

# 프로그래밍 설계 방법론

## 프로젝트 리듬게임

!Deemo

팀 monorail

2018044457김재훈

2018045150 최성환

# 목차

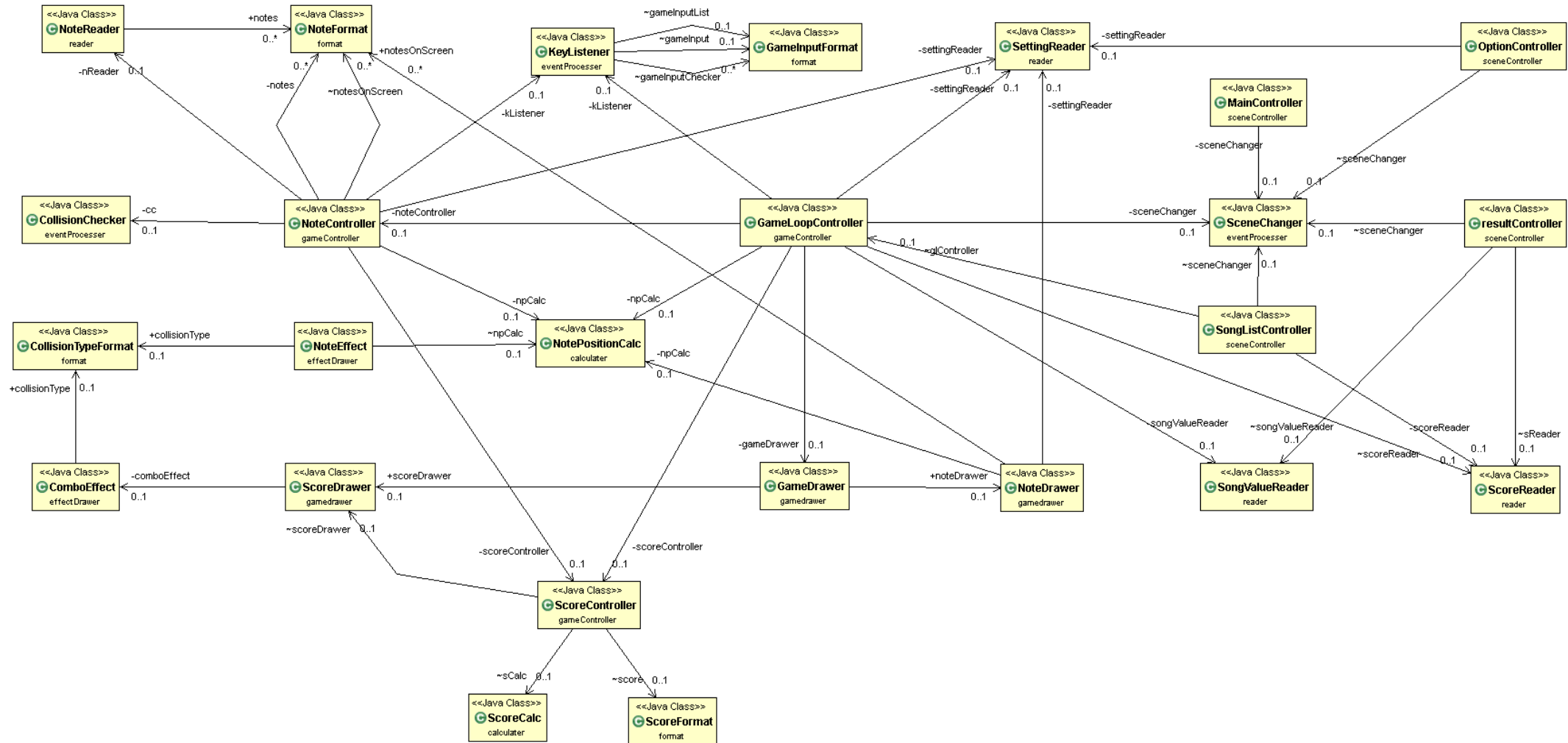
1. 구현된 기능 설명
2. 코드 설명
3. 예외처리 항목
4. 역할 분담

# 1. 구현된 기능

- 1) 게임 내에서 중요하게 쓰이는 옵션값(볼륨, 싱크, 노트 스피드 등)을 자유로이 수정하고, 저장하는 기능
- 2) 노트가 일직선으로 내려오는 것이 아닌, 점점 커지며 입체적으로 다가오는 기능
- 3) 지정된 형식의 노트파일을 읽어 그대로 화면에 출력해주는 기능
- 4) 판정에 따라 다른 색의 이펙트가 나오는 기능
- 5) 판정과 콤보에 따른 점수 측정 및 최고점수 저장 기능

