
벌레를 잡아라!

김가희, 신아영, 진민지 조

목차

1

구현 목표

2

구현 코드

3

구현 결과

구현 목표

Thread를 활용한 게임

JTabbedPane

JPanel

Event

Graphic

JLabel

JFrame

Image

BorderLayout

GridLayout

Textfield

구현 목표 - 레이아웃

1



전체 프레임 - Card Layout

`setLayout(card)`

//첫 번째 패널
Jpanel start

paintComponent로
이미지 그리기

이미지 버튼

구현 목표 - 레이아웃

2



//두 번째 패널
centerPanel
paintComponent 배경 넣기

menuPanel
메뉴 버튼 4가지

contentPanel "NORTH":
menuPanel
contentPanel "CENTER":
centerPanel

//재생 버튼
centerPanel-bugs[], deathbugs[]
//일시 정지 버튼
thread stop
//되돌아가기 버튼
thread resume

구현 목표 - 레이아웃

2



//타이머 레이블

60초 제한

1초 단위

일시정지 시 쓰레드와 함께 중단

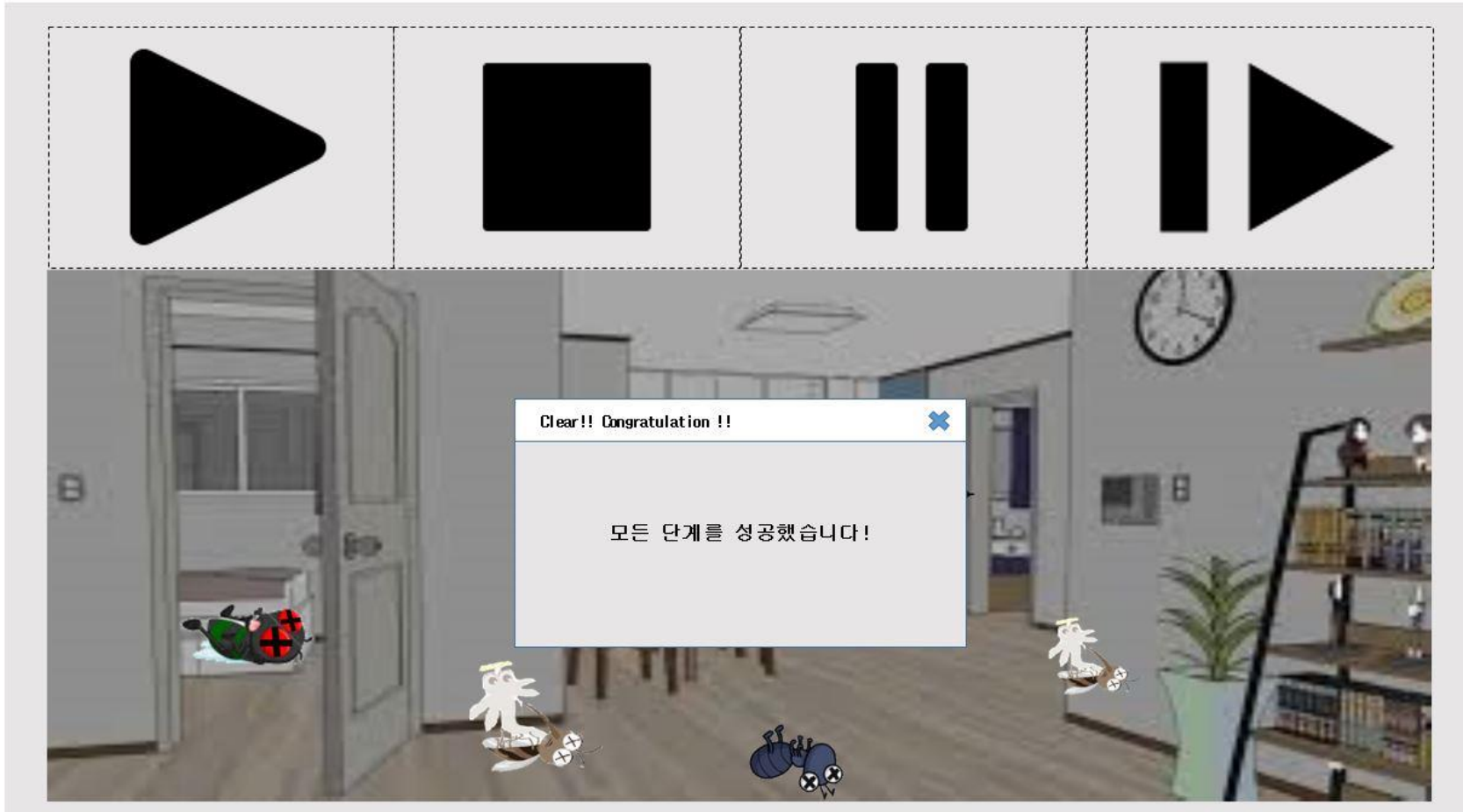
벌레들도 각자 쓰레드로 움직임
쓰레드에 ActionListener추가

time이 0이하일 때 (시간 초과 시)
팝업 창-실패 메시지
fail로 화면 전환

커서 이미지 변경

구현 목표 - 레이아웃

3



모든 레벨 달성 시
팝업 창-성공 메시지

음악 효과 추가

구현 목표 - 레이아웃

4

//성공 패널



구현 목표 - 레이아웃

4

//실패 패널



구현 코드 - layout_Main

```
*Bug2.java  layout.java  layout_Main.java  Death_bug.java
HW > src > bug_catch > layout_Main > main(String[]) : void
1 package bug_catch;
2 import javax.swing.*;
5
6 public class layout_Main {
7     public static void main(String args[]){
8         JFrame f = new JFrame("ButtonLabel0");
9         f.setLayout(new CardLayout());
10        f.add(new layout());
11        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12
13        f.setSize(1000,800);
14
15        f.setVisible(true);
16    }
17
18 }
19 |
```

구현 코드 Bug

```
*Bug2.java layout.java layout_Main.java Death_bug.java
HW > src > bug_catch > Bug2 > run() : void
1 package bug_catch;
2
3 import javax.swing.ImageIcon;
4
5
6
7
8
9
10 public class Bug2 extends JButton implements Runnable{
11     int x, y;
12     int speed;
13     public ImageIcon name;
14     JFrame app;
15
16
17 public Bug2(JFrame _app, ImageIcon _name, int w, int h)
18 {
19     app = _app;
20     this.name = _name;
21     x = w;
22     y = h;
23     this.setIcon(name);
24     this.setBackground(Color.pink);
25     this.setOpaque(false);
26     this.setBounds(x, y, 50, 50);
27 }
28
```

```
29 public void run() {
30     // TODO x, y 값을 받아서 바꿔줘야함.
31     // 몬스터가 달리는 부분
32     for (int i=0 ; i < 10000 ; i++) {
33         // 여기서 반복문 break 조건을 시간제한으로 주면 될 듯
34         speed = (int)(Math.random() * 50);
35         int move = speed % 4;
36         if (move == 0 ) {
37             x += speed;
38         } else if( move == 1) {
39             if(x > speed){x -= speed;} else {x += speed;}
40         } else if (move == 2) {
41             if(y < 1000){y += speed;} else {y -= speed;}
42         } else if (move == 3) {
43             if(y > speed){y -= speed;} else {y += speed;}
44         } else {
45             x += speed;
46             y += speed;
47         }
48         // 몬스터 움직이게 하는 부분
49         this.setBounds(x,y,50,50);
50         // 몬스터 마다 서로 다른 임의의 인터벌을 가진다.
51         try {
52             Thread.sleep((int)(Math.random()*400));
53         } catch (InterruptedException e) {
54             // TODO Auto-generated catch block
55             e.printStackTrace();
56         }
57     } // 잡지 못했을 때
58 }
59 }
```

