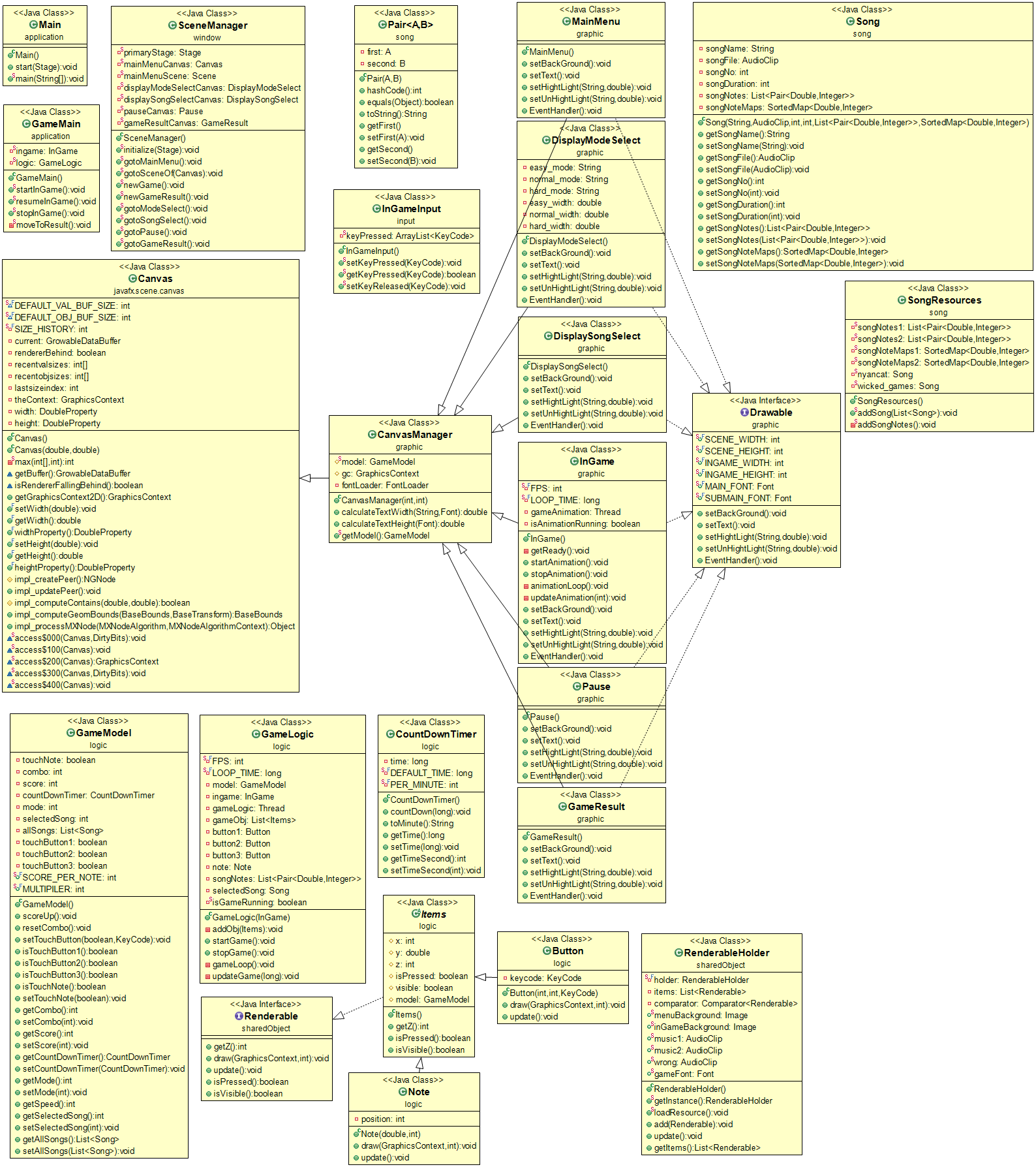
3. Implementation Detail

The diagram of the program is shown in Figure 6 below. There are 7 packages, in the total of 21 classes and 2 interfaces. The first package is application package which have 2 classes, Main and GameMain. Next package is graphic package, contains all canvases in this game and Canvas manager. It contains CanvasManager, DisplayModeSelect, DisplaySongSelect, GameResult, InGame, MainMenu, Pause and the interface, Drawable. Next package is input, contains InGameInput. Next one is logic package, contains Button, CountDownTimer, GameLogic, GameModel, Items, Note class. Next is sharedObject, which contains RenderableHolder class and Renderable interface. The song package contains Pair, Song, SongResources class. Last one is window package, contains SceneManager class. The UML diagram is shown below.



***Figure 6: The UML diagram of the program***

3.1 Package application

3.1.1 Class Main extends Application

3.1.1.1 Method

|  |  |
| --- | --- |
| + void start(Stage primaryStage) | The main entry for the JavaFX applications. There are the code setOnCloseRequest to completely close the application. |
| + void main(String[] args) | An entry of the application. |

3.1.2 Class GameMain

3.1.2.1 Field

|  |  |
| --- | --- |
| - InGame ingame | For call the canvas and startAnimation |
| - GameLogic logic | For call startGame |

3.1.2.2 Method

|  |  |
| --- | --- |
| + void startInGame() | - Initialize the fields  - Go to the scene of ingame  - Start the logic and animation threads |
| + void resumeInGame() | Move back to ingame canvas |
| + void stopInGame() | - Stop the logic and animation threads  - Call moveToResult method |
| + void moveToResult() | - Initialize the result canvas  - Move to that canvas |

3.2 Package graphic

3.2.1 Class CanvasManager extends Canvas

3.2.1.1 Field

|  |  |
| --- | --- |
| # GameModel model | To collect the data from user and use it in the whole game |
| # GraphicContext gc | To draw of each canvas |
| # FontLoader fontLoader | To use calculate width and height of the text |

3.2.1.2 Constructor

|  |  |
| --- | --- |
| + CanvasManager(int sceneWidth, int sceneHeight) | - Initialize canvas by setting the width and height  - Initialize fontLoader with Toolkit.getToolkit().getFontLoader();  - Initialize model when the model didn’t initial before. |

3.2.1.3 Method

|  |  |
| --- | --- |
| + double calculatedTextWidth(String text, Font font) | Calculate the string width |
| + double calculateTextHeight(Font font) | Calculate the string height |
| + GameModel getModel() | Return model |

3.2.2 Class DisplayModeSelect extends CanvasManager implements Drawable

3.2.2.1 Field

|  |  |
| --- | --- |
| - String easy\_mode | String “Easy” |
| - String normal\_mode | String “Normal” |
| - String hard\_mode | String “Hard” |
| - double easy\_width | Width position to draw easy\_mode |
| - double normal\_width | Width position to draw normal\_mode |
| - double hard\_width | Width position to draw hard\_mode |

3.2.2.2 Constructor

|  |  |
| --- | --- |
| + DisplayModeSelect() | - Set the width and height  - Initialize gc  - call the setBackground(), setText(), EventHandler() methods |

3.2.2.3 Method

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |