

## 一. 整体设计

框架与 project1 相同，只是这次有多个地图，并且添加了各种植物和僵尸。  
首先会弹出一个选择地图的界面

```
void MainWindow::choose()//显示出第一步的界面
{
    timer = new QTimer;
    QGraphicsScene* scene1 = new QGraphicsScene(this);
    scene1->setSceneRect(0,0,1800,1200);
    view = new QGraphicsView(scene1,this);
    view->resize(1802,1202);//为啥要+2? ?
    view->setRenderHint(QPainter::Antialiasing);
    scene1->setItemIndexMethod(QGraphicsScene::NoIndex);//If your scene uses many animations and you are experiencing slowness,
    scene1->setBackgroundBrush(QPixmap("D:/QtProjects/images/surface1.png"));
    button* button1 = new button("默认地图");
    button1->setPos(1305,250);//这个是基于本来的图片的坐标，需要加150，变成场景中的坐标
    scene1->addItem(button1);
    button* button2 = new button("自建地图");
    button2->setPos(1305,500);//这个是基于本来的图片的坐标，需要加150，变成场景中的坐标
    scene1->addItem(button2);
    connect(timer, &QTimer::timeout, this, &MainWindow::showmap);
    timer->start(50);
    view->show();
}
```

通过设置一个全局变量，让程序知道当前的地图是新建的还是默认的。  
然后对于不同的地图，我们进行读入文件操作，根据读入的数据，我们来在屏幕上画出不同的地图。

```
void MainWindow::showmap()
{
    if(MAP_NUM == "默认地图")
    {
        /*Life = 10; ...*/
        life = 10;
        timer->stop();
        timer = new QTimer;
        scene2 = new QGraphicsScene(this);
        scene2->setSceneRect(0,0,1800,1000);
        view->setScene(scene2);
        view->resize(1802,1002);//为啥要+2? ?
        view->setRenderHint(QPainter::Antialiasing);
        scene2->setItemIndexMethod(QGraphicsScene::NoIndex);
        ROW = 5;COL = 9;
        for(int i=0;i<=ROW;i++)
        {
            scene2->addLine(start_x,start_y+i*height1,COL*width1+start_x,start_y+i*height1);
        }

        for(int i=0;i<=COL;i++)
        {
            scene2->addLine(start_x+i*width1,start_y,start_x+i*width1,start_y+ROW*height1);
        }
        addPaths_default(); addShop();addCard();
        connect(timer, &QTimer::timeout, scene2, &QGraphicsScene::advance);//为啥不用多线程的就不行
        connect(timer, &QTimer::timeout, this, &MainWindow::addZombie);
        connect(timer, &QTimer::timeout, this, &MainWindow::check);
        timer->start(50);
        view->show();
    }
    else if(MAP_NUM == "自建地图")
    {
        //...
    }
}
```

```

else if(MAP_NUM == "自建地图")
{
    //从指定的地址读取自建的地图的信息，然后设置新的地图

    life = 10;
    timer->stop();
    timer = new QTimer;
    scene2 = new QGraphicsScene(this);
    scene2->setSceneRect(0,0,1800,1000);
    view->setScene(scene2);
    view->resize(1802,1002);//为啥要+2? ?
    view->setRenderHint(QPainter::Antialiasing);
    scene2->setItemIndexMethod(QGraphicsScene::NoIndex);//If your scene uses many animations and you are experiencing slowne
    QFile file("C:/Users/quanh/nju-documents/AdvancedProgramming-2021/homework/project-quanhaoli/src/map_self_build.txt");
    if (!file.open(QIODevice::ReadOnly | QIODevice::Text))
        return;
    QTextStream in(&file);
    in>>ROW>>COL;
    if(ROW>5)
        ROW=5;
    if(COL>10)
        COL=10;
    //绘制表格

    for(int i=0;i<=ROW;i++) { ... }

    for(int i=0;i<=COL;i++) { ... }
    int path_num;//有几条路径
    in>>path_num;
    for(int j=0;j<path_num;j++) { ... }
    int distant_num;//远战格子数目
    in>>distant_num;
    for(int j=0;j<distant_num*2;j++) { ... }
    addShop();addCard();
    connect(timer, &QTimer::timeout, scene2, &QGraphicsScene::advance);//为啥不用多线程的就不行
    connect(timer, &QTimer::timeout, this, &MainWindow::addZombie);
    connect(timer, &QTimer::timeout, this, &MainWindow::check);
    timer->start(50);
    view->show();
}

