ChooseShadingTypeDialog

- APPLICATIONSTATE :IApplicationState

\<get>DialogTitle() :String
\<get>DialogText() :String

\<get>DialogOptions() :ShapeShadingType[]
\<get>CurrentSelection() :ShapeShadingType

ChooseSecondaryColorDialog

- APPLICATIONSTATE :IApplicationState

\<get>DialogTitle() :String
\<get>DialogText() :String

\<get>DialogOptions() :ShapeColor[]
\<get>CurrentSelection() :ShapeColor

ChooseShapeDialog

- APPLICATIONSTATE :IApplicationState

\<get>DialogTitle() :String
\<get>DialogText() :String

\<get>DialogOptions() :ShapeType[]
\<get>CurrentSelection() :ShapeType

ChoosePrimaryColorDialog

- APPLICATIONSTATE :IApplicationState

\<get>DialogTitle() :String
\<get>DialogText() :String

\<get>DialogOptions() :ShapeColor[]
\<get>CurrentSelection() :ShapeColor

<< IDialogChoice >>

\<get>DialogTitle() :String
\<get>DialogText() :String
\<get>DialogOptions() :T[]
\<get>CurrentSelection() :T

ChooseStartAndEndPointModeDialog

- APPLICATIONSTATE :IApplicationState

\<get>DialogTitle() :String
\<get>DialogText() :String

\<get>DialogOptions() :StartAndEndPointMode[]
\<get>CurrentSelection() :StartAndEndPointMode

<< IApplicationState >>
Interface

+ setActiveShape() :void
+ setActivePrimaryColor() :void
+ setActiveSecondaryColor() :void
+ setActiveShadingType() :void
+ setActiveStartAndPointMode() :void

\<get>ActiveShapeType() :ShapeType
\<get>ActivePrimaryColor() :ShapeColor
\<get>ActiveSecondaryColor() :ShapeShadingType
\<get>ActiveShapeShadingType() :ShapeShadingType
\<get>ActiveStartAndEndPointMode() :StartAndEndPointMode

Implements

---

1. Mouse Listener
2. ShapeList Collection class

DialogProvider

- CHOOSESHAPEDIALOG :IDialogChoice<ShapeType>
- CHOOSEPRIMARYCOLORDIALOG :IDialogChoice<ShapeColor>
- CHOOSESECONDARYCOLORDIALOG :IDialogChoice<ShapeColor>
- CHOOSESHADINGTYPEDIALOG :IDialogChoice<ShapeShadingType>
- CHOOSESTARTANDENDPOINTMODEDIALOG :IDialogChoice<StartAndEndPointMode>

-APPLICATIONSTATE :IApplicationState

DialogProvider(IApplicationState) :DialogProvider

\<get>ChooseShapeDialog() :IDialogChoice<ShapeType>
\<get>ChoosePrimaryColorDialog() :IDialogChoice<ShapeColor>
\<get>ChooseSecondaryColorDialog() :IDialogChoice<ShapeColor>
\<get>ChooseShadingTypeDialog() :IDialogChoice<ShapeColor>
\<get>ChooseStartAndEndPointModeDialog() :IDialogChoice<StartAndEndPointMode>

ApplicationState

- UIMODULE :IUiModule
- DIALOGPROVIDER :IDialogProvider

- activeShapeType :ShapeType
- activePrimaryColor :ShapeColor
- activeSecondaryColor :ShapeColor
- activeShapeShadingType : ShapeShadingType
- activeStartAndEndPointMode :StartAndEndPointMode

ApplicationState(IUiModule) :ApplicationState

+ setActiveShape() :void
+ setActivePrimaryColor() :void
+ setActiveSecondaryColor() :void
+ setActiveShadingType() :void
+ setActiveStartAndPointMode() :void

\<get>ActiveShapeType() :ShapeType
\<get>ActivePrimaryColor() :ShapeColor
\<get>ActiveSecondaryColor() :ShapeColor
\<get>ActiveShapeShadingType() :ShapeShadingType
\<get>ActiveStartAndEndPointMode() :StartAndEndPointMode

- setDefaults() :void

Abstract Factory?

JPaint Controller

- UIMODULE :IUiModule
- APPLICATIONSTATE :IApplicationState

JPaintController(IUiModule, IApplicationState) :JPaintController

+ setup() :void
- setupEvents() :void

<< IJPaintController >>
Interface

+ setup() :void

<< IDialogProvider >>
Interface

\<get>ChooseShapeDialog() :IDialogChoice<ShapeType>
\<get>ChoosePrimaryColorDialog() :IDialogChoice<ShapeColor>
\<get>ChooseSecondaryColorDialog() :IDialogChoice<ShapeColor>
\<get>ChooseShadingTypeDialog() :IDialogChoice<ShapeColor>
\<get>ChooseStartAndEndPointModeDialog() :IDialogChoice<StartAndEndPointMode>

<< IUiModule >>
Interface

+ addEvent(EventName, IEventCallback) :void
\<get>DialogResponse(IDialogChoice) :T

Implements

Gui

GUI :IGuiWindow

Gui(IGuiWindow)

+ addEvent(EventName, IEventCallback) :void

\<get>DialogResponse(IDialogChoice) : T

COMMAND

<< IEventCallback >>
Interface

run() :void

COMMAND OBJECT GOES HERE
RUN IS EXECUTE();

GuiWindow

DEFAULTWIDTH :int
DEFAULTHEIGHT :int
DEFAULTTITLE :String
DEFAULTBUTTONDIMENSIONS :Insets

EVENTBUTTONS :Map<EventName, JButton> = new HashMap<>()

CANVAS :PaintCanvas

GuiWindow(PaintCanvas) :GuiWindow

+ getButton(EventName) :JButton

- createWindow() :JPanel
- createMenu() :JPanel
- addButtonToPanel(EventName, JPanel) :void
- createButton(EventName) :JButton
- createButtonBorder() :Border
- createButtonPanel() :JPanel
- createBackgroundPanel() :JPanel

EventName

enum

+CHOOSE_SHAPE{}
+CHOOSE_PRIMARY_COLOR{}
+CHOOSE_SECONDARY_COLOR{}
+CHOOSE_SHADING_TYPE{}
+CHOOSE_START_AND_ENDPOINT_MODE{}
+UNDO{}
+REDO{}
+COPY{}
+PASTE{}
+DELETE{}

<< IGuiWindow >>
Interface

\<get>Button(EventName, IEventCallback) :JButton

Implements

PaintCanvas

\<get>Graphics2D() :Graphics2D