编号	25

浙江经贸职业技术学院

实验实训报告

（职业技能训练课）

2020 / 2021 学年第 二 学期

课程名称： Java 基础开发实训

班　　级： 软件技术 203

学　　号： 20202030319

姓　　名： 王晨冰

分组名单： 王晨冰 周增远 王家辉

一、课程设计（实训）目的及要求

《Java 基础开发实训》是三年制高职软件技术专业的一门实践实训课程。教育目标是在学生学习了《Java 语言程序设计》课程以后进行实践操作与训练。使学生在掌握在面向对象程序设计的基本知识、基本理论的基础上，通过本次实训，培养学生实践操作的基本技能，掌握面向对象程序设计和 Windows 程序设计的方法。同时，提高学生分析问题解决问题的能力，为今后软件开发打下必要的基础。

课程设计（实训）相关要求：

1、学生实训以小组为单位，小组人数以 4 人为宜。

2、学生实训时认真做好小组分工和实训计划，并完成实训报告、答辩用 PPT、源程序代码（同时提供系统安装操作说明）。

3、积极参与课程实训，按质按量完成课程实训要求，成绩考核评定包含如下几个部分：

(1) 出勤情况，占 20%.

(2) 在项目开发过程中完成的工作量，占 30%.

(3) 对开发的项目涵盖的知识点的掌握情况，占 40%。

(4) 项目中有新功能或创意，占 10%。

二、设计思路

1. 需求分析：

2. 客户针对这个项目提出了一些需求，在下面列出：

1） 有两个账户：管理员账户 用户账户

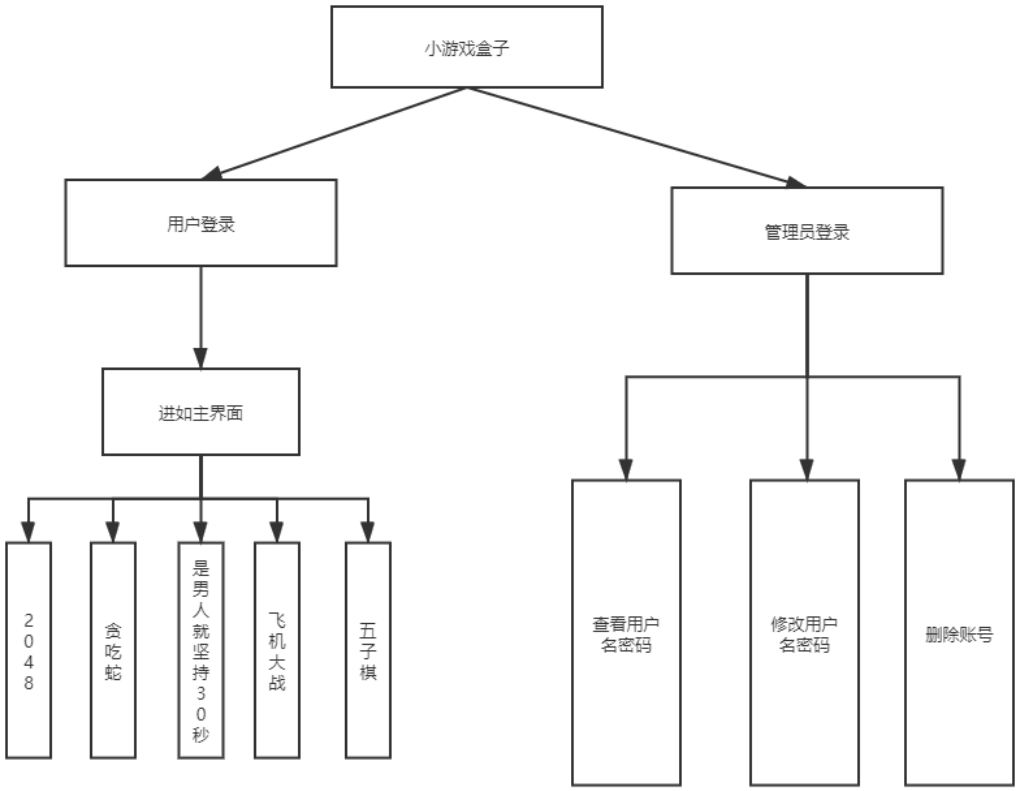