구현 코드 - Death_bug

```
*Bug2.java  layout.java  layout_Main.java  Death_bug.java ✖
HW > src > bug_catch > Death_bug > run() : void
1 package bug_catch;
2
3 import javax.swing.ImageIcon;
4
5
6
7
8
9
10 public class Death_bug extends JButton implements Runnable{
11     int x,y;
12     int speed;
13     public ImageIcon name;
14     JFrame app;
15
16
17 public Death_bug(JFrame _app, ImageIcon _name, int w, int h)
18 {
19     app = _app;
20     this.name = _name;
21     x = w;
22     y = h;
23     this.setIcon(name);
24     this.setBackground(Color.pink);
25     this.setOpaque(false);
26     this.setBounds(x, y, 50, 50);
27 }
28
```

```
29
30 // TODO x, y 값을 받아서 바꿔줘야함.
31 // 몬스터가 달리는 부분
32 for (int i=0 ; i < 10000 ; i++) {
33     // 여기서 반복문 break 조건을 시간제한으로 주면 될 듯
34     speed = (int)(Math.random() * 50);
35     y += 10;
36     // 몬스터 움직이게 하는 부분
37     this.setBounds(x,y,50,50);
38     // 몬스터 마다 서로 다른 임의의 인터벌을 가진다.
39     try {
40         Thread.sleep((int)(Math.random()*300));
41     } catch (InterruptedException e) {
42         // TODO Auto-generated catch block
43         e.printStackTrace();
44     }
45 } // 잡지 못했을 때
46
47
48
49 }
50
51
```


구현 코드 - layout (전체 레이아웃 & KillBug & timer)

```
*Bug2.java  layout.java  layout_Main.java  Death_bug.java
HW > src > bug_catch > layout > KillBug > KillBug(int) >
1 package bug_catch;
2
3 import javax.imageio.ImageIO;
4 import javax.swing.*;
5 import java.awt.*;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8 import java.awt.image.BufferedImage;
9 import java.util.Timer;
10 import java.util.TimerTask;
11
12 public class layout extends JPanel{
13     static CardLayout card = new CardLayout();
14
15     //초기화면
16     //JPanel start = new JPanel();
17     //게임화면
18     KillBug main = new KillBug(1);
19     //게임 성공화면
20     //JPanel success = new JPanel();
21
22     //this Panel 생성
23     JPanel base = this;
24     ImageIcon icon;
25     ImageIcon icon2;
26     ImageIcon icon3;
27     ImageIcon success_icon;
28     ImageIcon fail_icon;
29     ImageIcon explain_icon;
30
31
32     //게임 실패화면
33     static JPanel fail;
34     static JPanel success;
35
36     //시작화면
37 public layout() {
38     this.setBackground(Color.white);
39
40     icon = new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\maintitle.jpg");
41     //icon2 = new ImageIcon("image/background2.png");
42
43
44     //배경 Panel 생성후 컨테이너로 지정
45 JPanel start = new JPanel(){
46     public void paintComponent(Graphics g) {
47         // Approach 1: Display image at full size
48         g.drawImage(icon.getImage(), 150, 80, 900,700,null);
49
50         setOpaque(false); //그림을 표시하게 설정, 투명하게 조절
51         super.paintComponent(g);
52     }
53
54 };
55
```

구현 코드 - layout (전체 레이아웃 & KillBug & timer)

```
56 //성공화면
57 success_icon = new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\success.png");
58
59 //배경 Panel 생성후 컨트롤패널으로 지정
60 JPanel success = new JPanel(){
61     public void paintComponent(Graphics g) {
62         // Approach 1: Display image at at full size
63         g.drawImage(success_icon.getImage(), 0, 0, 1000,800,null);
64
65         setOpaque(false); //그림을 표시하게 설정, 투명하게 조절
66         super.paintComponent(g);
67     }
68 };
69
70 //실패화면
71 fail_icon = new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\fail.png");
72
73 //배경 Panel 생성후 컨트롤패널으로 지정
74 JPanel fail = new JPanel(){
75     public void paintComponent(Graphics g) {
76         // Approach 1: Display image at at full size
77         g.drawImage(fail_icon.getImage(), 0, 0, 1000,800,null);
78
79         setOpaque(false); //그림을 표시하게 설정, 투명하게 조절
80         super.paintComponent(g);
81     }
82 };
83
84 //레이아웃은 CardLayout
85
86 //레이아웃 적용
87 this.setLayout(card);
88
89 start.setBackground(Color.green);
90 //main.setBackground(Color.black);
91 success.setBackground(Color.red);
```

```
99
100 //MouseHandler handler = new MouseHandler();
101 icon2 = new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\startButtonBasic.png");
102 //버튼을 만듭니다.
103 JButton start_button = new JButton("Start",icon2);
104 //버튼에 이벤트 추가
105 start_button.addActionListener(new MyEventListener());
106
107 icon3 = new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\quitButtonBasic.png");
108 //버튼을 만듭니다.
109 JButton quit_button = new JButton("Quit",icon3);
110 //버튼에 이벤트 추가
111 quit_button.addActionListener(new MyEventListener());
112
113 //텍스트용 성공 버튼
114 JButton success_button = new JButton("Success");
115 success_button.addActionListener(new MyEventListener());
116
117 //텍스트용 실패 버튼
118 JButton fail_button = new JButton("Fail");
119 fail_button.addActionListener(new MyEventListener());
120
121 //도움말 버튼
122 JButton guide_button = new JButton("도움말");
123 guide_button.addActionListener(new MyEventListener());
124
125 //success 화면과 fail화면에 돌아가기 버튼을 추가합니다.
126 JButton s_back_button = new JButton("돌아가기");
127 s_back_button.addActionListener(new MyEventListener());
128
129 JButton f_back_button = new JButton("돌아가기");
130 f_back_button.addActionListener(new MyEventListener());
131
132 success.add(s_back_button);
133 fail.add(f_back_button);
134
135 JPanel menu = new JPanel();
136 menu.setBackground(Color.LightGray);
137
138 BorderLayout border = new BorderLayout();
139 menu.setLayout(border);
140
141 JLabel blank_L1 = new JLabel(" ");
142 JLabel blank_L2 = new JLabel(" ");
143 JLabel blank_L3 = new JLabel(" ");
144 JLabel blank_L4 = new JLabel(" ");
```