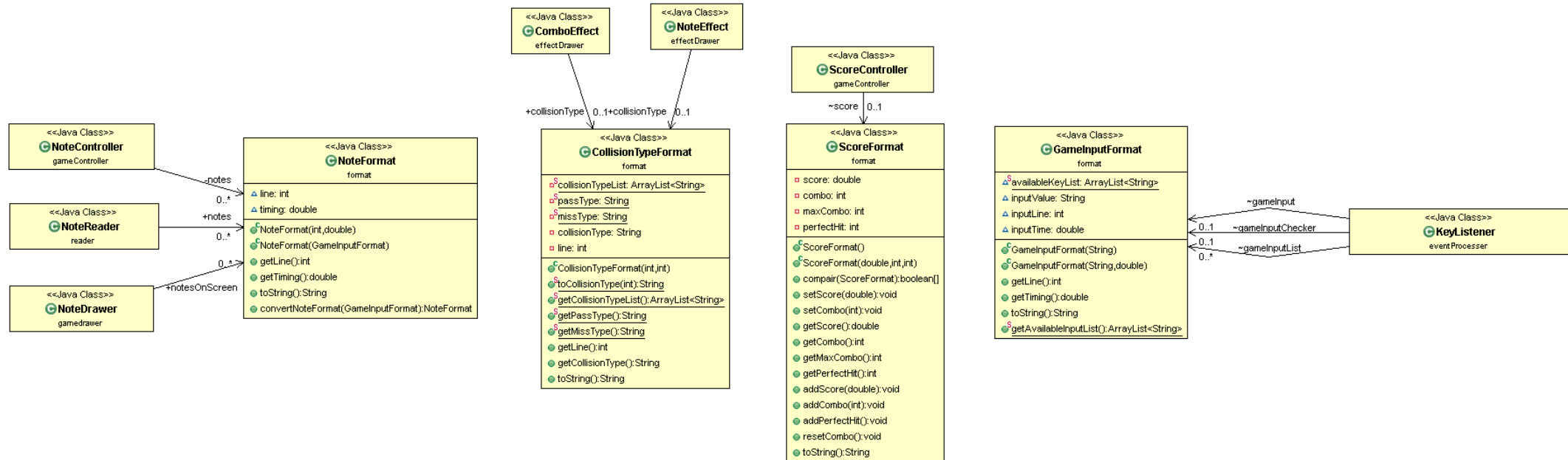
## 2. 코드 설명-클래스 구조도

### 1) 전체 클래스 구조도



## 2. 코드설명-클래스 구조도

### 2) Model\_formatPackage



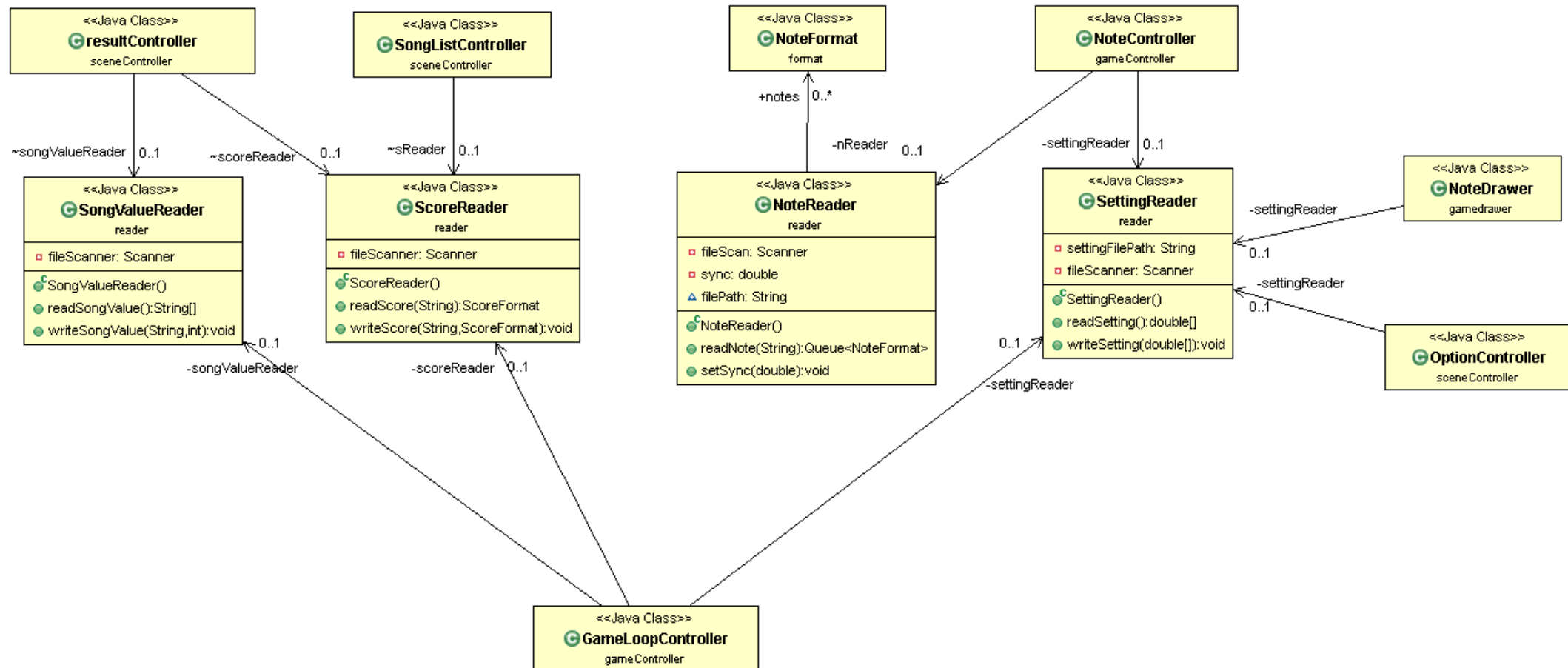
## 2. 코드설명-클래스 명세표

### 2) Model\_formatPackage

- noteFormat.java : 노트의 형식이 지정된 클래스
  - line : 노트가 나타날 라인
  - timing : 노트의 판정 시간
- collisionTypeFormat.java : 노트의 판정 타입 형식이 지정된 클래스
  - collisionTypeList : 노트의 판정 타입들이 저장된 리스트
  - collisionType : 확인중인 노트의 판정 타입
  - line : 확인중인 노트가 나타난 라인
- scoreFormat.java : 점수 정보를 가지는 클래스
  - score : 점수
  - combo : 콤보
  - maxCombo : 최대 콤보
  - perfectHit : perfect 판정을 받은 노트의 수
- gameInputFormat.java : 입력받은 키에 매칭되는 라인과 입력 타이밍을 가지는 클래스
  - availableKeyList : 게임 노트에 매핑되는 키값 리스트
  - line : 입력받은 키에 매핑되는 라인
  - timing : 키 입력이 들어온 시간

