

高级语言程序设计

实验报告

南开大学 计算机大类

姓名 张一萌

学号 2313636

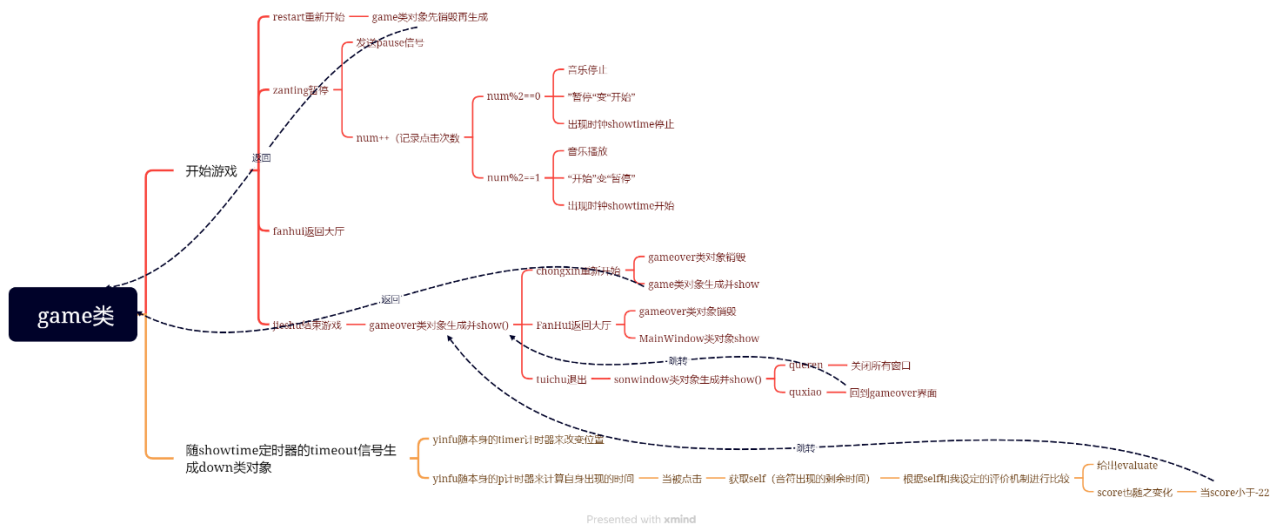
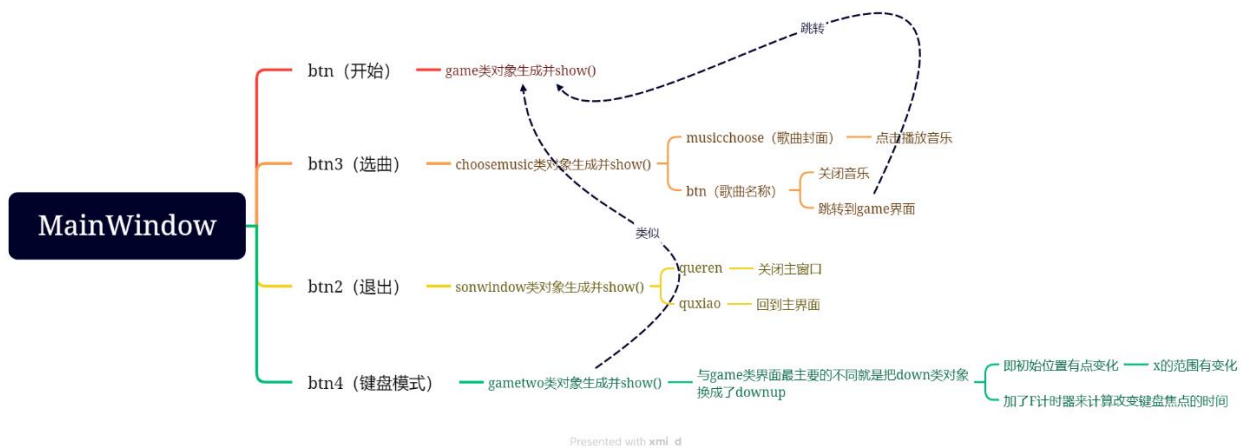
班级 0927

