能化重命名，如只改变分属不同域中的多个同名变量之一，但不影响其余变量。

* 轻松的调试与诊断

我们可以通过Qt Creator提供的时间线和火焰图，快速浏览CPU和内存的使用情况，找出性能瓶颈并作针对性优化。其本身支持通GNU 符号调试器 ( gdb )、 Microsoft 控制台调试器 ( CDB )与内部 Java 脚本调试器对程序进行调试，可以设置断点查看调用堆栈以及检查局部变量和表达式，使得调试过程简单便捷。

## 编译器介绍

本项目使用MinGW作为C/C++编译器。

MinGW 提供了一套简单方便的Windows下的基于GCC 程序开发环境，收集了一系列免费Windows使用的头文件和库文件；同时整合了 GNU工具集，如gcc，g++，make等。它在Windows平台上模拟了Linux下GCC的开发环境，为C++的跨平台开发提供了良好基础支持。

## 库与函数

### 库

* <QLabel> —— 对Qt提供的标签类控件进行操作
* <QString> —— 处理从输入框得到的字符串
* <QMetaEnum> —— 应用Qt内的枚举类
* <QPushButton> —— 对Qt提供的按钮类控件进行操作
* <QMainWindow> —— 对Qt提供的窗口类控件进行操作

### 1.3.2 函数

* setVisible – 使控件在可见/不可见的状态下进行切换
* setAlignment – 切换文字对其模式
* setStyleSheet – 更改控件样式，支持修改填充颜色、填充背景、字体样式
* resize – 重设控件大小
* move – 重设控件位置
* setText – 重设控件文字
* connect – 为信号绑定槽，对按钮的点击事件进行响应

# 项目功能与目标

## 基本功能

### 实现美观、交互友好的图形化用户界面

启动程序后，系统将首先进入标题页，用户可以通过鼠标操作实现不同界面间的切换，并随时通过返回按钮回到前移界面修改已经做出的选项。不同界面间应遵从一致的UI设计规范，提供统一的视觉反馈效果，实现良好的人机交互。

### 实现21点与德州扑克两种游戏模式

程序应至少实现21点（BlackJack）与德州扑克两种游戏模式，为用户提供更多的游戏选择。界面中应实时显示玩家当前拥有的筹码数量，并在对应轮次以图形化形式展现玩家拥有的手牌，以提供良好的游戏体验。

