

TWIN

Ball

객체지향 프로그래밍
1862007 김보람
1660013 김성수

INDEX



작품소개

개발과정 및 역할분담



프로그램 구조

프로그램 내용





작품소개



개발과정 및 역할분담



프로그램 구조



프로그램 내용



Twin Ball이란?



“ 혼자가 아닌
둘이 즐기는 게임!! ”

- 회원가입 및 로그인 을 통해 정보저장
- 컴퓨터와 게임할수있는 1p 연습모드
- 친구와 함께 즐기는 2p모드
- 목표물을 맞춰 점수를 가져가는 게임
- 랭킹 확인을 통해 흥미유발



작품소개



개발과정 및 역할분담



프로그램 구조



프로그램 내용

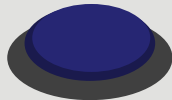




개발 과정1

19-10-30 ~ 19-11-02

개발프로그램에 대한
브레인스토밍 및 마이스
브레이킹



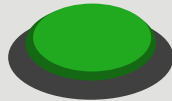
19-11-04~19-11-05

아이디어 구상 및
계획서 초안 작성



19-11-07~19-11-10

초기화면 및 프로그램
기본적인 GUI 구성



19-11-11~19-11-20

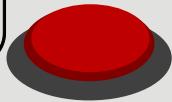
로그인,회원가입 기능 ,
회원정보 수정기능 구현
회원 데이터베이스 연동



개발 과정2

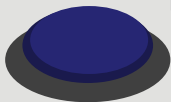
19-11-22 ~ 19-12-13

게임 화면 구성 및 기능
구현, 게임점수 및 랭킹
데이터 베이스 연동



19-12-14 ~ 19-12-16

회원정보 수정창 및
게임 결과창 작업



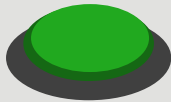
19-12-17 ~ 19-12-18

오류 수정 작업 및
보충 작업



19-12-19 ~ 19-12-20

마무리 작업 및 발표



역할 분담

1660013 김성수

전반적인 GUI 이미지 구성

회원정보, 게임정보 테이블
설계 및 제작

회원정보 DB 관리 기능

점수 및 top 랭킹 조회 기능

1862007 김보람

게임 GUI 구성

1P 모드, 2P 모드 제작
및 구현

timer, jump, game
thread 제작

게임 DB 관리 기능



작품소개



개발과정 및 역할분담



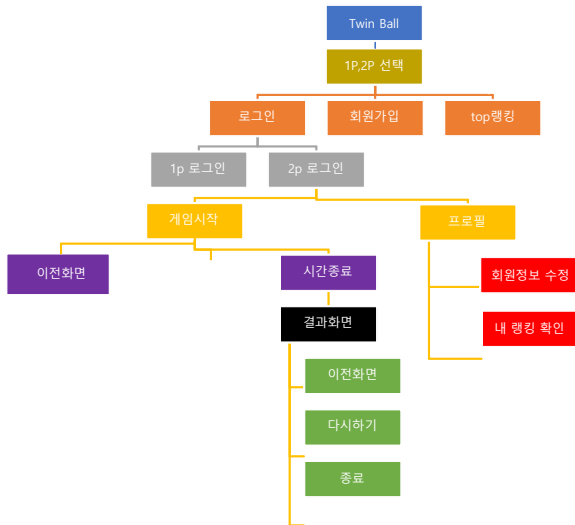
프로그램 구조

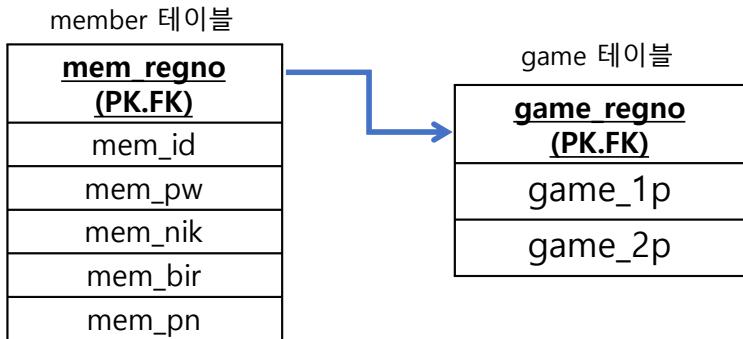


프로그램 내용



알고리즘





- DBMS : PostgreSQL 사용

- mem_regno : 성별 + 주민번호 13자리(ex) W9904202222222)
- mem_id : 회원 ID
- mem_pw : 회원 PW
- mem_nik : 회원 nickname
- mem_pn : 회원 휴대폰 번호
- game_regno : mem_regno 값
- game_1p : 1p모드 점수
- game_2p : 2p모드 점수

Game & member

- game
 - Ball1_2p.java
 - Ball1.java
 - Ball2_2p.java
 - Ball2.java
 - DrawingSurface_1p.java
 - DrawingSurface_2p.java
 - Goal.java
 - jumsuth.java
 - Main_1p.java
 - Main_2p_key.java
 - NorthPanel.java
 - NorthPanel2.java
 - Racket_com.java
 - Racket_key1.java
 - Racket_key2.java
 - Racket_mouse.java
 - result1p_screen.java
 - result2p_screen.java
 - Select_Screen.java
 - Select_Screen2.java
 - timerth.java
 - timerth2.java

- member
 - changeDAO.java
 - first_screen.java
 - joinDAO.java
 - joinmain.java
 - login_1p.java
 - login_1p
 - login_2p.java
 - loginDAO.java
 - Member.java
 - MyPage.java
 - nikPanel.java
 - pwPanel.java
 - Rank.java
 - rankDAO.java
 - rankDAO
 - rankPanel.java

- IDE : Java Eclipse 사용



작품소개



개발과정 및 역할분담



프로그램 구조



프로그램 내용



프로그램 초기 화면



- 1P 버튼 : 혼자 게임을 하고싶을 때
- 2P 버튼 : 누군가와 함께 하고싶을 때

회원가입 화면(1)

```
// 아이디 중복확인 눌렀을 때
else if (cmd == idOk) {

    dao.IDcheck(id);
    try {
        if (dao.rs.next()) {
            System.out.println("중복");
            la = "아이디가 있습니다!";
            JOptionPane.showMessageDialog(null, la, "ID 중복 확인", JOptionPane.ERROR_MESSAGE);
        } else {
            System.out.println("중복아님");
            la = "사용 가능한 아이디입니다!";
            sw1 = true;
            JOptionPane.showMessageDialog(null, la);
        }
    } catch (SQLException e1) {
        e1.printStackTrace();
    }
}
```



```
//닉네임 중복 확인
if (cmd == nikOk) {
    dao.NIKcheck(nik);
    try {
        if (dao.rs.next()) {
            System.out.println("중복");
            la = "닉네임이 있습니다!";
            JOptionPane.showMessageDialog(null, la, "닉네임 중복 확인", JOptionPane.ERROR_MESSAGE);
        } else {
            System.out.println("중복아님");
            sw3 = true;
            la = "사용 가능한 닉네임 입니다!";
            JOptionPane.showMessageDialog(null, la);
        }
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    dao.NIKcheck(nik);
}

if (sw1 == true && sw2 == true && sw3 == true) {
    //아이디중복확인과 닉네임중복확인을 했을 때 가입하기 활성화 된
    btnOk.setEnabled(true);
}
```



```
//PW 일치 확인
else if (cmd == pwOk) {
    if (!pw.equals(rpww))
    {
        JOptionPane.showMessageDialog(null, "pw를 다시 확인하세요", "PW 불일치", JOptionPane.ERROR_MESSAGE);
    }
    else
    {
        JOptionPane.showMessageDialog(null, "PW 일치");
        sw2=true;
    }
}
```



-입력받은 값 DB에서 검색 후 대조 => 3가지 모두 중복체크 후 가입하기 버튼 활성화
존재할 경우 경고창 생성

회원가입 화면(2)

```
// 가입 눌렀을때
if (cmd == btnOk) {

    member = new Member();
    member.setId(id);
    member.setPw(pw);
    member.setnik(nik);
    member.setbir(birthday);
    member.setPhone(phone);
    member.setRegno(regno);
    System.out.println(id + "\n" + pw + "\n" + nik + "\n" + birthday + "\n" + phone + "\n" + regno);
```

```
if (dao.newMember(member)) {
    nl.setText(nik + "님 회원가입을 축하드려요!!");
    JOptionPane.showMessageDialog(null, "회원가입완료");
    dispose();
} else {
    nl.setText("회원가입 실패");
}
```

```
if (sw1 == true && sw2 == true && sw3 == true) {
    // 아이디중복확인과 닉네임중복확인을 했을 때 가입하기 활성화 됨
    btnOk.setEnabled(true);
}
```

