

2016



MICT1 – Exercise Week 3

Delano Cörvers

Zuyd University

01-Mar-16

MICT1 – Exercise Week 3

Reverse Engineering Data – Exercise week 3

Group	Group 1	
Students	Delano Cörvers	(1306669)
	Davy Heutmekers	(1309730)
	Rik Kierkels	(1354442)
Module	MICT1 – Reverse Engineering Data	
Assignment	Exercise – Week 3	
School year	2015 – 2016	

Table of content

1	What we found	4
2	Where we found it	4
3	How we found it	6

1 What we found

We investigated the provided files, `f_z.data` and `e_r.data`. We quickly realized that the provided files were corrupted and were missing image frames. We reconstructed both files using the full frames that we were able to recover. The file `e_r.data` originally contained 16 frames and we were able to recover 9 frames. `F_z.data` originally contained 11 frames and we recovered 6 frames. We added both reconstructed images in our GitHub commit.

There are a few leftover frames still in the provided files. These frames were incomplete (missing data), so we decided not to include them in the reconstructed GIFs.

The reconstructed GIFs are now interpreted as valid media files by Windows, Chrome and Internet Explorer. An Exif analysis tool we used validated the file as a GIF and showed no errors.

2 Where we found it

We split of the large files into smaller sub-files, so it is easier to process for us.

E_r.data

Parts

Starting position	Ending position	File
0x00000000	0x00013fff	Start.data
0x0001b800	0x00027fff	Part 1.data
0x00037000	0x0003bfff	Part 2.data
0x00043800	0x0003afff	Part 3.data
0x00057800	0x00063fff	Part 4.data
0x0006e000	0x00081fff	Part 5.data
0x00089800	0x000cf52d	End.data

Frames

Files	Start	End
Start	0x00000320	0x0001063b
Part 1	None	
Part 2	None	
Part 3	None	
Part 4	None	
Part 5	0x00003df0	0x0000d668
End	0x00004758 0x0000f442 0x0001880d 0x00023ab4 0x0002d155 0x00038ebc 0x00044666	0x0000f441 0x0001880c 0x00023ab3 0x0002d154 0x00038ebb 0x00044665 0x00045d2b

F_z.data

Parts

Starting position	Ending position	File
0x00000000	0x00018fff	Start.data
0x0001e000	0x00031fff	Part 1.data
0x00032000	0x0004afff	Part 2.data
0x00050000	0x00054fff	Part 3.data
0x0005a000	0x00063fff	Part 4.data
0x00069000	0x00072fff	Part 5.data
0x00078000	0x0007cfff	Part 6.data
0x00082000	0x00090fff	Part 7.data
0x00096000	0x000e6160	End.data

Frames

Files	Start	End
Start	0x00000030d	0x00012a32
Part 1	Start.data -> 0x00012a33	0x0000c2bd
Part 2	None	
Part 3	None	
Part 4	None	
Part 5	None	
Part 6	None	
Part 7	0x000031ef	End.data -> 0x0006461
End	0x00006462 0x0001898f 0x0003d921	0x0001898e 0x0002b50a 0x0005015e

3 How we found it

The first thing we did after receiving the files on Monday was checking the structure of the file. The header showed GIF89a and the trailer had the correct ending bytes for a GIF file (0x00 0x3B). Both provided files had a lot of clear memory. These were likely to be deleted or corrupted parts from the original files. After deleting the free space between the actual parts of data (ending up with one large file without free space), we renamed the files from filename.data to filename.gif and tested the images.

We were able to see one and a half frame of each image, then we tried Google's image search functionality to find the original file to see what the GIF was supposed to look like.

Next we compared the original file and the provided file with each other to find the differences using Hex Editor Neo. File e_r.data and f_z.data had corrupted or was missing (part of) frames.

At first we thought we had to mix the parts that were already in the files to reconstruct the order of the bytes to get a valid image out of it. So we split the original files into separate files without the free space. After trying multiple different combinations we came to the conclusion that our method was wrong.

We started looking for a tool to help us figure out what was wrong. After Googling for a bit we came across Jeffrey's Exif Viewer. This is an online Exif analyzer which validates the provided image and gives back information about the file itself. Using this analyzer we verified that both files (original without free space) only had one or two frames in each of them. We also got an error code on the original files "Invalid block label found".

Because we only saw one full frame and a half frame for each we assumed the second frame was causing a problem (corrupted frame or a frame that was missing the terminator byte). We started looking for the second frame using Neo Editor. We discovered that some of the frames (like the second frame) were incomplete or missing data. We searched all our split-up files for frames where there was a proper terminator (0x00) and a starting position for a frame (0x21 0xF9). We reconstructed both files by starting with the header and ending with the trailer, in between we placed all full frames we could find.

Jeffrey's Exif Viewer <http://regex.info/exif.cgi>

Jeffrey's Exif Viewer

From Web

From File

Image URL:

View Image At Url

CT

Basic Image Information

Target file: E_r_RECONSTRUCTED.gif

File:	480 × 270 GIF 373.899 bytes (365 kilobytes)
Gif:	Animation: 9 frames [Show Composite Frames] [Show Raw Frames]
Color Encoding:	WARNING: No color-space metadata and no embedded color profile. Windows and Mac web browsers treat colors randomly. Images for the web are most widely viewable when in the sRGB color space and with an embedded color profile. See my Introduction to Digital-Image Color Spaces for more information.

