Classes

class Application

class DistributeVisitor

class House extends Store

class Player

class Store

class Turn

class ClickInfo

class Controller

class Gui extends JFrame implements ActionListener

Functions

```
/*
 *
```

```
 */
ClickInfo(boolean firstPlayer, int index)

/*
 * return the truth value of the sentence "this house belongs to first player".
 */
boolean belongsToFirstPlayer()

/*
 * Handle Gui events.
 * @param event The gui event
 * @param data Relevant data for event or null otherwise.
 */
void handleGuiEvent(GuiEvent event, Object data)

/*
 * Updates the store with given value of first or second player.
 * @param firstPlayer
 * @param score
 */
void updateStore(boolean firstPlayer, int score)

/*
 * Update a house.
 * @param firstPlayer
 * @param index
 * @param score
 * @param enableCell
 */
void updateHouse(boolean firstPlayer, int index, int score, boolean enableCell)

/*
 * refresh the display.
 * This method fetches seed counts from all houses
 * and stores and then forces the GUI to update.
 */
void refreshDisplay()

/*
 * Disable the house buttons.
 */
void disableHouses()

/*
```

```
 * Constructor for the controller.
 */
Controller()

/**
 * Create a mancala player button.
 * @param text
 * @param firstPlayer Does it belong to first player.
 * @param index The index of the button as it is in the UML model.
 */
MancalaButton(String text, boolean firstPlayer, int index)

/*
 * set the index of a house.
 * @param index to be given.
 */
void setIndex(int index)

/*
 * get the index of a house
 */
int getIndex()

/*
 * set a player as the first player.
 */
void setFirstPlayer(boolean firstPlayer)

/*
 * get the truth value of the sentence "this player is the first player"
 */
boolean isFirstPlayer()

/*
 * This function asks and sets the players names. Player name is asked with
 * a popup window.
 */
void askAndSetPlayerNames()

/*
 * Setup the menu items on the menubar.
 */
void setupMenus()
```

```
/**
 * Handle the clicks on the MancalaButtons.
 */
void actionPerformed(ActionEvent arg0)

/*
 * This function is used to initialize both players store and the houses.
 */
void setupStoresAndHouses()

/*
 * This function updates a buttons text from the button grid. The button grid
 * contains all the buttons that a user can press on the screen while playing.
 * This function is used to display different text on buttons on the grid.
 */
void updateCell(int x, int y, String text)

/*
 * This function enables a button from the button grid. The button grid contains
 * all the buttons that a user can press on the screen while playing.
 */
void enableCell(int x, int y, boolean enabled)

/*
 * This function creates a gui that is controlled via a controller. This is used
 * on the top level of the program forward user input to the modelling of events
 * and results of the modelling are again forwarded to graphical events that are
 * displayed to the players.
 */
Gui(Controller controller)

/*
 * Display a message that tells the players that the game is over.
 */
public void displayGameOver(String text)
```