

<구현 순서>

- 1) Frame 및 Panel 생성: JavaHW5, HW5MainPanel 클래스
- 2) Panel 추상 클래스 정의: HW5Panel 클래스
- 3) HW5Panel을 상속받는 HW5TitlePanel, HW5GamePanel, HW5GameOverPanel 클래스 생성
- 4) Object 추상 클래스 정의: HW5Object 클래스
- 5) HW5Object를 상속받는 HW5Wall, HW5Ball, HW5Brick, HW5Racket 클래스 생성
- 6) 블록 깨기 게임 기능 구현: HW5MainPanel 클래스 생성
- 7) Sound 추가

1. Frame 및 Panel 생성

- 1) public 클래스에서 JFrame 클래스를 상속받고, main 함수에서 JavaHW5 클래스를 객체화한다.

```
public class JavaHW5 extends JFrame {
    public static void main(String[] args){ new JavaHW5(); }
}
```

2. HW5Panel 추상 클래스 정의

- 1) 게임 화면 관련된 클래스에서 상속받을 HW5Panel 추상 클래스를 정의한다.

```
abstract class HW5Panel extends JPanel {
    HW5MainPanel main;
    Dimension d;
    public HW5Panel(HW5MainPanel panel) { main = panel; }
    abstract void draw(Graphics g); // 화면 그리기
    abstract void setDimension(Dimension _d); // 화면 크기 설정
}
```

- 2) HW5MainPanel에 접근하기 위한 변수와 Panel의 가로세로 길이를 알기 위한 변수를 갖는다.
- 3) 각 Panel에 맞게 화면을 그리기 위한 draw 함수를 갖는다.

* 시행착오

- HW5MainPanel의 생성자 안에서 각각의 Panel을 생성하므로, 모두 Panel의 크기를 알 수 없다.
따라서 나중에 설정해주기 위해 setDimension 함수를 추가했다.

3. HW5Panel 클래스 상속: HW5TitlePanel, HW5GamePanel, HW5GameOverPanel

클래스명	기능	설명
HW5TitlePanel	타이틀 화면	게임 타이틀을 그리고, 게임 시작 음악을 재생한다.
HW5GamePanel	게임 화면	게임 화면을 그리고, 단계, 현재 점수, 최고 점수 변수를 가진다. 라켓이나 벽돌에 충돌한 효과음, 다음 단계로 넘어가는 효과음을 재생한다.
HW5GameOverPanel	게임 오버 화면	게임 오버 화면을 그리고, 공이 떨어지는 효과음을 재생한다.

1) HW5TitlePanel



- 생성자 안에서 게임 시작 음악을 clip으로 불러온다.

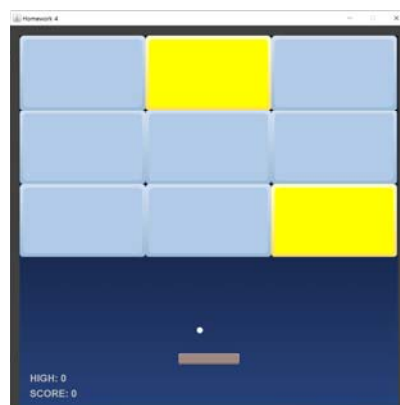
```
public HW5TitlePanel(HW5MainPanel panel) {  
    super(panel);  
    try {  
        clip = AudioSystem.getClip();  
        url = getClass().getClassLoader().getResource("gamestart.wav");  
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(url);  
        clip.open(audioStream);  
    } catch (UnsupportedAudioFileException | IOException e) {  
        e.printStackTrace();  
    } catch (LineUnavailableException e) {  
        e.printStackTrace();  
    }  
}
```

- 게임 시작 음악은 startBGM, stopBGM 함수를 통해서만 실행하거나 정지할 수 있다.

```
void startBGM() { clip.setFramePosition(0); clip.start(); }  
void stopBGM() { clip.stop(); }
```

- draw 함수를 오버라이딩해 게임 이름과 안내 메시지를 출력한다.
- 스페이스 바를 누르면 게임이 시작된다는 안내 메시지는 blink 변수를 통해 true일 때만 출력해 깜빡이는 것처럼 보이도록 한다.

2) HW5GamePanel



- 스테이지, 최고 점수, 현재 점수 변수를 private으로 만들고, 생성자 안에서 초기화한다.
- 현재 점수와 최고 점수를 확인할 수 있는 함수와 벽돌을 깨는 경우 점수를 얻는 함수를 구현한다.
- 벽돌을 모두 깬 경우 다음 스테이지로 넘어갈 수 있도록 함수를 구현한다.
- 게임 재시작 시 스테이지와 현재 점수를 초기화할 수 있도록 resetGame 함수를 구현한다.

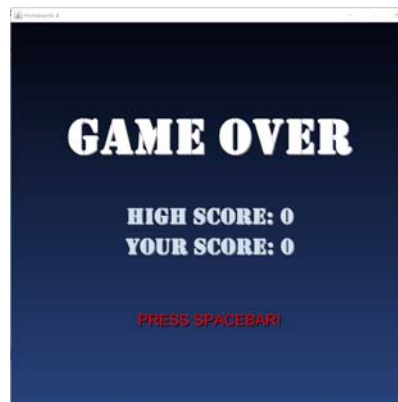
```
private int stage;
private int high;
private int score;
void nextStage() { stage++; }
int getStage() { return stage; }
int getScore() { return score; }
int getHighScore() { return high; }
void earnScore() { score += 10; if(high < score) high = score; }
void resetGame() { stage = 1; score = 0; }
```

- 생성자 안에서 라켓, 벽돌과 충돌한 효과음, 다음 스테이지로 넘어가는 효과음을 clip으로 불러온다.
- 각 효과음은 함수를 통해서 재생할 수 있다.

```
private Clip clipR, clipB, clipS;
void racketSound() { clipR.setFramePosition(0); clipR.start(); }
void brickSound() { clipB.setFramePosition(0); clipB.start(); }
void stageSound() { clipS.setFramePosition(0); clipS.start(); }
```

- draw 함수를 오버라이딩해 게임에 필요한 모든 물체와 현재 점수, 최고 점수를 그린다.

3) HW5GameOverPanel



- 생성자 안에서 공이 떨어지는 효과음을 clip으로 불러온다.
- 게임 시작 음악은 startBGM, stopBGM 함수를 통해서만 실행하거나 정지할 수 있다.

```
void startBGM() { clip.setFramePosition(0); clip.start(); }
void stopBGM() { clip.stop(); }
```

- draw 함수를 오버라이딩해 게임 오버 안내 메시지와 최고 점수, 현재 점수를 출력한다.
- 스페이스 바를 누르면 게임이 시작된다는 안내 메시지는 blink 변수를 통해 true일 때만 출력해 깜빡이는 것처럼 보이도록 한다.

* 시행착오

- 게임에 필요한 물체를 모두 HW5MainPanel 클래스에서 그리도록 구현했었지만, 게임 화면과 관련된 물체를 그리는 것이라고 생각해 HW5GamePanel의 draw() 함수 안에서 그리도록 구현했다.

4. HW5Object 추상 클래스 정의

1) 게임에 사용되는 물체와 관련된 클래스에서 상속받을 HW5Object 추상 클래스를 정의한다.

```
abstract class HW5Object {
    double x, y;
    Color color;
    public HW5Object(double _x, double _y, Color c) {
        x = _x;
        y = _y;
        color = c;
    }
    abstract void draw(Graphics g);
    void update(double dt) {}; // update
    void collisionResolution(HW5Object o) {};
}
```

2) 물체의 위치를 나타내는 x, y 변수와 색깔을 나타내는 color 변수를 갖는다.

3) 물체를 그리기 위한 draw 함수를 갖는다.

4) 물체의 위치를 변경하기 위한 update와 collisionResolution 함수는 선택적으로 구현할 수 있다.

* 시행착오

- update 함수도 abstract 함수로 구현했었지만, 벽이나 벽돌의 위치는 변하지 않기 때문에 선택적으로 구현할 수 있도록 변경했다.

5. HW5Object 클래스 상속: HW5Ball, HW5Brick, HW5Wall, HW5Racket

클래스명	기능	설명
HW5Ball	공	공의 반지름과 속도 변수를 추가로 가진다. 위치와 반지름, 색깔 값에 따라 공을 그리는 함수를 구현한다. 공의 위치를 변경하는 함수, 다른 물체와 충돌하는지 확인하고 위치와 방향을 수정하는 함수를 가진다.
HW5Brick	벽돌	벽돌의 가로세로 길이 변수와 그라데이션 색상 변수를 추가로 가진다. 벽돌의 위치와 가로세로 길이, 색깔 값에 따라 벽돌을 그리는 함수를 구현한다.
HW5Wall	벽	벽의 가로세로 길이 변수를 추가로 가진다. 벽의 위치와 가로세로 길이, 색깔 값에 따라 벽을 그리는 함수를 구현한다.
HW5Racket	라켓	라켓의 가로세로 길이 변수를 추가로 가진다. 라켓의 위치와 가로세로 길이, 색깔 값에 따라 라켓을 그리는 함수를 구현한다. 라켓의 위치를 변경하는 함수를 가진다.

1) HW5Ball

- 생성자 안에서 공의 반지름과 방향, 속도를 정한다.
- draw 함수를 오버라이딩해 흰색으로 x, y 좌표를 중심으로 하는 공을 그린다.
- update 함수를 오버라이딩해 공의 이전 위치를 저장하고, 현재 위치를 수정한다.
- isCollide 함수를 구현해 공이 다른 물체와 충돌하는지 확인한다.
- collisionResolution 함수를 오버라이딩해 공이 벽, 벽돌, 라켓과 충돌하는 경우 방향을 수정한다.
- 공이 라켓의 양쪽 끝에 부딪힌 경우, 부딪힌 방향으로 공을 다시 튕긴다. 라켓의 중앙에 부딪힌 경우, 단순히 반사되어 튕기도록 한다.

```

HW5Racket racket = (HW5Racket) o;
int range = racket.w / 5;
double dist = x - (racket.x + (racket.w / 2)); // 바 중심과 공 거리
if (dist < -range * 2) vx = 350 * Math.cos(140);
else if(dist < -range) vx = 350 * Math.cos(110);
else if(dist > range && dist < range*2) vx = 350 * Math.cos(70);
else vx = 350 * Math.cos(40);

```

2) HW5Brick

- 생성자 안에서 벽돌 색깔에 맞춰 그라데이션 색깔 변수를 초기화한다.
- draw 함수를 오버라이딩해 그라데이션으로 벽돌을 그린다.

3) HW5Wall

- draw 함수를 오버라이딩해 단색으로 벽을 그린다.

4) HW5Racket

- draw 함수를 오버라이딩해 그라데이션으로 라켓을 그린다.
- update 함수를 오버라이딩해 라켓이 양옆으로 움직일 수 있도록 x 좌표를 수정한다.

* 시행착오

- 공이 라켓과 충돌하는 경우, 벽이나 벽돌과 똑같이 방향을 바꾸는 것으로 구현했지만, 게임의 재미를 위해 라켓과 충돌한 위치에 따라 각도를 조절하는 것으로 변경했다.

6. 블록 깨기 게임 기능 구현: HW5MainPanel 클래스

- 1) HW5MainPanel 클래스를 생성하고, JavaHW5 클래스에서 Frame에 추가한다.
- 2) 생성자 안에서 게임과 관련된 타이틀, 게임, 게임 오버 클래스를 생성한다.

```

title = new HW5TitlePanel(this);
game = new HW5GamePanel(this);
over = new HW5GameOverPanel(this);

```

3) 게임에 사용될 물체를 저장할 LinkedList를 생성한다.

```

objs = new LinkedList<>(); // 라켓, 벽, 벽돌
balls = new LinkedList<>(); // 공
newBalls = new LinkedList<>(); // 노랑 벽돌과 충돌해 새로 추가될 공

```

4) KeyListener를 등록한다. 스페이스 바를 누르면 화면이 변경되고, 방향키를 누르면 라켓이 이동될 수 있게 구현한다.

```

public void keyPressed(KeyEvent e) {
    if(e.getKeyCode() == KeyEvent.VK_SPACE) { mode = (mode + 1) % 3; }
    if(e.getKeyCode() == KeyEvent.VK_LEFT && mode == 1) { // 게임 중에만 동작
        objs.get(0).update(-15); // 라켓 왼쪽으로 이동
        repaint();
    }
    if(e.getKeyCode() == KeyEvent.VK_RIGHT && mode == 1) { // 게임 중에만 동작
        objs.get(0).update(15); // 라켓 오른쪽으로 이동
        repaint();
    }
}

```

5) 스레드를 생성하고, run 함수를 오버라이딩해 구현한다.

- mode가 0인 경우, HW5TitlePanel의 스페이스바 문자열이 깜빡일 수 있도록 250ms마다 blink 변수의 값을 변경한다.

```
while(true) {  
    Thread.sleep(250);  
    title.blink = !title.blink;  
    repaint();  
}
```

- mode가 1인 경우, 16ms마다 공의 위치를 변경한다. LinkedList에 공이 남아있으면 게임을 진행 하고, 남아있지 않으면 mode를 2로 변경한다.

```
while(true) {  
    updateBall();  
    if(balls.size() == 0) {  
        over.startBGM();  
        mode = 2;  
    }  
    else playGame(game.getStage());  
    repaint();  
    Thread.sleep(16);  
}
```

- updateBall 함수에서는 공의 위치를 변경하고, 화면 밖으로 나간 경우 LinkedList에서 삭제한다.
- playGame 함수에서는 공과 충돌하는 물체를 확인해 공의 방향을 변경하고 노란 벽돌과 충돌하면 공을 2개 추가한다. 남아있는 벽돌이 없으면 다음 단계로 넘어가고, 모든 물체를 단계에 맞게 새로 생성한다.
- mode가 2인 경우, HW5GameOverPanel의 스페이스바 문자열이 깜빡일 수 있도록 200ms마다 blink 변수의 값을 변경한다.

6) paintComponent 함수를 오버라이딩해 모드에 따라 화면을 그린다.

```
if(mode == 0) title.draw(g);  
if(mode == 1) game.draw(g);  
if(mode==2) over.draw(g);
```

* 시행착오

- 화면이 전환될 때, frame에서 panel을 삭제하고 add 하는 방식으로 구현했었다가 HW5MainPanel 에서 모드에 따라 각 panel의 draw 함수를 호출하는 것으로 변경했다.
- panel마다 스레드를 가지고 실행하는 방식으로 구현했는데, 화면이 전환되면서 글씨가 깜빡이지 않거나 하는 문제가 발생했다. HW5MainPanel 한 곳에서 스레드를 실행하는 방식으로 변경했다.

7. Sound 추가

1) 게임 시작 배경 음악 추가

- HW5TitlePanel에서 게임 시작 음악을 clip으로 불러온다.
- HW5MainPanel에서 스레드가 시작될 때, mode가 2에서 0으로 바뀌었을 때 음악을 실행한다.
- HW5MainPanel에서 mode가 0에서 1로 바뀌었을 때 음악을 멈춘다.

2) 공이 벽돌과 충돌하는 효과음 추가

- HW5GamePanel에서 효과음을 clip으로 불러온다.
- HW5MainPanel에서 playGame 함수에서 공과 물체의 충돌을 확인할 때, 충돌한 물체가 벽돌이라면 해당 효과음을 실행한다.

3) 공이 라켓과 충돌하는 효과음 추가

- HW5GamePanel에서 효과음을 clip으로 불러온다.
- HW5MainPanel의 playGame 함수에서 공과 물체의 충돌을 확인할 때, 충돌한 물체가 라켓이라면 해당 효과음을 실행한다.

4) 다음 단계로 넘어가는 효과음 추가

- HW5GamePanel에서 효과음을 clip으로 불러온다.
- HW5MainPanel의 playGame 함수에서 남아있는 벽돌이 없으면 해당 효과음을 실행한다.

5) 게임 오버 효과음 추가

- HW5GameOverPanel에서 효과음을 clip으로 불러온다.
- HW5MainPanel의 스레드에서 LinkedList에 남아있는 공이 없으면 해당 효과음을 실행한다.

* 시행착오

- 효과음을 모두 HW5MainPanel에서 구현했었지만, 각 화면에서 구현해야 하는 기능이라고 생각했다. 각 화면의 클래스로 사운드와 관련된 코드를 이동하고, 효과음을 재생하는 함수를 구현해주었다.