

Plants Vs. Zombies Simplified Version

单博, 201220120

南京大学计算机科学与技术系

Plants Vs. Zombies Simplified Version

一、主要内容

二、实现目标

三、设计思路

1. 工具

2. 主要实现思路:

3. 具体实现过程:

1) 植物

相关代码

2) 僵尸

相关代码

3) 植物卡片

相关代码

4) 游戏场景

相关代码

四、功能亮点

五、操作方法

1. 游戏开始

2. 种植植物

3. 移除植物

4. 生成僵尸

5. 攻击僵尸

6. 抵抗入侵

7. 退出游戏

8. 其他功能

六、遇到的问题与解决方法

1. 使用MFC时遇到的困难

2. 安装Qt版本不当产生的困难

3. 频繁调用音效造成的主线程阻塞

4. 直接调用本地文件失败

一、主要内容

根据知名游戏《植物大战僵尸》，实现一个简化版本。

二、实现目标

在第二次课程设计的基础之上，将本项目拓展为GUI版本，并保证功能上有一定的增量，使其具有一定可玩性。

三、设计思路

1. 工具

- IDE: Qt Creator;
- Kits: Desktop Qt 5.12.11 MinGW 64-bit;

2. 主要实现思路：

1. 通过QGraphicsItem类, QGraphicsScene类, QGraphicsView类进行场景的建设，其中QGraphicsItem用于各种植物、僵尸以及卡片的实现，QGraphicsScene作为各图元的载体，而QGraphicsView用于呈现画面。
2. 使用状态机机制对游戏状态进行控制和转换。
3. 各图元之间以及图元与场景之间的交流使用Qt中的信号和槽机制实现。
4. 利用QTimer类实现场景中各部件的刷新和运动。
5. 利用Qt中的资源文件实现对图片、音效以及临时字体的管理与调用。
6. 利用QMovie类实现植物与僵尸GIF的播放。
7. 利用QSoundEffect类以及QSound类进行音效和背景音乐的播放，同时利用QThread类排除音效播放对主线程的阻塞。

3. 具体实现过程：

1) 植物

1. 实现一个继承自QObject与QGraphicsItem的类PlantBase作为所有植物的基类，添加一个表示植物类型的属性plantType以及各植物间相似的属性和方法，然后在各植物中针对不同功能实现具体的方法。
2. 属性：
 - 植物类型：int plantType;
 - 生命值：int hitPoint;
 - 攻击触发器：QTimer trigger;
 - 重绘触发器：QTimer repainter;
 - GIF播放器：QMovie plantGIF;
3. 方法：
 - 构造函数：PlantBase();
 - 边界矩形：QRectF boundingRect() const;
 - 右键删除：void contextMenuEvent(QGraphicsSceneContextMenuEvent* event);
 - 返回当前类型（主要用于区分植物、僵尸、子弹）：int type() const;
 - 返回当前植物类型（用于区分不同植物）：int getPlantType() const;
 - 减少血量：void reduceLife(int damage);
 - 返回是否是高坚果：virtual bool isTall() const;
4. 信号：
 - 攻击：void attack(PlantBase*);

- 删除: void deleteMe(PlantBase*);
5. 槽:
- 发送攻击信号: virtual void sendAttackSign();
 - 发送删除信号: void sendDeleteMe();
 - 重绘: void repaint();

