

*save
alpaka*





알파카를 살려 조

1

개발

석훈,주완,대윤,주원,은지,주희

2

개발 과정

메인 게임 > 미니게임**1,2** > 여러 밸런스 > **BGM**,디자인 > 그 외..

3

개발 목표

디자인에 많은 힘을 기울여 보는 눈을 사로잡고, 여러 재밌는 코딩스터디를 하기 위하여 귀여움을 택함

4

전략

삭막한 요즘시대에 귀엽고 소소한 재미를 느낄수 있도록 하기위함

5

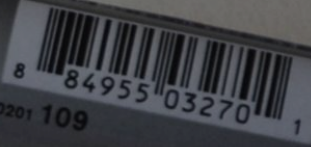
방안

디자인,**BGM**,미니게임등 심심하지 않도록 노력을 기울여 개발

Save alpaka

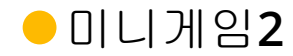
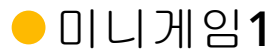
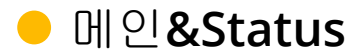
back to school
10-14 u

WINSOR
&
NEWTON
WATER COLOUR MARKER™



0201 109

CADMIUM YELLOW HUE



그외 BGM,Login



로그인,BGM

로그인 화면 캡처1: JFrame상속으로 로그인 화면 창 구현 후 USER NAME, ALPACA NAME 값 입력.

로그인 화면 캡처2: 입력받은 USER NAME, ALPACA NAME 값을 START버튼 클릭 시
메인화면으로 값을 넘겨줌.

BGM캡처 : 스레드를 이용해 로그인 화면, 메인화면, 미니게임 1, 미니게임2 에 BGM을 삽입
미니게임 종료시 다시 메인BGM을 실행.

