

Slippery Floor

Team #3

박준수 박상우 홍예진

Dept. Electronics and Computer Engineering

Chonnam National University

Gwangju

I .Motivation

프로그램이란 사용자의 명령에 맞는 결과를 구현해낼 수 있는 기능의 집합체라고 생각합니다. 따라서 저희 3조는 한 학기 내 명확히 구현 가능한 프로그램을 만드는 것을 첫 번째 목표로 선정하였고, 그에 따라 간단한 게임을 만드는 것이 가장 적절하다고 판단하였습니다. 이와 함께 현실적인 구현가능성과 각 조원의 흥미를 고려하여 상의한 결과 Slippery Floor라는 게임을 만들기로 하였습니다. 저희의 Slippery Floor는 어렸을 적 즐겨 플레이 하였던 포켓몬스터(일명 주머니 괴물) 게임의 일부인 '얼음동굴'이라는 스테이지의 독립적인 구현을 목표로 하고 있으며, 저희 조와 같이 포켓몬스터를 즐겨한 이들에게 과거를 추억할 수 있길 희망하고 있습니다.

II .Characteristics

1. 장애물에 부딪힐 때 까지 계속 움직인다.
2. 움직이는 도중에 다른 방향으로 이동할 수 없다.
3. 여러 개의 스테이지가 준비되어있다.
4. 스테이지가 거듭될수록 난이도가 올라간다.
5. 기록을 세울 수 있다.
6. 간단한 스토리가 있다.

III.Function Specification

GamePlayManager.GamePlaying	public void Draw_Background()	배경그리기 메소드
	public void Draw_Player()	플레이어 그리기 메소드
	public void Draw_Wall()	오브젝트 그리기 메소드
	public void init()	이미지 선언
	public void KeyEat()	열쇠 획득시 발동되는 메소드
	Public void KeyProcess()	KeyPressed에 의해 받은 키값을 통해 이동시켜주는 메소드
	public void	KeyListener의 키 입력값을 받아서 keyProcess를 설정해주는 메소드

keyPressed(KeyEvent e)	
public void keyReset()	모든 키 값을 초기화하는 메소드
public void nextScene()	조건에 부합하면 스테이지를 넘어가는 메소드
paint(Graphics g)	이미지를 그려주는 메소드
public void pauseGame()	일시 정지 메소드
public void prevScene()	조건에 부합하면 이전 스테이지로 돌아가는 메소드
public void resetPauseCounter()	pauseCounter를 0으로 세팅하는 메소드
public void resumeGame()	게임 재개 메소드
public void run()	쓰레드 메소드 run() 오버라이드
public void safeZone()	세이프존 입장시 발동되는 메소드
public void start()	키 리스너를 추가하고 게임 플레이 쓰레드 생성
public void update(Graphics g)	주기적으로 호출되는 메소드

GamePlayManager .MergedGamePanel	public void initGamePlaying()	게임 플레이 패널 초기화
	public void initKeySet()	키셋 초기화
	public void initMenuDialog()	메뉴 다이얼로그 초기화
	public void initStatePanel()	스테이터스 패널 초기화
	public void resetGame()	게임 화면 초기화 메소드

GamePlayManager.StatusPanel	public void initText()	기본적으로 보여질 상태창 설정
	public void printGetKey()	열쇠 획득시 상태창 설정
	public void printNotEnoughKey()	열쇠 조건 불충족시 상태창 설정
	public void printPause()	일시 정지시 상태창 설정

GamePlayManager.TimerPanel	public void resetCount()	게임 중 메뉴로 돌아가거나 게임이 끝났을 때 카운트 값을 리셋하는 메소드
	public void run()	Runnable run() 메소드 구현 쓰레드 인터럽트 발생시 리턴
	public void initLabel()	필요한 라벨들을 초기화하고 패널에 추가
	public void paintComponent (java.awt.Graphics g)	스톱워치(타이머) 이미지 셋팅
	public void setCountImage()	시간을 출력하는 count값을 받아 한자리수마다 그림으로 저장하는 메소드

SFDesignManager.Design	public JButton setbutton (JButton bt, int i)	버튼 크기, 이미지 지정 및 투영화 메소드 버튼 구분을 위한 정수형 변수를 매개변수로 받는다
-------------------------------	---	---

SFDesignManager.Sound	public void loop(int i)	매개 변수로 받은 수의 + 1만큼 반복 재생
	public void loop (String path, int i)	매개 변수로 받은 경로의 오디오 파일을 매개 변수로 받은 수의 + 1만큼 반복 재생
	public void play()	오디오 파일 재생
	public void play (java.lang.String path)	매개변수로 받은 경로에 있는 오디오 파일 재생
	public void stop()	오디오 파일 정지
	public void setbutton (JButton bt)	매개변수로 받은 버튼 컴포넌트에 사운드 부착

SFDesignManager. EndingPanel	public void run()	Runnable 메소드 run() 오버라이드 애니메이션 효과를 구현할 메소드
	public void start()	쓰레드 객체 초기화 및 시작
	public void paintComponent (Graphics g)	쓰레드에 따른 카운트 값 변동에 따라 애니메이션 효과 출력

WindowManager.ConfigPanel	public void actionPerformed (ActionEvent e)	키 셋팅 저장을 위한 액션리스너의 메소드 오버라이드
	public void changeAndSave()	입력된 키 코드 값들을 다시 환경설정 값으로 저장하는 메소드
	public void	키리스너 메소드 오버라이드

	keyPressed(KeyEvent e)	키를 입력하면 텍스트라벨에 입력받은 키를 출력하는 메소드
	public void updateKey (KeySet key)	저장된 키 세팅 값으로부터 키 코드를 다시 불러오기 위한 메소드

WindowManager.GamePanel	public void actionPerformed (ActionEvent e)	액션리스너 메소드 오버라이드 각각 버튼을 눌렀을 때 실행될 액션리스너 메소드 구현 통합 패널을 불러올 때 게임 스레드와 타이머 스레드가 시작
	public void setButton()	버튼의 이미지와 이벤트 발생시 실행될 메소드 구현
	public void setMain()	패널의 배경과 버튼 액션리스너 설정
	public void setSouth()	하단 패널 초기화

WindowManager.RankPanel	public void copy (RankInfo src, RankInfo dst)	순위 입력할 때 사용자의 순위 정보를 복사하는 메소드
	public void sort (RankInfo[] k)	순위를 표시할 때 정렬하기 위한 메소드
	public void updateRank()	메인 패널에서 랭크 패널을 불러올 때 랭크를 최신화 하기위한 메소드

WindowManager.MainPanel	public void setmain()	버튼 크기, 위치, 디자인 조정, 부착의 세부사항 처리.
--------------------------------	-----------------------	---------------------------------

IV.Requirement List

1. 조작법을 알려준다.(샘플 스테이지 or 초반 독백)
2. 소리 조절을 할 수 있다.(배경음,효과음)
3. 게임 시작을 눌러 튜토리얼(만화) 창으로 넘어 갈 수 있다.
4. skip 을 눌러 튜토리얼을 넘어가 스테이지 선택 창으로 이동할 수 있다. (stage 단위로 세이브)
5. stage 에 따라 장애물의 수와 방의 크기가 달라진다.
6. 일시정지 버튼이 있어 처음으로 돌아가기와 게임 계속하기, 환경설정을 선택 할 수 있다.
7. 게임 진행 화면에서는 게임 진행 시간을 볼 수 있다.
8. 방향키를 누르면 캐릭터가 그 방향으로 장애물에 닿을 때까지 이동한다.
9. 캐릭터는 벽과 장애물을 지나갈 수 없다.
10. 열쇠의 위치에 도달하면 열쇠를 획득하게 된다.
11. 스테이지별 일정 개수의 열쇠를 획득시 문이 열린다.
12. esc 키를 눌러 게임을 일시정지 시킬 수 있다.
13. 클리어시 다음 스테이지로 이동한다.
14. 특정 이벤트 발생시 엔딩 이미지를 볼 수 있다.

V.Usecase

1. 게임을 실행하면 주 메뉴가 나타난다.
 - 1-1. 게임시작, 순위, 환경설정, 종료 중 하나를 선택한다.
 - 1-1-1. 게임시작을 누르면 2 로 넘어간다.
 - 1-1-2. 순위를 누르면 저장된 순위를 보여준다.
 - 1-1-2-1. 순위 초기화 버튼을 누르면 순위가 초기화된다.
 - 1-1-2-2. 뒤로가기 버튼을 누르면 1-1 로 돌아간다.
 - 1-1-3. 환경설정을 누르면 키설정과 효과음 설정 창이 나타난다.
 - 1-1-3-1. 키설정 버튼을 누르면 사용자로부터 키를 입력받는다.
 - 1-1-3-2. 효과음 설정 버튼을 누르면 음소거 여부 체크박스를 띄운다.
 - 1-1-3-3. 뒤로가기 버튼을 누르면 1-1 로 돌아간다.
 - 1-1-4. 종료 버튼을 누르면 프로그램이 종료된다.
2. 게임시작을 누르면 게임이 진행된다.
 - 2-1. 게임의 스토리를 보여준다.

- 2-1-1. 다음 버튼을 누르면 다음 장면이 나타난다.
 - 2-1-1-1. 마지막 장면에 도달하면 게임 화면이 나타난다.
 - 2-1-2. 건너뛰기 버튼을 누르면 게임 화면이 나타난다.

3. 게임화면

- 3-1. 게임 화면은 게임 플레이 부분과 상태창 부분으로 나뉜다.
 - 3-1-1. 게임 플레이 부분에서 사용자의 조작이 나타난다.
 - 3-1-1-1. 스테이지에는 장애물, 아이템, 안전구역이 있다.
 - 3-1-1-2. 스테이지가 거듭될수록 각 객체의 위치 구조가 변한다.
 - 3-1-2. 화면의 오른쪽 상단에는 게임 클리어까지의 시간을 측정해서 출력한다.
 - 3-1-3. 화면의 하단부에는 상태창이 나타난다.
 - 3-1-3-1. 상태창은 텍스트 출력, 아이템 표시, 다시하기 버튼이 있다.
 - 3-1-3-1-1. 스토리가 진행되면 게임 플레이중 텍스트가 출력된다.
 - 3-1-3-1-2. 아이템을 획득하면 텍스트를 출력하고 상태창에 표시된다.
 - 3-1-3-1-3. 다시하기를 선택하면 스테이지를 다시 시작한다.

4. 게임 조작은 키보드로 이루어진다.

- 4-1. 기본 조작은 키보드의 방향키로 이루어진다.
- 4-2. 환경 설정에서 사용자가 설정한 키로 조작한다.
- 4-3. 플레이어가 장애물이나 안전지대에 도달할 때까지 계속 움직인다.
 - 4-3-1. 이동 중에는 다른 키입력을 무시한다.
 - 4-3-2. 안전지대에서는 자유롭게 이동한다.

5. 이벤트 발생

- 5-1. 장애물에 부딪힌다.
 - 5-1-1. 플레이어가 정지한다.
 - 5-1-2. 다음 키입력을 받을 준비를 한다.
- 5-2. 안전지대에 도달했다.
 - 5-2-1. 플레이어가 정지한다.
 - 5-2-2. 안전지대 내부에서 자유롭게 이동한다.

5-2-3. 안전지대를 벗어나는 순간 향하는

방향으로 이동을 계속한다.

5-3. 아이템을 획득했다.

5-3-1. 다음 스테이지로 넘어갈 수 있게 된다.

5-4. 다음 스테이지로 가는 문에 도달했다.

5-4-1. 조건이 충족되는 경우 다음 스테이지로 넘어간다.

5-4-2. 조건이 충족되지 않는 경우 상태창에 메시지를 출력한다.

6. 게임 클리어

6-1. 에필로그 이미지를 불러온다.

6-1-1. 클릭하여 다음 이미지를 불러온다.

6-1-1-1. 마지막 이미지라면 랭킹 입력으로 이동한다.

6-1-2. 넘어가기 버튼을 이용하여 바로 랭킹 입력으로 이동한다.

6-2. 사용자로부터 문자열을 입력 받아 걸린 시간에 따라 순위를 정한다.

6-3. 주 메뉴로 돌아간다.

VI. Usecase Diagram

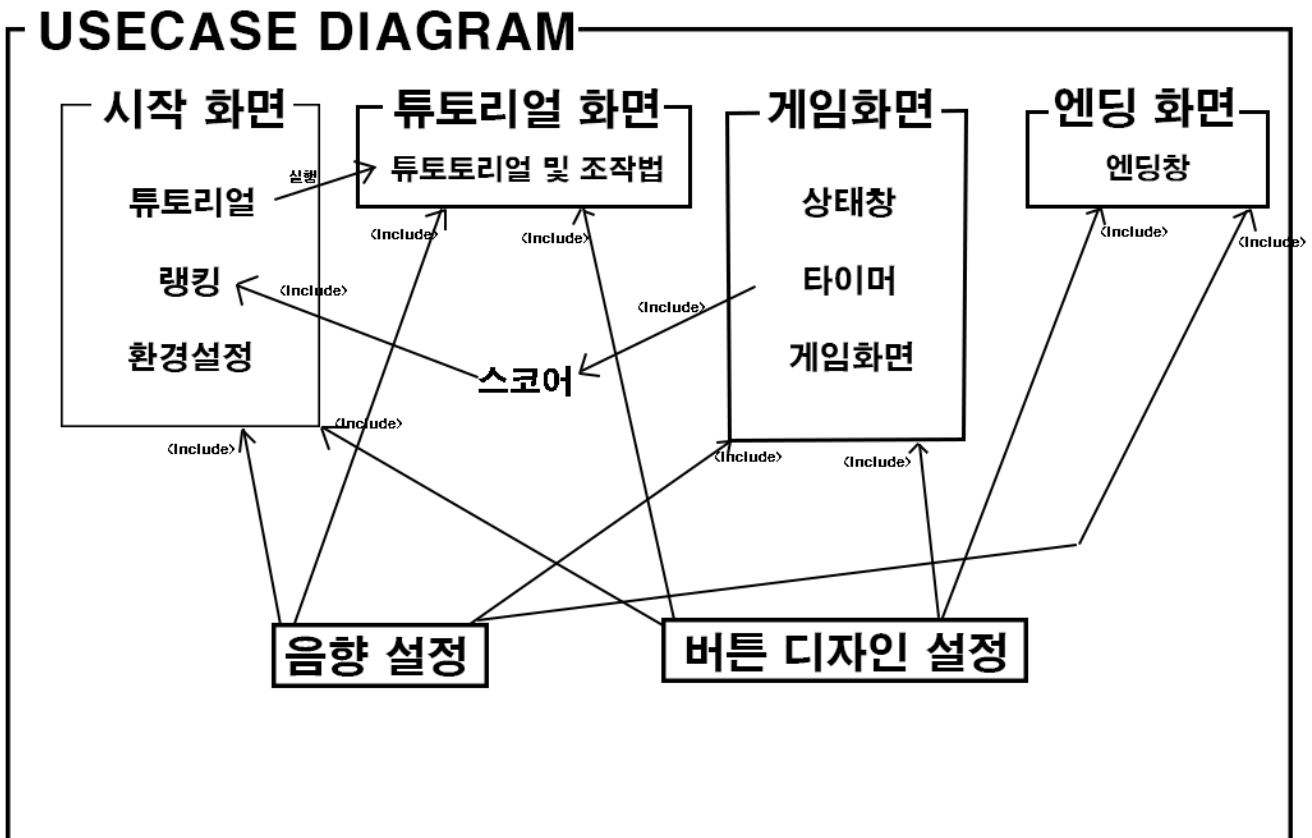


Fig. 1. Usecase Diagram

VII. Sequence Diagram

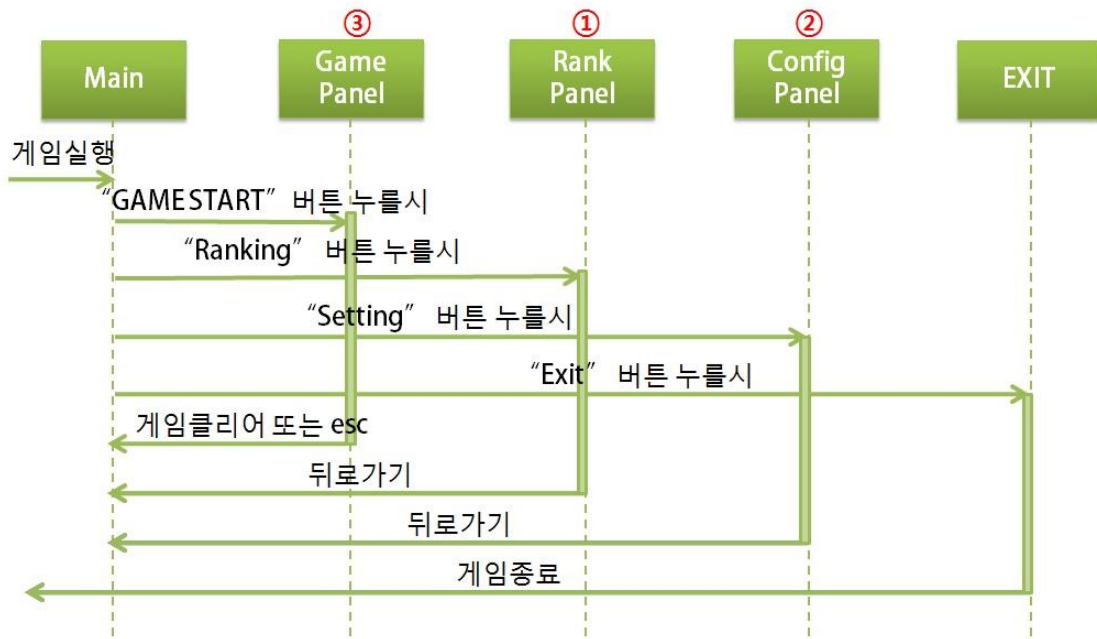


Fig. 2. 전체 Sequence Diagram

① Ranking Panel



Fig. 3. Ranking Panel

② Config Panel

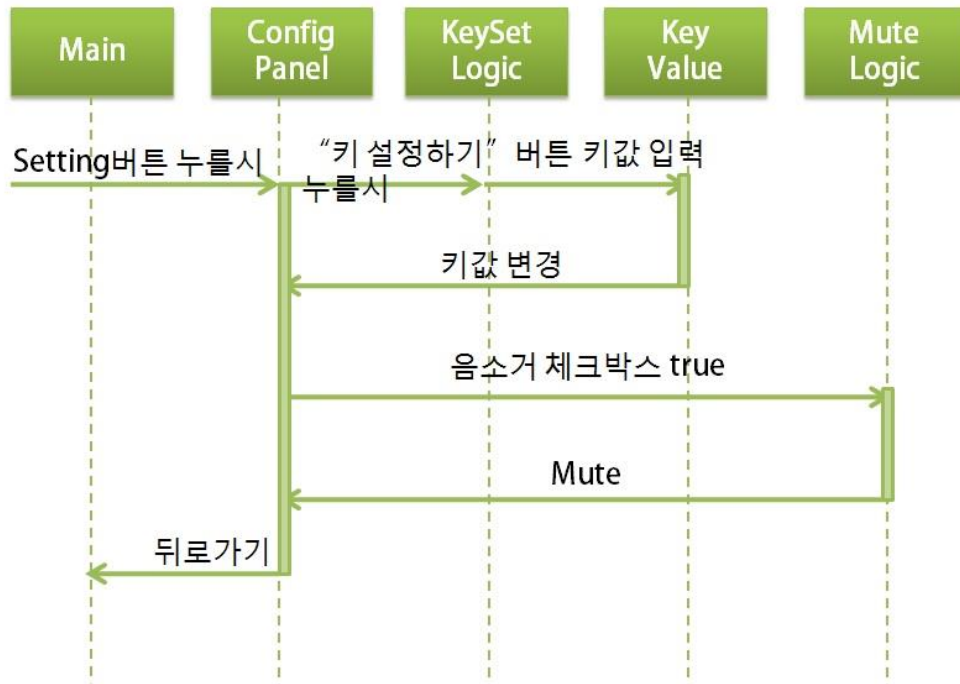


Fig. 4. Config Panel

③ Game Panel

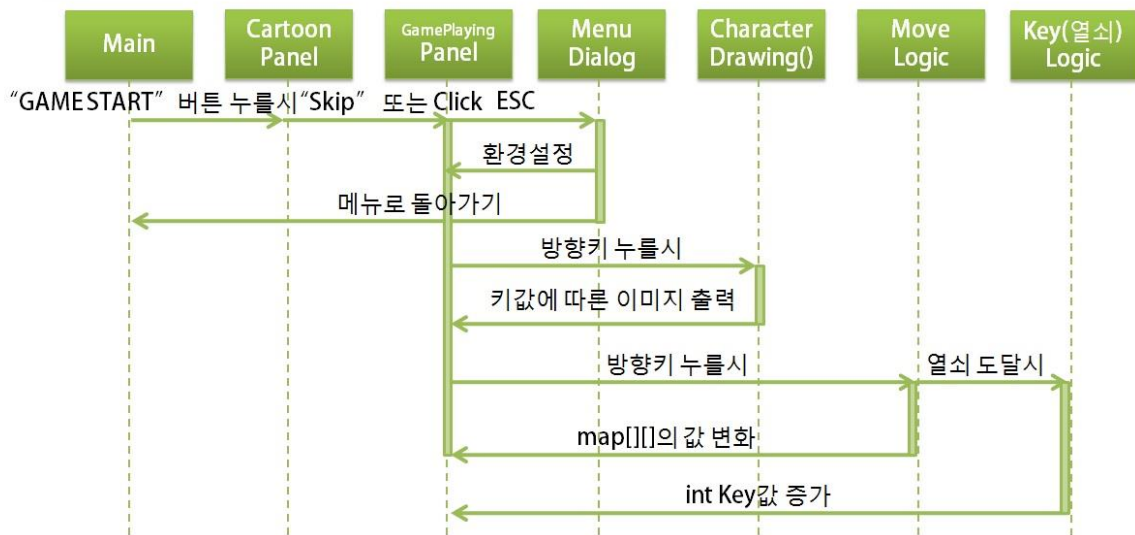


Fig. 5. Game Panel

VIII. User Interface Designs & Screenshot of Application Demo



(1)게임 시작 화면 (MainPanel)

1. 메인 이미지 로고1
2. 메인 이미지 로고2
3. 게임 패널로 넘어가는 버튼
4. 랭킹 패널로 넘어가는 버튼
5. 환경 설정 패널로 넘어가는 버튼
6. 다이얼로그창을 통해 게임을 종료할 것인지 확인하는 버튼



(2)랭크 패널 화면(RankPanel)

1. 메인 패널로 돌아가는 버튼
2. 외부데이터를 디폴드값으로 바꾸는 버튼
3. 시간 목록을 나타내는 시간 라벨
4. 이름 목록을 나타내는 이름 라벨
5. 순위정보(시간,이름)을 나타내는 라벨



(3)셋팅 패널 화면(ConfigPanel)

- | | |
|--|----------------------|
| 1. 메인 패널로 돌아가는 버튼 | 4. 배경화면 음소거 버튼 |
| 2. 상하좌우 이동 키를 지정할 수 있는 버튼 | 5. 키 셋팅 상태를 알려주는 라벨. |
| 3. 지정하고자하는 이동 키를 결정할수 있는 버튼
(key setting 클릭시 활성화) | |



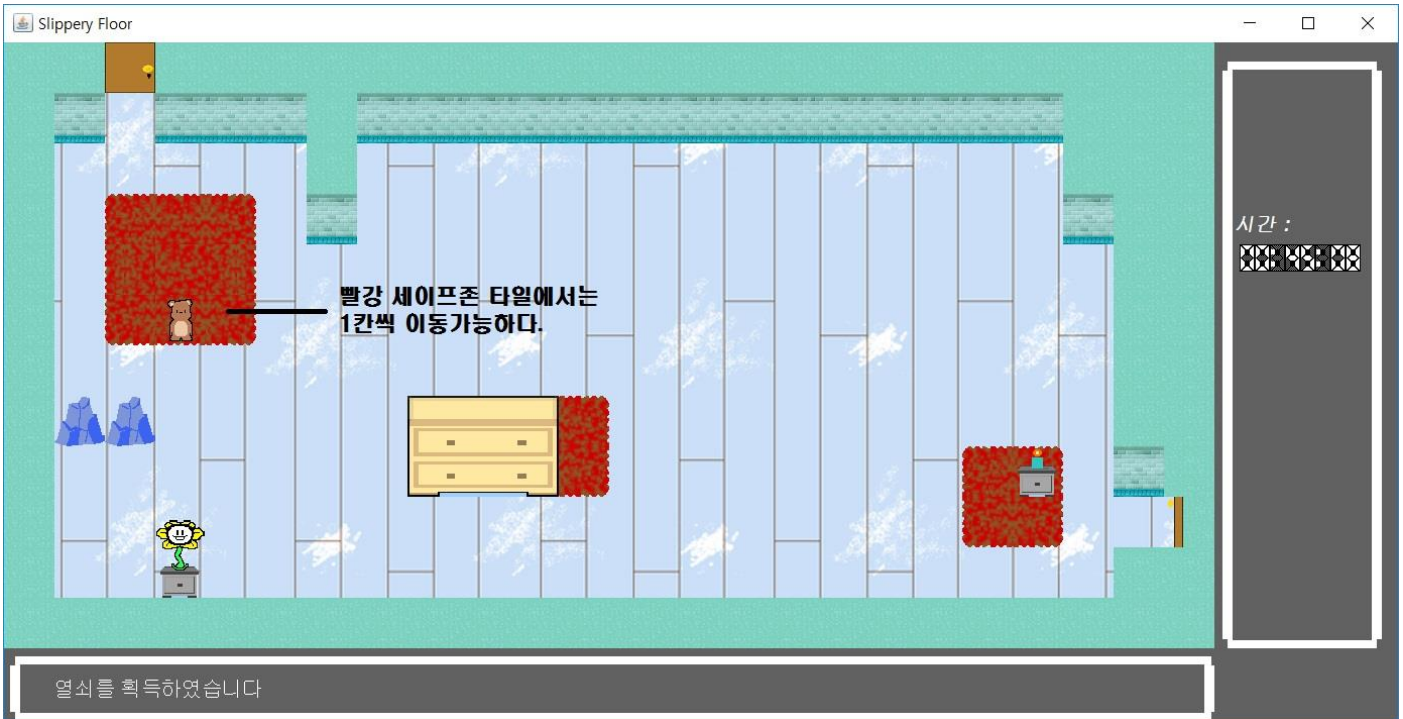
(4)종료 다이얼로그 화면(ExitDialog)