2） 几个单机小游戏使用户得到放松

3） 有一个登录界面、选择游戏界面及各小游戏界面

4） 系统对用户游戏的记录进行排名

5） 管理员账户能够更改所有用户的排名信息以及账户信息。

3. 功能介绍：



1）普通用户登录

a、进入 2048 模块；

b、进入贪吃蛇模块；

c、进入是男人就坚持 30 秒模块；

d、进入飞机大战模块

e、进入五子棋模块

2）管理员登录

a、对用户账号密码的查看；

b、对用户账号密码的修改；

c、对用户账号密码的删除；

模块划分：

登录界面及注册界面：周增远

管理员界面：周增远、王家辉

主界面：王晨冰

2048：周增远

飞机大战：王晨冰

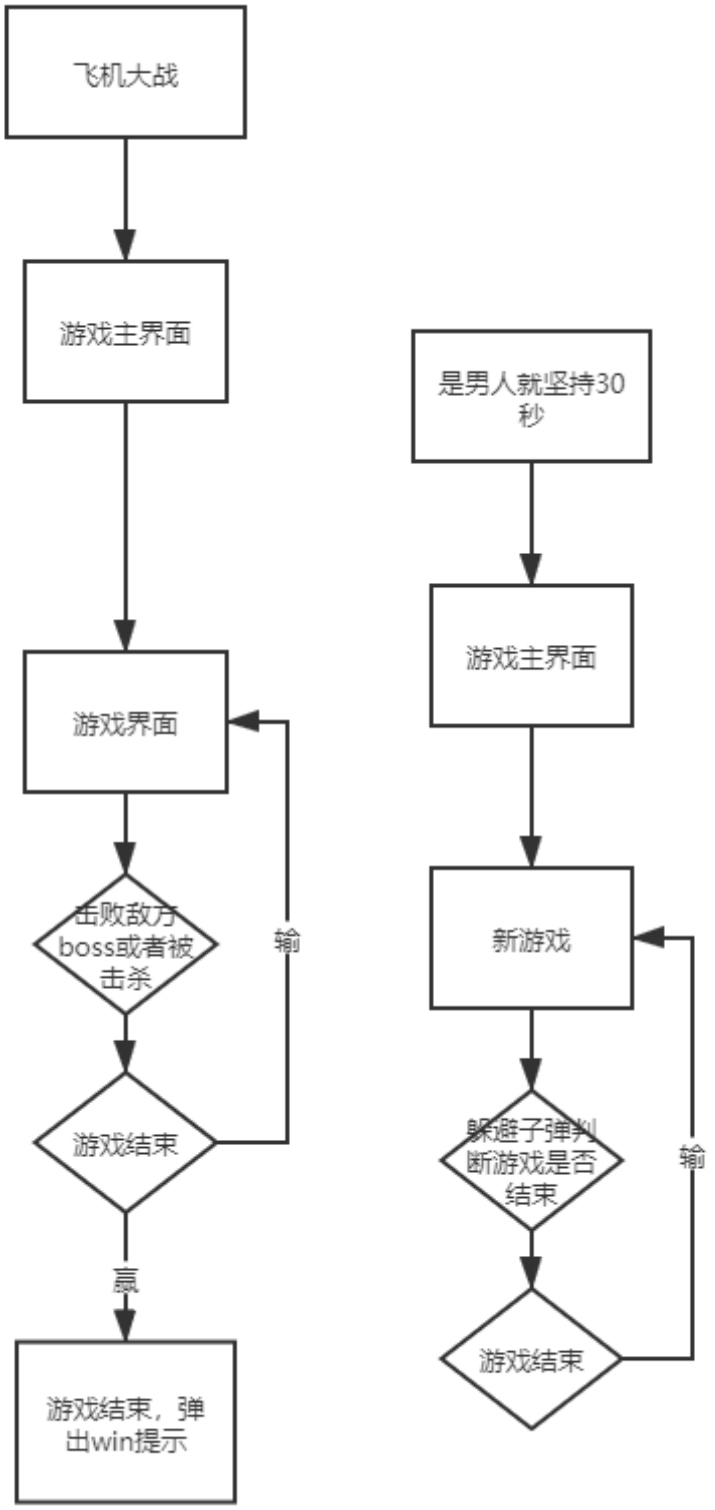
是男人就坚持 30 秒：王晨冰

贪吃蛇：王家辉

五子棋：王家辉

数据库：周增远

4. 流程图：



游戏大厅界面，供用户选择游戏进行游玩。

三、设计实现步骤

1、 基础知识准备

Java 的类和对象和基础知识，接口的定义和实现，GUI 界面的设计和常用的容器和组件，JDBC 数据库的链接和操作数据库，事件的处理和监听，数据库的增删改查

2、项目进度及任务分工

编号	时间	内容	负责人
1	6-28~7-2	登录界面及注册界面	周增远
2	6-28~7-2	管理员界面	周增远、王家辉
3	6-28~7-2	主界面	王晨冰
4	6-28~7-2	2048	周增远
5	6-28~7-2	飞机大战	王晨冰
6	6-28~7-2	是男人就坚持 30 秒	王晨冰
7	6-28~7-2	贪吃蛇	王家辉

8	6-28~7-2	五子棋	王家辉
9	6-28~7-2	数据库	周增远

2、 项目编程开发

主界面代码：


```
public class GameHill extends JFrame {
    /**
     * 面板
     */
    private JFrame jFrame;
    /**
     * 上容器
     */
    private JPanel topPanel;
    /**
     * 下容器
     */
    private JPanel bottomPanel;
    /**
```

```
             * 最小化按钮
             */
            private JButton min;
            /**
             * 关闭按钮
             */
            private JButton exit;

            private JLabel logo;
            private JLabel game2048;
            private JLabel game20481;
            private JLabel plane;
            private JLabel plane1;
            private JLabel man;
            private JLabel man1;
            private JLabel snake;
            private JLabel snake1;
            private JLabel gobang;
            private JLabel gobang1;
            private JLabel username;
            private JLabel jkxt;       //健康系统
            private JLabel jkxt2;       //健康系统


            public void init(User user) {
                  // 设置主题
                  this.setTitle("游戏大厅");
                  // 设置窗口图标

this.setIconImage(Toolkit.getDefaultToolkit().getImage("src/imgs/       游
戏.jpg"));
                  // 退出时关闭资源
      //
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                  // 设置窗口尺寸
                  this.setSize(1024, 768);
                  // 设置窗口屏幕居中
                  this.setLocationRelativeTo(null);
                  // 设置窗口不能改变大小
                  this.setResizable(false);
                  this.setLayout(null);


                  // 上容器
```

```
                topPanel = new JPanel();
                topPanel.setBounds(0, 0, 1050, 200);
                topPanel.setLayout(null);
                topPanel.setBackground(new Color(0, 0, 0));
                // 下容器
                bottomPanel = new JPanel();
                bottomPanel.setBounds(0, 200, 1050, 600);
                bottomPanel.setLayout(null);
                bottomPanel.setBackground(new Color(225, 225, 255));
                // 最小化按钮
                min = new JButton(new
ImageIcon("src/images/m3.png"));
                min.setBounds(900, 10, 50, 50);
                min.setBorder(null);
                min.setRolloverIcon(new
ImageIcon("src/images/m2.png"));
                // 关闭按钮
                exit = new JButton(new ImageIcon("src/images/3.png"));
                exit.setBounds(960, 10, 50, 50);
                exit.setBorder(null);
                exit.setRolloverIcon(new
ImageIcon("src/images/2.png"));
                // gameLogo
                logo = new JLabel(new
ImageIcon("src/images/logo1.jpg"));
                logo.setBounds(340, 23, 297, 161);
                //username
                username = new JLabel("尊敬的用户:" +
user.getUsername());
                username.setForeground(Color.BLUE);
                username.setFont(new Font("微软雅黑", Font.BOLD,
20));
                username.setBounds(50, 90, 500, 40);
      //          System.out.println(user.getUsername());
                //2048
                game2048 = new JLabel(new
ImageIcon("src/images/2048.jpg"));
                game2048.setBounds(140, 113, 100, 100);
                //20481
                game20481 = new JLabel(new
ImageIcon("src/images/20481.png"));
                game20481.setBounds(140, 113, 100, 100);
                game20481.setVisible(false);
                //plane
```

```
                plane = new JLabel(new
ImageIcon("src/images/plane.jpg"));
                plane.setBounds(440, 113, 100, 100);
                //plane1
                plane1 = new JLabel(new
ImageIcon("src/images/plane1.png"));
                plane1.setBounds(440, 113, 100, 100);
                plane1.setVisible(false);
                //man
                man = new JLabel(new
ImageIcon("src/images/man.png"));
                man.setBounds(740, 113, 100, 100);
                //man1
                man1 = new JLabel(new
ImageIcon("src/images/man1.png"));
                man1.setBounds(740, 113, 100, 100);
                man1.setVisible(false);
                //snake
                snake = new JLabel(new
ImageIcon("src/images/snake.jpg"));
                snake.setBounds(140, 363, 100, 100);
                //snake1
                snake1 = new JLabel(new
ImageIcon("src/images/snake1.png"));
                snake1.setBounds(140, 363, 100, 100);
                snake1.setVisible(false);
                //gobang
                gobang = new JLabel(new
ImageIcon("src/images/gobang.png"));
                gobang.setBounds(440, 363, 100, 100);
                //gobang1
                gobang1 = new JLabel(new
ImageIcon("src/images/gobang1.png"));
                gobang1.setBounds(440, 363, 100, 100);
                gobang1.setVisible(false);

                //健康系统
                jkxt = new JLabel("抵制不良游戏，拒绝盗版游戏。注意
自我保护，谨防受骗上当。");
                jkxt.setForeground(Color.WHITE);
                jkxt.setFont(new Font("微软雅黑", Font.BOLD, 11));
                jkxt.setBounds(710, 492, 800, 80);
                jkxt2 = new JLabel("适度游戏益脑，沉迷游戏伤身。合
理安排时间，享受健康生活。");
```

```
                jkxt2.setForeground(Color.WHITE);
                jkxt2.setFont(new Font("微软雅黑", Font.BOLD, 11));
                jkxt2.setBounds(710, 510, 800, 80);

                //1 图片的加载

                //添加面板
                topPanel.add(exit);
                topPanel.add(min);
                topPanel.add(logo);
                topPanel.add(username);
                this.add(topPanel);

                bottomPanel.add(game2048);
                bottomPanel.add(game20481);
                bottomPanel.add(plane);
                bottomPanel.add(plane1);
                bottomPanel.add(man);
                bottomPanel.add(man1);
                bottomPanel.add(snake);
                bottomPanel.add(snake1);
                bottomPanel.add(gobang);
                bottomPanel.add(gobang1);
                bottomPanel.add(jkxt);
                bottomPanel.add(jkxt2);
                this.add(bottomPanel);


                this.setUndecorated(true); //  去掉窗口的装饰


this.getRootPane().setWindowDecorationStyle(JRootPane.NONE);// 采 用
指定的窗口装饰风格

                {
                        /**
                         * 最小化
                         */
                        min.addActionListener(new ActionListener() {

                                @Override
                                public void actionPerformed(ActionEvent e) {
                                        setExtendedState(JFrame.ICONIFIED);
                                }
```

```
                });
                /**
                 * 关闭
                 */
                exit.addActionListener(new ActionListener() {

                    @Override
                    public void actionPerformed(ActionEvent e) {
                        System.exit(0);
                    }
                });

            }

            // 设置是否可见
            this.setVisible(true);
            Lister();
        }


        /**
         * 监听器方法
         */

        public void Lister() {
            {
                /**
                 * 鼠标移动到 plane      label 上
                 */
                game2048.addMouseMotionListener(new
MouseAdapter() {
                    @Override
                    public void mouseMoved(MouseEvent e) {
                        game2048.setVisible(false);
                        game20481.setVisible(true);
                        System.out.println("点击了");
                    }
                });

                /**
                 * 鼠标移除 plane1     label
                 */
                game20481.addMouseMotionListener(new
MouseAdapter() {
```

```
                        @Override
                        public void mouseMoved(MouseEvent e) {
                            try {
                                Thread.sleep(200);
                            } catch (InterruptedException
interruptedException) {

interruptedException.printStackTrace();
                            }
                            game2048.setVisible(true);
                            game20481.setVisible(false);

                            System.out.println("离开");
                        }

                    });
                    //2048 载入的鼠标方法
                    game2048.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e) {

                            JFrame f = new JFrame();
                            f.setTitle("game2048");
                            f.add(new Game2048(),
BorderLayout.CENTER);
                            f.pack();
                            f.setLocationRelativeTo(null);    //
窗口将置于屏幕的中央
                            f.setVisible(true);
                            f.setResizable(false);          //此
窗体不可由用户调整大小
                        }
                    });
                    game20481.addMouseListener(new MouseAdapter()
{
                        @Override
                        public void mouseClicked(MouseEvent e) {


System.out.println(Thread.currentThread().getName());
                            JFrame f = new JFrame();
                            f.setTitle("game2048");
                            f.add(new Game2048(),
BorderLayout.CENTER);
```

```
                                    f.pack();
                                    f.setLocationRelativeTo(null);     //
窗口将置于屏幕的中央
                                    f.setVisible(true);
                                    f.setResizable(false);            //此
窗体不可由用户调整大小
                              }
                        });
                  }

                  {
                        /**
                         * 鼠标移动到 plane      label 上
                         */
                        plane.addMouseMotionListener(new
MouseAdapter() {
                              @Override
                              public void mouseMoved(MouseEvent e) {
                                    plane.setVisible(false);
                                    plane1.setVisible(true);
                                    System.out.println("点击了");
                              }
                        });

                        /**
                         * 鼠标移除 plane1     label
                         */
                        plane1.addMouseMotionListener(new
MouseAdapter() {
                              @Override
                              public void mouseMoved(MouseEvent e) {
                                    try {
                                          Thread.sleep(200);
                                    }       catch       (InterruptedException
interruptedException) {

interruptedException.printStackTrace();
                                    }
                                    plane.setVisible(true);
                                    plane1.setVisible(false);

                                    System.out.println("离开");
                              }
```

```
                });
                plane1.addMouseListener(new MouseAdapter() {
                    @Override
                    public void mouseClicked(MouseEvent e) {
                        SwingWorker swingWorker = new
SwingWorker() {
                            @Override
                            protected Object doInBackground()
throws Exception {
                                new GameWin().launch();
                                return null;
                            }
                        };
                        swingWorker.execute();
                    }
                });

                plane.addMouseListener(new MouseAdapter() {
                    @Override
                    public void mouseClicked(MouseEvent e) {
                        SwingWorker swingWorker = new
SwingWorker() {
                            @Override
                            protected Object doInBackground()
throws Exception {
                                new GameWin().launch();
                                return null;
                            }
                        };
                        swingWorker.execute();
                    }
                });
            }
            {
                /**
                 * 鼠标移动到 man      label 上
                 */
                man.addMouseMotionListener(new MouseAdapter()
{
                    @Override
                    public void mouseMoved(MouseEvent e) {
                        man.setVisible(false);
                        man1.setVisible(true);
                        System.out.println("点击了");
```

```
                    }
                });

                /**
                 * 鼠标移除 man1    label
                 */
                man1.addMouseMotionListener(new
MouseAdapter() {
                    @Override
                    public void mouseMoved(MouseEvent e) {
                        try {
                            Thread.sleep(200);
                        }       catch       (InterruptedException
interruptedException) {

interruptedException.printStackTrace();
                        }
                        man.setVisible(true);
                        man1.setVisible(false);

                        System.out.println("离开");
                    }

