

contains

TurnResultPanel

+ setTurnResultText(textResult: String)

- turnResultText: JLabel

- turnResultTitle: JLabel

+ TurnResultPanel()

- bagLimitLabel: JLabel

roomNameLabel: JLabel

- buttonGroup: JButtonGroup

- bagLimitInputField: JTextField

- playerTypeLabel: JLabel

- submitButton: JButton

- startGameButton: JButto

- playerNameInputField: JTextField

- humanRadioButton: JRadioButton

- computerRadioButton: JRadioButton

- roomDropDown: JComboBox<String>

+ AddPlayerFrameImpl(world: ReadOnlyWorld)

- savedPlayerDeatils: List<String>

+ setFeatures(features: Features)

+ setDisplay(display: Boolean)

+ getPlayerDetails(): List<String>

+ savePlayerDetails()

+ dispatchEventClose()

+ updateModel(world:ReadOnlyWorld)

All these display methods will be

called from the controller and view

will decide how to display a

component by appropriately

defining these methods.

+ setFeatures(features: Features)

+ setDisplay(display: Boolean)

+ showItemChooser(): String

+ showRoomChooser(): String

+ dispatchEventClose()

TurnInfoPanel

- turnInfoText: JLabel

- turnInfoTitle: JLabel

+ TurnInfoPanel()

+ setTurnInfoText(textInfo: String)

+ updateModel(world:ReadOnlyWorld)

+ showAttackItemChooser(): String

- fileChooser: JFileChooser

- newSpecItem: JMenuItem

- exitItem: JMenuItem

- existingSpecItem: JMenuItem

- createMenuBar: JMenuBar

+ setFeatures(features: Features)

+ setDisplay(display: Boolean)

| + dispatchEventClose()

+ updateModel(world:ReadOnlyWorld)

+ WelcomeFrameImpl(world: ReadOnlyWorld)