CPlayer类：

数据成员：

坐标（Point类）

学分（int）

专业（QString或者enum，默认值为“新生”）

外观图片（jpg，=专业对应的图片，需为一个数组）

疲劳值（int，=0时可行动）

成员函数：

CPlayer(point, int, mature, Appearance, Fatigue=0);//构造函数

int getCredit(void) const;

void setMyCredit(int);//改变自己的学分时要发送一份改变后的学分给服务器

void setOthersCredit(int);//设置别人的学分时不用发送给服务器

int getFatigueValue(void) const;

void setFatigueValue(int);

point getPoint(void) const;

void move(int);//接受传来的行动步数

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

{

if(x1 == x2)

{

for(;y1 < y2; y1++)

//绘图

}

else

{ double k = XXX;//计算斜率

double b = XXX;//计算截距

for(;x1 < x2; x1++, y1 = k\*x2 + b)

//绘图

}

CPoint类：

数据成员：

double x, y;

Point \* next;

const int numOfEvent;

Event \* events;

成员函数：

Point(x, y, next, num, events[]);//构造函数，使用成员初始化器初始化

virtual bool isBranch(void) const;

double getX(void) const;

double getY(void) const;

virtual ~Point(void);

CBranchPoint类：public Point：

数据成员：

Point \* turn;

成员函数：

branchPoint(x, y, next, turn, num, events[]);//使用成员初始化器初始化基类

bool isBranch(void) const;

CHouse类：

数据成员：

QString name;//编号

QSstring owner;

int level;

int free\*;//存储不同等级对应的过路费

成员函数：

int getFree(void) const;//返回与当前等级相对应的过路费

void Upgrade(int);

void setOwner(QString);

int getLevel(void) const;

int getOwner(void) const;

CEvent类（ABC）：

数据成员：

QString id;

QString content;

工具函数：

int roll(int);//参数是随机数范围；

//种种子：qsrand(QTime(0, 0, 0).secsTo(QTime::currentTime()));

void login(void);//登陆函数，需与服务器进行数据交互

其他：

用QMessageBox来实现提示