프로그래밍 설계 방법론 프로젝트

리듬게임

팀 monorail

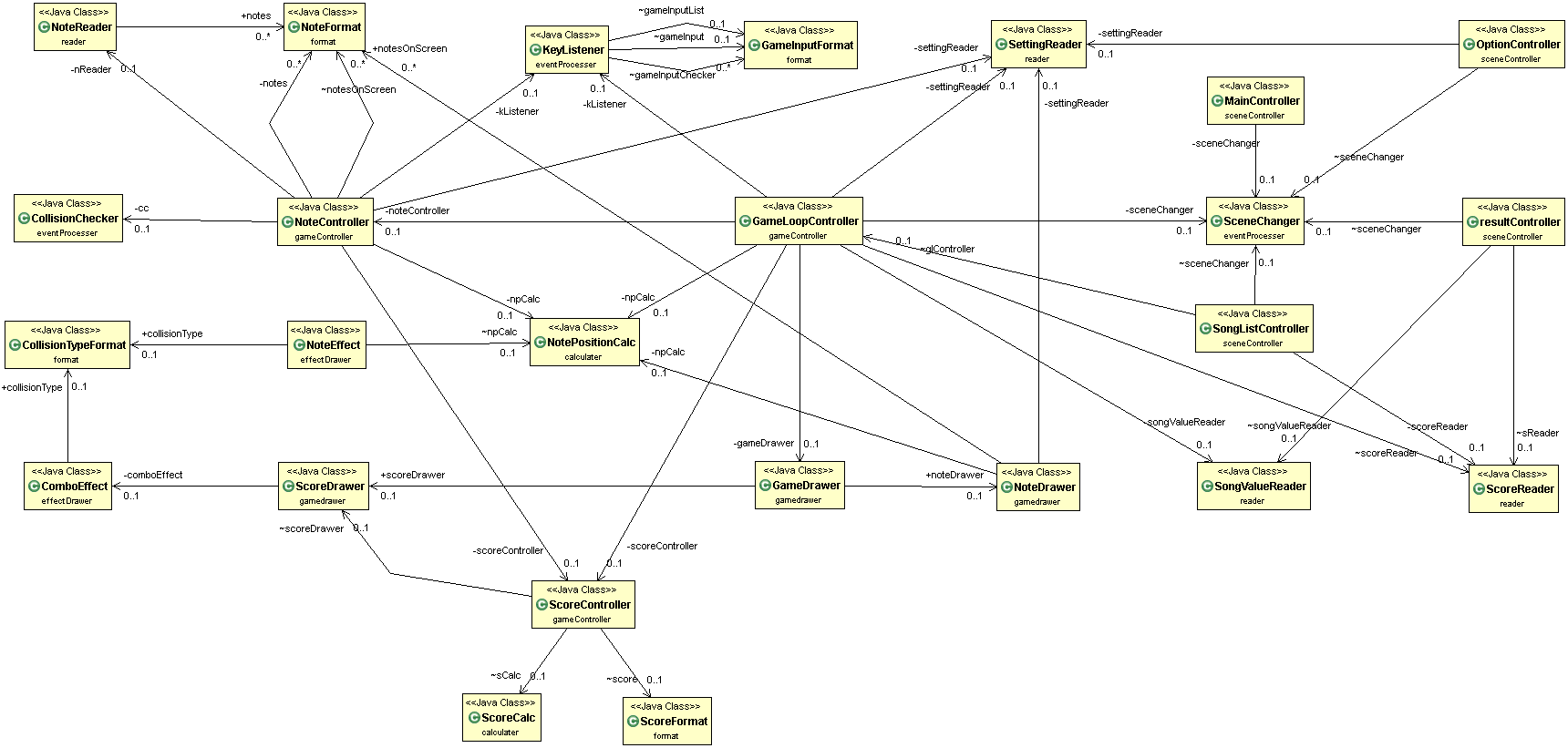
2018044457 김재훈

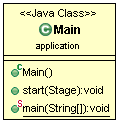
2018045150 최성환

1. 구현된 기능
   1. 다른 화면으로 넘어가는 기능
   2. 슬라이더를 이용해 값을 지정해 주는 기능
   3. 설정을 저장하고, 게임을 다시 켰을 때 이전에 저장된 설정을 불러오는 기능
   4. 점점 커지며 내려오는 노트
   5. 판정 범위에 따라 색이 달라지는 이펙트
   6. 판정에 따른 점수 합산 및 판정 확인 기능
   7. 지정된 형식에 맞춰 작성된 노트를 읽어오고, 그대로 노트를 그려주는 기능
2. 클래스 구조도 및 명세표

\* Getter, Setter, toString 및 생성자는 설명에서 제외

* 1. 전체 클래스 구조도



* 1. Main.java
  2. 