### 支持4名玩家单机参与游戏

程序应至少支持4名玩家在同一台电脑上参与游戏。

## 扩展功能

### 支持少于4名玩家参与游戏

本程序应在本机玩家少于四人时正常进行游戏，缺少的玩家应由计算机扮演。计算机应至少保证游戏的正常进行，并具备一定的运筹能力，保证游戏的趣味性。

### 支持多玩家通过局域网联机参与游戏

在实现玩家本地对战的基础上，支持玩家通过局域网进行联机。支持玩家自主创建房间，或通过搜索房间号加入已有房间进行游戏。

### 实现玩家历史游戏数据的存档与读取

支持以用户名作为识别用户的唯一表示，对用户的历史成绩进行保存。支持在用户以相同用户名登录时，继承其历史成绩以继续进行游戏。

### 21点模式：实现对特殊牌型的判定与注码结算

除基本的保险、分牌、投降功能外，游戏应支持对五龙、顺、同花、三条等特殊牌型的检测与相应的点数结算功能。

### 德州扑克模式：彩池限注与无限注模式

除了基本的有限注德州扑克外，应实现彩池限注与无限注两种德州扑克游戏模式，为拥有大量点数的玩家提供更丰富的游戏选择。

# 程序设计

## 页面架构

本程序共包含标题页、游戏选择页、21点游戏页、德州扑克游戏页四个页面。为保证页面刷新的相应速度，所有页面均在同一窗口中实现，以图层进行区分。

