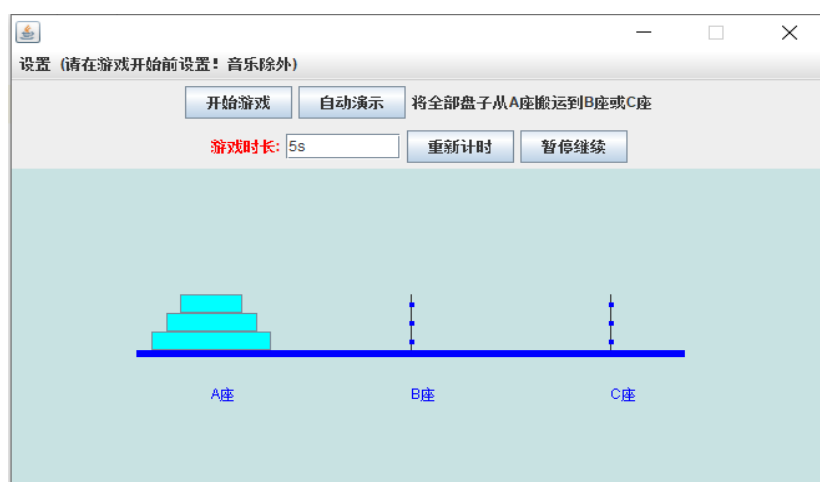


课程设计报告

一、课程设计要求

本项目名称为汉诺塔，要求最终需要实现以下几个效果：

1. 设计总体的 UI 界面，要求有三个塔以及塔针，在每个塔针上有若干塔点，于塔点上按照顺序放置盘子，然后可以通过鼠标对盘子进行移动拖放。
2. 大盘必须放在小盘子下面，一次只能移动一个盘子，最重要将所有盘子移动到 B 或者 C 塔上，并且要计算所需要的时间。
3. 用户可以根据自身的要求设置背景颜色，盘子颜色以及盘子的大小和数量，分为不同等级，难度依次提升。
4. 添加自动演示，可以将盘子自动的移动到 C 塔上，并且要动画效果，一秒演示一个步骤。
5. 在移动盘子的过程中，按下按钮重新计时以及并重新进行设置。
6. 添加音乐播放器，可以实现音乐的播放和暂停。
7. 使用文件 IO 流，将自动演示生成的搬运方法文本添加到 TEXT 文本当中。
8. 增加计时功能，在游戏完成后，自动停止计时，点击开始游戏，游戏重新开始，计时也同时重新计时。



在本设计中，全部的设置放在一个总的设置中，玩家可以在设置中找到想更改的项目。

二、总体设计

1 首先设置 GUI 界面的汉诺塔，要求每个塔的下面都需要标有名称，绘制一条水平线作为塔的底座，长度略大于盘子的大小，使用自由布局来实现拖拽过程中盘子位置的重新设置。只有最上面的盘子可以被拖动，假如说盘子没有放置到合适的塔点上时，松开鼠标，盘子将会回到之前的位置。

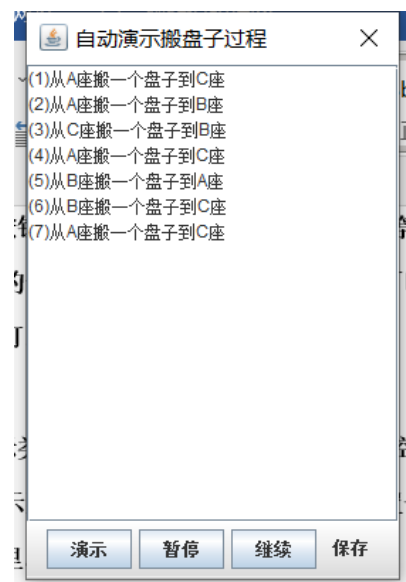
2 移动的规范是只能大盘子在小盘子上面，而且多个盘子叠加的时候不可以拖拽下面的盘子，最终的任务是将盘子全部移动到 B 或者是 C 塔上面。

3 在设置按钮里面选择背景颜色，盘子大小，等级以及其他的设置，采取边界布局管理器管理主页面的试图，中间是主要的操作区域，可以执行盘子的放置，本人在上面添加了一个计时器，可以动态的显示时间的流逝。

4 自动演示类的作用是弹出一个对话框，可以控制自动演示的开始暂停，并且可以将需要的步骤显示到对话框里面。增加的功能是设置一个按钮将文本区域里的文本放到电脑中的一个文档里。

5 在移动盘子的过程中，可以设置游戏的重新开始，此时计时器也会刷新，并且可以再设置游戏的其他参数，比如页面颜色，盘子颜色等。

6 音乐播放器可以播放复制到项目文件夹中的文件，设置音乐的播放和暂停。本人增加了循环播放列表歌曲的功能。



根据以上的要求，总共写出 6 个类，分别为：

主类（HanoiWindow.java）设置主窗口的底层容器，同时实现监听器功能，负责处理有关页面以及组件参数的设置，其中使用文本框、菜单选项等组件，成员变量主要用于接受对象以及设置组建的具体参数。同时具有计时和播放音乐的两个子线

程存在。

Tower 类是主类采取边界布局后，放在中间的面板，主要的操作再其中执行，调用绘图类在其中绘制出主要的页面外观，Disc 类和 TowerPoint 类放在其中作为成员变量，Tower 类并不处理事件，所以需要单独的设置一个事件监听器类对于鼠标事件进行处理。

TowerPoint 类在页面中不可见，可以称其为塔点类，它获取 tower 类中绘制的若干填充圆形的中心坐标，在获取相应容器后执行组件也就是盘子的放置，并且需要获取自身有没有盘子的 Boolean 值，方便后面实现盘子的移动规范以及检测胜利与否。

Disc 类也就是盘子类，继承自 JButton，在构造方法当中实现了大小的初始化，为了方便进行盘子大小的比较，每个盘子都设置了相应的大小参数，分别为 1, 2, 3... 盘子的，此外，为了实现封装性，需要将盘子的参数保存在 Tower 类当中，方便以后进行的一系列设置。

HandleMouse 类实现两个监听器，处理鼠标拖动和鼠标选择释放的事件，获取组建的外边界后，如果组件遮盖了塔点类的坐标而且放置符合规范，就会将盘子放置到其上。

AutoMoveDisc(自动演示类)类实现盘子移动的自动演示，处理文本区域内的文档，将其输出。

三、详细设计

1. 增加计时功能：

```
else if (Thread.currentThread() == gg) {  
    this.youXiMiaoShu = 0;  
    while (true) {
```

```

        try {
            gg.sleep(1000);
            this.youXiMiaoShu = this.youXiMiaoShu + 1;
            wenBen.setText(youXiMiaoShu + "s");
            if (tower.youWin()) {
                gg.stop();
            }
        }

```

此为计时的主要执行体

```

else if (e.getSource() == kaiShiJiShi) { //计时代码
    if (gg.isAlive()) {
        this.gg.stop();
    }

    gg = new Thread(this);
    gg.start();

} else if (e.getSource() == jieShuJiShi) {
    zanting+=1;
    if (zanting%2==0) {
        this.gg.suspend();
    } else {
        this.gg.resume();
    }
}

```

此为时间的暂停等操作;



2. 盘子。设置大小，颜色，数目

```

panZiShuMu = new JMenu("盘子数目");
zongSheZhi = new JMenu("设置");
panZiYanSe = new JMenu("盘子颜色");
beiJingYanSe = new JMenu("背景颜色");
three = new JMenuItem("3");
four = new JMenuItem("4");
five = new JMenuItem("5");
six = new JMenuItem("6");