                });
                man1.addMouseListener(new MouseAdapter() {
                    @Override
                    public void mouseClicked(MouseEvent e) {

                        MyGameFrame     frame     =     new
MyGameFrame();
                        //调用游戏类的初始方法
                        frame.launchFrame();

                    }
                });

                man.addMouseListener(new MouseAdapter() {
                    @Override
                    public void mouseClicked(MouseEvent e) {

                        MyGameFrame     frame     =     new
MyGameFrame();
                        //调用游戏类的初始方法
                        frame.launchFrame();
```

```
                        }
                    });
                }
                {
                    /**
                     * 鼠标移动到 snake      label 上
                     */
                    snake.addMouseMotionListener(new
MouseAdapter() {
                        @Override
                        public void mouseMoved(MouseEvent e) {
                            snake.setVisible(false);
                            snake1.setVisible(true);
                            System.out.println("点击了");
                        }
                    });

                    /**
                     * 鼠标移除 snake1     label
                     */
                    snake1.addMouseMotionListener(new
MouseAdapter() {
                        @Override
                        public void mouseMoved(MouseEvent e) {
                            try {
                                Thread.sleep(200);
                            }       catch       (InterruptedException
interruptedException) {

interruptedException.printStackTrace();
                            }
                            snake.setVisible(true);
                            snake1.setVisible(false);

                            System.out.println("离开");
                        }

                    });
                    snake.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e) {
                            JFrame frame = new JFrame("贪吃蛇");

                            // 添加游戏面板
```

```
                        frame.add(new GamePanel());

                        frame.setVisible(true);
                        frame.setResizable(false); // 禁止调整
大小
                        frame.setBounds(10, 10, 900, 720);
                    }
                });

                snake1.addMouseListener(new MouseAdapter() {
                    @Override
                    public void mouseClicked(MouseEvent e) {
                        JFrame frame = new JFrame("贪吃蛇");

                        // 添加游戏面板
                        frame.add(new GamePanel());

                        frame.setVisible(true);
                        frame.setResizable(false); // 禁止调整
大小
                        frame.setBounds(10, 10, 900, 720);
                    }
                });
            }
            {
                /**
                 * 鼠标移动到 gobang      label 上
                 */
                gobang.addMouseMotionListener(new
MouseAdapter() {
                    @Override
                    public void mouseMoved(MouseEvent e) {
                        gobang.setVisible(false);
                        gobang1.setVisible(true);
                        System.out.println("点击了");
                    }
                });

                /**
                 * 鼠标移除 gobang1     label
                 */
                gobang1.addMouseMotionListener(new
MouseAdapter() {
                    @Override
```

```
                        public void mouseMoved(MouseEvent e) {
                            try {
                                Thread.sleep(200);
                            } catch (InterruptedException
interruptedException) {

interruptedException.printStackTrace();
                            }
                            gobang.setVisible(true);
                            gobang1.setVisible(false);

                            System.out.println("离开");
                        }

                    });
                    gobang.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e) {
                            StartChessJFrame f = new
StartChessJFrame();//创建主框架
                            f.setVisible(true);//显示主框架
                        }
                    });

                    gobang1.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e) {
                            StartChessJFrame f = new
StartChessJFrame();//创建主框架
                            f.setVisible(true);//显示主框架
                        }
                    });
                }
            }



            二、飞机大战模块
```


```
public class GameWin extends JFrame {

    /** 定义双缓存图片 */
    Image offScreenImage = null;
    //游戏重绘次数
    int count = 1;
    //记录敌方飞机数量
    int ennemyCount = 0;
    //游戏状态 0 未开始 1 运行中 2 暂停 3 失败 4 成功
    int state = 0;
    int MY_WIDTH = 600;
    int MY_HEIGHT = 600;
    Image bg =
Toolkit.getDefaultToolkit().getImage("src/images/space.jpg");
    //爆炸效果图
    Image explode =
Toolkit.getDefaultToolkit().getImage("src/images/explode/e6.gif");
    //爆炸效果图坐标
    int explode_x = -300;
    int explode_y = 300;
    //定义大集合
```

```
        List<GameObject> objectList = new ArrayList<>();
        //被删除物体的集合
        List<GameObject> removeList = new ArrayList<>();
        //我方子弹集合
        List<ShellObj> shellObjList = new ArrayList<>();
        //敌方子弹集合
        List<BulletObj> bulletObjList = new ArrayList<>();
        //敌方战斗机集合
        List<EnemyObj> enemyObjList = new ArrayList<>();
        //爆炸对象集合
        List<ExplodeObj> explodeObjList = new ArrayList<>();
        //背景的实体类
        BgObj bgObj = new BgObj("src/images/space.jpg",0,-2000,2,this);
        //我方战斗机
        PlaneObj planeObj = new
PlaneObj("src/images/plane.png",290,550,0,this);
        //敌方 boss
        BossObj bossObj = null;
        //窗口的启动方法
        public void launch(){
            //设置图标

this.setIconImage(Toolkit.getDefaultToolkit().getImage("src/images/plane.
jpg"));
            //窗口是否可见
            this.setVisible(true);
            //窗口大小
            this.setSize(MY_WIDTH,MY_HEIGHT);
            //窗口的位置
            this.setLocationRelativeTo(null);
            //窗口的标题
            this.setTitle("飞机大战");

            //将游戏物体添加到大集合
            objectList.add(bgObj);
            objectList.add(planeObj);

            //为窗口添加开始鼠标事件
            this.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                    //鼠标左键代号是 1
                    if (e.getButton() == 1){
                        state = 1;
```