### 标题页

本页向玩家展示程序logo，并支持用户自定义用户名。

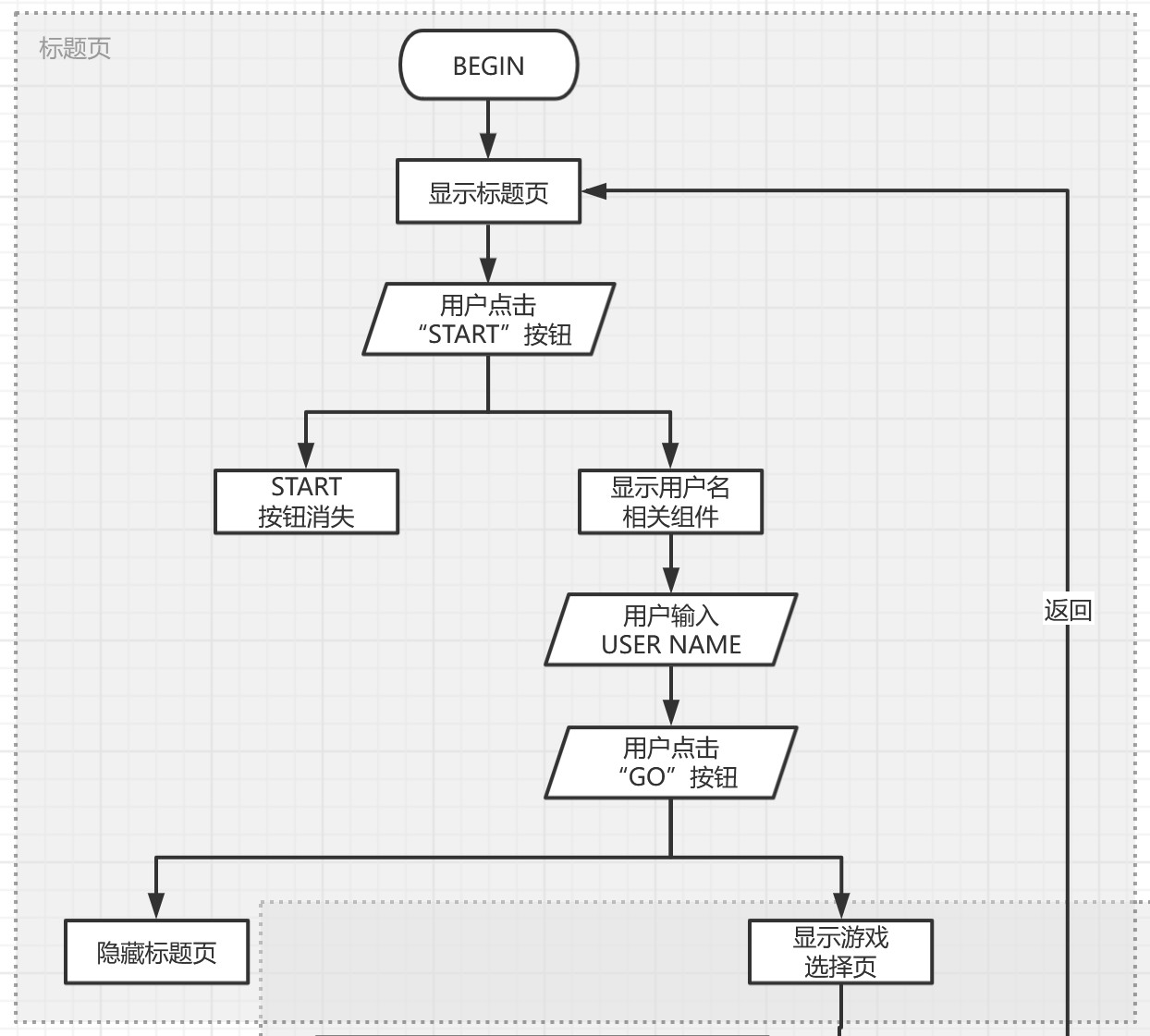


图3.1 标题页流程图

### 游戏选择页

本页支持玩家在两种支持的游戏（21点/德州扑克）与游戏模式（本地/联机）中做出选择。本地游玩时需指定玩家量。

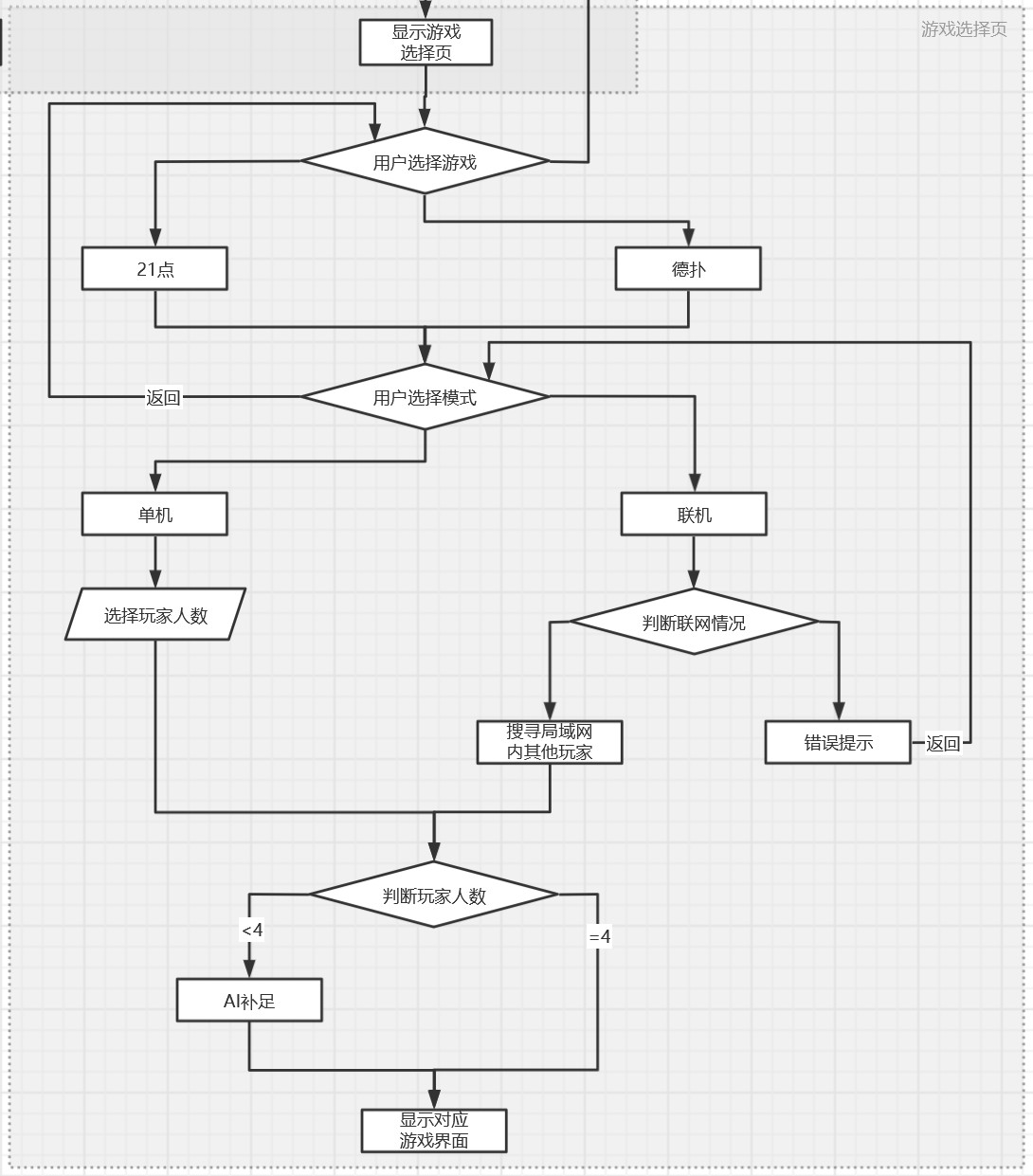


图3.2 游戏选择页流程图

### 21点游戏页

本页面支持用户进行21点游戏。

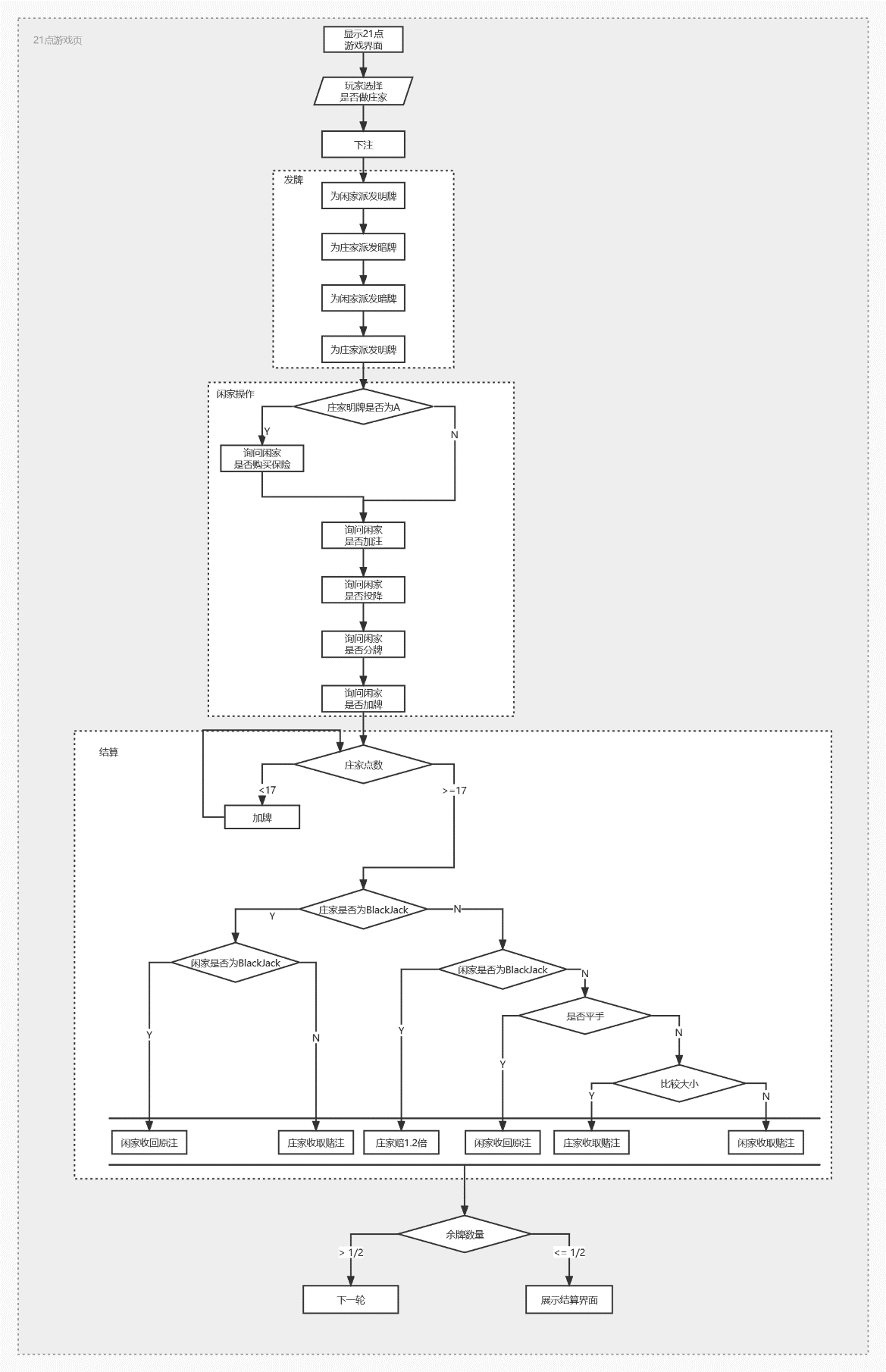


图3.3 21点游戏页流程图

### 德州扑克游戏页

本界面支持用户进行德州扑克游戏。

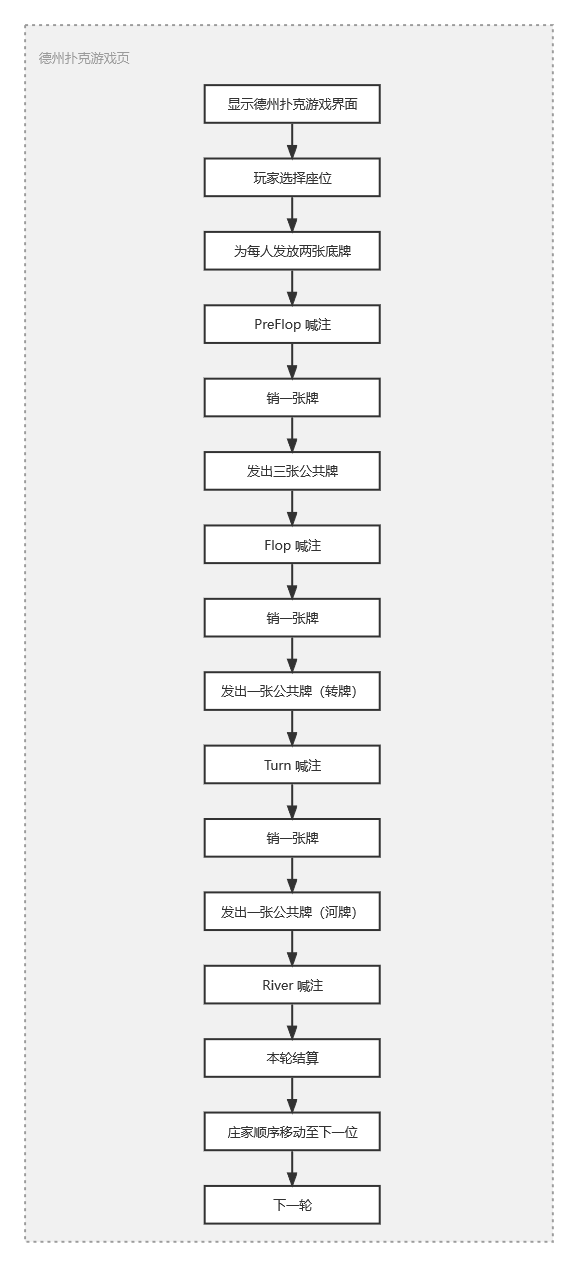


图3.4 德州扑克游戏页流程图

## 类的架构

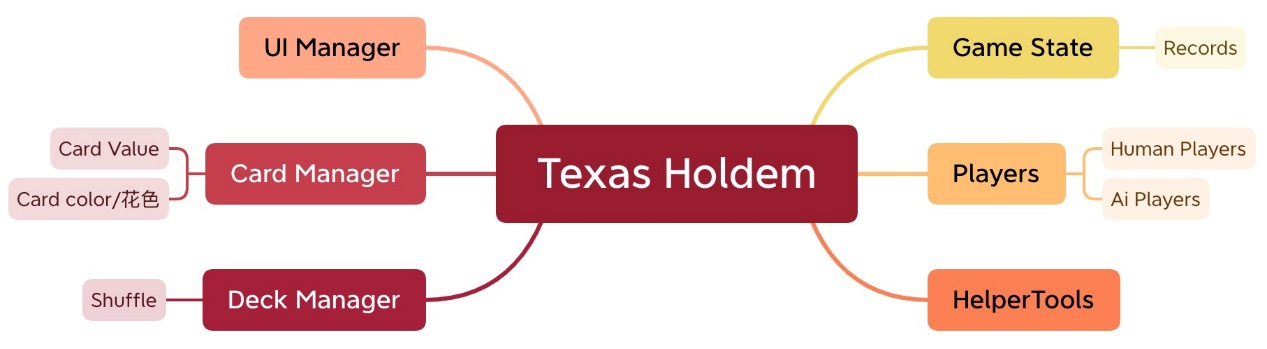
### 主窗口

1. class MainWindow : public QMainWindow //主界面类