```

```

seven = new JMenuItem("7");
eight = new JMenuItem("8");
red = new JMenuItem("红色");
blue = new JMenuItem("白色");
green = new JMenuItem("绿色");
yellow = new JMenuItem("黄色");
red1 = new JMenuItem("红色");
blue1 = new JMenuItem("黑色");
green1 = new JMenuItem("绿色");
yellow1 = new JMenuItem("黄色");
D1 = new JMenuItem("一级大");
D2 = new JMenuItem("二级大");
D3 = new JMenuItem("三级大");
X1 = new JMenuItem("一级小");
X2 = new JMenuItem("二级小");
X3 = new JMenuItem("三级小");

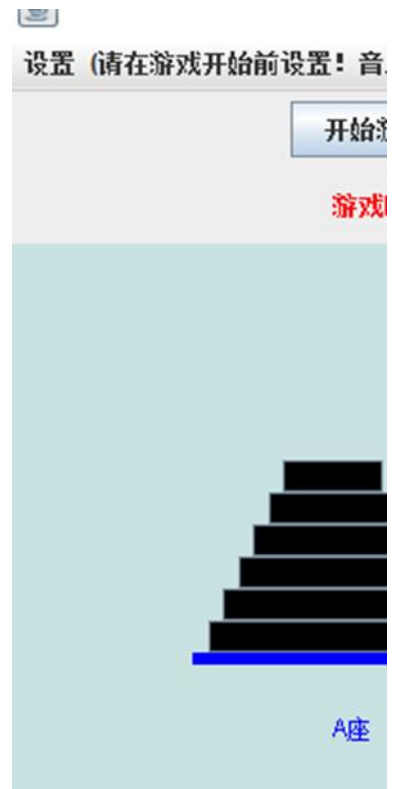
if (e.getSource() == three) { //盘
    子个数控制代码（监听
        amountOfDisc = 3;
        tower.setAmountOfDisc(amountOfDisc);
        tower.putDiscOnTower();

    else if (e.getSource() == green1)
    {
        //盘子颜色（由于避免蓝色字体冲突，蓝变黑

        tower.setbb(Color.green);
        tower.putDiscOnTower();

    else if (e.getSource() == D2) {
        tower.setMaxDiscWidth(120);
        setBounds(60, 60, 660, 410);
        tower.putDiscOnTower();

```



3. 播放和暂停背景音乐

```
Thread ff, gg;
```

```

ff = new Thread(this);
gg = new Thread(this);

ff.start();
gg.start();

public void run() {                                     //播放音乐和计时两个线程
    if (Thread.currentThread() == ff) {
        byte[] aubuffer = new byte[1024 * 128];
        do {
            AudioInputStream audioInputStream = null;
            SourceDataLine auline = null;
            try {
                audioInputStream =
AudioSystem.getAudioInputStream(
                    new
File("D:\\Ideajava\\Swwing\\src\\zuoye\\bgm.wav "));
                AudioFormat format =
audioInputStream.getFormat();
                DataLine.Info info = new
DataLine.Info(SourceDataLine.class, format);
                auline = (SourceDataLine)
AudioSystem.getLine(info);
                auline.open(format);
                auline.start();
                int byteCount = 0;
                while (byteCount != -1) {
                    byteCount =
audioInputStream.read(aubuffer, 0, aubuffer.length);
                    if (byteCount >= 0) {
                        auline.write(aubuffer, 0, byteCount);
                    }
                }

                } catch (UnsupportedAudioFileException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                } catch (LineUnavailableException e) {
                    e.printStackTrace();
                } finally {
                    auline.drain();
                }
            }
        }
    }
}

```

```

        auline.close();
    }
} while (true);

}

else if (e.getSource() == yinYve1) {           //音乐开关代码
    if (!ff.isAlive()) {
        this.ff = new Thread(this);
        ff.start();
    }
} else if (e.getSource() == yinYve2) {
    ff.stop();
}

