

!

from Yongwoo's Park

(SnowCraft)'

가

가 ?

?

ZIP

?

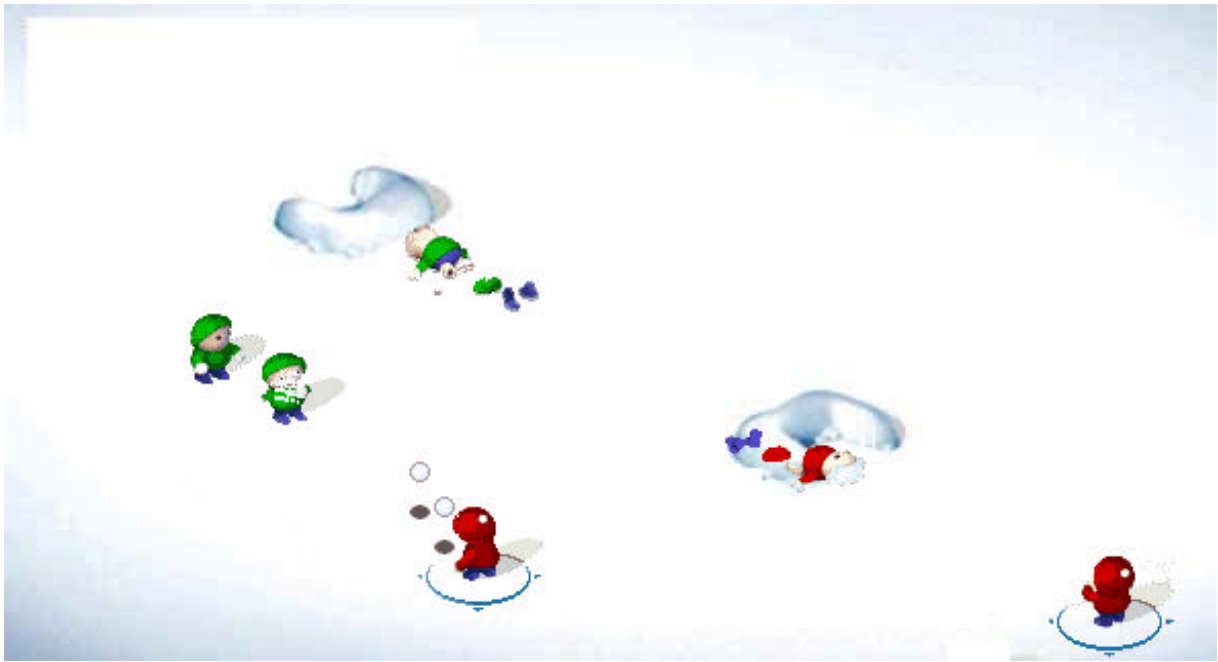
? ... !

?

?

1

가



1.

1

(Snow Ball)

1

☞

‘/’

☞

,5

가

☞

☞

가 가

가 가 ,

☞

가가

가

☞

가

☞

3

가

3,

5,7

가

가

가

!

ZIP ZIP

ZIP

가

☞ “images/logo.jpg”:

☞ “images/images.zip”:

“background.jpg”, “shadow.gif”, “fall.gif”, “ball.gif”, “sf1.gif”, “sf2.gif”, “sf3.gif”, “sf4.gif”, “asf1.gif”, “asf2.gif”, “asf3.gif”, “asf4.gif”, “asf5.gif”, “asf6.gif”, “asf7.gif”, “asf8.gif”

, ZIP
 ! , getImage Image 가
 , ZIP Image ? 가
 , ‘SnowCraft.java’ , getImagesFromZip

1) ZIP URL

```
URL url = new URL(getDocumentBase(), "images/images.zip");
```

2) URL , ZIP

ZipInputStream

```
ZipInputStream zipInputStream = new ZipInputStream(new DataInputStream(url.openStream()));
```

3) ZipInputStream ZipEntry ,

```
ZipInputStream
while((zipEntry = zipInputStream.getNextEntry()) != null) {
    String name = zipEntry.getName();
    int pos = 0;
    int bytesRead=0;
    int size = (int)zipEntry.getSize();
    byte[] buffer = new byte[size];
    for(;size>0;size-=bytesRead) {
        bytesRead = zipInputStream.read(buffer, pos, size);
        pos += bytesRead;
    }
    /* 4) */
}
```