2. {
3. Q\_OBJECT
4. public:
5. MainWindow(QWidget \*parent = nullptr);
6. ~MainWindow();         //- layer list -
7. void TitleLayer();     //0 - 标题页
8. void Unamelayer();     //1 - 用户名页
9. void GameTypeLayer();  //2 - 游戏选择页
10. void GameModeLayer();   //3 - 模式选择页
11. void InterErrLayer();   //4 - 网络错误页
12. void PlayerNumLayer(); //5 - 玩家数量页
13. private slots: //界面上各按钮点击事件的槽函数
14. void on\_b\_start\_clicked();
15. void on\_b\_go\_clicked();
16. void on\_b\_21\_clicked();
17. void on\_b\_tx\_clicked();
18. void on\_b\_local\_clicked();
19. void on\_b\_online\_clicked();
20. void on\_b\_back\_clicked();
21. void on\_b\_one\_clicked();
22. void on\_b\_two\_clicked();
23. void on\_b\_three\_clicked();
24. void on\_b\_four\_clicked();
25. private:
26. int layer{0}; //当前图层编号
27. QString  uname{""}; //用户名
28. int player\_num\_{0}; //玩家数量
29. int game\_type\_{0}; /\*
30. 游戏类型变量
31. 0 - 未选择
32. 1 - 21点
33. 2 - 德州扑克\*/
34. int game\_mode\_{0}; /\*
35. 游戏模式变量
36. 0 - 未选择
37. 1 - 本地
38. 2 - 联机\*/
39. Ui::MainWindow \*ui;};

