**斗地主软件开发设计文档**

**项目分工：UI界面--刘富康、周瑞杰**

**底层逻辑--周瑞杰**

**联网模块--罗满、周瑞杰**

1. **引言**

软件开发设计文档是软件开发过程中非常重要的一部分，它记录了软件开发的整个过程，包括需求分析、设计、实现、测试等各个阶段的内容。本文档是在开发“斗地主app”中，用于帮助规范软件开发工作，提高开发效率和开发质量。

1. **项目概述**
2. **项目名称**

基于QT6框架和QMl的斗地主APP.”牛牛斗地主”

1. **项目背景**

本项目旨在开发一款基于Qt框架和QML（Qt Modeling Language）的斗地主APP。斗地主是一款广受欢迎的扑克游戏，玩家通过出牌策略争夺胜利。本项目将利用Qt的跨平台特性和QML的直观UI设计优势，为用户提供流畅、美观的游戏体验。

1. **项目目标**

* **主界面**：使用QML设计游戏的主界面，包括玩家信息展示、牌面显示、按钮（如出牌、抢地主等）等。
* **出牌界面**：设计玩家出牌时的界面，展示可选的牌和已出的牌。
* **设置界面**：设计游戏设置界面，允许用户调整游戏参数，如音效、难度等。
* 如果你的斗地主APP需要支持多人在线对战，你需要实现网络通信功能。
* 使用Qt Network模块（如QTcpSocket、QUdpSocket等）来建立网络连接和传输数据。
* 实现自定义的网络协议，确保数据的可靠传输和解析。
* **牌组管理**：使用C++实现一个牌类（Card），以及一个管理所有牌的牌组类（Deck），用于洗牌、发牌等操作。
* **戏逻辑**：实现游戏的核心逻辑，包括判断出牌合法性、计算得分、处理游戏状态变化（如抢地主、出牌、游戏结束等）。
* **QML与C++交互**：在C++中创建QML可以访问的对象（通常是通过继承QObject并注册到QML中），用于处理QML界面的输入和更新QML界面的显示。

1. **项目范围**

**跨平台兼容性**：确保你的应用在多个平台上都能正常运行。

**用户体验**：注重界面设计和交互体验，确保用户能够轻松上手。

**安全性**：如果应用涉及网络通信，确保数据传输的安全性。

**性能优化**：对应用进行性能优化，以提高运行效率和响应速度。

1. **需求分析**
2. **用户需求**

满足用户游戏基本功能需求，斗地主软件需要严格遵循传统的斗地主游戏规则，包括发牌、抢地主、出牌、炸弹等规则，确保游戏的公平性和准确性。

斗地主软件的用户界面需要简洁、美观、易用。玩家可以方便地通过点击屏幕进行游戏操作，包括出牌、抢地主、加倍等。同时，界面设计需要符合玩家的使用习惯，提高游戏的可玩性。支持音效、音乐、背景等设置，增强游戏的沉浸感和趣味性。音效和背景音乐需要符合游戏的氛围，让玩家在游戏中获得更好的体验。

