

# QT 大作业实验报告

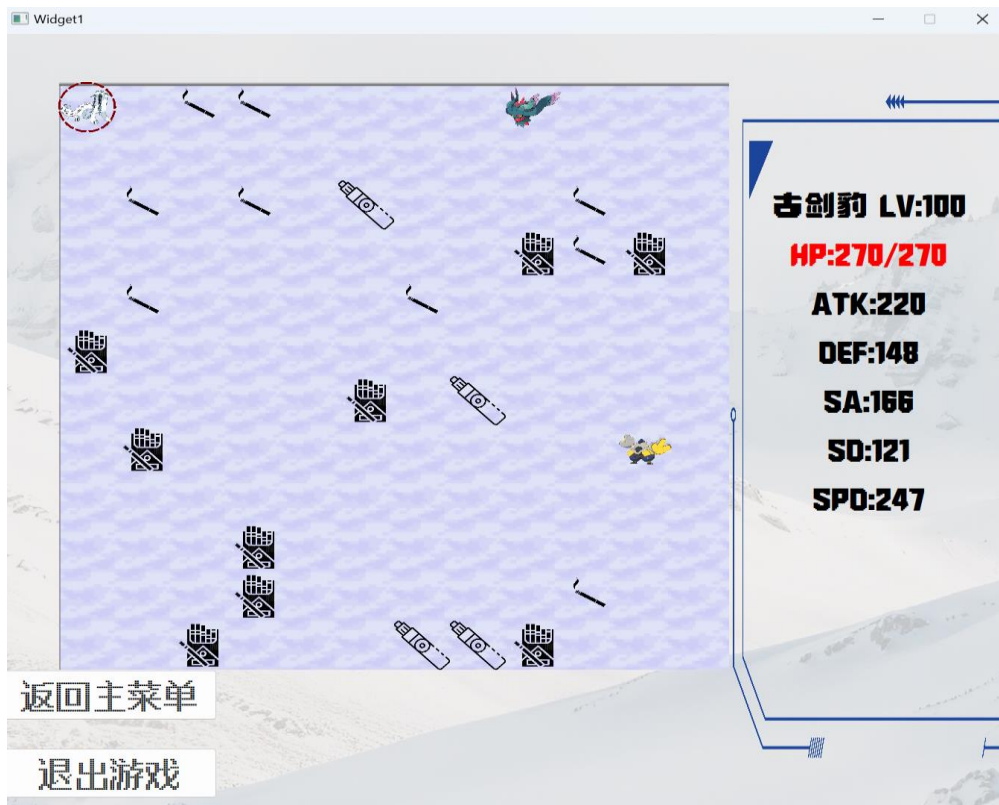
## （一）程序功能介绍

本程序是一款类宝可梦游戏，以下是游戏的画面截图：



游戏的主界面

进入程序后会先进入主菜单界面并播放背景音乐（BGM），点击“开始游戏”可以进入人物移动界面，点击“退出游戏”会直接关闭程序，将鼠标悬浮于相应选项上时可使其变色。



人物移动界面

在人物移动界面中，敲击“W”、“A”、“S”、“D”键或上下左右键可实现人物在地图上下左右四个方向的移动，进入地图上有特殊图标的位置可触发相应事件，如拾取道具会导致宝可梦属性的上升或下降，而进入敌人所在位置时会强制触发战斗。事件触发后，当前位置的特殊图标消失，无法再次触发该事件，击败地图上所有敌人后游戏胜利。同时，触发移动界面的相应按键可以退回主菜单或是直接退出游戏。



触发地图上交互事件后的界面



战斗界面

进入战斗界面后，玩家每个回合可选择四个技能中的一个作为当前

回合使用的技能，同时对方（电脑）也会从其四个技能中随机选择一个技能，此时会进入战斗环节，屏幕上会出现相应动画和文字展现战斗过程，当一方 hp 归 0 之后，战斗将直接结束，若己方宝可梦 hp 先归 0，则进入战斗失败的界面。鼠标悬置于技能上方时右侧会显示技能 pp 和技能类别。