### 21点

1. Card 纸牌类
2. {
3. Color           //花色
4. Value           //数值
5. }
6. Player 玩家类
7. {
8. community\_cards //明牌
9. get\_card        //拿牌
10. stop\_card       //停牌
11. double          //加倍
12. division        //分牌
13. surrender       //投降
14. judge           //判断是否超过21点
15. }
16. Banker  庄家类
17. {
18. community\_cards //明牌
19. under\_cards     //暗牌
20. udge            //判断是否超过21点
21. }
22. Deck  牌堆类
23. {
24. licensing       //发牌
25. shuffle         //洗牌
26. final\_judge     //判断场上的点数大小
27. }

### 德州扑克



1. class Card
2. {
3. public:
4. Card(size\_t const color, size\_t const value) :   
    color(color),value(value){}
5. string toString() const;
6. size\_t const color; //花色
7. size\_t const value; //数值
8. };
9. class Deck
10. {
11. public:
12. Card \* drawCard(); //绘制牌
13. void shuffle(); //快速洗牌
14. void riffleShuffle() //完全洗牌
15. private:
16. void swap(size\_t i, size\_t j)//换牌
17. vector<Card\*> deck;
18. vector<Card\*> cards;
19. };
20. class Player
21. {
22. public:
23. void setHand(Card\* card1, Card\* card2);
24. bool isActive() const {return !folded;}
25. void setBet(size\_t bet);// 只能追平call或者多于场上玩家的bet
26. virtual string getName();
27. size\_t getBet() const {return bet;}
28. size\_t getMoney(){return money;}
29. size\_t getId() const {return id;}
30. double getRaises() {return raises;}
31. double getFolds() {return folds;}
32. vector<Card\*> getHand() const {return hand;}
33. vector<Card\*> getCards() const; //返回玩家手牌与明牌
34. Command getLastCommand() const{return lastCommand;}
35. void resetLastCommand() {lastCommand = NONE;}
36. void resetFold(){folded = false;}
37. void giveMoney(size\_t amount){money += amount;}
38. bool isAllIn(){return money == 0;}
39. protected:
40. void raise();
41. void call();
42. void fold();
43. // player variables
44. const size\_t id;
45. vector<Card\*> hand;
46. size\_t money;
47. size\_t bet;
48. bool folded;
49. bool callWasCheck;
50. Command lastCommand;
51. double raises; //加价金额
52. double turns;
53. double calls;
54. double folds;
55. };

# 项目进度与后续计划

## 项目进度

### 4.1.1 总体进度

#### 2022/04/28 - 2022/05/01

* 对各选题的相关资料进行收集、整理
* 对可用开发框架信息进行收集、整理
* 进行第一次小组会议，确定选题（扑克牌游戏）与开发框架（Qt）

#### 2022/05/02 - 2022/05/08

* 制定编码规范
* 学习Qt编程技术
* 熟悉两种扑克牌游戏的规则
* 绘制程序整体流程图

### 4.1.2 前端进度

#### 2022/05/02 - 2022/05/08

* 完成UI设计与交互逻辑设计
* 完成类及函数设计
* 收集图片素材

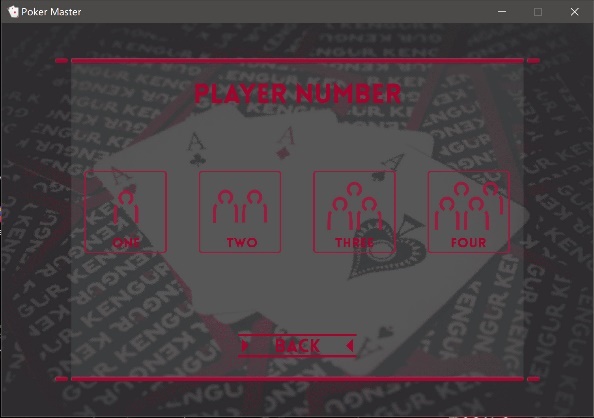
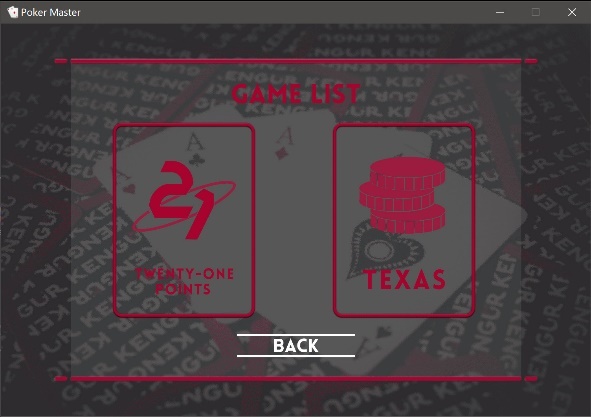
#### 2022/05/09 - 2022/05/15