구현 코드 - layout (전체 레이아웃 & KillBug & timer)

```
148
149 GridLayout grid = new GridLayout(2,1);
150 JPanel center = new JPanel();
151 center.setLayout(grid);
152
153
154
155
156 center.add(start_button);
157 start_button.setBackground(Color.white);
158
159 center.add(quit_button);
160 quit_button.setBackground(Color.white);
161
162 menu.add("East",blank_L1);
163 menu.add("South",blank_L2);
164 menu.add("North",blank_L3);
165 menu.add("West",blank_L4);
166 menu.add("Center",center);
167
168
169
170 start.setLayout(null);
171 start.add(menu);
172
173 menu.setBounds(500, 500, 450, 300);
174
175
176 base.add("Start", start);
177 base.add("Success", success);
178 base.add("Main", main.contentPane);
179 base.add("Fail", fail);
180
181
182 card.show(this, "Start");
183
184 }
185
186 @Override
187 public void actionPerformed(ActionEvent e) {
188     System.out.println("event 들어옴");
189     // 사용자가 Load 메뉴아이템을 선택하는 경우 처리할 작업 구현
190
191     if (e.getActionCommand() == "Start")
192     {
193         card.show(base, "Main");
194     }
195
196     else if(e.getActionCommand() == "Success")
197     {
198         card.show(base, "Success");
199     }
200
201 }
```

```
202
203 else if(e.getActionCommand() == "Fail")
204 {
205     card.show(base, "Fail");
206 }
207
208 else if(e.getActionCommand() == "돌아가기")
209 {
210     card.show(base, "Start");
211 }
212
213 else if(e.getActionCommand() == "도움말")
214 {
215     JFrame help_f = new JFrame();
216     help_f.setTitle("도움말");
217     // 주의, 여기서 setDefaultCloseOperation() 절의를 하지 않아야 한다
218     // 절의하게 되면 새 창을 닫으면 모든 창과 프로그램이 동시에 꺼진다
219
220 explain_icon = new ImageIcon("image/explain2.png");
221 //JLabel NewLabel = new JLabel("IT Cafe 개업방법");
222 //NewLabel.setBackground(Color.white);
223 JPanel NewWindowContainer = new JPanel() {
224     public void paintComponent(Graphics g) {
225         // Approach 1: Display image at full size
226         g.drawImage(explain_icon.getImage(), 0, 0, null);
227
228         setOpaque(false); //그림을 표시하게 설정, 투명하게 조절
229         super.paintComponent(g);
230     }
231 };
232
233 NewWindowContainer.setBackground(Color.white);
234 help_f.setContentPane(NewWindowContainer);
235
236
237 help_f.setSize(1300, 700);
238 help_f.setBackground(Color.white);
239 help_f.setVisible(true);
240
241 }
242
243 }
244
245 class KillBug extends JFrame{
246
247     ImageIcon[] originIcon = new ImageIcon[] {
248         new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\mos.png"),
249         new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\fly.png"),
250         new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\ant.png"),
251         // death Icon
252         new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\death_mos.png"),
253         new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\death_fly.png"),
254         new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\death_ant.png"),
255     };
256 }
```