```

实现以上功能需要调用的是 showmap 函数，根据文件流的读入设置地图的行数列数和不同的格子的位置。

再接下来，我们需要在放置植物的时候，查看当前鼠标点击的位置是否可以放置，如果当前的格子超过了地图范围，或者当前格子的种类与想安放的植物种类不同，那么就说明不能放。

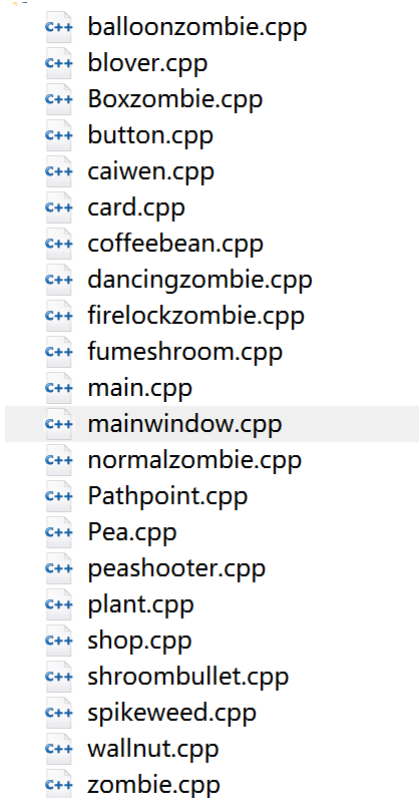
以默认地图为例：

```

-----
if(CUR_SRC == plant->cost && (scene2->itemAt(now_plantpos_x,now_plantpos_y,transform) == nullptr || scene2->itemAt(now_plantpos_x,now_plantpos_y,transform)->type() == Zombie::Type ||
{ //注意：我现改这里没法放植物，因为我这里给画格子之后，itemAt返回值不是nullptr了
    //这里要判断加的格子是在近战还是远战的格子
    int grid_type;//要放入的格子的种类，0近战，1远战，2不能放
    if(MAP_NUM == "默认地图")
    {
        //QDebug()<<x<<y;
        if((x==0&&y<=8&&y>=0) || (x==4&&y<=8&&y>=0))
            grid_type = 0;
        else if(x==1 && y==0)
            grid_type = 1;
        else
            grid_type = 2;
        if(plant->plant_type == grid_type)
        {
            //QDebug()<<"yes";
            plant->setPos(now_plantpos_x,now_plantpos_y);
            scene2->addItem(plant);
            CUR_SRC -= plant->cost;
        }
        else
            return;
    }
}

```

再接下来就是各种植物和僵尸种类的添加



这些植物都是 plant 的子类, 这些僵尸都是 zombie 的子类, 每个都有自己的不同属性。挺特别的是, 对于远程攻击植物, 我们设置了 pea 和 shroombullet, 这些远程植物会定时放出一定的子弹, 这些子弹也可以显示在屏幕上, 只有子弹打到僵尸, 才会造成伤害。

```
void Pea::advance(int phase)
{
    if(!phase)
        return;
    QList<QGraphicsItem *> items = collidingItems();
    if (!items.isEmpty())
    {
        Zombie *zombie = qgraphicsitem_cast<Zombie *>(items[qrnd() % items.size()]);
        zombie->hp -= atk;
        if(zombie->hp <= 0)
        {
            zombie->~Zombie();
        }
        delete this;
        return;
    }
    //飞离画面?
    setX(x() + speed);
    if(x()>1800)
        delete this;
}
```

## 二 . 整体效果