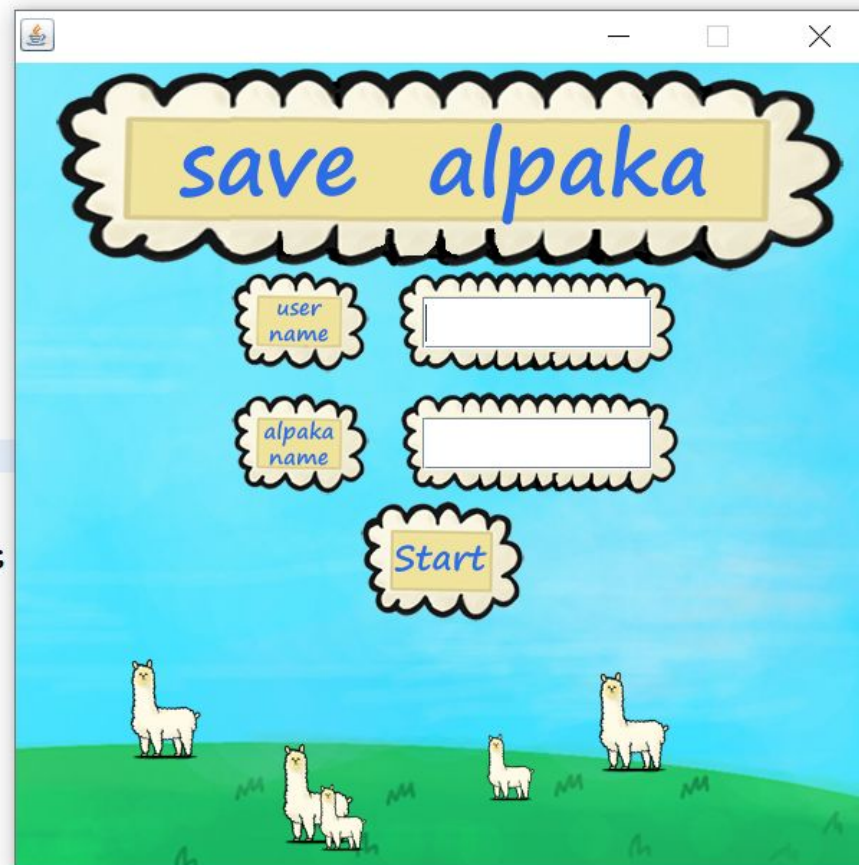




로그인1

CreateAlpaca.java

```
19 public class CreateAlpaca extends JFrame {
20
21     private static final long serialVersionUID = 9214831313090474843L;
22     MainController mainController = new MainController();
23
24     //로그인 화면
25     public CreateAlpaca() {
26         super("Save Alpaca Login");
27
28         //main bgm 시작
29         Sound mains = new Sound("mainbgm.wav",131000);
30         Sound.bgmThread = new Thread(mains);
31         Sound.bgmThread.start();
32
33         JFrame main = new JFrame();
34         main.setSize(600, 600);
35         main.setBounds(300,300,600,600);
36         setLocationRelativeTo(null);
37         main.setLayout(null);
38
39         JLabel Image = new JLabel(new ImageIcon("images/loginimages/login.png"));
40         Image.setSize(600,600);
41         main.add(Image);
42
43         //ALPACA NAME:
44         JLabel unname = new JLabel("    USER    NAME    ");
45         unname.setLocation(120,150);
46         unname.setSize(150,60);
47         Border border = BorderFactory.createLineBorder(Color.GRAY,3);
48         unname.setBorder(border);
49
50         //폰트
51         Font unnamefont = new Font("필기",Font.BOLD,16);
```



로그인 2

CreateAlpaca.java

```
77 JTextField alpacatextField = new JTextField(20);
78 alpacatextField.setOpaque(true);
79 alpacatextField.setLocation(283,247);
80 alpacatextField.setSize(160,36);
81
82 //폰트
83 Font font4 = new Font("필기",Font.BOLD,20);
84 alpacatextField.setFont(font4);
85
86 //로그인 버튼
87 JButton loginbtn = new JButton();
88 loginbtn.setLocation(255,326);
89 loginbtn.setSize(80,40);
90 loginbtn.setBackground(new Color(0,0,0,0));
91 loginbtn.setBorderPainted(false);
92 loginbtn.setOpaque(false);
93
94
95 main.add(uname);
96 main.add(aname);
97 main.add(usertextField);
98 main.add(alpacatextField);
99 main.add(loginbtn);
100
101 // SaveAlpaca와 연동
102 loginbtn.addActionListener(new ActionListener() {
103     public void actionPerformed(ActionEvent e) {
104         String usertextFieldValue = usertextField.getText(); //유저이름 받아오기
105         String alpacatextFieldValue = alpacatextField.getText(); //알파카이름 받아오기
106         mainController.createAlpaca(usertextFieldValue, alpacatextFieldValue);
107         main.dispose();
108     }
109 });
```





BGM

```
Sound.java
55 public class Sound implements Runnable{
56     public static Thread bgmThread;
57     private String musicName;
58     private int musicTime;
59     private Clip clip = null;
60
61     public Sound() {}
62
63     public Sound(String musicName, int musicTime) {
64         this.musicName = musicName;
65         this.musicTime = musicTime;
66     }
67
68     @Override
69     public void run() {
70         File bgm;
71         AudioInputStream stream;
72         AudioFormat format;
73         DataLine.Info info;
74         String path = "sound/";
75         bgm = new File(path + musicName); // bgm 폴더경로
76         try {
77             while(true) {
78                 stream = AudioSystem.getAudioInputStream(bgm);
79                 format = stream.getFormat();
80                 info = new DataLine.Info(Clip.class, format);
81                 clip = (Clip)AudioSystem.getLine(info);
82                 clip.open(stream);
83                 System.out.println("--- BGM 시작 ---");
84                 clip.start();
85                 Thread.sleep(musicTime);
86             }
87         } catch (Exception e) {
88             e.printStackTrace();
89         }
90     }
91 }
```

메인 구현

- * 메인화면 구성은 필드에 움직이는 알파카 객체를 생성하고
- * 시간에 따라 허기감소, 응가가 늘어나고 이에따라 건강도가 변경되도록 로직을 구성하였으며
- * 미니게임1,2를 실행하는 버튼, 상태창 구현, 스코어 타임 등을 보기좋게 구현하는데 힘썼습니다.
- * 이때 미니게임 1,2는 메인화면의 알파카 스테이터스와 연동되도록 구성하여
- * 사용자가 미니게임을 통하여 알파카 살리기의 목적인 알파카의 피딩을 가능케 하도록 구현하였습니다.
- * 비주얼 디렉터의 뼈와 살을 깎아내어 화면을 미려하게 구성하였습니다.