失败后进入的界面

战斗失败界面处可以退出游戏或返回主菜单。



## （二）项目各模块与类设计细节

项目的主要模块包括地图模块、人物移动模块以及战斗模块。

地图模块使用了一个二维数组来存储各个位置的状态，0 表示没有事件，1，2，3 分别对应不同的道具事件，4，5 则代表不同战斗事件。地图的初始化采用随机初始化的方式，实现方式如下图所示：

```
void GameMap::InitByRand() {
    srand(time(NULL));
    for(int i=0; i<mRow; ++i) {
        for(int j=0; j<mCol; ++j) {
            mPArr[i][j] = rand() % 15;
            if(mPArr[i][j] > 3) mPArr[i][j] = 0;
        }
    }
    int tx=rand()%mRow,ty=rand()%mCol;
    while(tx==0&&ty==0){
        tx=rand()%mRow,ty=rand()%mCol;
    }
    mPArr[rand()%mRow][rand()%mCol]=4;
    while(mPArr[tx][ty]==4||(tx==0&&ty==0)){
        tx=rand()%mRow,ty=rand()%mCol;
    }
    mPArr[tx][ty]=5;
    mPArr[0][0] = 0;
}
```

对于地图绘制，使用了 switch 语句来实现对不同状态的绘制：

```

void GameMap::Paint(QPainter* _p, QPoint _Pos) {
    for(int i=0; i<mRow; i++) {
        for(int j=0; j<mCol; j++) {
            QString imgUrl = ":/image/default.png";
            //QString imgUrl;
            switch(mPArr[i][j]) {
                case Grass:
                    continue;
                case Cigar:
                    imgUrl = ":/image/cigar.png"; break;
                case Elec:
                    imgUrl = ":/image/elec.png"; break;
                case Jiu:
                    imgUrl = ":/image/smoke.png"; break;
                case Mount:
                    imgUrl = ":/image/ironbundle.png"; break;
                case Enemy:
                    imgUrl = ":/image/Fluttermane.png"; break;
            }

            if(imgUrl == ":/image/default.png") {
                qDebug("error: %d\n", mPArr[i][j]);
            }
            QImage img(imgUrl);
            _p->drawImage(QRect(_Pos.x() + j*Size, _Pos.y() + i*Size, Size, Size), img);
        }
    }
    // _p->drawImage(QRect(600, 20, 200, 100), QImage(":/image/title.png"));
}

```

在人物移动模块中，定义了 role 类并赋予其位置属性，输入按键指令后，如果接受的是移动相关的按键指令，会通过一个 collision 函数来更新人物位置和地图状态：

```

void Widget1::Collision(int _dRow, int _dCol) {
    int newRow = mRole->mRow + _dRow;
    int newCol = mRole->mCol + _dCol;
    if(newRow < 0 or newRow >= N or newCol < 0 or newCol >= N) return ;
    // if(mPMap->mPArr[newRow][newCol] == 1) {
    //     return;
    // }
    //else move
    mRole->Move(_dRow, _dCol);
    if(mMap->mPArr[mRole->mRow][mRole->mCol] > 0 and mMap->mPArr[mRole->mRow][mRole->mCol] <= 5) {
        pao->boost(pickType = mMap->mPArr[mRole->mRow][mRole->mCol]);
        pick = true;
        pickType = mMap->mPArr[mRole->mRow][mRole->mCol];
        mMap->mPArr[mRole->mRow][mRole->mCol] = 0;
    }
}

```

并且在每次接受按键指令后，会进行画面的重绘更新。

战斗模块定义了宝可梦类和技能类。

技能类具有五种属性：名称、pp 数、威力、优先度以及技能类型。

```

class skill{
protected:
    int pp,pp_now;
    int pow;
    int prior;
    QString name,type;
public:
    skill(QString name0="",int pp0=0,int pow0=0,int prior0=0,QString type0=""):pp(pp0),pp_now(pp0),pow(pow0),name(name0),prior(prior0),type(type0){}
    int get_pp(){return pp_now;}
    int get_pp_max(){return pp;}
    int get_prior(){return prior;}
    int get_pow(){return pow;}
    void use_pp(){pp_now--; }
    QString& get_name(){return name;}
    QString& get_type(){return type;}
};

