

SE 3XA3: Test Plan
Gifitti

Team #2,Gifitti
Nicolai Kozel kozeln
Riley McGee mcgeer
Student 3 name and macid

October 24, 2016

Contents

1	General Information	1
1.1	Purpose	1
1.2	Scope	1
1.3	Acronyms, Abbreviations, and Symbols	1
1.4	Overview of Document	2
2	Plan	2
2.1	Software Description	2
2.2	Test Team	2
2.3	Automated Testing Approach	2
2.4	Testing Tools	2
2.5	Testing Schedule	2
3	System Test Description	2
3.1	Tests for Functional Requirements	2
3.1.1	The User is Able to Open a GIF from a specified location	2
3.1.2	Area of Testing2	4
3.2	Tests for Nonfunctional Requirements	4
3.2.1	Area of Testing1	4
3.2.2	Area of Testing2	4
4	Tests for Proof of Concept	4
4.1	Opening a GIF file for playback	4
4.2	Saving a GIF file's frames	5
5	Comparison to Existing Implementation	5
5.1	Graphics/UI	5
5.2	Performance	6
6	Unit Testing Plan	6
6.1	Unit testing of internal functions	6
6.2	Unit testing of output files	6
7	Appendix	7
7.1	Symbolic Parameters	7
7.2	Usability Survey Questions	7

List of Tables

1	Revision History	ii
2	Table of Abbreviations	1
3	Table of Definitions	1

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

This document describes the test plan for the Gifitti application developed for 3XA3 at McMaster University.

1 General Information

1.1 Purpose

1.2 Scope

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	Definition
Abbreviation1	Definition1
Abbreviation2	Definition2

Table 3: **Table of Definitions**

Term	Definition
Term1	Definition1
Term2	Definition2

1.4 Overview of Document

2 Plan

2.1 Software Description

2.2 Test Team

2.3 Automated Testing Approach

2.4 Testing Tools

2.5 Testing Schedule

See Gantt Chart at the following url ...

3 System Test Description

3.1 Tests for Functional Requirements

3.1.1 Open GIF

The User is Able to Open a GIF from a specified location

1. Select proper formatted gif- id1

Type: Manual Functional. Initial State: Program loaded; no GIF Loaded. Input: File name. Output: System loads gif into memory, displays it to the user in the gif view.

How test will be performed:

- (a) Launch the program
- (b) Select Open
- (c) From the Open dialog specify a path to a known gif image
- (d) After the image is loaded verify that it is being displayed, and resides in system memory

2. No File Selected in File Dialog-id2

Type: Manual Functional. Initial State: Program loaded; no GIF Loaded. Input: No file path. Output: No image loaded.

How test will be performed:

- (a) Launch the program
- (b) Select Open
- (c) Select open option with no file path specified
- (d) Verify program remains open, and no image is loaded

3. Close File Dialog-id3

Type: Manual Functional. Initial State: Program loaded; no GIF Loaded. Input: None. Output: No image loaded.

How test will be performed:

- (a) Launch the program
- (b) Select Open
- (c) Close the file dialog
- (d) Verify program remains open, and no image is loaded

4. Open random non gif file-id4

Type: Manual Functional. Initial State: Program loaded; no GIF Loaded. Input: File that is not a GIF. Output: No image loaded.

How test will be performed:

- (a) Launch the program
- (b) Select Open
- (c) Try and select a file that is not a GIF or specify a file path to a known file
- (d) Verify program remains open, and no image is loaded

3.1.2 Area of Testing2

...

3.2 Tests for Nonfunctional Requirements

3.2.1 Area of Testing1

Title for Test

1. test-id1

Type:

Initial State:

Input/Condition:

Output/Result:

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

3.2.2 Area of Testing2

...

4 Tests for Proof of Concept

4.1 Opening a GIF file for playback

Open GIF

1. OpenGif-01

Type: Manual Functional

Initial State: Program must be in normal state (form window is open and playback window is blank).

Input: File

Output: GIF is shown in playback window of the form.

How test will be performed: Click open button, select a file of type .gif, and verify that the GIF loads and begins to playback within the form window.

4.2 Saving a GIF file's frames

Save GIF

1. SaveGif-01

Type: Manual Functional

Initial State: Program must be in playback state (form window is open and playback window is playing GIF).

Input: GIF File

Output: GIF's frames are saved as .bmp in folder specified.

How test will be performed: Click save frames button, select a folder, and verify that the GIF's frames are saved to the specified folder as .bmp.

5 Comparison to Existing Implementation

5.1 Graphics/UI

1. GIF playback resolution is the same or better than Gif Viewer.
2. Program has the same button scheme as Gif Viewer (Open button to open file, Extract Frames button to save frames).

3. Program has a help menu available that is similiar to Gif Viewer, but is available at all times.
4. Program's color scheme and design resembles Gif Viewer.

5.2 Performance

1. GIF playback is at the same smoothness/framerate or better than Gif Viewer.
2. Opening and saving a file takes the same amount of time or less than Gif Viewer.

6 Unit Testing Plan

6.1 Unit testing of internal functions

6.2 Unit testing of output files

7 Appendix

This section contains symbolic parameters for this document and the usability survey that will be delivered to a focus group upon initial completion of the application.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions

The survey will be delivered in the same format as the Questionnaire for User Interface Satisfaction. This questionnaire is composed of various questions pertaining to several sub categories on a 0-9 scale. This includes the screen, terminology and system information, learning, and system capabilities. It also allows the user to list the most positive and negative aspects of the program. The questionnaire can be found at garyperلمان.com