

1. 미니프로젝트 주제

미니프로젝트는 자바 언어로 Swing GUI 와 스레드를 활용하는 자바 응용프로그램을 만든다. “움직이는 단어 타이핑 게임”으로 단어가 떨어지며 단어를 맞춰 점수를 올리는 타자게임을 만든다.

2. 작품개요

응용프로그램의 시나리오에 대해서 설명하겠다. “한컴타자게임”에서 “산성비”라는 게임을 모방하여 만들 것이다. 한성대의 마스코트 상상부기가 단어를 모두 맞추어 10단계를 모두 성공하여 한성대학교에서 살아남기 위한 게임이다. 배경에 상상부기 캐릭터가 있다.

위에서 단어가 떨어지고, 사용자는 그 단어나 문장을 맞춰 글자를 지워나가면서 점수를 얻는다. 레벨은 10단계 까지있고, 레벨이 높아질수록 속도가 빨라지도록 한다. 레벨1은 초당 10픽셀로 떨어지고 레벨이 오를수록 초당 5 픽셀이 빨라지고, 레벨 5부터는 초당 2~3 픽셀을 추가하여 떨어지게 한다.

또한 8단계 부터는 문장이 나타나게 하여 난이도를 높인다. 점수는 하나를 맞출 경우 10점씩 오르도록 한다. 틀릴 경우에는 점수가 깎이지 않게 한다. 만약 단어를 바닥까지 타이핑하지 못하고, 떨어진다면 상상부기의 목숨이 깎인다. 목숨값은 10개가 있다. 목숨은 이미지로 표시하여 하나씩 없어지는 것을 보이게 한다.

\* 단계별 특이점 (총 10단계로 진행된다.)

- 1단계 : 1초당 10픽셀의 속도로 떨어짐.
- 2단계 : 1초당 15픽셀의 속도로 떨어짐.
- 3단계 : 1초당 20픽셀의 속도로 떨어짐.
- 4단계 : 1초당 25픽셀의 속도로 떨어짐.
- 5단계 : 1초당 30픽셀의 속도로 떨어짐.
- 6단계 : 1초당 33픽셀의 속도로 떨어짐.
- 7단계 : 1초당 35픽셀의 속도로 떨어짐.
- 8단계 : 1초당 37픽셀의 속도로 떨어짐.
- 9단계 : 1초당 40픽셀의 속도로 떨어짐.
- 10단계 : 1초당 45픽셀의 속도로 떨어짐.

랜덤하게 빨강색, 파란색, 핑크색, 검정색 단어가 나오게한다. 각 단어마다 특별한 기능이 있다. 기능이 있는 단어는 20프로의 확률로 나오며 3개 단어는 똑같은 확률로 나오게 만들었다.

\* 단어별 기능

- 검정색단어 : 일반 단어 - 점수 +10
- 파란색단어 : 모든 단어를 3초간 모두 정지시키는 기능을 가짐 - 점수 +20
- 핑크색단어 : 모든 단어를 삭제시킨 - 점수 +50
- 빨간색단어 : 단어가 떨어지는 속도가 5초간 빨라진다. - 점수 +10



\* 한컴타자게임의 산성비

화면은 이와 비슷하게 만든다. 단어를 추가하는 창은 같은 UI에 만들지 않고, 메뉴창에서 새롭게 단어를 추가하고자 하는 단어가 있는 경우 들어가서 단어를 추가하도록 한다. 이를 위해 메뉴창 설정에 단어추가라는 창을 만든다. .

\* 설정 -> 단어추가 -> 단어 추가 창 실행하여 게임에 나올 수 있는 단어를 추가



\* 첫화면 디자인

게임 첫 화면의 경우 고전게임인 “피카츄 배구”를 모방하여 만들었다.

“혼자서 재미있게”를 누르면 혼자서 할 수 있는 게임을,

“둘이서 신나게”를 누르면 2p로 tcp통신으로 배틀 게임을 할 수 있도록 나중에 만든다. (미완성)

\* 오디오 기능

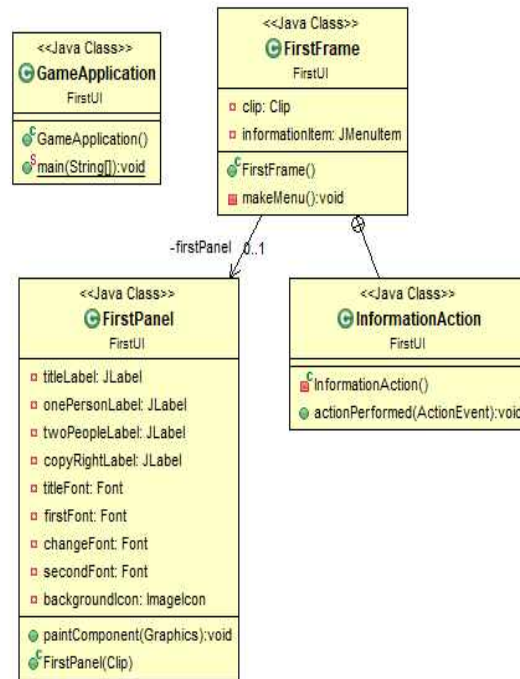
- 1) 맞혔을 때 소리
- 2) 틀렸을 때 소리
- 3) 목숨을 잃었을 때 소리
- 3) 배경음악
  - 소리 조절 기능
  - 켜고 끄는 기능
- 4) GameOver 소리

### 3. 프로그램 구조

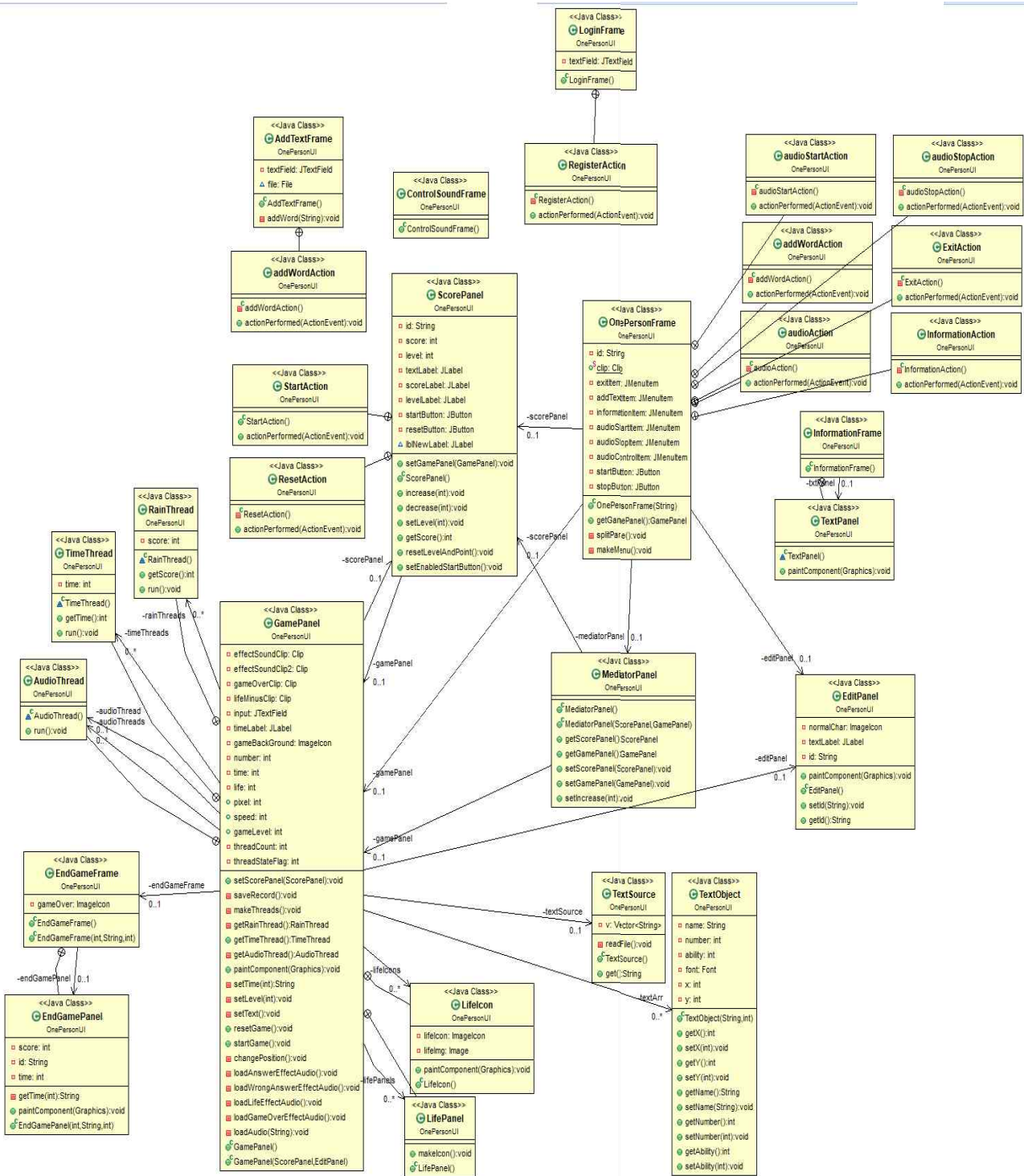
#### \* 파일 구조

- > FinalProjectClient1209 [FinalProjectClient1209 master]
    - > JRE System Library [JavaSE-13]
    - > src
      - > FirstUI
        - > FirstFrame.java
        - > FirstPanel.java
        - > GameApplication.java
        - FirstUI.ucls
      - > OnePersonUI
        - > AddTextFrame.java
        - > ControlSoundFrame.java
        - > EditPanel.java
        - > EndGameFrame.java
        - > GamePanel.java
        - > InformationFrame.java
        - > LoginFrame.java
        - > MediatorPanel.java
        - > OnePersonFrame.java
        - > ScorePanel.java
        - > TextObject.java
        - > TextSource.java
        - OnePersonUI.ucls
      - > TwoPeopleUI
    - > resource
      - > img
        - char\_normal.png
        - favcion\_bugi.png
        - FirstBackground.jpg
        - GameBackground.jpg
        - GameBackground2.jpg
        - gameoverlmg.jpg
        - life.png
        - lose.jpg
        - win.jpg
      - > sound
        - bgm.wav
        - correctEffect.wav
        - gameOverEffect.wav
        - lifeEffect.wav
        - wrongAnswerEffect.wav
      - > word
        - koreanWord.txt

#### 1) FirstUI package 프로그램 구조 UML



## 2) OnepersonUI package 프로그램 구조 UML



#### 4. 작품 상세

- “혼자서 재미있게” : 1p게임

##### (1) 첫 화면

도움 -> 타자게임 정보 프레임

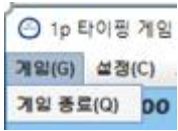
혼자서 재미있게 -> 로그인 프레임 -> 등록버튼을 누를 경우 아이디가 만들어짐

-> 1p 타이핑 게임 프레임이 만들어진다.



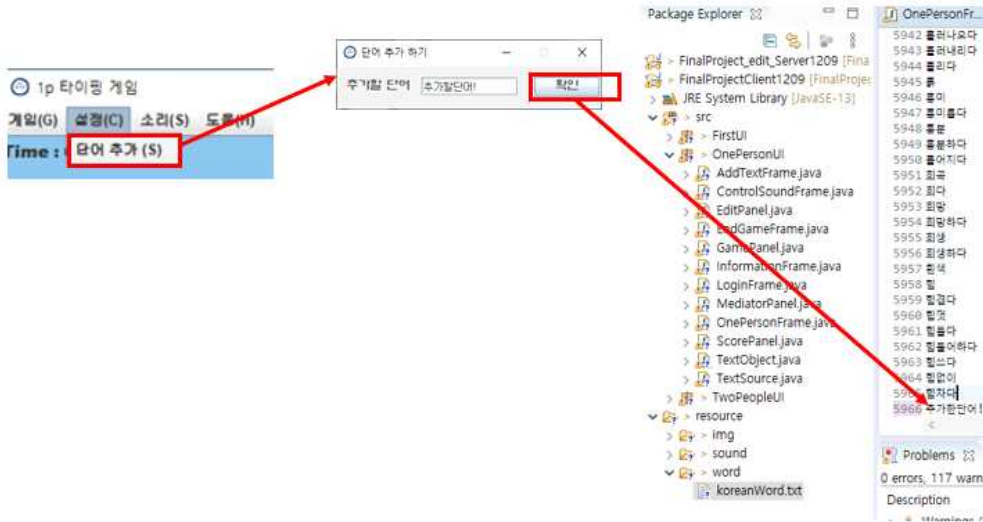
## (2) 1p게임 메뉴관련 기능

### 1) 게임(G)



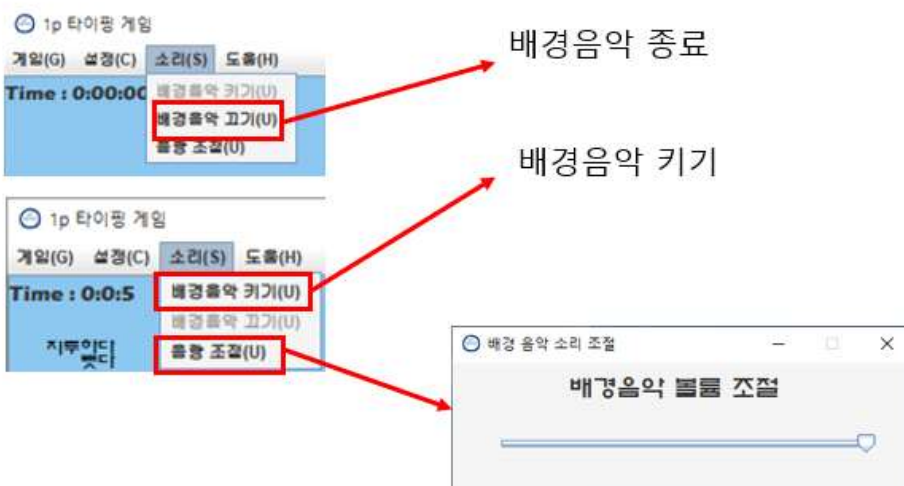
- 게임종료(Q) : 게임종료 기능

### 2) 설정(C)



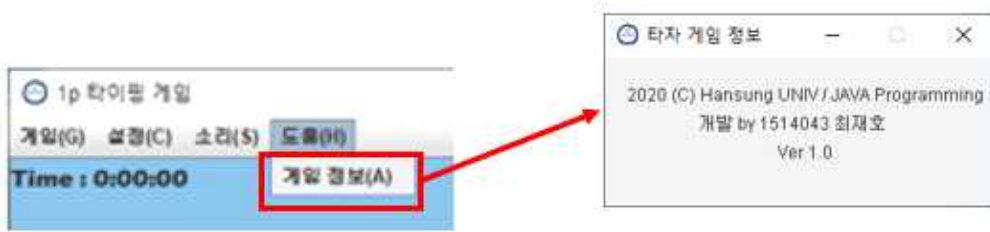
- 단어추가(S) : 단어 추가 기능

### 3) 소리(S)



- 배경음악 키기(U) : 배경음악 키기
- 배경음악 끄기(U) : 배경음악 끄기
- 음량 조절(U) : 배경음악 음량 조절

### 3) 도움(H)



- 게임정보 : 타자게임 정보 프레임 출력

### (2) 1p게임 관련 기능 설명

1) Start 버튼 클릭 -> 게임이 시작됨.



### 2) 게임 진행 특이점

TimerThread : 쓰레드로 만들어 시간을 측정할 수 있게 만들음.

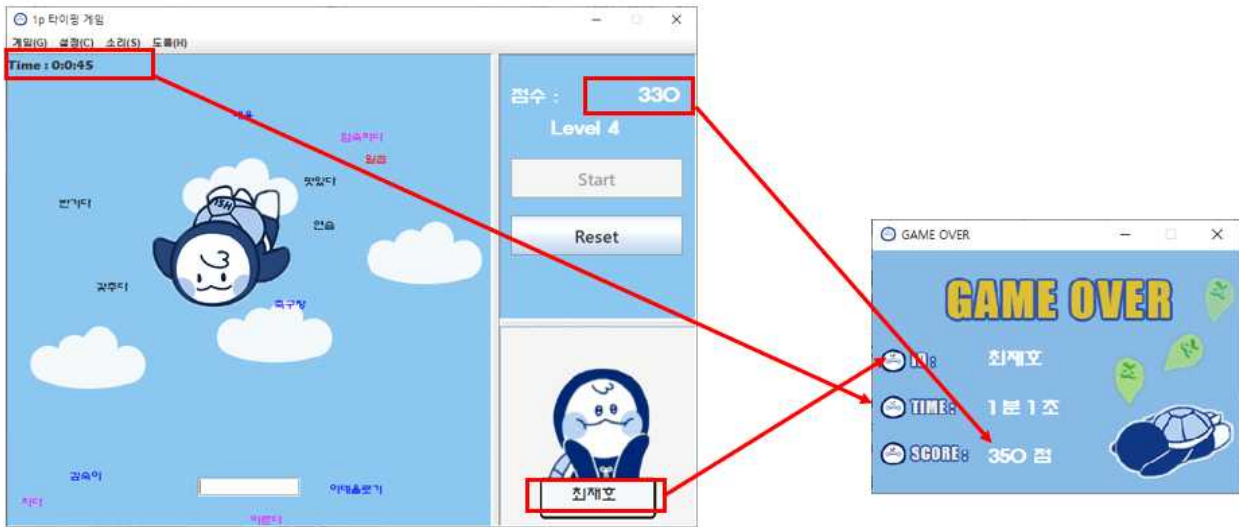
단어가 밑에 떨어지면 하나씩 깎인다. (기본 Life = 100)

- 분홍색 단어 : 모든 단어 삭제
- 검정색 단어 : 기본단어 (기능 X)
- 파란색 단어 : 잠시동안 정지 시킴
- 빨간색 단어 : 속도를 두배 빠르게

- \* 점수  
검정색 = 10점  
파란색 = 20점  
빨간색 = 10점  
분홍색 = 10점
- \* 레벨  
100점이 오를 수록  
레벨이 1씩 오름  
(레벨 별 속도는 위  
글 참조)

게임이 진행되는 판넬로 RainThread 쓰레드를 만들어 게임이 진행되게 만들음

### 3) 게임 종료



life 10개가 모두 깎이면 게임이 종료된다.

게임이 종료되면 id, 시간, 점수 값을 저장해서 GameOver 프레임을 만들어서 사용자에게 보여준다.

그리고 창에 있었던 모든 정보는 지워지게 된다.

게임 재시작 이 가능하며 재시작시 아이디는 저장되서 진행되게 된다.



## 5. 프로그램 소스 코드 (import 문은 제외함)

### (1) FirstUI Pacakage : 첫 화면을 만들기 위한 패키지

#### 1) FirstFrame.java

```
public class FirstFrame extends JFrame {

    private Clip clip;
    private FirstPanel firstPanel = new FirstPanel(clip);
    private JMenuItem informationItem = new JMenuItem("게임 정보(A)");

    public FirstFrame() {
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Image img = toolkit.getImage("resource/img/favcion_bugi.png");
        setIconImage(img);

        setTitle("타자 게임");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(500, 500);
        setLocationRelativeTo(null);
        getContentPane().add(firstPanel);
        makeMenu();
        setResizable(false);
        setVisible(true);
    }

    private void makeMenu() {
        JMenuBar menuBar = new JMenuBar();
        this.setJMenuBar(menuBar);
        JMenu informationMenu = new JMenu("도움(H)");
        informationMenu.add(informationItem);
        menuBar.add(informationMenu);
        informationItem.addActionListener(new InformationAction());
    }

    private class InformationAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            InformationFrame informationFrame = new InformationFrame();
        }
    }
}
```

#### 2) FirstPanel.java

```
public class FirstPanel extends JPanel {
    private JLabel titleLabel = new JLabel("부기 타자 게임");

    private JLabel onePersonLabel = new JLabel("혼자서 재미있게");
    private JLabel twoPeopleLabel = new JLabel("둘이서 신나게");
    private JLabel copyRightLabel = new JLabel("(C) 2020 Choijaeho");

    private Font titleFont = new Font("휴먼엑스포", Font.BOLD | Font.ITALIC, 50);
    private Font firstFont = new Font("휴먼엑스포", Font.BOLD | Font.ITALIC, 20);
    private Font changeFont = new Font("휴먼엑스포", Font.BOLD | Font.ITALIC, 22);
    private Font secondFont = new Font("휴먼엑스포", Font.BOLD | Font.ITALIC, 10);
    private ImageIcon backgroundIcon = new ImageIcon("resource/img/FirstBackground.jpg");

    public void paintComponent(Graphics g) {
        Image originImg = backgroundIcon.getImage();
        g.drawImage(originImg, 0, 0, null);
        setOpaque(false);
        super.paintComponent(g);
    }

    public FirstPanel(Clip clip) {
        setLayout(null);
        titleLabel.setFont(new Font("휴먼엑스포", Font.BOLD | Font.ITALIC, 47));

        titleLabel.setSize(402, 100);
        titleLabel.setLocation(65, 98);
        add(titleLabel);
        onePersonLabel.setForeground(Color.BLACK);

        onePersonLabel.setFont(new Font("휴먼엑스포", Font.BOLD | Font.ITALIC, 20));
        onePersonLabel.setSize(175, 50);
        onePersonLabel.setLocation(155, 260);
        add(onePersonLabel);
        twoPeopleLabel.setForeground(Color.BLACK);

        twoPeopleLabel.setFont(firstFont);
        twoPeopleLabel.setSize(175, 50);
        twoPeopleLabel.setLocation(155, 312);
        add(twoPeopleLabel);

        copyRightLabel.setFont(secondFont);
        copyRightLabel.setSize(220, 50);
        copyRightLabel.setLocation(155, 372);
        add(copyRightLabel);
    }
}
```

```

onePersonLabel.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        System.out.println("Click onePersonLabel");

        LoginFrame loginFrame = new LoginFrame();
    }
    @Override
    public void mouseEntered(MouseEvent e) {
        onePersonLabel.setFont(changeFont);
        onePersonLabel.setForeground(Color.BLUE);
    }
    @Override
    public void mouseExited(MouseEvent e) {
        onePersonLabel.setFont(firstFont);
        onePersonLabel.setForeground(Color.BLACK);
    }
});

twoPeopleLabel.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        System.out.println("Click twoPeopleLabel");
        TwoPeopleFrame twoPeopleFrame = new TwoPeopleFrame();
    }
    @Override
    public void mouseEntered(MouseEvent e) {
        twoPeopleLabel.setFont(changeFont);
        twoPeopleLabel.setForeground(Color.BLUE);
    }
    @Override
    public void mouseExited(MouseEvent e) {
        twoPeopleLabel.setFont(firstFont);
        twoPeopleLabel.setForeground(Color.BLACK);
    }
});
}
}

```

### 3) GameApplication.java

```

public class GameApplication {
    public static void main(String[] args) {
        FirstFrame firstFrame = new FirstFrame();
    }
}

```

## (2) OnePersonUI Pacakage : 1p용 게임 관련 패키지

1) LoginFrame.java : login을 할 때 필요.

```
public class LoginFrame extends JFrame {
    private JTextField textField;

    public LoginFrame() {
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Image img = toolkit.getImage("resource/img/favcion_bugi.png");
        setIconImage(img);

        setTitle("로그인");
        setSize(233, 131);
        getContentPane().setLayout(null);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null); // 화면 가운데 출력시키기
        textField = new JTextField();
        textField.setBounds(87, 10, 116, 21);
        getContentPane().add(textField);
        textField.setColumns(10);

        JButton registerButton = new JButton("\uB4F1\uB85D");
        registerButton.setFont(new Font("휴먼엑스포", Font.PLAIN, 16));

        registerButton.addActionListener(new ActionListener() {
            String id = (String) textField.getText();

            public void actionPerformed(ActionEvent e) {

                String id = textField.getText();
                System.out.println("id : " + id + " 입니다.");
                OnePersonFrame onePersonFrame = new OnePersonFrame(id);
                setVisible(false);
            }
        });

        registerButton.setBounds(24, 41, 179, 35);
        getContentPane().add(registerButton);

        textField.addKeyListener(new KeyAdapter() {
            @Override
            public void keyPressed(KeyEvent e) {
                if (e.getKeyChar() == '\n') {
                    System.out.println("click enter");
                    String id = textField.getText();
                    System.out.println("id : " + id + " 입니다.");
                    OnePersonFrame onePersonFrame = new OnePersonFrame(id);
                    setVisible(false);
                }
            }
        });

        textField.requestFocus();

        JLabel registerLabel = new JLabel("\uC544\uC774\uB514 :");
        registerLabel.setFont(new Font("휴먼엑스포", Font.PLAIN, 12));
        registerLabel.setBounds(24, 13, 81, 15);

        getContentPane().add(registerLabel);
        setResizable(false);
        setVisible(true);
    }

    private class RegisterAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String id = textField.getText();
            System.out.println("id : " + id + " 입니다.");
            OnePersonFrame onePersonFrame = new OnePersonFrame(id);
            setVisible(false);
        }
    }
}
```

2) InformationFrame.java : 게임정보 관련 프레임

```
public class InformationFrame extends JFrame{
    private TextPanel txtPanel = new TextPanel();

    public InformationFrame() {
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Image img = toolkit.getImage("resource/img/favcion_bugi.png");
        setIconImage(img);
        setTitle("타자 게임 정보");
        setSize(288, 142);
        setContentPane(txtPanel);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null); // 화면 가운데 출력시키기
        setResizable(false); //함부로 크기를 변경하지 않게.
        setVisible(true);
    }
    class TextPanel extends JPanel{
        public void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.drawString("2020 (C) Hansung UNIV / JAVA Programming", 15, 30);
            g.drawString("개발 by 1514043 최재호", 65, 50);
            g.drawString("Ver 1.0", 120, 70);
        }
    }
}
```

### 3) OnePersonFrame.java : 게임이 진행되는 프레임

```
public class OnePersonFrame extends JFrame {
    private String id = "";

    public static Clip clip;

    // 전반적으로 액세스 할 수 있게 하기 위해서 빼놓는다.
    private JMenuItem exitItem = new JMenuItem("게임 종료(Q)");
    private JMenuItem addTextItem = new JMenuItem("단어 추가 (S)");
    private JMenuItem informationItem = new JMenuItem("게임 정보(A)");
    private JMenuItem audioStartItem = new JMenuItem("배경음악 키기(U)");
    private JMenuItem audioStopItem = new JMenuItem("배경음악 끄기(U)");
    private JMenuItem audioControlItem = new JMenuItem("음량 조절(U)");
    private JButton startButton = new JButton("Start");
    private JButton stopButton = new JButton("Stop");

    private ScorePanel scorePanel = new ScorePanel();
    private EditPanel editPanel = new EditPanel();
    private GamePanel gamePanel = new GamePanel(scorePanel, editPanel);
    private MediatorPanel mediatorPanel;

    public OnePersonFrame(String id) {
        mediatorPanel = new MediatorPanel();
        mediatorPanel.setGamePanel(gamePanel);
        mediatorPanel.setScorePanel(scorePanel);
        gamePanel.setScorePanel(scorePanel);
        scorePanel.setGamePanel(mediatorPanel.getGamePanel());

        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Image img = toolkit.getImage("resource/img/favcion_bugi.png");
        setIconImage(img);

        this.id = id;
        editPanel.setId(id);
        setTitle("1p 타이핑 게임");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE); // 이창만꺼지기
        setSize(800, 600);

        splitPane(); // JSplitPane을 생성하여 컨테트팬에 CENTER에 부착.
        makeMenu();

        setResizable(false); // 함부로 크기를 변경하지 않게.
        setVisible(true);
    }

    public GamePanel getGamePanel() {
        return gamePanel;
    }

    // ContentPane을 SplitPanne으로 나눌거임.
    private void splitPane() {
        JSplitPane hPane = new JSplitPane();
        // ContentPane()은 BorderLayout이 기본.
        getContentPane().add(hPane, BorderLayout.CENTER);

        hPane.setOrientation(JSplitPane.HORIZONTAL_SPLIT);
        hPane.setDividerLocation(550);

        // 못움직이게 고정시키기. - 이걸 모든 컴포넌트에 다 적용된다.
        hPane.setEnabled(false);

        // 왼쪽에 게임패널 넣기.
        hPane.setLeftComponent(gamePanel);

        // 왼쪽에 붙일 패널과 오른쪽에 붙일 패널을 만들어줘야된다.
        JSplitPane pPane = new JSplitPane();
        pPane.setOrientation(JSplitPane.VERTICAL_SPLIT);
        pPane.setDividerLocation(300);
        pPane.setEnabled(false);

        // ScorePanel붙이기. id 붙이기
        pPane.setTopComponent(scorePanel);

        // EditPanel붙이기
        pPane.setBottomComponent(editPanel);

        // 오른쪽에 pPane 넣기.
        hPane.setRightComponent(pPane);
    }

    // 메뉴바를 만드는 함수. - 14장 1절
    private void makeMenu() {
        JMenuBar menuBar = new JMenuBar();
        this.setJMenuBar(menuBar);

        // 게임 관련 메뉴
        JMenu gameMenu = new JMenu("게임(G)");
        gameMenu.add(exitItem);

        // 설정 관련 메뉴
        JMenu setMenu = new JMenu("설정(C)");
    }
}
```

```

setMenu.add(addTextItem);

// 소리 관련 메뉴
JMenu soundMenu = new JMenu("소리(S)");
soundMenu.add(audioStartItem);
soundMenu.add(audioStopItem);
soundMenu.add(audioControlItem);

// 개발자 정보관련 메뉴
JMenu informationMenu = new JMenu("도움(H)");
informationMenu.add(informationItem);

menuBar.add(gameMenu);
menuBar.add(setMenu);
menuBar.add(soundMenu);
menuBar.add(informationMenu);

// 액션리스너
exitItem.addActionListener(new ExitAction());
addTextItem.addActionListener(new addWordAction());

audioStopItem.addActionListener(new audioStopAction());
audioStartItem.addActionListener(new audioStartAction());
audioControlItem.addActionListener(new audioAction());
informationItem.addActionListener(new InformationAction());

audioStartItem.setEnabled(false);
}

// 액션 리스너 작성
// 게임 종료
private class ExitAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
}

// 게임 단어추가
private class addWordAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        //새로운 창 추가.
        //단어를 추가할 수 있는 창.
        AddTextFrame addTextFrame = new AddTextFrame();
    }
}

private class audioStopAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        clip.stop();
        audioStopItem.setEnabled(false);
        audioStartItem.setEnabled(true);
    }
}

private class audioStartAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        clip.loop(clip.LOOP_CONTINUOUSLY);
        audioStartItem.setEnabled(false);
        audioStopItem.setEnabled(true);
    }
}

private class audioAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        ControlSoundFrame controlSoundFrame = new ControlSoundFrame();
    }
}

// 게임 정보
private class InformationAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        InformationFrame informationFrame = new InformationFrame();
    }
}
}