메인 화면





알파카 오토무빙

```
public void animation() {  
    Point point = alpacaPanel.getLocation();  
    if (point.x > 900 || point.x < 100) {  
        r *= -1;  
        timer.restart();  
        alpaca.setIcon(new ImageIcon("images/main/alpaca" + r + ".png"));  
    }  
    alpacaPanel.setLocation((point.x + (10 * r)), (point.y));  
}
```

똥만들기 및 초기화

```
public void setPoopLocation() {  
  
    if (poopCount < 5) {  
        poop[poopCount].setSize(75, 15);  
        Point point = poop[poopCount].getLocation();  
        poop[poopCount].setLocation((point.x) + (int)(Math.random() * 150) + 10, (point.y) + (int)(Math.random() * -20) - 1);  
  
        poop[poopCount].setVisible(true);  
  
        if (poopCount == 0) {  
            hygieneIcon.setOpaque(false);  
            hygieneIcon.setIcon(new ImageIcon("images/statusimages/hygieneicon10.png"));  
        } else if (poopCount == 1) {  
            hygieneIcon.setOpaque(false);  
            hygieneIcon.setIcon(new ImageIcon("images/statusimages/hygieneicon20.png"));  
        } else if (poopCount == 2) {  
            hygieneIcon.setIcon(new ImageIcon("images/statusimages/hygieneicon30.png"));  
            hygieneIcon.setOpaque(false);  
        } else if (poopCount == 3) {  
            hygieneIcon.setIcon(new ImageIcon("images/statusimages/hygieneicon40.png"));  
            hygieneIcon.setOpaque(false);  
        } else if (poopCount == 4) {  
            hygieneIcon.setIcon(new ImageIcon("images/statusimages/hygieneicon50.png"));  
            hygieneIcon.setOpaque(false);  
        }  
  
        System.out.println(poopCount + "번째 똥");  
        poopCount++;  
  
    } else {  
        for (int i = 0; i < poop.length; i++) {  
            poop[i].setVisible(false);  
            hygieneIcon.setIcon(new ImageIcon());  
            hygieneIcon.setOpaque(false);  
        }  
  
        mainController.PoopDamage(-10);  
        poopCount = 0;  
    }  
}
```



똥 치우기

```
public void removePoop() {  
    poopTimer.stop();  
    setPoopCount(0);  
    for (int i = 0; i < poop.length; i++) {  
  
        poop[i].setVisible(false);  
        hygieneIcon.setIcon(new ImageIcon());  
        hygieneIcon.setOpaque(false);  
  
    }  
    System.out.println("똥 다 지워짐");  
    poopTimer.start();  
}
```




게임 클리어 ,오버

```
public void playTimer() {
    playTime++;

    if(playTime==300) { //300초가 지나면 게임 클리어
        backfiled.setIcon(new ImageIcon("images/main/GameClear.png"));
        cleenPoop.setVisible(false);
        alpacaPanel.setVisible(false);
        poopPanel.setVisible(false);
        mainButtons.getGamePanel1().setVisible(false);
        mainButtons.getGamePanel2().setVisible(false);
        status.getLabelPane().setVisible(false);
        hygieneIcon.setVisible(false);

        timer.stop();
        poopTimer.stop();
        closeTimer.stop();
        playTimer.stop();
    }
}
```

```
public void close() {
    if(mainController.getHealth() == 40) {
        backfiled.setIcon(new ImageIcon("images/main/GameOver.png"));
        cleenPoop.setVisible(false);
        alpacaPanel.setVisible(false);
        poopPanel.setVisible(false);
        mainButtons.getGamePanel1().setVisible(false);
        mainButtons.getGamePanel2().setVisible(false);
        status.getLabelPane().setVisible(false);
        hygieneIcon.setVisible(false);

        timer.stop();
        poopTimer.stop();
        closeTimer.stop();
        playTimer.stop();
    }
}
```