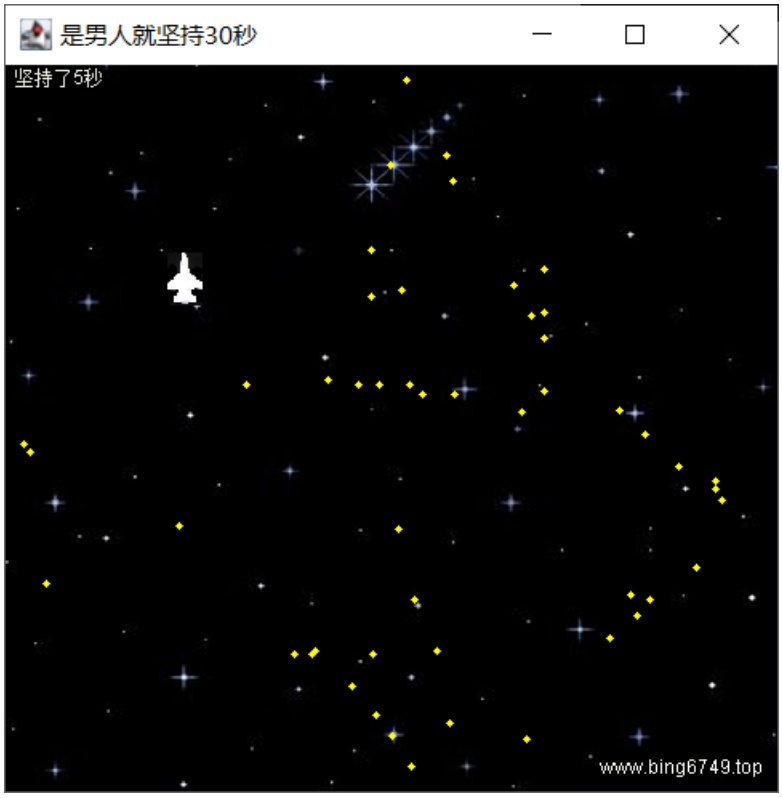
```
                    repaint();
                }
            }
        });
        //为暂停添加一个键盘事件
        this.addKeyListener(new KeyAdapter() {
            @Override
            public void keyPressed(KeyEvent e) {
                super.keyPressed(e);
                //空格键被按下
                if (e.getKeyCode() == 32){
                    switch (state){
                        //运行改为暂停
                        case 1:
                            state = 2;
                            break;
                        //暂停改为运行
                        case 2:
                            state = 1;
                            break;
                    }
                }
            }
        });

        while (true){
            if (state == 1){
                createObj();
                repaint();
            }

            try {
                //线程休眠   1 秒  = 1000 毫秒
                Thread.sleep(10);
            }catch (Exception e){
                e.printStackTrace();
            }
        }
    }

    @Override
    public void paint(Graphics g) {
        /** 创建和容器一样大小的 Image 图片 */
        if(offScreenImage ==null){
```

```
                offScreenImage=this.createImage(MY_WIDTH,
MY_HEIGHT);
            }
            /** 获得该图片的画布*/
            Graphics gImage= offScreenImage.getGraphics();
            /** 填充整个画布*/
            gImage.fillRect(0, 0, MY_WIDTH, MY_HEIGHT);
            //游戏未开始
            if (state == 0){
                  gImage.drawImage(bg,0,0,null);
                  //改变画笔的颜色
                  gImage.setColor(Color.BLUE);
                  //改变文字大小和样式
                  gImage.setFont(new Font("仿宋",Font.BOLD,50));
                  //添加文字
                  gImage.drawString("点击开始游戏",150,300);
            }
            if (state == 1){
                  //爆炸对象添加到大集合
                  objectList.addAll(explodeObjList);

                  //绘制所有游戏物体
                  for (GameObject object : objectList){
                        object.paintSelf(gImage);
                  }
                  //绘制完后将要删除的元素进行处理
                  objectList.removeAll(removeList);
            }
            //失败显示内容
            if (state == 3){
                  //改变画笔的颜色
                  gImage.setColor(Color.red);
                  //改变文字大小和样式
                  gImage.setFont(new Font("仿宋",Font.BOLD,50));
                  //添加文字
                  gImage.drawString("GAME OVER",180,300);
            }
            //通关显示内容
            if (state == 4){
                  //改变画笔的颜色
                  gImage.setColor(Color.green);
                  //改变文字大小和样式
                  gImage.setFont(new Font("仿宋",Font.BOLD,50));
                  //添加文字
```

```
                gImage.drawString("游戏通关",150,300);
            }
            gImage.drawImage(explode,explode_x,explode_y,null);
            /** 将缓冲区绘制好哦的图形整个绘制到容器的画布中 */
            g.drawImage(offScreenImage, 0, 0, null);
            //count 自增
            count++;
            System.out.println("大集合的长度" + objectList.size());
        }

        //添加子弹或敌机
        public void createObj(){
            //控制我方子弹生成速度
            if (count % 20 == 0){
                shellObjList.add(new
ShellObj("src/images/bulletGreen.png",planeObj.x + 3,planeObj.y -
10,10,this));
                objectList.add(shellObjList.get(shellObjList.size() - 1));
            }
            //敌方子弹生成速度
            if (count % 10 == 0 && bossObj != null){
                bulletObjList.add(new
BulletObj("src/images/bulletYellow.png",bossObj.x + 45,bossObj.y +
76,10,this));
                objectList.add(bulletObjList.get(bulletObjList.size() -
1));
            }
            //生成敌方战斗机
            if (count % 15 == 0) {
                enemyObjList.add(new
EnemyObj("src/images/enemy.png",this));
                objectList.add(enemyObjList.get(enemyObjList.size() -
1));
                ennemyCount++;
            }
            if (ennemyCount > 100){
                if (bossObj == null){
                    bossObj = new
BossObj("src/images/boss.png",250,30,5,this);
                    objectList.add(bossObj);
                }
            }
        }
```

```
    public static void main(String[] args) {
        GameWin gameWin = new GameWin();
        gameWin.launch();
    }
}
```

