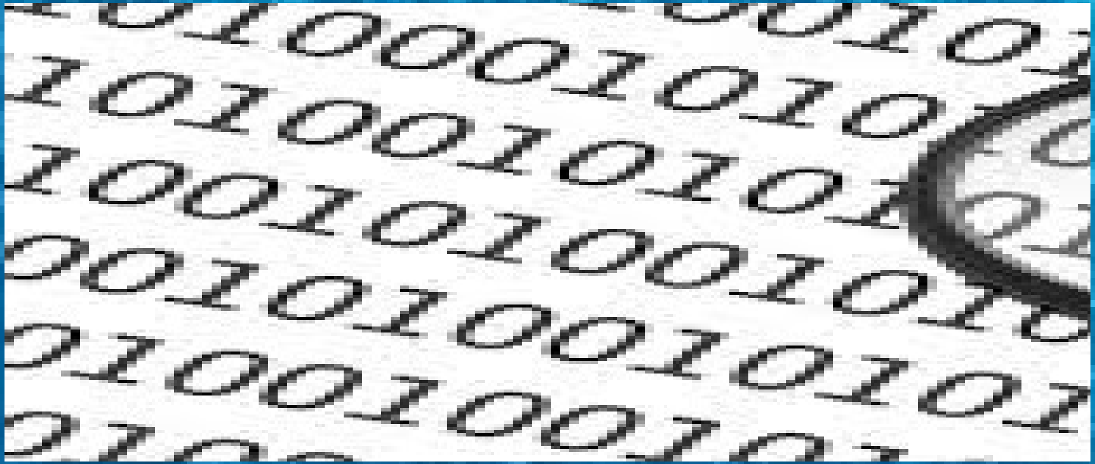
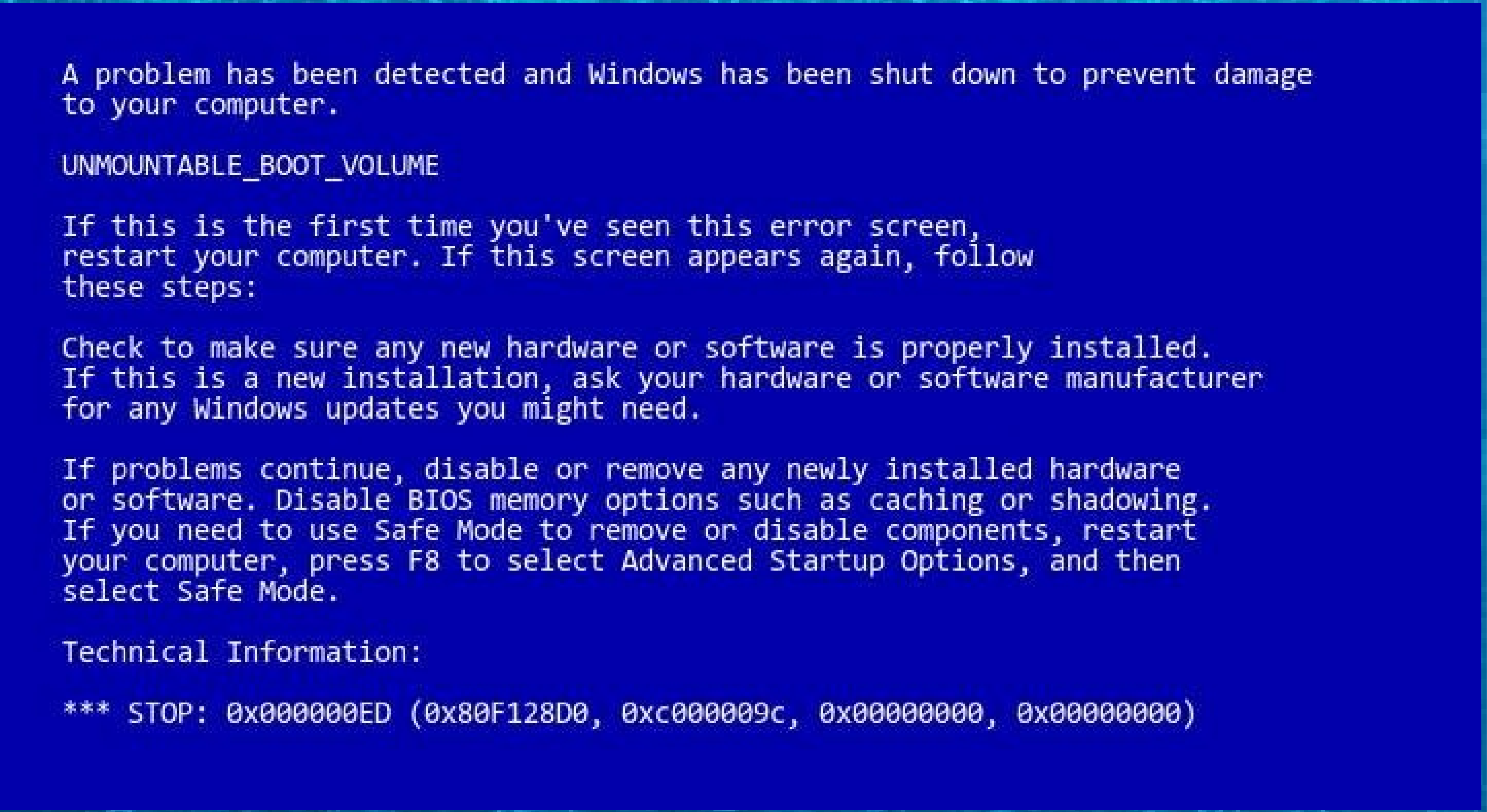
# Contexto

