设计：

假设我是红方。

此时CGame的成员变量currentRound为RED，所以此刻只有红方棋子能被选中并移动。所以当任意一个棋子被触发鼠标点击事件，首先判断自己的颜色是否对应现在的行动方，只有对应的才能继续，不对应的直接返回。

当我点击一枚红色的棋子，这枚棋子被选中，同时发送信号(PieceSelected)给scene，把scene中的PieceIsSelected设为true，此时：

一、 再此点击这枚棋子，这枚棋子取消选中

二、点击了另一枚己方棋子

二、点击地图上任意一点，计算出这一点对应的棋盘坐标，同时由scene调用被选中棋子的Move() 函数，Move()函数会自行判断是否能移动到那一点，如果不能，每点击一次会发出提示音；如果能，棋子移动到那点，轮到黑方走

一方完成一次行动分为两步：1. 选择棋子 2. 选择地点

整个界面分为三种区域：1. 己方棋子 2. 敌方棋子 3. 空白棋盘

在第一步中，PieceIsSelected为false：

1. 点击己方棋子：选中

2. 点击敌方棋子：无

3. 点击空地：无

在第二步中，PieceIsSelected为true：

1. 点击己方棋子：取消原来选中的棋子，选中现在的，继续第二步

2. 点击敌方棋子：计算敌方棋子坐标，发送给scene，触发PieceTriesToMove();

3. 点击空地：计算空地坐标，发送给scene，触发PieceTriesToMove();

棋子的鼠标点击函数：

判断是否是自己回合？

是🡪发射信号触发scene的RecordSelectedPiece()函数。

否🡪发射信号触发scene的PieceTriesToMove()函数。

棋盘点击函数：

发射信号触发scene的PieceTriesToMove()函数。

RecordSelectedPiece()函数：

判断PieceIsSelected是否为真？

真🡪触发**m\_SelectedPiece 的**ChangeIconOnSelection()函数，并判断触发信号的对象是否是**m\_SelectedPiece** 自己？

是🡪返回

否🡪触发发射信号对象的ChangeIconOnSelection()函数

假🡪触发发射信号对象的ChangeIconOnSelection()函数

PieceTriesToMove()函数：

判断PieceIsSelected是否为真？

真🡪触发发射信号对象的Move函数，若得到的返回值为真，触发**m\_SelectedPiece** 的ChangeIconOnSelection()函数，**m\_SelectedPiece** 指向NULL，PieceIsSelected设为false，其他的会自动刷新。返回值若为假，返回

假🡪返回

判断PieceIsSelected是否为真？是🡪发射信号触发scene的PieceTriesToMove()函数。

否🡪直接返回