ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/342040047

THE FEATURES AND ARCHITECTURE OF THE GIMP

Article · June 2020

CITATIONS
0

READS
96

1 author:

Hanisah Zainuddin
University Malaysia of Computer Science and Engineering
1 PUBLICATION **0** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

The GIMP, Open Source Software View project

All content following this page was uploaded by Hanisah Zainuddin on 09 June 2020.

The user has requested enhancement of the downloaded file.

THE FEATURES AND ARCHITECTURE OF THE GIMP

Hanisah Zainuddin [1*]

UNIMY, Cyberjaya, Selangor, Malaysia

d09190007@student.unimy.edu.my

ABSTRACT

In this article, two aspects of the open-source software, which is the GNU Image Manipulation Program known as GIMP, will be discussed. The first aspect is the features of the software. We will discuss what makes this software the best open-source software for image manipulation.

Then, we will discuss the architecture of the software for better understanding of how this software was created and the functions of each component available in the software. The architecture is done in a bottom-up manner.

Keywords: [open source, multimedia, operating systems, proprietary software]

INTRODUCTION

Overall, there are two types of software, open-source and proprietary software. Open-source software is a type of software where the source code is made available by the author for everyone to view, copy, learn, modify and share it. The source code is made by the programmers to define how the software works. Only the programmers who have access to the source code can modify the code to add, remove or improve the features of the software they are working with.

Proprietary software, also known as closed source software, does not make their source code available to everyone. The source code is made available only to the person, team and organization who created the software. Only those who maintain the software has exclusive control over it. They are the ones who can modify the source code to make improvements, add or remove features of the software. Usually, computer users have to agree to not change anything with the software without the permission of the software author. The examples of proprietary software are Adobe Photoshop and Microsoft Office.

There are many reasons why some people prefer to use open-source software rather than proprietary software. First, the users have full control of the software. With the source code made available, they can view, add or remove any parts they do not want in the software. Secondly, open-source software helps them learn how to program a software from the source

code. Thirdly, since the source code is accessible for everyone to view and modify, someone might find errors missed by the original author and able to fix it, making the software more secure. The errors can be corrected quickly due to the fact that programmers do not need permission from the author to make modification towards the software. Besides that, the community that helps maintain the OSS by removing bugs helps make the software more reliable. Finally, it helps lower the cost to be carried for students or hobbyists and will help avoid illegal copying of the original software.