```

4. 将移动盘子的文本信息保存到文件

```

File file =new File("a.txt");

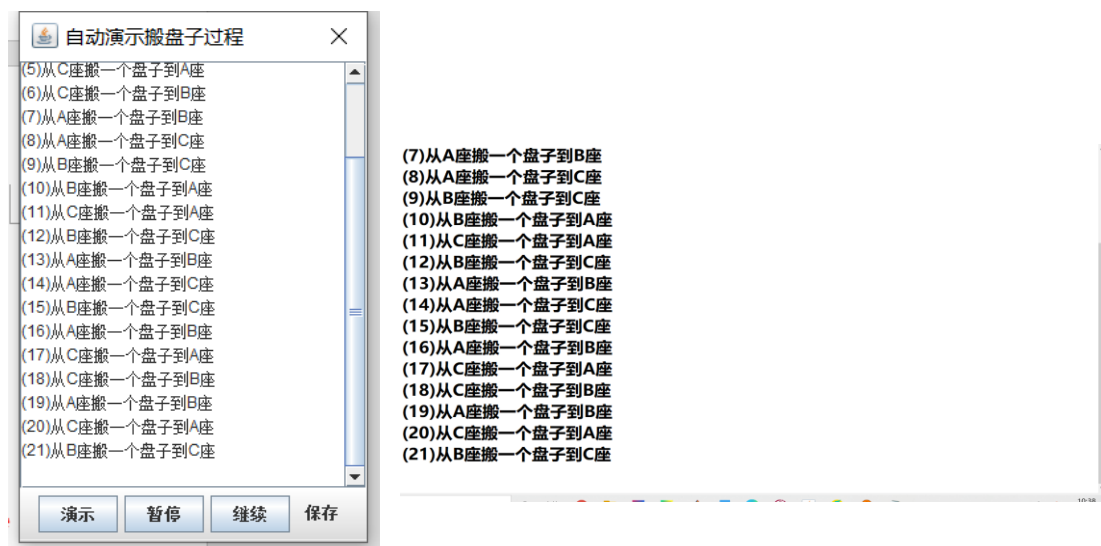
item=new JMenuItem("将文本保存到 a.txt");

else if (e.getSource() == item) {           //把关闭改成了保存
    time.stop();

    byte fileContent[] = showStep.getText()    //读取 JArea 里面
    的文本信息，下面为写入文件
        .getBytes();
    try {
        OutputStream out =new FileOutputStream(file);
        out.write(fileContent);
        out.close();
    } catch (FileNotFoundException fileNotFoundException)
    {
        fileNotFoundException.printStackTrace();
    } catch (IOException ioException) {
        ioException.printStackTrace();
    }

    //setVisible(false);
}

```



四、心得体会

转眼之间，Java 程序设计课程已结束，虽时间仓促，但收获甚多。

本人于五一假期期间完成大部分课设题目的解答。

此设计共有 6 题，每一题都花了半天左右的时间才在书本和网络中找到比较满意的解答。

经验：

首先我把代码看了一遍之后，对代码的结构及原理有了大致的一点了解。便从比较容易的 4, 5, 两题开始下手，一天过去，两题终于写完，期间看了很多次书和网上的资料。这个过程其实也不很容易。比如在盘子大小改变的时候，本来的空间可能就不够用了，于是在多次调参后，才确定最终的界面大小。

到最后一问，（第一题）确实我认为最困难的，时间的显示方法，及最后的结束状态都不好控制。最后用了数字休眠加一的方法才加以解决。

但一个人做事还是容易分心，以至于在干活之前的准备时间太长，时间浪费了好多。

教训：

做事应专心，做自己该做的事，对周围的花花世界持有一种淡然的心态，去追逐真正的自我所爱。

Java 课程建议开的多一点，这样没学过 c 语言的同学更容易跟上课些。学过的同学也能对 java 编程的思想有更加深入的体会。

最后，感谢组员的帮助及老师的指导。

改进本程序的 Hanoi 塔程序，增加一些新的功能，具体要求如下：
(1) 增加计时功能，即显示用户完成移动盘子所花费的时间。

Java 课程设计（第2版）

- (2) 用户可以设置最大和最小盘子的大小。
- (3) 用户可以选择播放和暂停背景音乐。
- (4) 用户可以设置盘子的数目。
- (5) 用户可以设置盘子的颜色以及背景的颜色。
- (6) 用户可以将自动移动盘子的文本信息保存到文件。