* 通过PhotoShop完成对图片素材的处理
* 实现标题页



图4.1 标题页界面

* 实现游戏选择页

​

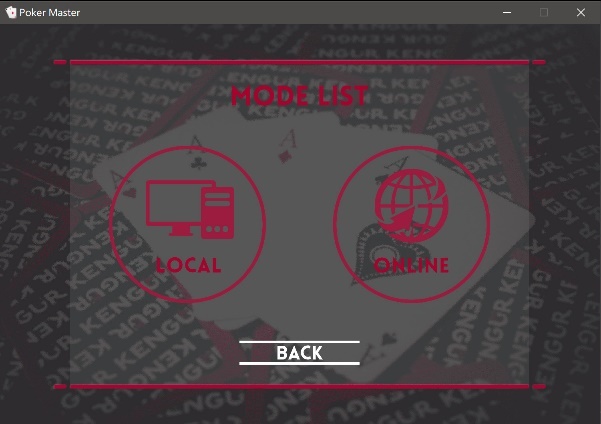


图4.2 游戏选择页各分页面

#### 2022/05/016 - 2022/05/22

* 完成21点游戏页面
* 完成德州扑克游戏页面

### 4.1.3 后端进度

#### 2022/05/02 - 2022/05/08

* 完成对类之间继承关系的设计
* 完成对各类成员函数的设计

#### 2022/05/09 - 2022/05/15

* 定义供前端调用的接口
* 完成各类中较为独立的函数

#### 2022/05/016 - 2022/05/22

* 实现类中的其余函数
* 规范现有代码的格式，并进行注释

## 4.2 后续计划

1. 完成对于单机四人游戏模式的前后端联调

2. 实现单机游戏玩家不足时由计算机扮演

3. 尝试实现联网游戏功能

# 5 成员分工

|  |  |
| --- | --- |
| 人员 | 分工 |
| 沈韵沨 （组长）  3200104392 | * 总体架构设计、协调组内活动 * 用户界面设计与实现 * 21点游戏设计与实现 |
| 王柯棣 （组员）  3200105119 | * 德州扑克游戏设计与实现 * 程序模块测试与整体测试 * 程序开发文档整理 |
| 魏鼎坤 （组员）  3200105652 | * 会议记录与资料整理 * 21点游戏设计与实现 * 中期报告撰写 |
| 赵伊蕾 （组员）  3200104866 | * 界面流程设计与流程图绘制 * 德州扑克游戏设计与实现 * 代码规范整理 |

# 附录

**OOP中期会议记录**

**会议时间：**2022 年 5 月 10 日

**会议地点：**碧峰二幢

**出席人员：**沈韵沨、王柯棣、魏鼎坤、赵伊蕾

**会议内容：**

**1. 讨论确定选题**

经过讨论和分析，我们认为扑克牌游戏在实现上具有一定挑战性，可以加强我们的代码能力，并且其功能的实现具有较强逻辑，更加适合我们。以扑克牌游戏为目标，我们可以丰富自己的专业知识，学习更丰富的本领和技能。

* 选题内容：扑克牌游戏
* 选题要求：
* 友好的图形用户界面
* 支持至少 MAX(2, 组员人数-2) 种扑克牌游戏
* 支持四玩家参与游戏（单机参与）
* 支持少于四玩家参与游戏，不足人数由计算机扮演
* 支持不同玩家局域网内联机参与游戏

**2. 讨论确定任务分工**

经过各位组员的自我展示和相关特长分析，讨论决定本次将实现任务分为两个部分，一个负责二十一点游戏，一个负责德州扑克游戏，游戏开始界面等公共界面共同开发。具体人员分工见上。

**3. 讨论确定了实现细节**

* 经过讨论分析，决定使用VS2017作为IDE，并使用QT框架辅助开发。
* 讨论确定了页面风格，确定了前端样式。
* 讨论确定了高级功能的实现
* 讨论了类的设计和架构