中国象棋项目开发说明文档

本文档旨在说明本组关于中国象棋项目的开发说明，以帮助了解项目需求和实施细节。

1. **项目概述**

中国象棋项目是一个基于计算机的两人对弈游戏，旨在提供一个可交互、具有挑战性的游戏体验。玩家可以与朋友对战或与电脑进行对弈。

1. **技术要求**
2. 开发语言

采用QML和C++交互进行开发。

1. 开发平台

Felgo3、QtCreater

1. 用户界面

参考中国象棋APP做。

1. 游戏规则算法（必做）

实现一个规则引擎，用于验证玩家的移动是否符合中国象棋的规则，并检测胜利的情况。

1. AI对手（选做）

如果计划实现人机对战模式，需要设计和实现一个AI 引擎，能够计算电脑玩家的下一步走法。

1. 联机对战（选做）

如果计划实现联机对战模式，需要通过局域网连接，能够连接两台设备。

1. **开发过程**
2. 任务划分

首先分出来了几个任务，包括架构设计，用户界面（UI）设计，棋子移动规则算法，网络编程实现联机对战，完成以上功能后再根据时间考虑是否要完成AI对战。

1. 个人分工

秦昊池：分析架构，学习棋子移动规则算法

聂安芹：学习网络编程，学习QML如何与C++交互和完成主菜单界面

傅佳乐：分析架构，大致了解网络编程，学习棋子移动方法和完成游戏界面

1. 时间分配

确认选题后，计划在七天内完成。

第一天进度

所需的图片资源整理完成，完成了主菜单、帮助关于、系统设置界面和界面之间的切换。

第二天进度

完成单人游戏和联机对战的界面，加入了音量控制的功能。

第三天进度

大家都在学习。初步完成游戏界面的设计和棋子、棋盘类，加入了游戏界面的系统对话框，实现了主题更换的功能。

第四天进度

完成棋子和棋盘类的设计，基本完成游戏界面，可以实现棋子的移动但是没有添加移动规则。完成了网络编程C++部分，尚未交互。

第五天进度

加入了棋子移动算法，现在每个棋子可以按照其对应的规则移动了。联机对战可以连接了，但不能同步双方的棋盘。加入了一些提示的对话框，如非己方回合的提示，游戏结束的提示对话框等。

第六天进度

加入了计时器和计步器。联机对战功能实现了！

第七天进度

找BUG，找用户做测试，做最后的细节调整和优化，提交版本到GitHub上。决定不做AI对战了。

1. **棋子移动规则算法说明**

1.将或帅

（1）移动范围：它只能在王宫内移动。

（2）移动规则：它每一步只可以水平或垂直移动一点。

（3）算法：

function relation( row1, col1, row, col) {

return Math.abs(row1-row)\*10+ Math.abs(col1-col);

}计算当前位置和下一步要走位置的关系

r=relation(row1, col1, row2, col2)

if(r !== 1 && r !== 10||col2<4||col2>6||(row2>=4&&row2<=7))

{

return false;

}else{

return true;

}

通过r来判定当前位置和要下一步的位置的关系，如果不满足r=1和r=10,或者将未在九宫格内，不能移动

2.士

（1）移动范围：它只能在王宫内移动。

（2）移动规则：它每一步只可以沿对角线方向移动一点。

（3）算法： r=relation( row1, col1, row2, col2)

if(r !== 11 ||col2<4||col2>6||(row2>=4&&row2<=7)){

return false;

}else{

return true;

}

3.象

（1）移动范围：河界的一侧。

（2）移动规则：它每一步只可以沿对角线方向移动两点，另外，在移动的过程中不能够穿越障碍。

（3）算法：if(getID((row1+row2)/2,(col1+col2)/2))

{

return false

}

r=relation( row1, col1, row2, col2)

if(getID(row1,col1).camp!==camp)//camp==hong

{

if(row2>5){

return false;

}else if(r !== 22){

return false

}else{

return true;

}

}else {

if(row2<6){

return false;

}else if(r !== 22){//判定塞象眼

return false;

}else{

return true;

}

}

4.马

（1）移动范围：任何位置

（2）移动规则：每一步只可以水平或垂直移动一点，再按对角线方面向左或者右移动。另外，在移动的过程中不能够穿越障碍。

（3）算法：

r=relation( row1, col1, row2, col2)

if((r !== 12 && r !== 21)){

return false;

} else {

if(r===12){

if(getID(row1,(col1+col2)/2)){

return false

} else { return true }

} else if (r===21){ //判定撇脚马

if(getID((row2+row1)/2,col1)){

return false

}else{ return true }

}

return true

}

5.车

（1）移动范围：任何位置

（1）移动规则：可以水平或垂直方向移动任意个无阻碍的点。

function getStoneCountAtLine( row1, col1, row2, col2) {//判定点击的棋子到下一步要到的格子中间隔了几颗棋子

var ret = 0;

if(row1 !== row2 && col1 !== col2)

return false;

if(row1 === row2 && col1 === col2)

return false;

if(row1 === row2) {

var min = col1 < col2 ? col1 : col2;

var max = col1 < col2 ? col2 : col1;

for(var col = min + 1; col<max; ++col) {

if(getID(row1, col)) {

++ret;

}

}

} else {

min = row1 < row2 ? row1 : row2;

max = row1 < row2 ? row2 : row1;

for(var row = min+1; row<max; ++row) {

if(getID(row, col1)) {

++ret;

}

}

}

return ret;

}//ret表示棋子中间隔的其他棋子数

var ret = getStoneCountAtLine(row1, col1, row2, col2);

if(ret===0)

{

return true;

}else

{

return false

}

6.炮

（1）移动范围：任何位置

（2）移动规则：移动起来和车很相似，但它必须跳过一个棋子来吃掉对方的一个棋子。

（3）算法：ret = getStoneCountAtLine(row2, col2, row1, col1);

if(getID(row2,col2)) {

if(ret===1) {

return true

} else {

return false

}

}else {

if(ret===0) {

return true

} else {

return false

}

7.兵

（1）移动范围：任何位置

（2）移动规则：每步只能向前移动一点。过河以后，它便增加了向左右移动的能力，兵不允许向后移动。

（3）算法： var r=relation( row1, col1, row2, col2)

if(getID(row1, col1).camp!==camp) {

if(row2 < 6) {

if(row2 - row1 == 1 && (r === 1 || r === 10)) {

return true;

} else {

return false;

}

} else {

if((row2 - row1 == 1 || Math.abs(col2 - col1) == 1) && r === 1 || r === 10){

return true;

} else {

return false;

}

}

}else {

if(row2 > 5) {

if(row2 - row1 == -1 && (r === 1||r === 10)) {

return true;

} else {

return false;

}

} else {

if((row2 - row1 == -1 || Math.abs(col2 - col1) == 1)&&(r === 1 || r === 10)){

return true;

} else {

return false;

}

}

}