1. **功能需求**

1.完成发牌、抢地主、叫地主、出牌、结算等基本游戏功能

**2.完成用户界面与交互：**

**牌面显示**：清晰展示每位玩家的手牌、出牌记录以及底牌（如果地主已查看）。

操作提示：提供出牌规则、操作说明等提示信息，帮助新用户快速上手。

音效与动画：出牌、抢地主、结算等关键操作配以音效和动画效果，增强游戏体验等功能。

3.更多扩展功能等（如：游戏模式、设计与互动）

1. **非功能需求**
2. 性能需求（响应时间、并发处理能力、资源利用率）

2. 可靠性需求（稳定性、数据一致性、容错性）

3.易用性需求（用户界面友好、帮助与文档、反馈机制）

4.可维护性需求（模块化设计、代码质量）

5.安全性需求（数据加密）

6.可扩展性需求（新功能支持、跨平台支持）

1. **系统设计**

**1.系统架构**

一、系统架构概述

二、前端展示层

三、业务逻辑层

四、数据访问层

五、网络通信层

六、模块化与可扩展性

七、实例参考

**2.模块设计**

### 用户界面模块：**游戏界面**：设计直观、友好的游戏界面，包括玩家信息展示、牌面显示、操作按钮（如出牌、抢地主、过牌等）等。界面应支持多种分辨率和设备，确保在不同平台上的良好显示效果。

### 游戏逻辑模块**发牌**：实现自动发牌功能，确保每个玩家获得相同数量的牌，并保留三张底牌作为地主牌。**叫地主**：实现叫地主的逻辑，允许玩家在发牌后根据手中的牌型选择是否抢地主。**出牌与过牌**：根据斗地主的规则，实现玩家出牌和过牌的功能。确保出牌符合规则，并处理各种复杂的牌型组合。**胜负判定**：在游戏结束时，根据剩余的牌量和牌型大小判定胜负，并给出相应的结算结果。

### 网络通信模块：**服务器与客户端通信**：实现服务器与客户端之间的网络通信功能，确保游戏数据的实时传输和同步。**房间管理**：管理游戏房间，包括创建房间、加入房间、离开房间等功能。确保玩家能够顺利地与其他玩家匹配并开始游戏。**消息处理**：处理各种游戏内消息，如出牌消息、抢地主消息等，并实时更新游戏状态。

1. **接口设计**

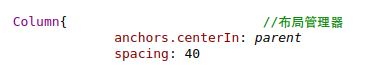
### 游戏大厅接口：**1.房间列表接口**：用于获取当前可加入的游戏房间列表。该接口可能返回房间的ID、名称、玩家人数、游戏状态等信息。**2.创建房间接口**：允许玩家创建新的游戏房间，并设置房间的基本参数（如游戏模式、底分等）。该接口可能接收房间名称、游戏模式、底分等作为输入参数，并返回房间ID和创建结果。

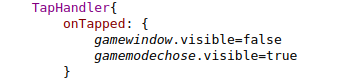
### 2. 游戏逻辑接口：1.**发牌接口**：在游戏开始时，为每位玩家分发牌。该接口可能不需要外部输入参数，直接在游戏内部调用，并返回每位玩家的手牌信息。**2.叫地主接口**：允许玩家在发牌后抢地主。该接口可能接收玩家的叫地主请求（包括是否叫地主、叫地主的倍数等）作为输入参数，并更新游戏状态和地主身份。**3.牌接口**：处理玩家的出牌操作。该接口接收玩家的出牌信息（包括牌型和具体的牌）作为输入参数，并验证出牌是否合法。如果合法，则更新游戏状态和牌面信息；如果不合法，则返回错误信息。**4.负判定接口**：在游戏结束时，根据剩余牌量和牌型大小判定胜负，并计算积分。该接口可能不需要外部输入参数，直接在游戏内部调用，并返回胜负结果和积分信息。

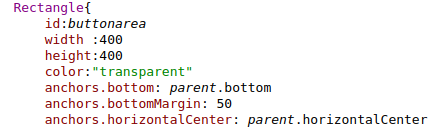
1. **详细设计**
2. **页面设计：**

1.主页面设计：

使用了简单的布局器，以及按钮，rectangle元素，窗口界面的可视化（visible：）进行了页面变换：大致函数如下:



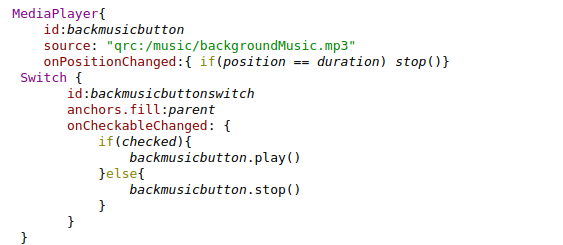




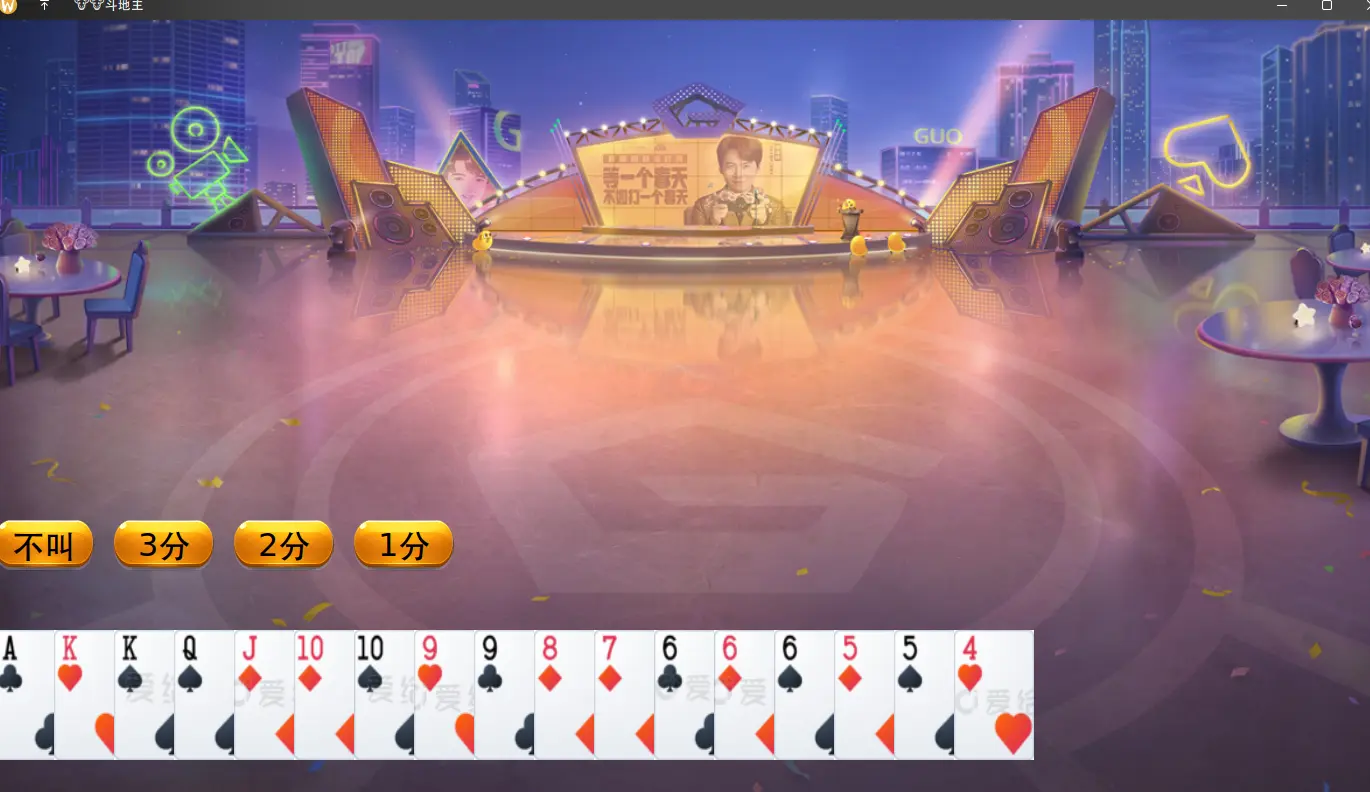
1. 设置界面与模式选择界面



采用了switch开关来设置背景音乐和游戏音乐的状态，使用了SoundEffect类型来播放未压缩的音频，我们生成了一个mediaplayert类型的实例来在任意位置引用他。用户和界面交互主要是使用了taphander的方法进行交互。