2024 年 5 月 14 日

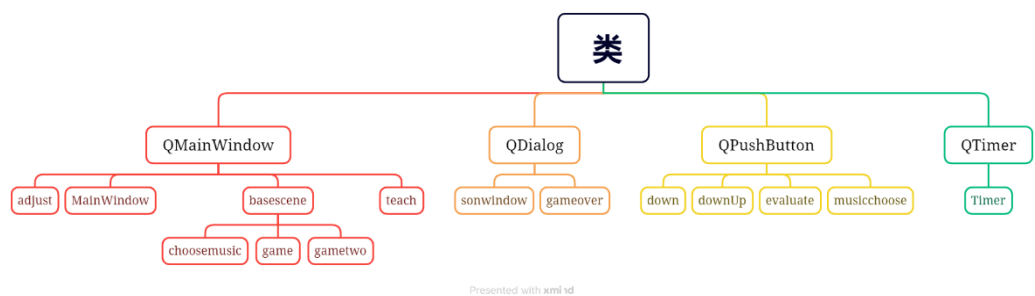
目录

- 一、 作业题目2
- 二、 开发软件2
- 三、 课题要求2
- 四、 主要流程2
 - 1. 整体流程2
 - 2. 各种类的相互关系3
 - 3. 一些重要函数的实现3
- 五、 单元测试6
- 六、 收获7





2. 各种类的相互关系



3. 一些重要函数的实现

- 有关音符的位置
 - down 类
 - //出现计时器
 - p=new Timer(this);

```

    p->setParent(this);
    p->setSingleShot(true); //表示只计时一次
    emit p->Start(500*xia+Time);
    //判断时间
    connect(p,&Timer::timerTimeout,[=]() {
        emit down::miss();
    });

void down::changeLocation()
{
    x = QRandomGenerator::global()->bounded(40,800);
    y = 50;
    this->move(x,y);
    //设置初始位置
    //音符位置要随时间的变化而变化

    //位置计时器
    timer = new Timer(this);
    timer->setParent(this);
    emit timer->Start(downSpeed);

    //改变音符位置
    connect(timer,&Timer::timerTimeout,[=]() {
        this->move(x,++y);
    });
}

```

downUp 类

```

void downUp::changeLocation()
{
    x = QRandomGenerator::global()->bounded(1,9);
    y = 50;
    this->move(6+82*(x-1)+40*x,y);
    //设置初始位置
    //音符位置要随时间的变化而变化
    //位置计时器
    timer = new Timer(this);
    timer->setParent(this);
    emit timer->Start(downSpeed);

    //改变音符位置
    connect(timer,&Timer::timerTimeout,[=]() {
        this->move(6+82*(x-1)+40*x,++y);
    });
}

```

```

    }
(2) 信号与槽的连接函数
void MainWindow::connectMainWindow(choosemusic*c)
{
    connect(btn,&QPushButton::clicked,[=]() {
        game* music2=new game(":/Kirara Magic -
Dragonflame.wav",this);
        music2->setParent(this);
        connectGameWindow(music2,c);
        push->play();
        background->stop();
        QTimer::singleShot(400,music2,[=]() {
            music2->show();
            emit music2->showScene();
        });
        this->hide();
    });

    connect(btn2,&QPushButton::clicked,[=]() {
        push->play();
        sonwindow*choose=new sonwindow(this);
        choose->setParent(this);
        connectSonWindow(choose,this);
        QTimer::singleShot(400,choose,[=]() {
            choose->show();
        });
    });

    connect(btn3,&QPushButton::clicked,[=]() {
        //延时 0.5 秒后， 进入选择场景
        push->play();
        this->hide();
        QTimer::singleShot(500,this,[=]() {
            c->show();
        });
    });

    connect(btn4,&QPushButton::clicked,[=]() {
        gametwo* music1=new gametwo(":/夢幻.wav",this);
        music1->setParent(this);
        connectGameWindow(music1,c);
        push->play();
        background->stop();
        QTimer::singleShot(400,music1,[=]() {