- | | |
|--------------------|-----------------------------|
| 1. 종료확인질문을 나타내는 라벨 | 3. 다이얼로그를 닫고 메인 패널로 돌아가는 버튼 |
| 2. 프로그램을 종료하는 버튼 | |



(5)게임 패널 화면(GamePanel)

1. 게임 패널 설명 라벨
2. 스토리라인 및 조작법 안내 패널
3. 게임 패널로 넘어가는 버튼
4. 메인 패널로 돌아가는 버튼



(6)게임 패널, 상태 패널, 타이머 패널(MergedPanel)

1. 스테이지 이동 시 필요한 열쇠.
2. 5번의 위치 외의 곳에 있을 때 상하좌우로 연속적인 이동
3. 클리어 소요시간을 나타내는 count up 타이머
4. 정지상태, 아이템 획득, 도움말 출력 등 상태 출력 라벨



(7)랭크 다이얼로그 화면(GetNameDialog)

1. 랭크 이름 입력을 안내하는 라벨
2. 이름을 입력받는 텍스트 필드
3. 입력받은 값을 기록, 엔딩패널을 불러오는 버튼



(8)엔딩 패널 화면(EndingPanel)

- 1~3. 동작에 따라순서대로 출력되는 이미지

IX. ClassDiagram

(1) 클래스 다이어그램 설계

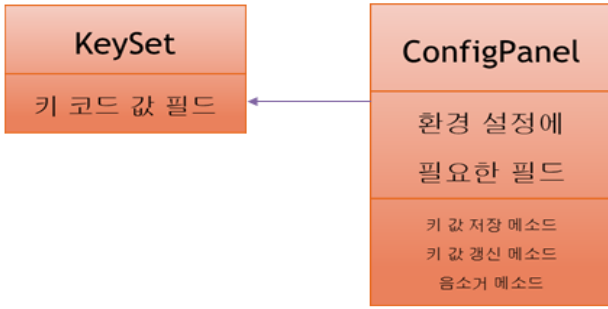


Fig. 6. ConfigPanel 과 KeySet 의 관계 설계

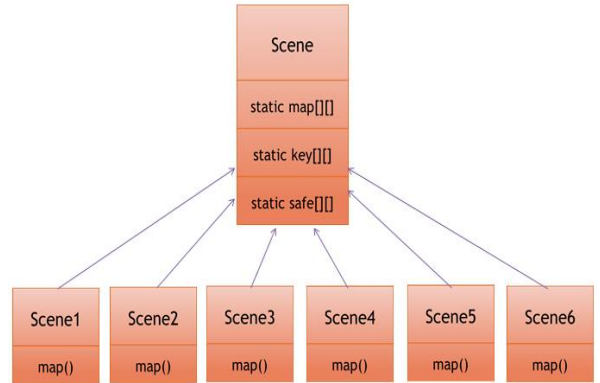


Fig. 9. Scene class 설계

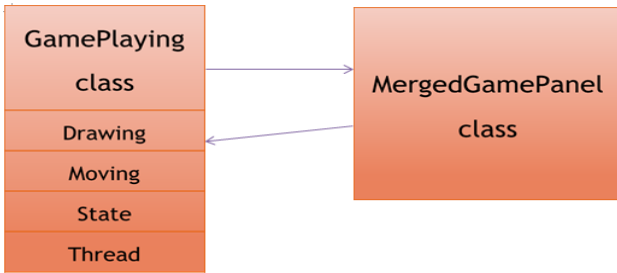


Fig. 7. MergedGamePanel 과 GamePlaying class 의 관계 설계

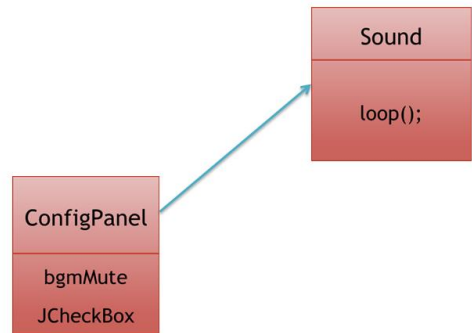


Fig. 10. Sound 와 config 클래스관계 설계