4) , Toolkit

ZipEntry

(1) 3)

```
while((zipEntry = zipInputStream.getNextEntry()) != null) {
    String name = zipEntry.getName();
    /* 3) */
    if("background.jpg".equals(name)) {
        bg = tk.createImage(buffer);
    } else if("shadow.gif".equals(name)) {
        shadow = tk.createImage(buffer);
    } else if("fall.gif".equals(name)) {
        burst = tk.createImage(buffer);
    } else if("ball.gif".equals(name)) {
        ball = tk.createImage(buffer);
    } else {
        /* sf1.gif ~ sf4.gif, asf1.gif ~ asf8.gif */
    }
}
```

5) , MediaTracker

Image

```
getImagesFromZip();
tracker.addImage(bg, 0);
tracker.addImage(ball, 1);
tracker.addImage(shadow, 1);
tracker.addImage(burst, 1);
for(int i=0;i<images1.length;i++) {
    tracker.addImage(images1[i], 2);
}
for(int i=0;i<images2.length;i++) {
    tracker.addImage(images2[i], 2);
}
try {
    tracker.waitForAll();
} catch(Exception exception1) {}
```

가 , Toolkit createImage Image
ZIP , AudioClip

가

가

ImageCanvas : , Canvas

Jr. 8

TimeBar : , Canvas

Jr. 8

LoadingBarWindow : ImageCanvas

TimeBar

modal

Dialog

< 1 >

SnowCraft.java' , init, loadImages, getImagesFromZip, loadSounds, getAudioClip

< 1 >

```
public class SnowCraft extends Applet implements Runnable, MouseListener, MouseMotionListener {
    /* ... */
    LoadingBarWindow loadingBarWindow=null;
    int loadingBarCount = 0;

    public void init() {
        logoImage = getImage(getDocumentBase(), "images/logo.jpg");
        /* ... */

        // LoadingBarWindow
        loadingBarWindow = new LoadingBarWindow((Frame)this.getParent(),
            logoImage, "loading images...");

        loadingBarWindow.show(); //

        loadingBarWindow.setMessage("loading images..."); //
        loadingBarWindow.setStatusLevel(0); // 0
    }
```

```

loadImages();           //

loadingBarWindow.setMessage("loading sounds..."); //
loadingBarWindow.setStatusLevel(0); //      0
loadSounds();           //

loadingBarWindow.dispose(); //
/* ... */
}

private void getImagesFromZip() {
    try {
        /* ... */
        while((zipEntry = zipInputStream.getNextEntry()) != null) {
            /* ... */
            loadingBarCount++;
            int level = (int)(((double)loadingBarCount/16.0)*100.0) ; //
            loadingBarWindow.setStatusLevel(level); //
        }
    } catch(Exception anyException) {
        anyException.printStackTrace();
    }
}

private void loadImages() {
    /* ... */
}

public AudioClip getAudioClip(URL url, String name) {
    AudioClip clip = super.getAudioClip(url, name);
    clip.play(); //                play
    int level = (int)(((double)(++loadingBarCount) /11.0) * 100.0) ; //
    loadingBarWindow.setStatusLevel(level) ; //
    clip.stop(); //                가
    return(clip);
}

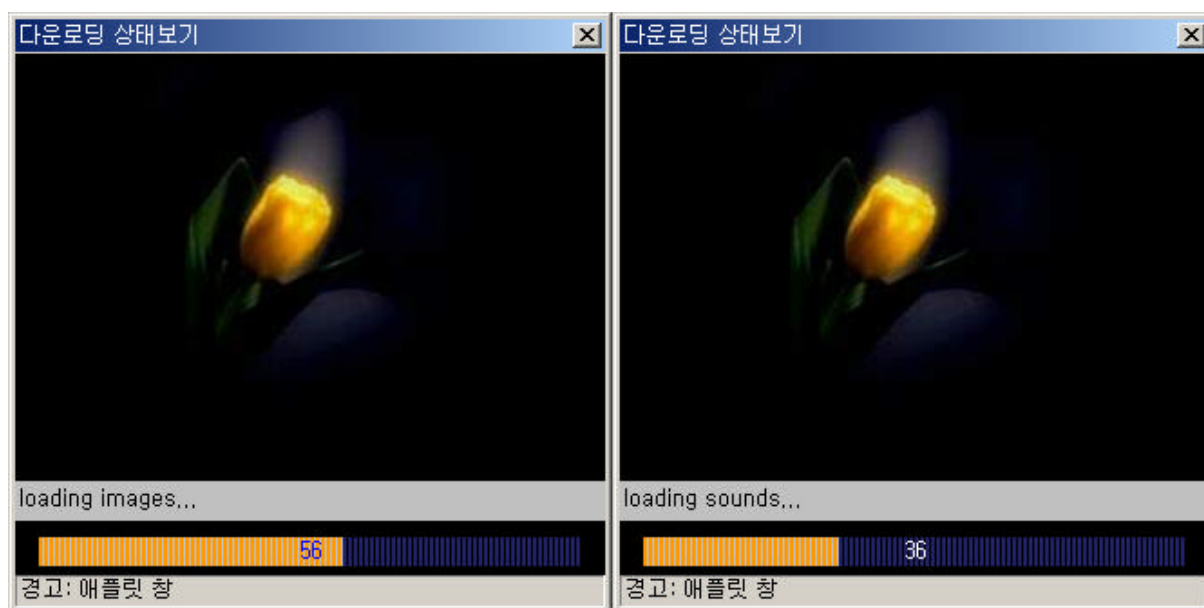
private void loadSounds() {
    loadingBarCount = 0; //