Um bug de software é um erro ou falha em um programa de computador ou sistema que faz com que ele produza um resultado incorreto ou inesperado, ou se comporte de maneira não intencional.

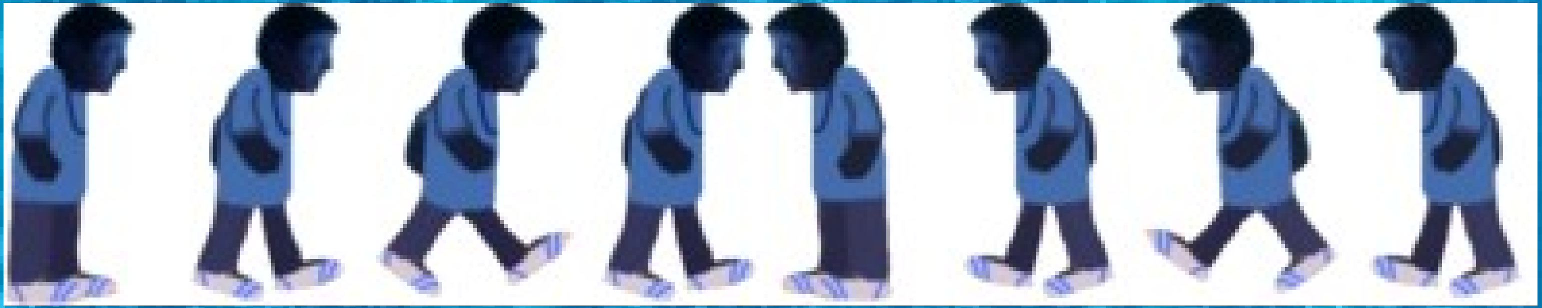


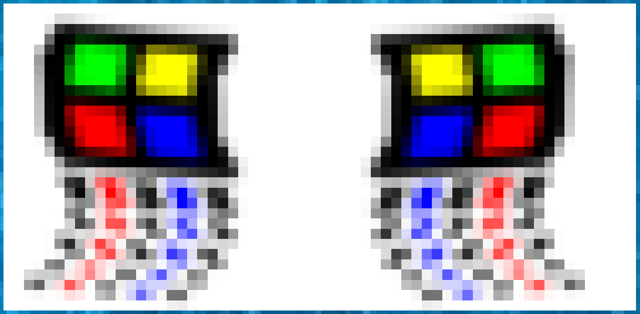
Tela Azul é um ecrã (tela) apresentado nos sistemas operativos Windows em caso de erro grave de sistema. Este ecrã tornou-se muito famoso pelas suas regulares aparições. A mais famosa foi durante a demonstração do Windows 98, em 1998, quando Bill Gates acidentalmente, exibiu esta tela ao público. A tela também é encontrada em consoles portáteis de videogame como PlayStation Portable, Nintendo DS e Game Boy Advance.



```
A problem has been detected and Windows has been shut down to prevent damage
to your computer.

UNMOUNTABLE_BOOT_VOLUME

If this is the first time you've seen this error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical Information:

*** STOP: 0x000000ED (0x80F128D0, 0xc000009c, 0x00000000, 0x00000000)
```

# Sprites





# Inimigos

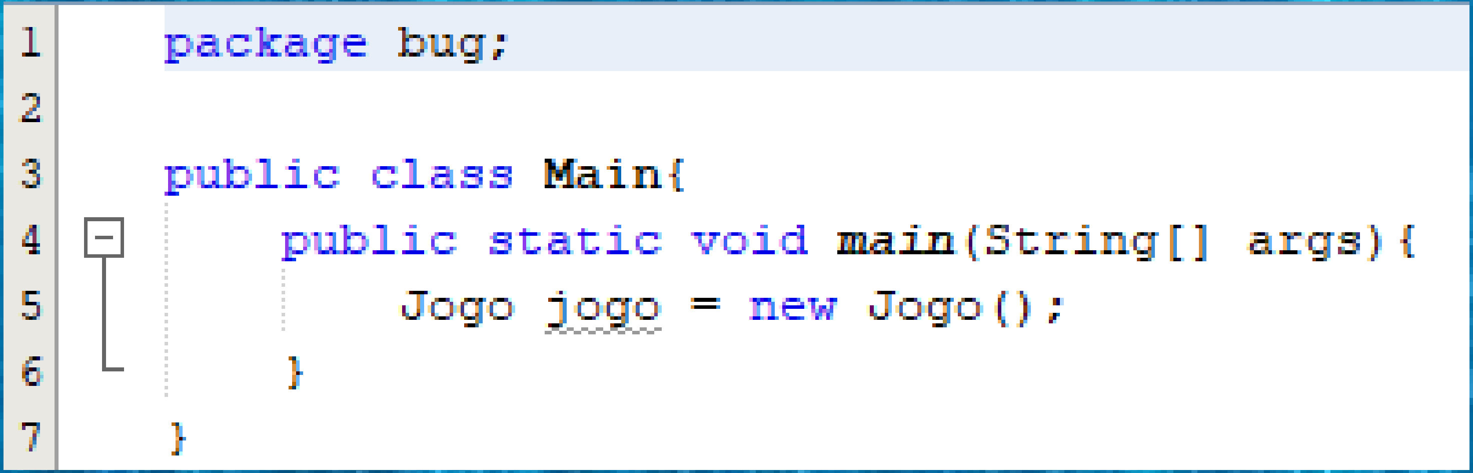
Após o personagem perder 2 pontos de vida, os inimigos começam a se teletransportar.

Caso perca 4, os inimigos se duplicam conforme uma porcentagem definida.

# CLASSES

MAIN
JOGO
SYSADM
BUG
SHOT

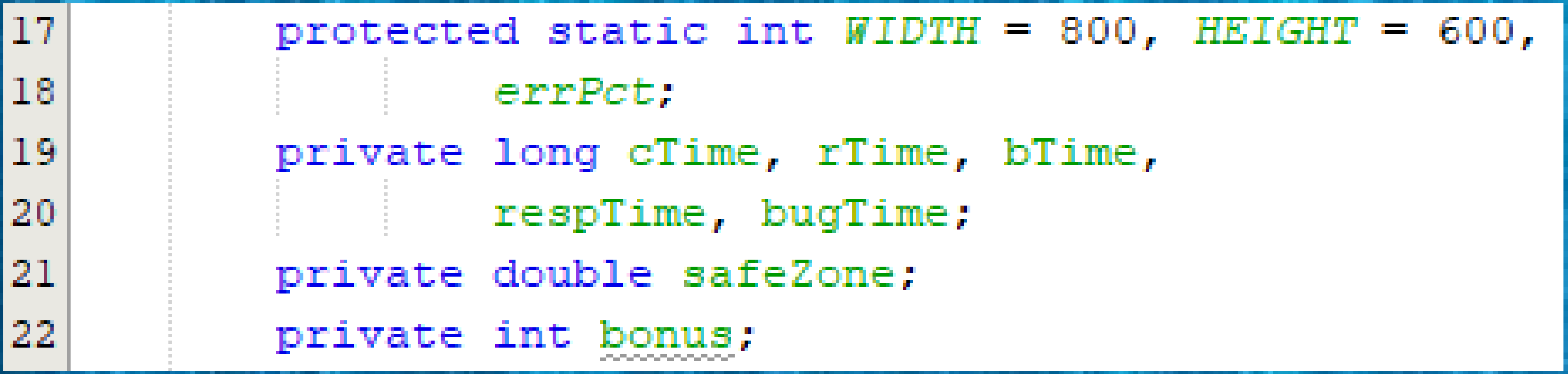
# MAIN



```java
package bug;

public class Main{
    public static void main(String[] args){
        Jogo jogo = new Jogo();
    }
}
```