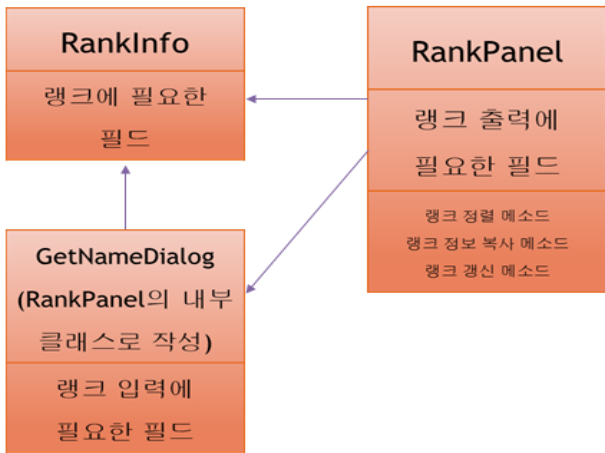


Fig. 8. RankPanel 설계

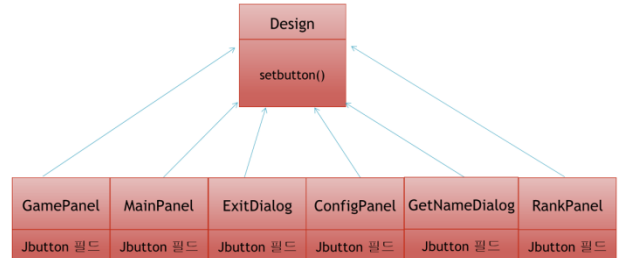


Fig. 11. Design 클래스 설계

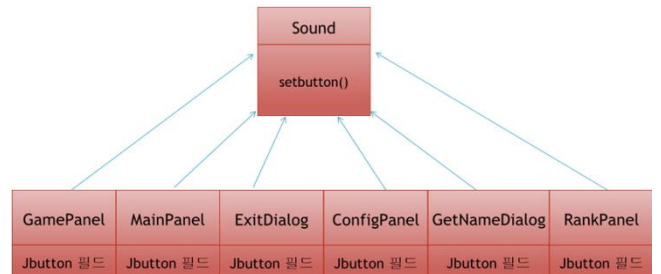


Fig. 12. Sound 클래스 설계

(2) 실제 클래스 다이어그램

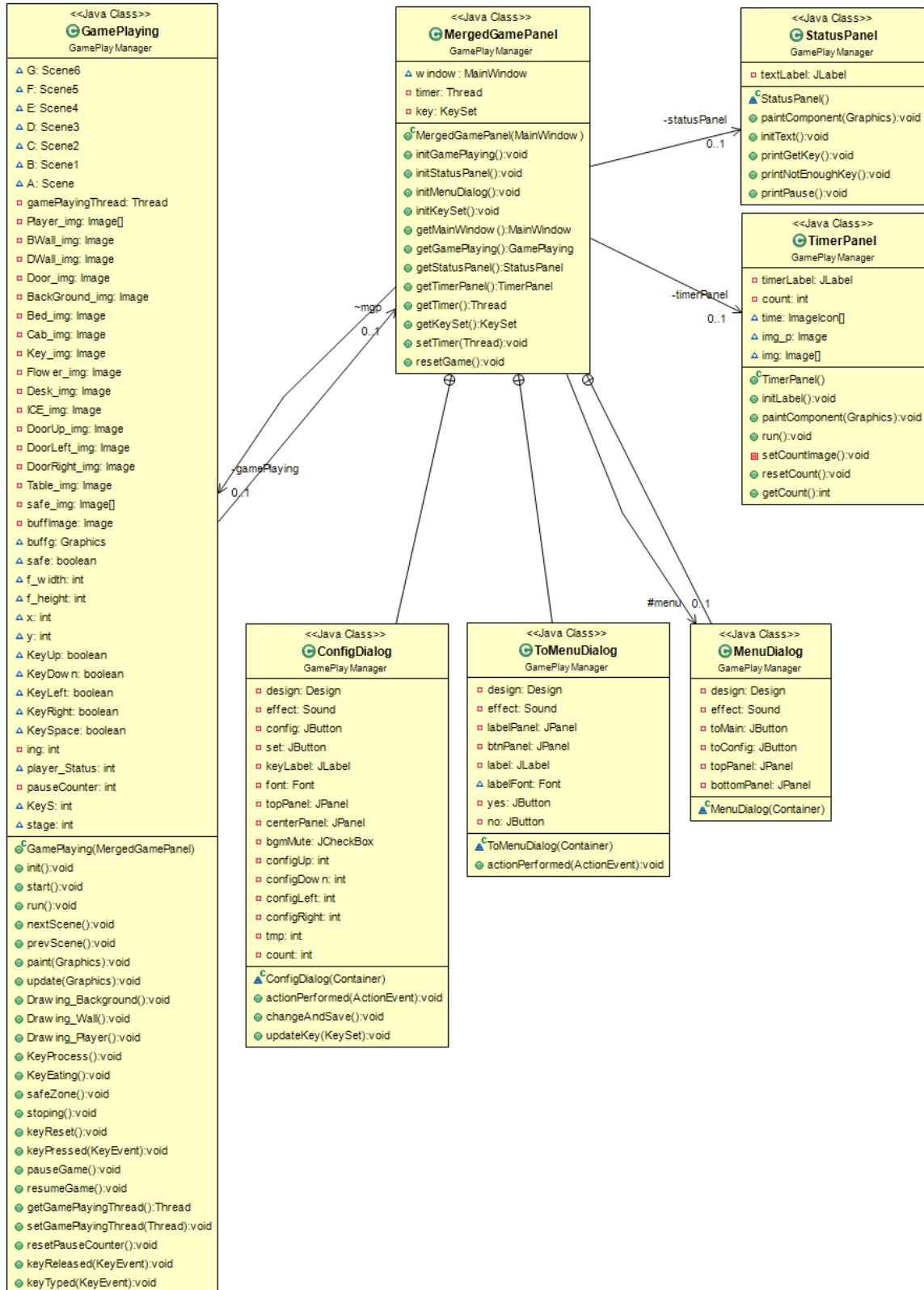


Fig. 13. 게임 플레이에 관련된 클래스 다이어그램

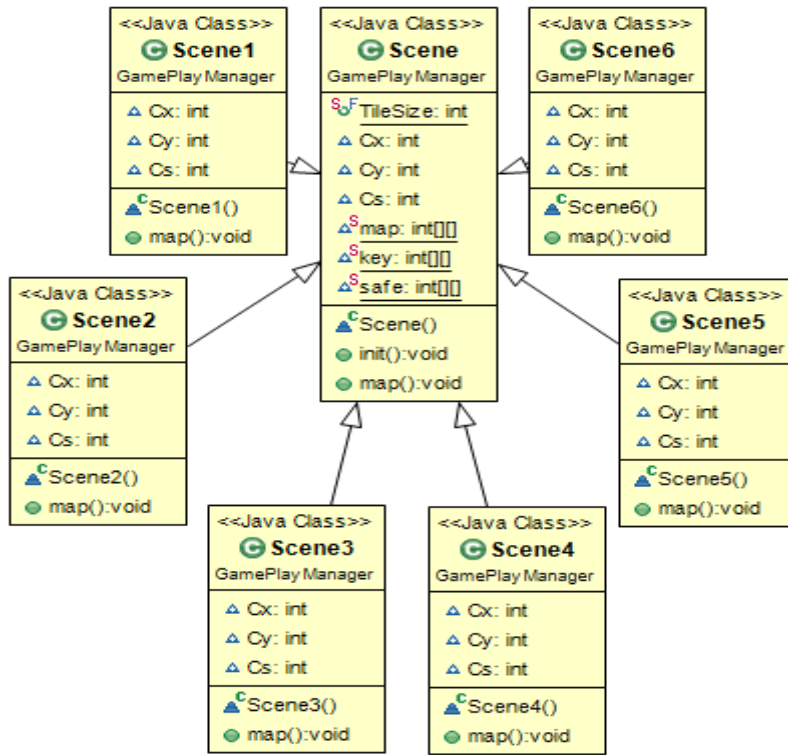


Fig. 14. 씬 클래스 다이어그램

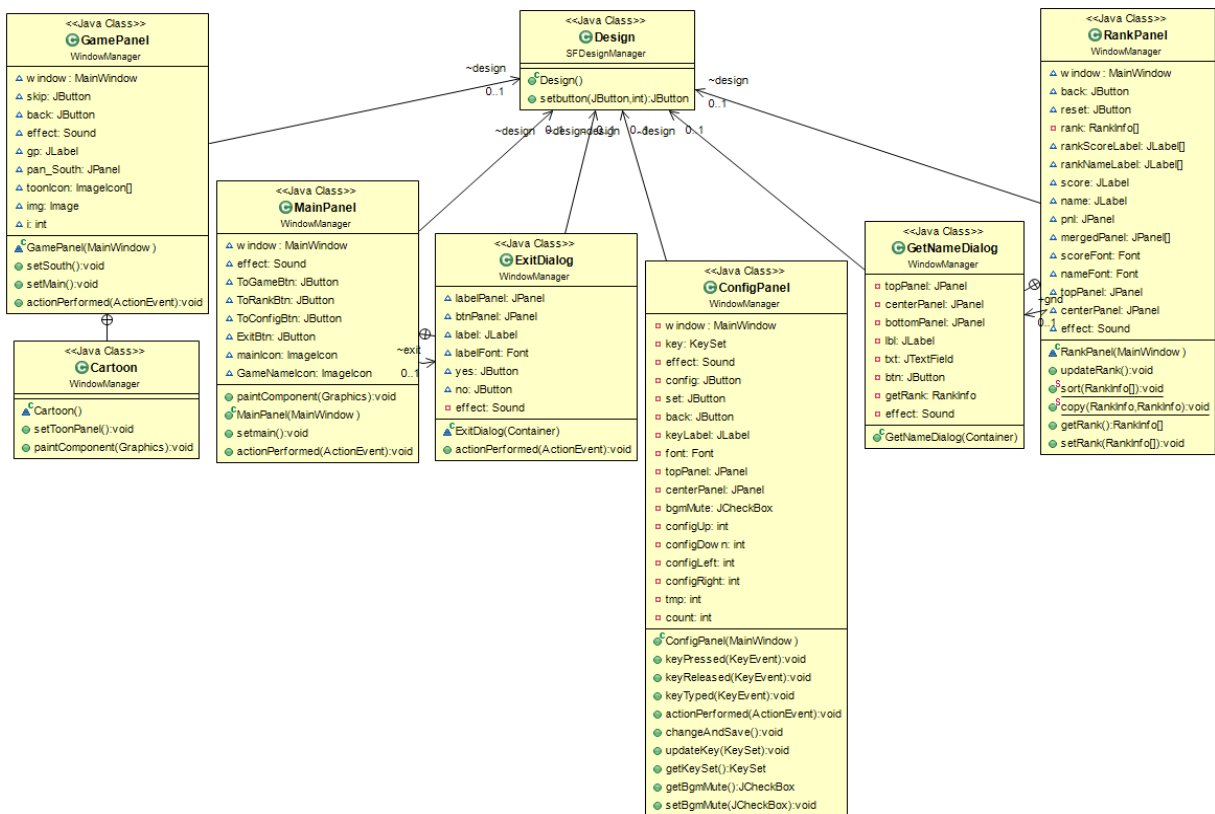


Fig. 15. 디자인 클래스와 영향을 받는 클래스 다이어그램

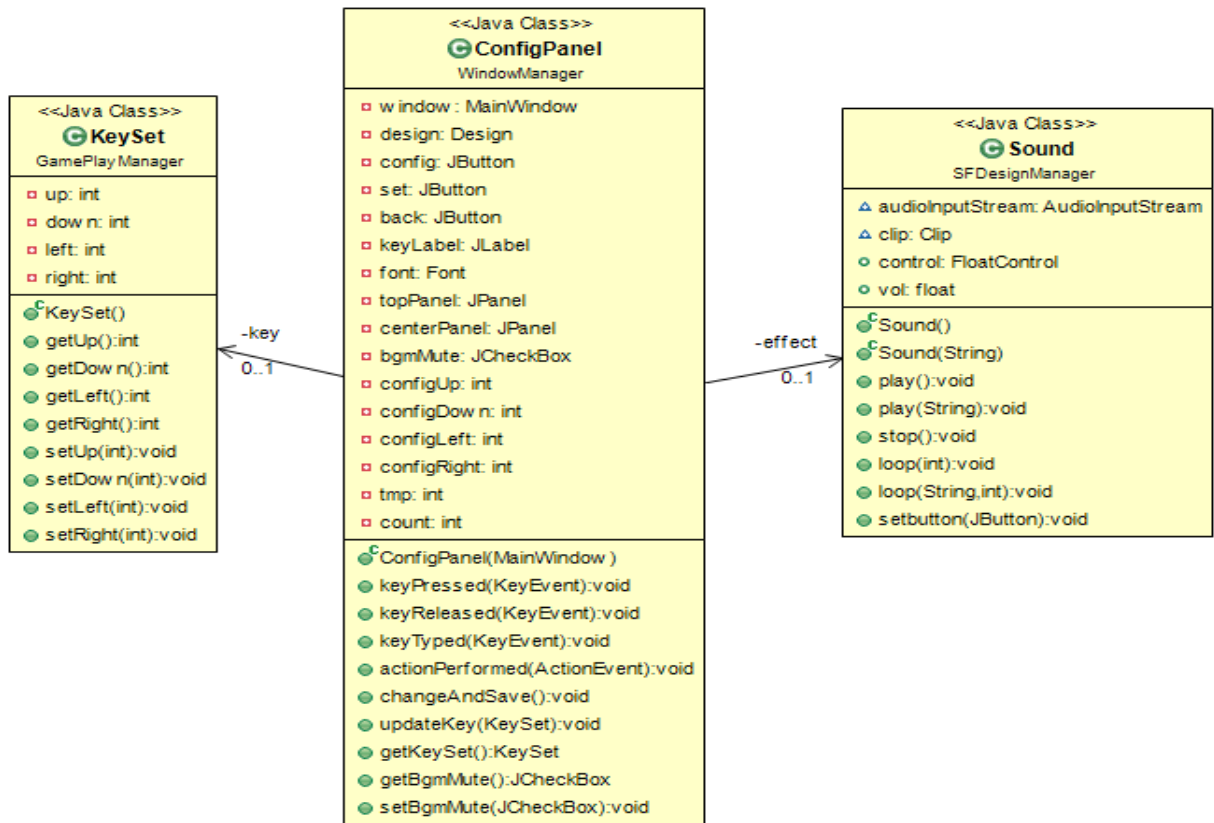


Fig. 16. 환경설정과 관련된 클래스 다이어그램

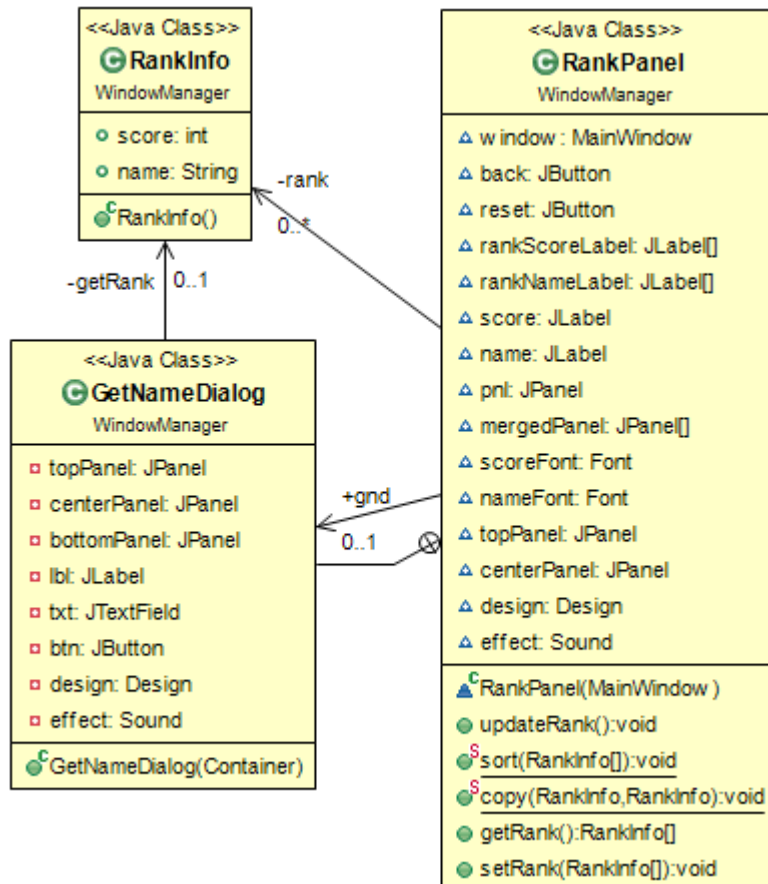


Fig. 17. 랭킹과 관련된 클래스 다이어그램

X .Screenshot of API Manual

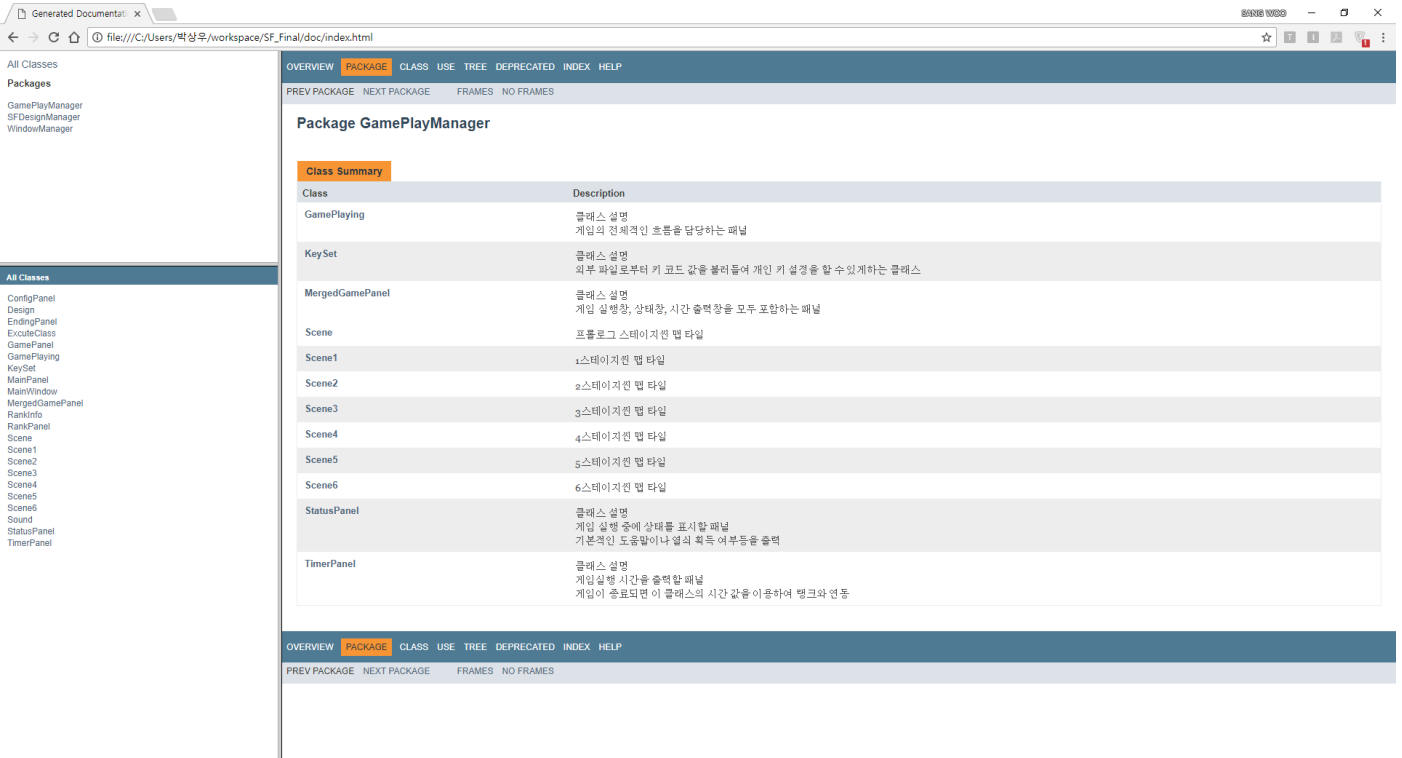


Fig. 18. 게임플레이매니저 패키지 API 매뉴얼

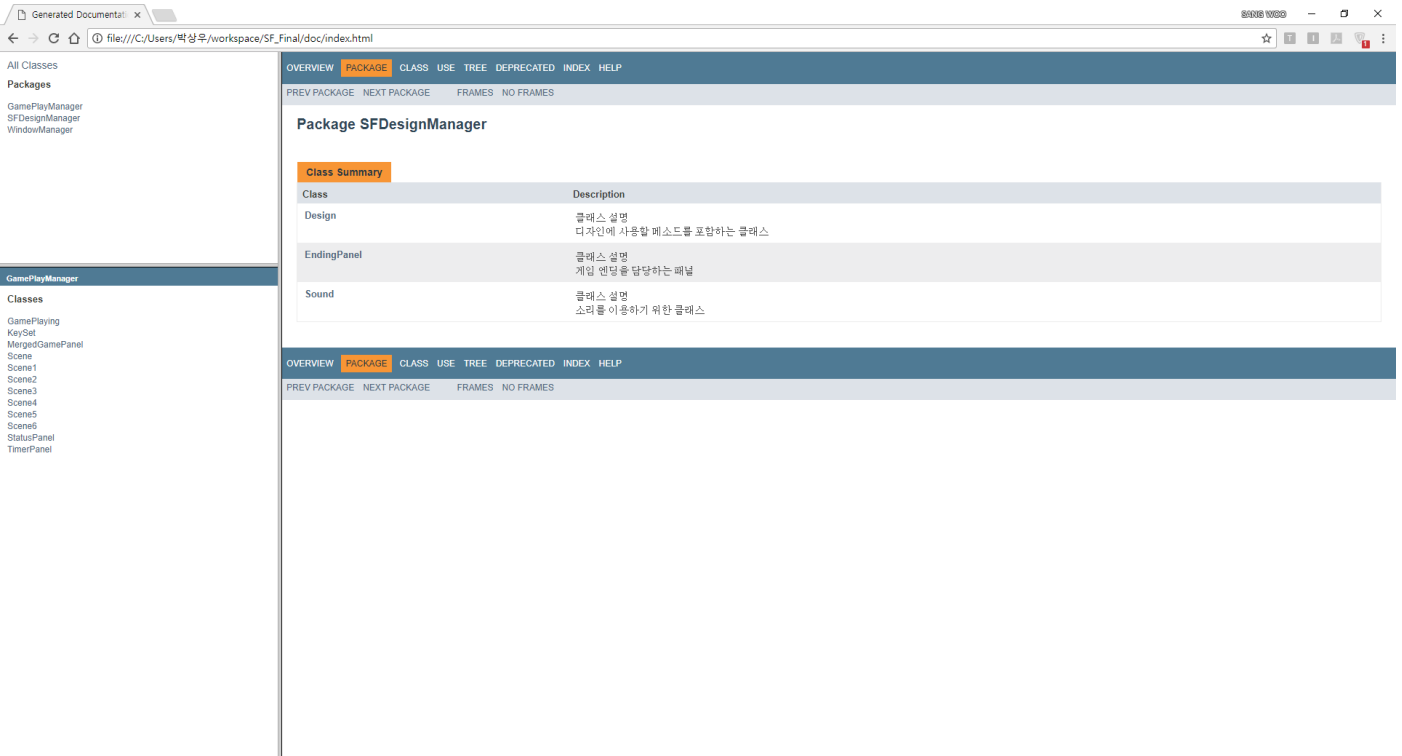


Fig. 19. 디자인매니저 패키지 API 매뉴얼

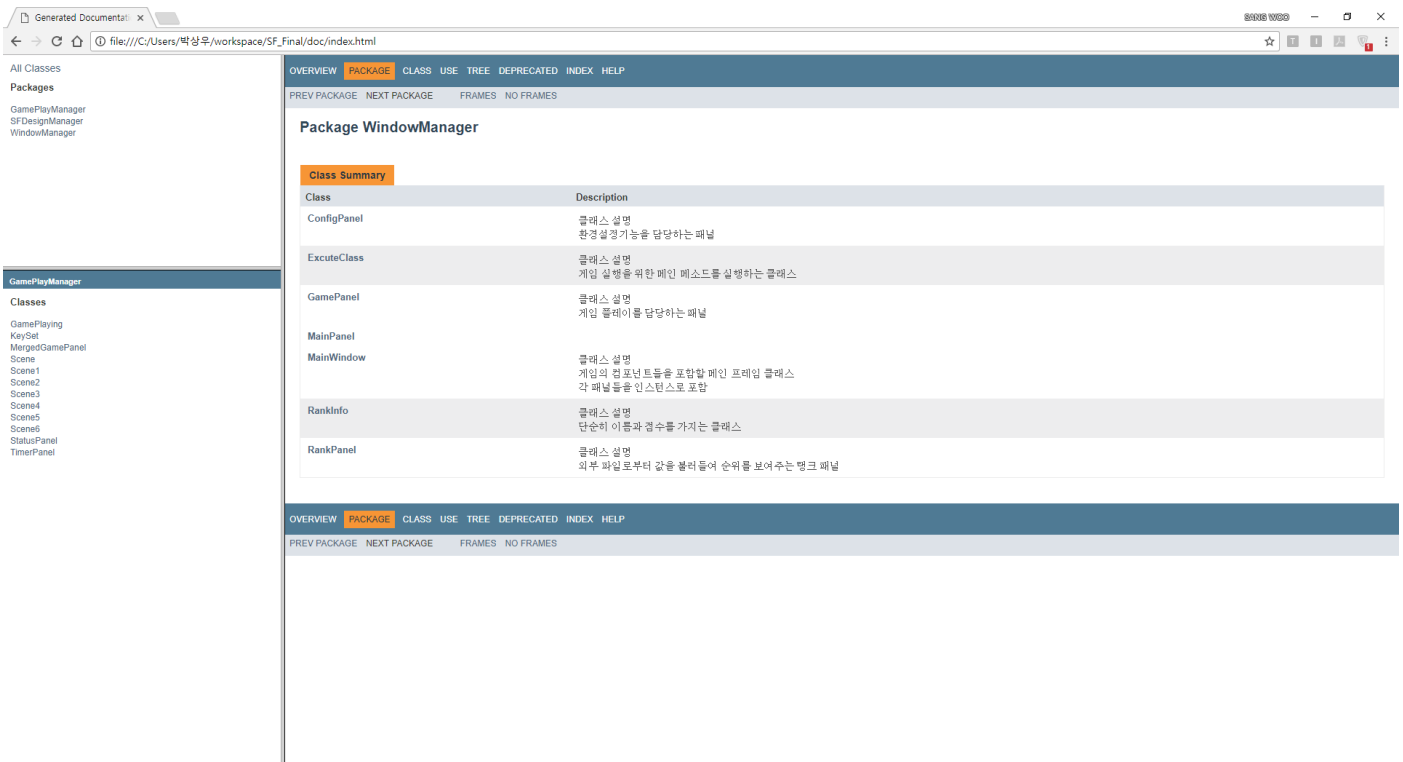


Fig. 20. 윈도우 매니저 패키지 API 메뉴얼

XI. Screenshot of Github Pages

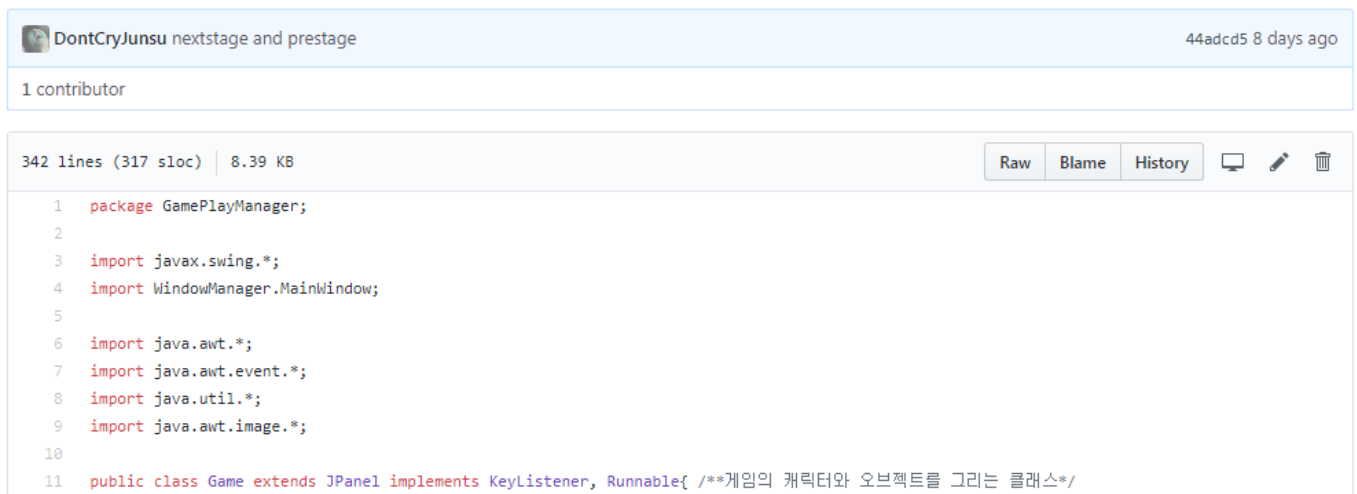


Fig. 21. Github 프로젝트속의 코드(1)

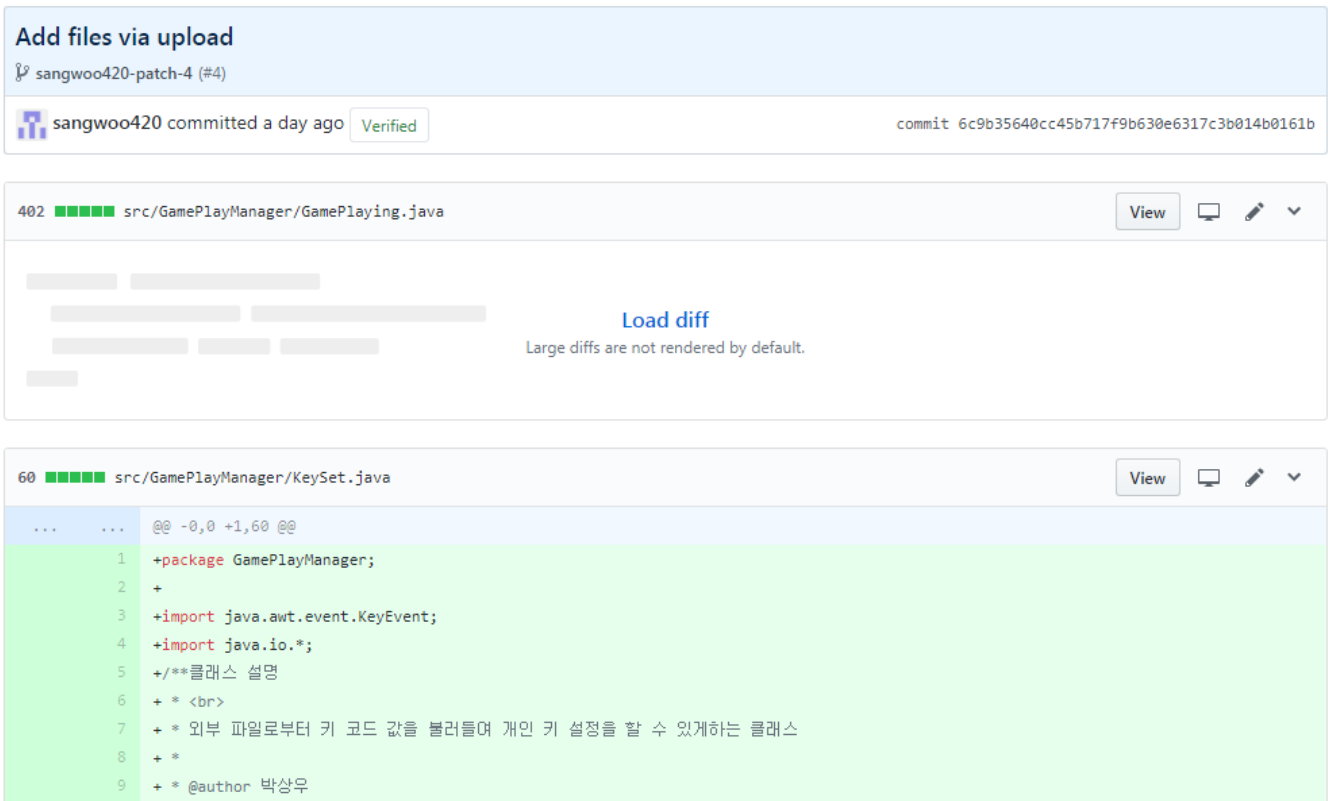


Fig. 22. Github 프로젝트속의 코드(2)

XII. Work Distribution

Chart. 1 역할분담표

역할분담	박상우	박준수	홍예진
랭킹	Red		
환경설정	Red		Blue
게임플레이		Green	
스태이지		Green	
그래픽/애니메이션			Blue
오디오	Red		Blue

<p>박준수 코드 : 1303줄</p>	<p>직접적인 게임구현과 관련된 부분을 작업하였다. 24*12크기의 배열 타일을 생성한뒤 이미지 파일을 불러와서 게임 맵의 배경을 만들었다. 각 타일위의 값에 따라 오브젝트(벽,문,열쇠)를 맵에 표시했고 방향키를 누를시 플레이어의 위치를 이동시키는 이동 메소드에 충돌판정을 합쳐서 이동을 구현하였다. 1페이지부터 7페이지까지 모든 게임 스테이지와 다음씬, 이전씬 전환을 구현하였다. 특정지역(safe zone)에서는 1칸만 움직이도록 추가적인 이동메소드를 구현하였다. GamePlayingPanel속에 그리기, 이동 클래스를 분리하지 않은 것은 아쉬운 점이다.</p>	<p>주 작업 클래스 GamePlaying, Scene, Scene1, Scene2, Scene3, Scene4, Scene5, Scene6,</p>
<p>박상우 코드 : 1319줄</p>	<p>디자인과 게임 맵을 제외한 클래스 구현 및 사운드, 게임 플레이의 부가기능 등의 기능을 중점으로 작업하였다. 외부 파일에서 키 설정을 관리할 수 있게 하였고 배경음 음소거 기능, 게임 중 일시정지 기능과 서버 메뉴 다이얼로그 작업을 하였다. 랭킹을 관리하기 위해 랭킹정보 클래스를 만들고 외부 파일에서 랭킹을 관리할 수 있게 하였다. 메인 화면에서 버튼을 눌렀을시 패널을 전환하는 메소드를 구현하였다. 게임 화면에서 상태창과 타이머 패널을 구현하였다. 아쉬운 점은 효과음도 음소거 기능을 구현하고 싶었으나, 음소거 기능이 오디오 재생이 약간 되고 나서 적용되기 때문에 짧은 효과음에는 구현할 수 없었다.</p>	<p>주 작업 클래스 KeySet, MergedGamePanel, StatusPanel, TimerPanel, ConfigPanel, ExcuteClass, RankPanel MainWindow RankInfo</p>
<p>홍예진</p>	<p>게임에 들어가는 모든 리소스 파일을 찾아 편집하거나</p>	<p>주 작업 클래스</p>

<p>코드: 672줄</p>	<p>직접 그렸으며, 이에 관련한 기능 구현을 중심으로 작업하였다. 모든 버튼에 시각적, 청각적 효과를 줄 수 있도록 이미지와 소리를 지정하는 메소드를 구현하였고, 메인화면, 게임화면, 엔딩화면 각 경우에 따라 맞는 배경음악이 재생되도록 클래스를 구성하였다. 마우스 클릭으로 다음 이미지를 볼 수 있도록 구성한 JPanel 상속 클래스를 통해 게임의 튜토리얼과 조작법을 설명하는 패널을 구성하였다. 타이머 패널에서 값을 받아 그 값에 따른 이미지를 출력하도록 하여 시각적 효과를 더해준 count-up타이머를 구성하였다. 스프레드와 JPanel의 repaint 메소드를 이용하여 텍스트가 자동적으로 출력되는 듯한 애니메이션 효과를 낸 엔딩 패널을 구성하였다. 편집은 했지만 패널이 다소 통일성 없이 난잡하게 작업된 점이 아쉽다. 프로젝트 계획을 짜던 초반, 디자인 클래스에서 인스턴스를 인수로 받아 모든 패널의 디자인을 편집하려했으나 버튼 편집 메소드밖에 하지 못 했던것도 큰 아쉬움 남는다.</p>	<p>Sound Design MainPane Ending GamePanel</p>
---------------------	---	---