三、是男人就坚持 30 秒模块




```
public class MyGameFrame extends JFrame {

    int planeX = 100;
    int planeY = 100;

    Image planeImg = GameUtil.getImage("images/plane.png");
    Image bg = GameUtil.getImage("images/bg.jpg");

    Plane plane = new Plane(planeImg,planeX,planeY,5);  //创建飞机对象
    Shell[] shells = new Shell[50];  //创建炮弹们的对象
    Explode explode = null;  //创建爆炸对象
    Date start= new Date();  //创建游戏开始时间
    Date end;
```

```
//创建游戏结束时间（飞机死亡时刻）
    long                period                =                0;
//玩了多少秒

    @Override
    public void paint(Graphics g) {        //把 g 当作是一支画笔

        Color color = g.getColor();
        //画背景
        g.drawImage(bg,      0,      0,      Constant.GAME_WIDTH,
Constant.GAME_HEIGHT, null);

        //画时间
        drawTime(g);

        //画小飞机
        plane.drawMyself(g);
        //画小炮弹
        for (Shell shell : shells) {
            shell.drawMyself(g);

            //实现碰撞检测，将所有的炮弹和飞机进行飞行检测，
看有没有碰到
            boolean peng = shell.getRec().intersects(plane.getRec());
            if(peng){

                plane.live = false;

                //处理爆炸效果
                if(explode == null) {
                    explode = new Explode(plane.x, plane.y);
                }
                explode.drawMySelf(g);
            }
        }

        //添加水印
        g.setColor(new Color(253, 255, 246));
        g.drawString("www.bing6749.top", 380, 480);
        g.setColor(color);
    }
```

```
	//画时间
	public void drawTime(Graphics g){
		g.setColor(new Color(234,231,222));
		if(plane.live){
			period =
(System.currentTimeMillis()-start.getTime())/1000;
			g.drawString("坚持了" + period + "秒", 15, 50);

		}else{
			if(end == null) {
				end = new Date();
				period = (end.getTime()-start.getTime())/1000;
			}
			g.setColor(Color.RED);
			Font font = g.getFont();
			g.setFont(new Font("微软雅黑",Font.BOLD,50));
			g.drawString("最终时间:"+period+"秒",100,250);
			g.setFont(font);


		}
	}
	//关闭类
	public void close(){
		this.dispose();

//this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
	}

	//初始化窗口
	public void launchFrame() {
		this.setTitle("是男人就坚持 30 秒");	//窗口标题
		setVisible(true);				//窗口是否可见
		setSize(Constant.GAME_WIDTH,  Constant.GAME_HEIGHT);
//窗口大小

		setLocation(400, 500);			//窗口位置

		//增加关闭窗口的动作
		this.addWindowListener(new WindowAdapter() {
			@Override
			public void windowClosing(WindowEvent e) {

setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
                }
            });
            //初始化创建 50 个炮弹对象
            for (int i = 0; i < 50; i++) {
                shells[i] = new Shell();
            }
            new PaintThread().start();          //启动重画方法的线程



            this.addKeyListener(new KeyMonitor());   //启动键盘监听
        }

        /**
         * 定义另一个重画窗口的线程类
         * 定义内部类调用外部方法方便
         */
        class PaintThread extends Thread {
            @Override
            public void run() {
                while (true) {
                    repaint();          //内部类可以直接使用外部类的
成员
                    //条件符合显示弹窗
                    if(end!=        null&&end.getTime()+2000        <=
System.currentTimeMillis()&&!plane.live){
                        new MyOptionPane();
                        plane.live= true;
                    }
                    try {
                        Thread.sleep(20);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        }

        //弹窗类
        class MyOptionPane {
            public MyOptionPane() {
                int                     userOption                =
JOptionPane.showConfirmDialog(null,"是否重新开始游戏？","您已死亡
",JOptionPane.OK_OPTION,JOptionPane.QUESTION_MESSAGE);     //确
```

```
认对话框
                //如果用户选择的是 OK
                if (userOption == JOptionPane.OK_OPTION) {
                    close();
                    new MyGameFrame().launchFrame();
                }else {
                    close();
                }
            }
        }

        //内部类，实现键盘的监听处理
        class KeyMonitor extends KeyAdapter {
            //调用飞机类中的键盘控制方法

            @Override
            public void keyPressed(KeyEvent e) {
                plane.addDirection(e);
            }

            @Override
            public void keyReleased(KeyEvent e) {
                plane.minusDirection(e);

            }
        }

        //双缓冲技术
        private Image offScreenImage = null;
        @Override
        public void update(Graphics g){
            if(offScreenImage == null){
                offScreenImage                                          =
this.createImage(Constant.GAME_WIDTH,Constant.GAME_HEIGHT);
            }

            Graphics goff = offScreenImage.getGraphics();
            paint(goff);
            g.drawImage(offScreenImage,0,0,null);
        }
}
```