구현 코드 - layout (전체 레이아웃 & KillBug & timer)

```
256# Image[] originImg = {
257     originIcon[0].getImage(),
258     originIcon[1].getImage(),
259     originIcon[2].getImage(),
260     // death Icon
261     originIcon[3].getImage(),
262     originIcon[4].getImage(),
263     originIcon[5].getImage(),
264 };
265
266# Image[] changedImg = {
267     originImg[0].getScaledInstance(50, 50, Image.SCALE_SMOOTH),
268     originImg[1].getScaledInstance(50, 50, Image.SCALE_SMOOTH),
269     originImg[2].getScaledInstance(50, 50, Image.SCALE_SMOOTH),
270     // death Icon
271     originImg[3].getScaledInstance(50, 50, Image.SCALE_SMOOTH),
272     originImg[4].getScaledInstance(50, 50, Image.SCALE_SMOOTH),
273     originImg[5].getScaledInstance(50, 50, Image.SCALE_SMOOTH),
274 };
275
276# ImageIcon[] bugimg = new ImageIcon[] {
277     new ImageIcon(changedImg[0]),
278     new ImageIcon(changedImg[1]),
279     new ImageIcon(changedImg[2]),
280     // death Icon
281     new ImageIcon(changedImg[3]),
282     new ImageIcon(changedImg[4]),
283     new ImageIcon(changedImg[5])
284 };
285
286# Bug2[] bugs = new Bug2[]{
287     new Bug2(this, bugimg[0], 0, 100),
288     new Bug2(this, bugimg[0], 300, 100),
289     new Bug2(this, bugimg[0], 600, 100),
290
291     new Bug2(this, bugimg[1], 0, 300),
292     new Bug2(this, bugimg[1], 300, 300),
293     new Bug2(this, bugimg[1], 600, 300),
294
295     new Bug2(this, bugimg[2], 0, 600),
296     new Bug2(this, bugimg[2], 300, 600),
297     new Bug2(this, bugimg[2], 600, 600),
298
299 };
300
301# Death_bug[] death_bugs = new Death_bug[] {
302     new Death_bug(this, bugimg[3], 1000, 1000),
303     new Death_bug(this, bugimg[3], 1000, 1000),
304     new Death_bug(this, bugimg[3], 1000, 1000),
305
306     new Death_bug(this, bugimg[4], 1000, 1000),
307     new Death_bug(this, bugimg[4], 1000, 1000),
308     new Death_bug(this, bugimg[4], 1000, 1000),
309 }
```

```
310     new Death_bug(this, bugimg[5], 1000, 1000),
311     new Death_bug(this, bugimg[5], 1000, 1000),
312     new Death_bug(this, bugimg[5], 1000, 1000),
313
314 };
315
316
317 timer time_label = new timer(this, 0, 0);
318
319
320 JScrollPane scrollPane;
321 ImageIcon back;
322 JButton startBt, stopBt, pauseBt, suspendBt, resumeBt;
323
324 JPanel contentPane;
325 //ImagePanel contentPane;
326 JPanel menuPanel;
327 JPanel centerPanel;
328
329 BufferedImage img = null;
330 // 일단 바닥쪽에
331 int img_x = 800;
332 int img_y = 800;
333 Thread[] ths;
334 int score;
335 int level;
336
337
338# public KillBug(int _level)
339 {
340     //super("Bug Catching Game");
341
342     // 타이머 스레드도 추가
343     //ths = new Thread[bugs.length*2];
344     ths = new Thread[bugs.length*2 + 1];
345
346     level = _level;
347
348     setDefaultCloseOperation(EXIT_ON_CLOSE);
349     setBounds(100, 10, 1300, 800);
350
351     //back = new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\grass.png");
352     //contentPane = new ImagePanel("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\house.png");
353
354     contentPane = new JPanel();
355     contentPane.setLayout(new BorderLayout(0, 0));
356     setContentPane(contentPane);
357
358     centerPanel = new JPanel() {
359         public void paintComponent(Graphics g){
360             back=new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\house.png");
361
362             g.drawImage(back.getImage(), 0, 0, 1000,800,null);
363             setOpaque(false);
364         }
365     };
```