1. 游戏界面设计





游戏内界面截图，主要与底层逻辑结合

1. **测试**
2. **测试计划**

#### 发牌功能：

#### ****测试目标****：确保发牌逻辑正确，每位玩家手牌数量正确，底牌数量正确。

**测试方法**： 验证初始发牌时，三位玩家手牌数量均为17张，底牌3张。

验证洗牌算法的随机性，确保每次发牌都不相同。

#### 2. 叫地主功能

**测试目标**：确保叫地主逻辑正确，包括随机选定首叫玩家、明牌优先权等。

**测试方法**： 验证第一轮是否由系统随机选定首叫玩家。

验证后续轮次是否由上一轮获胜玩家（先出完牌的玩家）先叫地主。

验证有玩家明牌时，是否优先获得叫地主权。

验证多人同时叫地主时，系统如何处理（如按叫分高低、随机等）。

#### 3. 出牌功能

**测试目标**：确保出牌逻辑正确，包括出牌顺序、牌型判断、出牌特效等。

**测试方法**：

证出牌顺序是否正确，即地主先出，然后是农民按逆时针顺序出牌。

验证各种牌型（单牌、对子、顺子、飞机、炸弹等）的出牌逻辑是否正确。

验证出牌时的声音、特效是否显示正确。

验证出牌倒计时功能是否正常，包括提醒和托管逻辑。

#### 4. 结算功能

**测试目标**：确保结算逻辑正确，包括倍数计算、积分加减等。

**测试方法**：

验证不同倍数（叫地主分数、炸弹、火箭等）对最终积分的影响。

验证春天/反春天对积分的加倍效果。

验证托管情况下的积分处理逻辑。

验证积分封顶逻辑是否正确。

### 二、性能测试

**测试目标**：确保应用在不同设备和网络环境下的性能表现良好。

**测试方法**：

使用不同配置的手机、平板等设备进行测试，验证应用的兼容性。

在不同网络环境下（如Wi-Fi、4G/5G）进行测试，验证应用的网络稳定性和响应速度。

进行压力测试，模拟大量用户同时在线的情况，验证应用的稳定性和承载能力。

### 三、用户体验测试

**测试目标**：确保应用的界面友好、操作便捷、反馈及时。

**测试方法**：

组织用户进行试用，收集用户对界面设计、操作流程、功能布局等方面的反馈。

验证应用的提示信息、错误处理是否清晰易懂。

验证应用的音效、动画等是否增加了游戏的趣味性。

### 四、安全测试

**测试目标**：确保应用的数据安全、防作弊机制有效。

**测试方法**：

验证应用的数据加密、传输安全等机制是否有效。

模拟作弊行为（如使用外挂、修改数据等），验证应用的防作弊机制是否有效。

验证应用的用户权限管理是否合理，防止非法访问和篡改数据。

### 五、兼容性测试

**测试目标**：确保应用在不同操作系统版本、不同分辨率的设备上均能正常运行。

**测试方法**：

在不同操作系统版本（如Android不同版本、iOS不同版本）的设备上进行测试。

在不同分辨率的设备上进行测试，验证应用的适配性。

### 六、回归测试

**测试目标**：确保在修复bug或添加新功能后，原有功能不受影响。

**测试方法**：

对修复的bug进行复现测试，验证是否已解决。

对新增功能进行全面测试，确保符合需求。

对原有功能进行随机抽样测试，确保未引入新的问题。

1. **测试结果**

实现了完整的出牌逻辑，实现了简单的人机对弈，完成了计算机之间的连接。

**七、总结**

**界面友好**：游戏的界面设计应简洁明了，操作直观易懂，避免过多的复杂操作，让玩家能够快速上手。**反馈及时**：游戏中的各种操作都应给予及时的反馈，比如出牌成功、失败、轮到自己出牌等，增强玩家的沉浸感。**音效与动画**：适当的音效和动画可以增加游戏的趣味性，但要注意不要过于繁琐，以免影响游戏性能。**规则细化**：在开始编程之前，务必确保对斗地主的规则有全面且细致的理解，包括发牌规则、叫地主规则、出牌规则、结算方式等。细节决定成败，任何一个规则的遗漏或误解都可能导致游戏逻辑出现问题。**异常处理**：设计时要考虑到各种边界情况和异常情况，比如牌面组合是否合法、玩家出牌顺序是否正确等，确保游戏的鲁棒性。**网络同步**：对于网络版斗地主，确保各个玩家之间的数据同步至关重要。需要设计合理的同步机制，保证游戏的公平性和实时性。**团队协作**：斗地主程序的开发往往需要多人协作，团队成员之间应保持良好的沟通和协作，共同解决问题，推动项目进展。**知识共享**：在团队内部建立知识共享机制，鼓励成员分享技术心得、经验教训等，提升团队整体技术水平。