## 2. 코드설명-클래스 구조도

### 3) Model\_readerPackage



## 2. 코드설명-클래스 명세표

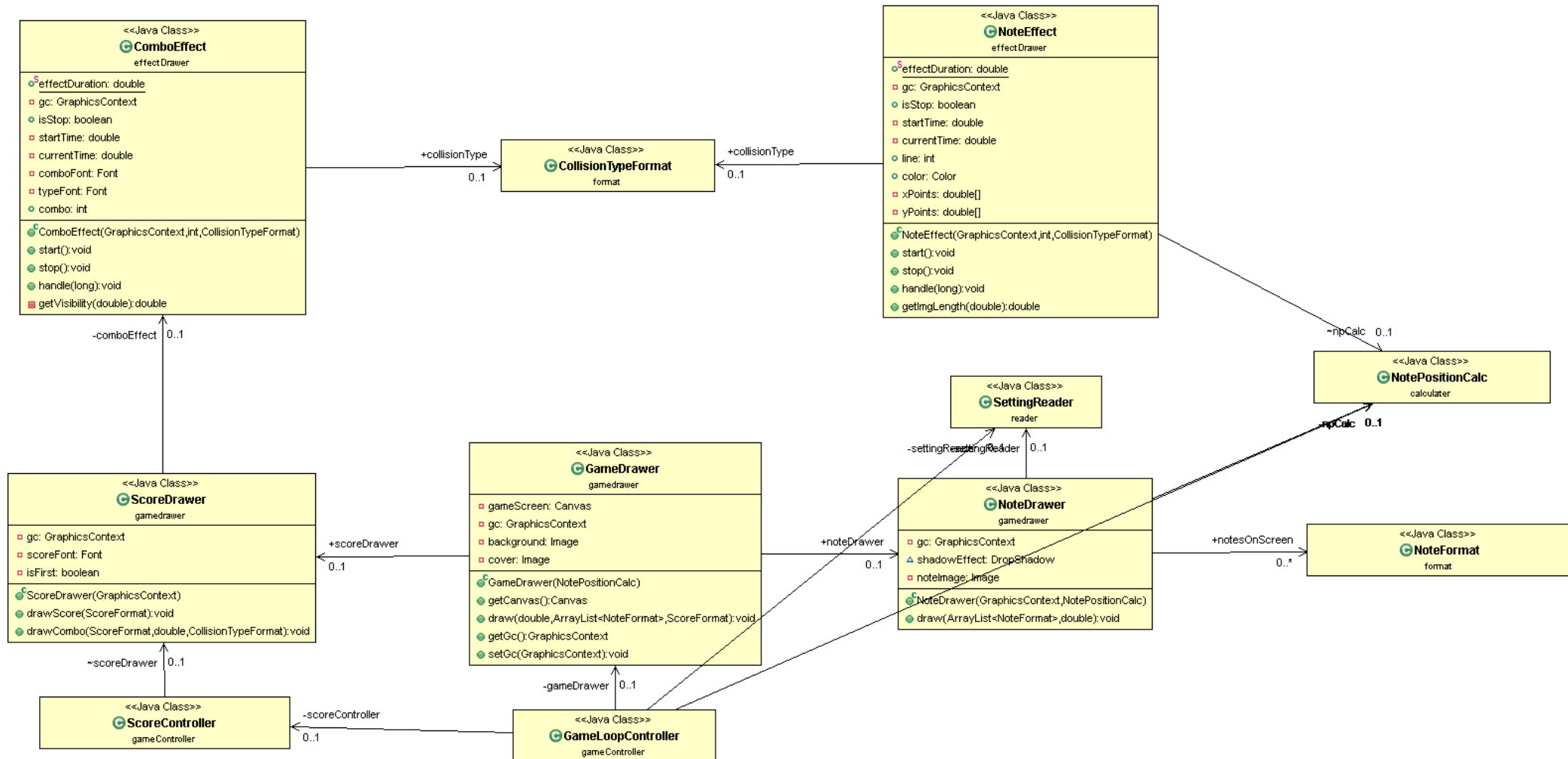
### 3) Model\_readerPackage

- SongValue.java : 노트의 형식이 지정된 클래스
  - line : 노트가 나타날 라인
  - timing : 노트의 판정 시간
- collisionTypeFormat.java : 노트의 판정 타입 형식이 지정된 클래스
  - collisionTypeList : 노트의 판정 타입들이 저장된 리스트
  - collisionType : 확인중인 노트의 판정 타입
  - line : 확인중인 노트가 나타난 라인
- scoreFormat.java : 점수 정보를 가지는 클래스
  - score : 점수
  - combo : 콤보
  - maxCombo : 최대 콤보
  - perfectHit : perfect 판정을 받은 노트의 수
- gameInputFormat.java : 입력받은 키에 매칭되는 라인과 입력 타이밍을 가지는 클래스
  - availableKeyList : 게임 노트에 매핑되는 키값 리스트
  - line : 입력받은 키에 매핑되는 라인
  - timing : 키 입력이 들어온 시간



## 2. 코드설명-클래스 구조도

### 4) View\_effectDrawerPackage, View\_gameDrawerPackage



## 2. 코드설명-클래스 명세표

### 4) View\_effectDrawerPackage

- ComboEffect.java : 게임 내 화면 우측에 콤보 이펙트를 그리는 클래스,  
AnimationTimer 추상 클래스를 상속 받음
  - 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
  - getVisibility( ) : 시간에 따른 투명도를 반환하는 메소드

## 2. 코드설명-클래스 명세표

### 4) View\_effectDrawerPackage

- NoteEffect.java : 노트를 쳤을 때 판정에 따라 이펙트를 그리는 클래스,  
AnimationTimer 추상 클래스를 상속 받음
  - 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
  - getImgLength( ) : 이펙트가 그려지는 위치 계산

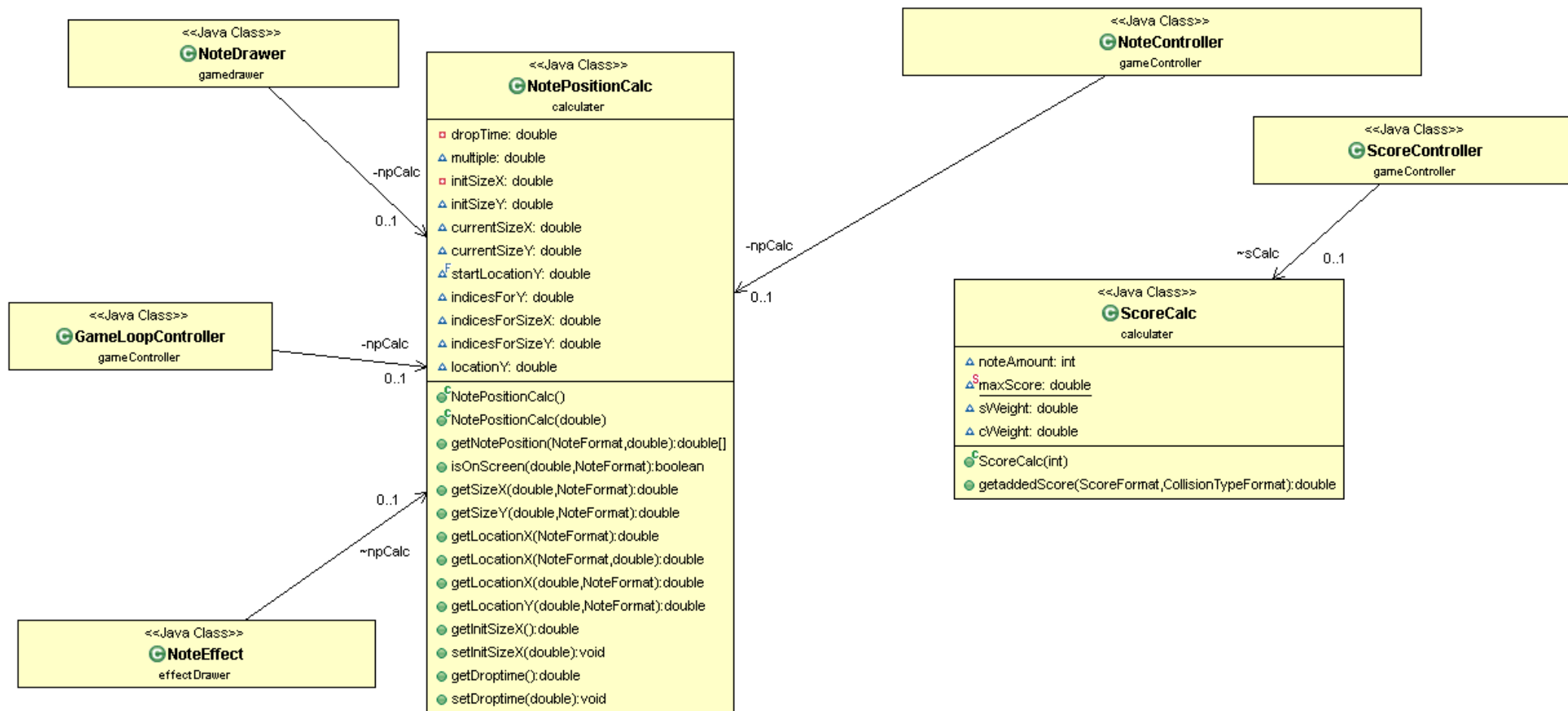
## 2. 코드설명-클래스 명세표

### 4) View\_gameDrawerPackage

- GameDrawer.java : 다른 Drawer클래스들을 불러와 전체적인 게임 화면을 그리는 클래스
  - draw( ) : 게임 플레이 화면을 구성하는 요소들을 그려주는 메소드
- NoteDrawer.java : 화면에 나타나 있을 노트를 받아 뒤에서부터 그리는 클래스
  - draw( ) : 노트를 뒤에 있는 순으로 그려주는 메소드
- ScoreDrawer.java : 화면 우측에 점수와 판정을 표시하는 클래스
  - comboEffect : 콤보의 변화를 보여줄 때 사용할 이펙트
  - drawScore( ) : 점수를 보여주는 메소드
  - drawCombo( ) : 콤보를 보여주는 메소드

## 2. 코드설명-클래스 구조도

### 5) Controller\_calculatorPackage



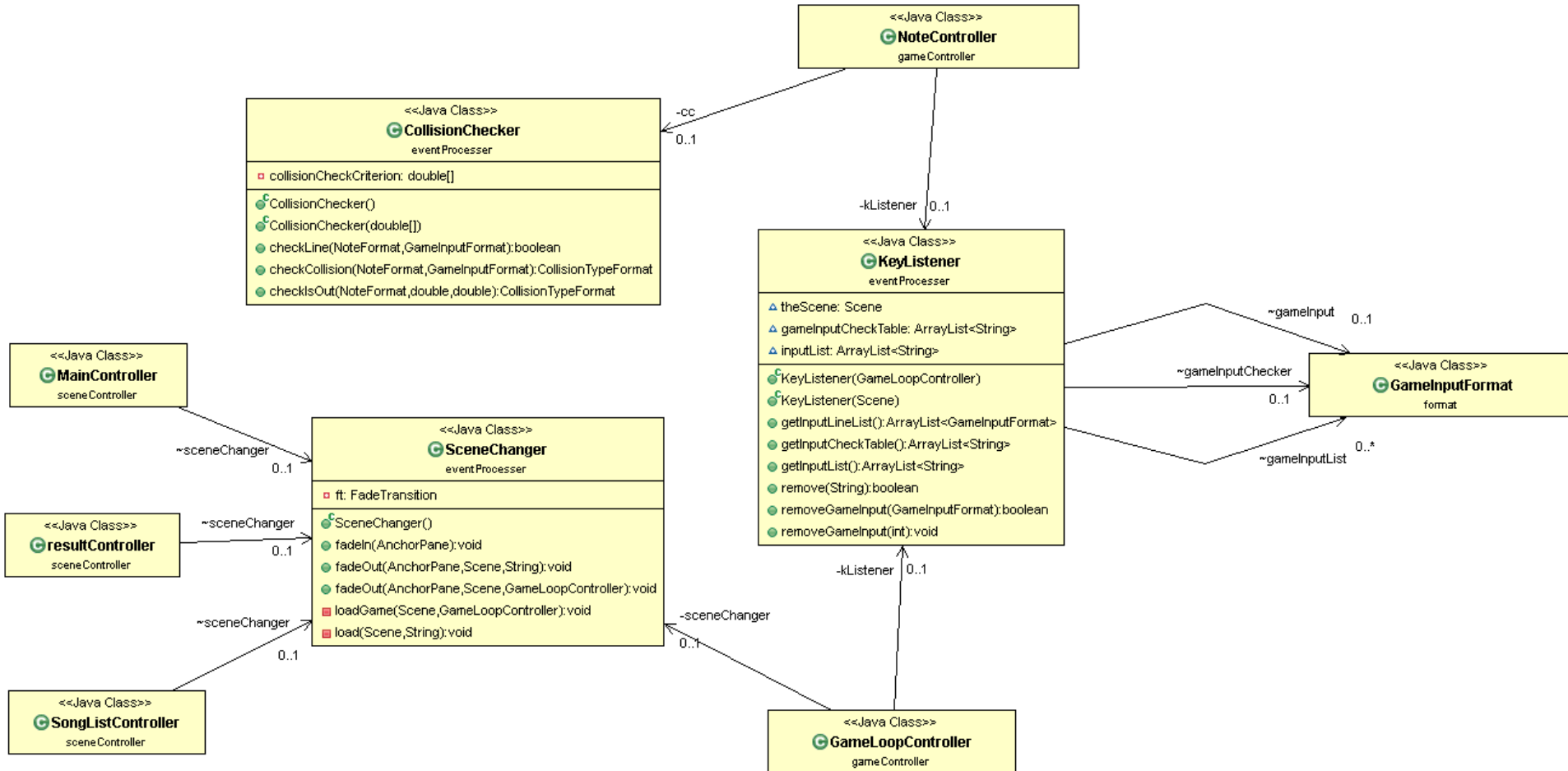
## 2. 코드설명-클래스 명세표

### 5) Controller\_calculatorPackage

- NotePositionCalc.java : 노트의 크기와 위치 변화를 계산하는 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 계산하는 클래스
  - noteAmount : 노트의 총 개수
  - maxScore : 최대 점수
  - sWeight : 점수 추가에 사용될 가중치
  - cWeight : 콤보 추가에 사용될 가중치

## 2. 코드설명-클래스 구조도

### 6) Controller\_eventProcessorPackage



## 2. 코드설명-클래스 명세표

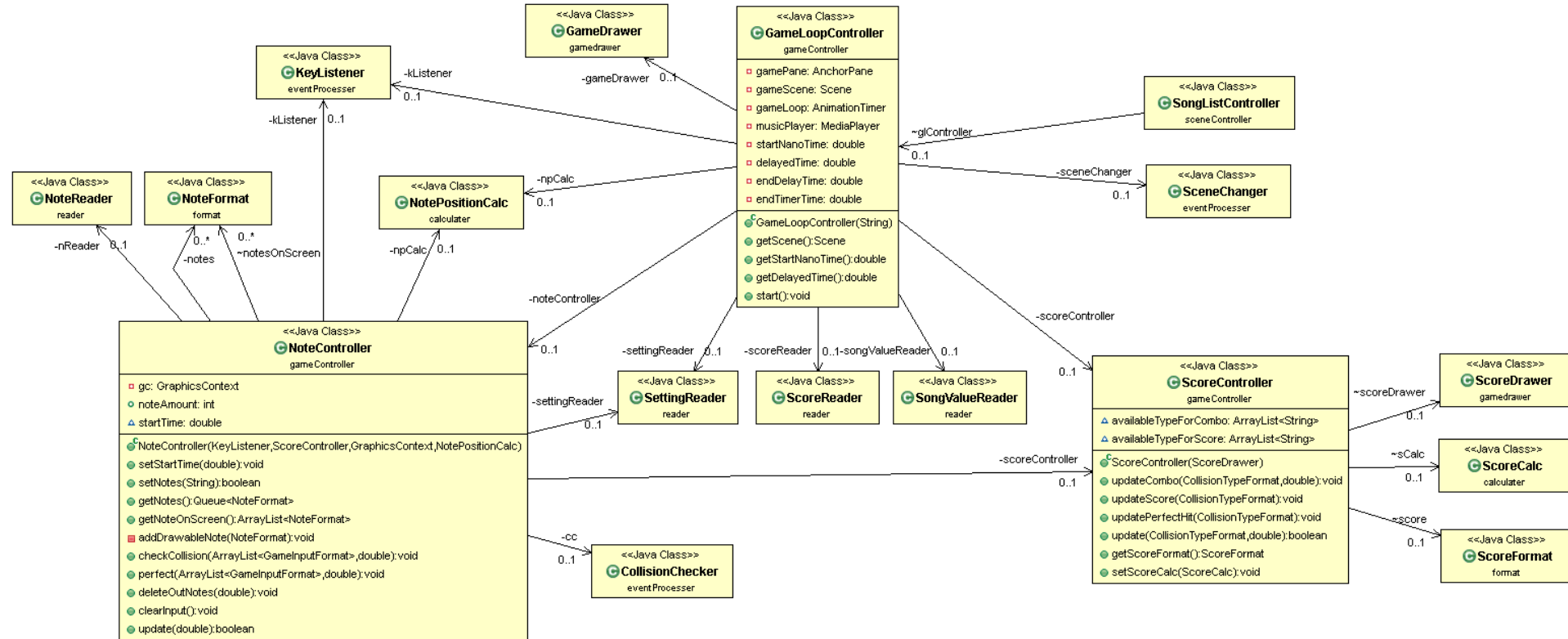
### 6) Controller\_eventProcessorPackage