구현 코드 - layout (전체 레이아웃 & KillBug & timer)

```
364         super.paintComponent(g);
365     };
366
367
368     scrollPane = new JScrollPane(centerPanel);
369     menuPanel = new JPanel();
370     menuPanel.setLayout(new GridLayout(1,4,5,5));
371     //1행 4열 5개로 나눈다
372     startBt = new JButton(new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\car\\play.png"));
373     stopBt = new JButton(new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\car\\stop.png"));
374     suspendBt = new JButton(new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\car\\suspend.png"));
375     resumeBt = new JButton(new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\car\\resume.png"));
376
377     menuPanel.add(startBt);
378     menuPanel.add(stopBt);
379     menuPanel.add(suspendBt);
380     menuPanel.add(resumeBt);
381
382     contentPane.add("North", menuPanel);
383
384     Toolkit tk = Toolkit.getDefaultToolkit();
385     Image img = tk.getImage("C:\\Users\\admin\\eclipse-workspace\\HW\\src\\bug_catch\\spiderweb1.png");
386     Cursor mycursor = tk.createCustomCursor(img, new Point(10,10), "snow");
387     centerPanel.setCursor(mycursor);
388
389     //centerPanel.setBackground(Color.pink);
390     //centerPanel.setSize(1000,800);
391     //centerPanel.setOpaque(false);
392
393     contentPane.add("Center", centerPanel);
394
395     time_label.setFont(new Font("HY애펬드림M", Font.BOLD+Font.ITALIC,30));
396
397     centerPanel.add(time_label);
398     time_label.setBackground(Color.pink);
399     time_label.setOpaque(true);
400
401     startBt.addActionListener(new ActionListener() {
402     public void actionPerformed(ActionEvent arg0) {
403         int level_control = (bugs.length/3)*level;
404         for (int i=0 ; i < level_control; i++) {
405             bugs[i].setVisible(true);
406             centerPanel.add(bugs[i]);
407             centerPanel.add(death_bugs[i]);
408         }
409         centerPanel.updateUI();
410         time_label.start_time = System.currentTimeMillis()/1000;
411         game_play();
412     }
413 });
414
```

```
415     stopBt.addActionListener(new ActionListener(){
416     public void actionPerformed(ActionEvent e){
417         for(Thread t:ths){
418             t.stop();
419         }
420     }
421 });
422
423     suspendBt.addActionListener(new ActionListener(){
424     public void actionPerformed(ActionEvent e){
425         for(Thread t:ths){
426             t.suspend();
427         }
428     }
429 });
430
431     resumeBt.addActionListener(new ActionListener(){
432     public void actionPerformed(ActionEvent e){
433         for(Thread t:ths){
434             t.resume();
435         }
436     }
437 });
438
439     public void game_play(){
440     makeThread();
441     for(int i=0; i<bugs.length*2+1; i++){
442         ths[i].start();
443     }
444 }
445
446     public void makeThread(){
447     for(int i=0; i<bugs.length;i++){
448         ths[i] = new Thread(bugs[i]);
449         score = 0;
450         bugs[i].addActionListener(attack);
451     }
452     for(int i=0; i<bugs.length;i++){
453         ths[bugs.length+i] = new Thread(death_bugs[i]);
454     }
455     ths[bugs.length*2] = new Thread(time_label);
456 }
457
458     ActionListener attack = new ActionListener() {
459     public void actionPerformed(ActionEvent e){
460     int level_control = (bugs.length/3)*level;
461     for (int i=0 ; i<bugs.length;i++ {
462         if( e.getSource() == bugs[i]) {
463             Thread t = ths[i];
464             t.stop();
465             death_bugs[i].x = bugs[i].x;
466             death_bugs[i].y = bugs[i].y;
467             score += 1;
468
```

