**Sokoban – Project Description**

This project is designed following the MCV paradigm.

The UML diagram is at the end of the doctument.

# **View**

The view package holds the classes in charge the GUI. The icons for the GUI are placed inside the package's folder.

## Sokoban

This class hodls the Main() method for the project.

This class is a JFrame implementation. It holds an instance of a **BoardModel** that it then parses to instances of all the controllers. In addition, It uses an instance of the **BoardDraw** class to draw the board. It receives messages from the Controllers (mainly a view update request) and uses the **BoardDraw** class to do so.

## BoardDraw

This class contains the "drawing" logic of the game board. It receives Cell arrays and returns an appropriate **JPanel** instance. The Sokoban class has an instance of this object that it uses when it receives a view-update request.

# **Model**

The view package holds the classes in charge the GUI.

## BoardModel

Holds a 2D **Cell** array representing the board. This class calculates moves and verifies if victory is achieved. The class receives messages from the classes in the Controller package.

## Direction (Enum)

The **ENUM** represents a direction on the board – *UP*, *DOWN*, *LEFT*, *RIGHT*.

It is used for the **BoardModel**.makeMove() method.

# **LevelLoader**

The levelLoader package holds the classes that load different levels from text files.

## LevelLoader

This class reads .txt files and returns a corresponding 2D **Cell** array.

## Cell

This class is the building block of the game board. It represents a single cell in the game.

# **Controller**

The Controller package contains the listener classes. This "listens" to user input and sends messages to the model package and view package accordingly.

## MainController

The main controller in the project.

Upon arrow key-press will send a message to the **BoardModel** to make a move in the designated direction

Upon the user pressing the Reset button, the controller sends a message to the **BoardModel** to reset.

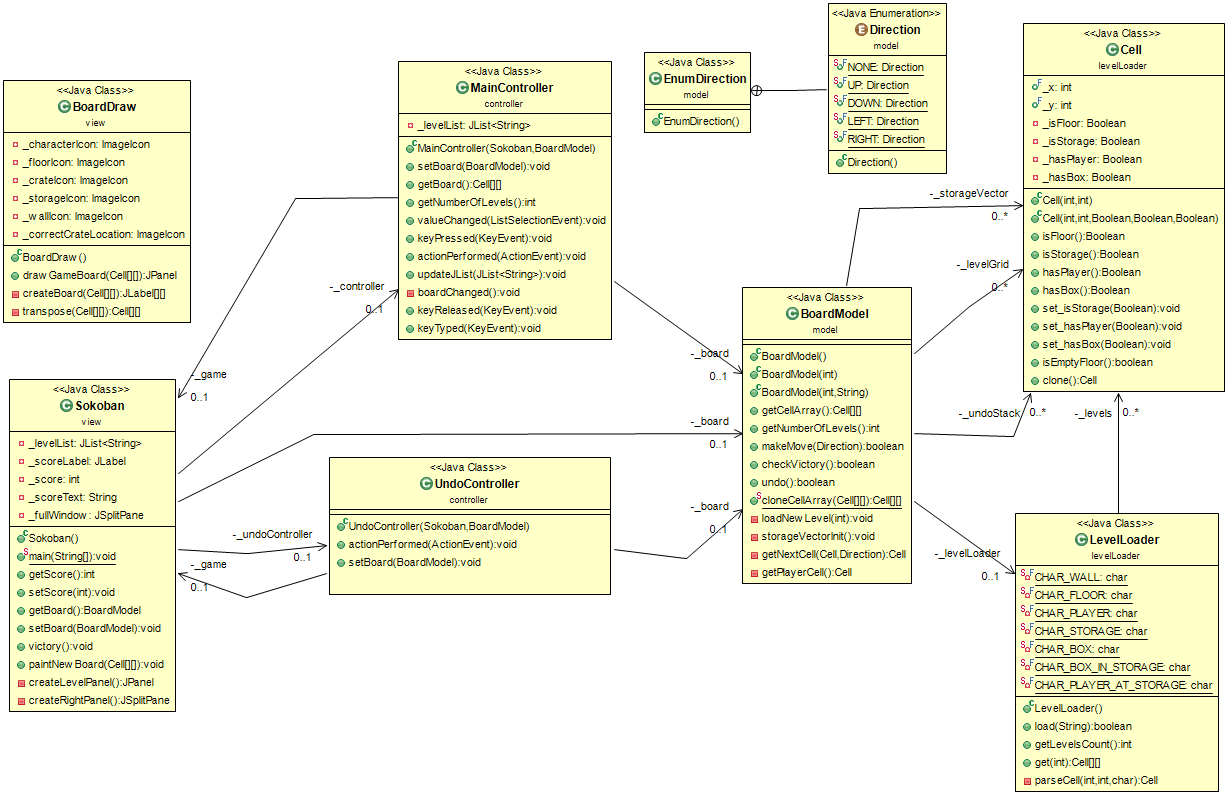
Upon user choosing an element in the level list the controller sends a message to the **BoardModel** to load a new level accordingly.

After a message is sent to the Model package, a message is then sent to the Sokoban class that instructs it to update the view according to the board.

## UndoController

This class is the action listener for the Undo button.

Upon the user pressing the button calls the **BoardModel**.undo() method. Afterwards a message instructing a view update is sent to the Sokoban class.

****