Mini game 1

- *게임 화면이 시작하는 동시에 실행 시간이 작동하고,
20개의 숫자를 다 누름과 동시에 실행시간이 끝나는 타이머 구현.
- *배열로 20개의 숫자가 나올 수 있게 만들었고,
숫자를 클릭하게되면 클릭한 숫자는 사라지고 다음 숫자만 클릭가능하게 만들었습니다.
- *숫자의 위치는 x,y를 사용 하여 랜덤으로 위치
- *화면에 실행시간 및 실행시간에 따라 성공 or 실패 > 성공시 배고픔 +1상승





미니게임 1

숫자 누르기!

Tip!



를 이용하여

숫자 0~19까지

빠르게 눌러

지우세요!!



Game Start

push!!

1

2

4

3





숫자클릭,실행시간

```
public Game1() {

    super("game 중");
    long start = System.currentTimeMillis(); // 게임 실행시작과 동시에 타임 작동
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setResizable(false);

    c = getContentPane();
    c.setLayout(null);

    for(int i=0; i<gameLabels.length; i++) {
        String numstr = Integer.toString(i);
        gameLabels[i] = new JLabel(numstr,new ImageIcon("images/MiniGame_01_images/1.png"),JLabel.CENTER);
        gameLabels[i].setVerticalTextPosition(0);
        gameLabels[i].setHorizontalTextPosition(0);
        gameLabels[i].setFont(new Font("고딕",Font.BOLD,22));
        gameLabels[i].setLayout(null);
        gameLabels[i].setForeground(Color.black);

        c.add(gameLabels[i]);

        gameLabels[i].addMouseListener(new MouseAdapter() {

            @Override
            public void mousePressed(MouseEvent e) {
                JLabel la = (JLabel)e.getSource();
                if(Integer.parseInt(la.getText()) == nextPressed) {
                    nextPressed++;
                    if(nextPressed == 21) {

                    }else {
                        la.setVisible(false);
                    }
                    if(nextPressed == 20) {
                        long end = System.currentTimeMillis();
                        result = ((end-start)/1000);
                        System.out.println("실행시간 : " + result);

                        End tm2 = new End();
                        tm2.Timer3(result);

                        dispose();
                    }
                }
            }
        });
    }
}
```



랜덤 숫자 위치

```
public long getResult() {  
    return result;  
}  
  
public void setResult(long result) {  
    this.result = result;  
}  
  
private void configure() {  
    for(int i=0; i<gameLabels.length; i++) {  
        gameLabels[i].setSize(45,45);  
        int x = (int)(Math.random()*470)+40;  
        int y = (int)(Math.random()*310)+70;  
        gameLabels[i].setLayout(null);  
        gameLabels[i].setLocation(x,y);  
        gameLabels[i].setVisible(true);  
    }  
}
```



런타임 결과값

```
pan.setBounds(0,0,600,600);

JLabel lbl = new JLabel(" " + result);
lbl.setBounds(280, 139, 230, 40);
lbl.setForeground(Color.pink);
lbl.setFont(new Font("굴림",Font.BOLD,35));

JLabel lbl2 = new JLabel();

int num = (int)result;
if(num > 0 && num <= 15) {
    lbl2 = new JLabel("Win!!");
    System.out.println(" 올~ 성공추키팡키!");
    mc.gameHungerResult(10);

} else if(num > 15){
    lbl2 = new JLabel("Fail");
    System.out.println(" 실패지를 ^__^");

}
lbl2.setForeground(Color.pink);
lbl2.setBounds(330, 240, 260, 40);
lbl2.setFont(new Font("굴림",Font.BOLD,35));

pan.add(lbl);
pan.add(lbl2);

JLabel game1Image = new JLabel(new ImageIcon("images/MiniGame_01_images/6.png"));
game1Image.setSize(600,600);
pan.add(game1Image);
game1Image.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        frame.dispose();
    }
});
}
```


Mini Game2

- * 미니게임 2는 필드에서 위에서 좌로 움직이는 횡스크롤식 장애물 피하기 게임입니다.
- * 움직이는 플레이어캐릭터와 장애물을 구현하기 위해
- * 타이머를 사용하여 일정 시간마다 각각의 좌표를 조금씩 반복적으로 움직이게 구현하여
- * 실제로 눈으로 보았을때 요소들이 움직이는것 처럼 구현하기 위해 노력하였습니다.
- * 각각 장애물 마다의 게임성을 위한 랜덤한 요소(장애물 생성위치,속도)가 적용되었으며
- * 이를 각각 구현하려고 노력하다보니 매우 많은 배움을 얻을 수 있었고
- * 비주얼 디렉터와 그래픽 디자이너의 피담으로 보다 훌륭한 퀄리티의 게임을 제작할 수 있었습니다.



미니 게임2





알파카 무빙

```
for (int i = 0; i < obstacleArr.length; i++) {
    Rectangle obstacleR = obstacleArr[i].getBounds();
    if (alpacaR.intersects(obstacleR)) {
        System.out.println("코뿔소에 부딪혀서 알파카가 아파해요!! " + "체력 : " + life);
        Alpaca_mini.setLocation(0, 0);
        AlpacaRunTimer.stop();
        AlpacaL.setIcon(new ImageIcon("images/MiniGame_02_images/alpacaG2Die.png"));
        AlpacaRetrunTimer.start();

        // 알파카 이미지와 로케이션이 변경된 후 2초 뒤에 원래 위치, 원래 이미지로 바뀌어야함.
        // Alpaca_mini.setLocation(Alpaca.getLocation());

    }
}

for (int i = 0; i < snakeArr.length; i++) {
    Rectangle snakeR = snakeArr[i].getBounds();
    if (alpacaR.intersects(snakeR)) {
        FLYING_UNIT = 10;
        System.out.println("독사에 물려서 느려졌어요!!!");
        Alpaca_mini.setLocation(0, 0);
        AlpacaRunTimer.stop();
        AlpacaL.setIcon(new ImageIcon("images/MiniGame_02_images/alpacaG2_1slow.png"));
        SnakeSlowTimer.start();
        // 알파카 이미지와 로케이션이 변경된 후 2초 뒤에 원래 위치, 원래 이미지로 바뀌어야함.
        // Alpaca_mini.setLocation(Alpaca.getLocation());

    }
}
```




게임 인터페이스

* @메소드 설명 : 알파카가 맵 밖으로 나가는지 감시. 알파카 캐릭터와 장애물의 충돌을 감지하고 라이프 포인트를 1씩 감소. 뱀에게 물렸을 시 알파카 캐릭터의 일정시간 속도 저하. 라이프 포인트가 0이 되었을 때 게임오버 출력. 게임종료후 결과값 메인컨트롤러에 리턴.

```
=====*/  
public void checkcollisionTimer() {  
    count++;  
    Alpaca.requestFocus();  
  
    Rectangle alpacaR = Alpaca_mini.getBounds();  
  
    // 알파카가 맵밖으로 나가는지 감시하는 조건문.  
    if (Alpaca.getY() <= 80) {  
        Alpaca.setLocation(Alpaca.getX(), Alpaca.getY() + 50);  
    }  
    if (Alpaca.getY() >= 750) {  
        Alpaca.setLocation(Alpaca.getX(), Alpaca.getY() - 50);  
    }  
    if (Alpaca.getX() <= 100) {  
        Alpaca.setLocation(Alpaca.getX() + 50, Alpaca.getY());  
    }  
    if (Alpaca.getX() >= 1280) {  
        Alpaca.setLocation(Alpaca.getX() - 50, Alpaca.getY());  
    }  
  
    obstacle1_mini = new JPanel();  
    obstacle1_mini.setSize(180, 105);  
    obstacle1_mini.setLocation(obstacle1.getLocation());  
    obstacle1_mini.setVisible(true);  
}
```

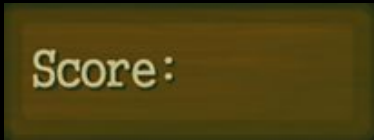
Design,gui

직관적으로 귀엽게, 이쁘게
그리는게 목표

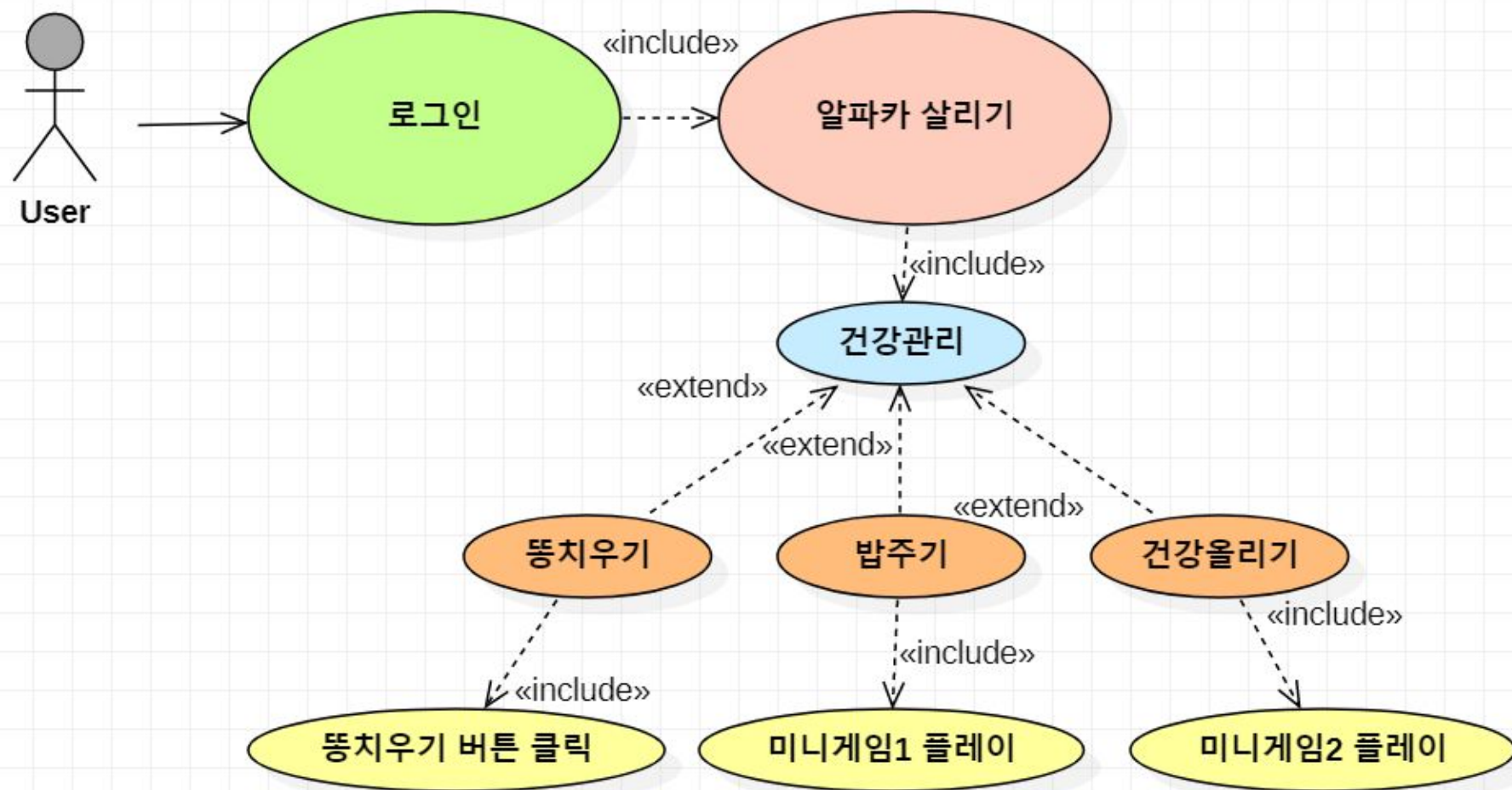


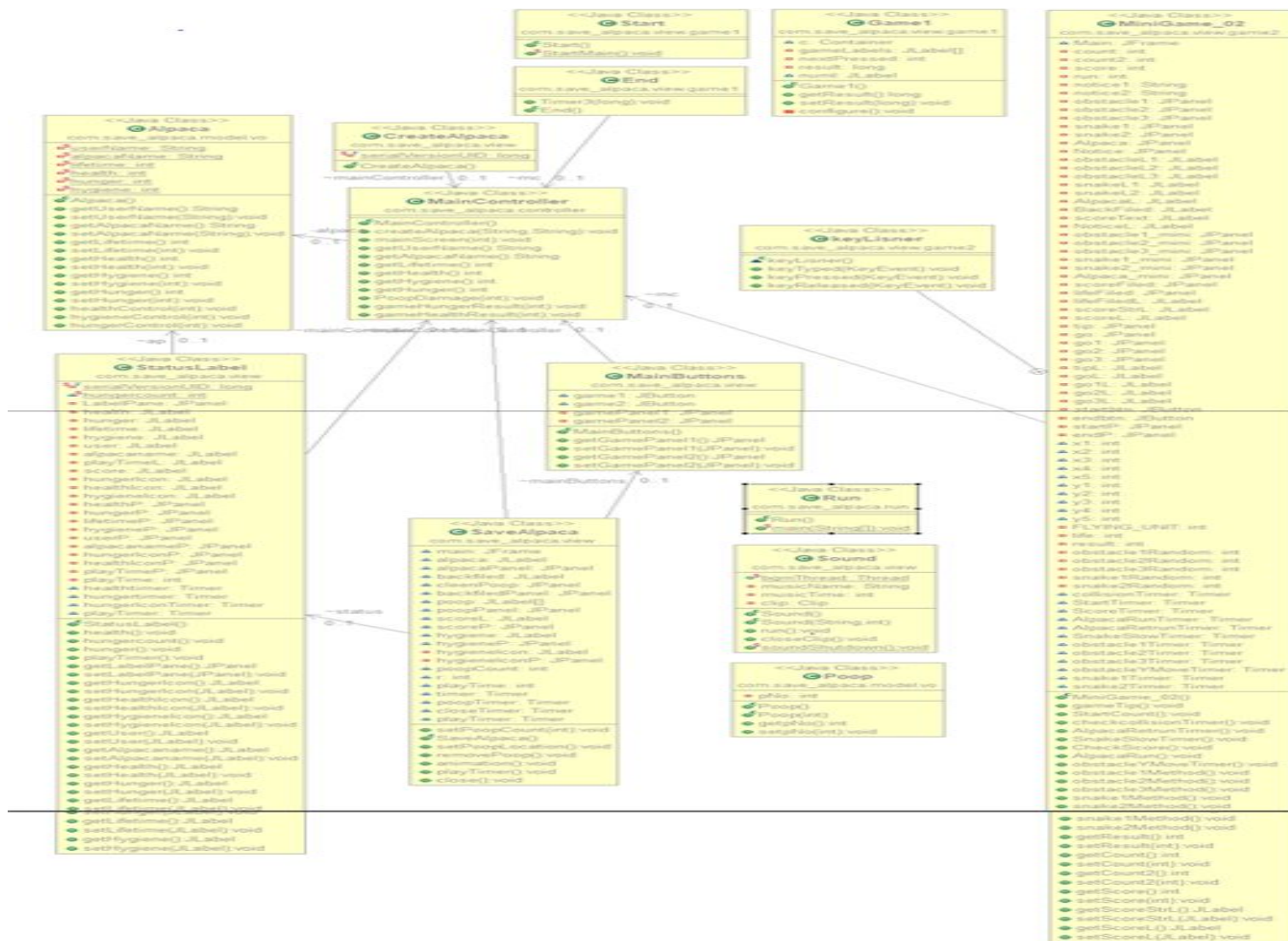


디자인



유스케이스 다이어그램







Thank You