전체 프로그램을 실행시키는 클래스

Start( ) : javafx를 실행시켜주는 메소드, 부모 클래스인application에 Override함.

* 1. Model-format

CollisionTypeFormat.java

노트의 판정 타입의 형식을 지정해준 클래스

collisionTypeList : 판정 리스트

passType : 판정 리스트 중 검사할 필요가 없다는 판정

missType : 판정 리스트 중 놓친 것에 대한 판정

collisionType : 판정 타입

line : 노트가 내려올 구간

toCollisionType( ) : 판정이 올바른 형태인지 받아서 반환해주는 메소드

GameInputFormat.java

게임 내에서 유효한 입력들의 형식을 지정해준 클래스

availableKeyList : 입력 받을 키의 값을 저장한 리스트

inputValue : 입력 받은 키 값

inputLine : 입력 받은 라인

inputeTime : 입력 받은 시간

NoteFormat.java

노트의 형식을 지정해준 클래스

line : 노트가 나타날 구간

timing : 노트가 나타날 시간

ScoreFormat.java

게임 내에서의 점수의 형식을 지정해준 클래스

score : 점수

combo : 현재 콤보

maxCombo : 최대 콤보

Compare() : 점수와 콤보가 다른지 아닌지 확인하는 메소드

addScore() : 현재 점수에 추가 점수를 더해주는 메소드

addCombo() : 현재 콤보에 추가로 콤보를 더해주는 메소드

resetCombo() : 콤보를 0으로 리셋 해주는 메소드

Model-reader

NoteReader.java

저장된 노트 파일을 읽어와 지정된 노트 형식의 배열로 만들어주는 클래스

notes : 노트의 정보를 저장할 큐

sync : 노트가 내려올 타이밍을 조절하기 위한 변수

readNote( ) : 모든 노트의 정보가 저장된 텍스트 파일을 불러와 어느 라인에 어느

시간에 나타날 지 구별해 놓는 메소드

View-effectDrawer

ComboEffect.java

게임 내 화면 우측에 콤보 이펙트를 그려주는 클래스

AnimationTimer를 상속받음

effectDuration : 이펙트가 나타나는 시간

isStop : 이펙트가 지속되는 상태인지 확인하는 변수

startTime : 이펙트가 시작된 시간

currentTime : 이펙트가 시작된 시간부터 측정하기 시작하는 시간

combo : 현재 콤보

collisionType : 판정 타입

start( ) : AnimationTimer에 있는 메소드를 Override하는 메소드

stop( ) : AnimationTimer에 있는 메소드를 Override하는 메소드

handle( ) : AnimationTime가 돌아갈 때 작업을 반복하는 메소드

getVisibility( ) : 시간에 따른 투명도를 반환하는 메소드

NoteEffect.java

노트를 쳤을 때 판정에 따라 이펙트를 그려주는 클래스

effectDuration : 이펙트를 보여주는 시간

isStop : 이펙트가 지속되는지 확인하는 변수

startTime : 이펙트가 시작된 시간

currentTime : 이펙트가 시작된 시간부터 측정하기 시작하는 시간

collisionType : 판정 타입

line : 노트가 내려오는 구간

color : 이펙트의 색

xPoints : 이펙트의 표시하기 위한 3점의 x좌표

yPoints : 이펙트를 표시하기 위한 3점의 y좌표

NoteEffect( ) : 노트 위치에 이펙트를 그려주는 메소드

start( ) : AnimationTimer에 있는 메소드를 Override하는 메소드

stop( ) : AnimationTimer에 있는 메소드를 Override하는 메소드

handle( ) : AnimationTimer에 있는 메소드를 Override하는 메소드

getImgLength( ) : 이펙트가 그려지는 위치 계산

View-gameDrawer

GameDrawer.java

다른 Drawer클래스들을 불러와 전체적인 게임 화면을 그려주는 클래스

draw( ) : 게임 플레이 화면을 구성하는 요소들을 그려주는 메소드

NoteDrawer.java

화면에 나타나 있을 노트를 받아 뒤에서부터 그려주는 클래스

notesOnScreen : 화면에 나타나 있을 노트들

draw( ) : 노트를 뒤에 있는 순으로 그려주는 메소드;