```

```

audioClip = getAudioClip(getCodeBase(), "audio/audioclip.au");
levelClip = getAudioClip(getCodeBase(), "audio/levelclip.au");
/*
loadingBarWindow.setStatusLevel(100);
}

```



2.

update

가

가

가

Component 가
 repaint
 가
 repaint 가

≦ Repaint() : update(Graphics g)

≦ update(Graphics g) : update
 paint(Graphics g)

≦ paint(Graphics g) :

, update
 , update

```
public void update(Graphics g) {  

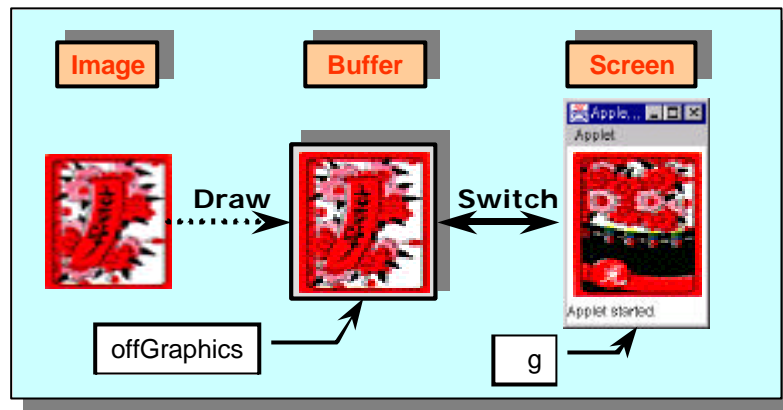
  paint(g);  

}
```

, 가

가 , 가 가 ,

(double-buffering) 가 가 , 3 ,
 가



3.

. ,
, << 2>

< 2>

```

public class SnowCraft extends Applet implements Runnable, MouseListener, MouseMotionListener {
    int[] type = new int[] { Cursor.CROSSHAIR_CURSOR };
    Cursor cursor=null;

    Image bufferImage=null;
    Graphics bufferGraphics=null;
    Thread manMoveThread, ballMoveThread, computerActionThread;
    public void init() {
        //          CROSSHAIR      (+)
        cursor = Cursor.getPredefinedCursor(type[0]);
        this.setCursor(cursor);
        /* ... */
        //
        bufferImage = createImage(this.size().width, this.size().height);
        bufferGraphics = bufferImage.getGraphics();
    }

    public void run() {
        Thread currentThread=null;

        while(true) {

```

```

if(status[who] == 1 && power < 50) {
    //
    power++;
}
currentThread = Thread.currentThread();
if(currentThread == displayThread) {
    try {
        Thread.sleep(DISPLAY_DELAY);
    } catch(InterruptedException e) {
        showStatus(e.toString());
    };
    //
    repaint();
} else if(currentThread == computerActionThread) {
    try {
        Thread.sleep(DISPLAY_COMPUTING);
    } catch(InterruptedException e) {
        showStatus(e.toString());
    };
    //
    computerAction();
}
}

//
public void tranject(int dirx, int diry, int dirz, int ith) {
    /*
    ,
    .*/
}
public void atranject(int adirx, int adiry, int adirz, int aith) {
    /*
    ,
    .*/
}
public void comMove(int i) {
    /*
    ,
    .*/
}
}

```

```

public void comThrow(int i) {
    /*                                     .*/
}
public void computerAction() {
    for(int i=0;i<AMS;i++) {
        if(astatus[i] != 6 && astatus[i] != 7) {
            if(aicom[i] == 0) {
                aicom[i] =(int)(Math.random()*5);
            }
            switch(aicom[i]) {
                case 0 : break;
                case 1 : comMove(i); break;
                case 2 : comThrow(i); break;
                case 3 : comMove(i); break;
                case 4 : comThrow(i); break;
            }
        } else if(astatus[i] == 6) {
            if(agetUp[i]++ == 10) {
                agetUp[i] = 0;
                astatus[i] = 0;
            }
        }
    }
}
public void ballXYZ() {
    /*                                     ,
        가                                     .*/
}

//
public void showAll(Graphics g) {
    bufferGraphics.drawImage(bg, 0, 0, this);
    for(int j=0;j<MS;j++) {
        if(j != who) {
            bufferGraphics.drawImage(images1[status[j]], xpos[j], ypos[j], this);
        }
        tempX[j]=xpos[j]; tempY[j]=ypos[j];
    }
    bufferGraphics.drawImage(images1[status[who]], xpos[who], ypos[who], this);
    /*                                     */
}

```

```

}

public void initGame(Graphics g) {
    int i, cx=400, cy=30;
    bufferGraphics.drawImage(bg, 0, 0, this);
    first = false;
    for(int j=0;j<MS;j++) {
        xpos[j] = 300+(int)(Math.random()*230);
        ypos[j] = 160+(int)(Math.random()*110);
        bufferGraphics.drawImage(images1[status[j]], xpos[j], ypos[j], this);
    }
    try {
        Thread.sleep(1000);
    } catch(InterruptedException e) {
        showStatus(e.toString());
    };
    /*
    */
}

public void update(Graphics g) {
    paint(g);
}

public void paint(Graphics g) {
    if(first == true) {
        initGame(g); //
    }
    showAll(g); //
    ballXYZ();

    g.drawImage(bufferImage, 0, 0, this); //
}
}

```

☞ : AIFF, AU, WAV

☞ : MIDI TYPE 0, MIDI TYPE 1, RMF.

	8kHz	48khz	(sample rate)	8	16
(mono)		(stereo)		.	,
	,	8		.	,
					AudioClip
					가
				,<<	3>

< 3>

```

public class SnowCraft extends Applet implements Runnable, MouseListener, MouseMotionListener {
    AudioClip audioClip, levelClip, strikeClip, astrikeClip, endClip, notcomeClip, shootClip;
    AudioClip deadClip, adeadClip, dropClip, ashootClip;

    public void init() {
        loadSounds();
        /* ... */
    }

    public AudioClip getAudioClip(URL url, String name) {
        AudioClip clip = super.getAudioClip(url, name);

        clip.play(); // play
        int level = (int)(((double)(++loadingBarCount) / (double)11) * (double)100);
        loadingBarWindow.setStatusLevel(level);
        clip.stop(); // 가

        return(clip);
    }

    private void loadSounds() {
        audioClip = getAudioClip(getCodeBase(), "audio/audioclip.au");
        /* */
    }

```

```
public void start() {
    //          /          ,
    if(audioClip != null) {
        audioClip.loop();
    }
    /* ... */
}

public void stop() {
    //          ,          .
    if(audioClip != null) audioClip.stop();
    /*          .*/
}

public void tranject(int dirx, int diry, int dirz, int ith) {
    //          ,          .
    if(shootClip != null) {
        shootClip.play(); //
    }
    /* ... */
}

public void atranject(int adirx, int adiry, int adirz, int aith) {
    if(ashootClip != null) {
        ashootClip.play(); //
    }
    /* ... */
}

public void gameOver() {
    if(endClip != null) {
        endClip.play(); //
    }
    stop(); //
}
}
```

(stop) HTML , , 가
 , stop , 가
 , 가 , 가
 가 가 가

가 “ ” ,
 , 가 ,
 ,
 가 가 가
 가
 가

(MouseListener)

MouseListener
 MouseListener 가
 , MouseAdapter MouseListener
 MouseAdapter ,
 void mouseClicked(MouseEvent e): 가
 void mouseEntered(MouseEvent e): 가
 void mouseExited(MouseEvent e): 가

```

< void mousePressed(MouseEvent e):

```

```

< void mouseReleased(MouseEvent e):

```

```

, < 4>

```

```

< 4>

```

```

public class SnowCraft extends Applet implements Runnable, MouseListener, MouseMotionListener {
    public void init() {
        /* ... */
        addMouseListener(this);
        addMouseMotionListener(this);
    }

    public void mousePressed(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        for(int i=0;i<MS;i++) {
            if(status[i] != 3) {
                dx = x - xpos[i];
                dy = y - ypos[i];
                if(dx < width && dx >= 0 && dy < height && dy >= 0) {
                    status[i] = 1;
                    who = i;
                    break;
                }
            }
        }
    }

    public void run() {
        while(true) {
            if(status[who] == 1 && power < 50) {
                power++;
            }
            /* ... */
        }
    }
}

```



```

public void mouseReleased(MouseEvent e) {
    if(status[who] != 3 && dx<width && dx>=0 && dy<height && dy>=0) {
        vx.addElement(new Integer(xpos[who]+5));
        vy.addElement(new Integer(ypos[who]+10));
        vz.addElement(new Integer(power/2));
        status[who] = 2;
    }
    power = 0;
}

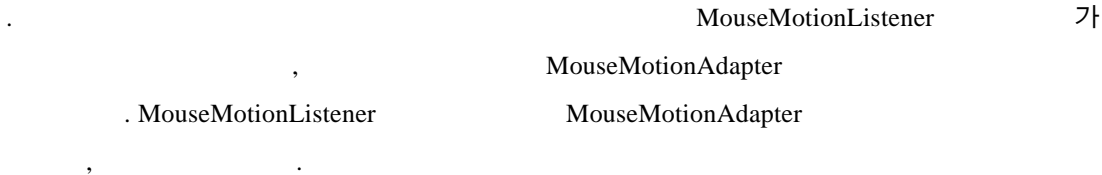
public void mouseClicked(MouseEvent e) { }
public void mouseEntered(MouseEvent e) { }
public void mouseExited(MouseEvent e) { }
}

```

가 (mousePressed), 가
 (MS=3) , 가 (x, y) (xpos
 [i], ypos[i]) 가 0 (width, height) . ,
 가 , status[i] 1 , (i) who .
 , (x,y) (xpos[i], ypos[i]) (dx, dy)
 , who status
 [who] , (power) 가 . ,
 status[who] 1 (poser) 가 . ,
 , (mouseReleased), vx, vy, vz (x+5,y+10) (power/2)
 . , status[who] 2 (power) 0 .
 vx, vy, vz ,
 가 .

(MouseListener)

MouseListener



```

< void mouseDragged(MouseEvent e):

```

```

< void mouseMoved(MouseEvent e):

```

가

, < 5

< 5>

```

public class SnowCraft extends Applet implements Runnable, MouseListener, MouseMotionListener {
    public void init() {
        /* ... */
        addMouseListener(this);
        addMouseMotionListener(this);
    }

    public void mouseDragged(MouseEvent e) {
        if(status[who] != 3) {
            int x = e.getX();
            int y = e.getY();
            if((x+y) <= 400) { // 가 가
                if(notcomeClip != null) {
                    notcomeClip.play();
                }
            }

            if(((x+y) > 400) && dx < width && dx >= 0 &&
                dy < height && dy >= 0 && x < 582 && y < 312) {
                xpos[who] = x - dx;
                ypos[who] = y - dy;
                status[who] = 1;
            }
        }
    }

    public void mouseMoved(MouseEvent e) { }
}

```

```

, (x,y) , x y 400
“가 !” ,
, 가 (mousePressed),
(dx,dy)

```

```
dx = x - xpos[i]
```

```
dy = y - ypos[i]
```

```
xpos[i] = x - dx
```

```
ypos[i] = y - dy
```

```
ZIP
```

```
login.asp
```

```
, ScoreBoard.asp
```

```
HTML
```