# CLASSE JOGO

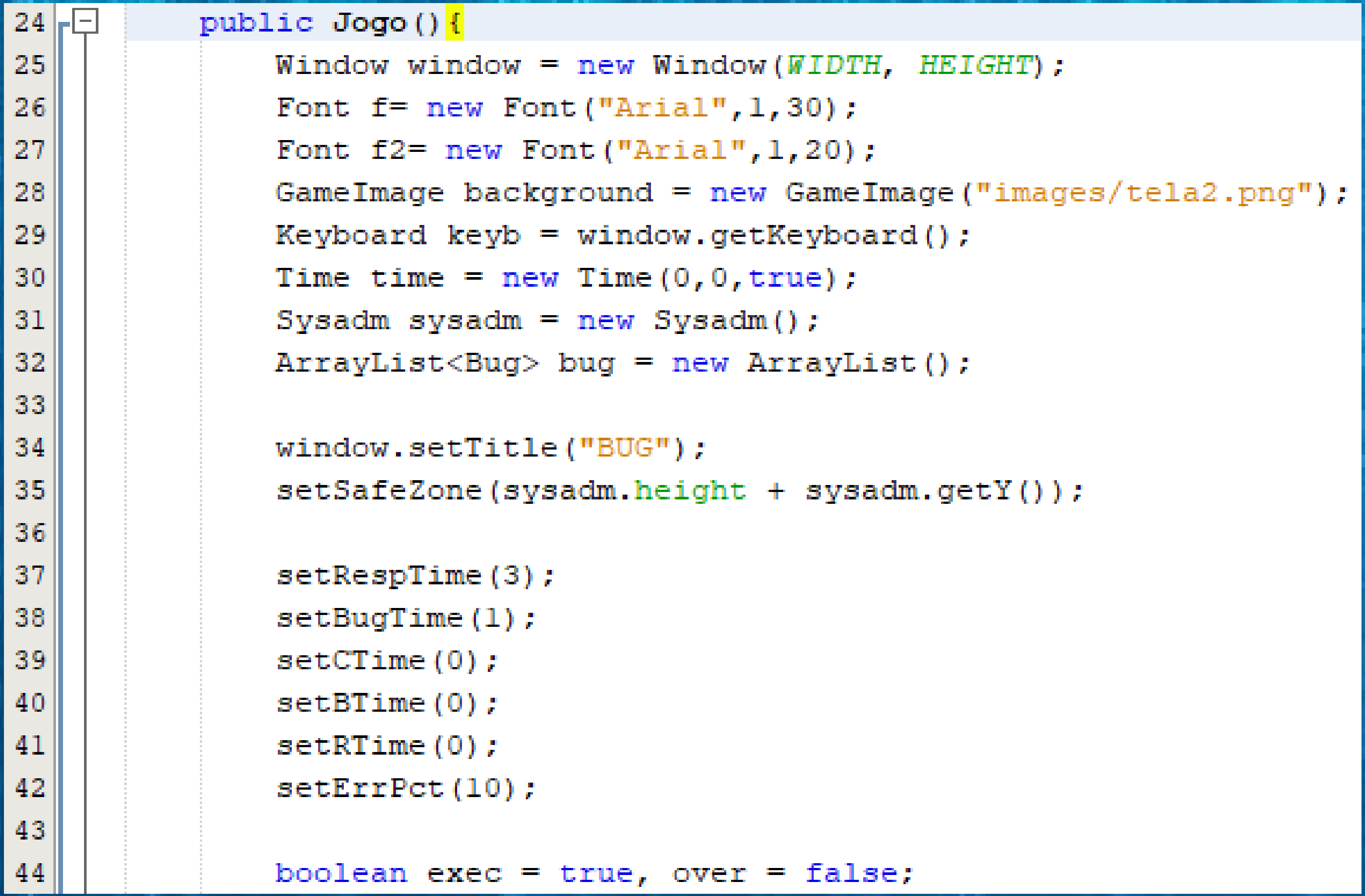
## ATRIBUTOS



```
    protected static int WIDTH = 800, HEIGHT = 600,
            errPct;
    private long cTime, rTime, bTime,
            respTime, bugTime;
    private double safeZone;
    private int bonus;
```