구현 코드 - layout (전체 레이아웃 & KillBug & timer)

```
468 score += 1;
469 bugs[i].setVisible(false);
470 if (score == level_control*level) {
471     for(Thread ts:ths){
472         ts.stop();
473     }
474     int answer=JOptionPane.showConfirmDialog(null, "다음 단계로 넘어 가시겠습니까?", "Clear!! Congraturation!!", JOptionPane.YES_NO_OPTION);
475     if (answer==0){
476         level ++;
477         score = 0;
478         if (level == 2) {
479             //ImageIcon success = new ImageIcon("C:\\Users\\admin\\eclipse-workspace\\HM\\src\\bug_catch\\grass.png");
480             //JOptionPane.ICON_PROPERTY success = new JOptionPane.ICON_PROPERTY();
481
482             JOptionPane.showConfirmDialog(null, "모든 단계를 성공했습니다!", "Clear!! Congraturation!!", JOptionPane.CLOSED_OPTION);
483             contentPane.setVisible(false);
484             card.show(base, "Success");
485             level = 1;
486         }
487     }
488     break;
489 }
490 }
491 }
492 }
493 };
494
495 class ImagePanel extends JPanel {
496     private Image img;
497
498     public ImagePanel(String img) {
499         this(new ImageIcon(img).getImage());
500     }
501
502     public ImagePanel(Image img) {
503         this.img = img;
504         Dimension size = new Dimension(img.getWidth(null), img.getHeight(null));
505         setPreferredSize(size);
506         setMinimumSize(size);
507         setMaximumSize(size);
508         setSize(size);
509         setLayout(null);
510     }
511
512     public void paintComponent(Graphics g) {
513         g.drawImage(img, 0, 0, null);
514     }
515 }
516 }
517 }
518 }
```

```
519
520 class timer extends JLabel implements Runnable{
521     long start_time;
522     long rest_time;
523     JFrame app;
524     int x, y;
525
526     public timer (JFrame _app, int w, int h) {
527         app = _app;
528         this.setText("제한시간 : " + String.valueOf(rest_time));
529         x = w;
530         y = h;
531     }
532
533     public void run() {
534         rest_time = 60;
535         System.out.println(rest_time);
536         while(61 > rest_time) {
537             rest_time -= 1;
538             System.out.println(rest_time+ " " + System.currentTimeMillis()+ " " +start_time);
539
540             this.setText("제한 시간 : " +String.valueOf(rest_time));
541             //this.setBounds(x,y,50,50);
542
543             try {
544                 Thread.sleep(1000);
545             } catch (InterruptedException e) {
546                 // TODO Auto-generated catch block
547                 e.printStackTrace();
548             }
549             if (rest_time <= 0) {
550                 JOptionPane.showConfirmDialog(null, "시간초과", "Lose.....", JOptionPane.CLOSED_OPTION);
551                 card.show(base, "Fail");
552                 break;
553             }
554         }
555     }
556 }
557
558 }
559 }
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

구현 코드 - layout (전체 레이아웃 & KillBug & timer)

```
JFXPanel panel2 = new JFXPanel();  
Media m = new Media( source: "file:/C:/Users/SM-PC/IdeaProjects/bug_catch/src/bug_catch/bgm.mp3");  
MediaPlayer p = new MediaPlayer(m);  
p.play();
```

```
public void run() {  
    rest_time = 60;  
    System.out.println(rest_time);  
    while(61 > rest_time) {  
        rest_time -= 1;  
        System.out.println(rest_time+ " " + System.currentTimeMillis()+ " " +start_time);  
  
        this.setText("제한 시간 : " +String.valueOf(rest_time));  
        //this.setBounds(x,y,50,50);  
  
        try {  
            Thread.sleep( millis: 1000);  
        } catch (InterruptedException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
  
        if (rest_time <= 0) {  
            JFXPanel panel2 = new JFXPanel();  
            //new JFXPanel();  
            Media m = new Media( source: "file:/C:/Users/SM-PC/IdeaProjects/bug_catch/src/bug_catch/loose.mp3");  
            MediaPlayer p = new MediaPlayer(m);  
            p.play();  
            JOptionPane.showConfirmDialog( parentComponent: null, message: "시간초과", title: "Lose.....", JOptionPane.CLOSED_OPTION);  
            card.show(base, name: "Fail");  
  
            break;  
        }  
    }  
}
```

구현결과

시연

감사합니다