The '회원가입' window has a yellow background. It contains the following fields and buttons:

- HELLO
- 회원 가입
- *아이디: [text box] 중복 확인
- *비밀번호: [text box] PW 확인
- 비밀번호확인: [text box]
- *닉네임: [text box] 중복 확인
- 생년월일: [year] 년 [month] 월 [day] 일
- 핸드폰: [area code] - [number] - [number]
- 주민등록번호: [text box] - [text box]
- 가입하기 (disabled) 가입취소



The '회원가입' window is in its final state with the following values:

- HELLO
- 회원 가입
- *아이디: AAA111 중복 확인
- *비밀번호: **** PW 확인
- 비밀번호확인: ****
- *닉네임: 김두 중복 확인
- 생년월일: 1997 년 07 월 28 일
- 핸드폰: 010 - 0000 - 0000
- 주민등록번호: 123456 - 000000
- 가입하기 (active) 가입취소

- 아이디 중복확인과 닉네임 중복확인이 완료됐을때 가입하기 버튼 활성화

The '메시지' dialog box has a title bar with '메시지' and a close button. It contains an information icon, the text '회원가입완료' (Sign Up Complete), and a '확인' (Confirm) button.

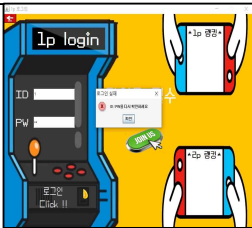
1P 로그인화면

```
JButton logbtn = new JButton(loginbtn);
logbtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        id1 = idtf.getText();
        pw1 = "";
        char[] getPw = pwtf.getPassword();
        for (int i = 0; i < getPw.length; i++) {
            pw1 += getPw[i];
        }
```

```
//System.out.print("click");
System.out.println(id1);
dao.IDPW1(id1);
```

```
System.out.println(dao.IDPW1(id1).getPw());
```

```
if(pw1.equals(dao.IDPW1(id1).getPw()))
{
    JOptionPane.showMessageDialog(null, "로그인성공");
    dispose();
    new Select_Screen();
}
else {
    JOptionPane.showMessageDialog(null, "ID / PW를 다시 확인하세요", "로그인 실패", JOptionPane.ERROR_MESSAGE);
}
```



```
String sql = "SELECT * from member WHERE mem_id = '" + id + "'";
try {
    pstmt = conn.prepareStatement(sql);
    rs = pstmt.executeQuery();
    if(rs.next()) {
        mem.setId(rs.getString("mem_id"));
        System.out.println(1);
        mem.setPw(rs.getString("mem_pw"));
        System.out.println(2);
        mem.setnik(rs.getString("mem_nik"));
        System.out.println(3);
        mem.setbir(rs.getInt("mem_bir"));
        System.out.println(4);
        mem.setPhone(rs.getInt("mem_pn"));
        System.out.println(5);
        mem.setPhone(rs.getInt("mem_regno"));
        System.out.println(6);

        System.out.println(mem.getId()+"!");
        System.out.println(mem.getPw()+"!");
        System.out.println(mem.getnik()+"!");
        System.out.println(mem.getbir()+"!");
        System.out.println(mem.getPhone()+"!");
        System.out.println(mem.getRegno()+"!");
    }
    else {
        System.out.println("아이디없음");
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```

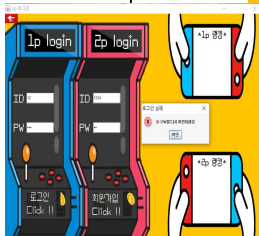
- 입력받은 ID를 DB에서 검색 => 그 ID의 PW와 입력받은 PW를 대조 => 로그인성공
- 다이얼로그를 이용해 정보가 일치하지 않을경우 경고창 생성

2P 로그인화면

```
dao.IDPW1(id1);
dao.IDPW2(id2);
if(id1.equals(id2))
{
JOptionPane.showMessageDialog(null,"다른 ID를 입력하세요", "로그인 실패",JOptionPane.ERROR_MESSAGE);
}
else
{
if(pw1.equals(dao.IDPW1(id1).getPw())&&pw2.equals(dao.IDPW2(id2).getPw()))//둘 다 로그인 성공
{
JOptionPane.showMessageDialog(null, "로그인성공");
dispose();
new Select_Screen();//일단
}
else if(pw1.equals(dao.IDPW1(id1).getPw()) && !pw2.equals(dao.IDPW2(id2).getPw()))//1p만성공
{
JOptionPane.showMessageDialog(null,"2p의 정보를 확인하세요", "로그인 실패",JOptionPane.ERROR_MESSAGE);
}
else if(!pw1.equals(dao.IDPW1(id1).getPw()) && pw2.equals(dao.IDPW2(id2).getPw()))//2p만성공
{
JOptionPane.showMessageDialog(null,"1p의 정보를 확인하세요", "로그인 실패",JOptionPane.ERROR_MESSAGE);
}
else
{
JOptionPane.showMessageDialog(null,"ID / PW를 다시 확인하세요", "로그인 실패",JOptionPane.ERROR_MESSAGE);
}
```

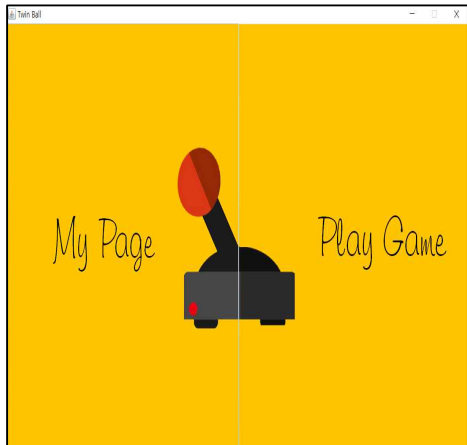
```
String sql = "SELECT * from member WHERE mem_id = '"+id+"'";
try {
    pstmt = conn.prepareStatement(sql);
    rs = pstmt.executeQuery();
    if(rs.next()) {
        mem.setId(rs.getString("mem_id"));
        System.out.println(1);
        mem.setPw(rs.getString("mem_pw"));
        System.out.println(2);
        mem.setnik(rs.getString("mem_nik"));
        System.out.println(3);
        mem.setbir(rs.getInt("mem_bir"));
        System.out.println(4);
        mem.setPhone(rs.getInt("mem_pn"));
        System.out.println(5);
        mem.setPhone(rs.getInt("mem_regno"));
        System.out.println(6);

        System.out.println(mem.getId()+"!");
        System.out.println(mem.getPw()+"!");
        System.out.println(mem.getnik()+"!");
        System.out.println(mem.getbir()+"!");
        System.out.println(mem.getPhone()+"!");
        System.out.println(mem.getRegno()+"!");
    }
    else {
        System.out.println("아이디없음");
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```



- 1P,2P 두가지 입력 => 입력받은 ID를 DB에서 검색 => 그 ID의 PW와 입력받은 PW를 1P, 2P 둘다 대조 => 로그인성공

선택 화면



- MY PAGE : 비밀번호 수정이 가능
- PLAYGAME: 친구와 함께 하고싶을 때

MY PAGE 화면

```
okbtn = new JButton("확인");
okbtn.setFont(font);
okbtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        login_ip player1 = new login_ip();
        player1.dispose();
        String pw1 = "";
        char[] getPw = nowpw.getPassword();
        for (int i = 0; i < getPw.length; i++) {
            pw1 += getPw[i];
        }

        if (pw1.equals(dao.IDPW1(player1.id1).getPw())) {
            JOptionPane.showMessageDialog(null, "올바른 비밀번호입니다.");
            sw1 = true;
            regno = dao.IDPW1(player1.id1).getRegno();
            System.out.println("regno : " + regno);
            if (sw1 == true && sw2 == true) {
                // 아이디등록확인과 비밀번호등록확인을 했을 때 변경하기가 활성화됨
                changebtn.setEnabled(true);
            }
        } else {
            JOptionPane.showMessageDialog(null, "PW를 다시 확인하세요", "실패", JOptionPane.ERROR_MESSAGE);
        }
    }
});
```

```
eqbtn = new JButton("확인");
eqbtn.setFont(font);
eqbtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        String new1 = newpw1.getText();
        String new2 = newpw2.getText();
        if (new1.equals(new2)) {
            JOptionPane.showMessageDialog(null, "비밀번호가 일치합니다.");
            sw2 = true;
            realpw = new1;

            if (sw1 == true && sw2 == true) {
                // 아이디등록확인과 비밀번호등록확인을 했을 때 변경하기가 활성화됨
                changebtn.setEnabled(true);
            }
        } else {
            JOptionPane.showMessageDialog(null, "PW를 다시 확인하세요", "실패", JOptionPane.ERROR_MESSAGE);
        }
    }
});
eqbtn.setBounds(290, 250, 180, 30);
add(eqbtn);

changebtn = new JButton("변경");
changebtn.setFont(font);
changebtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        // 비밀번호 정보변경
        if (Cdao.PwChange(realpw, regno)) {
            JOptionPane.showMessageDialog(null, "PW 변경 성공");
        } else {
            JOptionPane.showMessageDialog(null, "PW 변경에 실패하였습니다.", "PW 변경 실패", JOptionPane.ERROR_MESSAGE);
        }
    }
});
```



- 현재 로그인한 비밀번호를 입력시에 일치하면 확인성공
- 바꿀 비밀번호입력 -> 한번더 입력 -> 확인
- 두번다 일치하면 변경버튼 활성화

MY PAGE 화면

```
okbtn = new JButton("확인");
okbtn.setFont(font);
okbtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        login_ip player1 = new login_ip();
        player1.dispose();

        String pw1 = "";
        char[] getPw = ncmpw.getPassword();
        for (int i = 0; i < getPw.length; i++) {
            pw1 += getPw[i];
        }

        if (pw1.equals(dao.IDPW1(player1.id1).getPw()))
        {
            JOptionPane.showMessageDialog(null, "올바른 비밀번호입니다");

            regno = dao.IDPW1(player1.id1).getRegno();

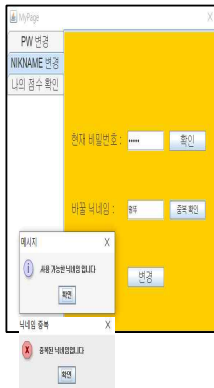
            sw1=true;

            if(sw1==true && sw2==true) {
                //비밀번호를 확인하고 중복을 확인하여 비밀번호 변경 가능
                changebtn.setEnabled(true);
            }
        }
        else
        {
            JOptionPane.showMessageDialog(null, "PW를 다시 확인하세요", "실패", JOptionPane.ERROR_MESSAGE);
        }
    }
});
```

```
System.out.println("종료");
JOptionPane.showMessageDialog(null, "종료된 비밀번호입니다", "비밀번호 종료", JOptionPane.ERROR_MESSAGE);
} else {
    System.out.println("종료하는");
    JOptionPane.showMessageDialog(null, "사용 가능한 비밀번호입니다");

    sw2=true;
    if(sw1==true && sw2==true) {
        //비밀번호를 확인하고 중복을 확인하여 비밀번호 변경 가능
        changebtn.setEnabled(true);
    }
} catch (SQLException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
});
eqbtn.setBounds(290, 250, 100, 30);
add(eqbtn);

changebtn = new JButton("변경");
changebtn.setFont(font);
changebtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        // 비밀번호 정보변경
        if (CHdao.NIKchange(realnik, regno)) {
            JOptionPane.showMessageDialog(null, "비밀번호 변경 성공");
        } else {
            JOptionPane.showMessageDialog(null, "비밀번호 변경에 실패하였습니다", "비밀번호 변경 실패", JOptionPane.ERROR_MESSAGE);
        }
    }
});
```



- 현재 로그인한 비밀번호를 입력시에 일치하면 확인성공
- 바꿀 닉네임 입력 -> 입력한 닉네임을 DB의 저장된값과 대조 -> 일치하는값이 없을시 중복확인 성공
- 두번다 일치하면 변경버튼 활성화

MY PAGE 랭킹화면

```
//1p점수 조회 메소드
public int jumsu1LOOK(String regno) {
    connectDB();

    String sql = "SELECT game_1p FROM game WHERE game_regno = '"+regno+"'";
    try {
        pstmt = conn.prepareStatement(sql);

        rs = pstmt.executeQuery();
        rs.next();

        jumsu1= rs.getInt("game_1p");

        System.out.println("regno : "+ regno);
        System.out.println("jumsu1 : "+ jumsu1);

    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
    }
}
```

```
//2p점수 조회 메소드
public int jumsu2LOOK(String regno) {
    connectDB();

    String sql = "SELECT game_2p FROM game WHERE game_regno = '"+regno+"'";
    try {
        pstmt = conn.prepareStatement(sql);

        rs = pstmt.executeQuery();
        rs.next();

        jumsu2= rs.getInt("game_2p");

        System.out.println("regno : "+ regno);
        System.out.println("jumsu2 : "+ jumsu2);

    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
    }
}
```

MyPage

PW 변경

NICKNAME 변경

나의 점수 확인

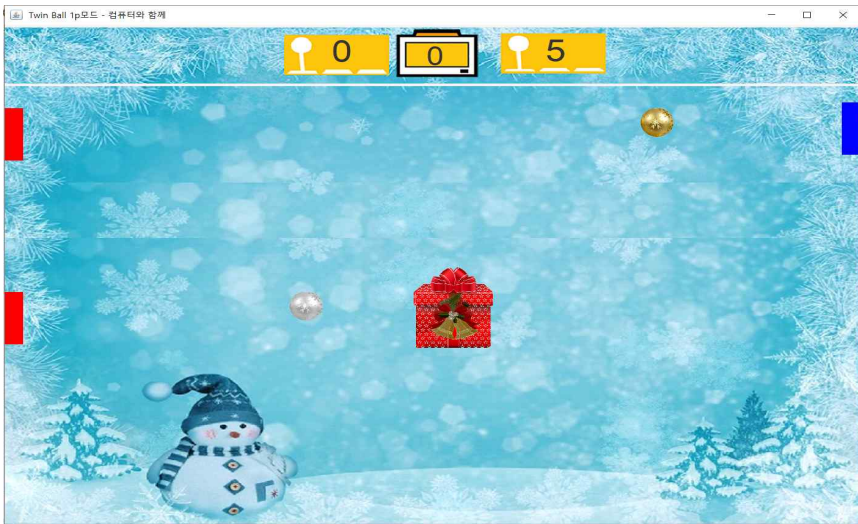
나의 1p 최고점수: 0

나의 2p 최고점수: 0

조회

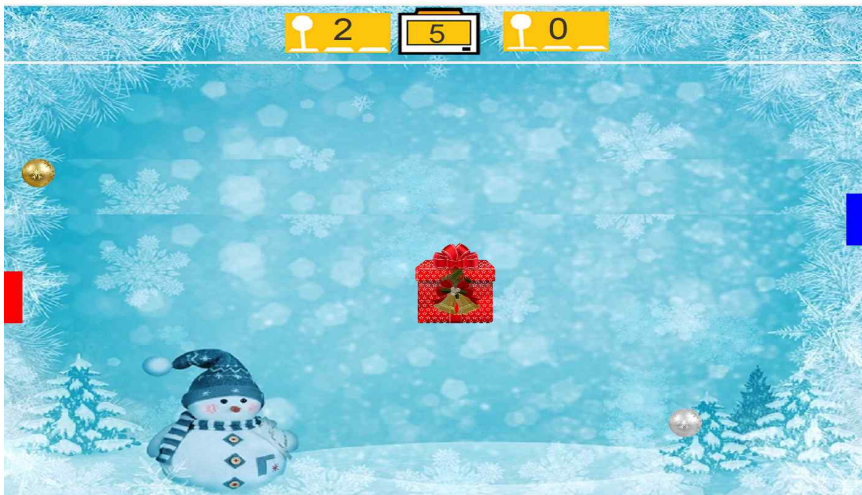
- SELECT 명령문을 사용해 DB에 있는 GAME 테이블의 1P 값을 불러온다
- SELECT 명령문을 사용해 DB에 있는 GAME 테이블의 2P 값을 불러온다

게임화면 1P모드



게임화면 2P모드

Twin Ball 2p모드 - 친구와 함께



로그인화면에서 랭킹조회

```
//2p로드 T03랭킹 불러오는 메소드
public ArrayList<Rank> get2p() {
    connectDB();

    sql = "SELECT member.mem_nik ,\r\n" +
        " member.mem_id ,\r\n" +
        " game.game_2p\r\n" +
        "FROM member, game \r\n" +
        "WHERE member.mem_regno = game.game_regno\r\n" +
        "ORDER BY game.game_2p DESC;";

    //현재 검색 데이터를 전달하기 위한 ArrayList
    ArrayList<Rank> datas = new ArrayList<Rank>();
    // 관리자도 정보박스 데이터를 위한 벡터 초기화
    Vector<Rank> rank = new Vector<Rank>();
    try {
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery(); // 검색된 데이터수 만큼 루프를 돌려 Product 객체를 만들고 이를 다시 ArrayList 에 추가함.
        for(int i=0; i<3;i++) //top3의 랭킹만 조회
        {
            rs.next();
            Rank r = new Rank();

            r.setnik(rs.getString("mem_nik"));
            r.setId(rs.getString("mem_id"));
            r.setjumsu2(rs.getInt("game_2p"));
            datas.add(r);
        }
    } catch (SQLException e) {
        e.printStackTrace();
        return null;
    } finally {
```

```
//1p로드 T03랭킹 불러오는 메소드
public ArrayList<Rank> get1p() {
    connectDB();

    sql = "SELECT member.mem_nik ,\r\n" +
        " member.mem_id ,\r\n" +
        " game.game_1p\r\n" +
        "FROM member, game \r\n" +
        "WHERE member.mem_regno = game.game_regno\r\n" +
        "ORDER BY game.game_1p DESC;";

    //현재 검색 데이터를 전달하기 위한 ArrayList
    ArrayList<Rank> datas = new ArrayList<Rank>();
    // 관리자도 정보박스 데이터를 위한 벡터 초기화
    Vector<Rank> rank = new Vector<Rank>();

    try {
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery(); // 검색된 데이터수 만큼 루프를 돌려 Product 객체를 만들고 이를 다시 ArrayList 에 추가함.
        for(int i=0; i<3;i++) //top3의 랭킹만 조회
        {
            rs.next();
            Rank r = new Rank();

            System.out.println("mem_nik:" + rs.getString("mem_nik"));
            System.out.println(rs.getString("mem_id"));
            System.out.println(rs.getString("game_1p"));

            r.setnik(rs.getString("mem_nik"));
            r.setId(rs.getString("mem_id"));
            r.setjumsu1(rs.getInt("game_1p"));

            datas.add(r);
```

- 테이블 조인을 이용하여 member테이블의 ID, 닉네임을 불러오고 game테이블의 점수를 조회한다

게임 DB 관리

//1p최고점수 업데이트 메소드

```
public void save(String regno, int jumsu1) {
    connectDB();
    sql = "UPDATE game SET game_1p = "+jumsu1+" WHERE game_regno = '"+regno+"'AND game_1p<"+jumsu1+"";

    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        closeDB();
    }
    System.out.println("변경성공");
}
```

//2p최고점수 업데이트 메소드

```
public void save2(String regno, int jumsu2) {
    connectDB();
    sql = "UPDATE game SET game_2p = "+jumsu2+" WHERE game_regno = '"+regno+"'AND game_1p<"+jumsu2+"";

    try {
        System.out.println(jumsu1);
        pstmt = conn.prepareStatement(sql);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        closeDB();
    }
    System.out.println("변경성공");
}
```

- 결과화면에서 점수를 game테이블에 update를 한다

게임화면 2P모드



1p와 2p의 점수를 출력해주고 결과를 출력한다
결과는 승,패,무승부로 나오며 유저1 승리시 옆에 승이라고 나오며
유저1이 패할시 옆에 패라고 나온다. 유저2 또한 같다.



감사합니다