```

#### 4) GamePanel.java: 게임이 진행되는 로직을 가지는 Panel

```
public class GamePanel extends JPanel {  
  
    // Audio 관련 클립  
    private Clip effectSoundClip; // 정답  
    private Clip effectSoundClip2; // 오답.  
    private Clip gameOverClip; // 게임오버  
    private Clip lifeMinusClip; // 라이프감소  
  
    private JTextField input = new JTextField(25);  
    private JLabel timeLabel = new JLabel("Time : 0:00:00");  
    private ImageIcon gameBackGround = new ImageIcon("resource/img/GameBackground.jpg");  
  
    private Vector<TextObject> textArr = new Vector<TextObject>(50);  
    private ScorePanel scorePanel = null;  
    private EditPanel editPanel = null;  
    private TextSource textSource = new TextSource(); // 단어 백터 생성.  
    private int number = 0;  
    private int time = 0;  
  
    // 초기 라이프 값 = 10  
    private int life = 10;  
    private Vector<Lifelcon> lifelcons = new Vector<Lifelcon>();  
    private Vector<LifePanel> lifePanels = new Vector<LifePanel>();  
  
    public int pixel = 10;  
    public int speed = 10;  
    public int gameLevel = 1;  
  
    // Thread 재시작 하기 위해서 Vector를 사용.  
    private int threadCount = 0;  
    private Vector<RainThread> rainThreads = new Vector<RainThread>();  
    private Vector<TimeThread> timeThreads = new Vector<TimeThread>();  
    private Vector<AudioThread> audioThreads = new Vector<AudioThread>();  
    private int threadStateFlag = 2; // set ..  
    private AudioThread audioThread = new AudioThread();  
    private EndGameFrame endGameFrame = new EndGameFrame();  
  
    public void setScorePanel(ScorePanel scorePanel) {  
        this.scorePanel = scorePanel;  
        System.out.println(scorePanel.getName() + " ScorePanel 만들어짐." + scorePanel.getScore());  
    }  
  
    private void saveRecord() {  
        int scoreTmp = rainThreads.get(threadCount).getScore();  
        String idTmp = editPanel.getId();  
        int timeTmp = timeThreads.get(threadCount).getTime();  
  
        System.out.println(scoreTmp + " " + idTmp + " " + timeTmp);  
        endGameFrame = new EndGameFrame(scoreTmp, idTmp, timeTmp);  
    }  
  
    private void makeThreads() {  
        RainThread rainThread = new RainThread();  
        TimeThread timeThread = new TimeThread();  
        AudioThread audioThread = new AudioThread();  
  
        rainThreads.add(rainThread);  
        timeThreads.add(timeThread);  
        audioThreads.add(audioThread);  
    }  
  
    private RainThread getRainThread() {  
        return rainThreads.get(threadCount);  
    }  
  
    public TimeThread getTimeThread() {  
        return timeThreads.get(threadCount);  
    }  
  
    private AudioThread getAudioThread() {  
        return audioThreads.get(threadCount);  
    }  
  
    public void paintComponent(Graphics g) {  
        Dimension d = getSize();  
        g.drawImage(gameBackGround.getImage(), 0, 0, d.width, d.height, null);  
    }  
  
    private String setTime(int sec) {  
        int min, hour;  
        min = sec / 60;  
        hour = min / 60;  
        sec = sec % 60;  
        min = min % 60;  
        return "Time : " + hour + ":" + min + ":" + sec;  
    }  
  
    class RainThread extends Thread { // 산성비 메소드 쓰레드를 상속받는다.  
        private int score = 0;  
    }  
}
```

```

public int getScore() {
    return score;
}

synchronized public void run() {
    try {
        while (true) {
            // *****Game over
            *****//
            if (Thread.interrupted()) {
                break;
            }
            if (life <= 0) {
                System.out.println("Game over");
                resetGame();
                break;
            }
            score = scorePanel.getScore();
            if (score > 100 && score <= 200) {
                setLevel(2);
            } else if (score > 200 && score <= 300) {
                setLevel(3);
            } else if (score > 300 && score <= 400) {
                setLevel(4);
            } else if (score > 400 && score <= 500) {
                setLevel(5);
            } else if (score > 500 && score <= 600) {
                setLevel(6);
            } else if (score > 600 && score <= 700) {
                setLevel(7);
            } else if (score > 700 && score <= 800) {
                setLevel(8);
            } else if (score > 800 && score <= 900) {
                setLevel(9);
            } else if (score > 900 && score <= 1000) {
                setLevel(10);
            }
            // System.out.println("RainThread start");
            setText();
            changePosition();
            GamePanel.this.repaint();
            // 능력사용 - 속도를 변경 한다.
            if (threadStateFlag >= 200) { // Blue Ability
                System.out.println("** blue text ability run **");
                Thread.sleep(3000);
                threadStateFlag = 0;
            } else if (threadStateFlag < 0) {
                Thread.sleep(500);
            } else {
                threadStateFlag = 0;
                Thread.sleep(1000);
            }
            System.out.println("ability protocol : " + threadStateFlag);
            threadStateFlag++;
        }
    } catch (InterruptedException e) {
        System.out.println("Thread stop");
        e.printStackTrace();
    }
}

class TimeThread extends Thread {
    private int time = 0;

    public int getTime() {
        return time;
    }

    synchronized public void run() {
        while (true) {
            try {
                if (Thread.interrupted()) {
                    break;
                }
                System.out.println("시간 : " + time + " 초");
                time++;
                timeLabel.setText(setTime(time));
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                timeLabel.setText(setTime(0));
                System.out.println("Kill TimeThread");
                e.printStackTrace();
                break;
            }
        }
    }
}

// Score를 기준으로 Level이 변경된다.
private void setLevel(int level) {
    switch (level) {

```



```

    case 1:
        scorePanel.setLevel(1);
        this.speed = 10;
        break;
    case 2:
        scorePanel.setLevel(2);
        this.speed = 15;
        break;
    case 3:
        scorePanel.setLevel(3);
        this.speed = 20;
        break;
    case 4:
        scorePanel.setLevel(4);
        this.speed = 25;
        break;
    case 5:
        scorePanel.setLevel(5);
        this.speed = 30;
        break;
    case 6:
        scorePanel.setLevel(6);
        this.speed = 33;
        break;
    case 7:
        scorePanel.setLevel(7);
        this.speed = 35;
        break;
    case 8:
        scorePanel.setLevel(8);
        this.speed = 37;
        break;
    case 9:
        scorePanel.setLevel(9);
        this.speed = 40;
        break;
    case 10:
        scorePanel.setLevel(10);
        this.speed = 45;
        break;
    }
}

private void setText() {
    // System.out.println("SetText");
    String newWord = textSource.get();
    TextObject textTmp = new TextObject(newWord, number);
    // System.out.println(textTmp.getText() + " 단어 출력.");

    for (int i = 0; i < textArr.size(); i++) {
        TextObject tmp = (TextObject) textArr.get(i);
        tmp.setSize(100, 30);

        // 단어 특징
        switch (tmp.getAbility()) {
            case 0:
                tmp.setForeground(Color.BLACK);
                break;
            case 1:
                tmp.setForeground(Color.BLUE);
                break;
            case 2:
                tmp.setForeground(Color.MAGENTA);
                break;
            case 3:
                tmp.setForeground(Color.RED);
                break;
        }
        tmp.setLocation(tmp.getX(), tmp.getY());
        add(tmp);
    }
    textArr.add(textTmp);
    number++;
}

public void resetGame() {
    scorePanel.setEnabledStartButton();
    // life객체 지우기
    LifePanel lifePanel = lifePanels.get(threadCount);
    lifePanel.removeAll();
    remove(lifePanel);

    number = 0;
    time = 0;
    life = 10;
    pixel = 10;
    speed = 10;
    gameLevel = 1;
    for (int i = 0; i < textArr.size(); i++) {
        TextObject textTmp = (TextObject) textArr.get(i);
        textTmp.setText("");
    }
}

```

```

        getTimeThread().interrupt();
        getRainThread().interrupt();
        getAudioThread().interrupt();

        scorePanel.resetLevelAndPoint();
        textArr.removeAllElements();
        OnePersonFrame.clip.stop();
        loadGameOverEffectAudio();
        saveRecord();
        threadCount++;
    }

    public void startGame() {

        LifePanel lifePanel = new LifePanel();
        lifePanels.add(lifePanel);
        add(lifePanel);

        makeThreads();
        getTimeThread().start();
        getRainThread().start();
        System.out.println((threadCount + 1) + "번째 시도 시작.(threadMake)");

        // 배경음악 스타트!
        loadAudio("resource/sound/bgm.wav");
        getAudioThread().start();
    }

    private synchronized void changePosition() {
        for (int i = 0; i < textArr.size(); i++) {
            TextObject textTmp = (TextObject) textArr.get(i);
            int y = textTmp.getY();

            // *****Life 감소 *****//
            if (y > 500) {
                // 못친 상황.
                textArr.remove(i);
                textTmp.setText("");
                GamePanel.this.repaint();

                Lifelcon icon = (Lifelcon) lifelcons.get(life - 1);
                icon.setText("");
                lifelcons.remove(icon);

                System.out.println("life 감소.");
                life--;
                loadLifeEffectAudio();
            }
            textTmp.setY(y + speed);
        }
    }

    // Life Logic
    class Lifelcon extends JLabel {
        private ImageIccon lifelcon = new ImageIccon("resource/img/life.png");
        private Image lifelmg = lifelcon.getImage();

        public void paintComponent(Graphics g) {
            setOpaque(false);
            super.paintComponent(g);
            g.drawImage(lifelmg, 0, 0, this.getWidth(), this.getHeight(), null);
        }

        public Lifelcon() {
            this.setSize(40, 30);
            this.setForeground(new Color(136, 194, 234));
            this.setText("000");
            System.out.println("그림 아이콘그림");
        }
    }

    class LifePanel extends JPanel {

        public void makeIccon() {
            JLabel tmp = new JLabel();
            tmp.setSize(40, 30);
        }

        public LifePanel() {
            this.setBounds(162, 0, 400, 30);
            this.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
            this.setBackground(new Color(255, 0, 0, 0)); // 투명

            for (int i = 0; i < life; i++) {
                Lifelcon lifelcon = new Lifelcon();
                lifelcons.add(lifelcon);
                add(lifelcon);
            }
        }
    }

```

```

/*
 * * * 오디오 효과 * *
 */
// 맞출때 효과
private void loadAnswerEffectAudio() {
    try {
        effectSoundClip = AudioSystem.getClip();
        File soundFile = new File("resource/sound/correctEffect.wav");
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(soundFile);
        effectSoundClip.open(audioStream);
        effectSoundClip.start();
        System.out.println("맞춤");
    } catch (Exception e) {
        System.out.println("오디오 에러");
    }
}

// 틀릴때 효과
private void loadWrongAnswerEffectAudio() {
    try {
        effectSoundClip2 = AudioSystem.getClip();
        File soundFile = new File("resource/sound/wrongAnswerEffect.wav");
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(soundFile);
        effectSoundClip2.open(audioStream);
        effectSoundClip2.start();
    } catch (Exception e) {
        System.out.println("오디오 에러");
    }
}

// 라이프 감소 효과
private void loadLifeEffectAudio() {
    try {
        lifeMinusClip = AudioSystem.getClip();
        File soundFile = new File("resource/sound/lifeEffect.wav");
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(soundFile);
        lifeMinusClip.open(audioStream);
        lifeMinusClip.start();
        // System.out.println("라이프 감소 사운드");
    } catch (Exception e) {
        System.out.println("오디오 에러");
    }
}

// 게임오버 효과
private void loadGameOverEffectAudio() {
    try {
        gameOverClip = AudioSystem.getClip();
        File soundFile = new File("resource/sound/gameOverEffect.wav");
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(soundFile);
        gameOverClip.open(audioStream);
        gameOverClip.start();
        // System.out.println("게임오버 사운드");
    } catch (Exception e) {
        System.out.println("오디오 에러");
    }
}

// 배경 오디오 올려놓기
private void loadAudio(String pathName) {
    try {
        OnePersonFrame.clip = AudioSystem.getClip();
        File soundFile = new File(pathName);
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(soundFile);
        OnePersonFrame.clip.open(audioStream);
        // System.out.println("배경음악 start");
    } catch (Exception e) {
        System.out.println("오디오 에러");
    }
}

// 배경 오디오 관련 쓰레드
class AudioThread extends Thread {
    @Override
    synchronized public void run() {
        try {
            OnePersonFrame.clip.loop(Clip.LOOP_CONTINUOUSLY); // 무한재생
        } catch (Exception e) {
            timeLabel.setText(setTime(0));
            System.out.println("Kill AudioThread");
            e.printStackTrace();
        }
    }
}

public GamePanel() {
}

public GamePanel(ScorePanel scorePanel, EditPanel editPanel) {
    this.scorePanel = scorePanel;
}

```

```

this.editPanel = editPanel;

setLayout(null);

input.setBounds(216, 481, 116, 21);
add(input);
input.setColumns(10);
timeLabel.setFont(new Font("Eras Bold ITC", Font.PLAIN, 15));
timeLabel.setSize(127, 24);
timeLabel.setLocation(0, 0);
add(timeLabel);

input.addActionListener(new ActionListener() {
    int flag = 0;

    @Override
    public void actionPerformed(ActionEvent e) {
        JTextField t = (JTextField) e.getSource();
        String inWord = t.getText();
        for (int i = 0; i < textArr.size(); i++) {
            TextObject textTmp = (TextObject) textArr.get(i);
            if (textTmp.getText().equals(inWord)) {
                loadAnswerEffectAudio(); // 정답 맞추면 싸운드
                switch (textTmp.getAbility()) {
                    case 0: // Black
                        scorePanel.increase(10);
                        textTmp.setText("");
                        textArr.remove(i); // 지우기
                        repaint();
                        t.setText("");
                        // System.out.println("black ability");
                        break;

                    case 1: // Blue
                        threadStateFlag = 200; // 5초간 stop
                        scorePanel.increase(20);
                        textTmp.setText("");
                        textArr.remove(i); // 지우기
                        repaint();
                        t.setText("");
                        // System.out.println("blue ability");
                        break;

                    case 2: // Pink
                        scorePanel.increase(50);
                        textTmp.setText("");
                        // 모든 내용 지우기.
                        for (int j = 0; j < textArr.size(); j++) {
                            TextObject tmp = (TextObject) textArr.get(j);
                            tmp.setText("");
                        }
                        t.setText("");
                        textArr.removeAllElements();
                        repaint();
                        // System.out.println("Green ability");
                        break;

                    case 3: // RED
                        threadStateFlag = -10; // 5초간 빨라짐.
                        scorePanel.increase(10);
                        textTmp.setText("");
                        textArr.remove(i); // 지우기
                        repaint();
                        t.setText("");
                        // System.out.println("Red ability");
                        break;

                }
                flag = -1; // 정답이 있었다.
            }
            t.setText(""); // 맞추든 틀리든 지워준다.
        } // End for
        // 틀린경우
        if (flag == 0) { // flag가 그냥 나온 경우 -> 틀렸다.
            loadWrongAnswerEffectAudio();
        }
        flag = 0; // 다시 0으로
    }
});
}
}
}

```

## 5) EditPanel.java: 아이디가 저장되는 Panel

```
public class EditPanel extends JPanel{
    private ImageIcon normalChar = new ImageIcon("resource/img/char_normal.png");
    private JLabel textLabel = new JLabel("");
    private String id = " ";

    public void paintComponent(Graphics g) {
        Dimension d = getSize();
        g.drawImage(normalChar.getImage(), 0, 0, d.width, d.height, null);
    }

    public EditPanel() {
        this.setLayout(null);
        textLabel = new JLabel("\uC544\uC774\uB514 \uC790\uB9AC");
        textLabel.setHorizontalAlignment(SwingConstants.CENTER);
        textLabel.setFont(new Font("휴먼엑스포", Font.PLAIN, 17));
        textLabel.setBounds(52, 173, 109, 37);
        add(textLabel);
    }

    public void setId(String id) {
        this.id = id;
        textLabel.setText(id);
    }

    public String getId() {
        return id;
    }
}
```

## 6) MediatorPanel.java: GamePanel과 ScorePanel을 이어주는 중재자역할 Panel

```
public class MediatorPanel extends JFrame {
    private ScorePanel scorePanel;
    private GamePanel gamePanel;

    public MediatorPanel() {}
    public MediatorPanel(ScorePanel scorePanel, GamePanel gamePanel) {
        this.scorePanel = scorePanel;
        this.gamePanel = gamePanel;
    }
    public ScorePanel getScorePanel() {
        return scorePanel;
    }
    public GamePanel getGamePanel() {
        return gamePanel;
    }
    public void setScorePanel(ScorePanel scorePanel) {
        this.scorePanel = scorePanel;
    }
    public void setGamePanel(GamePanel gamePanel) {
        this.gamePanel = gamePanel;
    }
    public void setIncrease(int n) {
        this.scorePanel.increase(n);
    }
}
```

7) ScorePanel.java: 게임 점수를 구현하는 로직을 가지는 Panel

```
public class ScorePanel extends JPanel {
    private String id;

    private int score = 0;
    private int level = 1;
    private JLabel textLabel = null;
    private JLabel scoreLabel = new JLabel("0", SwingConstants.CENTER); // 기본점수.
    private JLabel levelLabel = new JLabel("Level " + level);
    private JButton startButton = new JButton("Start");
    private JButton resetButton = new JButton("Reset");
    private GamePanel gamePanel;
    JLabel lblNewLabel = new JLabel("New label");

    public void setGamePanel(GamePanel gamePanel) {
        this.gamePanel = gamePanel;
    }

    public ScorePanel() {
        this.setBackground(new Color(136, 194, 234));

        textLabel = new JLabel("점수 : ");
        textLabel.setForeground(Color.WHITE);
        setLayout(null);
        textLabel.setFont(new Font("휴먼엑스포", Font.BOLD, 20));
        add(textLabel);
        scoreLabel.setForeground(Color.WHITE);
        scoreLabel.setFont(new Font("휴먼엑스포", Font.BOLD, 22));
        scoreLabel.setSize(80, 31);
        scoreLabel.setLocation(139, 32);
        add(scoreLabel); // scorePanel에 붙였음.
        levelLabel.setForeground(Color.WHITE);
        levelLabel.setFont(new Font("휴먼엑스포", Font.BOLD, 20));
        levelLabel.setSize(117, 45);
        levelLabel.setLocation(57, 62);
        add(levelLabel); // scorePanel에 붙였음.
        startButton.setFont(new Font("Yu Gothic UI Semibold", Font.BOLD, 20));
        startButton.setBounds(12, 117, 195, 45);
        add(startButton);
        textLabel.setSize(135, 32);
        textLabel.setLocation(12, 32);
        add(textLabel); // scorePanel에 붙였음.
        resetButton.setFont(new Font("Yu Gothic UI Semibold", Font.BOLD, 20));
        resetButton.setBounds(12, 182, 195, 45);
        add(resetButton);
        resetButton.setEnabled(false);
        startButton.addActionListener(new StartAction());
        resetButton.addActionListener(new ResetAction());
    }

    public synchronized void increase(int n) {
        score += n;
        scoreLabel.setText(Integer.toString(score));
    }

    public void decrease(int n) {
        score -= n;
        scoreLabel.setText(Integer.toString(score));
    }

    public synchronized void setLevel(int level) {
        this.level = level;
        levelLabel.setText("Level " + level);
    }

    public synchronized int getScore() {
        return score;
    }

    public void resetLevelAndPoint() {
        this.score = 0;
        this.level = 1;
        levelLabel.setText("Level " + 1);
        scoreLabel.setText("0");
    }

    public void setEnabledStartButton() {
        System.out.println("startButton enabled");
        startButton.setEnabled(true);
    }

    public class StartAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            gamePanel.startGame();
            startButton.setEnabled(false);
            resetButton.setEnabled(true);
        }
    }

    // 게임 리셋
    private class ResetAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            gamePanel.resetGame();
            startButton.setEnabled(true);
            resetButton.setEnabled(false);
        }
    }
}
```

}  
8) AddTextFrame.java : 단어를 추가해서 txt파일에 넣을 수 있는 기능을 하는 프레임

```
public class AddTextFrame extends JFrame{
    private JTextField textField;
    File file = new File("resource/word/koreanWord.txt");
    public AddTextFrame() {
        Toolkit toolkit = Toolkit.getDefaultToolkit();
        Image img = toolkit.getImage("resource/img/favcion_bugi.png");
        setIconImage(img);
        setTitle("단어 추가 하기");
        setSize(334, 88);
        getContentPane().setLayout(null);

        JLabel lblNewLabel = new JLabel("\uD944\uAC00\uD560 \uB2E8\uC5B4");
        lblNewLabel.setFont(new Font("휴먼엑스포", Font.PLAIN, 14));
        lblNewLabel.setBounds(12, 10, 78, 26);
        getContentPane().add(lblNewLabel);

        JButton btnNewButton = new JButton("\uD655\uC778");
        btnNewButton.setFont(new Font("휴먼엑스포", Font.PLAIN, 14));
        btnNewButton.setBounds(238, 12, 78, 23);

        getContentPane().add(btnNewButton);

        textField = new JTextField();
        textField.setBounds(102, 13, 116, 21);
        getContentPane().add(textField);
        textField.setColumns(10);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null); // 화면 가운데 출력시키기
        setResizable(false);
        setVisible(true);

        btnNewButton.addActionListener(new addWordAction());
    }

    private void addWord(String word) {
        try {
            //파일에 문자열을 쓴다
            //하지만 파일이 존재한다면 덮어쓰는게 아니라 .
            //그 뒤에 문자열을 이어서 작성한다.
            FileWriter fw = new FileWriter(file, true);
            fw.write(word);
        } catch (Exception e) {
            System.out.println(word + " 저장 완료.");
            System.out.println("단어 저장 오류");
            e.printStackTrace();
        }
    }

    private class addWordAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            //새로운 창 추가.
            //단어를 추가할 수 있는 창.
            String word = textField.getText();
            addWord(word);
            textField.setText("");
        }
    }
}
```

9) ControlSoundFrame.java : 배경음악을 조절할 수 있는 프레임

```
public class ControlSoundFrame extends JFrame {  
    public ControlSoundFrame() {  
        Toolkit toolkit = Toolkit.getDefaultToolkit();  
        Image img = toolkit.getImage("resource/img/favcion_bugi.png");  
        setIconImage(img);  
        setTitle("배경 음악 소리 조절");  
        setSize(403, 146);  
        setVisible(true);  
        getContentPane().setLayout(null);  
        JSlider slider = new JSlider();  
  
        slider.setMinimum(-80);  
        slider.setMaximum(6);  
        slider.setFont(new Font("휴먼엑스포", Font.PLAIN, 14));  
        slider.setPaintLabels(true);  
        slider.setMinorTickSpacing(1);  
        slider.setBounds(34, 46, 320, 45);  
        getContentPane().add(slider);  
  
        JLabel lblNewLabel = new JLabel("\uBC30\uACBD\uC74C\uC545 \uBCFC\uB968 \uC870\uC808");  
        lblNewLabel.setFont(new Font("휴먼엑스포", Font.PLAIN, 20));  
        lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);  
        lblNewLabel.setBounds(34, 10, 305, 26);  
        getContentPane().add(lblNewLabel);  
  
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
        setLocationRelativeTo(null); // 화면 가운데 출력시키기  
        setResizable(false);  
  
        slider.addChangeListener(new ChangeListener() {  
            public void stateChanged(ChangeEvent e) {  
                float value = (float) slider.getValue();  
                FloatControl volume = (FloatControl)  
                OnePersonFrame.clip.getControl(FloatControl.Type.MASTER_GAIN);  
                //System.out.println("최솟값 : " + volume.getMinimum()+ "최대값 : "  
                +volume.getMaximum());  
                volume.setValue((float) value);  
            }  
        });  
    }  
}
```



10) TextObject.java : 떨어트릴 단어 객체

```
class TextObject extends JLabel {
    private String name;
    private int number;
    private int ability; // , 2 = ability 1 , 3 = ability 2 , 4 = ability 3
    private Font font = new Font("휴먼엑스포", Font.PLAIN, 12);

    private int x = (int) (Math.random() * 480) + 10; // 첫 위치값 랜덤하게 준다.
    private int y = 5; // 기본 y값

    public TextObject(String name, int number) {
        super(name);
        this.setFont(font);
        this.number = number;
        this.ability = Math.random() < 0.7 ? 0 : -1;
        if (ability == 0) {
            ability = 0; // color == Black;
        }
        // 20프로 확률로 능력 랜덤으로 출력.
        else if (ability == -1) {
            double percent = Math.random();
            // 3가지 능력
            if (percent < 0.33) {
                ability = 1;
            } else if (percent > 0.33 && percent < 0.66) {
                ability = 2;
            } else if (percent < 1 && percent > 0.66) {
                ability = 3;
            }
        }
    }
    public int getX() {
        return x;
    }
    public void setX(int x) {
        this.x = x;
    }
    public int getY() {
        return y;
    }
    public void setY(int y) {
        this.y = y;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
    public int getAbility() {
        return ability;
    }
    public void setAbility(int ability) {
        this.ability = ability;
    }
}
}
```

11) TextSource.java : koreanWord.txt 파일에서 단어를 꺼내와 Vector에 저장하는 클래스

```
public class TextSource {
    private Vector<String> v = new Vector<String>();
    private void readFile() {
        FileInputStream fstream;
        try {
            fstream = new FileInputStream("resource/word/koreanWord.txt");
            DataInputStream in = new DataInputStream(fstream);
            BufferedReader br = new BufferedReader(new InputStreamReader(in));
            String strLine;
            int i = 0;
            while ((strLine = br.readLine()) != null) {
                v.add(strLine);
                i++;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public TextSource() {
        readFile();
    }
    public String get() {
        int index = (int) (Math.random() * v.size());
        return v.get(index);
    }
}
}
```

12) EndGameFrame.java : 게임오버 되면 나타나는 프레임 클래스

```
public class EndGameFrame extends JFrame {

    private EndGamePanel endGamePanel;
    private ImageIcon gameOver = new ImageIcon("resource/img/gameoverimg.jpg");

    public EndGameFrame() {}
    public EndGameFrame(int score, String id, int time) {
        endGamePanel = new EndGamePanel(score, id, time);

        Toolkit toolkit = Toolkit.getDefaultToolkit();

        Image img = toolkit.getImage("resource/img/favcion_bugi.png");
        setIconImage(img);
        setSize(380, 280);
        setTitle("GAME OVER");

        getContentPane().add(endGamePanel);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        this.setLocationRelativeTo(null); // 화면 가운데 출력시키기
        setVisible(true);
        setResizable(false); // 함부로 크기를 변경하지 않게.
    }

    class EndGamePanel extends JPanel {

        private int score;
        private String id;
        private int time;

        private String getTime(int sec) {
            int min, hour;
            min = sec / 60;
            hour = min / 60;
            sec = sec % 60;
            min = min % 60;
            return min + " 분 " + sec + " 초";
        }

        public void paintComponent(Graphics g) {
            Dimension d = getSize();
            g.drawImage(gameOver.getImage(), 0, 0, d.width, d.height, null);
        }

        public EndGamePanel(int score, String id, int time) {
            this.score = score;
            this.id = id;
            this.time = time;

            setLayout(null);
            setBounds(0, 0, 392, 261);

            JLabel timeLabel = new JLabel("");
            timeLabel.setFont(new Font("휴먼엑스포", Font.BOLD, 18));
            timeLabel.setForeground(Color.WHITE);
            timeLabel.setBounds(114, 145, 178, 26);
            timeLabel.setText(getTime(time));
            add(timeLabel);

            JLabel idLabel = new JLabel(id);
            idLabel.setFont(new Font("휴먼엑스포", Font.BOLD, 18));
            idLabel.setForeground(Color.WHITE);
            idLabel.setBounds(114, 96, 178, 26);
            add(idLabel);

            JLabel scoreLabel = new JLabel("0");
            scoreLabel.setFont(new Font("휴먼엑스포", Font.BOLD, 18));
            scoreLabel.setForeground(Color.WHITE);
            scoreLabel.setBounds(114, 193, 178, 26);
            String strScore = Integer.toString(score);
            scoreLabel.setText(strScore + " 점");
            add(scoreLabel);
        }
    }
}
```

## 6. 결론

이번 프로젝트를 통해서 많은 것을 배운 것 같습니다. 처음 계획한 것과는 다르게 결과물이 나왔지만 프로젝트를 만들면서 책에서는 배울수 없는 걸 많이 배웠습니다. 멀티 쓰레드를 이용해서 어떤 프로그램을 만드는 것이 처음이라 어려웠지만 하나씩 해결해나가면서 너무 재밌게 만들었던 것 같습니다.

자바 스윙, 멀티스레드, 그래픽, 오디오, 파일입출력 등의 기술을 종합적으로 사용해서 만들어 이부분에 대해서 많이 배웠습니다. 어려웠던 부분은 쓰레드를 재시작 하는 부분이었는데 사실 쓰레드는 죽어 버리면 다시 재시작 할 수 없는 것이었습니다. 쓰레드의 라이프사이클의 개념에 대해 부족해서 이부분에 대해서 공부를 하면서 해결했습니다. 쓰레드를 벡터에 넣어서 벡터에서 꺼내서 쓰면서 재시작을 하게 했습니다. 메모리낭비가 있을 것 같다는 생각을 했는데 어떨지 모르겠습니다.

그리고 결합도에 대한 문제가 많이 있었습니다. 어떤 필드를 계속 참조하고 참조하다 보니 너무 복잡해져서 어떻게 동작하는지 모르겠을 때가 많았습니다. 특히 ScorePanel, GamePanel, OnePersonFrame 이 세 클래스의 참조가 어려웠습니다. ScorePanel 에서의 점수를 GamePanel에서 참조해야하고, GamePanel에서의 필드를 ScorePanel 에서도 참조해야하는데 어떻게 해야 할지 몰랐는데, 검색을 하다 보니 설계 패턴 중에 “중재자 패턴”이라는 것이 있어서 공부한 뒤 완벽하지 않지만 MediatorPanel을 넣어서 참조에 대한 결합도를 조금 낮췄습니다.

이번 과제를 통해서 가장 공부의 필요성을 느낀 부분은 “설계”입니다. 처음에 설계를 잘해야지 좀더 효율적으로 코딩을 할 수 있을 것 같았습니다. 처음에 그냥 아무런 계획없이 이렇게 이렇게 해야 겠다 생각만 하고 코딩을 하다보니 얽히고 설킨 복잡해지는 것을 알게 되었습니다. 설계패턴에 대해서 공부를 많이해서 좀더 효율적인 코드를 만드는 컴퓨터 전문가로 성장하고자 결심을 하였습니다.

이번과제로 정말 많이 느꼈고 종강하고 2p 로 네트워크에서 게임이 동작하게 만들도록 해야겠다고 생각이 들었습니다.