四、课程设计（实训）总结

经过之前的学习,对程序设计有了一定的认识与理解。在校期间,一直都是学

习理论知识,没有机会去参与项目的开发。所以说实话,这次实训,软件项目开发对我来说是比较抽象的,一个完整的项目要怎么分工以及完成该项(转载于:软件工程实训总结)目所要的步骤也不是很明确。而经过这次实训,让我明白了一个完整项目的开发,必须由团队来分工合作,并在每个阶段中进行必要的总结与论

证个完整项目的开发它所要经历的阶段包括:远景范围规划和用例说明、项目结构和风险评估、业务功能说明书、详细设计说明书、代码实现、测试和安装包等等。个项目的开发所需要的财力、人力都是很多的,如果没有个好的远景规划,对以后的开发进度会有很大的影响,甚至会出现在预定时间内不能完成项目或者完成的项目跟原来预想的不一样。一份好的项目结构、业务功能和详细设计说明书对一个项目的开发有明确的指引作用,它可以使开发人员对这个项目所要实现的功能在总体上有比较明确的认识,还能减少在开发过程中出现不必要的麻烦。代码的实现是一个项目开发成功与否的关键,也就是说,前期作业都是为代码的实现所做的准备。

实训期间让我学到很多东西,不仅在理论上让我对 IT 领域有了全新的认识,在实践能力上也得到了很大的提高真正的学到了学以致用,更学到很多做人的道理,对我来说受益匪浅。我意识到自己知识的缺少,这激励我在以后的学习、工作、生活中要不断了解信息技术发展动态以及信息发展中出现的新的技术。

五、指导教师评语及成绩

评　语:

成绩评定:

指导教师签名:　　批阅日期:

备注: