자바 프로그래밍

기말 프로젝트 최종 보고서 ‘미니 공막기 게임’

정보보호학과 2017111370 이혜민

목차

1. 프로젝트 주제
2. 기존 프로그램의 분석 결과 및 기획 내용
3. 소스 코드
4. 실행 결과
5. 완성도
6. 후기 (소감)

1. 프로젝트 주제

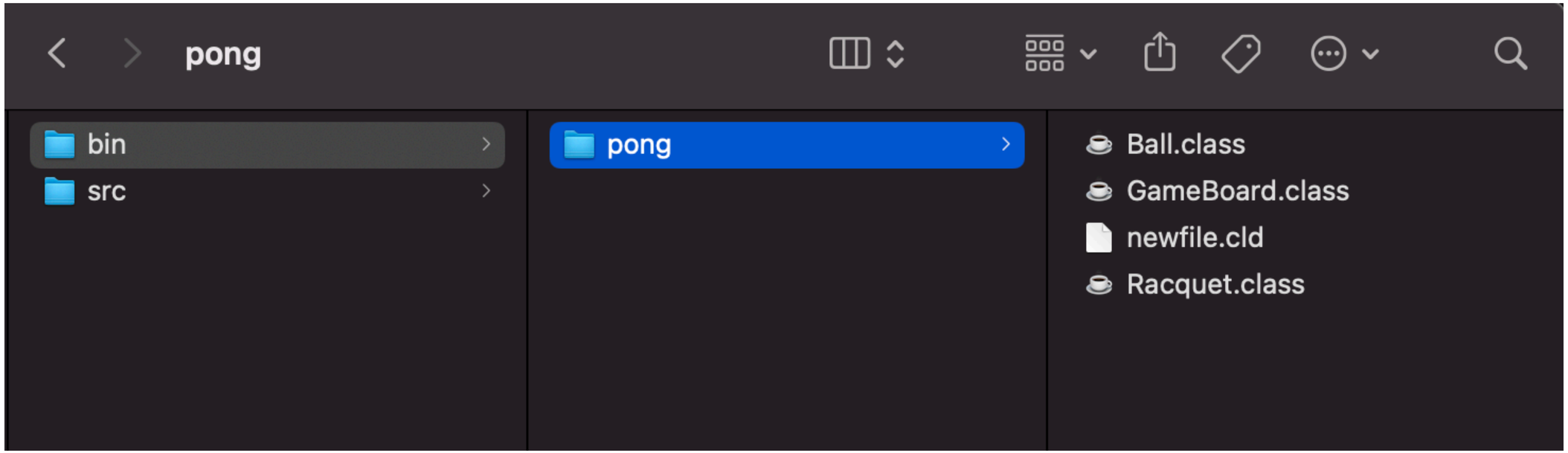
‘미니 공막기 게임’ 소개

- 두 개의 사이드 막대로 벽으로 향해 튕기는 여러 개의 공들을 막는 게임

2. 기존 프로그램 분석 결과 및 기획 내용

- 기존 프로그램 출처

- 수업 초 교수님께서 공유해주신 ‘source’ zip의 **chap13 ‘pong’ 프로젝트**



2. 기존 프로그램 분석 결과 및 기획 내용

- 기존 프로그램 분석

1. 공의 개수는 1개, 대각선 방향으로 움직임
   - 처음 위치는 x,y 좌표가 (0,0)에서 시작함

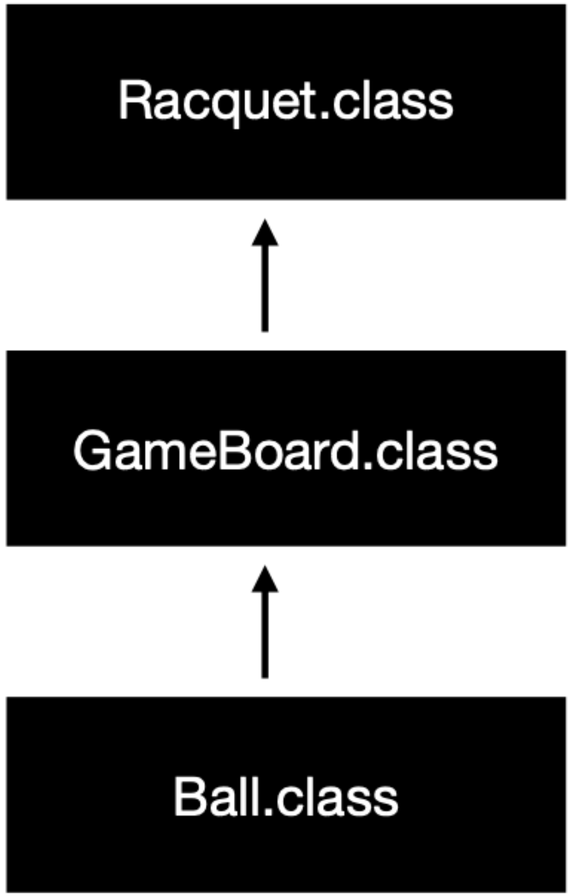
2. **공이 벽에 충돌하면 가고 있던 방향의 좌/우, 상/하의 반대로 방향이 전환**되어 움직임

3. 공이 사이드 바에 닿지 않고 좌/우 벽에 충돌하면?
   - 프로그램이 종료되거나 하는 변화 없이 2번을 그대로 수행

4. 공이 사이드 바에 닿는다면?
   - 2번처럼 사이드 바에 닿은 후 방향이 전환되어 움직임

2. 기존 프로그램 분석 결과 및 기획 내용

- 프로그램 구조



- KeyRelesead, KeyPressed 함수를 사용하여 키보드의 움직임을 받아와 변화를 정의함
- move() 정의하여 사이드 바의 움직임을 정의함

- Ball의 객체를 선언하여 각각의 색깔 및 초기 위치를 선언해줌
- main 함수에서 JFrame, JLabel에 대한 객체 선언 및 다양한 인터페이스 설정을 해줌 (fail 변화 설정)

- Ball.class에서 공의 좌표, 움직임의 스피드, 색깔등 변수 선언
- move()를 통해서 좌/우, 상/하 범위에 따라서 벽과 바에 닿는 경우 어떤 실행을 할지 지정해줌

2. 기존 프로그램 분석 결과 및 기획 내용

- 분석 (사용된 라이브러리 종류 및 의미)

- import java.awt.Color - 색을 표현하기 위해서 사용되는 라이브러리
- import java.awt.Graphics2D - 2D그래픽을 표현하기 위한 라이브러리
- import java.awt.Rectangle - Rectangle객체의 좌표 공간에서의 좌상의 점 (x, y), 좌표 공간내의 영역을 지정하는 라이브러리

- import javax.swing.JFrame - 프레임을 불러오는 라이브러리
- import javax.swing.JPanel - 패널을 불러오는 라이브러리

- import java.awt.event.KeyEvent - 키보드 키를 누를 때마다 해당 클래스의 개체를 KeyListener에 알리는 인터페이스
- import java.awt.event.KeyListener - KeyListener 인터페이스는 주요 이벤트를 수신하고 일부 작업을 수행하는 인터페이스

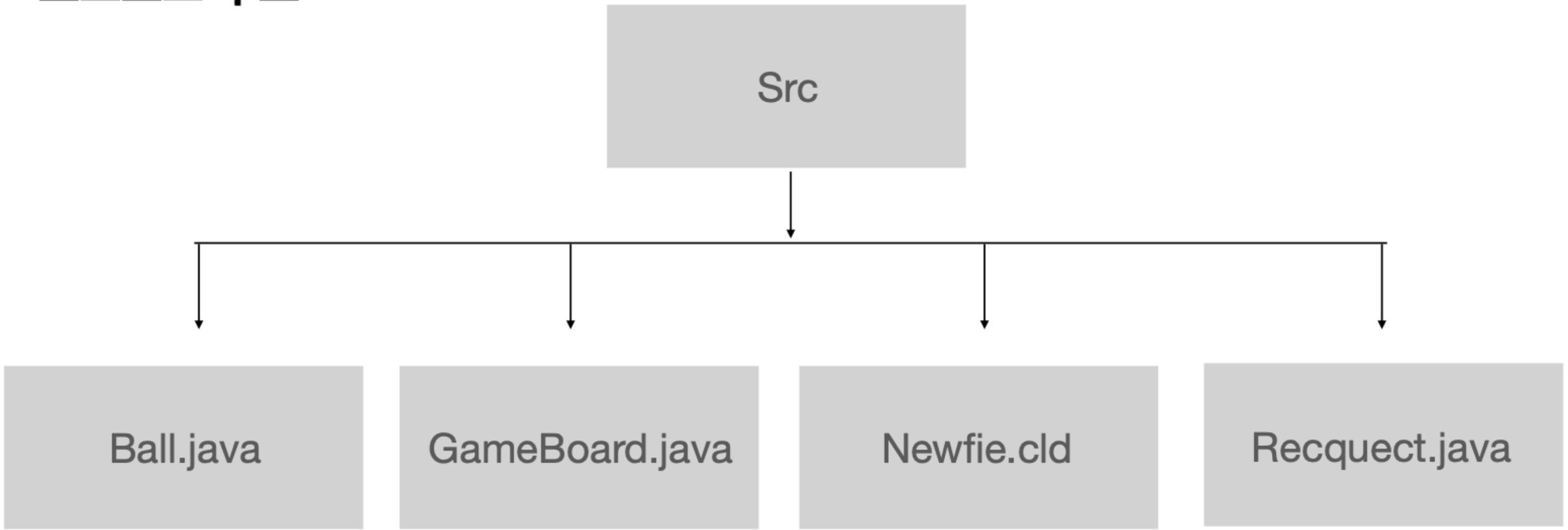
2. 기존 프로그램 분석 결과 및 기획 내용

- 기획 내용

- 공의 개수 늘리기 - 현재 Ball의 개수를 3개로 늘리기
- 공의 색깔과 모양 다양하게 변경하기
- 공이 좌,우 양쪽 바가 아닌 벽에 닿으면 “Fail” 창 뜨도록 변경하기
- 공이 좌,우 양쪽 바가 아닌 벽에 닿으면 배경 색깔 다르게 바뀌도록 변경하기

3. 소스코드

- 소스코드 구조



소스코드 깃헙 : https://han.gl/fMlrg

3. 소스코드

- Ball.java

```
package pong;

import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import javax.swing.*;

public class Ball extends JFrame {

        private static final int RADIUS = 20;
        int x = 0;
        int y = 0;
        int xspeed = 1;
        int yspeed = 1;
        private GameBoard game;
        boolean hitwall = false;
        Color color;

        public Ball(GameBoard game, Color color) {
                this.game = game;
                this.color = color;
        }

        void move() {
                if (y + yspeed < 0)
                        yspeed = 1;
                if (y + yspeed > game.getHeight() - 2 * RADIUS)
                        yspeed = -1;
                if (collision()) {
                        xspeed = -xspeed;
                        //System.out.println("HITRacquet");
                } else if ((x + xspeed < 0 || x + xspeed > game.getWidth() - 2 *
RADIUS) && game.getWidth() > 0) {
                        //System.out.println(("HITWall"));
                        hitwall = true;
                        game.setBackground(Color.pink);
                        game.deleteballs();
                } else {
                        //System.out.println("Moving");
                }
                x = x + xspeed;
                y = y + yspeed;
        }
```

소스코드 깃헙 : https://han.gl/fMlrg

3. 소스코드

- Ball.java

```
private boolean collision() {
                return game.racquet1.getBounds().intersects(getBounds())
                                || game.racquet2.getBounds().intersects(getBounds());
        }


        public void draw(Graphics2D g) {
                g.setColor(color);
                g.fillOval(x, y, 2 * RADIUS, 2 * RADIUS);
        }

        public Rectangle getBounds() {
                return new Rectangle(x, y, 2 * RADIUS, 2 * RADIUS);
        }}
```

소스코드 깃헙 : https://han.gl/fMlrg

3. 소스코드

- GameBoard.java

```
package pong;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.*;


import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;

public class GameBoard extends JPanel implements KeyListener {
                Ball ball1;
                Ball ball2;
                Ball ball3;
                Racquet racquet1;
                Racquet racquet2;

                boolean gamefail = false;
```

```
public GameBoard() {
                ball1 = new Ball(this, Color.red);
                ball1.x = 200; ball1.y = 300;
                ball2 = new Ball(this, Color.blue);
                ball2.x = 250; ball2.y = 50;
                ball3 = new Ball(this, Color.yellow);
                ball3.x = 100; ball3.y = 100;
                this.setBackground(Color.green);
                racquet1 = new Racquet(this, 10, 150, Color.blue);
                racquet2 = new Racquet(this, 560, 150, Color.yellow);
                setFocusable(true);
                addKeyListener(this);
                }

public void deleteballs() {
                ball1.color = Color.pink;
                ball2.color = Color.pink;
                ball3.color = Color.pink;
                }
```

소스코드 깃헙 : https://han.gl/fMlrg

3. 소스코드

- GameBoard.java

```
@Override
public void keyTyped(KeyEvent e) {
}

@Override
public void keyReleased(KeyEvent e) {
                racquet1.keyReleased(e);
                racquet2.keyReleased(e);
}

@Override
public void keyPressed(KeyEvent e) {
                racquet1.keyPressed(e);
                racquet2.keyPressed(e);
}

private void move() {
                ball1.move();
                ball2.move();
                ball3.move();
                racquet1.move();
                racquet2.move();
                if (ball1.hitwall || ball2.hitwall || ball3.hitwall) {
                                gamefail = true;
                }
}
```

소스코드 깃헙 : https://han.gl/fMlrg

3. 소스코드

- GameBoard.java

```
@Override
        public void paint(Graphics g) {
                super.paint(g);
                Graphics2D g2d = (Graphics2D) g;
                ball1.draw(g2d);
                ball2.draw(g2d);
                ball3.draw(g2d);
                racquet1.draw(g2d);
                racquet2.draw(g2d);
        }

        public static void main(String[] args) {

                JFrame frame = new JFrame("Pong 게임");
                JLabel labels;

                frame.setSize(600, 400);
                frame.setVisible(true);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                labels = new JLabel("Fail");
                labels.setForeground(Color.black);
                labels.setFont(new Font("Serif", Font.BOLD, 50));
                labels.setLocation(250,70);
                labels.setSize(200,200);
```

```
GameBoard game = new GameBoard();
frame.add(game);


while (true) {
                game.move();
                game.repaint();
                if (game.gamefail) {
                        game.add(labels);
                        break;
                }
                try {
                        Thread.sleep(10);
                } catch (InterruptedException e) {
                        e.printStackTrace();
                }
}}}
```

소스코드 깃헙 : https://han.gl/fMlrg

3. 소스코드

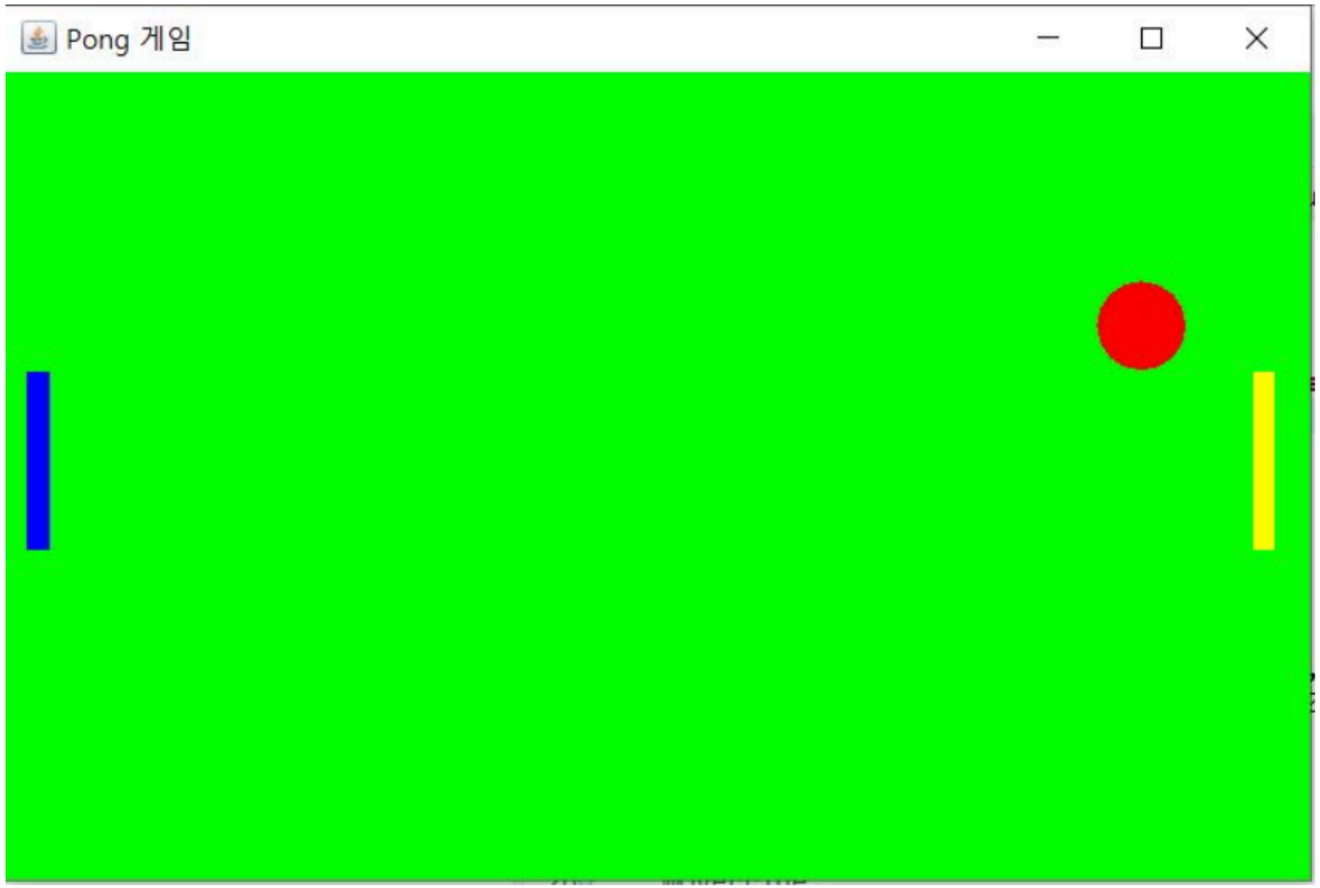
- Recquect.java

```
package pong;

import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.event.KeyEvent;

public class Racquet {
		private static final int WIDTH = 10;
		private static final int HEIGHT = 80;
		int x = 0;
		int y = 0;
		Color color;
		int xspeed = 0;
		int yspeed = 0;
		private GameBoard game;

		public Racquet(GameBoard game, int x, int y, Color color) {
			this.game = game;
			this.x = x;
			this.y = y;
			this.color = color;
		}
		public void move() {
			if (y + yspeed > 0 && y + yspeed <
game.getHeight() - HEIGHT)
				y = y + yspeed;
		}

		public void draw(Graphics2D g) {
			g.setColor(color);
			g.fillRect(x, y, WIDTH, HEIGHT);
		}

		public void keyReleased(KeyEvent e) {
			yspeed = 0;
		}

		public void keyPressed(KeyEvent e) {
			if (e.getKeyCode() == KeyEvent.VK_UP)
				yspeed = -10;
			if (e.getKeyCode() == KeyEvent.VK_DOWN)
				yspeed = 10;
		}

		public Rectangle getBounds() {
			return new Rectangle(x, y, WIDTH, HEIGHT);
		}
}
```

소스코드 깃헙 : https://han.gl/fMlrg

4. 실행 결과

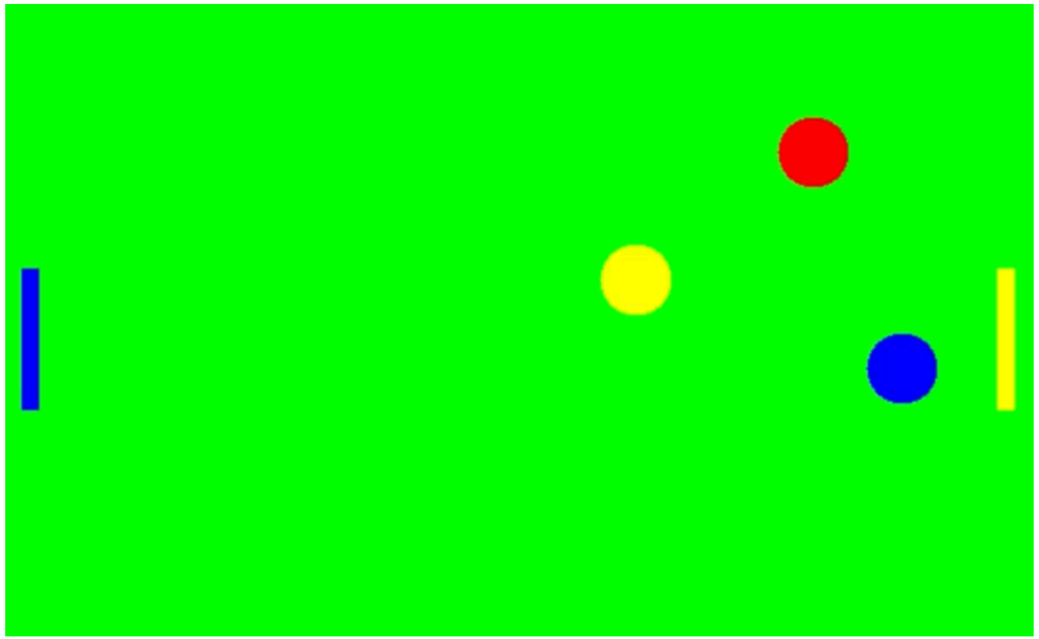
- 이전 프로그램 실행 결과

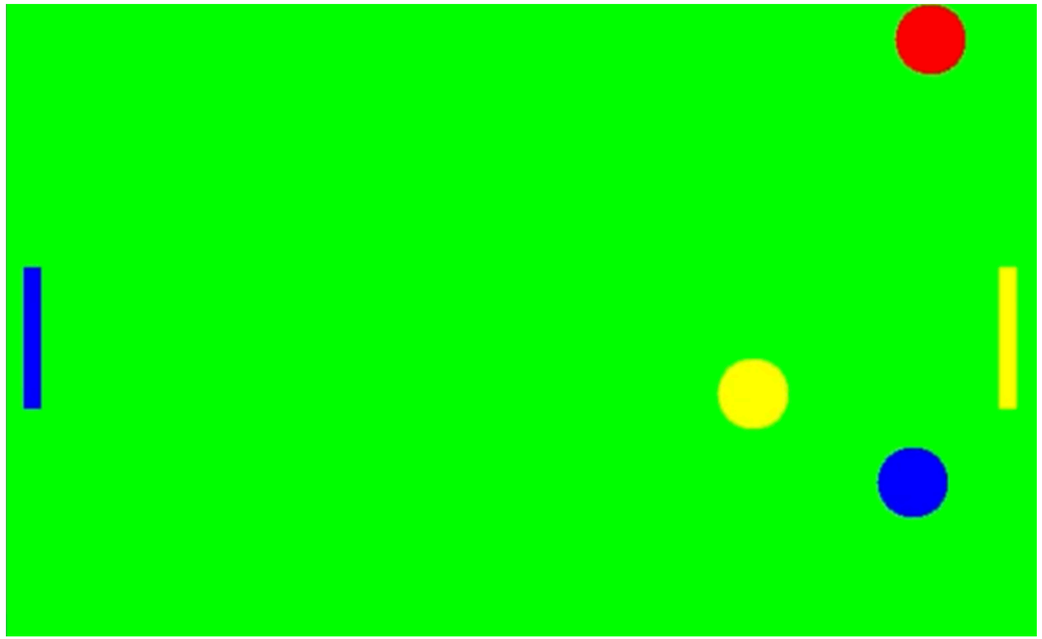


- 공이 대각선 방향으로 이동
- 사이드 바에 부딪히면 반대 방향으로 전환
- 부딪히지 않는 경우 (벽에 닿은 경우)
  - 별다른 변화 없이 반대로 방향 전환 후 이동

4. 실행 결과

- 최종 프로그램 결과 (시연 링크 : https://youtu.be/vhkUzsVuFpU)

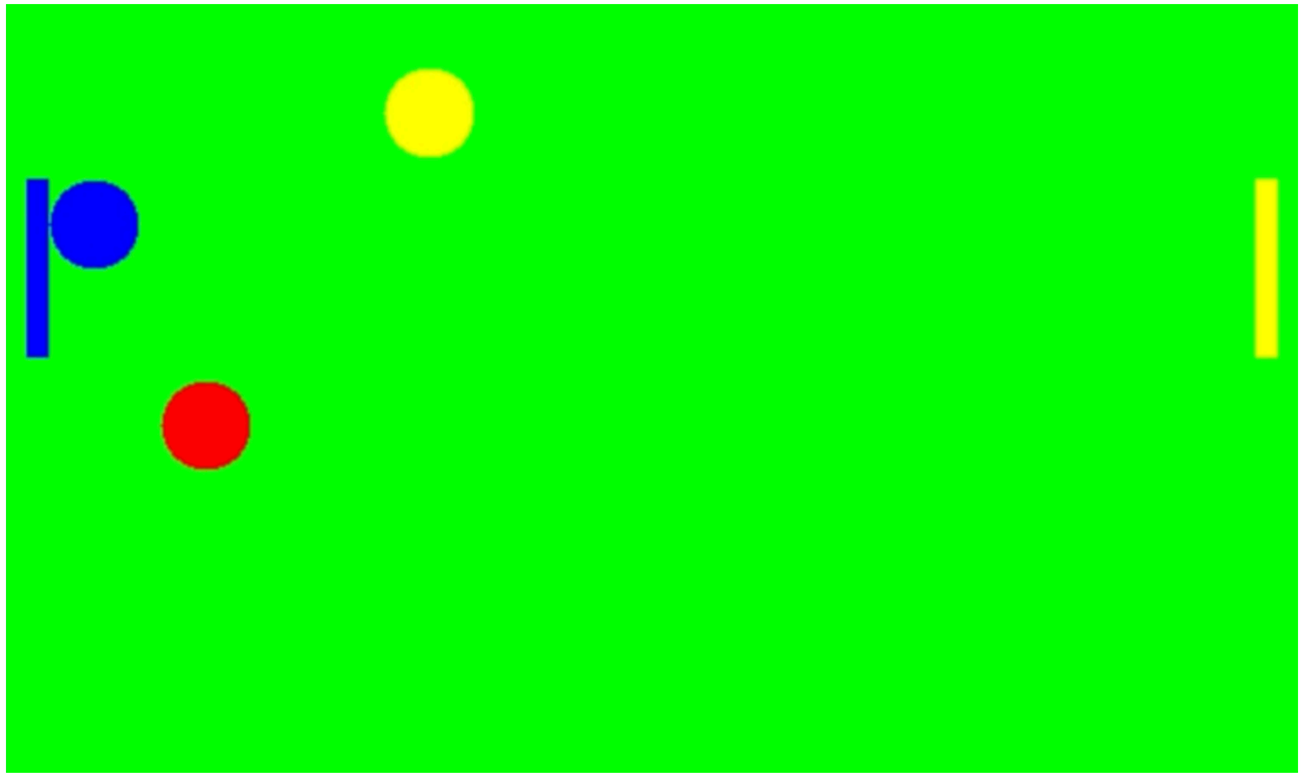




(Case 1. 공이 돌아다니는 경우, 상/하 벽에 부딪히는 경우)

4. 실행 결과

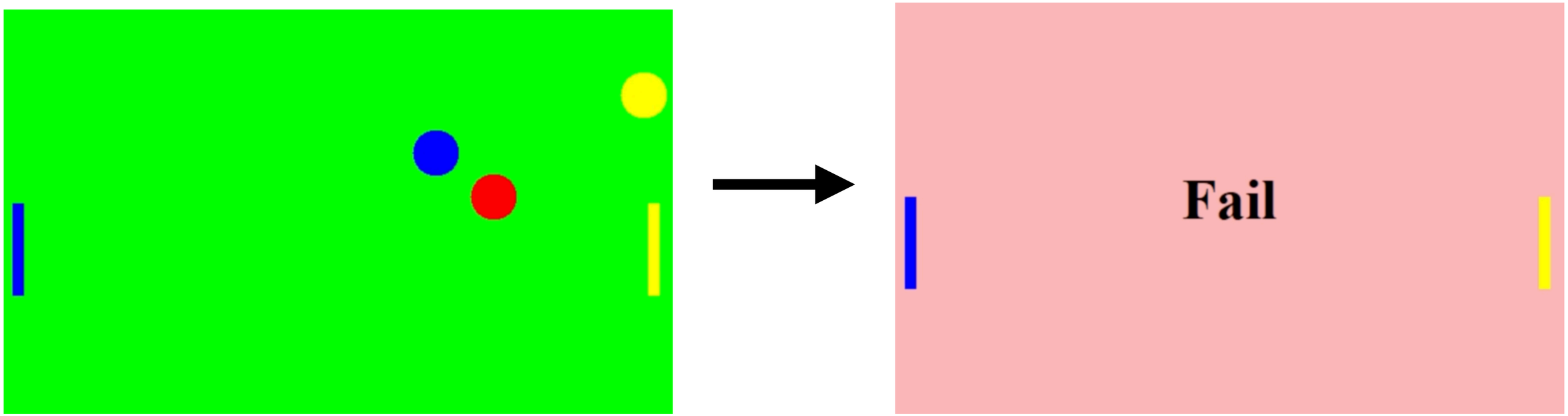
- 최종 프로그램 결과 (시연 링크 : https://youtu.be/vhkUzsVuFpU)



(Case 2. 공이 바에 닿는 경우)

4. 실행 결과

- 최종 프로그램 결과 (시연 링크 : https://youtu.be/vhkUzsVuFpU)



(Case 3. 공이 좌/우 벽에 닿는 경우, Fail로 화면 전환)

5. 완성도

계획 대비 프로그램 완성도

- 공의 개수 늘리기 - 현재 Ball의 개수를 3개로 늘리기 난이도 (하) ✓
- 공의 색깔과 모양 ~~다양하게 변경하기~~ 난이도 (중) ▲
- 공이 좌,우 양쪽 바가 아닌 벽에 닿으면 “Fail” 창 뜨도록 변경하기 난이도 (상) ✓
- 공이 좌,우 양쪽 바가 아닌 벽에 닿으면 배경 색깔 다르게 바뀌도록 변경하기 난이도 (상) ✓

✓ 총 88% 완성 (공들의 모양 변경까지는 구현하지 못하였음)

6. 후기 및 소감

- 처음 원 객체를 2개 더 추가하는 과정에서, 아무리 설정을 해보아도 **한 개 밖에** 보이지 않았습니다. class도 만들어보고, 함수도 다시 짜보았는데도 되지 않아서 초반에 시간을 많이 사용했던 것 같습니다.

  -> 원의 첫 좌표 위치를 다르게 해보니 각각의 원들이 보임을 알 수 있었습니다.
  좌표 설정이 겹쳐서 원이 여러개가 아닌 하나로 보였다는 것을 알 수 있었습니다.

6. 후기 및 소감

- 원이 벽에 닿을 경우, background의 색깔이 변경되도록 설정하는 코드를 짤 때, 어떤식으로 if문을 만들어야 할 지 고민을 했었습니다. if bar에 닿지 않는다면 색깔이 변경되도록 코딩을 하였더니, bar가 아닌 모든 범위일 경우에 색깔이 변경됨을 확인할 수 있었습니다.

  -> 따라서 **|| or 조건**을 사용하여 bar가 아니면서 y좌표 (벽)의 위치일 경우 색이 변경되도록 짜는 부분을 통해서 문제 해결 사고가 코딩할 때 중요하다는 것을 느낄 수 있었습니다.

6. 후기 및 소감

- 기존의 프로그래밍을 파악하는 과정도 어렵기는 하였지만, 하나씩 목표를 잡고 해나갈 때마다 뿌듯함을 많이 느낄 수 있었던 프로젝트여서 너무 의미가 있었습니다 :)
- 특히나 java가 단순히 언어가 아닌 인터페이스를 구현하는데 큰 장점이 되는 프로그래밍인 것을 깨달을 수 있었던 겨울 계절학기 수업이였습니다!

한 학기 수고하셨습니다.
감사합니다 :)

END