相关代码

```
1  #ifndef PLANTBASE_H
2  #define PLANTBASE_H
3
4  #include <QGraphicsItem>
5
6  #include <QTimer>
7  #include <QMovie>
8  #include <QGraphicsSceneContextMenuEvent>
9
10 #define PLANT_HEIGHT 90
11 #define PLANT_WIDTH 90
12
13 #define PLANTTYPE_NULL -1
14 #define PLANTTYPE_SUNFLOWER 0
15 #define PLANTTYPE_PEASHOOTER 1
16 #define PLANTTYPE_REPEATER 2
17 #define PLANTTYPE_SNOWPEA 3
18 #define PLANTTYPE_WALLNUT 4
19 #define PLANTTYPE_TALLNUT 5
20 #define PLANTTYPE_SQUASH 6
21 #define PLANTTYPE_CHERRYBOMB 7
22 #define PLANTTYPE_GARLIC 8
23 #define PLANTTYPE_PUMPKIN 9
24
25 class PlantBase : public QObject, public QGraphicsItem
26 {
27     Q_OBJECT
28     // Q_PROPERTY(QRectF geometry READ geometry WRITE setGeometry)
29     // Q_INTERFACES(QGraphicsItem)
30 public:
31     PlantBase();
32 protected:
33     int hitPoint;
34     int plantType;
35     QTimer trigger;
36 public:
37     QRectF boundingRect() const;
38     void contextMenuEvent(QGraphicsSceneContextMenuEvent* event);
39     int type() const;
40     int getPlantType() const;
41     void reduceLife(int damage);
42     virtual bool isTall() const;
43 signals:
44     void attack(PlantBase*);
45     void deleteMe(PlantBase*);
```

```

46 public slots:
47     virtual void sendAttackSign();
48     void sendDeleteMe();
49     void repaint();
50 //private:
51 //    QRectF geometryRect;
52 //public:
53 //    QRectF geometry() const;
54 //    void setGeometry(const QRectF& value);
55
56 // Gif Display
57 protected:
58     QMovie plantGIF;
59     QTimer repainter;
60 };
61
62 #endif // PLANTBASE_H
63

```

2) 僵尸

1. 大致思路同植物;

2. 属性:

- 僵尸类型: int zombieType;
- 攻击触发器: QTimer trigger;
- 移动触发器: QTimer tick;
- 冰冻判定: bool isFrozen;
- 冰冻计时: QTimer frozenTime;
- 音效线程: QThread soundEffectThread;
- 攻击音效: QSoundEffect* chompSoundEffect[2];
- GIF播放器: QMovie zombieGIF;

3. 方法:

- 构造函数: ZombieBase();
- 析构函数: ~ZombieBase();
- 边界矩形: QRectF boundingRect() const;
- 返回当前类型 (主要用于区分植物、僵尸、子弹) : int type() const;
- 返回当前僵尸类型 (用于区分不同僵尸) : int getZombieType() const;
- 减少血量: virtual void reduceLife(int damage);
- 获取击倒僵尸应获得的分数: virtual void getPoint() const;

4. 信号:

- 删除: void deleteMe(ZombieBase*);
- 游戏结束: void gameOver();

5. 槽:

- 攻击: virtual void attack();
- 移动: virtual void zombieMove();
- 接触冰冻状态: void melt();
- 进入冰冻状态: void freeze();

相关代码

```
1  #ifndef ZOMBIEBASE_H
2  #define ZOMBIEBASE_H
3
4  #include <QGraphicsItem>
5
6  #include <QTimer>
7
8  #include <QThread>
9  #include <QSoundEffect>
10
11 #include <QMovie>
12
13 #define ZOMBIETYPE_NULL -1
14 #define ZOMBIETYPE_NORMAL 0
15 #define ZOMBIETYPE_CONEHEAD 1
16 #define ZOMBIETYPE_NEWSPAPER 2
17 #define ZOMBIETYPE_POLEVAULTING 3
18 #define ZOMBIETYPE_JACKINTHEBOX 4
19 #define ZOMBIETYPE_CATAPULT 5
20
21 #define ZOMBIE_HEIGHT 90
22 #define ZOMBIE_WIDTH 50
23
24 #define ZOMBIE_TICK 10
25 #define ZOMBIE_SPEED -0.08
26 #define ZOMBIE_ATK 100
27
28 #define ZOMBIE_FROZENTIME 10000
29
30 class ZombieBase : public QObject, public QGraphicsItem
31 {
32     Q_OBJECT
33 public:
34     ZombieBase();
35     ~ZombieBase();
36     QRectF boundingRect() const;
37     int type() const;
38     int getZombieType() const;
39     virtual void reduceLife(int damage);
40     virtual int getPoint() const;
41 protected:
42     int hitPoint;
43     int zombieType;
44     QTimer trigger, tick;
45     bool isFrozen;
46     QTimer frozenTime;
47 signals:
48     void deleteMe(ZombieBase*);
49     void gameOver();
50 public slots:
51     virtual void attack();
52     virtual void zombieMove();
```

```

53     void melt();
54     void freeze();
55
56     // Sound Effect
57 protected:
58     QThread soundEffectThread;
59     QSoundEffect* chompSoundEffect[2];
60
61     // Gif Display
62 protected:
63     QMovie zombieGIF;
64 };
65
66 #endif // ZOMBIEBASE_H
67

```

3) 植物卡片

1. 植物卡片PlantCard类继承自QObject类和QGraphicsItem类，用来进行所表示植物的选中以及相关判定。

2. 属性：

- 所表示植物类型：int plantType;
- 冷却判定：bool isReady;
- 冷却计时：QTimer coolDownTimer;
- 重绘触发器：QTimer repainter;
- 选中音效：QSoundEffect seedliftSoundEffect;
- 选中失败音效：QSoundEffect notenoughsunSoundEffect;
- 音效线程：QThread soundEffectThread;

3. 方法：

- 构造函数：PlantCard();
- 构造函数：PlantCard(int);
- 获取植物所需阳光：int getCost() const;
- 边界矩形：QRectF boundingRect() const;
- 绘制图元：void paint(QPainter* painter, const QStyleOptionGraphicsItem* option, QWidget* widget);
- 鼠标悬停事件：void hoverEnterEvent(QGraphicsSceneHoverEvent event);
- 鼠标点击事件：void mousePressEvent(QGraphicsSceneMouseEvent* event);
- 焦点离开事件：void focusOutEvent(QFocusEvent* event);

4. 信号：

- 点击：void clicked(int type);
- 播放选中音效：void sendSeedlift();
- 播放选中失败音效：void sendNotenoughsun();

5. 槽：

- 启动冷却计时器：void startTimer();
- 启动冷却计时器：void startTimer(int type);
- 冷却完毕：void getReady();
- 重绘：void repaint();

相关代码

```
1  #ifndef PLANTCARD_H
2  #define PLANTCARD_H
3
4  #include <QGraphicsItem>
5
6  #include <QTimer>
7  #include <QFocusEvent>
8
9  #include <QThread>
10 #include <QSoundEffect>
11
12 #define CARD_HEIGHT 60
13 #define CARD_WIDTH 81
14
15 #define PLANT_CNT 10
16
17 class PlantCard : public QObject, public QGraphicsItem
18 {
19     Q_OBJECT
20 public:
21     PlantCard();
22     PlantCard(int);
23     int getCost() const;
24     QRectF boundingRect() const;
25     void paint(QPainter* painter, const QStyleOptionGraphicsItem* option,
26               QWidget* widget);
27 public slots:
28     void startTimer();
29     void startTimer(int type);
30     void getReady();
31     void repaint();
32 signals:
33     void clicked(int type);
34 protected:
35     void hoverEnterEvent(QGraphicsSceneHoverEvent *event);
36     void mousePressEvent(QGraphicsSceneMouseEvent* event);
37     void focusOutEvent(QFocusEvent* event);
38 private:
39     int plantType;
40     bool isReady;
41     QTimer coolDownTimer;
42     QTimer repainter;
43
44     // Sound Effect
45 private:
46     QSoundEffect seedliftSoundEffect;
47     QSoundEffect notenoughsunSoundEffect;
48     QThread soundEffectThread;
49 signals:
50     void sendSeedlift();
51     void sendNotenoughsun();
52 };
```

53

54 **#endif** *// PLANTCARD_H*

55

4) 游戏场景

1. 作为各图元的载体，实现了继承自QGraphicsScene的游戏场景InterfaceScene类。

2. 属性：

- 游戏开始按钮：QPushButton gameStartButton;
- 游戏状态机：QStateMachine machine;
- 游戏状态：QState* gameStart, * gameRunning, * gamePausing;
- 终止状态：QFinalState* gameOver;
- 暂停对话框：QMessageBox* pauseMessageBox;
- 布局矩形：QRectF platformsRect, cardsRect, infoRect, sloganRect;
- 被选中植物：int selectedPlant;
- 植物卡片：PlantCard* cards[PLANT_CNT];
- 阳光：int sunlight;
- 阳光自然增加触发器：QTimer sunlightAdder;
- 阳光显示部件：QLabel sunlightDisplay;
- 分数：int point;
- 分数展示部件：QLabel pointDisplay;
- 僵尸生成向量：QVector <int> zombieProducer;
- 僵尸生成触发器：QTimer zombieProducerTick;
- 游戏音效：QSoundEffect* awoogaSoundEffect, pointsSoundEffect, setplantSoundEffect, shoopSoundEffect, notenoughsunSoundEffect, kernelpultSoundEffect, sirenSoundEffect;
- 音效线程：QThread soundEffectThread;
- 背景音乐：QSound* bgmSound;

3. 方法：

- 构造函数：InterfaceScene();
- 析构函数：~InterfaceScene();
- 键盘事件：void keyPressEvent(QKeyEvent* event);
- 添加植物：bool addPlant(const QPointF point);
- 鼠标单击事件：void mousePressEvent(QGraphicsSceneMouseEvent* event);
- 鼠标双击事件（调试用）：void mouseDoubleClickEvent(QGraphicsSceneMouseEvent* event);

4. 信号：

- 游戏结束：void gamelsOver();
- 游戏暂停：void gamelsPausing();
- 游戏继续：void gamelsContinuing();
- 关闭窗口：void closeView();
- 鼠标单击：void clicked();
- 植物添加成功：void plantAddedSuccessfully(int type);

5. 槽：

- 设置欢迎界面背景：void setCoverBackground();
- 设置游戏界面背景：void setInterfaceBackground();
- 呼出暂停消息框：void callPauseMessageBox();
- 发送游戏暂停信号：void sendGamePausingSignal();
- 发送游戏继续信号：void sendGameContinuingSignal();
- 发送游戏结束信号：void sendGameOverSignal();

- 发送关闭窗口信号: void sendCloseViewSignal();
- 选择植物: void selectPlant(int type);
- 植物攻击: void plantAttack(PlantBase* plant);
- 移除植物: void plantDelete(PlantBase* plant);
- 移除子弹: void bulletDelete(Bullet* bullet);
- 移除僵尸: void zombieDelete(ZombieBase* zombie);
- 增加阳光: void addSunlight();
- 更新阳光的显示: void updateSunlight();
- 添加子弹 (投石车僵尸) : void addBullet(QPointF zombiePos);
- 生产僵尸: void produceZombie();
- 打印状态 (测试用) : void printGameStartState();
- 打印状态 (测试用) : void printGameRunningState();
- 打印状态 (测试用) : void printGamePausingState();
- 打印状态 (测试用) : void printGameOverState();

相关代码

```

1  #ifndef INTERFACESCENE_H
2  #define INTERFACESCENE_H
3
4  #include <QGraphicsScene>
5
6  #include <QRectF>
7  #include <QTimer>
8  #include <QLabel>
9  #include <QPointF>
10 #include <QPushButton>
11 #include <QMessageBox>
12 #include <QGraphicsWidget>
13 //#include <QtStateMachine/QFinalState>
14 //#include <QtStateMachine/QStateMachine>
15 //#include <QtStateMachine/QHistoryState>
16 #include <QFinalState>
17 #include <QStateMachine>
18 #include <QHistoryState>
19
20 #include <QSound>
21 #include <QThread>
22 #include <QSoundEffect>
23
24 #include "plantcard.h"
25 #include "plantbase.h"
26
27 #include "bullet.h"
28
29 #include "zombiebase.h"
30
31 #define INTERFACE_HEIGHT 600
32 #define INTERFACE_WIDTH 900
33 #define STATE_GAMESTART 0
34 #define STATE_GAMERUNNING 1
35 #define STATE_GAMEPAUSING 2
36 #define STATE_GAMEOVER 3

