

## Changelog0614

周昊男 2019202458

(注：这个 changelog 按照每个.h 头文件和.cpp 资源文件的顺序来记录这个版本相对上个版本的改动)

### 一、新增功能

- 新增了 soundcontrol 类，加入了背景音乐，相应的事件（选中要建的塔的类型、防御塔发射子弹、敌人被击毙、敌人到达基地使得玩家 Hp 降低）也加入了相应的音效作为提示
- Tower、Bullet 类均实现了继承，IceTower（对敌人造成冰冻伤害，一定时间的减速）和 PoisonTower（对敌人造成中毒伤害，一段时间的持续扣血）均继承于 Tower 类，对应的 IceBullet 和 PoisonBullet 均继承于 Bullet 类，击中敌人后分别触发冰冻和中毒效果
- 增加了攻击塔的种类，可在游戏界面左下方"PLEASE SELECT TOWER HERE!"区域选择防御塔类型（选中会有音效提示）并在空白的 TowerPosition 位置建造三种不同属性和攻击效果的防御塔
- 为攻击塔添加了升级和拆除按钮，这两个功能完成了一部分，还没有完全实现

### 二、修复的 bug

- 解决了上一个版本中未进入游戏界面游戏就自动开始的问题，现在在开始界面点击"START"按钮后才会触发第一关游戏的开始

### 三、具体改动内容

(注：增加用+表示，删除用-表示，修改用#表示)

(注：由于我在写程序的过程中相较于第一版程序有变动的地方都作了//+标识，而函数具体的改动实在是太繁琐了，故在这里只体现了做了改动或添加的函数的名称，具体实现的修改可通过程序中//+标识具体看到)

#### Bullet.h

```
+ class IceBullet:public Bullet //冰冻子弹类
+ IceBullet(QPoint startPos, QPoint targetPoint, int damage, Enemy *target,
            MainWindow *game, const QPixmap &sprite =
            QPixmap(":/image/icebullet.png"));
+ ~IceBullet(); //构造函数和析构函数
+ void hitTarget(); //槽函数：击中敌人时与敌人相连接，使得敌人受到伤害
            //同时改变被击中 enemy 的 m_isSlowed

+ class PoisonBullet:public Bullet //中毒子弹类
+ PoisonBullet(QPoint startPos, QPoint targetPoint, int damage, Enemy *target,
            MainWindow *game, const QPixmap &sprite =
            QPixmap(":/image/poisonbullet.png"));
+ ~PoisonBullet(); //构造函数和析构函数
+ void hitTarget(); //槽函数：击中敌人时与敌人相连接，使得敌人受到伤害
            //同时改变被击中 enemy 的 m_isPoisoned
```

#### Bullet.cpp

```

//IceBullet 和 PoisonBullet 的构造函数和析构函数
+ IceBullet::IceBullet(QPoint startPos, QPoint targetPoint, int damage, Enemy *target,
                      MainWindow *game, const QPixmap &sprite/* =
                      QPixmap(":/image/icebullet.png")*/)
    : Bullet(startPos,targetPoint,damage,target,game,sprite)
{}
+ IceBullet::~IceBullet()
{}
+ PoisonBullet::PoisonBullet(QPoint startPos, QPoint targetPoint, int damage, Enemy
*target,
                      MainWindow *game, const QPixmap &sprite/* =
                      QPixmap(":/image/poisonbullet.png")*/)
    : Bullet(startPos,targetPoint,damage,target,game,sprite)
{}
+ PoisonBullet::~PoisonBullet()
{}
//IceBullet 和 PoisonBullet 的 hitTarget 方法（虚函数重载）
+ void IceBullet::hitTarget()//扣血的同时造成冰冻伤害
+ void PoisonBullet::hitTarget()//扣血的同时造成中毒伤害

