

SE 3XA3: MIS Gifitti

Team #2, Gifitti
Pavle Arezina, arezinp
Nicolai Kozel, kozeln
Riley McGee, mcgeer

December 7, 2016

Contents

1	Introduction	1
2	Module Declarations	1
3	Module Specifications	1
3.1	Hardware-Hiding Module	1
3.2	Behavior-Hiding Module	2
3.3	Software Decision Module	2
3.4	Image Loading Module	2
3.4.1	Variables	2
3.4.2	Methods	2
3.5	Image Processing Module	3
3.6	Image Conversion Module	3
3.6.1	Variables	3
3.6.2	Methods	3
3.7	GIF Transformation Module	3
3.7.1	Variables	4
3.7.2	Methods	4
3.8	GIF Model Module	4
3.8.1	Variables	4
3.8.2	Methods	4
3.9	View-Model Module	5
3.9.1	Variables	5
3.9.2	Methods	5

List of Tables

1	Revision History	ii
----------	-----------------------------------	-----------

List of Figures

Table 1: **Revision History**

Date			Version	Notes
November	12th	2016	1.0	Initial creation of document
November	13th	2016	1.1	Spellcheck and formatting
November	30th	2016	2.0	Updated to reflect the release of Giftiti.

1 Introduction

Gifitti is a software solution for allowing users with a varying knowledge base on image manipulation, as well as technical experience with computers to be able to load, decompose, and modify GIF images.

This document describes the internal workings of each module implemented as a part of Gifitti. Described are the methods, input/output, as well as any environment and state variables that the module uses. The MIS will continue to be updated as more methods and features are added to each module.

2 Module Declarations

This section simply lists the modules used by Gifitti. More details on these modules (and how they interact) can be found in the MG document.

M1: Hardware-Hiding Module

M2: Behaviour-Hiding Module

M3: Software Decision Module

M4: Image Processing Module

M5: Image Loading Module

M6: Image Conversion Module

M7: GIF Transformation Module

M8: GIF Model Module

M9: View-Model Module

3 Module Specifications

3.1 Hardware-Hiding Module

Since the hardware hiding module is implemented through the OS and is not actually coded as part of Gifitti, there is nothing to add to the MIS.

3.2 Behavior-Hiding Module

This module serves as a communication layer between the hardware-hiding module and the software decision module. This module covers M4, M5, and M8. Again, this module is not actually coded so there is nothing to add to the MIS.

3.3 Software Decision Module

This module includes any data structures and algorithms used in the system that do not provide direct interaction with the user. This module covers M7, and M6. Again, this module is not actually coded so there is nothing to add to the MIS.

3.4 Image Loading Module

This module allows for the conversion of system paths to images in a usable form within the software.

3.4.1 Variables

1. Filepath (Input)
2. originalImage (Output)
3. gifImage (Output)
4. numberOfFrames (Output)
5. frames[] (Output)

3.4.2 Methods

1. loadImage(string FilePath)
This method takes as input the file path and converts it to an image file type using M4. This image is saved in the Image variable.
2. frameConstruction(Image img)
Decomposes GIF into its frames.

3.5 Image Processing Module

This module converts the input images into a system usable form. Implementation is handled by the ImageMagick.NET API.

3.6 Image Conversion Module

This module converts the GIF from a GIF Model (M8) to an image exportable by the system (GIF, TIFF, JPEG, PNG, BMP).

3.6.1 Variables

1. allowedFileTypes[] (Static)
2. exportFileType (Input)
3. curLoadedGIF (Input)
4. startFrame (Input)
5. endFrame (Input)
6. delay (Input)
7. filePath (Input)
8. outputImage (Output)

3.6.2 Methods

1. exportGIF(string filePath)
This handles exporting the currently represented GIF.
2. ConvertTo(string filePath, FileType ft)
Export of all GIF frames as type ft to path filePath.

3.7 GIF Transformation Module

This module handles frame by frame manipulation of the GIF Models (M8) in order to complete operations such as resizing and modifying the speed, etc. All the methods and variables of this module are listed below.

3.7.1 Variables

1. curLoadedGIF (Input)
2. modifiedGIF (Output)

3.7.2 Methods

1. Resize(int x, int y)
This method will resize the GIF to the size passed as parameters.
2. ChangeSpeed(int newDelay)
This method will alter the speed of the GIF by the value passed as a parameter. This is information hidden, it is meant to extend more for future releases.

3.8 GIF Model Module

This module represents a GIF object within the system. It contains methods and properties that allow the system to properly manipulate the GIF.

3.8.1 Variables

1. Frames[]
2. delay
3. startFrame
4. endFrame

3.8.2 Methods

1. GetNextFrame()
Returns the next frame to be handled by the controller based on Model state.
2. GetFrame(int index, [bool isResize])
Fetches frame number index (index is an element of [0, numberOfFrames - 1]), if isResize return a clone, else return a reference to the image.

3. Dispose()
Memory management and system cleaning.

3.9 View-Model Module

The View-Model Module handles all View elements created by C# WinForms, as well as their static link to the model representation of a view. These links are referred to as View-Models. View-Models and Views are not abstracted into separate modules due to how the implementation occurs in the C# coding paradigm. Views (called Forms), are linked to their View-Model via partial classes, this keeps coupling high between the View-Models and Views, but low coupling to the Model.

3.9.1 Variables

1. MainForm
2. HelpContextForm
3. SaveAsXbyYForm
4. GifModel (input)

3.9.2 Methods

1. MenuOpenHelp()
Show the HelpContextForm.
2. MenuItemResize()
Show the SaveAsXbyYForm.
3. StartStopClickEvent()
Toggles between play and stop for GIFs.
4. MenuOpenFile()
Launches a OpenFileDialog prompting the user to open a GIF, the opened GIF is then loaded as a GifModel.
5. MenuItemFileSaveAs()
Launches a SaveFileDialog prompting the user for a path to save the loaded GIF to a location.

6. StopFrameTextChanged()
Validation and Verification of the text box representing endFrame.
7. StartFrameTextChanged()
Validation and Verification of the text box representing startFrame.
8. GifFPSSliderUpdate()
Updates delay of the GIF via slider position.
9. AdvanceFrameTimer()
System interrupt to draw the next frame, modeling the GIF.
10. exportFrames(ImageFormat format)
Exports the GIF as a bundle of Images representing each frame from startFrame to endFrame, exported image will be of type Format.
11. MenuItemExportTIFF()
Exports the GIF frames as TIFF file type
12. MenuItemExportPNG()
Exports the GIF frames as PNG file type
13. MenuItemExportJPEG()
Exports the GIF frames as JPEG file type
14. MenuItemExportBMP()
Exports the GIF frames as BMP file type
15. chkImage()
Validates the Image loaded is legal.
16. openingGIFLinkClicked()
Displays information about opening GIFs in the help context.
17. savingGIFLinkClicked()
Displays information about saving GIFs in the help context.
18. savingImageLinkClicked()
Displays information about saving GIFs as other image types in the help context.

19. pickFramesLinkClicked()
Displays information about selecting sub GIFs the help context.
20. playbackSpeedLinkClicked()
Displays information about modifying GIF playback speed in the help context.
21. resizeLinkClicked()
Displays information about resizing GIFs in the help context.
22. SaveAsButtonClick()
Utilized in the SaveAsXbyYForm, Prompts the user for a file path for a resized GIF.
23. widthChanged()
Validation and Verification on the width text box in the SaveAsXbtY-Form
24. heightChanged()
Validation and Verification on the height text box in the SaveAsXbtY-Form