# CLASSE JOGO

## CONSTRUTOR



```java
    public Jogo(){
        Window window = new Window(WIDTH, HEIGHT);
        Font f= new Font("Arial",1,30);
        Font f2= new Font("Arial",1,20);
        GameImage background = new GameImage("images/tela2.png");
        Keyboard keyb = window.getKeyboard();
        Time time = new Time(0,0,true);
        Sysadm sysadm = new Sysadm();
        ArrayList<Bug> bug = new ArrayList();

        window.setTitle("BUG");
        setSafeZone(sysadm.height + sysadm.getY());

        setRespTime(3);
        setBugTime(1);
        setCTime(0);
        setBTime(0);
        setRTime(0);
        setErrPct(10);

        boolean exec = true, over = false;
```

# CLASSE JOGO

CONSTRUTOR(LOOP GAME)

```
while(exec){
        background.draw();
        window.drawText(String.valueOf(sysadm.getKills()), 710, 40, Color.red,f);
        int a=3;
        a=a- sysadm.getDeaths();
        window.drawText(String.valueOf(a), 610, 40, Color.green,f);
        sysadm.draw();

        if(keyb.keyDown(Keyboard.SPACE_KEY)) sysadm.shoot();
        else sysadm.moveX();

        if(cTime - lTime >= respTime){lTime = cTime; bug.add(new Bug());}
        for(int i = 0; i < bug.size(); i++){
           Bug b = bug.get(i);
           b.moveToUp();
           b.draw();
           b.runAnimation();
           if(getSafeZone() >= b.getY()){
              bug.remove(b);
              sysadm.die();
           }else if(sysadm.kill(b))
              if(sysadm.getDeaths() == 0) bug.remove(b);
              else if(Math.abs(gen.nextInt(100)) >= errPct) bug.remove(b);
        }
```

# CLASSE JOGO

CONSTRUTOR(LOOP GAME)

```
if((0 < sysadm.getDeaths()) && (Math.abs(gen.nextInt(100)) <= errPct) && (!
  bug.isEmpty())){
            int i = Math.abs(gen.nextInt(bug.size()));
            Bug b = bug.get(i);
            if(!b.tele){
               b.setX(Math.abs(gen.nextInt(WIDTH - b.width)));
               b.tele = true;
            }
         }
         if((1 < sysadm.getDeaths()) && (Math.abs(gen.nextInt(100)) <=
  errPct) && (!bug.isEmpty())){
            int i = Math.abs(gen.nextInt(bug.size()));
            Bug b = bug.get(i);
            if(!b.clone){
               Bug c = new Bug();
               if((b.getX() + (2 * b.width)) > WIDTH) c.setX(b.getX() - b.width);
               else c.setX(b.getX() + b.width);
               c.setY(b.getY());
               b.clone = true;
               c.clone = true;
               bug.add(c);
            }
         }
```

```
cTime = time.getTotalSecond();
           if(sysadm.getDeaths() > 2){
               background.loadImage("images/telaazul.jpg");
               new Sound("images/desligando.wav").play();
               background.draw();
               window.delay(1024);
               window.drawText(String.valueOf(sysadm.getKills()),
  65,450 , Color.white,f2);
               window.display();
               over = true;
               exec = false;
           }

           window.display();
           if(keyb.keyDown(Keyboard.ESCAPE_KEY)) exec = false;
       }
       bug.clear();
       while(!keyb.keyDown(Keyboard.ESCAPE_KEY) && over)
  {window.display();}
       window.exit();
   }
```

# CLASSE JOGO

OUTROS METODOS

```
    protected void setResp(long x){respTime = x;}
    protected long getResp(){return respTime;}
    protected void setCTime(long x){cTime = x;}
    protected long getCTime(){return cTime;}
    protected void setLTime(long x){lTime = x;}
    protected long getLTime(){return lTime;}
    protected void setErrPct(int x){errPct = x;}
    protected int getErrPct(){return errPct;}
    protected void setSafeZone(double x){safeZone = x;}
    protected double getSafeZone(){return safeZone;}

}//FIM DA CLASSE
```

# CLASSE SYSADM

# CLASSE SISADM

ATRIBUTOS E CONSTRUTOR

```
public final class Sysadm extends Sprite{
    protected static double X,Y,
            WIDTH, HEIGHT,
            VEL;
    private int deaths, kills;
    private final ArrayList<Shot> shot = new ArrayList();
    private final Time time = new Time(0,0,true);
    private long lTime, shotTime;
    public Sysadm(){
        super("images/sysadm.png",12);
        WIDTH = this.width;
        HEIGHT = this.height;
        setX((Jogo.WIDTH - this.width) / 2);
        kills = 0;
        setDeaths(0);
        setVel(1);
        setY(55);
        setLTime(0);
        setShotTime(0);
        this.setPosition(X, Y);
        this.setInitialFrame(0);
        this.setRangeOfFrames(0, 0);
        this.setRepeatAnimation(true);
    }
```

```
protected long getLTime(){return lTime;}
    protected void setLTime(long y){lTime = y;}
    protected long getShotTime(){return shotTime;
    protected void setShotTime(long y){shotTime =
y;}
    protected void setVel(double x){VEL = x;}
    protected void setX(double x){X = x;}
    protected double getX(){return X;}
    protected void setY(double y){Y = y;}
    protected double getY(){return Y;}
    private void setDeaths(int x){deaths = x;}
    protected int getDeaths(){return deaths;}
    protected int getKills(){return kills;}
    private long getHr(){
        Calendar cal = Calendar.getInstance();
        return ((cal.get(Calendar.HOUR) * 60 +
cal.get(Calendar.MINUTE) * 60 +
cal.get(Calendar.SECOND)) * 60 +
cal.get(Calendar.MILLISECOND));
    }
```

# CLASSE SYSADM

```
protected void shoot(){
        shot.add(new Shot());
        lTime = getHr();
        if(this.getCurrFrame() >= 6) this.setCurrFrame(10);
        else this.setCurrFrame(4);
    }

    protected boolean kill(Bug b){
        for(int i = 0; i < shot.size(); i++){
            Shot t = shot.get(i);
            if(t.collided(b)){
                shot.remove(t);
                kills += 1;
                return true;
            }
        }
        return false;
    }

    protected void die(){setDeaths(getDeaths() + 1);}


    protected boolean dead(){
        if(getDeaths() > 0) return false;
        else return true;
    }
```

```
    @Override
    public void draw(){
        super.draw();
        for(int i = 0; i < shot.size(); i++){
            Shot t = shot.get(i);
            t.fall();
            t.draw();
            if(t.isOnFloor()) shot.remove(t);
        }
    }

    @Override
    public void moveX(){
        super.moveX();
        X = this.x;
        Y = this.y;
        if(getHr() - lTime  > shotTime){
            if(this.getStateOfX() == Sprite.RIGHT){
                this.setCurrFrame((this.getCurrFrame() + 1)%4);
            }
            else if(this.getStateOfX() == Sprite.LEFT){
                this.setCurrFrame((this.getCurrFrame() + 1)%4 + 6);
            }
            else{
                if(this.getCurrFrame() == 4){this.setCurrFrame(5);lTime = getHr();}
                else if(this.getCurrFrame() == 10){this.setCurrFrame(11);lTime = getHr(
                else if(this.getCurrFrame() >= 6) this.setCurrFrame(6);
                else this.setCurrFrame(0);
            }
        }
    }
}
```

# CLASSE BUG

# CLASSE BUG

```
public class Bug extends Sprite{
    protected static double WIDTH, HEIGHT;
    private final Random gerador = new Random();
    public boolean tele, clone;

    public Bug(){
        super("images/bug.png",5);
        WIDTH = this.width;
        HEIGHT = this.height;
        setX(Math.abs(gerador.nextInt(Jogo.WIDTH - this.width)));
        setY(Jogo.HEIGHT - this.height);
        this.setPosition(getX(), getY());
        this.setVelocityY(0.05);
        this.setTimeChangeFrame(2);
        this.setRepeatAnimation(true);
        tele = false;
        clone = false;
    }

    protected void setX(double x){this.x = x;}
    protected double getX(){return this.x;}
    protected void setY(double y){this.y = y;}
    protected double getY(){return this.y;}

}
```

# CLASSE SHOT

# IMAGENS

# CLASSE SHOT

```
package bug;

import JPlay.Sprite;

public class Shot extends Sprite{
    public Shot(){
        super("images/teclado.png");
        this.setPosition(Sysadm.X, Sysadm.Y + Sysadm.HEIGHT);
        this.setFloor(Jogo.HEIGHT);
        this.setGravity(0.000098);
    }
}
```