- CollisionChecker.java : 노트를 칠 때 판정을 계산하는 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 받는 클래스
  - gameInputList : 게임에 관련된 입력이 저장되는 리스트
  - gameInputChecktable : 입력이 저장된 리스트에 중복 저장 하지 않기 위한 리스트
  - inputList : 입력된 키 값을 저장하는 리스트
- SceneChanger.java : 화면을 전환하는 클래스
  - fadeIn( ) : 페이드 인 실행 메소드
  - fadeOut( ) : 페이드 아웃 및 하면 전환 메소드를 실행하는 메소드
  - loadGame( ) : 게임 플레이 화면으로 전환하는 메소드
  - load( ) : 화면을 전환하는 메소드



## 2. 코드설명-클래스 구조도

### 7) Controller\_gameControllerPackage



## 2. 코드설명-클래스 명세표

### 7) Controller\_gameControllerPackage

- GameLoopController.java : 전체적인 컨트롤러를 불러와서 게임 화면을 총괄하는 클래스
  - gameloop : 게임의 상태를 업데이트 해주고 그려주는 루프를 가진 변수
  - startNanoTime : 게임이 시작된 시간
  - delayedTime : 게임이 지연 될 시간
- NoteController.java : 노트가 나타날 시간을 불러와 알맞은 시간에 출력하는 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

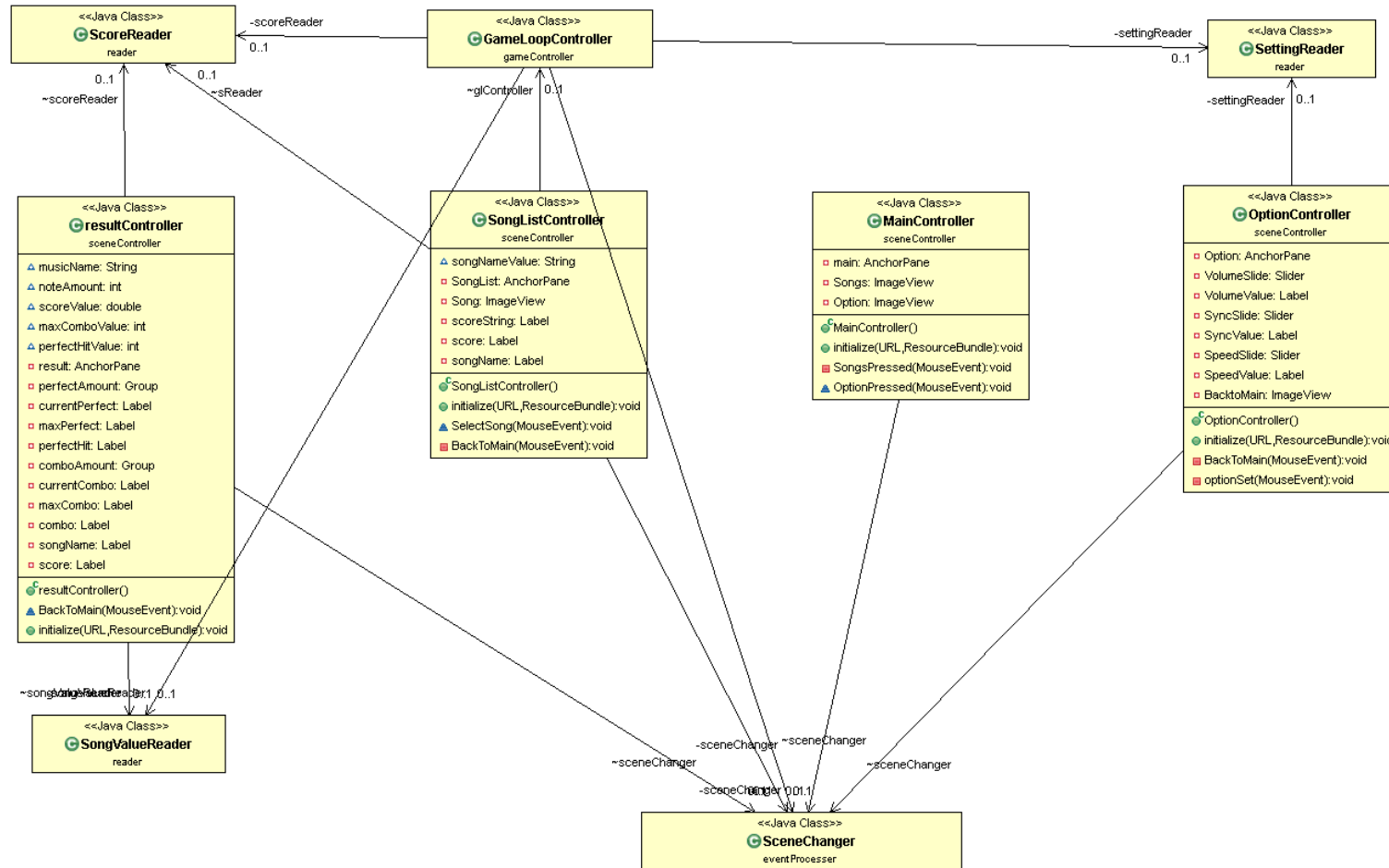
## 2. 코드설명-클래스 명세표

### 7) Controller\_gameControllerPackage

- ScoreController.java : 점수와 콤보 계산을 받아 업데이트 하는 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

## 2. 코드설명-클래스 구조도

### 8) Controller\_sceneControllerPackage



## 2. 코드설명-클래스 명세표

### 8) Controller\_sceneControllerPackage

- MainController.java : 메인 화면에서 입력을 받아 동작하는 클래스, Initializable 인터페이스를 상속받음
  - Songs : 곡 목록 버튼
  - Option : 설정 버튼
  - 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 동작하는 클래스, Initializable 인터페이스를 상속받음
  - 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
  - BackToMain( ) : 초기 화면으로 돌아가는 메소드
  - changed( ) : 슬라이더를 이용해 변경된 설정 값을 변수에 저장하는 메소드

## 2. 코드설명-클래스 명세표

### 8) Controller\_sceneControllerPackage

- ResultController.java : 결과 화면에서 입력을 받아 동작하는 클래스, Initializable 인터페이스를 상속받음
  - BackToMain( ) : 초기 화면으로 돌아가는 메소드
  - scoreValue : 점수
  - maxComboValue : 최대 콤보
  - perfectHitValue : 퍼펙트 판정을 받은 횟수;
- SongListController.java : 곡 선택 화면에서 입력을 받아 동작하는 클래스, Initializable 인터페이스를 상속받음
  - Song : 노래 실행 버튼
  - SelectSong( ) : 게임을 플레이 하는 메소드
  - BackToMain( ) : 초기 화면으로 돌아가는 메소드

# 3. 예외처리 항목

- FileloadException
  - 파일을 불러올 수 없을 때 처리 (Model\_reader\_NoteReader.java)
- IOException
  - fxml 파일을 읽을 수 없을 때 처리 (Controller\_eventProcessor\_SceneChanger.java)
  - 화면 전환을 하지 못 할 때 처리 (Controller\_eventProcessor\_SceneChanger.java)
  - 점수 및 콤보를 업데이트 하지 못 할 때 처리 (Controller\_gameController\_ScoreController.java)
  - 메인 화면을 돌아가지 못 할 때 처리 (Controller\_sceneController\_OptionController.java, Controller\_sceneController\_SongListController.java)
  - 설정 값을 덮어쓰지 못 할 때 처리 (Model\_reader\_ScoreReader.java, Model\_reader\_SettingReader.java, Model\_reader\_SongValueReader.java)
- Exception
  - 파일을 불러오거나 정보를 읽지 못 할 때 처리 (Model\_reader\_ScoreReader.java, Model\_reader\_SettingReader.java, Model\_reader\_SongValueReader.java)

## 4. 역할 분담

- 김재훈

- 게임 화면 구성
- 노트 움직임 구현
- 콤보 이펙트, 노트 이펙트 구현
- 코드 분리 및 정리

- 최성환

- 각 화면 제작
- 설정 기능 구현
- 보고서 작성