```

```

37
38 #define PLATFORM_HEIGHT 90
39 #define PLATFORM_WIDTH 90
40
41 #define SUNLIGHT_INCREMENT 25
42 #define SUNLIGHT_INTERVAL 10050
43 #define SUNLIGHT_ICON_HEIGHT 40
44 #define SUNLIGHT_ICON_WIDTH 40
45
46 #define TYPE_PLANT 1
47 #define TYPE_BULLET 2
48 #define TYPE_ZOMBIE 3
49 #define TYPE_SPECIAL_GARLIC 4
50 #define TYPE_SPECIAL_PUMPKIN 5
51
52 #define BOX_WIDTH 700
53 #define BOX_HEIGHT 450
54
55 class InterfaceScene : public QGraphicsScene
56 {
57     Q_OBJECT
58 public:
59     InterfaceScene();
60     ~InterfaceScene();
61 public slots:
62     void setCoverBackground();
63     void setInterfaceBackground();
64     void callPauseMessageBox();
65     void sendGamePausingSignal();
66     void sendGameContinuingSignal();
67     void sendGameOverSignal();
68     void sendCloseViewSignal();
69 signals:
70     void gamelsOver();
71     void gamelsPausing();
72     void gamelsContinuing();
73     void closeView();
74     void clicked();
75 protected:
76     void keyPressEvent(QKeyEvent* event);
77 private:
78     QPushButton gameStartButton;
79     QStateMachine machine;
80     QState* gameStart, * gameRunning, *gamePausing;
81     QFinalState* gameOver;
82     QHistoryState* history;
83     QMessageBox* pauseMessageBox;
84
85 // Part Rects
86 private:
87     QRectF platformsRect, cardsRect, infoRect, sloganRect;
88
89 // Plant Cards
90 private:
91     int selectedPlant;

```

```

92     PlantCard* cards[PLANT_CNT];
93 public slots:
94     void selectPlant(int type);
95 signals:
96     void plantAddedSuccessfully(int type);
97 public:
98     bool addPlant(const QPointF point);
99 protected:
100     void mousePressEvent(QGraphicsSceneMouseEvent* event);
101
102 // Plants and Bullets
103 public slots:
104     void plantAttack(PlantBase* plant);
105     void plantDelete(PlantBase* plant);
106     void bulletDelete(Bullet* bullet);
107
108 // Zombies
109 public slots:
110     void zombieDelete(ZombieBase* zombie);
111
112 // Game Info
113 private:
114     int sunlight;
115     QTimer sunlightAdder;
116     QLabel sunlightDisplay;
117
118     int point;
119     QLabel pointDisplay;
120 public slots:
121     void addSunlight();
122     void updateSunlight();
123
124 // Catapult Zombie
125 public slots:
126     void addBullet(QPointF zombiePos);
127
128 // Zombie Producer
129 private:
130     QVector<int> zombieProducer;
131     QTimer zombieProducerTick;
132 public slots:
133     void produceZombie();
134
135 // Sound Effect
136 private:
137     QSoundEffect* awoogaSoundEffect;
138     QSoundEffect* pointsSoundEffect;
139     QSoundEffect* setplantSoundEffect;
140     QSoundEffect* shoopSoundEffect;
141     QSoundEffect* notenoughsunSoundEffect;
142     QSoundEffect* kernelpultSoundEffect;
143     QSoundEffect* sirenSoundEffect;
144     QThread soundEffectThread;
145 //signals:
146 // void sendAwooga();

```

```

147 // void sendPoints();
148 // void sendSetplant();
149 // void sendShoop();
150 // void sendNotenoughsun();
151 // void sendKernelpult();
152
153 // Background Music
154 private:
155     QSound* bgmSound;
156
157 // Debug Functions
158 private slots:
159     void printGameStartState();
160     void printGameRunningState();
161     void printGamePausingState();
162     void printGameOverState();
163 protected:
164 // void mousePressEvent(QGraphicsSceneMouseEvent* event);
165     void mouseDoubleClickEvent(QGraphicsSceneMouseEvent *event);
166 };
167
168 #endif // INTERFACESCENE_H
169

```

四、功能亮点

- 使用临时字体显示文字，使窗口更加美观；
- 使用GIF来显示植物与僵尸，使画面更加生动；
- 增加音效与背景音乐，使游戏更具可玩性；
- 利用右键菜单实现植物的移除，使得操作更加简洁；
- 增加历史记录积分榜，加强了游戏的竞争性；
- 制作游戏文件安装程序，完善发布过程。

五、操作方法

1. 游戏开始

游戏加载完成后，单击GAME START游戏开始。



2. 种植植物

1) 开始游戏后，各植物卡片开始冷却。



2) 植物冷却完毕后，可以单击植物进行选择，选中后单击其他区域（除草地区域）可以取消选中。



3) 单击草地中某一地块，如果此时阳光充足，植物种植成功。



附，植物花费阳光数一览：

植物种类	阳光花费
向日葵	50阳光
豌豆射手	100阳光
双发射手	200阳光
寒冰射手	175阳光
坚果墙	50阳光
高坚果	125阳光
倭瓜	50阳光
樱桃炸弹	150阳光
大蒜	50阳光
南瓜头	125阳光

3. 移除植物

右键点击植物呼出菜单，点击DELETE移除植物。



4. 生成僵尸

生成僵尸阶段一览：

- 1. 分数在0到5内只生成普通僵尸，此时生成僵尸速度为30s/只，生成速度下同；
- 2. 分数达到5分之后开始增加路障僵尸，其与普通僵尸生成的概率相等，即每30s生成一只普通僵尸或者路障僵尸，生成逻辑下同，即等概率生成每一种僵尸；

3. 分数达到10分之后开始增加读报僵尸和撑杆跳僵尸；
4. 分数达到50之后开始增加小丑僵尸，同时将生成僵尸的间隔修改为10s/只；
5. 分数达到70之后开始生成投石车僵尸。

5. 攻击僵尸

- 1) 种植攻击性植物，对僵尸进行攻击。



- 2) 击倒僵尸后获得相应分数。



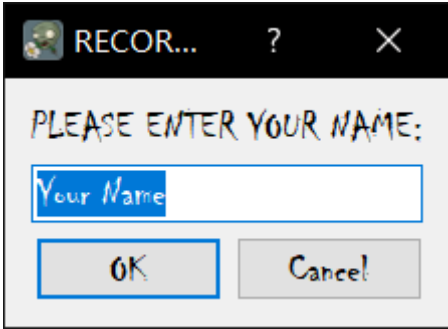
附，僵尸分数一览：

僵尸种类	对应分值
普通僵尸	1分
路障僵尸	5分
读报僵尸	7分
撑杆跳僵尸	15分
小丑僵尸	50分/0分
投石车僵尸	70分

注：小丑僵尸自然引爆时不得分，如果在引爆前击倒小丑僵尸可以获得50分。

6. 抵抗入侵

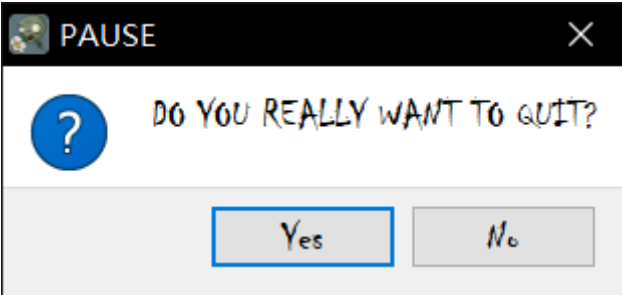
抵抗僵尸的入侵，如果僵尸抵达终点线（游戏左侧的边界），则游戏失败。关闭窗口前将弹出对话框，记录成绩后显示排行榜。



7. 退出游戏

按空格键呼出对话框，单击 “Yes” 键以退出游戏，记录成绩并显示排行榜后将关闭窗口。

注意：此时游戏并未暂停，单击 “No” 键可以返回进行中的游戏。



8. 其他功能

按 “0” 键关闭背景音乐，再次按 “0” 键可以开启背景音乐。

六、遇到的问题与解决方法

1. 使用MFC时遇到的困难

起初对于这次课程设计我选用的是MFC来作为图形化的工具，但是在中途设计植物与僵尸的行为时遇到了障碍，发现MFC似乎不能很好的支持这样类型的实现，于是转战Qt。通过大约三天的学习与实操练习，在大致掌握Qt中QGraphicsItem、QGraphicsScene、QGraphicsView等与本次课程设计紧密相关的类之后，开始着手使用Qt实现本次设计。

2. 安装Qt版本不当产生的困难

出于对最新版的盲目追求，在未经了解的情况下选择下载Qt 6.0，但是由于Qt 6版本升级时做出了较大的更改，对于QStateMachine以及QMovie等类的使用与我了解的内容出现较大偏差，尤其是QMovie和QSoundEffect等类甚至在Qt 6.0 中被移除，需要等待Qt 6.2才能再次加入。在寻找替代方案失败后，我重新安装了Qt 5.12，继续未完成的设计。

3. 频繁调用音效造成的主线程阻塞

调用QSoundEffect类中的moveToThread方法，将其转移到另外的线程中运行。

4. 直接调用本地文件失败

选择Qt的资源文件，将要调用的字体、音效等文件添加到资源文件即可很方便的使用本地文件。