ScoreDrawer.java

화면 우측에 점수와 판정을 표시해주는 클래스

comboEffect : 콤보의 변화를 보여줄 때 사용할 이펙트

drawScore( ) : 점수를 보여주는 메소드

drawCombo( ) : 콤보를 보여주는 메소드

Controller-calculator

NotePositionCalc.java

노트의 크기와 위치 변화를 계산해주는 클래스

droptime : 노트가 떨어지는 시간

multiple : 노트의 크기 변화를 줄 변수

intiSizeX : 노트의 최종 가로 길이

initSizeY : 노트의 최종 세로 길이

currentSizeX : 현재 노트의 가로 길이

currentSizeY : 현재 노트의 세로 길이

startLocationY : 노트의 시작 지점

indicesForY : 노트의 현재 y값 계산에 사용될 계수

indicesForSizeX : 노트의 가로 길이 계산에 사용될 계수

indicesForSizeY : 노트의 세로 길이 계산에 사용될 계수

isOnScreen( ) : 노트가 화면에 나와 있는지 확인하는 메소드

ScoreCalc.java

노트를 칠 때의 판정에 따라 점수를 계산해주는 클래스

noteAmount : 노트의 총 개수

maxScore : 최대 점수

sWeight : 점수 추가에 사용될 가중치

cWeight : 콤보 추가에 사용될 가중치

Controller-eventProcessor

CollisionChecker.java

노트를 칠 때 판정을 계산해주는 클래스

collisionCheckCriterion : 판정 범위

checkLine( ) : 알맞은 노트의 라인을 눌렀는지 확인하는 메소드

checkCollision( ) : 노트를 눌렀을 때 판정 타입을 결정해주는 메소드

checkIsOut( ) : 판정 범위에서 키를 했을 때의 판정을 해주는 메소드

KeyListener.java

노트를 치기 위한 키 입력을 받게 해주는 클래스

gameInputList :

gameInputChecktable :