Main GIF image displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)

Animated frame #1 (65-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)

Animated frame #2 (38-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Animated frame #3 (43-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Animated frame #4 (37-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Animated frame #5 (45-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Animated frame #6 (38-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Animated frame #7 (47-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Animated frame #8 (46-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Animated frame #9 (5.7-kilobyte)
Displayed here at 94% width (88% the area of the original)



[Click image to isolate, click this text to show histogram](#)
[show composite frame](#)

Here's the full data:

GIF

Animation	yes
Frames	9
GIF Version	89a
Has Color Map	Yes
Color Resolution Depth	8
Bits Per Pixel	8
Background Color	2
Animation Iterations	Infinite
Frame Count	9
Duration	0.60 s
Image Size	480 × 270

File — basic information derived from the file.

File Size	365 kB
File Type	GIF
File Type Extension	gif
MIME Type	image/gif

Composite

This block of data is computed based upon other items. Some of it may be wildly incorrect, especially if the image has been resized.

Megapixels	0.130
------------	-------

E_r.data successfully validated.

Jeffrey's Exif Viewer

[CI

From Web

From File

Image URL:


View Image At Url

Basic Image Information

Target file: F_Z_RECONSTRUCTED.gif

File:	500 × 223 GIF 454,170 bytes (444 kilobytes)
Gif:	Animation: 6 frames [Show Composite Frames] [Show Raw Frames]
Color Encoding:	WARNING: No color-space metadata and no embedded color profile: Windows and Mac web browsers treat colors randomly. Images for the web are most widely viewable when in the sRGB color space and with an embedded color profile. See my Introduction to Digital-Image Color Spaces for more information.

Main GIF image displayed here at 90% width (81% the area of the original)



Click image to isolate, click this text to show histogram

Animated frame #1 (74-kilobyte)
Displayed here at 90% width (81% the area of the original)



Click image to isolate, click this text to show histogram

Animated frame #2 (74-kilobyte)
Displayed here at 90% width (81% the area of the original)



Click image to isolate, click this text to show histogram
show composite frame

Animated frame #3 (73-kilobyte)
Displayed here at 90% width (81% the area of the original)




Click image to isolate, click this text to show histogram
show composite frame

Animated frame #4 (73-kilobyte)
Displayed here at 90% width (81% the area of the original)



Click image to isolate, click this text to show histogram
show composite frame

Animated frame #5 (75-kilobyte)
Displayed here at 90% width (81% the area of the original)



Click image to isolate, click this text to show histogram
show composite frame

Animated frame #6 (74-kilobyte)
Displayed here at 90% width (81% the area of the original)



Click image to isolate, click this text to show histogram
show composite frame

Here's the full data:

GIF

Animation	yes
Frames	6
GIF Version	89a
Has Color Map	Yes
Color Resolution Depth	8
Bits Per Pixel	8
Background Color	255
Animation Iterations	Infinite
Frame Count	6
Duration	0.48 s
Image Size	500 × 223

File — basic information derived from the file.

File Size	444 kB
File Type	GIF
File Type Extension	gif
MIME Type	image/gif

Composite

This block of data is computed based upon other items. Some of it may be wildly incorrect, especially if the image has been resized.

Megapixels	0.112
------------	-------

F_z.data successfully validated.

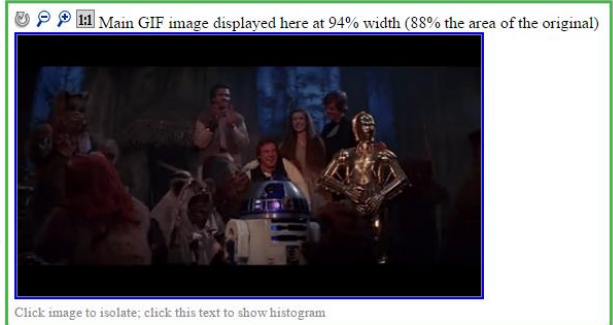
Jeffrey's Exif Viewer

☒ From Web
 ☐ From File
 Image URL:

Basic Image Information

Target file: e_r.gif

File:	480 × 270 GIF 849,198 bytes (0.81 megabytes)
Gif:	Animation: unknown
Color Encoding:	WARNING: No color-space metadata and no embedded color profile: Windows and Mac web browsers treat colors randomly. Images for the web are most widely viewable when in the sRGB color space and with an embedded color profile. See my Introduction to Digital-Image Color Spaces for more information.



Here's the full data:

GIF

Animation	unknown
Error	invalid block label found [0]
GIF Version	89a
Has Color Map	Yes
Color Resolution Depth	8
Bits Per Pixel	8
Background Color	2
Animation Iterations	Infinite
Frame Count	2
Duration	0.13 s
Image Size	480 × 270

File — basic information derived from the file.

File Size	829 kB
File Type	GIF
File Type Extension	gif
MIME Type	image/gif

Composite

This block of data is computed based upon other items. Some of it may be wildly incorrect, especially if the image has been resized.

Megapixels	0.130
------------	-------

Unsuccessful file.