```

宝可梦类包含宝可梦名称、宝可梦的六维属性：hp、atk、def、sa、sd、spd 以及宝可梦的四个技能，以及判断守住相关技能连续成功概率的参数，同时还有判断出手顺序的友元函数和在移动界面中触发相关事件后属性增减的成员函数、获取相关属性的成员函数：

```

class pokemon{
protected:
    int hp,atk,spd,def,sa,sd;
    int level[6];
    QString name;
public:
    int hp_now,suc,fir;
    friend skill;
    skill skl[4];
    pokemon(QString name0,int hp0,int atk0,int spd0,int def0,int sa0,int sd0,skill skl10,skill skl11,skill skl12,skill skl13:hp(hp0),atk(atk0),spd(spd0),def(def0),sa(sa0),sd(sd0){
        name=name0;
        skl[0]=skl10;
        skl[1]=skl11;
        skl[2]=skl12;
        skl[3]=skl13;
        hp_now=hp;
        suc=0,fir=1;
    }
    int get_pp(int idx){ return skl[idx].get_pp();}
    int get_pp_max(int idx){return skl[idx].get_pp_max();}
    int get_hp_now(){return hp_now;}
    int get_hp(){return hp;}
    int get_atk(){return atk;}
    int get_def(){return def;}
    int get_sa(){return sa;}
    int get_sd(){return sd;}
    int get_spd(){return spd;}
    QString get_name(){return name;}
    void boost(int idx){
        if(idx==1){
            hp_now+=25;
            if(hp_now>1) hp_now=1;
        }
        if(idx==2){
            atk+=10;def+=10;
        }
        if(idx==3){
            hp_now+=50;
            if(hp_now>hp) hp_now=hp;
        }
    }
    bool cmpspd(pokemon& p){
        return spd>p.spd;
    }
    friend bool judge_prior(pokemon* poke1,pokemon* poke2,skill& skl1, skill& skl2){
        if(skl1.get_name()=="突袭"&&skl2.get_name()=="守住") return 1;
        if(skl2.get_name()=="突袭"&&skl1.get_name()=="守住") return 0;
        if(skl1.get_prior()>skl2.get_prior()) return 1;
        if(skl2.get_prior()>skl1.get_prior()) return 0;
        if(poke1->spd>poke2->spd) return 1;
        if(poke2->spd>poke1->spd) return 0;
        return rand()%2;
    }
};

```

定义了函数 battle 和 attack，其中 battle 为模拟两宝可梦间的战斗过程，attack 模拟一方的宝可梦向另一方宝可梦使用技能的过程，battle 函数先判断两宝可梦的出手顺序再轮流执行 attack。

### （三）小组成员分工情况

王振宇：完成主界面设计开发，收集素材，提供部分接口与基本类，

实现部分模块拼接。

王子昂：收集素材，编写地图、人物移动模块，实现了背景音乐，实现了基础的移动界面 ui。

张力元：收集素材，编写战斗模块和战斗界面，对移动界面进行美化和改动，添加地图事件，实现地图模块与战斗模块的拼接，编写了战斗胜利和战斗失败的相关界面，负责了实验报告的撰写。

#### （四）项目总结与反思

本项目借助 qt 程序开发平台，使用 C++中面向对象的内容与 qt 中的开发工具完成了当初设计文档中的设计规划，成功实现了可交互且兼具美观性的程序。各组员密切交流与深入合作，使用 github 的代码托管功能实现对代码版本的控制，使得代码开发便捷且高效。

该项目增长了组员们的代码开发经验，使得我们对 C++中面向对象的部分有了更为成熟的理解和更加熟练的掌握，同时也更加具有团队意识和沟通意识，这对以后的工作和学习都有着极大裨益。