inputList : 입력된 키 값을 저장하는 리스트

SceneChanger.java

화면을 전환 시켜주는 클래스

fadeIn( ) : 페이드 인 실행 메소드

fadeOut( ) : 페이드 아웃 및 하면 전환 메소드를 실행하는 메소드

loadGame( ) : 게임 플레이 화면으로 전환하는 메소드

load( ) : 화면을 전환하는 메소든

Controller-gameController

GameLoopController.java

전체적인 컨트롤러를 불러와서 게임 화면을 총괄하는 컨트롤러 클래스

gameloop : 게임의 상태를 업데이트 해주고 그려주는 루프를 가진 변수

startNanoTime : 게임이 시작된 시간

delayedTime : 게임이 지연 될 시간

start( ) : 게임이 시작할 때 화면 전환 및 시간을 측정하기 시작하는 메소드

handle( ) : 게임 이벤트를 처리하고 화면에 그려주는 루프 메소드

NoteController.java

노트가 나타날 시간을 불러와 알맞은 시간에 출력을 시켜주는 명령을 내리는 클래스

notes : 노트들의 정보가 들어있는 큐

noteAmount : 노트의 총 개수

notesOnScreen : 화면에 나와 있는 노트 리스트

startTime : 게임이 시작된 시간

addDrawableNote( ) : 화면에 나와있을 노트를 추가하는 메소드

checkCollision( ) : 노트 판정을 받아서 점수 및 콤보 에 변화를 주고 노트를 지우는

메소드

deleteOutNotes( ) : 놓친 노트를 화면에서 지우는 메소드

clearInput( ) : 입력 받은 키 값을 지우는 메소드

update( ) : 노트에 대한 정보를 업데이트 하는 메소드

ScoreController.java

점수와 콤보 계산을 받아 업데이트 시켜주는 클래스

availableTypeForCombo : 노트 판정 중 콤보가 계속되는 판정 목록

availableTypeForScore : 노트 판정 중 점수가 추가되는 판정 목록

updateCombo( ) : 콤보를 추가하거나 리셋하는 메소드

updateScore( ) : 점수를 추가하는 메소드

update( ) : 콤보와 점수를 업데이트 하는 메소드

Controller-sceneController

MainController.java

메인 화면에서 입력을 받아 동작을 시켜주는 클래스

Songs : 곡 목록 버튼

Option : 설정 버튼

initialize( ) : 화면을 불러왔을 때 실행하는 메소드

SongPressed( ) : 곡 목록 버튼을 눌렀을 때 곡 목록 화면으로 넘어가는 메소드

OptionPressed( ) : 설정 버튼을 눌렀을 때 설정 화면으로 넘어가는 메소드

OptionController.java

설정 화면에서 입력을 받아 동작을 시켜주는 클래스

VolumeSlide : 음량을 조절하는 슬라이더

SyncSlide : 노트가 내려올 타이밍을 조절하는 슬라이더

SpeedSlide : 노트의 속도를 조절하는 슬라이더

VolumeValue : 설정된 음량을 보여주는 변수

SyncValue : 설정된 노트의 타이밍을 보여주는 변수

SpeedValue : 설정된 노트의 속도를 보여주는 변수

BacktoMain : 초기 화면으로 돌아가는 버튼

initialize( ) : 화면을 불러왔을 때 실행하는 메소드

BackToMain( ) : 초기 화면으로 돌아가는 메소드

changed( ) : 슬라이더를 이용해 변경된 설정 값을 변수에 저장하는 메소드

SongListController.java

곡 선택 화면에서 입력을 받아 동작을 시켜주는 클래스

Song : 노래 실행 버튼

initialize( ) : 화면을 불러왔을 때 실행하는 메소드

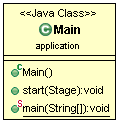
SelectSong( ) : 게임을 플레이 하는 메소드

BackToMain( ) : 초기 화면으로 돌아가는 메소드

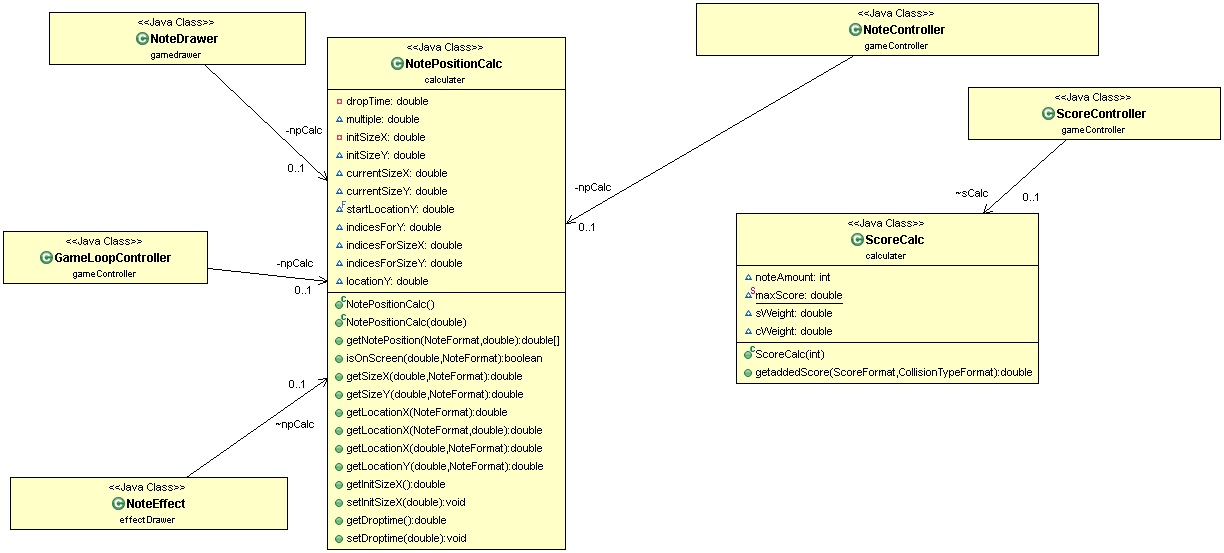
클래스 구조도

전체 구조도

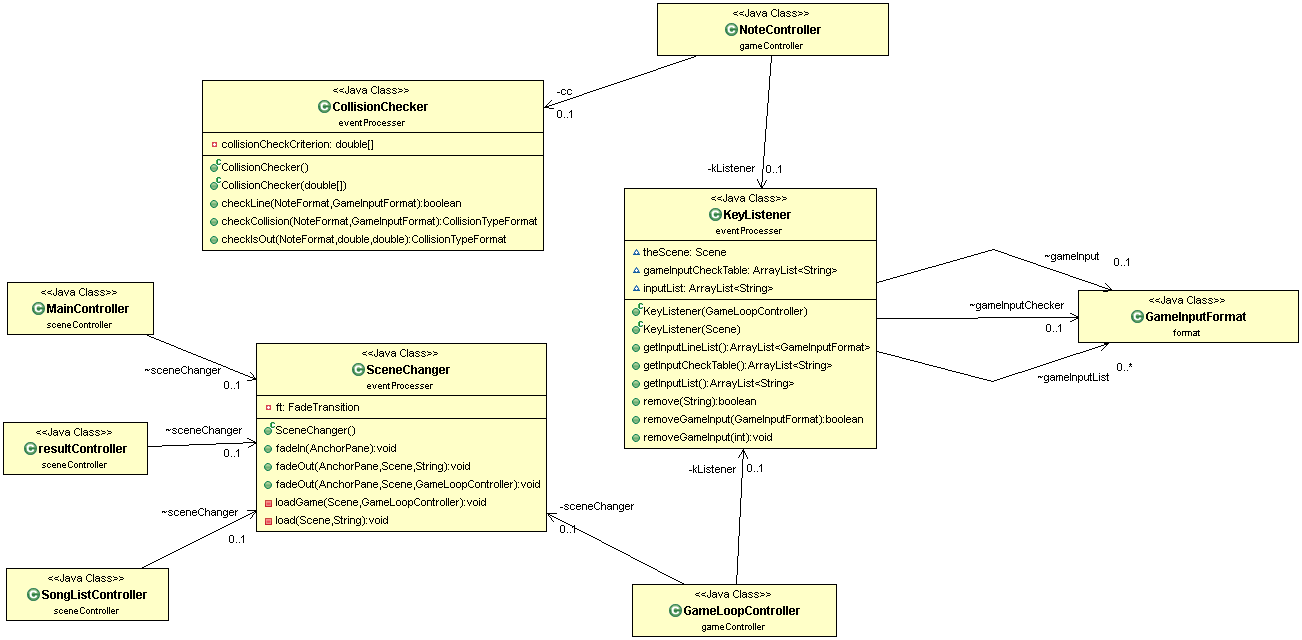
Application



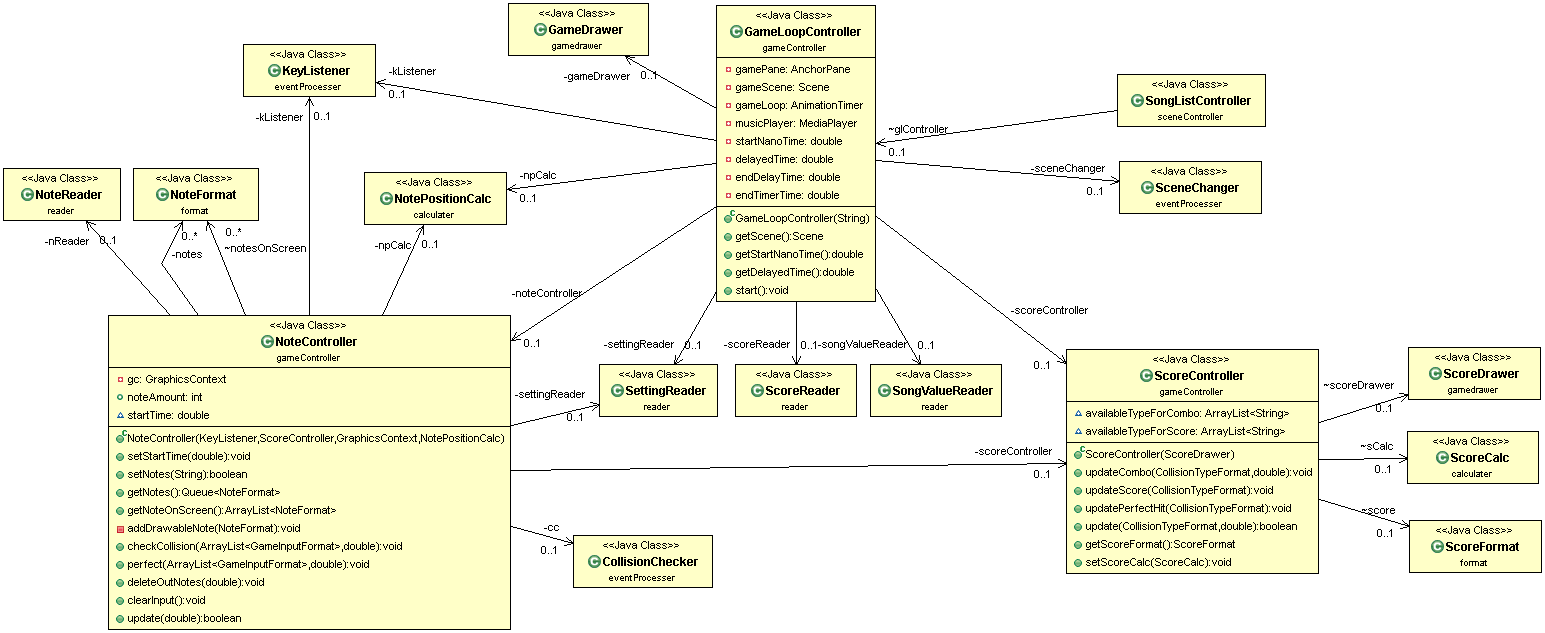
controller – calculator



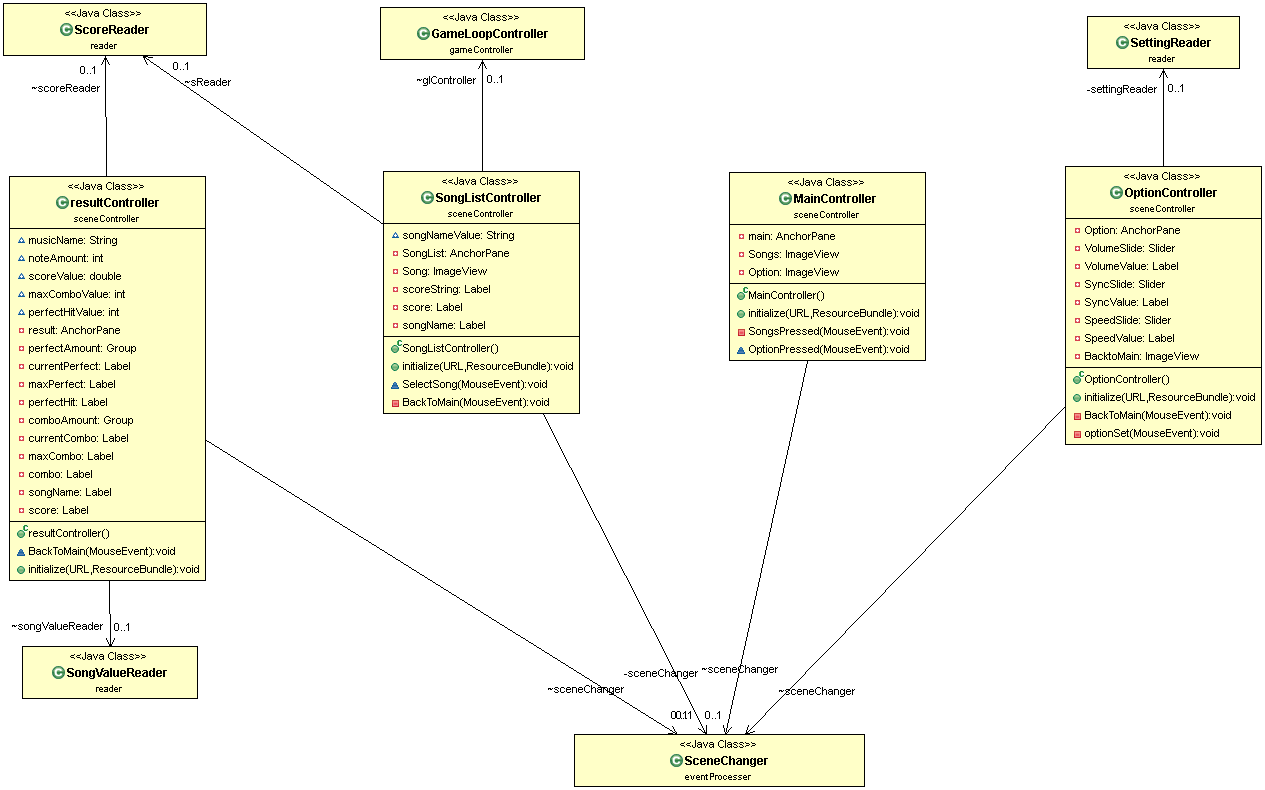
controller – eventProcessor



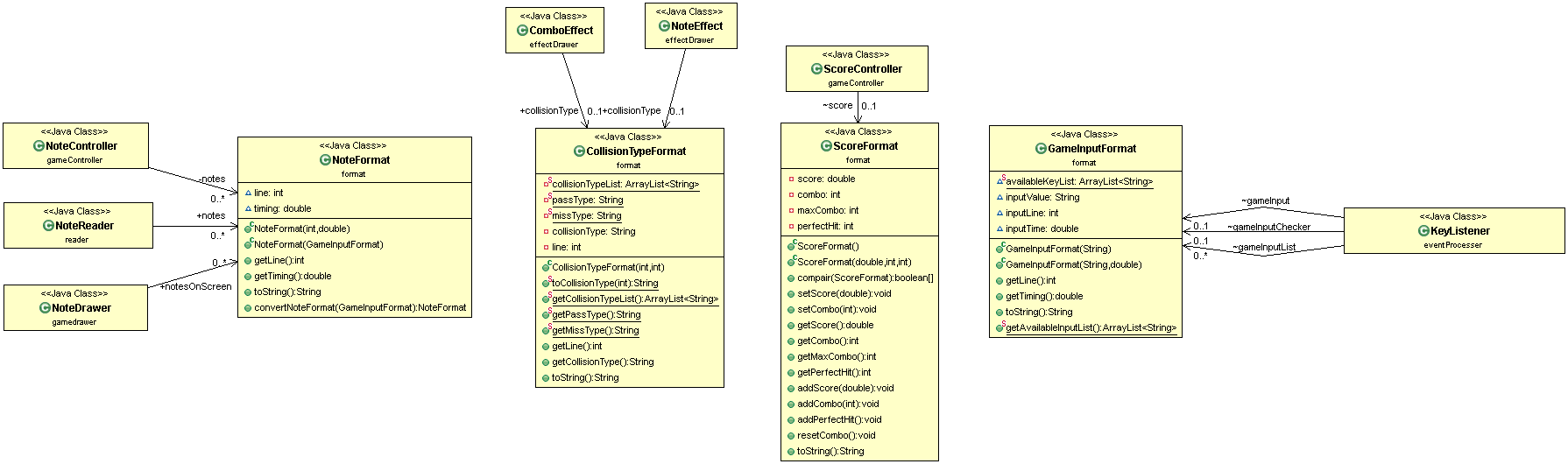
controller – gamController



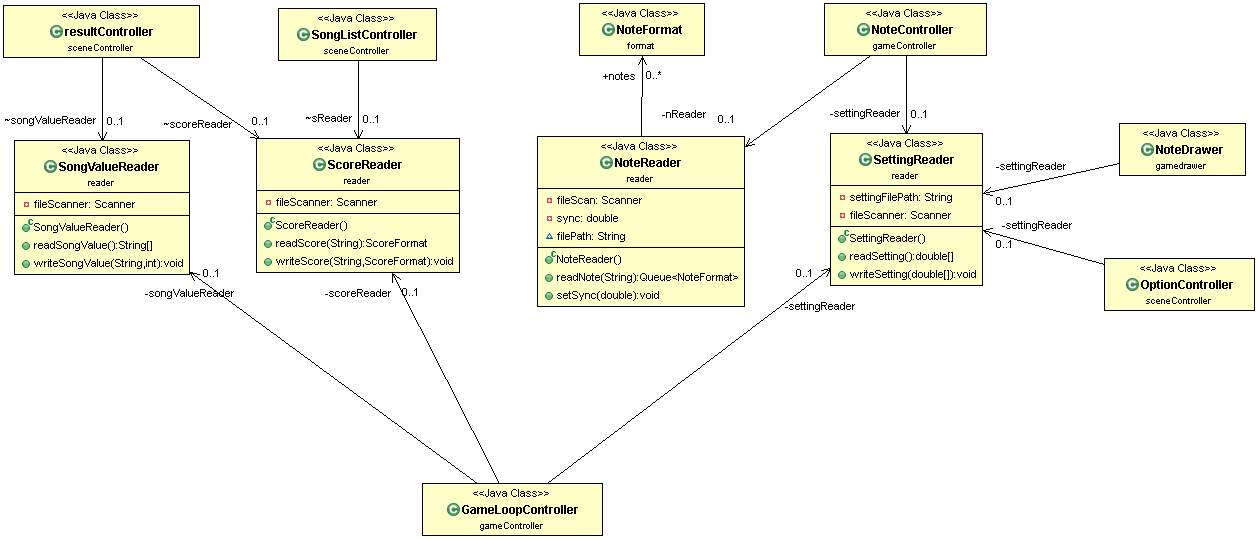
Controller – sceneController



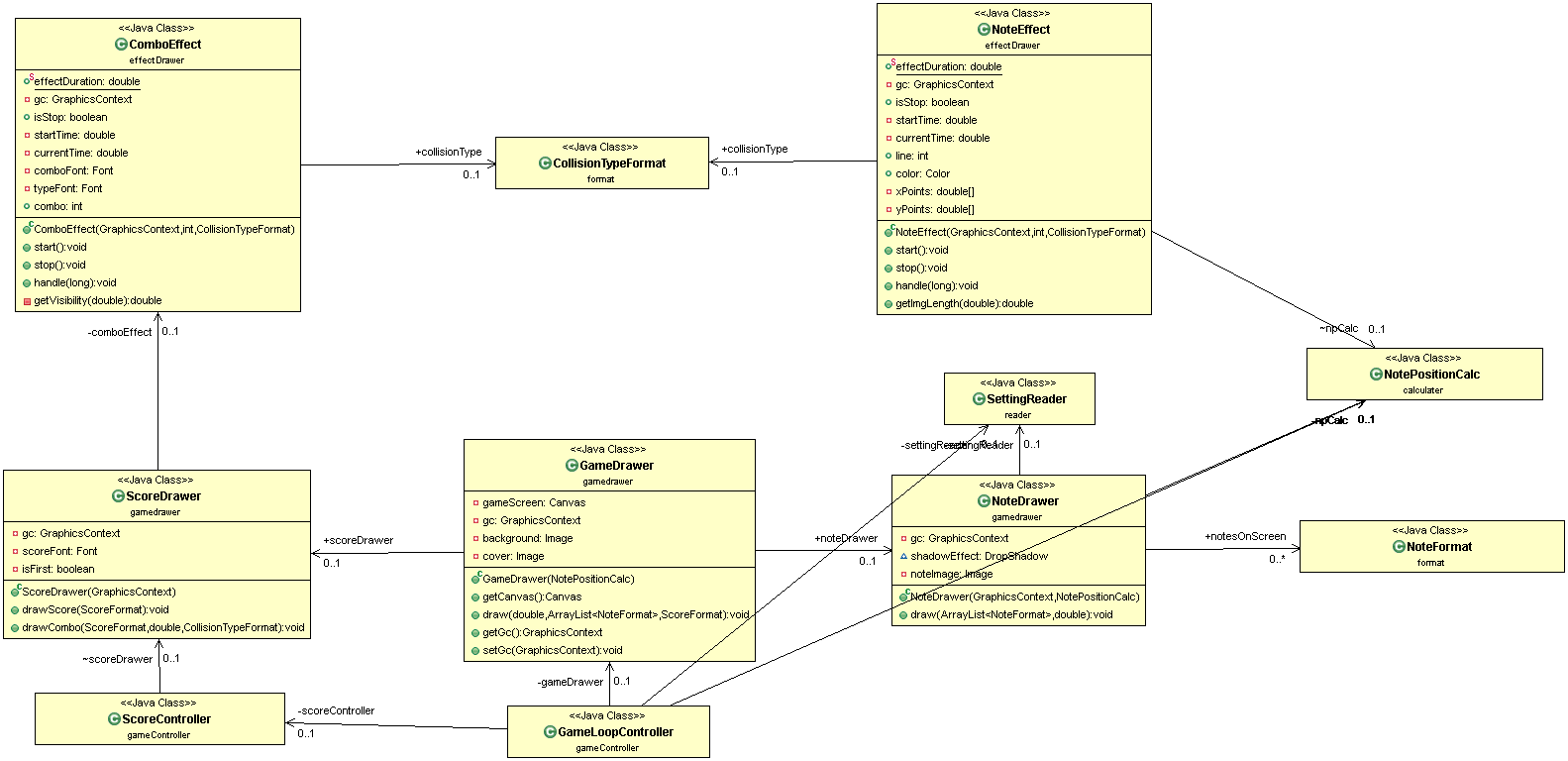
model – format



model – reader



View – effectDrawer & gamedrawer



추상 클래스 및 인터페이스 사용 내역

예외 케이스 커버 목록