```

## Button.h/Button.cpp 无变动

### Enemy.h

```

Public:
# virtual int  getCurrentHp() const; //判断是被打死了还是走到基地了，加声音
+ void slowDown(); //被 icebullet 击中，修改 m_isSlowed 的值为 1
+ void getPoisoned(); //被 poisonbullet 击中，怪兽一段时间的持续伤害
Public slots:
+ void SpeedUp(); //槽函数，修改 m_isSlowed 的值为 0
+ void CurePoison(); //槽函数，修改 m_isPoisoned 的值为 1
+ void getPoisonDamage(); //槽函数，中毒持续伤害
Protected:
+ bool m_isSlowed=0; //被 icebullet 击中，怪兽一段时间的减速 //默认为 0，被击中后一
段时间此值为 1，造成冰冻减速
+ bool m_isPoisoned=0; //被 poisonbullet 击中，怪兽一段时间的持续伤害 //默认为 0，被
击中一段时间后此值为 1，造成持续中毒伤害

```

### Mainwindow.h

```

Public:
+ void drawOtherTexts(QPainter *painter); //其他文字信息
+ SoundControl * soundcontrol() const; //音效列表

Private:
+ void loadTowerSelectButtons(); //初始化选塔按钮

```

```
+ int m_currentTowerMode=1; //通过这个变量来确定安放的不同塔
+ SoundControl * m_soundcontrol; //音效列表
```

### SoundControl.h

```
+ enum SoundType//枚举类型的数组
{
    PlantTower,          //建塔
    SellTower,           //拆塔
    UpgradeTower,        //升级塔
    SelectTower,         //选中某一种塔
    LifeLoss,            //怪物到基地，掉血
    Bulletshot,          //子弹发射
    EnemyKilled          //怪物击毙
};
+ class SoundControl: public QObject
Public:
+ explicit SoundControl(QObject *parent = 0); //构造函数
+ void startBGM(); //背景音乐
+ void playSound(SoundType soundType); //事件触发各种音效
Private:
+ QMediaPlayer *m_backgroundMusic; // 只用来播放背景音乐
```

### Tower.h

```
Public:
# Tower(QPoint pos, MainWindow *game, const QPixmap &sprite =
QPixmap(":/image/tower1.png")); //构造函数 //增加了升级和拆塔的按钮，做了修改
# virtual void draw(QPainter *painter) const; //增加了升级和拆塔的按钮，做了修改
+ int getTowerType(); //不同类型的塔
//升级和拆除
//本来尝试以 button 形式实现，不知道为什么无法加载出 button，所以改用点击相应位置
//建塔的思路实现（结合 mousepressevent）
+ virtual void setUpgradeTower(); //升级后更新塔的等级和攻击能力参数
+ virtual bool isMaxlevel() const; //该处的塔是否满级，还可以升级否
+ const QPoint upgradebuttoncenterPos() const; //升级塔按钮的中心点
+ const QPoint sellbuttoncenterPos() const; //卖塔按钮的中心点
+ virtual bool containupgradebutton(const QPoint &pos) const; //是否点到了升级按钮上
+ virtual bool containsellbutton(const QPoint &pos) const; //是否点到了卖塔按钮上
Protected:
+ const QPixmap m_upgradebutton; //升级按钮的图片
+ const QPixmap m_sellbutton; //拆除按钮的图片
+ static const QSize ms_buttonfixedSize; //按钮的尺寸

+ class IceTower: public Tower //冰冻塔，继承于 Tower
Public:
```

```

//构造函数和析构函数
+ IceTower(QPoint pos, MainWindow *game,const QPixmap &sprite =
QPixmap(":/image/tower2.png"));
+ ~IceTower();
+ void draw(QPainter *painter) const;//因为图片资源的原因，三个塔的中心点不一样，所以 draw()也要重载
Public slots:
void shootWeapon();//子弹变化，所以需要重载
protected:
int m_towerType=2;//冰冻塔类型标识

+ class PoisonTower:public Tower //中毒塔，继承于 Tower
Public:
//构造函数和析构函数
+ PoisonTower(QPoint pos, MainWindow *game,const QPixmap &sprite =
QPixmap(":/image/tower3.png"));
+ ~PoisonTower();
+ void draw(QPainter *painter) const;//因为图片资源的原因，三个塔的中心点不一样，所以 draw()也要重载
Public slots:
+ void shootWeapon();//子弹变化，所以需要重载
protected:
int m_towerType=2;//冰冻塔类型标识

```

**Towerposition.h 未修改**

**Utility.h 未修改**

**Waypoint.h 未修改**