INTRODUCTION OF THE GIMP

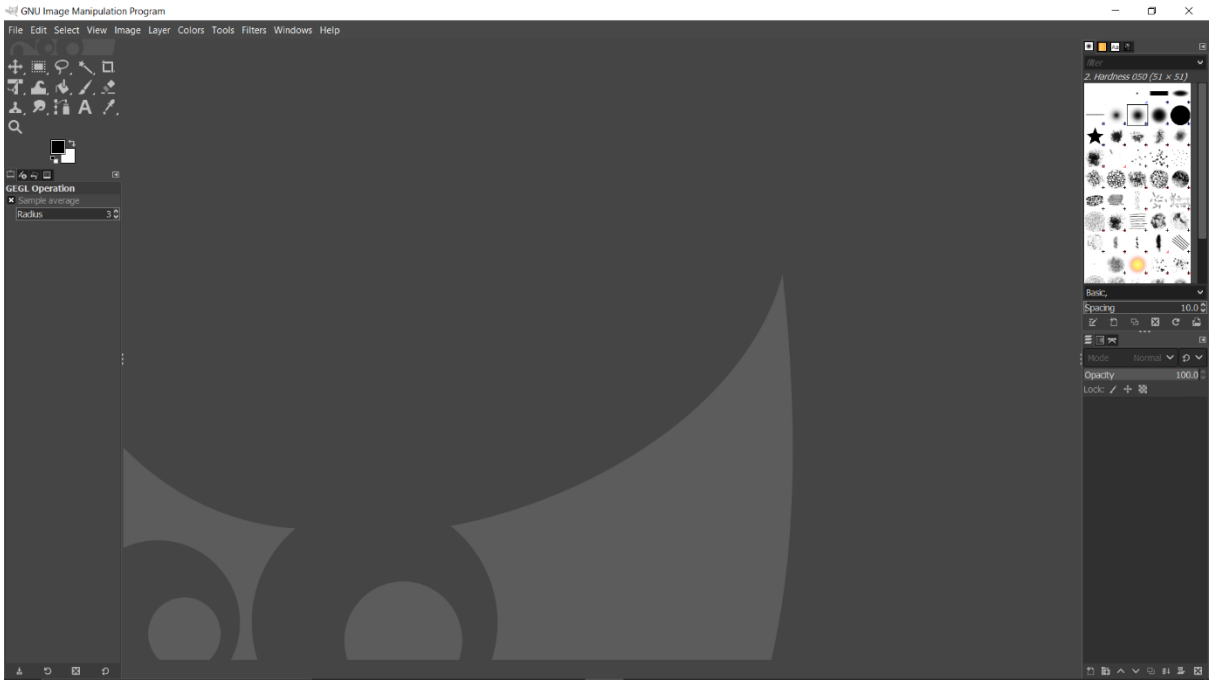

Figure 1: Windows GIMP 2.10

GIMP stands for GNU Image Manipulation Program, a type of open-source software that can be used for image retouching, free-form drawing, converting image formats, photo editing and a lot more. GIMP is also known as 'The Photoshop' of the open-source software.

This open-source software was created by Spenser Kimball and Peter Mattis as a semester-long project at their university, University of Califonia, Berkeley. This software was first published to the public in 1996. At first, the letter 'G' in the name stands for General but later was changed to GNU based on the name of the operating system produced by Richard Stallman under the GNU project when Stallman approved.

After the GIMP passed from beta development in 1998, the GIMP version 1.0 had its first stable release to the internet society. Since the released, the development of the software has grown at a rapid pace.

METHODOLOGY

In this article, there are two elements of the open-source software called GIMP will be discussed. The two elements are the features and architecture of the software.

FEATURES

The GIMP software is capable of many tasks from a simple one such as a paint program to an advanced image manipulation due to its expandable and advanced scripting interface program that supports plug-ins and extensions.

First, painting is one of the features available in the software. This feature provides a full suite of painting tools such as pencil, brush and clone. It also has sub-pixel sampling for all paint tools for high-quality anti-aliasing that helps smooth jagged lines or texture.

Second, the system is made with tile-based memory management which makes the size of the image is only limited by the availability of the disk space.

Third, the software has advanced manipulation that supports full alpha channel, layers, undo and redo. It also provides with various tools to perform the task such as transformation tool that includes scale, rotates, flip and shear, selection tools including rectangle, rounded rectangle, free and fuzzy, foreground extraction tool and advanced path tools.

Fourth, GIMP is an extensible software. It has a Procedural Database use for calling the internal function from the external programs. The fact that GIMP supports plug-ins will allow the user to add new file formats and new effect filters to the software.

The fifth is file handling. GIMP supports various file format including BMP, gif, jpeg, mng, pcx, pdf, ps, png, SVG, tiff, TGA and many more. The software may load, display, save and convert to many file formats available.

Last, because GIMP allows users to customize the interface and the behaviour of the software based on the user's need such as the widget spacing, icon sizes and toolsets in the toolbox.

ARCHITECTURE

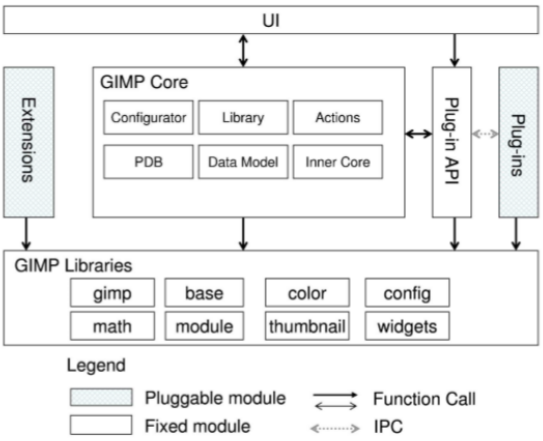

Figure 2: Overview of GIMP Architecture

The architecture included in this article was obtained through a bottom-up approach. The architecture is classified into User Interface, Extensions, Plug-ins, GIMP Libraries and GIMP Core.

The User Interface is a component that determines the presence of software through a general Graphical User Interface (GUI) layout such as the position of the menu bar and the palette. The GIMP User Interface is made up of five sub-components which are

- Dialogues: A template which let the plug-ins and the core to carry out procedures.

- Menus: The function of lists is about the same as dialogues. Through generic menu templates, the plug-ins and the core are able to differentiate the type of menus and menu items.
- Widgets: A sub-system component that carries out complex compound-widgets.
- Display: A place which holds the User Interface Components.
- Graphical User Interface: A component that plays the part of Application Programming Interface between the User Interface, the plug-ins and the core.

The Application Programming Interface is used to help the GIMP User Interface to communicate with the GIMP Core and plug-ins.

The next component is the extension which made up of 12 files that do not change the document model as its function. The Graphic User Interface and the User Interface sub-system do not interact with the extension.

The Plug-ins in the architecture is handled through the GIMP plug-in Application Interface. The plug-in Application Interface passes the plug-ins communication to the GIMP core, which cause the dialogue boxes, menus, and other procedures to take place on behalf of the plug-ins. The separation that takes place between the GIMP User Interface and the plug-ins is possible through abstract of user interface definition that defines the user interface in a textual form. This form is a common approach in many open source and modular software system.

The GIMP libraries consist of many mathematical, graphic and colour manipulation functions. All the libraries are separated and organized in each of their directories. The functions of the UI are called within the libraries.

The GIMP Core, it works on the system level tasks such as loading extensions plug-ins, configure the system and act as a central administrator for extensions and plug-ins. Basically, what the core does is it connects the components between the local layer and the whole architecture. The GIMP Core has six sub-components that complement each other in the system. The sub-components are:

- Actions
  This sub-component shows the functions that take place from the GIMP Core to the User Interface component. The Actions purpose is to hide design pattern making the interface of the GIMP Core functionality looks simple to the developers.
- Libraries
  The libraries sub-component consists of image manipulation functions for image manipulation tools such as colour picker and airbrush tool to the user interface components. All the components in the sub-components are Graphic User Interface-centric and applied with several static functions for the user interface event interaction.
- Data Model
  The Data Model is responsible to the functions that are related to the image creation and modification internally within the GIMP, so does the reading and writing of the vector and raster information in the image to the file system.
- Procedural Database
  The Procedural Database in the GIMP is executed as a hash table data structure. The libraries in sub-component are called by the Procedural Database to enrol or register a new user tool. For example, a new paintbrush tool.
- Configurator
  The main configurator task is to maintain and initialize the configuration of the components that are found in the software upon start-up and saving when required by

the user. Functions to serialize a configuration can be found in this sub-component in a LISP-like syntax.

- Inner Core
  The Inner core sub-component consist of the codes linked the parts of the GIMP together. A *GimpClass* data structure is used to store the components, and the lists of plug-ins, images, Graphic User Interface and the procedural database references available. Besides that, it also has the supporting functions which have relation to the undo, redo, memory management and error handling routines.

RECOMMENDATIONS

Although the GIMP is considered the Photoshop of open source software, most people still prefer to use the Adobe software for them to do their works. It is because the GIMP still lacks in many aspects.

First, because the GIMP is open-source software, it makes them invest less in marketing than the popular proprietary software. This caused the image manipulation software to gain less attention from the computer user.

Second, most institutions thought using the Adobe proprietary software. Students did not get much exposure to the open-source software. All the institutions have their reasons to use proprietary software such as collaboration, and it is what used in the industry, which brings us to the third point.

Photoshop has now become an industry standard. As most of us can see, most well-established organizations or designers use photoshop to do their work. They did not find it as a burden to pay for the software because they consider it as an investment with expected returns.

Fourth, photoshop integrates with other Adobe products such as adobe lightroom which makes works even more accessible to the user like a photographer.

Although many advance tasks can be done through GIMP, proprietary software such as Adobe Photoshop can always do better. The lack of GIMP support and tutorials also contributes to less interest in the user. For the first-timer, navigating and using the software might be hard without the supports. Since this software does not get much attention, not many people would do tutorials for the new users.

Finally, the mistrust of open-source software caused computer users afraid of the open-source software. Most people are having a misconception of the open-source software such as the software contains viruses and other malicious software that were attached to the software source code by irresponsible individuals.

CONCLUSION

The GIMP or GNU Image Manipulation Program is the best open-source software which offers image manipulation. With the source code made available, computer users around the world can share, upgrades and learn from this software. The software's architecture helps us understand better the functions of each component. We also get to know how all the major and sub-components work and compliments each other in the system. It also helps us discover the limitless possibilities an open-source software can have.

REFERENCES

[1] M Boricha (2020), **18 Best Open Source Softwares of 2020**

(https://techrrival.com/best-open-source-softwares/)

[2] Opensource.com Team, **What is Open Source?**

(https://opensource.com/resources/what-open-source)

[3] B Rancea (2019), **If GIMP is so Good, Why Does Everyone Use Photoshop?**

(https://ecommerce-platforms.com/articles/if-gimp-is-so-good-why-does-everyone-use-photoshop)

[4] The GIMP Team, **About GIMP**

(https://www.gimp.org/about/introduction.html)

[5] N Safie (2008). **Usage of OSS in multimedia courses: OUM experience**. FORMAMENTE, No 3-4, Anno 3. 249-260.

[6] The GIMP Team, **Feature Overview**

(https://www.gimp.org/features/)

[7] E Oliver, J Ruiz, S She and J Wang (December 7, 2006), **The Software Architecture of the GIMP**

View publication stats