SWEN PRIXOS DESIGN DOCUMENT

Authors: Isabella Sturm, Ivan Kovacevic, Srdan Lazarevic

Date: 7 April 2017
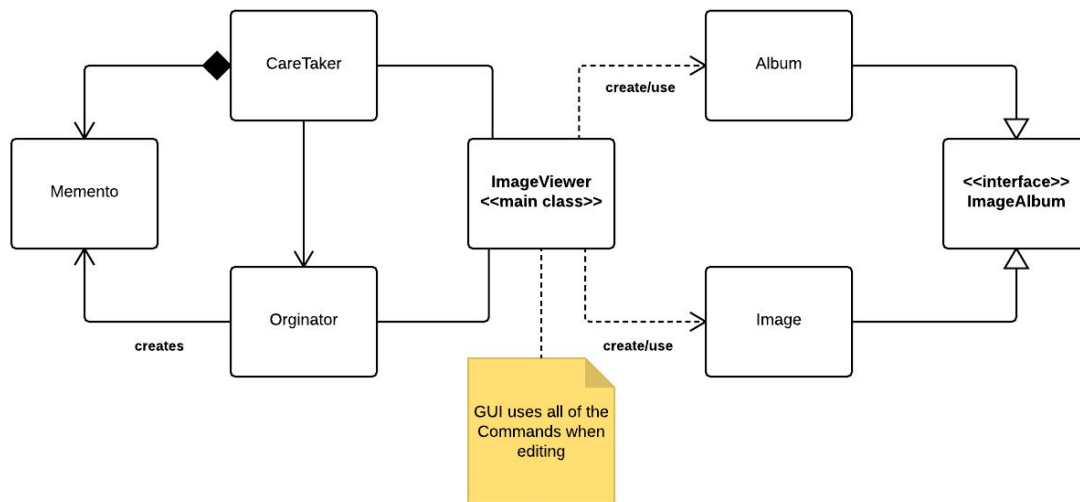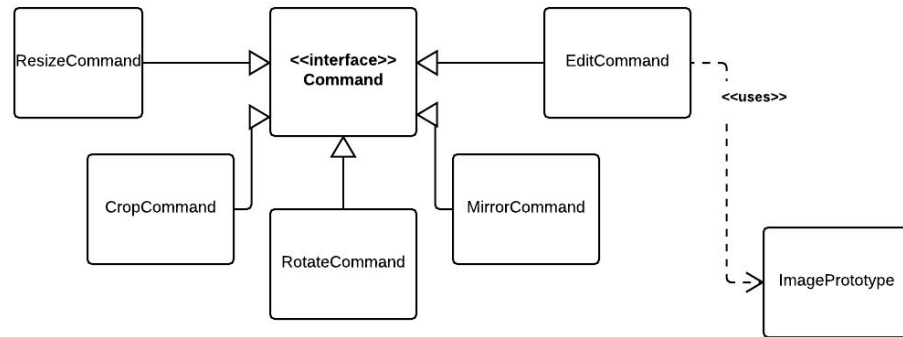
Class table

Class Name	Class Description	Uses	Used By	Methods
ImageViewer	Main class; GUI. This class provides the user interface of the program.	ResizeCommand, CropCommand, RotateCommand, MirrorCommand, CareTaker, Originator		main() ImageViewer()
Command	Interface for commands that will control the functionalities allowed by the user		ResizeCommand, CropCommand, MirrorCommand, RotateCommand, EditCommand	execute()
ResizeCommand	Implements Command interface; when called by user, resize command prompts user to change size of image	Command interface	ImageViewer	resize() execute()
CropCommand	Implements Command interface; when called, user can crop image	Command interface	ImageViewer	crop() execute()
RotateCommand	Implements Command interface, when called upon, user can chose to rotate image by custom number of degrees or by the standard 90 or -90	Command interface	ImageViewer	rotate() execute()

	degrees			
MirrorCommand	Implements Command interface, when called, user can mirror original image vertically or horizontally	Command interface	ImageViewer	mirror() execute()
EditCommand	Implements Command interface; when user wants to edit page, EditCommand calls prototype to make copy of original photo before editing	Command interface, ImagePrototype	ImageViewer	edit() execute()
ImageAlbum	Interface for images and album/ labels (Composite Pattern)		Album, Image	open()
Album	Implements ImageAlbum; "Directory" level; Array or ArrayList to store images in a specific album	ImageAlbum interface	ImageViewer, Command	open() addImage() showNext() showPrevious()
Image	Implements ImageAlbum; "File" level;	ImageAlbum interface	ImageViewer, Command	open()
CareTaker	Stores mementos and can restore (redo/undo) to another state	Memento	ImageViewer	checkpoint() restore()

	of the image			
Originator	Creates and saves mementos of the image	Memento	ImageViewer	createMemento) reset()
Memento	Keeps state of mementos of image when image is edited		CareTaker, Originator	save(State s) retrieve()
ImagePrototype	Provides functionality to clone image that user wants to edit so the original can remain and the new edited image can be created		EditCommand	clone()

UML Diagrams

Diagram 1 (Command pattern):

ResizeCommand → <<interface>> Command
CropCommand → <<interface>> Command
RotateCommand → <<interface>> Command
MirrorCommand → <<interface>> Command
EditCommand → <<interface>> Command
EditCommand <<uses>> → ImagePrototype

Diagram 2 (Memento / Album pattern):

CareTaker ◆→ Memento
CareTaker → Orginator
Orginator → Memento (creates)
CareTaker → ImageViewer <<main class>>
ImageViewer <<main class>> create/use → Album
ImageViewer <<main class>> create/use → Image
Album → <<interface>> ImageAlbum
Image → <<interface>> ImageAlbum

GUI uses all of the Commands when editing

Design Patterns

<u>Memento</u>: Handles Undo and Redo functionality when editing images
<u>Prototype [+ Factory]</u>: Clones (makes copy) of original image and creates new image for editing
<u>Command</u>: Handles functionality of JButtons in GUI
<u>Composite</u>: Organizes into and allows user to go through hierarchical collection of albums and images inside.

Patterns that we are under consideration:

<u>MVC</u> - On linking the GUI with Command and adding a Model.

<u>Virtual Proxy:</u> may having something to do with editing image or with displaying

<u>Facade</u>: We will probably use this if we ever run into any trouble with a lot of interfaces in our project. So for example if we want to provide our user with the cleanest experience we might consider using this, however at the current point we do not know if we will use so many interfaces.