```

```

        music1->show();
        emit music1->showScene();
    });
    this->hide();
});
}

```

等……

- (3) 派生的计时器类 Timer 的自我管控的实现（信号）

```

Timer::Timer(QObject*parent)
    : QTimer{parent}
{
    //定时任务放在线程中执行，必须指定参数
    connect(this,&Timer::Start,this,&Timer::startN);
    connect(this,&Timer::Stop,this,&Timer::stopN);
}
void Timer::startN(int msec)
{
    this->start(msec);
    connect(this, &Timer::timeout,[=]() {
        emit this->timerTimeout();
    });
}
void Timer::stopN()
{
    this->stop();
}
}

```

五、单元测试

- 对 MainWindow 界面的按钮点击引起的页面跳转进行测试。
- 对 MainWindow 关联的 sonwindow 界面的生成进行测试
- 对 MainWindow 关联的 sonwindow 界面的按钮效果进行测试
- 对 QSoundEffect 类对象的播放与停止进行测试。
- 对 game 界面的生成进行测试
- 对选曲界面的生成进行测试
- 对选曲界面中按钮的点击效果进行测试
- 对选曲界面中按钮点击后 game 界面的生成进行测试
- 对游戏界面音符的下落位置进行测试
- 对游戏界面的音符块的下落与点击以及键盘点击的响应进行测试。
- 对游戏界面的按钮进行测试。
- 对分数判定的效果进行测试
- 对评价的出现时间与持续时间进行测试
- 对 gameover 界面的生成与关闭进行测试

对 gameover 界面的按钮进行测试
对 gameover 关联的 sonwindow 界面的生成进行测试
对 gameover 关联的 sonwindow 界面的按钮效果进行测试

对调整界面的生成进行测试
对教学界面的生成进行测试
对调整界面 QSpinBox 和 QSlider 的同步效果进行测试
对音符的下落速度与出现速度与调整界面的同步性进行测试
对调整界面中调整的持续可见性进行测试。
对程序能否正常关闭进行测试。

六、收获

函数模板的使用

```
template <typename T>void generateDownObject();  
template<typename T>void connectChooseWindow(choosemusic*c);
```

重载函数的使用

```
void connectSonWindow(sonwindow*choose,QMainWindow*parent);  
void connectSonWindow(sonwindow*choose,gameover*parent,game*music1);  
void connectSonWindow(sonwindow*choose,gameover*parent,gametwo*music1);
```

```
void connectGameOver(game*music1,gameover*over,choosemusic*c,int score);  
void connectGameOver(gametwo*music1,gameover*over,choosemusic*c,int score);
```

友元类的使用

Private

静态变量的重要作用

```
private:  
    static int xiaSpeed;  
    static int chuSpeed;  
    static int TimeD;  
    friend class game;  
    friend class gametwo;
```

函数的嵌套和递归

函数模板中模板参数 T 在递归函数中的默认问题，及其引发的无法进行类型转换的问题

虚基类的注意事项

类的继承与派生中，同名函数的选择问题：基类的函数中包含基类的另一函数，若在子类中调用基类的这一函数，则其中被包含的函数会默认使用基类的同名函数，即便子类中也有这一同名函数，但在这一继承自基类的函数中不会被使用。

包含纯虚函数的基类是虚基类，无法生成类对象

虚函数的选择问题：virtual 关键字的使用问题

虚析构函数的作用

类的继承和派生的使用

多文件编写的注意事项：避免头文件的相互包含

多文件编写的好处

QTimer 无法跨线程打开和关闭的问题：可使用信号和槽实现让定时器自己根据信号来控制自己的打开和关闭，而不是让其他线程的对象控制计时器。

信号和槽的使用

自定义信号和槽的使用