ગુજરાત રાજ્યના શિક્ષણવિભાગના પત્ર-ક્રમાંક
મશબ/1213/110/છ, તા. 18-3-2013-થી મંજૂર

COMPUTER STUDIES

Standard 11

PLEDGE

India is my country.
All Indians are my brothers and sisters.
I love my country and I am proud of its rich and varied heritage.
I shall always strive to be worthy of it.
I shall respect my parents, teachers and all my elders and treat everyone with courtesy.
I pledge my devotion to my country and its people.
My happiness lies in their well-being and prosperity.

Price : ₹ 96.00

Gujarat State Board of School Textbooks
'Vidyayan', Sector 10-A, Gandhinagar-382010

Subject Adviser

Prof. R. P. Soni

Authors

Dr Harshal Arolkar (Convener)
Dr Sonalben Jain
Dr Nilesh Modi
Shri Triptiben Dodiya

Reviewers

Dr Kuntal Patel
Shri Jyotika R. Doshi
Shri Girish S. Brahmbhatt
Shri Saket A. Dave
Shri Ankita S. Dwivedi
Shri Bimal K. Raval
Shri Mayuri I. Shah
Shri R. L. Gilatar
Shri Rajnikant P. Pandya
Shri Pankaj R. Shukla
Shri Nishita N. Gandhi
Shri Rajshree N. Padia
Shri Dimple U. Shukla

Co-ordinator

Shri Ashish H. Borisagar
(Subject Co-ordinator : Mathematics)

Preparation and Planning

Shri Haren Shah
(Dy. Director : Academic)

Lay-out and Planning

Shri Haresh S. Limbachiya
(Dy. Director : Production)

PREFACE

The Gujarat State Secondary and Higher Secondary Education Board has prepared new syllabi based on the open source operating system and compitible open source software tools for various topics of Computer Studies. These syllabi are sanctioned by the Government of Gujarat.

It is a matter of pleasure for the Gujarat State Board of School Textbooks to place this textbook of **Computer Studies** before the students of **Standard 11** prepared according to the new syllabus.

Before publishing the textbook, its manuscript has been fully reviewed by experts and teachers teaching at this level. Carrying out suggestions given by teachers and experts, we have made necessary changes in the manuscript and then have published the textbook.

The board has taken special care to ensure that this textbook is interesting, useful and free from errors. However, we welcome suggestions to enhance the quality of the textbook.

P. bharathi (IAS)
Director
Executive President
Date : 13-12-2019
Gandhinagar

First Edition : 2013, Reprint : 2017, 2018, 2019, 2020

Published by : P. Bharathi, Director, on behalf of Gujarat State Board of School Textbooks, 'Vidhyayn', Sector 10-A, Gandhinagar.

Printed by :

FUNDAMENTAL DUTIES

It shall be the duty of every citizen of India :*

(a) to abide by the Constitution and respect its ideals and institutions, the National Flag and the National Anthem;

(b) to cherish and follow the noble ideals which inspired our national struggle for freedom;

(c) to uphold and protect the soverenty, unity and integrity of India;

(d) to defend the country and render national service when called upon to do so;

(e) to promote harmony and the spirit of common brotherhood amongst all the people of India transcending religious, linguistic and regional or sectional diversities; to renounce practices derogatory to the dignity of women;

(f) to value and preserve the rich heritage of our composite culture;

(g) to protect and improve the natural environment including forests, lakes, rivers and wild life, and to have compassion for living creatures;

(h) to develop the scientific temper, humanism and the spirit of inquiry and reform;

(i) to safeguard public property and to abjure violence;

(j) to strive towards excellence in all spheres of individual and collective activity so that the national constantly rises to higher levels of endeavour and achievement;

(k) to provide opportunities for education by the parent or the guardian to his child or ward between age of 6 to 14 years, as the case may be.

* Constitution of India : Section 51-A

CONTENTS

About This Textbook...

Dear Teachers,

With a mission to spread computer literacy on a fast track, the Gujarat Government has provided latest computer equipments to more than 6000 aided schools under the ICT@School program. As a new policy initiative all the schools are given the Ubuntu (a variant of Linux) Operating System and other Open Source software packages so that schools can use and exchange the software freely without botheration of the licensing issues. Since earlier text books were largely based on proprietary software, there was a need to rewrite the text books based on new syllabus. This was also necessary in view of the fact that the 8th standard has been transferred to primary section. Therefore, new content has been provided for 9th to 12th standard in a phased manner based on the Open Source Operating System and compatible Open Source software tools for various topics of computer studies.

This text book for 11th standard is the third in series for the subject of 'Computer Studies'. In this text book, the aim is to provide elementary knowledge of multimedia, introduction to animation using Open Source multimedia tool on Linux known as Synfig, discuss essential Ubuntu Linux commands for various operations, introduction to Shell scripting, data base operations on Open Office Data Base Management System tool known as Base, and an overview of certain emerging technologies which may gain ground in the days to come.

We hope the coverage will be useful to the students to get insight into computer applications and you will enjoy teaching and conducting practical using open source software tools.

Dear Students,

We assume that you are familiar with Open Source Operating System Ubuntu Linux and Open Office components. In this text book you will be learning fundamentals of multimedia, basic Linux commands for file and general operations, working with editors and Shell scripting for elementary administrative tasks. You will also learn how to create database, write queries for information retrieval and generate forms and reports. Finally, few emerging technologies and trends will be discussed so that you would become aware of the information technology which is likely to take in close future.

The chapters 1 to 5 cover introduction to multimedia, features of multimedia tool Synfig, creating animation using Synfig, advanced concepts of animation using layers and what can be achieved by superimposing layers, how to handle pictures and images in Synfig tool. Chapters 6 introduces basic Linux commands for general purpose operations, file and directory manipulations, i/o redirection and pipe as well as for sorting of records. Chapter 7 covers the Vim editor for creating and editing files followed by introduction to shell scripting. In chapter 8 several examples of Shell scripting have been presented so that you can develop confidence and write scripts for elementary administrative tasks.

Chapter 9 and 10 introduce the Open Office tool Base for creating database (tables) and populating the created tables with actual data. Chapters 11 and 12 discuss writing queries for data retrieval and creation of forms as well as reports for output presentation. The last chapter 13 deals with few emerging technologies and trends such as Mobile Computing, Cloud Computing, RFID, Biometrics, and Green Computing among others. We are sure these will make you reading and learning of current and future trends in computing interesting.

It is sure that if you carefully study the text and practice the laboratory exercises, you will develop reasonable confidence in working with Synfig animation tool, writing Shell scripts, creating data base and generating forms, reports and queries for information retrieval.

1 Introduction to Multimedia

Communication is an integral part of our life. We use various means of communication like radio, newspaper, television, theatre, movies, internet and others. All these medium help us gain knowledge or educate us. A single medium of information is just not enough to convey meaningful messages to variety of audiences. For example while talking on telephone to a friend, we can hear his voice but cannot see his facial expressions. When you write letter to a friend describing him about your trip to Kashmir, only the text can be read. You cannot hear the voice of the person. If you send him a picture along with the letter, he can imagine the fun you had during your trip. However, if you send a video clip, he could visualize more about the fun you had. As you can see, the more mediums of information you use the impact of the communication increases. Multimedia is all about adding various effects to make communication effective. In this chapter we will learn about multimedia, basic elements of multimedia and applications of multimedia.

Multimedia

The term multimedia has been coined from two terms : multiple and media. Hence multimedia means usage of multiple medias to communicate. In other words, multimedia is a combination of different media elements like text, audio, graphics, video and animation. The combination of all the media elements makes the communication more structured and understandable to the user. Multimedia is now-a-days used in every field like education, theatre, advertising, fashion, gaming to name a few. Figure 1.1 shows different components of multimedia.

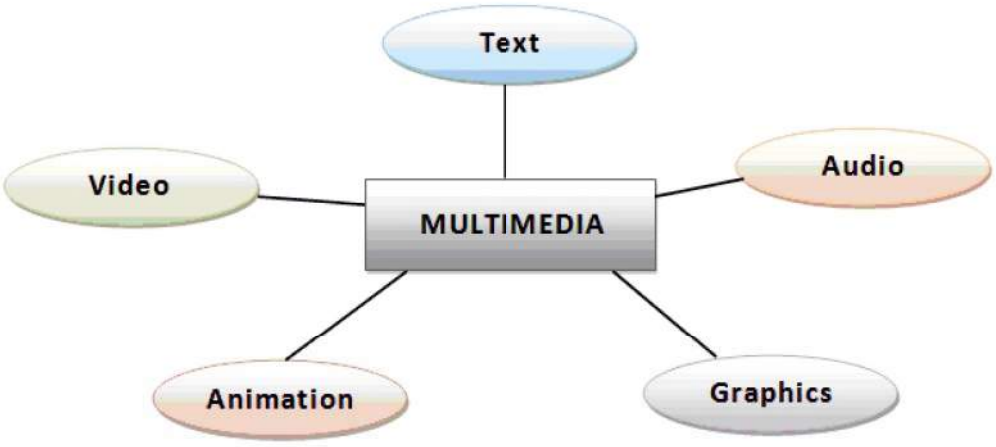


Figure 1.1 : Components of Multimedia

Computers play a vital role in the development of these media elements. A computer capable of handling text, graphics, audio, video and animation is sometimes called multimedia computer. Figure 1.2 shows use of the various elements of multimedia.

Figure 1.2 : Use of elements of Multimedia

Elements of multimedia

We defined multimedia as combination of various media elements like text, audio, graphics, video and animation. Let us now describe each element.

Text

Including text in multimedia is the basic step towards development of multimedia presentation. Text is used to communicate information to the user. Proper use of text and words in multimedia presentation will help to communicate the idea and message to the user. Text can be of any type, a word, a single line, or a paragraph. The textual data for multimedia can be developed using any text editor. However to give special effects we need graphics software. We can even use word processing software like OpenOffice Writer or MS-Word to create textual data. The text can have different type, size, color and style to suit the professional requirement of the multimedia software.

Observe that the sample shown in figure 1.3 does not look the same. This is because they have a different typeface, font and size.

GUJARAT

GUJARAT

GUJARAT

Figure 1.3 : Sample Text

A typeface is family of characters or letters which have similar look. For example : Times, Arial, Courier and others. Typefaces have different style and size known as Font. Styles are Italic, Bold, Underline and others. Size is the distance between the top of the letter to the bottom point. It is expressed in points for example 12 point, 20 point and so on. For example, if we have a font of 'Times new roman 14 point', then Times new roman is style and 14 point is size.

Typeface is categorized into : Serif and Sans Serif. Serif is a little decoration at the end of the character. Times, Century, Bookman are some examples of serif fonts. Serif fonts are easy to read on a printed page as they guide the eye along the line of text. So they are usually used when there is a lot of text to be read. On the other hand, as the name suggests, Sans Serif (sans in French means 'without') are without the decoration. Arial, Verdana and Helvetica are some examples of Sans Serif. They are usually used for headlines and titles. But on computer screen due to low resolution, serif fonts are difficult to read than the sans serif. Figure 1.4 shows the sample of typeface.

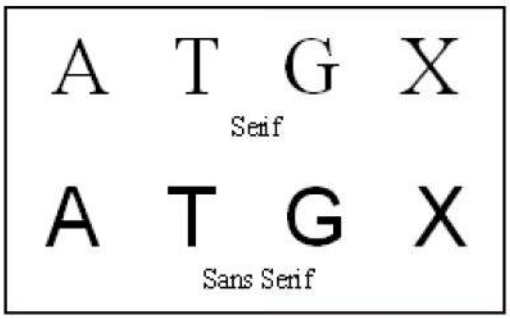


Figure 1.4 : Sample Typeface

Text in multimedia is different from the text used in traditional pen and paper method. In traditional method we convey the information using only text while in multimedia we have other elements along with text. And these elements work together to convey meaningful information. Thus we can say that the information is in the form of text, sound, pictures etc. It should be presented to the user in such a way that he/she is not confused. Thus the text in multimedia should be short, to the point and attractive to get the attention of the user.

Assume that, we want to make a presentation on "Gujarat Tourism" using OpenOffice Impress. First, we need to create a main page with the title "Gujarat Tourism". We can simply type the text but to make it more attractive we can use Fontwork. The steps to create the text are as mentioned :

- Open a blank presentation in OpenOffice Impress.
- Click on View → Toolbars → Drawing. This will open the drawing toolbox.
- On the *Drawing* toolbar click on the *Fontwork Gallery* icon  . A *Fonwork Gallery* dialog box as shown in the figure 1.5 will open.

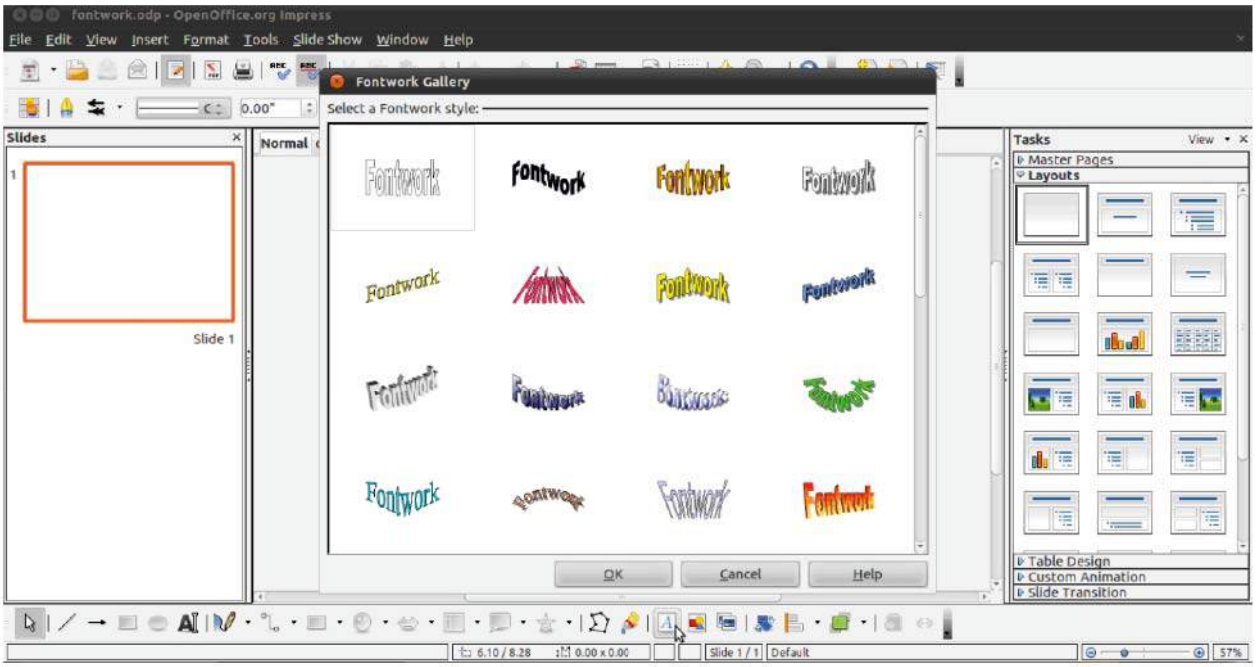


Figure 1.5 : Fontwork Gallery dialog box

- In the *Fontwork Gallery* dialog box, select any style that you like and click on the OK button. The Fontwork object is inserted into the document as shown in figure 1.6.
- Double click on the object to enter into the text edit mode.

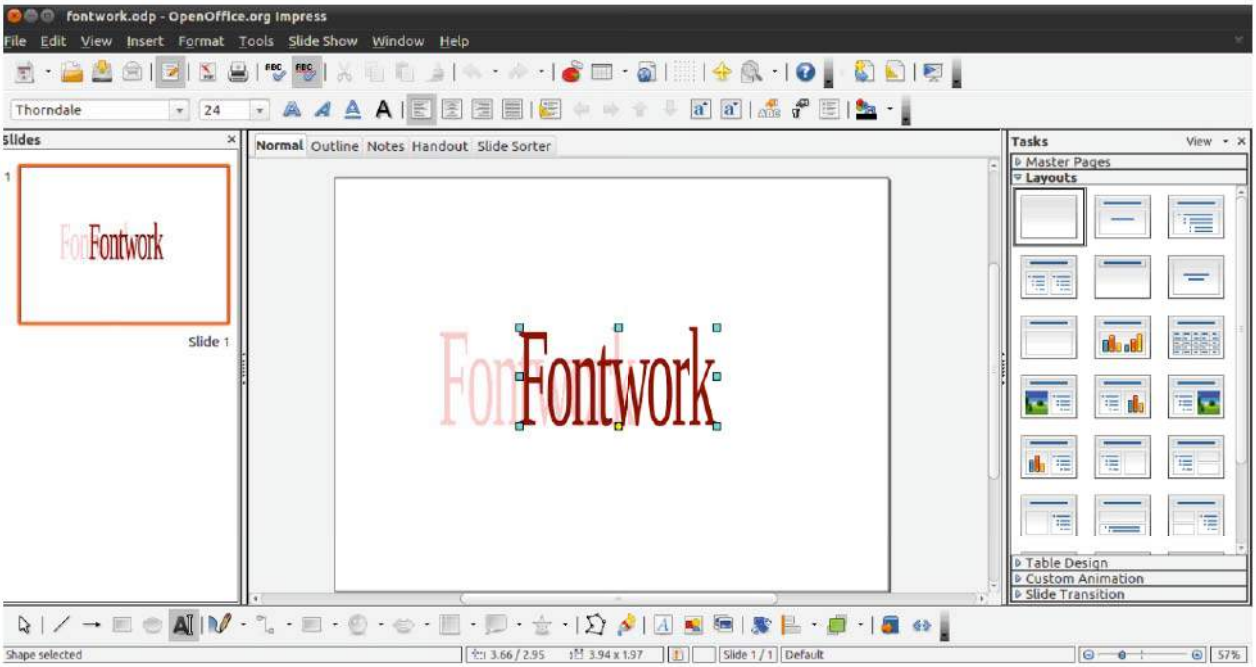


Figure 1.6 : Edit the fontwork text dialog box

- Replace the default "Fontwork" text with text "Gujarat Tourism".
- Press ESC key to exit the text edit mode.

- You can now see that the text "Gujarat Tourism" appears on the slide as shown in the figure 1.7.
- Save the file for further use.

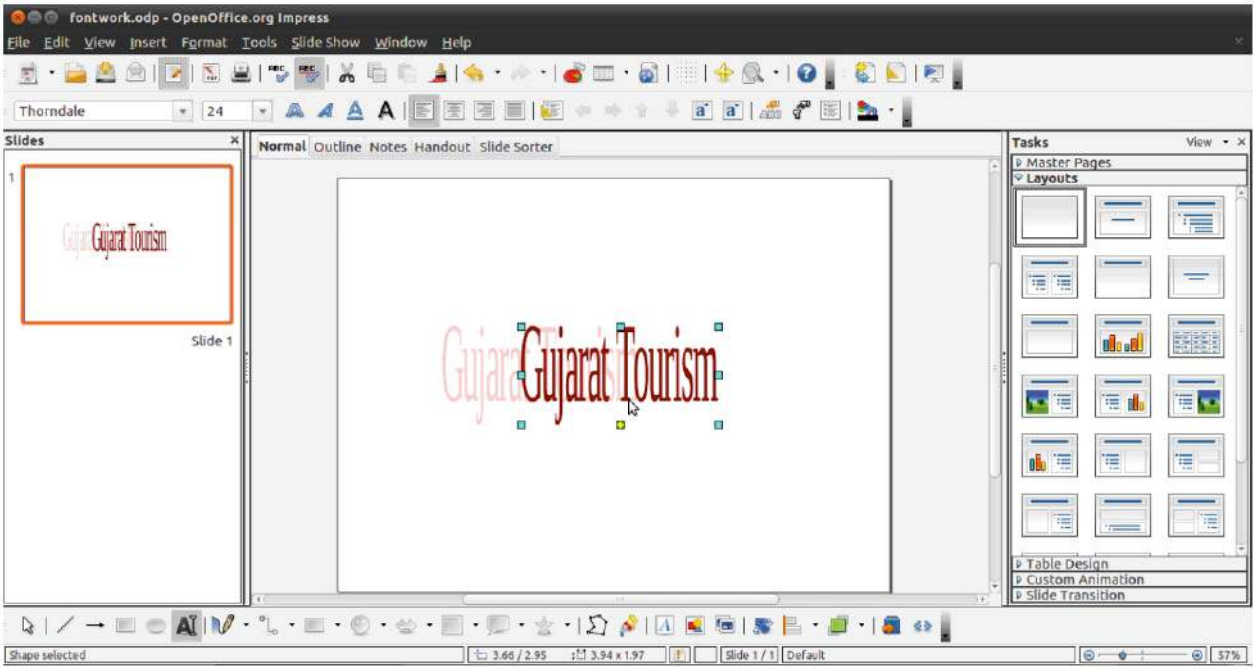


Figure 1.7 : Final slide with the text

Audio

Sound is the most important element of multimedia. We can define sound as a meaningful speech in any language. The term sound is used in the analog form, and the digitized form of the sound is called audio.

Text and images without any sound are helpful when we are presenting in front of the audience. But in the absence of the presenter, sound plays an important role. The best example is when we are viewing an educational CD. The user can enable or disable the sound option. But the learning process is more enjoyable when it is played with sound. To make such a kind of multimedia presentation, we need to first record the voice and play it according to the animation in the presentation. Thus when a user sees the presentation he/she is able to understand it easily by seeing the animation along with the voice.

Now-a-days, computers support multimedia applications. They come along with required hardware and software needed for multimedia applications. We get inbuilt sound card, speaker, microphone, and web camera which are required for multimedia applications. The operating system provides application that can be directly used for capturing audio and video.

Let us have a look at an audio recording application provided in Ubuntu Linux. Be sure you have microphone and speaker connected to the computer. Let us record our voice using the application by following the steps given :

- Select Applications → Sound and video → Sound recorder. This opens a *Sound Recorder* application as shown in figure 1.8.
- Choose File → New.
- Use the Record as drop-down list to select one of the following recording options :
  - CD quality, Lossless
  - CD quality, Lossy
  - Voice

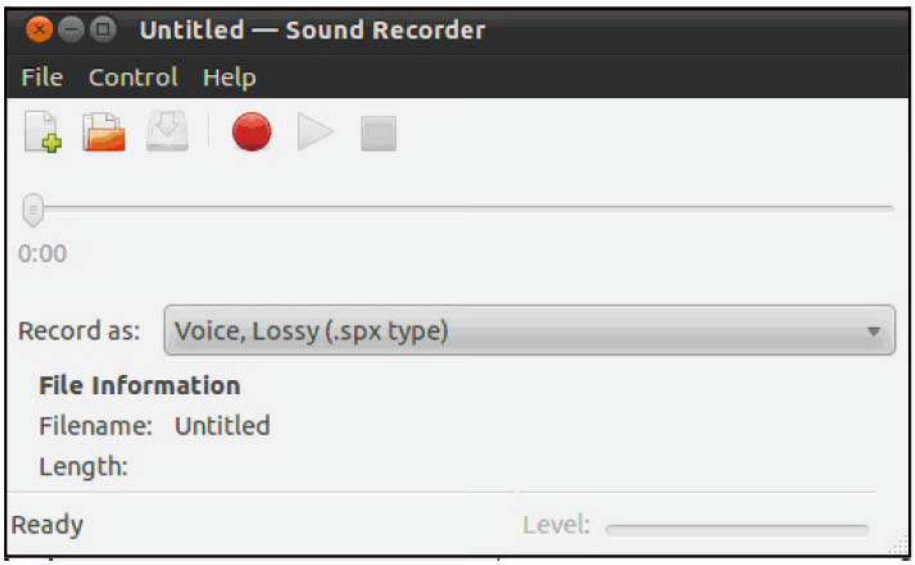


Figure 1.8 : Sound Recorder

- To start recording, choose Control → Record or click on record button  as in shown figure 1.8.
- Now start recording the voice. For example, describe a tourist place of Gujarat for our multimedia presentation on "Gujarat Tourism". Figure 1.9 shows the audio recording process.

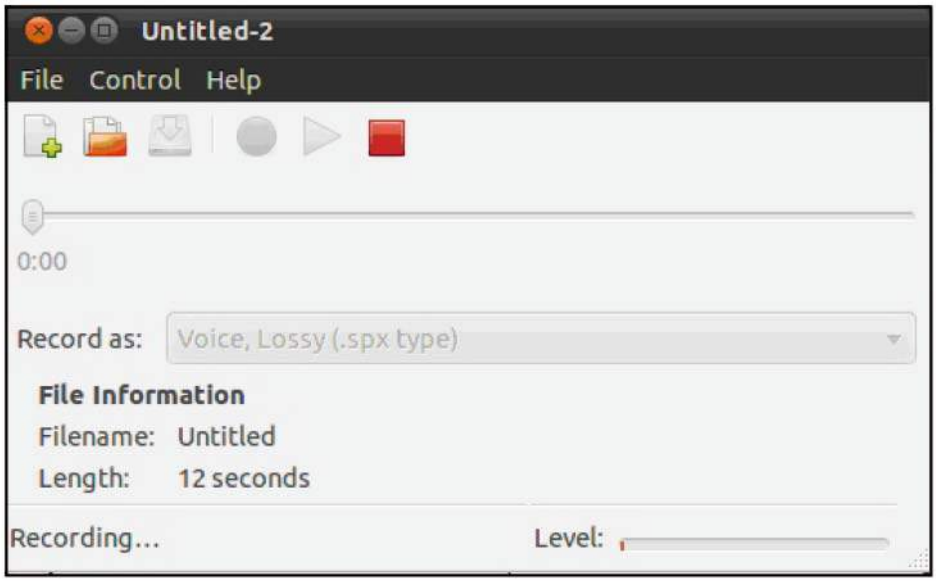


Figure 1.9 : Audio Recording

- To stop recording, choose Control → Stop or click on Stop button ( ).
- To play back the recording, choose Control → Play or click on Play button ( ).
- To run the audio mixer, choose File → Open Volume Control.
- To save the recording, choose File → Save As, and type a name for the sound file.

We can also make changes to this sound file like, delete a part of it, change the playback speed, change the playback volume and add echo etc. Multimedia elements like sounds or videos are stored in media files. To discover the type of media file we should look at the file extension. The commonly used audio formats are described in table 1.1.

File Extension	Type of File	Description
.mid, .midi	MIDI File	MIDI (Musical instrument digital interface) files contain music data.
.rm, .ram	Real Audio File	.ram (real audio metadata) is combination of audio and video.
.wav	Wave File	Waveform audio file.
.wma	Windows Media Audio File	Audio file compressed with windows media compression.
.mp3, .mpga	MP3 Audio File	Compressed audio format.

Table 1.1 : Commonly used Audio File Extension

.wav is the most popular uncompressed sound format on the Internet, and it is supported by all popular browsers. The .mp3 is the new compressed format for recorded music. So if you want to use the recorded music, .mp3 is the best choice.

Image

Generally multimedia presentations are graphics/image based. Information communicated through images is easier to remember and understand. Images used in a multimedia application can be photographs converted into digital form with the help of scanners or generated on the computer. Computers store the pictures in the form of pixels maps also known as bitmap or raster images. A Pixel (short for picture element) is the small dot on the screen. A map is a two dimensional matrix of these dots. Thus, a bitmap is a simple matrix of tiny dots which forms an image on the computer screen. Each pixel contains values representing a color. When an image is sent to be displayed on the screen, the picture data is converted to pixels.

Another form of image generated by computers is vector graphics. They store the images in the form of mathematical equations. On the screen both bitmap and vector graphics looks the same. But when you enlarge the bitmap image, it will blur and the pixels will get enlarged as shown in figure 1.10. The vector images on the other hand looks the same when enlarged. This is because,

the data is stored in the mathematical form and the values are recalculated when displayed on the screen.

Bitmap image

Enlarged bitmap image showing pixels

Figure 1.10 : Bitmap Image

Both the types of images are stored in different file formats as mentioned in table 1.2. Typically, these files are compressed to save memory space. Bitmap image file formats like gif, jpeg and png use compression within the file itself. The commonly used image formats are described in table 1.2.

File Extension	Type of File	Description
.bmp	Bitmap Image	Uncompressed image file used to store bitmap digital images.
.gif	Graphical Interchange Format File	Common for web graphics with small images and images with text. Uses limited number of colors.
.png	Portable Network Graphic	Used to store graphics for web images and supports upto 32-bit colors. It was an improvement over gif.
.jpeg / .jpg	Joint Photographic Expert Group Image File	Common image format used by digital cameras. Supports upto 24-bit colors.
.psd	Photoshop Document	Image file created by Adobe Photoshop.
.tif	Tagged Image File Format	Highly flexible and platform independent format which is widely used today.

Table 1.2 : Commonly used Image File Extension

Using inbuilt images is a common practice while preparing a presentation. For example, in Open office Writer, we can insert a picture by going to Insert → Picture. The pictures used can be a

file already present in the computer or we can scan the picture using a scanner and convert it into digital format to be used further by the computer. Figure 1.11 shows a document with an image inserted using OpenOffice Writer.

The image used in figure 1.11 was already available to us. We may use any other image of our liking. We can also make changes to our images by using programs like Adobe Photoshop, GIMP, MS-Paint and others.

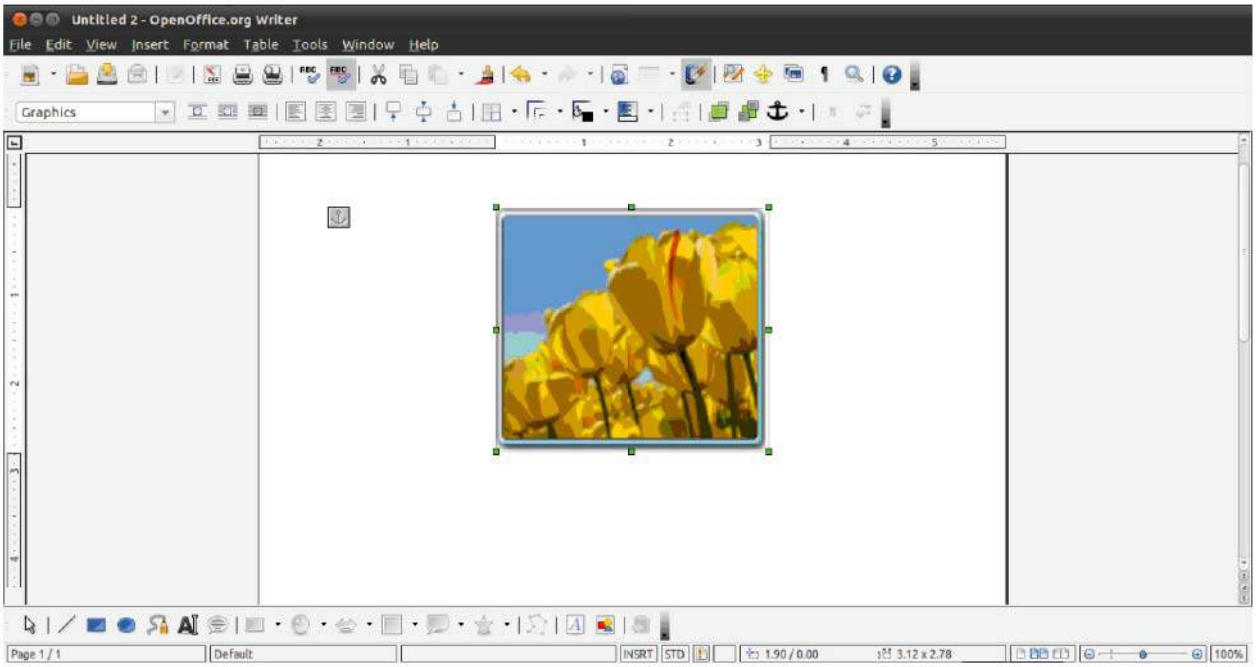


Figure 1.11 : Document with Image

Video and Animation

Animation can add great power to any multimedia application. Any static presentation becomes lively by adding a video or animation. Let us first differentiate between animation and video. The term video refers to the sequence of natural scenes captured using analog or digital video capturing device. This device can be a web camera, digital camera or even mobile phones.

Animation is a visual change over time. The digital images are played one after the other to create a moving effect. We can say that, animation is created from drawn pictures and video is created using real time visuals. Carefully planned and well-executed video clips or animations can make a great difference in a multimedia application. Some examples of animation are movies like Kung Fu Panda and Smurfs.

Animations can be classified as two-dimensional (2D) or three dimensional (3D). In 2D animation the visual change occurs on the x and y axis of the screen. This type of animation is simple. Some examples of 2D animation software are Macromedia Flash, Synfig studio and Pencil. In 3D animation the visual change occurs along three axis namely x, y and z. This type of animation gives almost

a realistic view of the image as can be seen by the human eye. Some examples of 3D animation software are Maya, Blender and 3D Max. Figure 1.12 shows the view of an object in 2D animation software and figure 1.13 shows the view of an object in 3D animation software. We will learn how to create animation in later chapters.

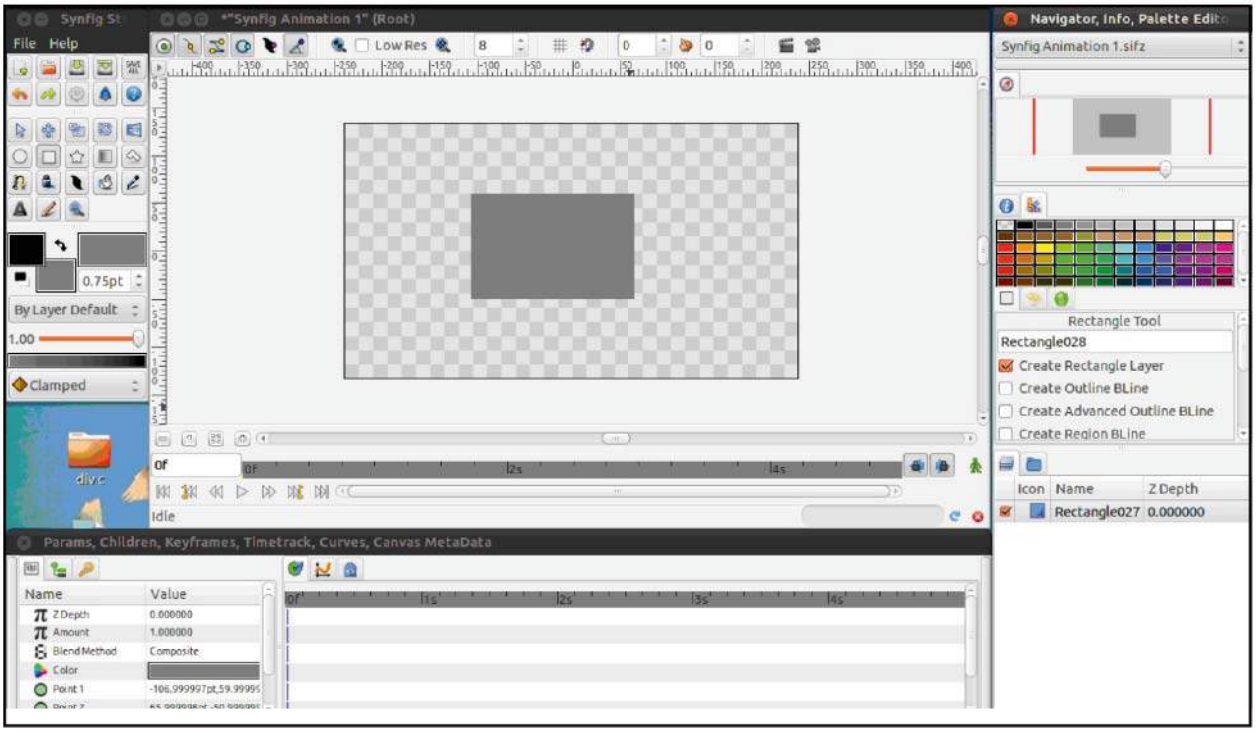


Figure 1.12 : 2D view of an object

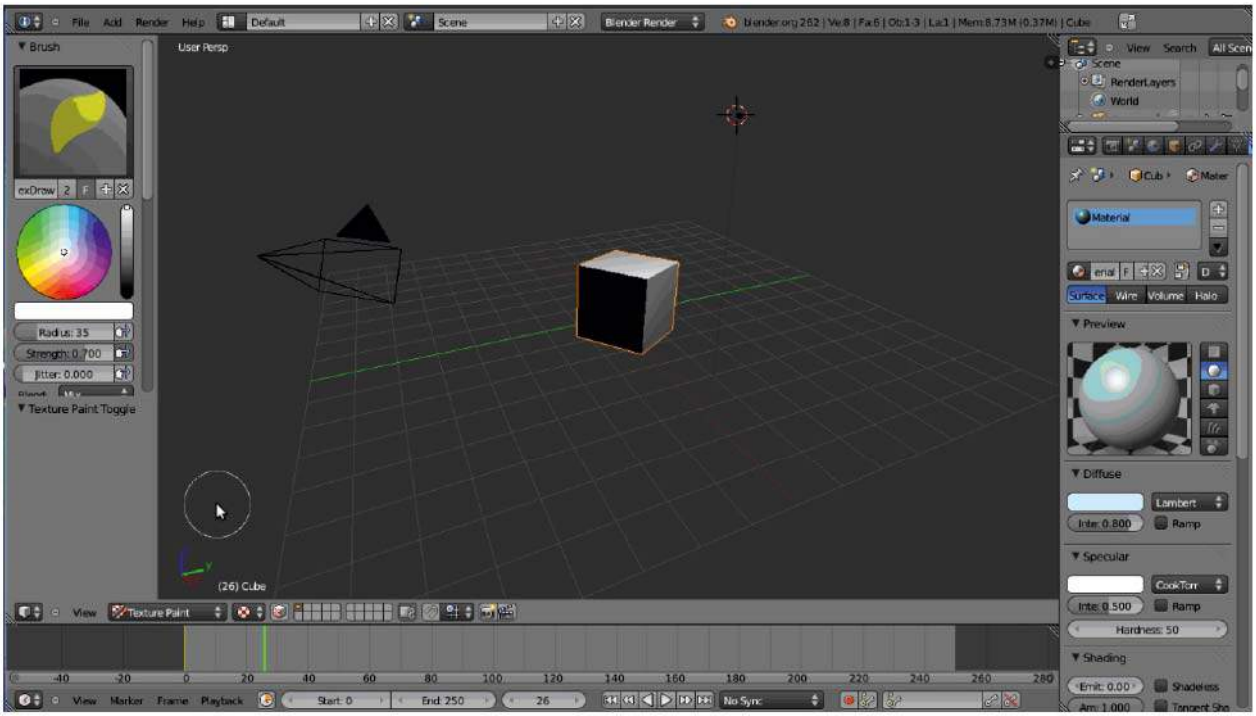


Figure 1.13 : 3D view of an object

Let us now see how to record a video using VLC media player. It is essential that you have VLC Media Player installed in your computer, if not then download it from Ubuntu Software Center. It is a free open source multimedia player. Be sure you have webcam connected to the computer. Let us start recording a video by following the steps given :

- Open VLC media player.
- Select Media → Open Capturing Device. This will take you to the Open Media dialog as shown in figure 1.14. Select the video device name and audio device name. Figure 1.14 shows the selected options.

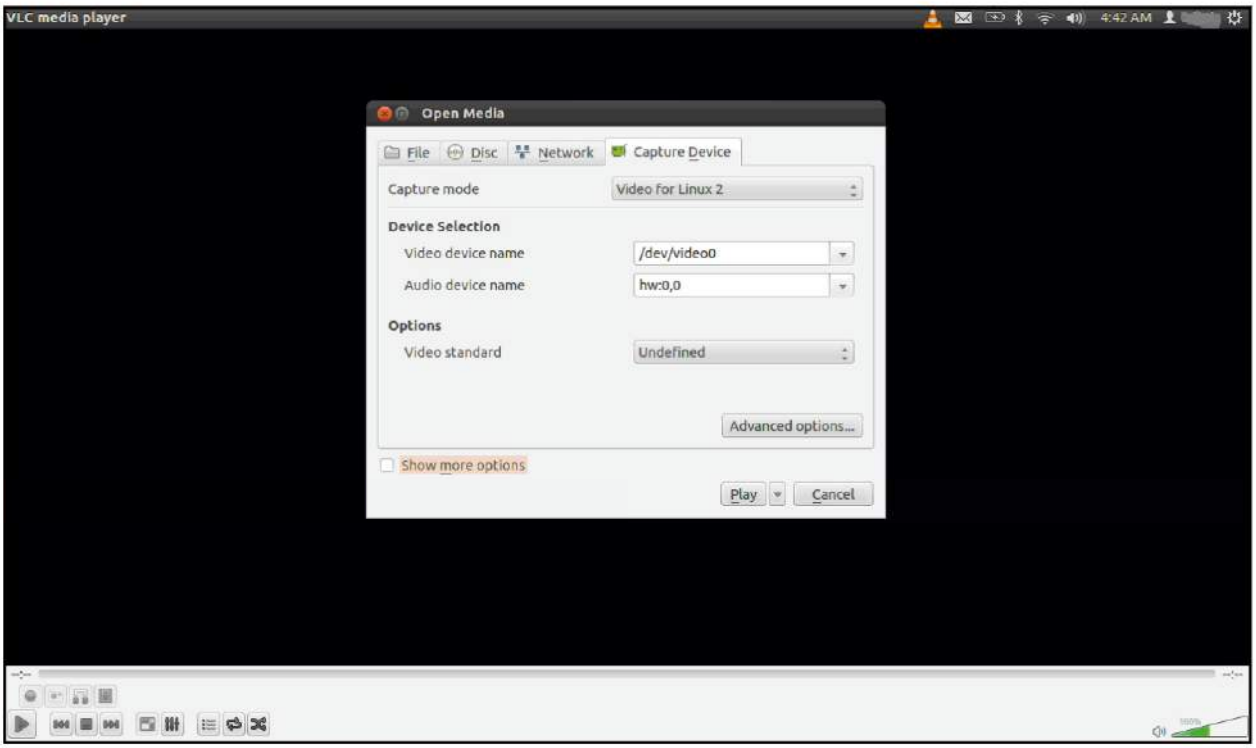


Figure 1.14 : Open Media dialog box

Note :

The device name and the audio device name may vary as per the settings of reader's machine. In case you have multiple devices you can choose the one that you would like to use.

- From the Play drop down menu visible in the *Open Media* dialog box of figure 1.14, select Convert option. This will open the *Convert* dialog box as shown in figure 1.15. In the textbox after *Destination file* label, select the destination and filename. Check 'Display the output' checkbox to see what you are recording.
- Click on the Start button.

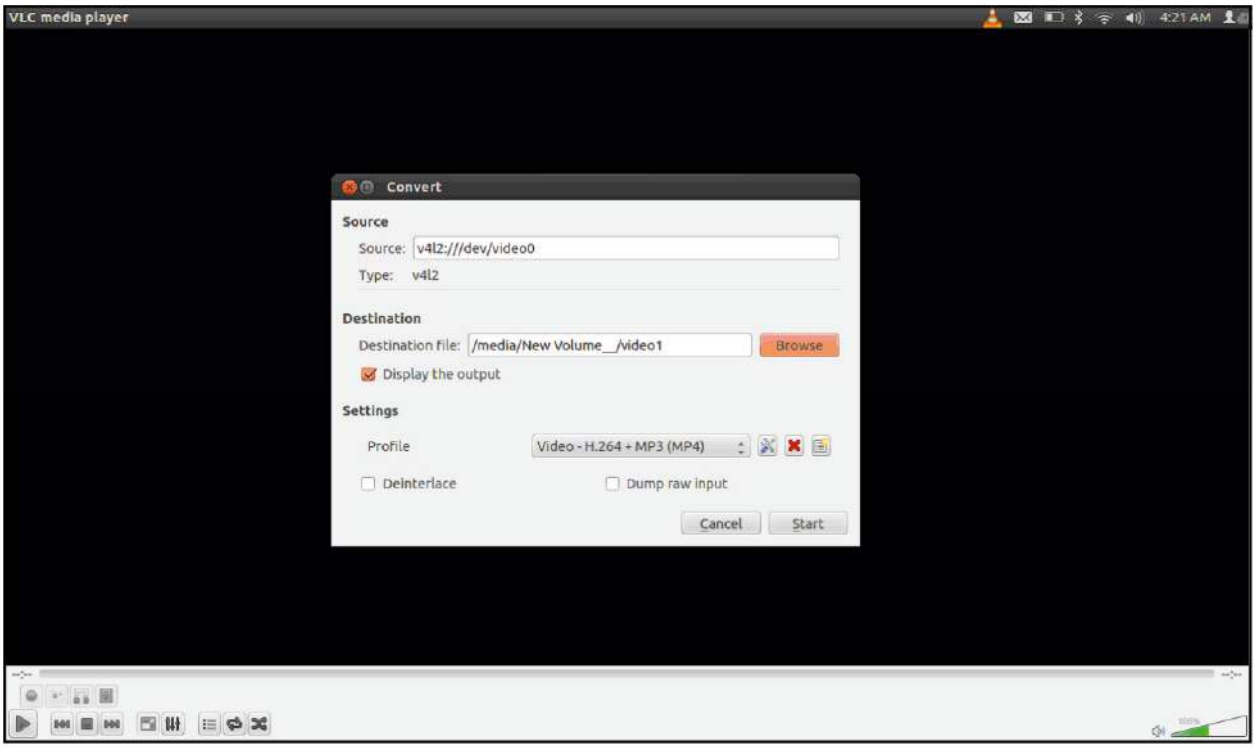


Figure 1.15 : Save the Video file

Figure 1.16 shows the video recording in progress. Click on the Stop button  to stop the recording.

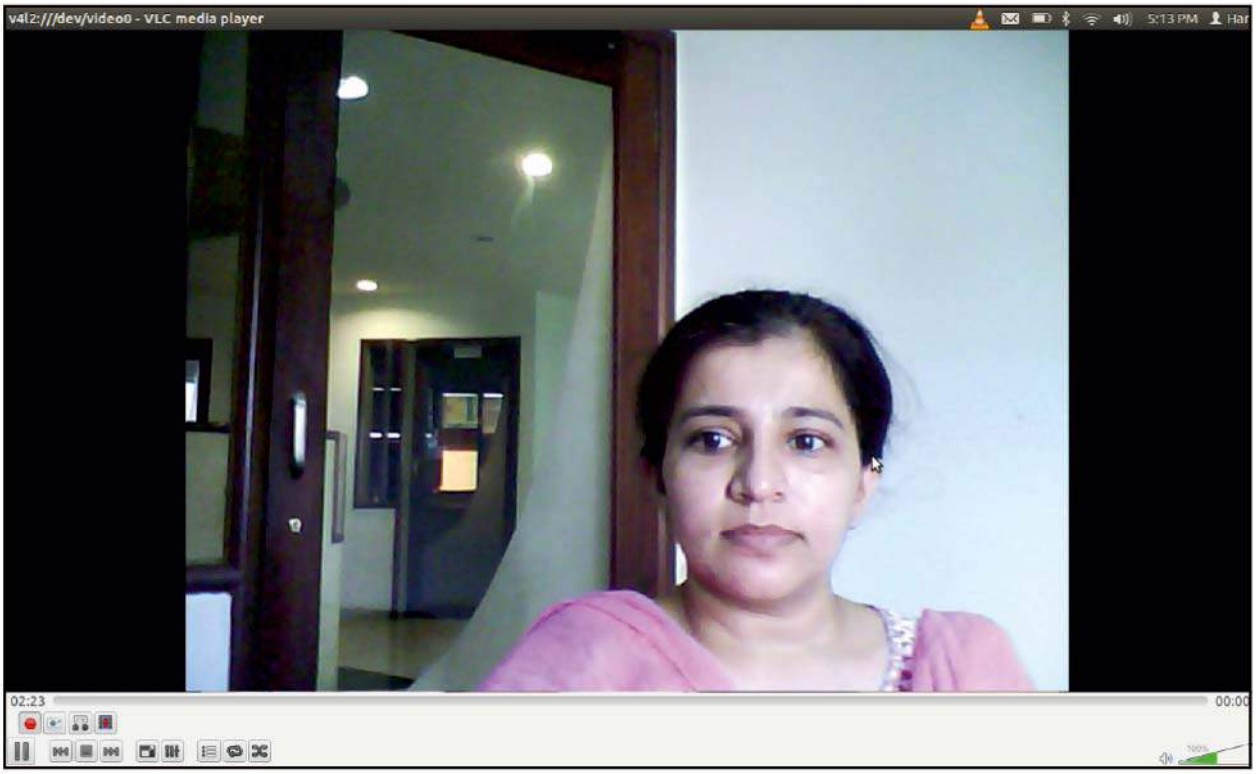


Figure 1.16 : Video recording in progress

Later the recorded video can be attached to the multimedia application. You can also try out another open source video recording software called "Cheese Webcam Booth" available in Ubuntu Software Center. The commonly used video formats are shown in table 1.3.

File Extension	Type of File	Description
.avi	Audio Video Interleave File	Developed by Microsoft to play videos in windows environment.
.wmv	Windows Media Format File	.wmv (windows media video) is a compressed video format developed by Microsoft for internet streaming applications.
.mpg, .mpeg	Moving Pictures Expert Group File	Popular video format used for creating movies distributed over internet.
.mov	Quick Time Format File	Developed by Apple. Files are compressed and are easier to download from internet.
.rm, .ram	Real Video Format File	.ram (real audio metadata) are combination of audio and video.
.swf, .flv	Adobe Flash Shockwave File	Animated file containing text and images. Created by using Adobe Flash software.
.mp4	MPEG4 Video File	Commonly used for sharing video files on the internet.

Table 1.3 : Commonly used Video File Extension

Today mp4 is most widely used format for the Internet video. It is supported by Flash players, YouTube as well as HTML5.

Interactive objects

For a multimedia presentation to be effective it should have maximum interactivity. At times the user may want to skip a portion of the presentation or want to again go through that same portion. The user is given buttons or hyperlinks for navigation in such cases. When these interactive objects are clicked, the flow of the presentation changes as per the user's choice.

Earlier the Internet browsers supported only one media element : text. Text was also limited to single font and color. All the media elements described earlier are now supported by latest Internet browsers in different ways. Some media elements can be handled inline and some require plug-in. Inline media objects are specified inside the program using a specific <inlinemediaobject> tag. Plug-in are extra helper programs that can be easily installed and used as a part of the web browser. The extent to which a particular media element is employed is determined by the nature

and scope of the project. For example, multimedia project on "Gujarat Tourism" may use more images, video clips and text about various places of Gujarat. We will learn more about these elements in later chapters.

Classification of Multimedia

Generally multimedia is classified into two namely : Interactive and Non-Interactive multimedia.

In Interactive multimedia, the sequence and timing of media elements can be controlled by the user. For example, a multimedia CD on "Learn Musical Instruments", the user can select on the different musical instruments he/she wants to play. User can also select his favourite song which he wants to play using that instrument. Most of the CD-ROM titles and games available in the market are interactive in nature. Generally, interactivity increases the user participation and he/she enjoys it more. The degree of interactivity depends upon the field of the multimedia project.

In traditional mass media like television, radio or newspaper the communication is one-way. In other words we can say that the process originates from the source and is delivered to the mass audience. For example, a movie has a beginning and an end irrespective of the user watching it on the television or not. Though these technologies also use video, audio, text and graphics but in an inflexible way. While in multimedia the user does not remain passive but can control the elements. Thus, the difference between mass media and multimedia is that there is a shift from audience to user and the communication becomes two-way instead of one-way.

In Non-Interactive multimedia the user simply watches the media as it plays from beginning to the end. He/she has no control over the flow. For example : a corporate presentation or a multimedia demo.

Another classification of multimedia is based on its applications : Entertainment or Edutainment.

All types of games, movies are examples of Entertainment multimedia wherein the purpose is to entertain the user. Whereas the multimedia titles that educate the user are classified under edutainment multimedia. All types of educational CD's are example of edutainment multimedia.

Usage of Multimedia

Multimedia has become an integral part of our life. Multimedia finds its application in various areas including, but not limited to, advertisements, art, education, entertainment, engineering, medicine, mathematics, business and scientific research. Let us discuss some of the usage of multimedia in various fields :

Education and Training

In the area of education, multimedia has a great importance. We have been using presentations consisting of charts, tables and other objects to impart knowledge since a long time. But now-a-days the classroom education is not limited to the earlier conventional method rather it needs audio and visual media. Multimedia is used to produce computer based training courses commonly known as CBT. A CBT lets the user go through a series of presentations, animations, text about a particular

topic along with examples. Today CBT are used for almost all the age groups right from kindergarten to post graduates. For example, edutainment CD like Nursery Rhymes or Learn Computers. Edutainment is an informal term used to describe combining education with entertainment, especially multimedia entertainment.

Multimedia is used extensively in training programs. Medical trainers and doctors can practice surgery methods via simulation before the actual surgery. Mechanics can learn to repair the engines or a salesperson can learn about the product details.

Some of the uses of multimedia in education and training are as mentioned below :

- Recording or broadcasting lectures.
- Using video conferencing we can hear an expert speaker from a distant location.
- Demonstrating surgeries or other techniques that learners may not otherwise have the opportunity to see and later put them in practice.
- Record student's performance to enable feedback.

Advertisements

In the field of advertising multimedia plays a vital role. Today, advertising is the major source of launching and promoting a product by the manufacturing companies in the market. Exciting animations, effects and slogans can make an advertisement popular. This can lead to promotion of the product. We see so many advertisements today and some of them catch our attention and we ultimately end up buying that product. For example, Jujus and pug dog in Vodafone advertisement were used to promote the various plans and packages of the company.

Entertainment

Multimedia is heavily used in the entertainment industry to develop special effects in animations and movies. Movies like Ice age, Jurassic park, Avatar will always be remembered for their special effects and animations.

Multimedia games are very popular among children and a variety of these games are available either as CD-ROM's or online. With availability of lot of gaming software programs for individuals or groups, virtual gaming has become a reality today. With usage of such software two players in different countries can play a game sitting on their computers.

Journalism

Multimedia is used a lot in the field of journalism. There are many magazines and newspaper that are published periodically. Today not only we see the text in the newspaper, but can also see the photographs. E-newspaper and E-magazines are also available online where we can see the videos related to particular news.

Stages in Multimedia Production

Any multimedia presentation must be carefully planned and designed. As we studied that multimedia is applicable to almost all the areas, a plan should be according to the field selected. For example,

an application made for entertainment may not work with an educational application and vice versa. Irrespective of the field, the stages in multimedia production remain the same. Let us discuss the stages in multimedia production.

Research and Analysis

During this stage we need to find out about the audience, their skills, needs and qualification. We should also gather as much information as possible about the content to be presented.

Scripting (or Flowcharting)

Scripting or flowcharting means deciding the flow of the multimedia project. This is done by making a flowchart to show the main menu and the branching when a user selects a particular option. For example, if we were designing educational multimedia project on Science for Standard XI, the home page would display all the chapter names. Selecting a particular chapter would display the explanation, exercises, test and other aspects related to the chapter. Thus we can draw a flowchart starting with our main menu and then the subpages that are linked with each other.

Note : The term scripting here does not refer to action scripting or java scripting.

Storyboarding

During this stage the actual visualization of the project takes place. The designer decides how each screen should look like, which media elements are to be used and where to place them on the screen. The storyboard in our earlier example, will contain the design of the home page, where we will place the buttons, how will they look like, what happens when we click this button and other such required elements.

Collection of media elements and construction

After the storyboarding, the designer is ready with the prototype and starts creating the graphics and other media elements to be used in the project. For example, characters to be used are created, sounds and videos are recorded and animations are created.

Programming

After the collection and construction of media elements, they are combined together into a final product using software packages like Macromedia Flash, Synfig, Hypercard to name a few.

Testing

This is the final stage of any multimedia project. In this stage we check if all the media elements are working as per the requirements or not. Also, whether the audience targeted will find the design and the content attractive or not.

If we work as per the stages mentioned here then most of the applications that we may develop will turn out to be good.

Summary

In this chapter we discussed multimedia and its components text, audio, graphics, video and animation. Each of the components plays a vital role in the development of the multimedia project. We introduced each component and discussed it in detail. Multimedia is classified into Interactive, Non-Interactive, Entertainment and Edutainment multimedia. Today the usage of multimedia can be found in almost all the fields like education and training, advertisements, entertainment, journalism to name a few. In the systematic and fruitful development of a multimedia application the stages play an important role.

EXERCISE

1. What is multimedia? List the elements of multimedia.
2. What is typeface and font?
3. What are interactive objects?
4. Explain the classification of multimedia giving suitable example for each.
5. Explain how multimedia is used in various fields.
6. List the stages in multimedia production. Write about each in brief.
7. List various file formats used for the following files :

   (a) Audio (b) Video (c) Image

8. **Choose the most appropriate option from those given below :**

   (1) The term multimedia has been coined from which of the following terms :

   (a) mix, media (b) mid, mediate

   (c) multi, mediator (d) multiple, media

   (2) How many maximum media elements are combined in Multimedia?

   (a) three (b) four

   (c) five (d) eight

   (3) Which of the following terms defines a meaningful speech in any language?

   (a) video (b) sound

   (c) text (d) animation

   (4) Times, Century, Bookman are some examples of which of the following font?

   (a) arial (b) typeface

   (c) serif (d) sans serif

   (5) Which of the following is an audio file compressed with windows media compression?

   (a) .mid (b) .wav

   (c) .mp3 (d) .wma

(6) Which of the following terms represent the distance between the top of the letter to the bottom point?

(a) size (b) style

(c) length (d) width

(7) Which of the following terms represent typefaces that have different styles and sizes?

(a) font (b) text

(c) letter (d) character

(8) Macromedia Flash, Synfig studio and Pencil are examples of which of the following animation types?

(a) 3D (b) 2D

(c) 4D (d) 5D

(9) Which of the following terms represent a small dot on the computer screen?

(a) pixel (b) point

(c) cursor (d) cell

(10) Which of the following terms represent a classification of images?

(a) vector, raster (b) bitmap, raster

(c) picture, clipart (d) graphics, picture

(11) Which of the following terms represent a visual change over time?

(a) text (b) audio

(c) animation (d) graphics

(12) Which of the following terms represent a two dimensional matrix of dots?

(a) screen (b) map

(c) cell (d) array

(13) Which of the following terms is full form of MIDI?

(a) musical instrument data interface (b) musical instrument digital interface

(c) musical image digital interface (d) musical instrument digital image

(14) Which of the following terms represent the most common image format used by digital cameras?

(a) .gif (b) .jpeg

(c) .tif (d) .bmp

(15) Which of the following is used to store the images are stored in form of mathematical equations?

(a) vector (b) raster

(c) bitmap (d) picture

(16) Which of the following devices help us to convert photographs into a digital format?

(a) scanners (b) modem

(c) printer (d) monitor

(17) Which of the following terms represent PNG?

(a) programmable network graphic (b) photo network graphic

(c) portable new graph (d) portable network graphic

(18) Which of the following terms represent a helper programs that can be easily installed and used as a part of the web browser?

(a) plug-in (b) text

(c) video (d) audio

(19) In which of the following terms can Multimedia be classified?

(a) intelligent and non-intelligent (b) interactive and non-Interactive multimedia.

(c) intuitive and non-intuitive (d) informative and non-informative

(20) In which of the following animation types does the visual change occur in the x and y axis?

(a) 2D (b) 3D

(c) 4D (d) 5D

LABORATORY EXERCISES

1. Create a title text using OpenOffice Impress or Writer for the text "Gujarat".
2. Record your voice using sound recorder describing about "Adalaj ni Vav".
3. Record a video describing "What is Multimedia".

2 Animation Tool : Synfig

In previous chapter, we discussed about multimedia and the basic building blocks of multimedia. To create a multimedia presentation using these building blocks we need application programs known as multimedia authoring tools. These authoring tools provide the important framework to organize and edit the multimedia elements like graphics, sound, video and animation.

Generally, the authoring tools are classified into:

- Card or Page based tools
- Icon and Event based tools
- Time based tools

In Card or Page based authoring tools the elements are organized as pages of a book or pile of cards. The authoring tool links these pages into an organized sequence. These types of tool are useful when the different elements can be viewed individually like the pages of a book. Some examples of card based authoring tools are Hypercard and Multimedia Toolbook.

In Icon and Event based authoring tools the elements are organized as objects in a structural framework or process. Here you need to build the flowchart of the events or tasks and then add the elements as per the structure. Some examples of icon and event based authoring tools are Authorware and IconAuthor.

In Time based authoring tools the elements are organized along a timeline. They are useful when you want to give a message that has a beginning and an end. The elements are displayed as per the time or location of the events. Time based authoring tools are the most popular amongst all. Some examples of time based authoring tools are Synfig, Pencil, Flash and Director.

There are many multimedia tools available in the market. Some of these tools are proprietary while others are open source. Some of the open source animation tools easily available on internet are Pencil and Synfig.

Pencil

Pencil is a 2D animation software that lets you create traditional hand-drawn animation using both bitmap and vector graphics. It can be freely downloaded from http://www.pencil-animation.org. It is available on various operating systems like Linux, Windows and MacOS X.

Figure 2.1 shows the user interface of pencil software. The interface looks quite similar to other animation or drawing softwares. In pencil animation software, you can create good animation if you are good at free hand drawing instead of object drawing.

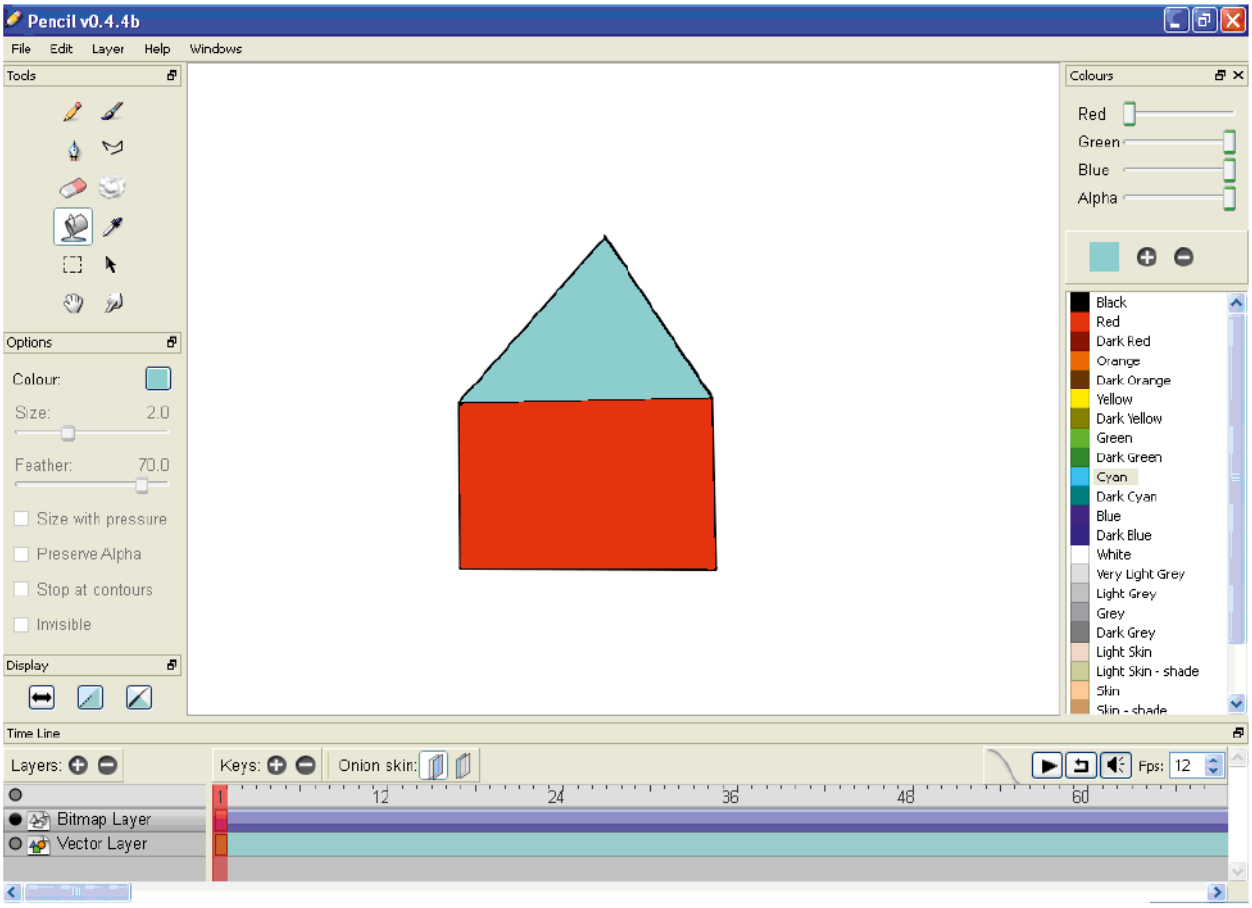


Figure 2.1 : User interface of Pencil animation software

For object drawing and animation, let us learn about an open source animation tool called Synfig Studio.

Synfig Studio

Synfig Studio is a time based multimedia authoring tool that falls in the category of open source. It is available free of cost on the internet and can be downloaded from http://www.synfig.org . Synfig Studio (or simply Synfig) is 2D vector animation software released in 2005 under the GNU GPL (General Public License). It is designed to produce film-quality animation with fewer people and resources. It is available on various operating systems like Linux, Windows and MacOS X. The first version of Synfig was 0.61.05 and new versions have been regularly launched with additional features till date. The latest version of Synfig is 0.63.05. Since it is not available as a standard package in Ubuntu Linux, we need to install it first. Once installed, it will be available under the Graphics option of Applications menu.

Synfig stores animations in XML (extensible markup language) file format, often compressed using compression software gzip which stands for GNU zip. These files use the filename extension .sif in uncompressed form or .sifz in compressed form.

Getting Started

To start Synfig, in the menu bar click Applications → Graphics → Synfig. When you start Synfig Studio you must have observed that it does not start in one window, but opens a number of individual

windows on the desktop. We can drag the window to change the size. If you want to reset the Synfig studio window to default window arrangement as shown in the figure 2.2 then go to Toolbox → File menu → Panels → Reset Windows to Original layout.

Let us first get familiar with the user interface of Synfig. In figure 2.2 you can see the Synfig studio window layout.

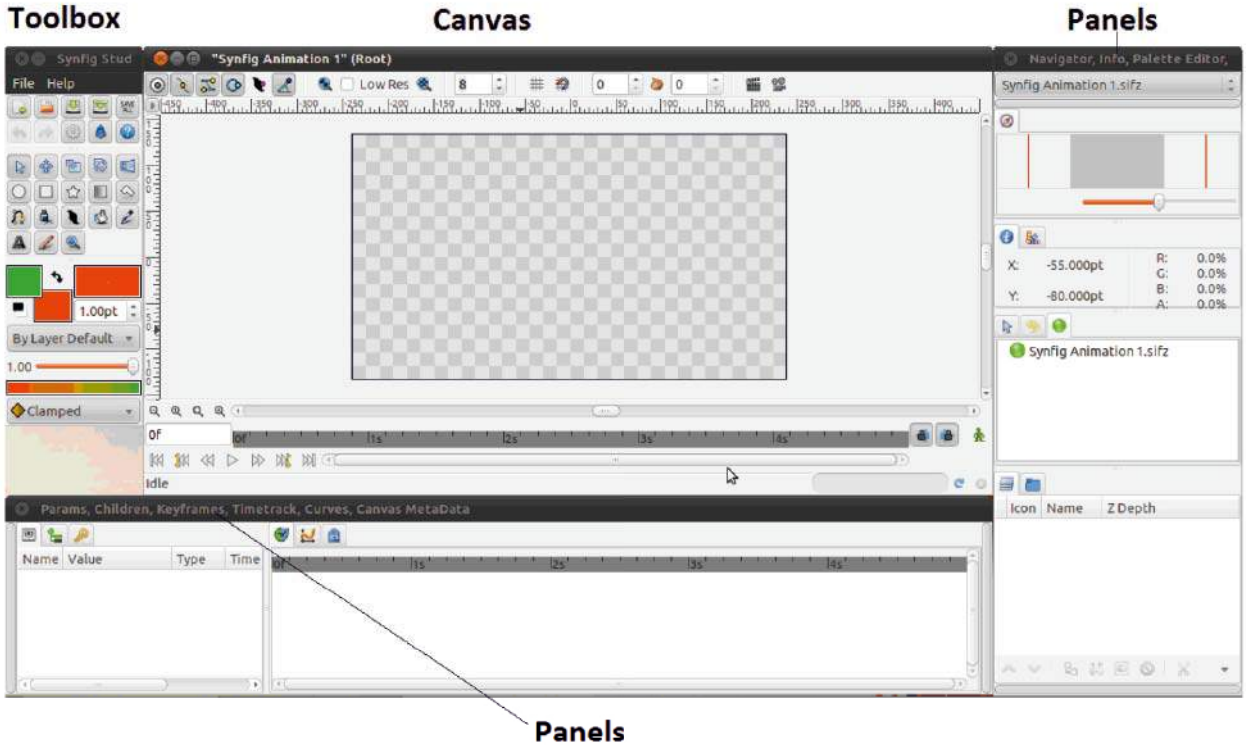


Figure 2.2 : Synfig user interface

As seen in the figure 2.2 the components in the user interface are: toolbox, canvas and panels. Let us discuss the role of each of them.

Toolbox

Toolbox is the main window which contains system menu and tools to create and edit your artwork as shown in figure 2.3. Closing the toolbox exits the application. Even if you have several projects open there will only be one Toolbox for all.

Using the toolbox we can create a new file, open an existing file, save the file and setup properties. The toolbox also contains the general tools like circle, rectangle, transform and others, for creating and editing an object. The toolbox window is separated into three areas or palette:

- The upper palette contains buttons for standard file operations like create a new file, open a file, save, save all files, undo and redo as well as access to the settings dialog and the help system.
- In the middle palette we have the tools like circle, rectangle, transform and others for creating and manipulating an object. We will learn more about each tool later in this chapter.

- The lowest palette contains the default settings for new layers like :
  - **Foreground and background color** : it creates the fill and the outline colors for the object. When we create a layer without outline then it has only fill color. We can change the color by clicking on fill or outline color.
  - **Brush size :** sets the size of the line for new outline layers.
  - **Blend method :** sets the blend method which will be discussed in the later chapters. It is set to "By layer default".
  - **Opacity :** it controls the layers visibility. 0 means the layer is invisible and 1 means the layer is visible.
  - **Gradient :** used by Gradient layers as their fill. It sets to a default gradient between the foreground color and background color.
  - **Interpolation :** each waypoint has an interpolation setting which determines the manner in which the parameter changes.

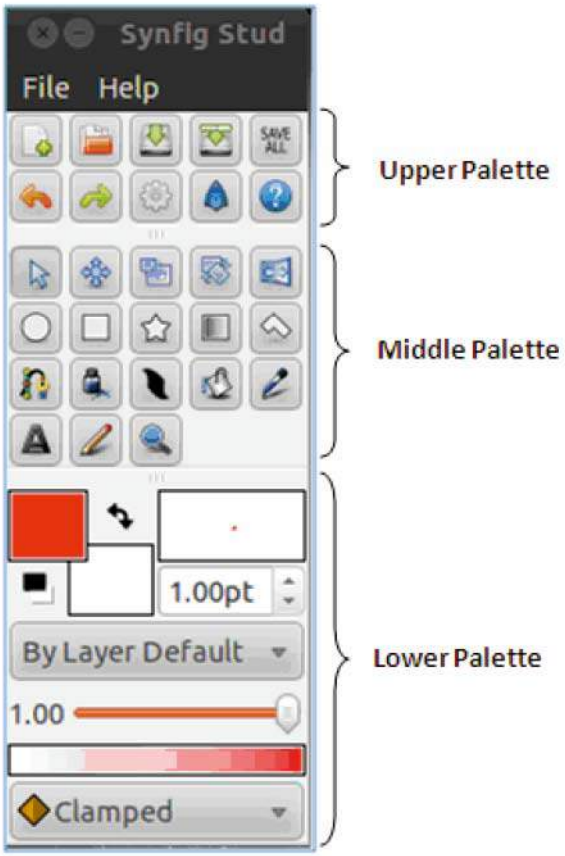


Figure 2.3 : Toolbox

Canvas

When you open a new file or create a new project you will see the canvas window. It can be seen at the center of the user interface as shown in figure 2.4. This is the place where you display your artwork and create animation. Whenever we start Synfig studio a new canvas window appears.

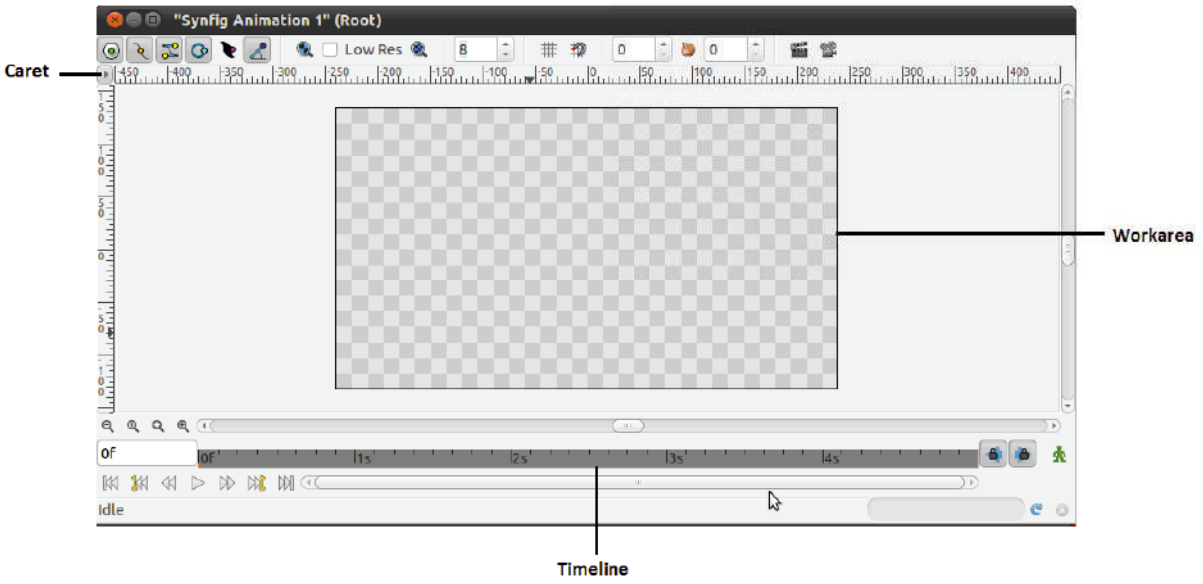


Figure 2.4 : Canvas Window

The area with the grey check-board pattern is the working area in which you can create elements/ layers and manipulate them.

In the upper left corner of the canvas window you can see caret , a little black triangle. Clicking on this button, the canvas window menu will pop up which allows access to most of the Synfig features. Generally, in most of the applications you find the set of menus at the top of the screen or at the top of the drawing window.

You can also see the timeline which appears only when you have non-zero duration in the settings dialog of your project. To the left you can see the number of the current frame and on the right side two buttons to switch the animation status and to lock/unlock the keyframes.

Panels

Synfig has different type of panels like layers, parameter, history etc. Panels contain tools and information about certain elements of your project. Some panels will allow you to modify these elements.

The two windows on the bottom and to the right are customizable dock dialogs. Each dock dialog contains a set of panels which are arranged horizontally or vertically. Some panels share the same space inside the dock dialog and we can switch between them by clicking on their tabs. We can rearrange the contents of dock dialogs by dragging the panel tab to where we want it. If accidentally we close a panel go to the Toolbox, select File → Panels and click on the name of the panel you need. Then drag the panel into the dock dialog where you want to place it. Figure 2.5 and 2.6 shows the panels arranged vertically and horizontally.

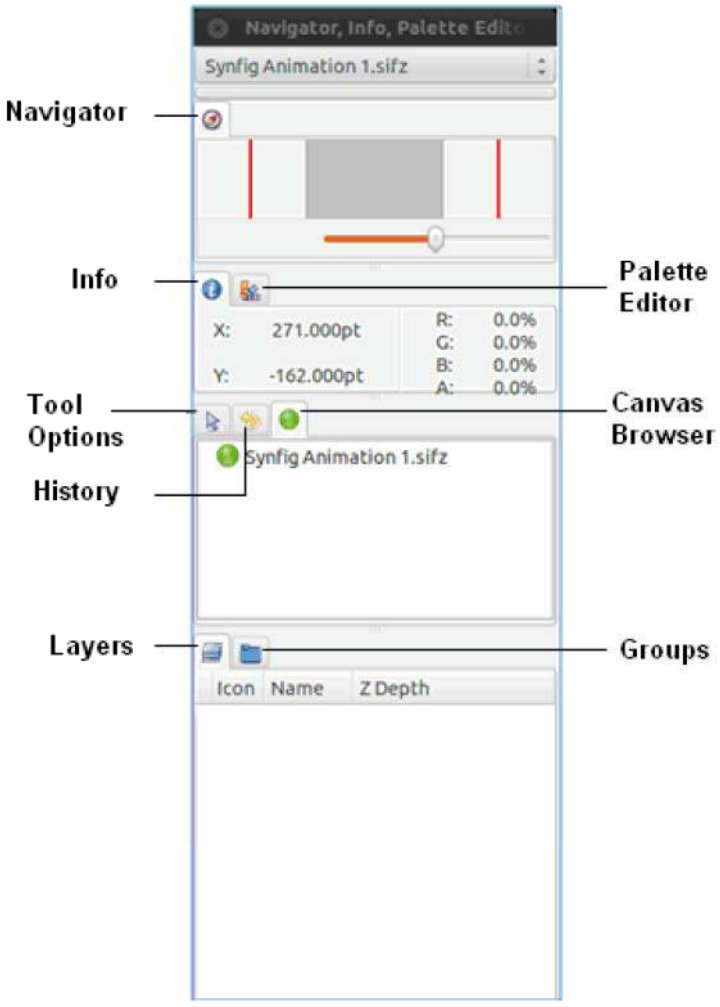


Figure 2.5 : Panels arranged vertically

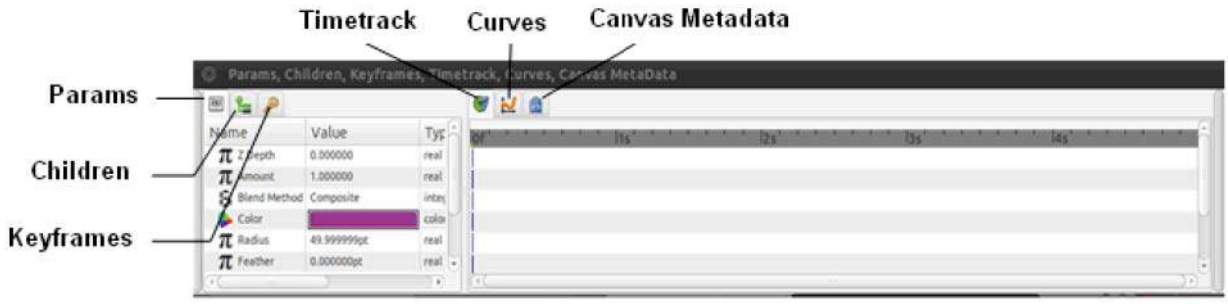


Figure 2.6 : Panels arranged horizontally

Let us discuss the function of some of the important panels:

Layers Panel

It displays the layers of your working canvas. It also allows you to manipulate these layers.

Params Panel

It shows the currently selected layer parameters. When multiple layers are selected, only the parameters that are common in the selected layers are displayed.

Tool Options Panel

It shows options that are specific to the currently selected tool.

Navigator

It shows a thumbnail image of what the currently selected canvas looks like. We can also zoom in and move the focus around.

History Panel

It keeps track of all the actions that are done while editing the file. The check box column is used to disable specific actions without going through the entire action list. So if we want to redo or undo an action or group of actions just click on the check box.

There are many other panels and to know about each simply hold mouse over its icon and a tooltip will pop up describing its function.

Creating new file

Now let us see how to create a new file and set its properties. To create a new file, perform the following steps:

- Select Toolbox → New file. Alternatively, when you start Synfig a new file is automatically created.

Click on the Caret → Edit → Properties. This will open a dialog box as shown in figure 2.7.

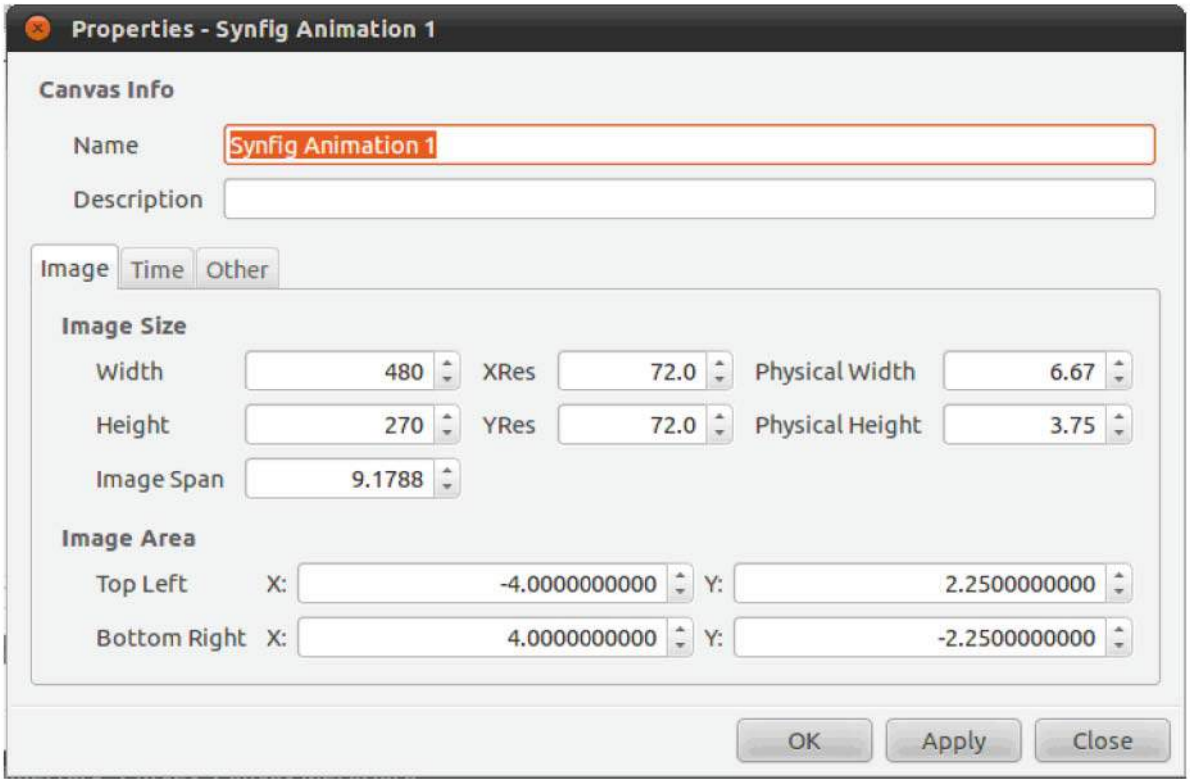


Figure 2.7 : File properties dialog box

You can set the name of the file and give a description to your animation. In the window, you can see three tabs namely *Image, Time* and *Other*. The *Image* tab is used to set the image size and image area as shown in figure 2.7. The *Time* tab is used for setting the start time, end time and frames per second. The default settings are as shown in figure 2.8. We will learn more about these properties later.

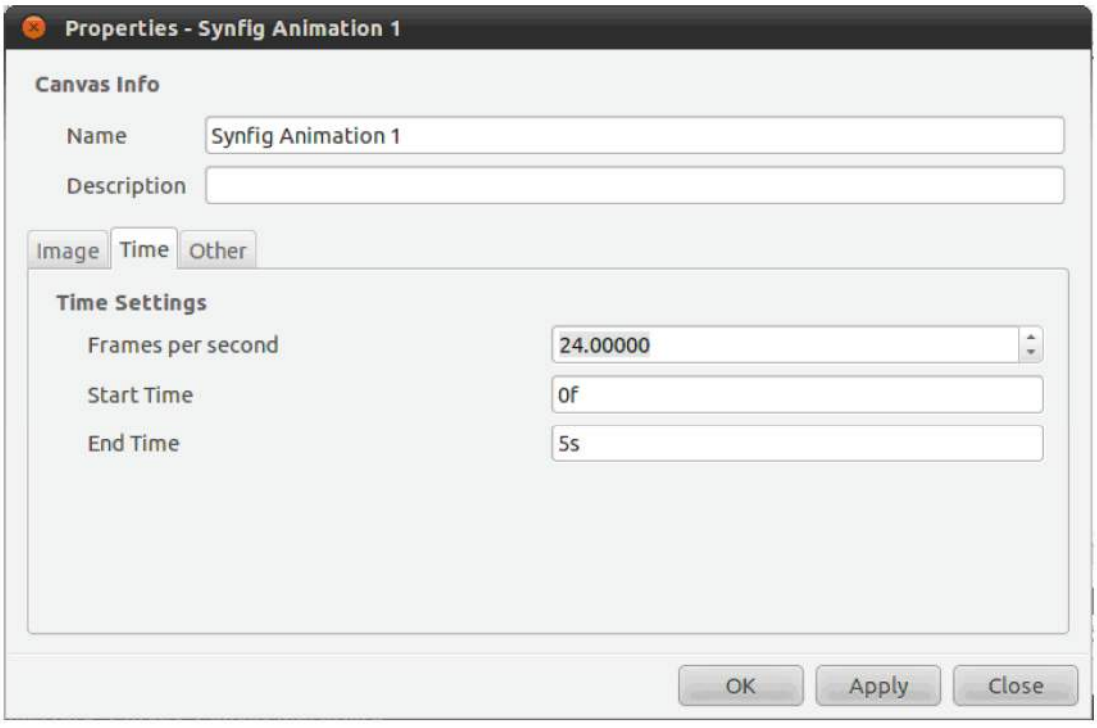


Figure 2.8 : File property dialog box with time tab selected

- Now click OK button and the changes will be applied.
- After creating the file and setting properties we can now start creating our artwork and animation.

Let us now see different tools used to create and manipulate an object.

Tools

Tools help you to create the artwork for your multimedia project. Synfig provides various tools to create and manipulate an object. Figure 2.9 shows these tools, each tool has its own set of options and parameters. Tools are visible in the tool options panel. And parameters are visible in the parameter panel. When you select a particular tool, the tool options panel displays the options that are available for that tool.

Figure 2.9 : Tools to create and manipulate objects

Let us first discuss the tools used to draw an object.

Circle Tool

The Circle tool is used to create new circle layer. Circle layers are used to represent circles. A circle has two important parameters: centre and radius. You can see the parameters in the parameter panel. We can change any of the parameter after drawing the object.

To draw a circle select the circle tool, Click in the working area where you want the center of the circle to be and then drag to set the radius as shown in figure 2.10.

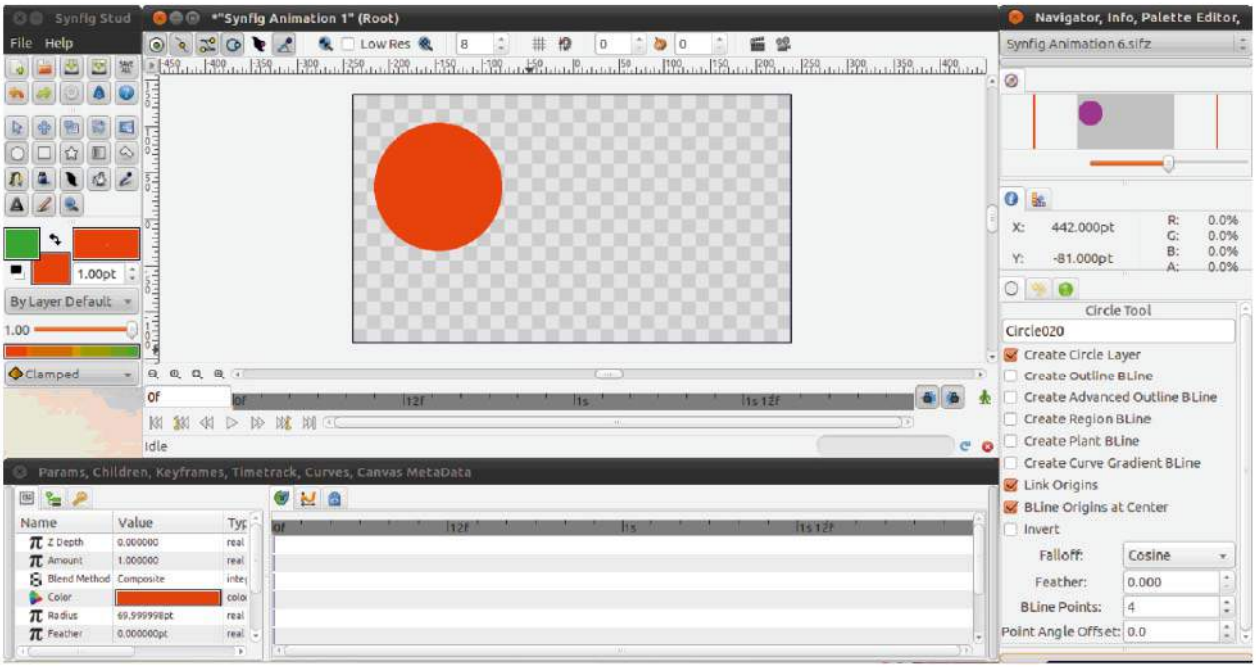


Figure 2.10 : Use of Circle tool

Using the tool options panel you can also change the name of the circle layer, create additional layers using the checkboxes like create outline layer which is used to create an outline around the circle. In the figure 2.10, only the *Create Circle Layer* is selected. If we select the *Create Outline Bline* option then the circle object will be created with an outline.

If you want to edit the circle, say for example you want to increase the radius of the circle or move the circle then click on the duck as shown in the figure 2.11. In case the ducks are not visible then click on the transform tool to see the ducks. You may be wondering what a duck is?

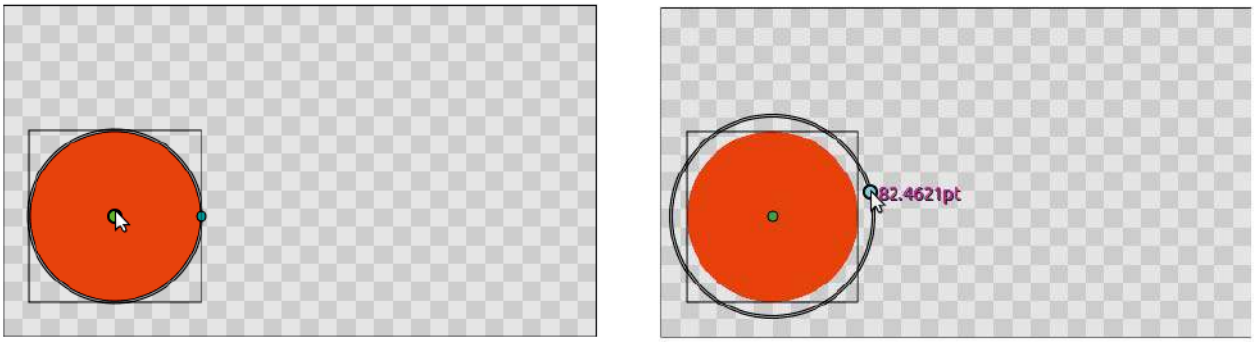


Figure 2.11 : Ducks for circle

A duck is a handle that is used to control some parameter of a layer. For example, a circle will have two ducks: one to control the position of its centre, and second to control its radius. In figure 2.11, you can see the green and blue colored ducks. Ducks come in different color and they have specific meaning. The color of the duck signifies a change in the following aspects of the object:

- Green – position of the object
- Blue – radius of the circle
- Orange – vertices
- Yellow – curves
- Dark blue – to change the angle in star

In figure 2.11 the green duck is used to change the position of the circle and the blue color duck is to change the radius of the circle. Ducks are displayed in the work area for all the currently selected layers. You have to drag the ducks to change the parameters. You can even use the cursor keys for more precise control. If you want to constrain the movement to be either horizontal or vertical then hold down the shift key while dragging.

Sometimes, one type of duck interferes or gets into the way of another duck. You can turn the ducks on/off by selecting the option present on the top of the canvas window as shown in figure 2.12.

Figure 2.12 : Turn on/off various ducks

Rectangle Tool

We can use the rectangle tool in the same way we used the circle tool. It is used to create a rectangle layer. To draw a rectangle, select the rectangle tool, click where you want any corner of the rectangle to be, and then drag to the opposite corner as shown in figure 2.13.

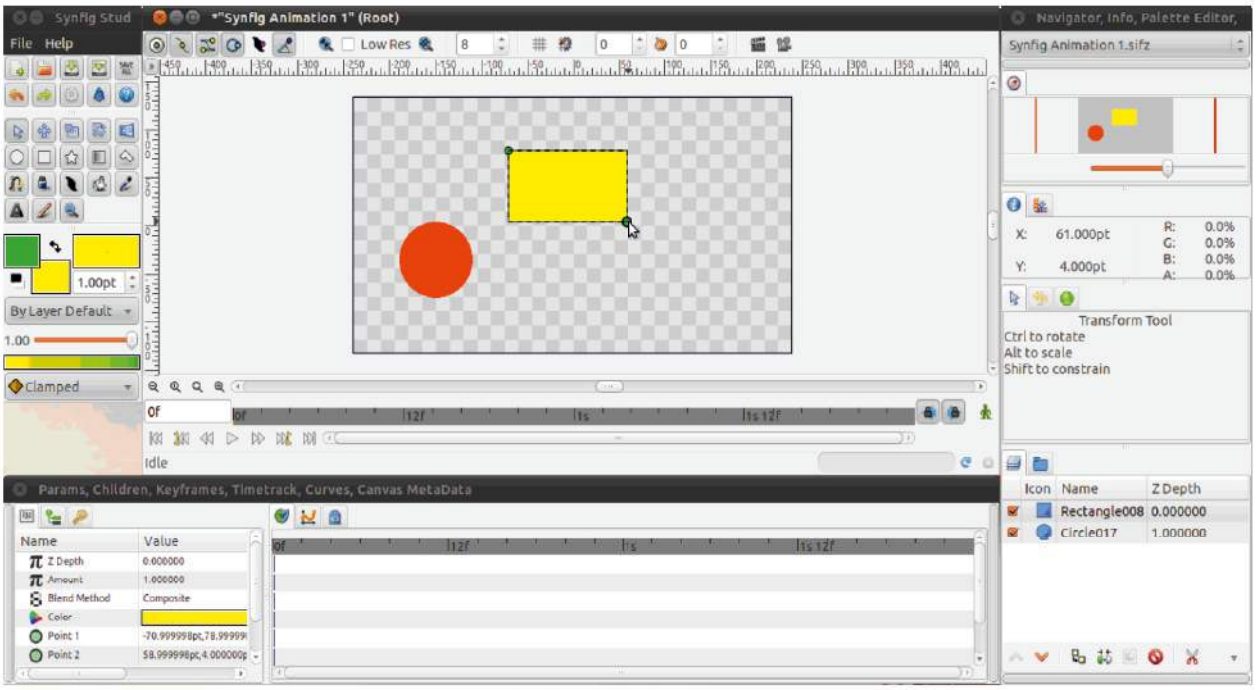


Figure 2.13 : Use of Rectangle tool

You can also see the tool options available in the tool options panel. Also the parameters are displayed in the parameter panel. As we have two objects on the working area, the parameter panel displays the parameters that are common to circle and rectangle. This way we can change a single parameter to make changes in all the objects. If we want to see the parameters of only the rectangle object; then in the layers panel we can select rectangle layer which will now display the parameters of rectangle only.

We can make the changes to the rectangle by selecting the duck and then dragging it. Figure 2.14 shows the ducks for the rectangle tool.

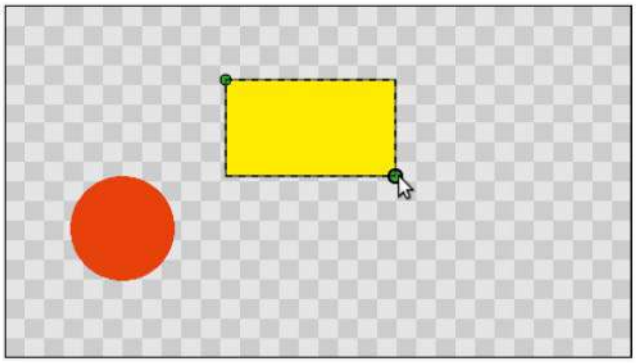


Figure 2.14 : Ducks for Rectangle tool

Star and Polygon Tool

Likewise, we can use the star and polygon tool to create the star layer and polygon layer. The only change that you can see is while editing them. The ducks for each shape is different and change accordingly. Figure 2.15 (a) shows use of star tool and ducks to edit the star. Make the changes to see the difference. Figure 2.15 (b) shows how to draw polygon with polygon tool. A polygon can be created using any number of points and then click on the first point to close it.

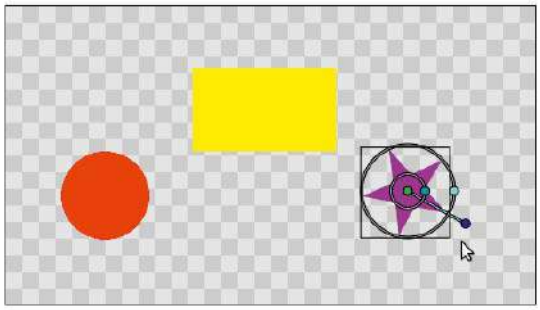


Figure 2.15(a) : Use of Star tool

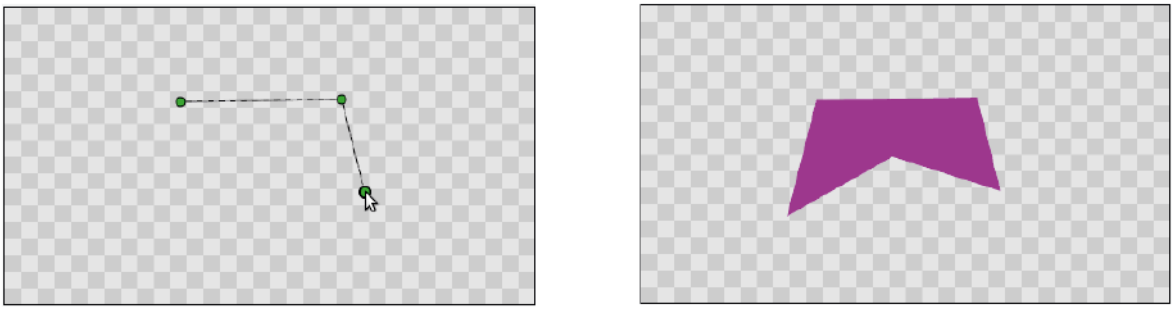


Figure 2.15(b) : Use of Polygon tool

BLine Tool

Bline tool is used to make objects with any number of points and curves. Click on the Bline tool and start inserting the points. In figure 2.16 (a) you can see a shape drawn using Bline tool. To stop drawing the object, right click on the last point. After drawing, the object can be edited using the ducks. Figure 2.16 (b) shows the ducks of the shape drawn using Bline tool.

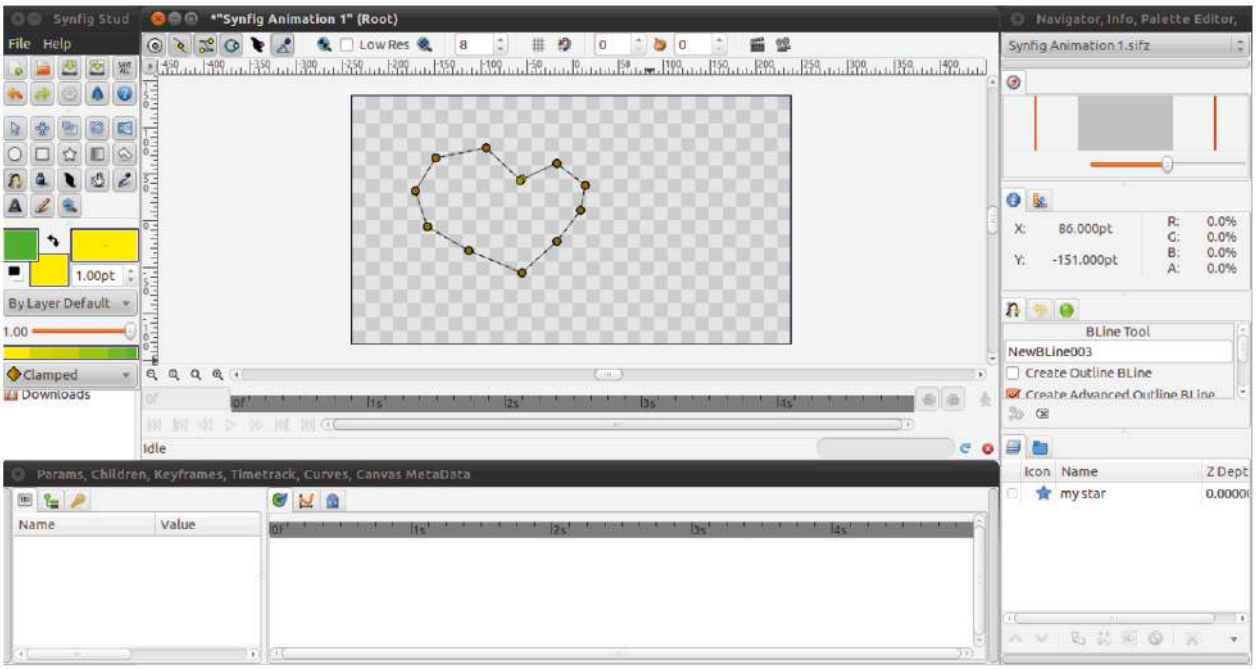


Figure 2.16(a) : BLine tool

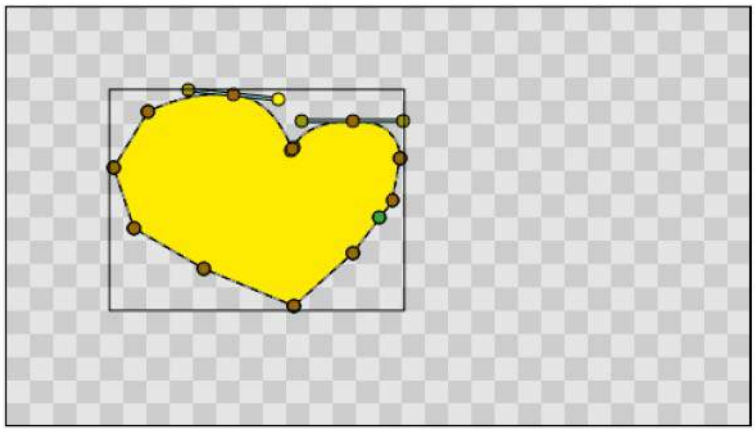


Figure 2.16(b) : Drawn shape with the ducks

After discussing about the object drawing tools, now let us discuss some other tools.

Draw Tool

You must be familiar with the draw tool as it is the most common tool used for drawing. You can draw with the help of mouse. Figure 2.17 shows the use of draw tool.

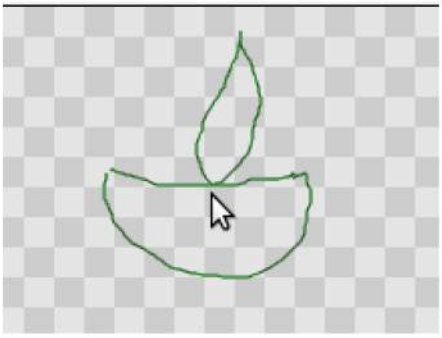


Figure 2.17: Draw tool

Fill Tool

The Fill tool changes the color of objects. To fill a color in the object, select the color from the color palette. Select the Fill tool button and then click on the object. This tool has an effect on the following layers:

- Circle Layer
- Rectangle Layer
- Polygon Layer
- Star Layer
- Region Layer
- Outline Layer
- Checkboard Layer

The other layers are not affected by this tool because they use gradient to fill the visual area.

Eyedrop Tool

The Eyedrop tool allows you to select colors from the Eyedrop tool. Select the Eyedrop tool, and then click in the work area to set the default foreground color in the Eyedrop tool to be the color you are currently pointing at.

Gradient Tool

The Gradient tool is used to create smooth transitions between two or more colors in an object. When you select the Gradient tool, the tool options panel will show the options for the Gradient tool as in figure 2.18.

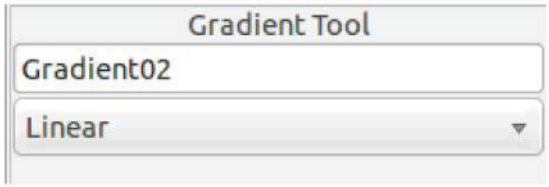


Figure 2.18 : Gradient tool options panel

It allows you to set a name for the layer and choose the type of gradient.

There are 4 types of gradients namely linear, radial, conical and spiral.

Linear - This produces a simple transition in a straight line. Set the foreground and background colors. Click where you want the gradient to begin, and drag to where you want the transition to end. The gradient will be created perpendicular to the line you drag out. You can edit the gradient by moving the endpoint in any direction using the normal tool. Figure 2.19 shows the effect of linear gradient.

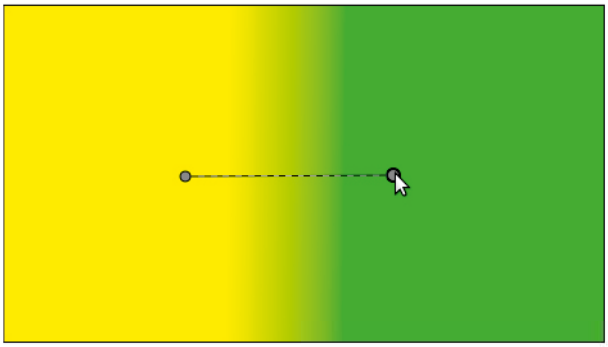


Figure 2.19 : Linear gradient

Radial - This produces circular colors with the transition being at the center of those circles. Click where you want the center of the circles to be, and drag to set the radius of the transition. Figure 2.20 shows the effect of radial gradient.

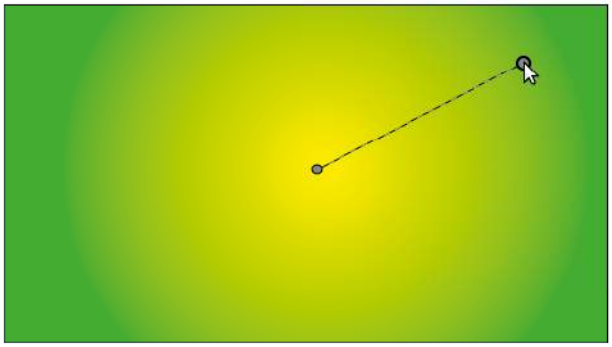


Figure 2.20 : Radial gradient

Conical - This has the appearance of looking down on a tip of a cone. The gradient is along the circular arc of the center and goes in all directions. Click to set the center, and drag to indicate the direction in which the foreground and background colors should go. The center endpoint adjusts the center of the gradient and the other endpoint adjusts the direction of the gradient. Figure 2.21 shows the effect of conical gradient.

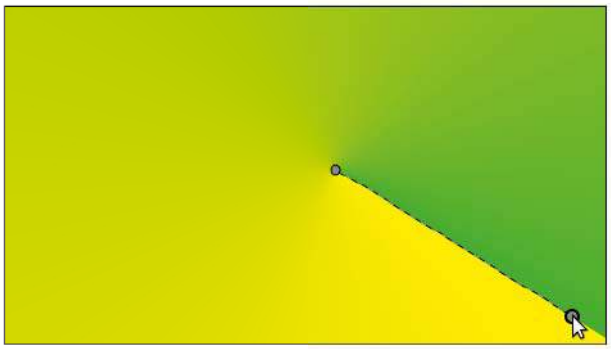


Figure 2.21 : Conical gradient

Spiral - This produces a spiral gradient. Click to set the center of the spiral and drag to set the 'tightness'. Figure 2.22 shows the effect of spiral gradient.

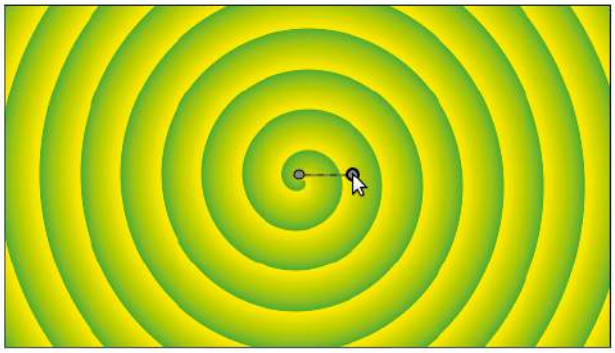


Figure 2.22 : Spiral gradient

Text tool

Text tool allows you to enter text by creating a text layer. Select the text tool icon from the toolbox, and then click in the canvas window where you want the text layer to be created. A dialog box will appear, allowing you to enter the text you want in the created layer as shown in figure 2.23 (a).

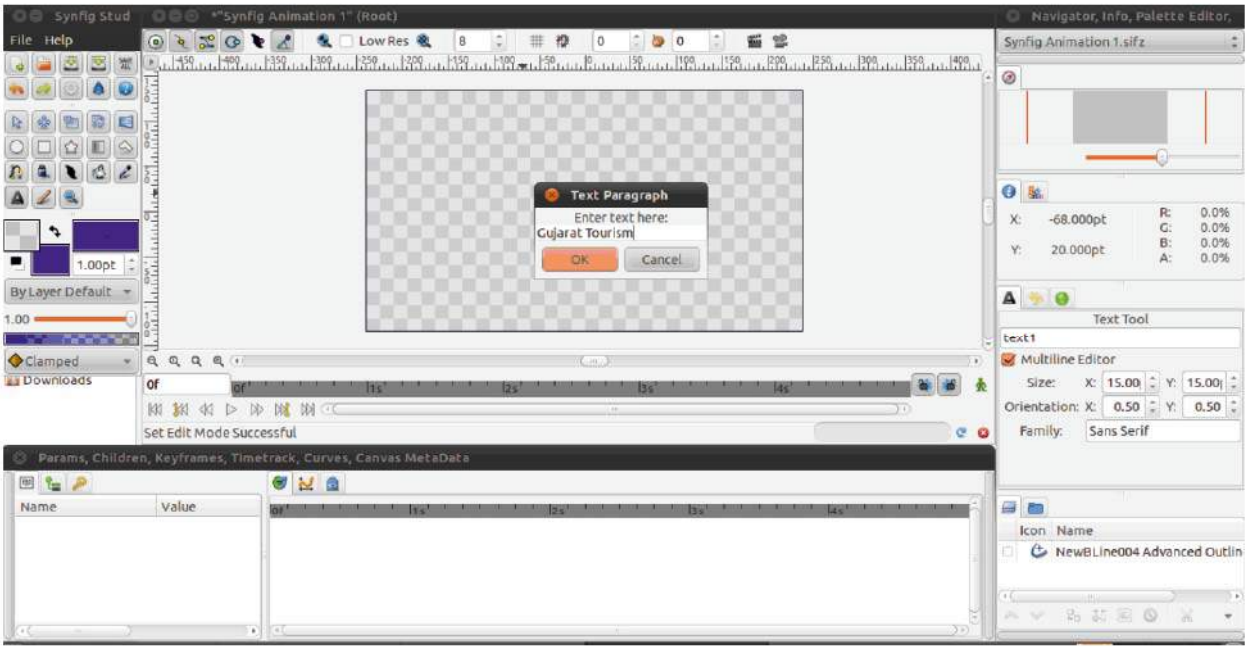


Figure 2.23(a) : Text layer dialog box

Figure 2.23 (b) shows a new text layer created along with its ducks. Using the parameter panel we can change the entered text later on according to our need.

Figure 2.23(b) : Text with ducks

The tool options panel will allow us to:

- Select the name for the new layer; this number will increment as we add new layer.
- Select single or multi line editor for entering the text. If single line is selected then click the enter key to submit the value while in multi-line editor, click the OK button to submit the value.
- Select the horizontal and vertical size of the text.
- Select the 'orientation' of the text. By default (0.5, 0.5) is selected which means that the text will be centered around the point you clicked on.
- Select the font family.

Zoom Tool

The Zoom tool is used to get a closer or far away view of the objects in the working area. The zooming does not affect the output results. It is used to just view the objects from close or far range.

Width Tool

The Width tool helps to increase or decrease the width of a line. It acts like the way you use pencil to make a line thicker or eraser to make it thinner. Select the outline layer to change the width on particular segment of a line, and then move mouse over that line, press the left mouse button and move cursor back and forth along the line. The width of outline will be increased at the places where you moved the cursor. To decrease the width hold "CTRL" key while moving the cursor. This tool is used to fine tune the line width.

Transform Tool

Transform tool allows you to select objects and to move, rotate or scale the ducks. Select the object on the canvas window and click the transform tool. If the object cannot get

selected as it is obscured by another layer then you can select its layer in the layer panel. This will display the ducks of the object. Select the duck and make the change according to your need.

Smooth Move Tool

Smooth move tool works just like transform tool, except when multiple ducks are selected. Select multiple ducks by pressing CTRL+a or dragging a rectangle around the ducks. Now when you drag one of the selected duck, the other selected ducks will also move along with it.

Scale Tool

Just as we discussed smooth move tool, here also when we select multiple ducks, and drag one of the selected ducks the other selected ducks will scale relative to the centre of the selected group.

Rotate Tool

As in smooth move and scale tool, when we select multiple ducks, and drag one of the selected ducks the other selected ducks will move rotating about the centre of the selected group.

Summary

In this chapter we learned about the authoring tools and its types. We have become familiar with the Synfig user interface and also learned to create a new file and set its properties. We discussed the purpose of each panel. The use of each tools along with the tool options were discussed. We can use these tools to create and manipulate the objects and then animate later on. So let us now learn to animate these objects. Having learnt functionalities of various tools available in Synfig, we can now proceed to create a multimedia animation using these tools.

EXERCISE

1. What are authoring tools? List different types of authoring tools.
2. What is the purpose of gradient tool? State the use of four types of gradient learned in this chapter.
3. Briefly explain the three palettes of toolbox window.
4. List the important panels and their function.

8. **Choose the most appropriate option from those given below :**

   (1) The elements are organized as pages of a book or pile of cards in which of the following forms?

   (a) card or page based tools
   (b) icon and event based tools
   (c) time based tools
   (d) animation tools

(2) Which of the following tool helps in changing the color of objects?

(a) draw (b) fill

(c) circle (d) rectangle

(3) Synfig Studio represents which of the following type of authoring tools?

(a) time based authoring tool (b) event based authoring tool

(c) page based authoring tool (d) icon based authoring tool

(4) Which of the following terms represents the number of palettes a toolbox window separated into?

(a) two (b) three

(c) four (d) five

(5) Which of the following is the file extension of a file created in Synfig Studio?

(a) .fiz (b) .sifz

(c) .zif (d) .fis

(6) Which of the following panel shows the currently selected layer parameter?

(a) layer (b) params

(c) timetrack (d) history

(7) Which of the following represents the colour of the duck used to change the radius of the circle?

(a) green (b) yellow

(c) orange (d) blue

(8) Which of the following tools is used to make objects with any number of points and curves?

(a) bline (b) circle

(c) rectangle (d) star

(9) Which of the following tools is used to create smooth transitions between two or more colours in an object?

(a) width (b) fill

(c) eyedrop (d) gradient

(10) Which of the following gradient produces simple transition in a straight line?

(a) conical (b) radial

(c) linear (d) spiral

(11) Authorware and IconAuthor are examples of which of the following types of tools?

(a) card or page based tools (b) icon and event based tools

(c) time based tools (d) animation tools

(12) Which of the following animation software lets you create traditional hand-drawn animation?

(a) pencil (b) synfig

(c) flash (d) director

(13) Which of the following are two important parameters of circle tool?

(a) centre, diameter (b) diameter, point

(c) radius, diameter (d) centre, radius

(14) Which of the following terms represent a handle used to control some parameter of the layer?

(a) point (b) tool

(c) duck (d) panel

(15) The duck used to change the position of the object has which of the following colour?

(a) green (b) blue

(c) red (d) yellow

(16) The layers visibility is controlled by which of the following operations?

(a) opacity (b) gradient

(c) interpolation (d) blend

(17) Which of the following tools allow you to enter text by creating a text layer?

(a) circle (b) text

(c) draw (d) fill

LABORATORY EXERCISES

1. Draw a circle, rectangle and star on the canvas. Select different color for each object. Also insert text as per the object.
2. Using Bline tool draw a candle.
3. Using appropriate tools draw a flower and a candle.

3 Creating Animation Using Synfig

In previous chapter we discussed about the Synfig Interface. We also saw how to use different tools to create and manipulate the basic artwork. Let us now learn how to create an animation. As we have discussed earlier, animation is a visual change over time. In other words, it is the sequence of images displayed one after the other in a given time frame. Let us discuss some basics of animation.

Animation

Animation is a visual change over time wherein the digital images are played one after the other to create a moving effect. The images used in animation can be first created and then displayed in a sequence. This type of animation is known as frame by frame animation. Frames are defined as the length of time in the document. Each frame consists of some objects to be displayed over a period of time. Hence in frame by frame animation, each small change in the image is created and then displayed in sequence. This method is slow and tedious. It also requires lot of time and resource.

The other simple way to create animation is through keyframe. A keyframe is a frame in which we define a change to an object's properties. Using keyframes you need to specify the first position and the last position of the object. The in-between frames are automatically understood and need not be specified. For example, say we want to show a circle moving from left to right. Here you need to create the first keyframe on the left and include the last keyframe on the right. The series of frames in between are drawn automatically. This effect is known as tweening. Tweening is an action which calculates the number of frames between keyframes and path of the action. This type of animation is known as cel animation and the computers generally employ this type animation. The real difference between varieties of animation software is how much to be drawn by the user and how much is automatically generated by the computer.

Another type of computer animation is known as kinematics. It is the study of movement and motion of objects that have joints such as walking man or running leopard. In some animations we can use an effect known as morphing, wherein the image transforms into another. Say for example you must have seen that a person's face transforms into another face and so on. Many products are available in the market that supports this type of effect.

Creating Animation

Now let us create our first animation of a circle moving from left to right. Follow the steps given to create this animation:

- Start Synfig (or Select Application → Graphics → Synfig), this will automatically open a new file for you.
- Click the Caret → Edit → Properties. This opens the canvas properties dialog box as shown in the figure 3.1.

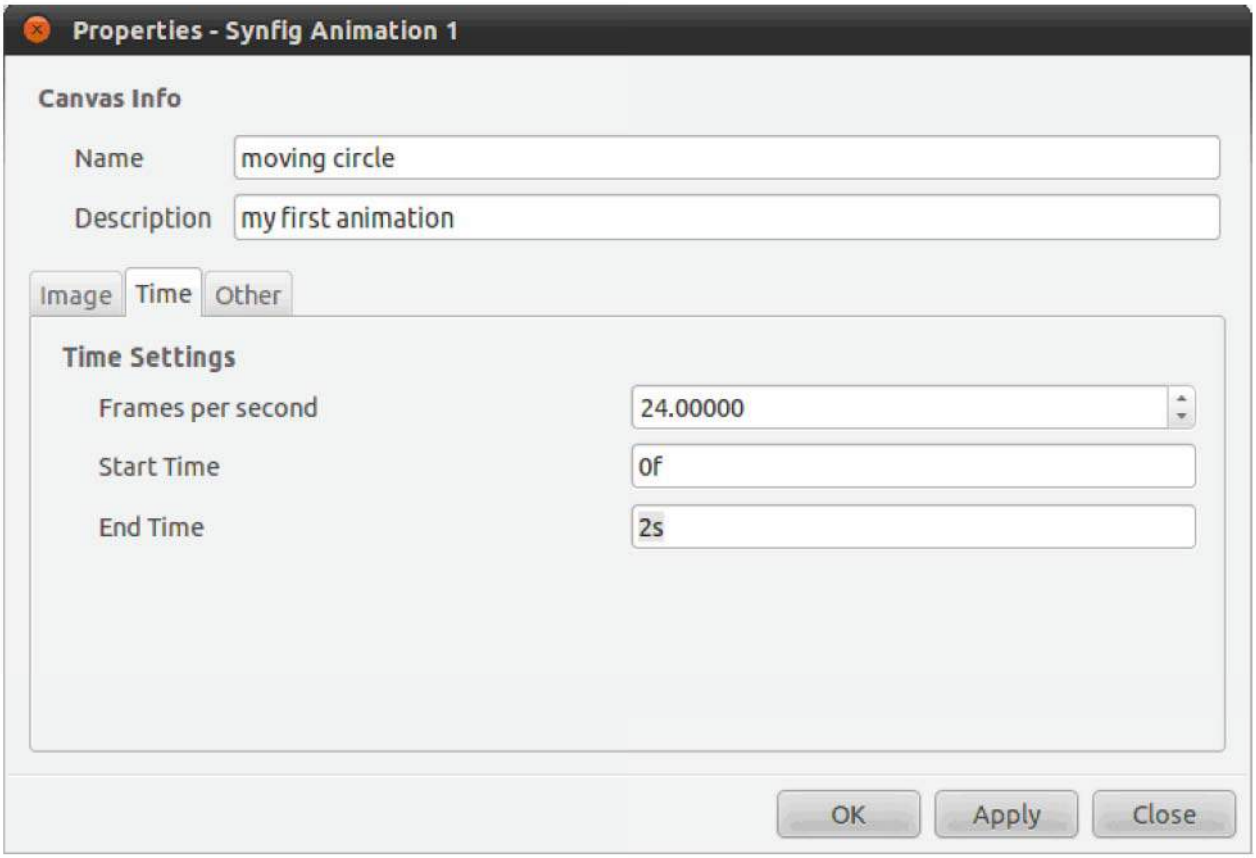


Figure 3.1 : Canvas property dialog box

- Give a name and description to your first animation. As shown in figure 3.1 we have given the name to our animation as "moving circle" and description "my first animation". Now click on Time tab and change the End time to 2s. It means that our animation will be of 2 seconds duration. By default the End time is set as 5s.
- Click Ok. This will close the canvas properties dialog box and the changes we made can be seen as the file name has changed and the timeline is now 2s long.
- Now we need to draw a circle. Select the color of the circle by clicking on foreground color or using the palette. We can also change the color later by using the parameter panel. Currently we choose the fill color as yellow. Then select the circle tool and create a circle as shown in figure 3.2. You can use the transform tool to make changes in the circle drawn.

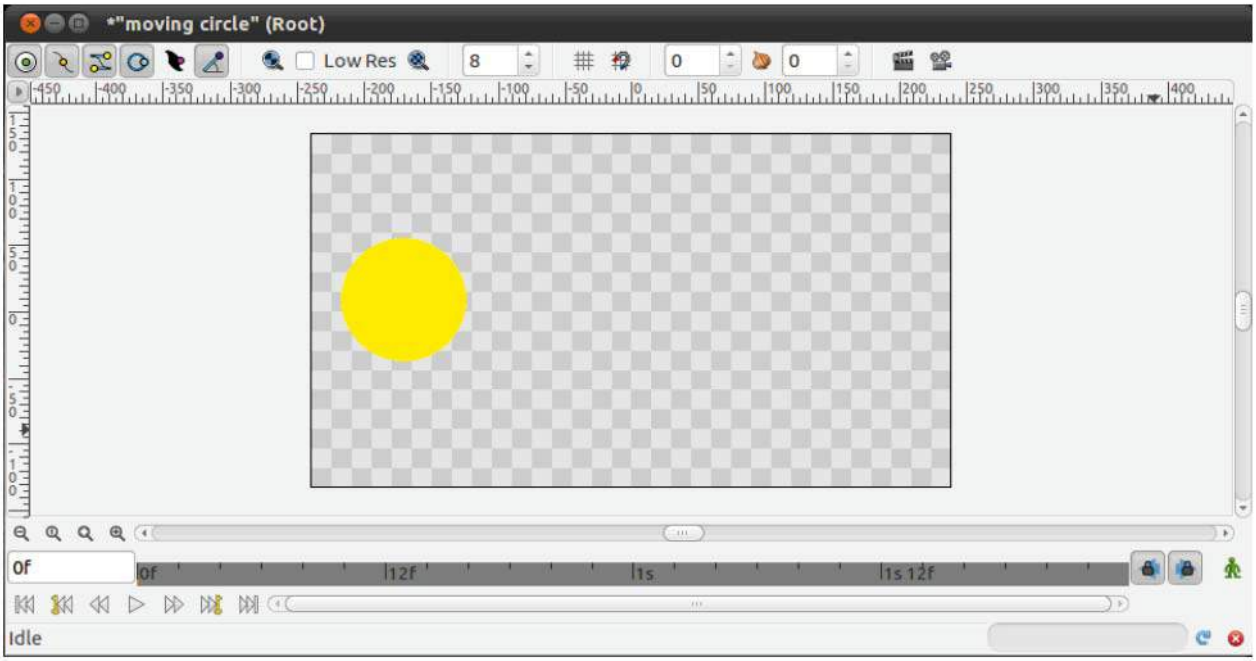


Figure 3.2 : Use of circle tool

Now we are ready with the object that is to be animated. Before starting with animation let us understand two important things: Timebar and Animate editing mode.

Timebar

As shown in figure 3.3 click on the timebar, you can see the orange indicator showing the position in time. Clicking at various points, you can see values "0f, 1s, 1s 10f" etc. in the entry field. You can also type the position in the entry field and press Enter, the indicator will take you to that position in timeslider. These values indicate a specific point on the timeline in terms of seconds (s) and frames (f). By default each second is divided into 24 frames. As you can see in figure 3.1 frames per second (fps) is set as 24. The frame starts at 0 and goes up to 24. For example, when one second and thirteen frames have passed, the entry box displays "1s 13f".

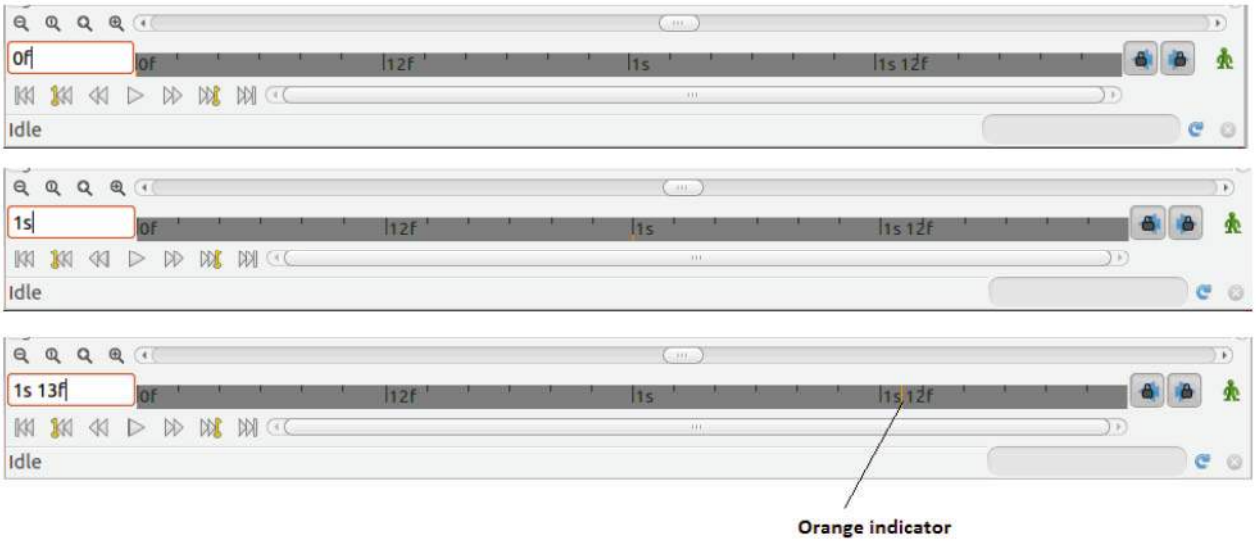


Figure 3.3 : Timebar with the indicator at 0f, 1s and 1s 13f

Animate editing mode

On the right side of the timebar shown in the figure 3.3, you can see a green button ( ) . It will be visible if you have a non-zero end time. Clicking on this button, turns it into red button ( ) and the canvas displays a red outline as in figure 3.4. This means that now the animate editing mode is active and any changes made will have an effect on the animation in the timeslider.

In animate mode, whenever a parameter of an object is changed, a waypoint is created to remember the change in the position and the time on the timeslider at which the change happened.

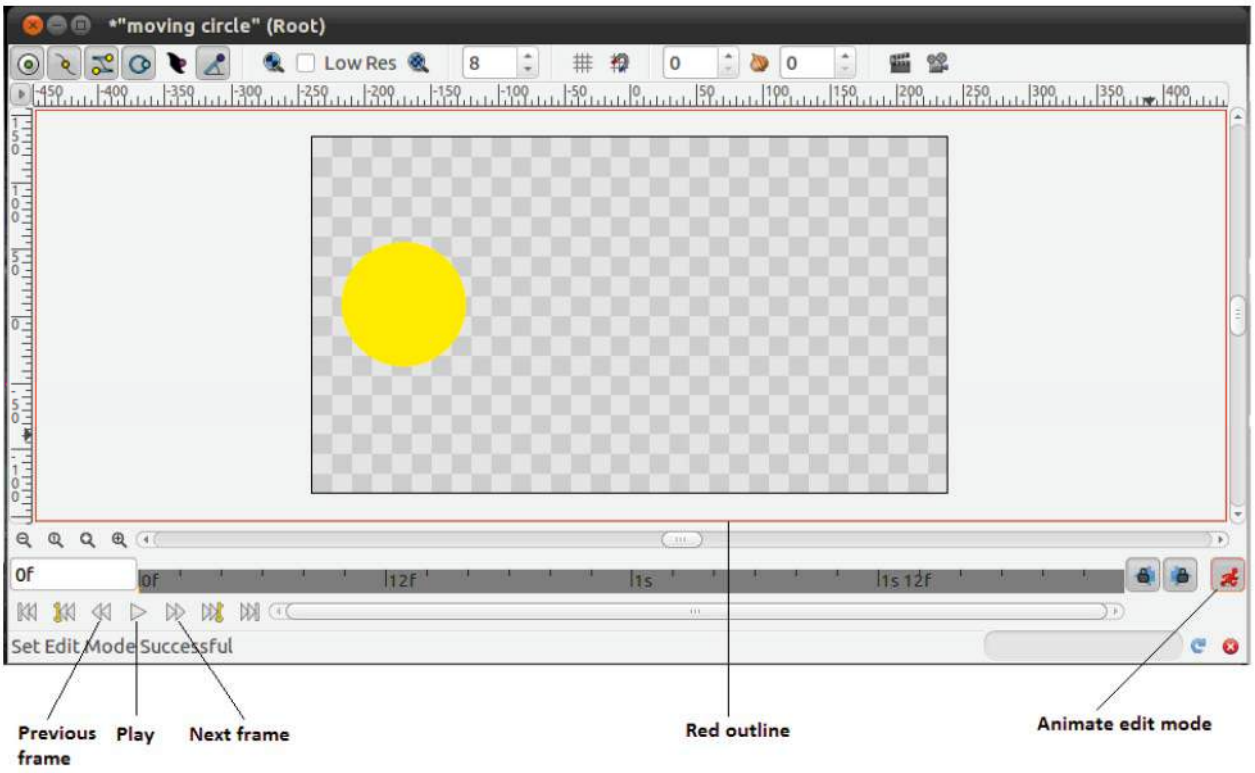


Figure 3.4 : Animate editing mode

When not in animate editing mode, changes to a parameter are applied throughout the entire timeline of the animation.

- Let us now start creating our animation. Click on Animate editing mode.
- You can see "0f" in timeline entry field. Move the circle slightly so that it records the location for the first frame. You can also move the circle to a new location on the working area as the starting point.
- Now click at the end of the timeline. This will show the '2s' mark in the entry field. You can even type '2s' in the entry field to reach the end of timeline. Grab the circle

and hold the shift key to move it in a straight line to the right of the working area as shown in figure 3.5.

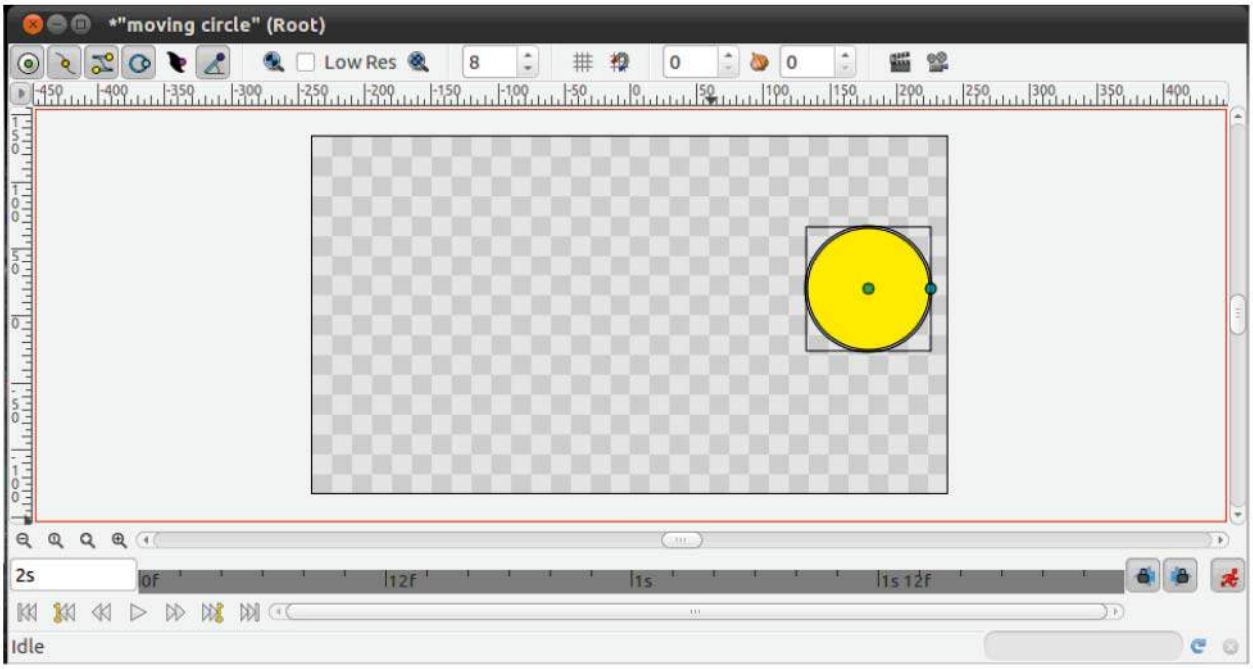


Figure 3.5 : Moving the circle to the right

- Click on any position on the timeslider and you will see the circle on a new position. We had specified only the beginning and the end locations of our circle, but Synfig has automatically created the in-between images or frames. As we discussed earlier, this process is known as tweening. These frames when displayed together will give an effect of moving circle.
- Click on the play key as shown in figure 3.4 to see how our animation will look. You can also click on the previous frame and next frames buttons to see frame by frame animation.
- Finally we need to render our work. Rendering means the computer uses various algorithms to apply the final effects you have specified on the objects created.
- Before rendering, click on the  button to leave the animate editing mode.
- Click on Caret → File → Save. Save the file by the name "moving-circle.sifz".
- Click on Caret à File à Render. Change the filename to "moving-circle.gif" and the same location where you saved "moving-circle.sifz". Select target format as "gif" instead of "Auto", then click Render as shown in figure 3.6 (a). It will take few seconds for the file to render. You can see the message "file rendered successfully" on window status bar located on the bottom of the window.

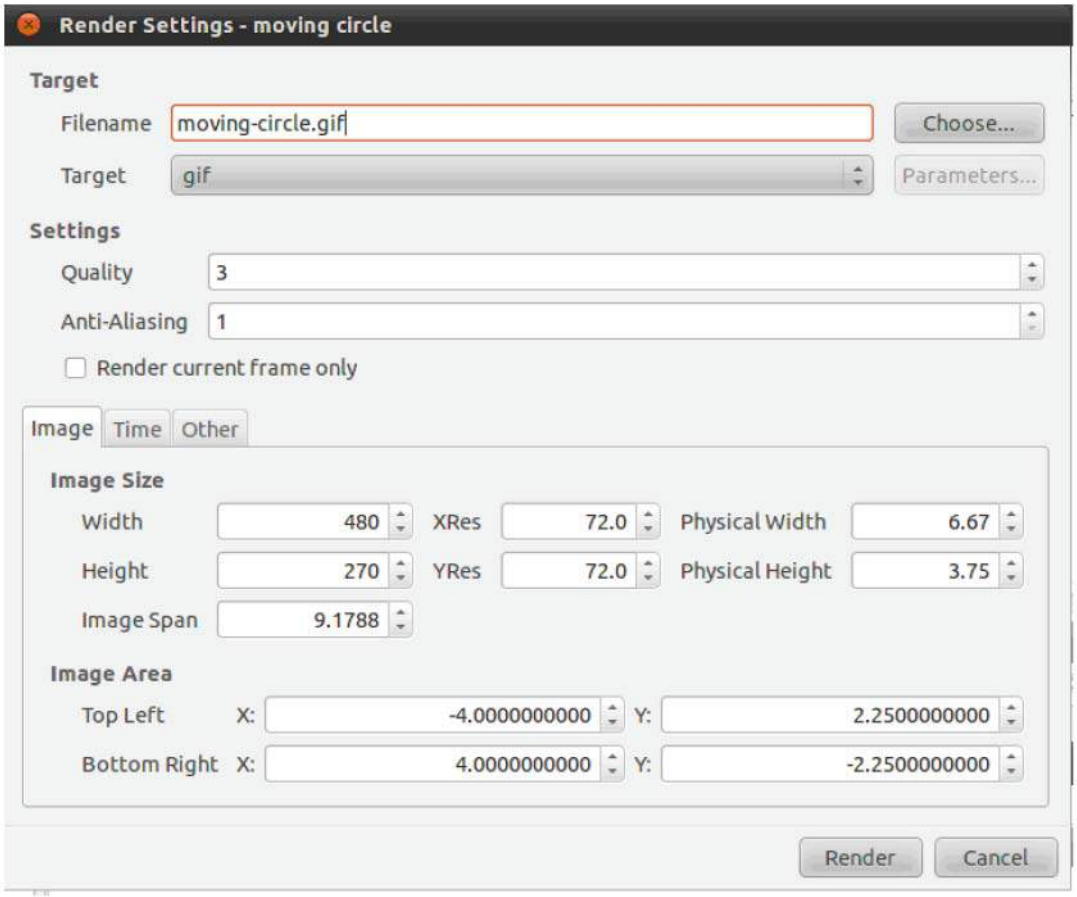


Figure 3.6(a) : Render the file

- Open moving-circle.gif file in a web browser or image viewer to see your animation. Figure 3.6 (b) shows the output as visible in the web browser.

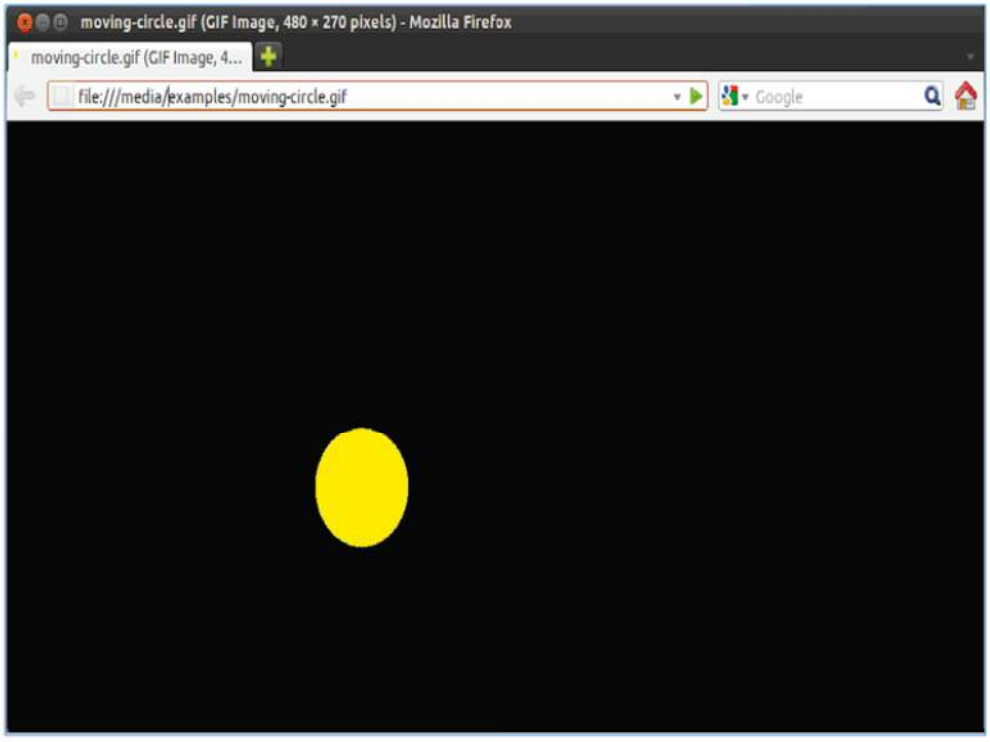


Figure 3.6(b) : Output displayed in web browser

Good. So, you have created your first animation.

Animation using keyframe

Let us create an animation of moving the circle from left to right and then left again using keyframes. For this we need to create following three steps:

- First the circle is on the left.
- Second, the circle is on the right.
- And third, the circle is back on the left.

Follow the steps given to create this animation:

- Start Synfig and open a new file.
- Click the Caret → Edit → Properties. Give a name and description to your animation. As shown in figure 3.7 we have given the name to our animation as "moving circle2" and description "moving circle using keyframes". Now click on Time tab and change the End time to 2s. Click OK to make the changes.

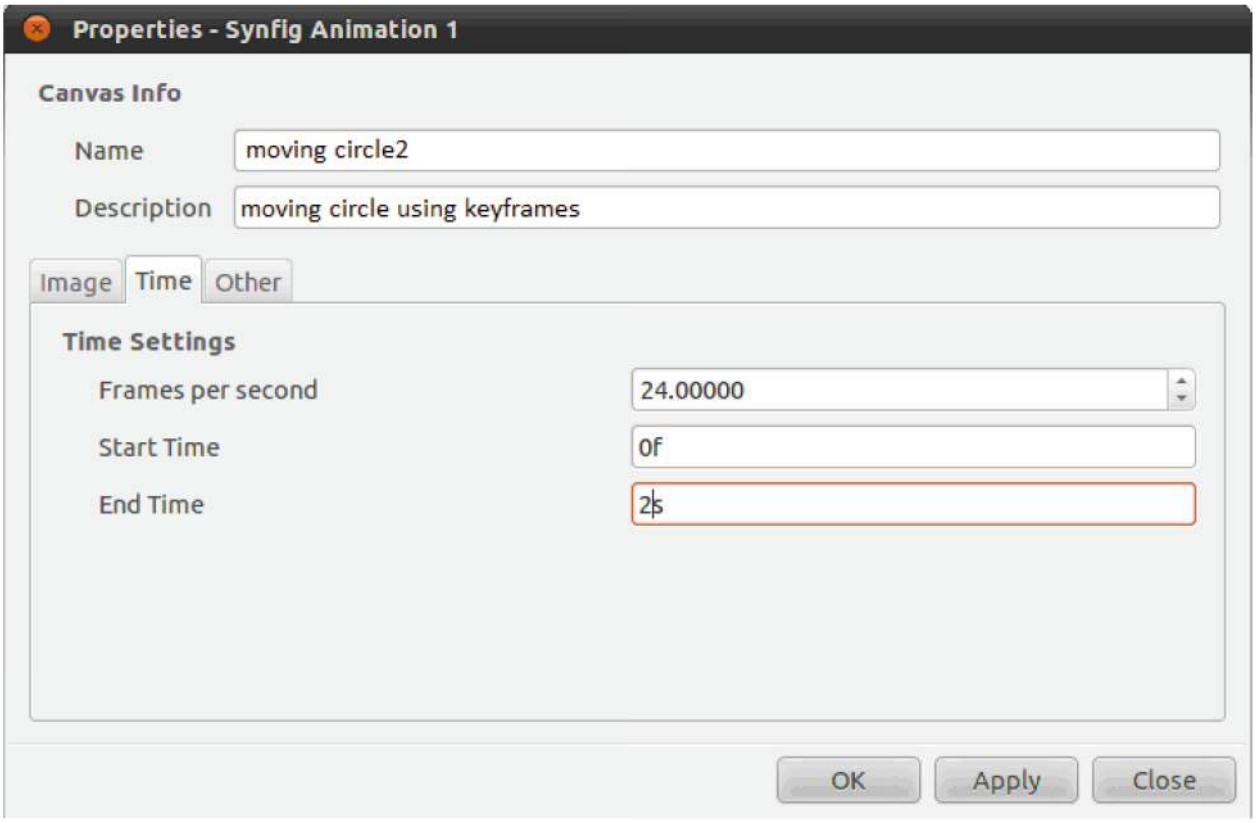


Figure 3.7 : Canvas property dialog box

- Now let us draw the background for our animation. Select the rectangle tool and create a simple black rectangle as shown in the figure 3.8.

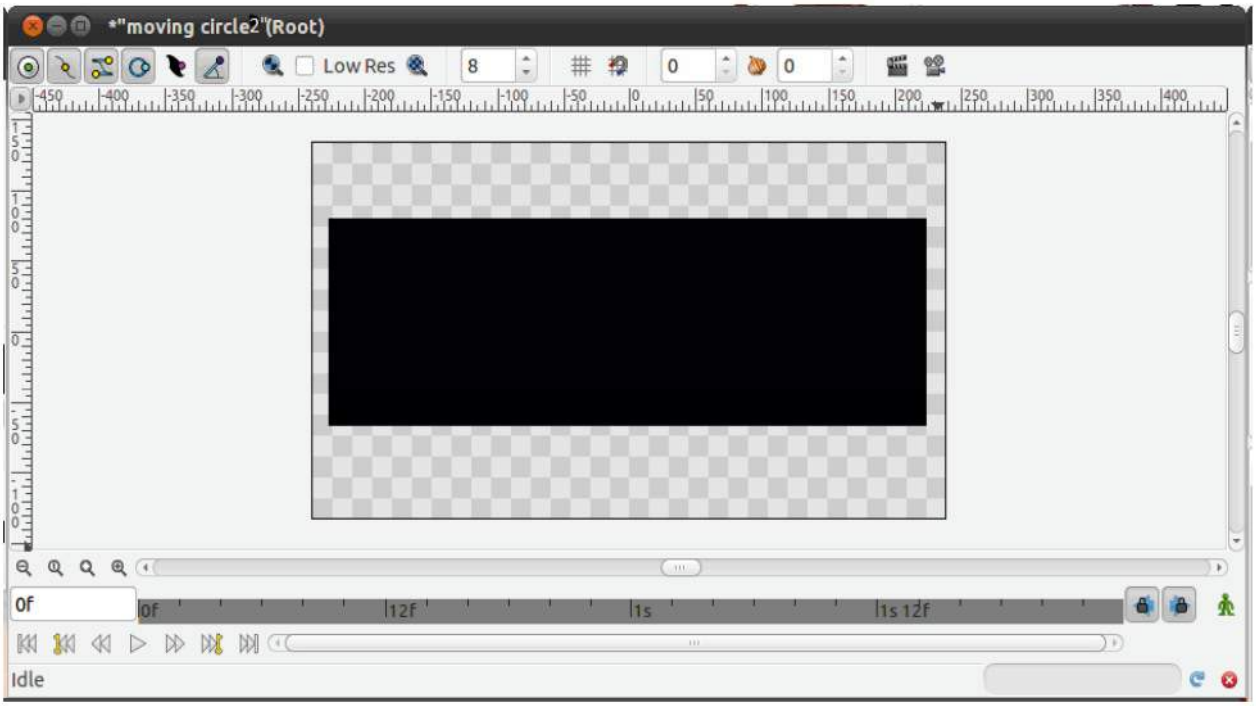


Figure 3.8 : Rectangle drawn using rectangle tool

- Now we need to draw a circle. First change the fill color to green. Then select the circle tool and create a circle as shown in figure 3.9. You can use the transform tool to make changes in the circle drawn.

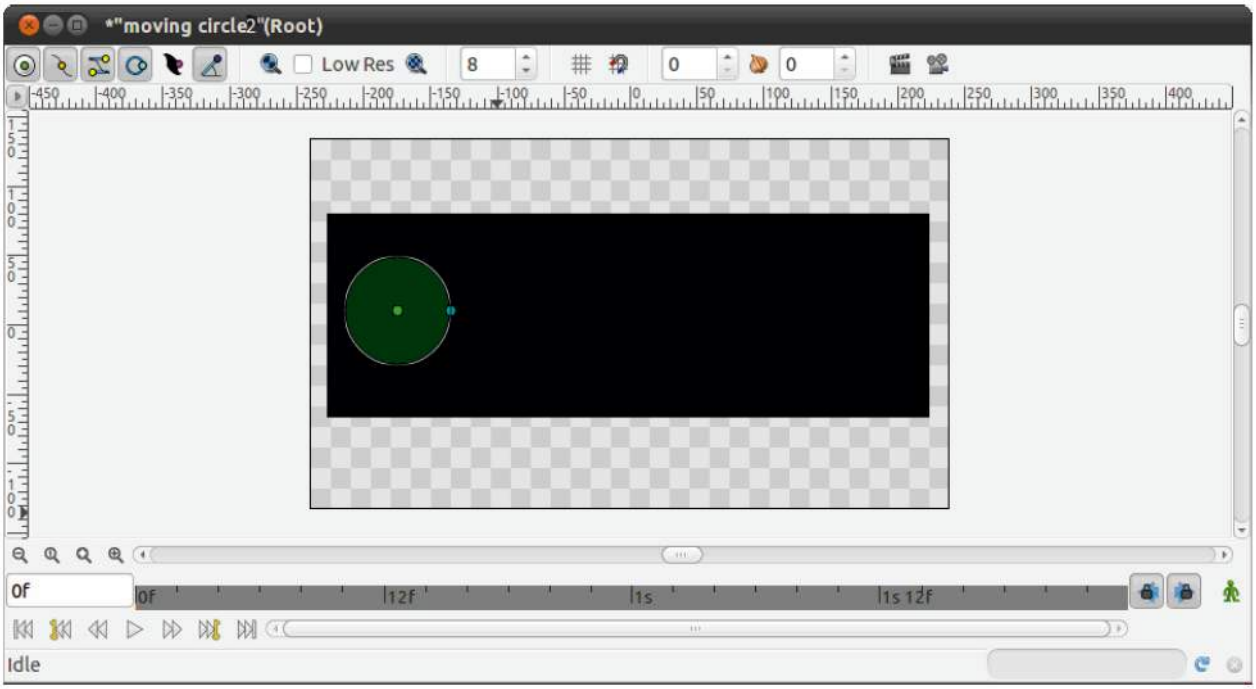


Figure 3.9 : Circle drawn using circle tool

- As the object is ready let us now start creating our animation by clicking on Animate editing mode.

- Go to parameter panel and click on  to open keyframe panel to add keyframes. Earlier we discussed the three steps; these will now become our keyframes. Click the small button  to add a new keyframe as shown in figure 3.10. This will make a new entry in the list displaying "0f, 0f, (JMP)".

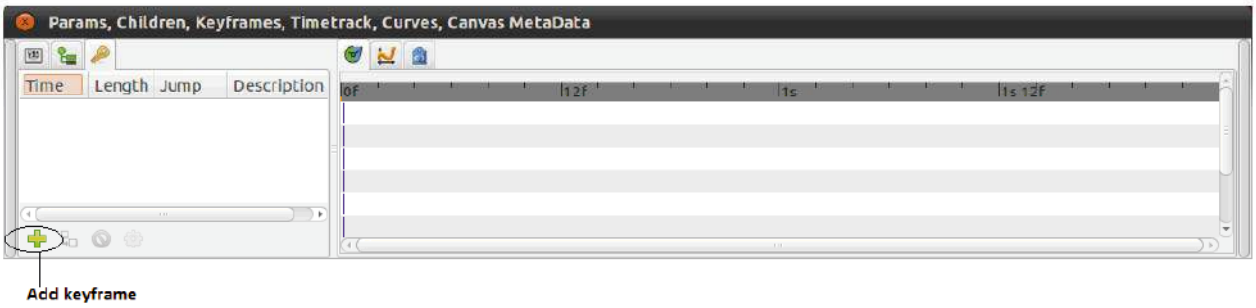


Figure 3.10 : Keyframe panel

- Go to the "1s" mark in the time slider. Click on  to add another keyframe. Again go to "2s" in the timeslider and add another keyframe. So now you have three keyframes in the list as shown in figure 3.11.

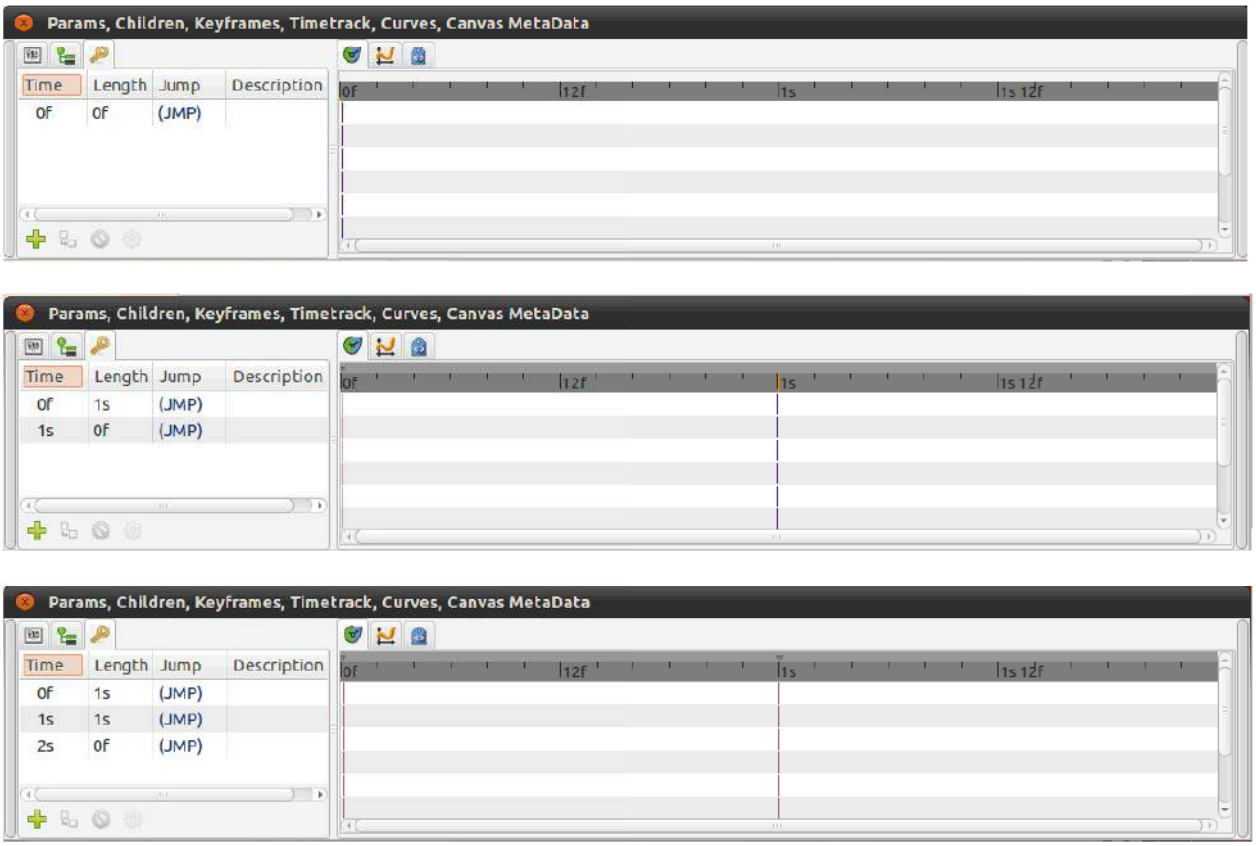


Figure 3.11 : Three keyframes at 0f, 1s and 2s

Let us understand the keyframe panel. It displays "Time" which indicates the start time, "Length" which indicates duration, "Jump" which are links used to jump to correct time and "Description".

- Here, in all the three keyframes the circle is on the left. But as discussed in our steps the second position of the circle is on the right. So now let us edit the circle at second keyframe i.e at 1s mark. Click on the "(JMP)" of 1s mark and move the circle from left to right as shown in figure 3.12 (a). So finally, we have our circle at 0f on the left, then at 1s we have it on right and on 2s we have it on left again. On the Timetrack panel you can see three orange diamonds known as Waypoints as shown in figure 3.12 (b). They represent the time at which the parameter (like color or location) of the object changes.

- Click on any position on the timeslider and you will see the circle on a new position. We specified only three locations of our circle, but Synfig has automatically created the in-between images or frames. These frames when displayed together will give an effect of moving circle.

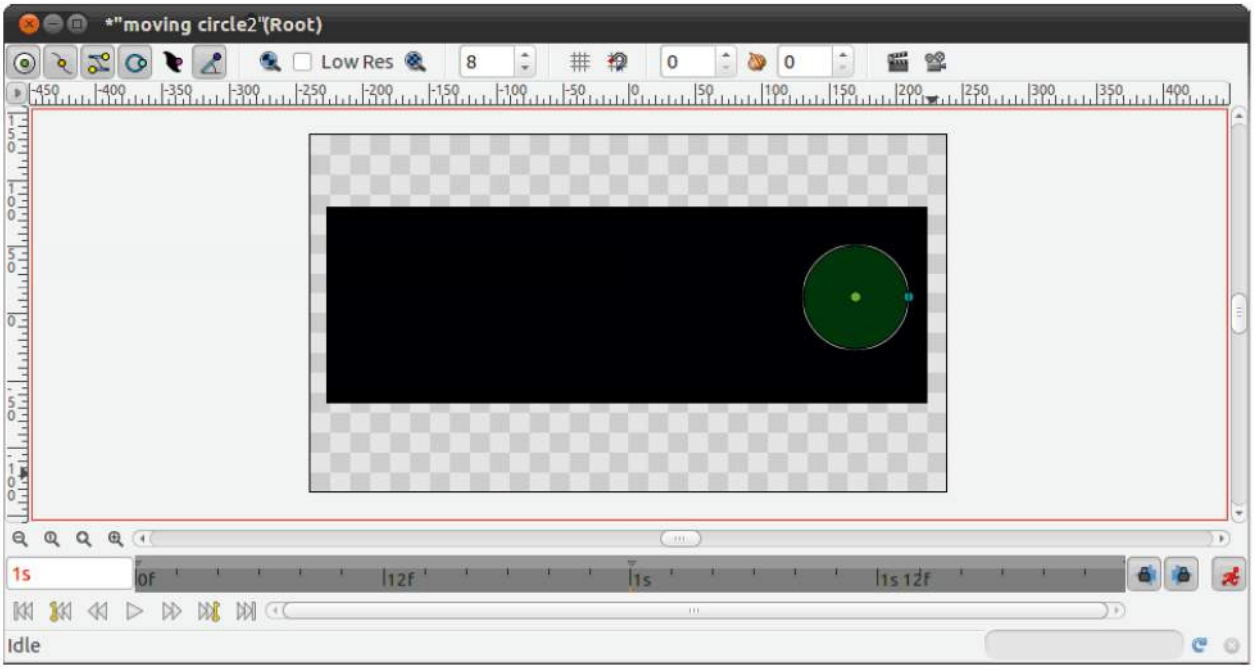


Figure 3.12(a) : Moving circle at 1s to its right

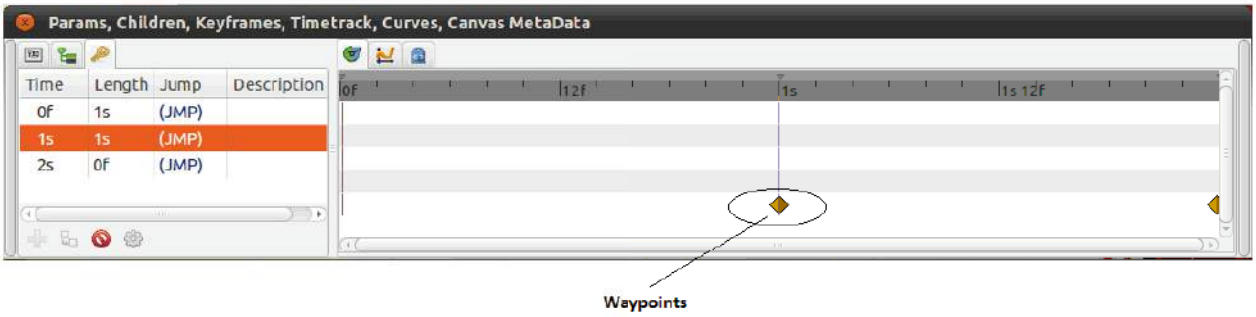


Figure 3.12(b) : Waypoints

- Now to see the animation we need to render our work.
- Before rendering, click on the  button to leave the animate editing mode.
- Click on Caret → File → Save. Save the file by the name "moving-circle2.sifz".
- Click on Caret → File → Render. Change the filename to "moving-circle2.gif" and the same location where you saved "moving-circle2.sifz". Select target format as "gif" instead of "Auto", then click Render as shown in figure 3.13 (a). You can see the message "file rendered successfully" on window status bar located on the bottom of the window.

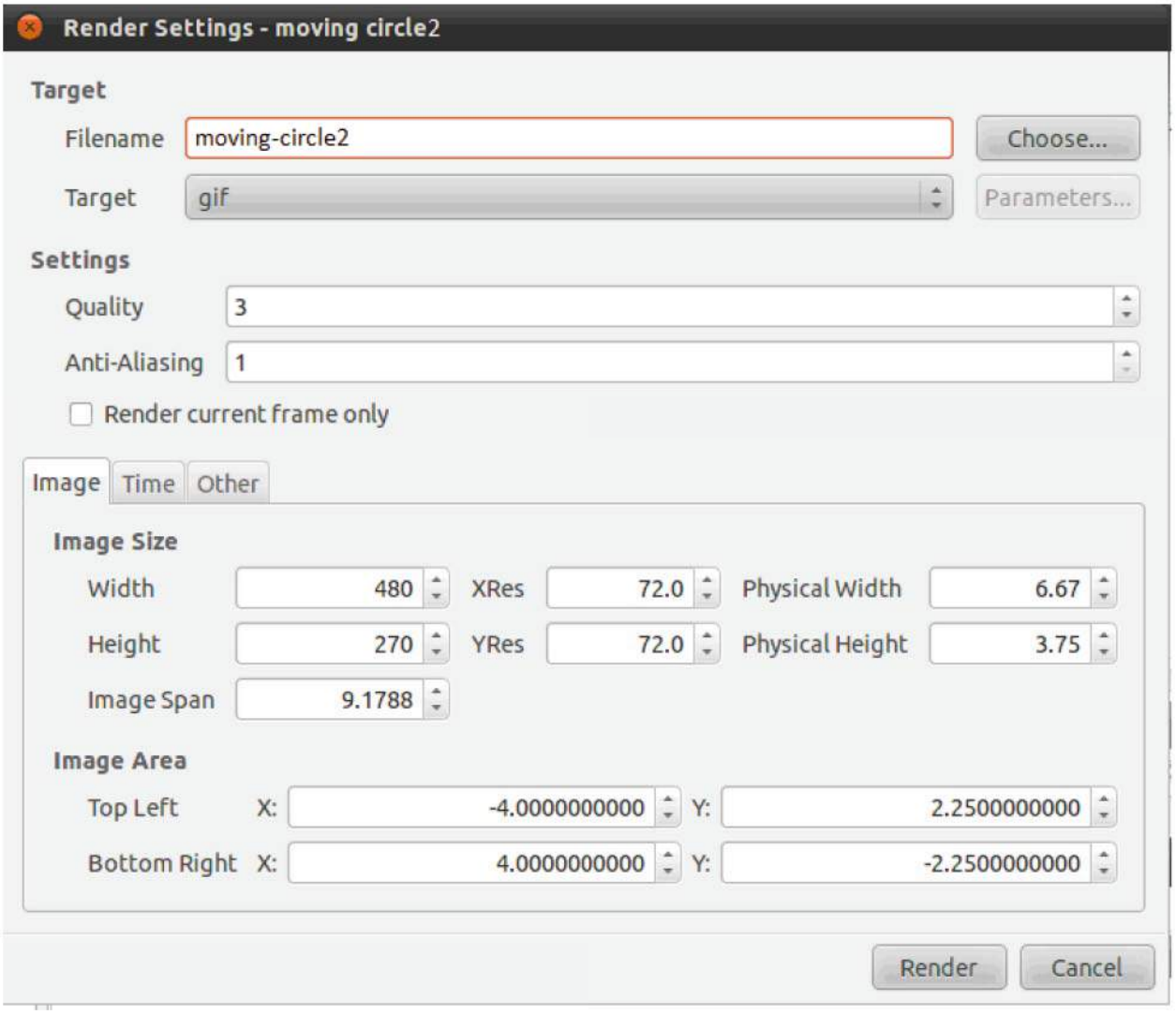


Figure 3.13(a) : Render the file

- Open moving-circle.gif file in web browser to see your animation. Figure 3.13 (b) shows the output.

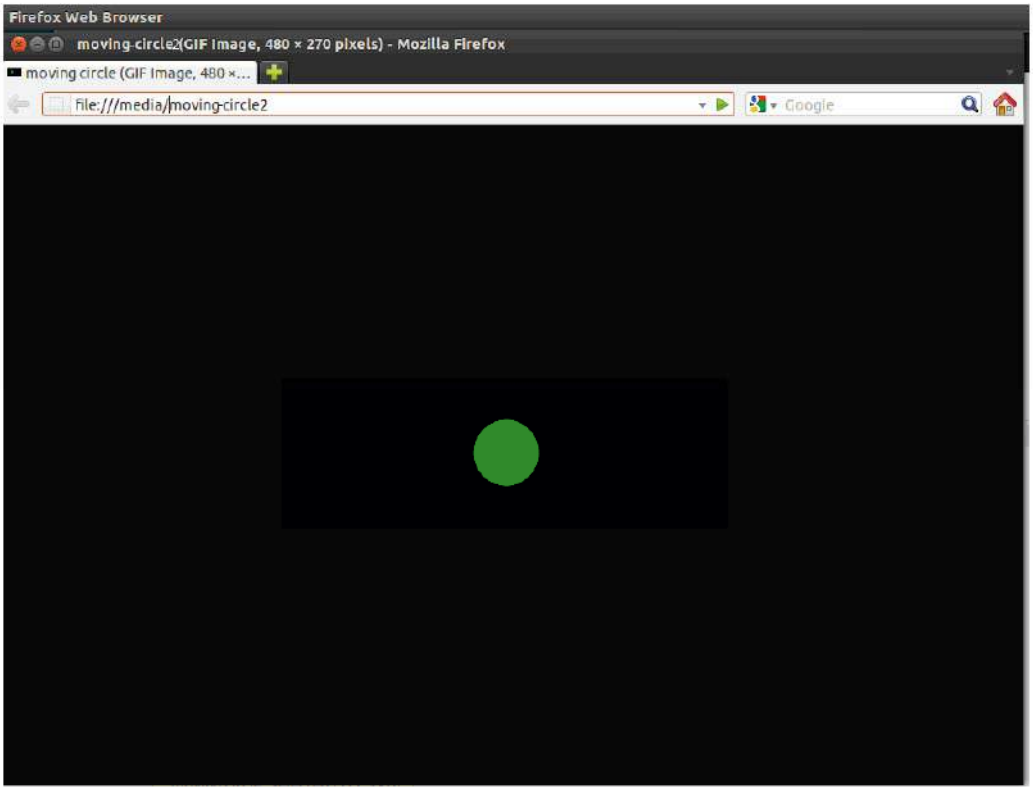


Figure 3.13(b) : Output displayed in web browser

Using Time Loop

If you want your animation to repeat again use the time loop. Let us take an example of a bouncing ball to understand Time loop. We will also see how to duplicate a keyframe which saves time in positioning the object in the animation all over again. In the example we will bounce the ball on the wall. The animation will be created twice using duplicate keyframes concept. And then use the time loop to show the bouncing continue till the end time of the animation.

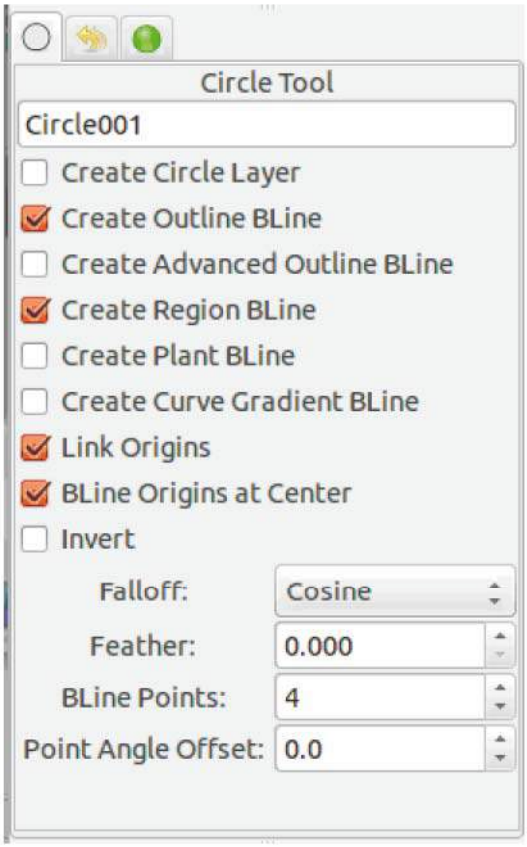


Figure 3.14: Circle tool options

- Create a new file.
- Select the Circle tool and in the tool options select only the region and the outline layer as in figure 3.14. Draw a circle on the canvas as shown in figure 3.15.

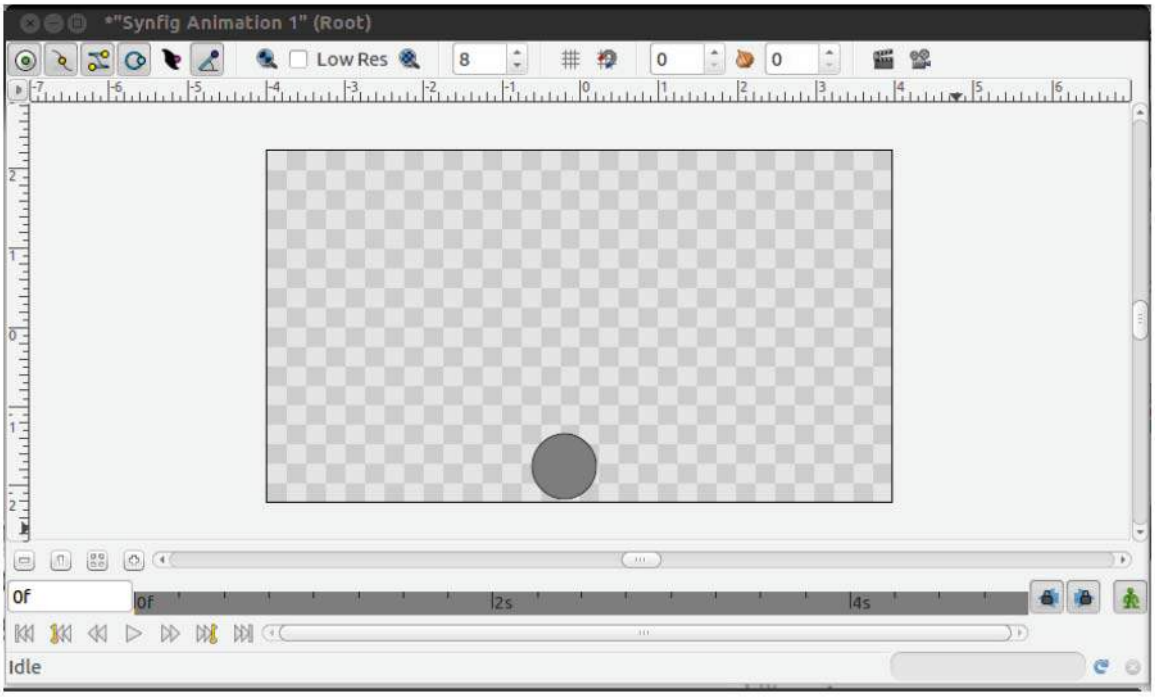


Figure 3.15 : Ball drawn using Circle tool

- Turn on the animate mode. In the timeline, record the position at "0f" by moving the ball slightly from its position. Now select the keyframe panel and add a keyframe by pressing  at the bottom of the keyframe panel. Figure 3.16 shows the keyframe added at "0f".

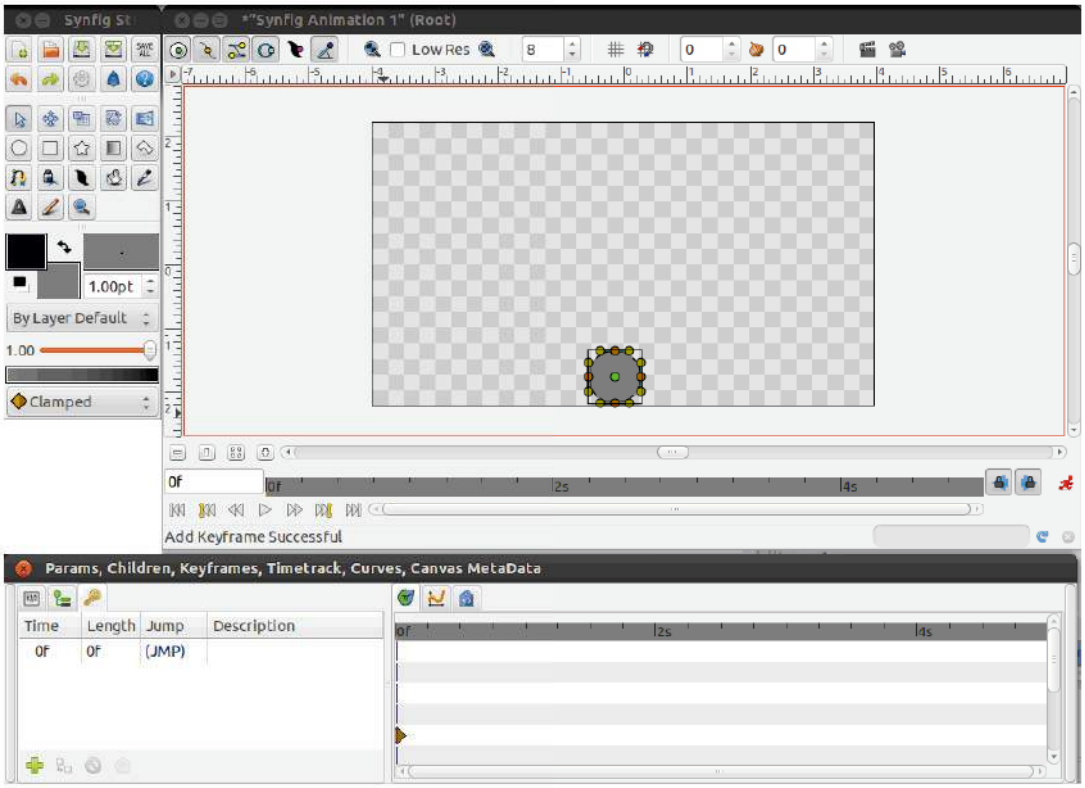


Figure 3.16 : Keyframe at 0f

- Now in the timeline come to 6f and drag the ball to a new position. Add a new keyframe at "6f" as shown in figure 3.17.

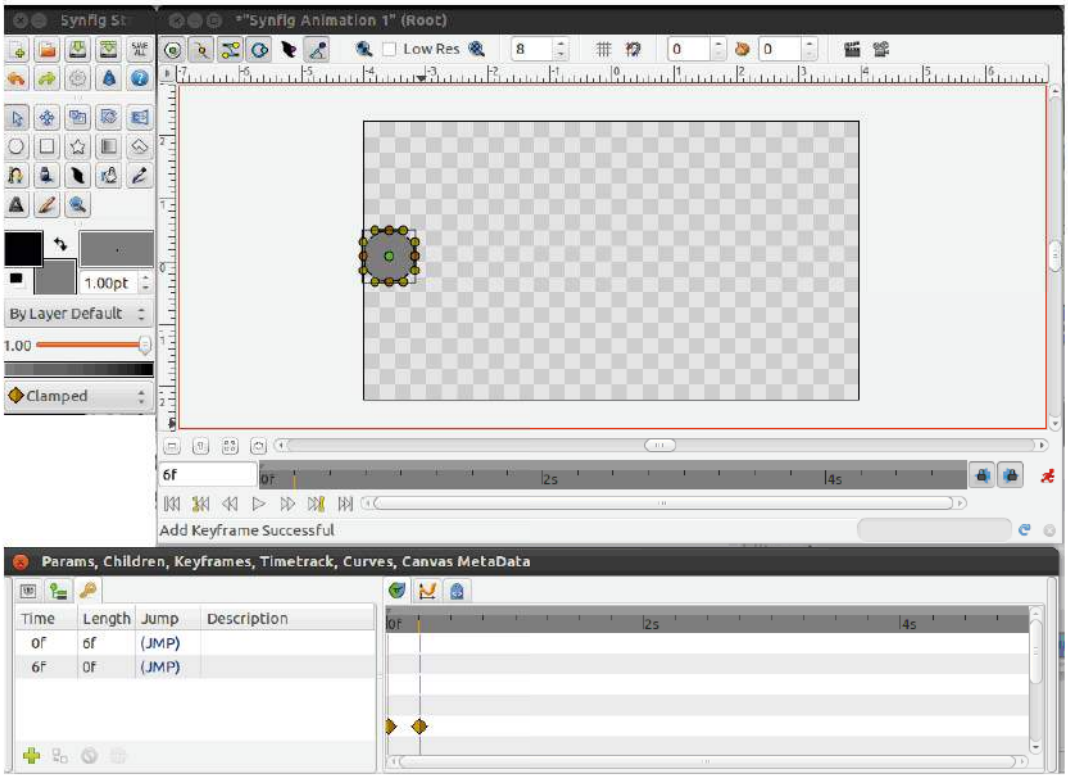


Figure 3.17 : Keyframe added at 6f

- Similarly, go to "12f", "18f" and drag the ball to the location as shown in figure 3.18 and 3.19. Add a keyframe at both the timing location.

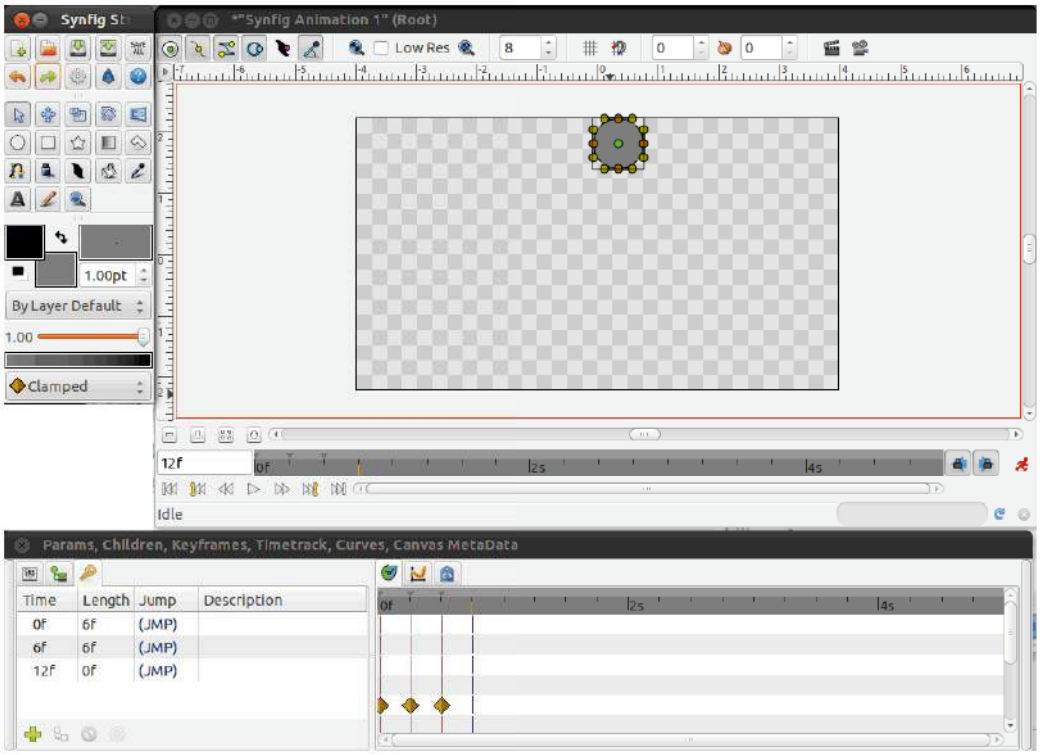


Figure 3.18 : Keyframe added at 12f

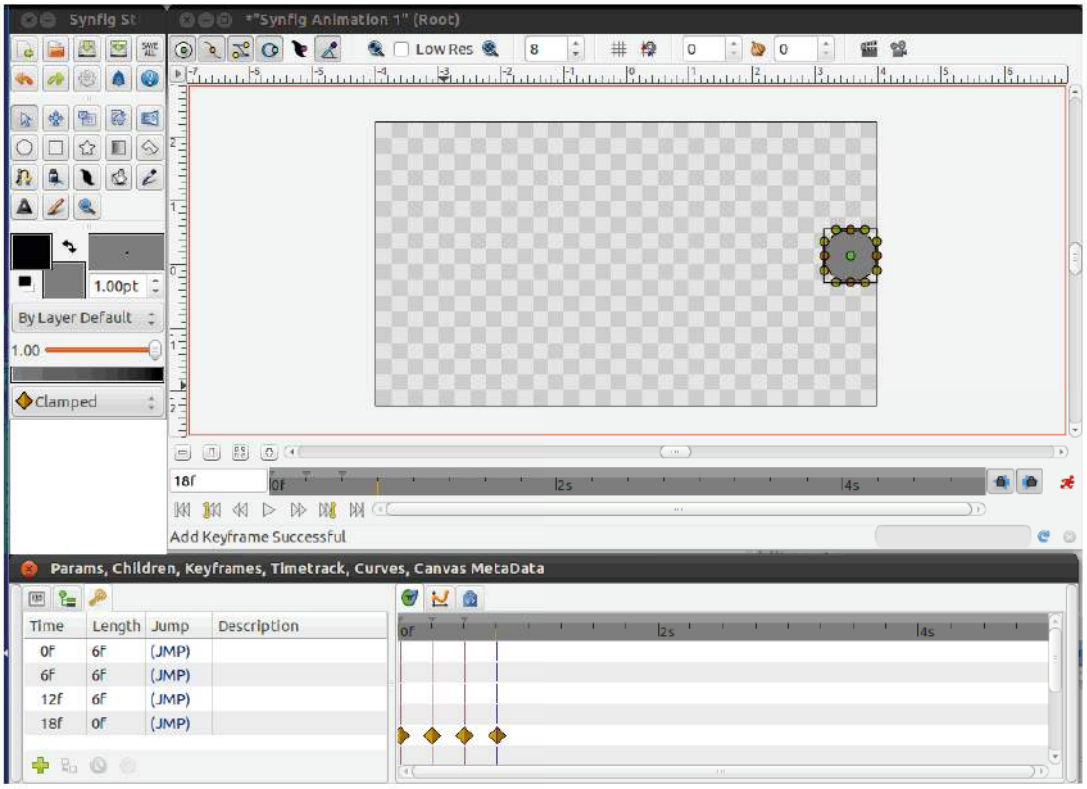


Figure 3.19 : Keyframe added at 18f

- Now come back to "1s" in the timeline, here we need the ball to come to its original position which we had at "0f". So we need to duplicate the keyframe at "0f". Select the "0f" keyframe from the keyframe panel and press duplicate keyframe as shown in figure 3.20. This will add a new duplicate keyframe at "1s" as shown in figure 3.21.

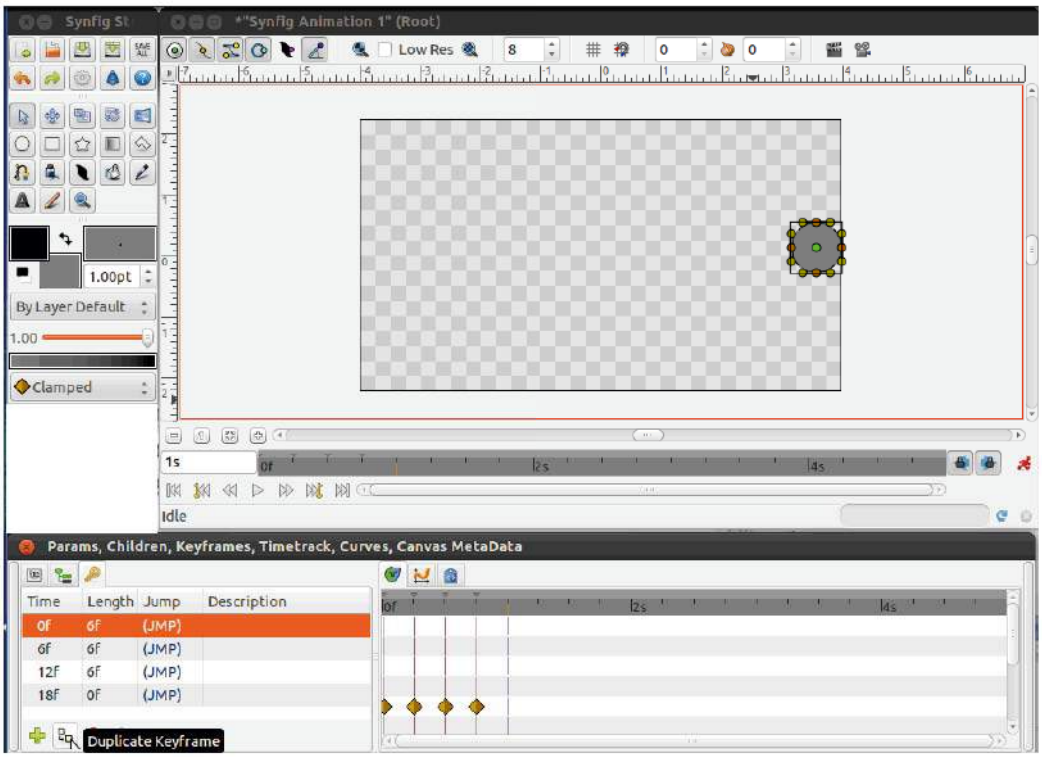


Figure 3.20 : Add a duplicate keyframe at 1s

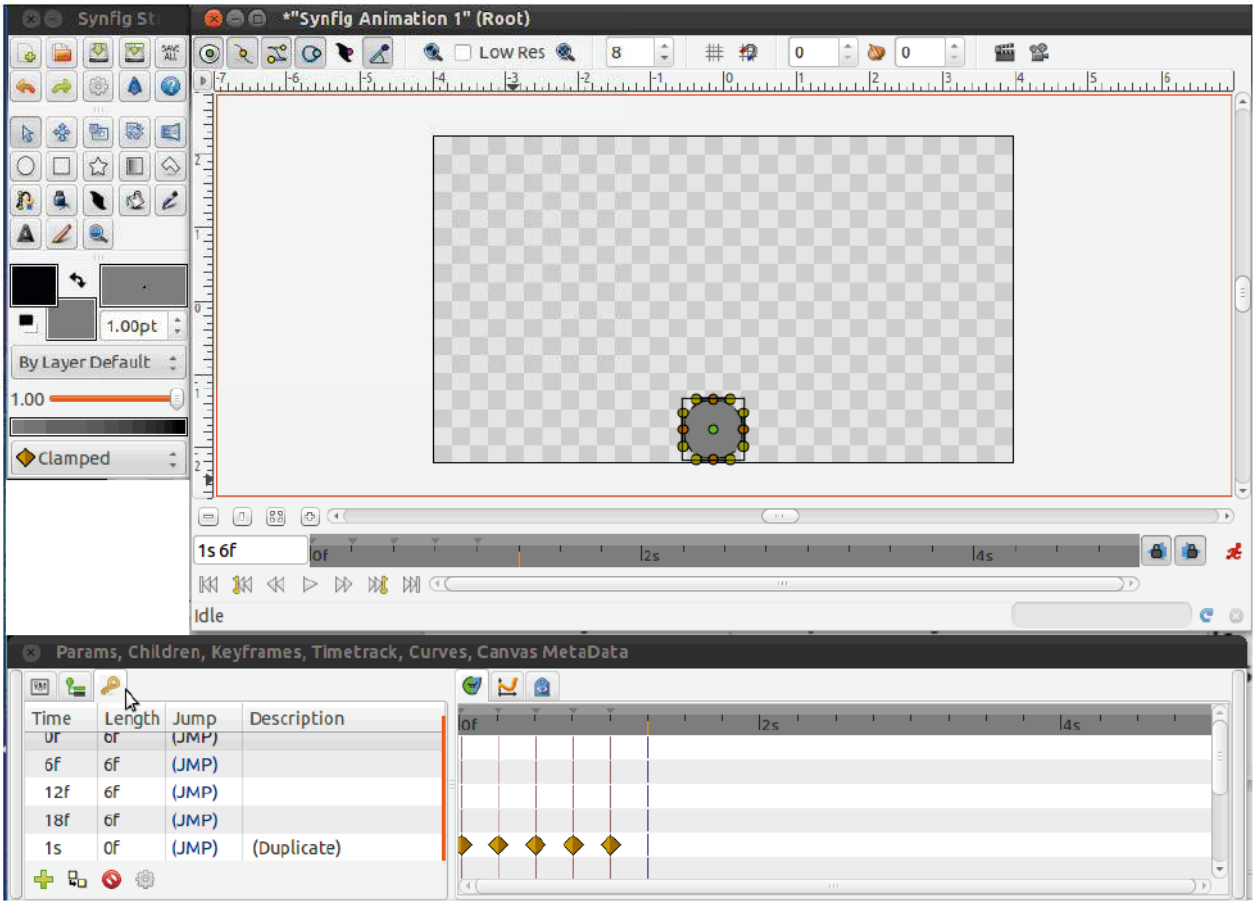


Figure 3.21 : Panel showing duplicate keyframe added

- Likewise to animate the ball twice we need to create the duplicate keyframes at 1s6f, 1s12f, 1s18f and 2s. Figure 3.22 shows the duplicate keyframes added.

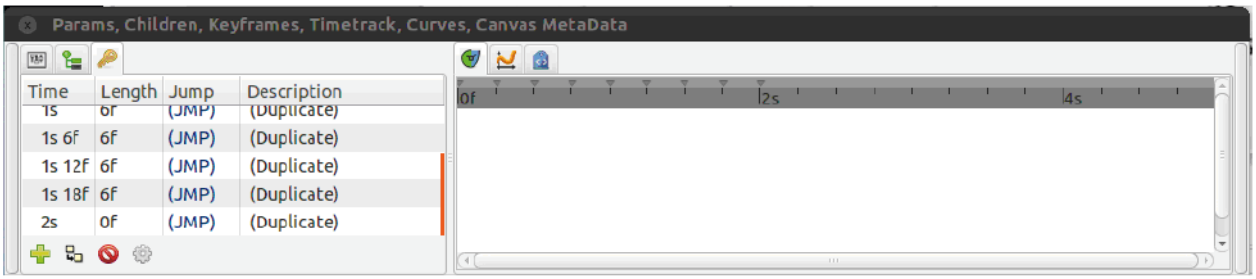


Figure 3.22 : Duplicate keyframes added

- Now preview the animation using File → Preview. You will see the ball bouncing twice till 2 seconds and then stop. We want the animation to go on till the end time. In our case the end time is 5 seconds.

  We will insert a new time loop layer above the circle layer.

- Select topmost circle outline layer → Right click → Add new layer → Other Time loop. This will add a new layer as shown in figure 3.23.

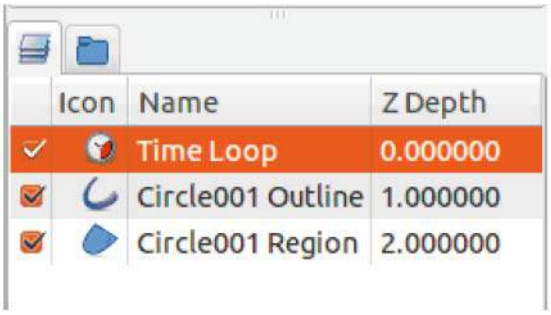


Figure 3.23 : Inserting time loop layer

- The time loop parameters in figure 3.24 shows the duration which denotes the frames or seconds that are looped. We can change it to 2s as our animation is playing twice till 2s.

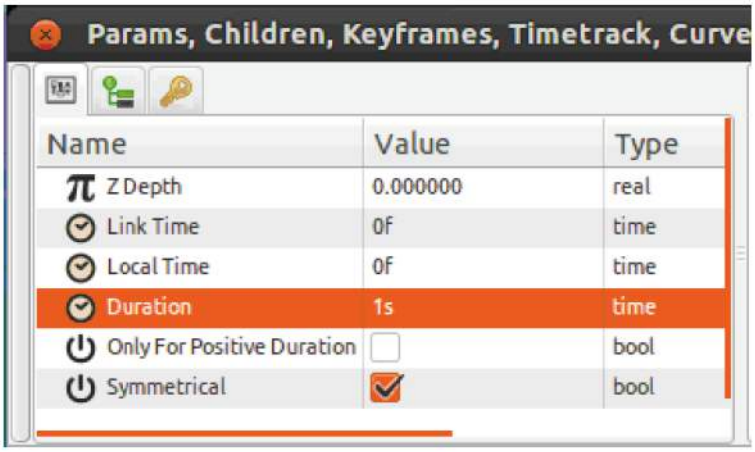


Figure 3.24 : Duration parameter in the Time loop

- Preview the animation and you will see that the animation goes on playing till the end time. If you change the endtime to "10s" in the file properties window then the animation will go on playing till 10s.
- Save and render the file. Figure 3.25 shows the output on web browser.

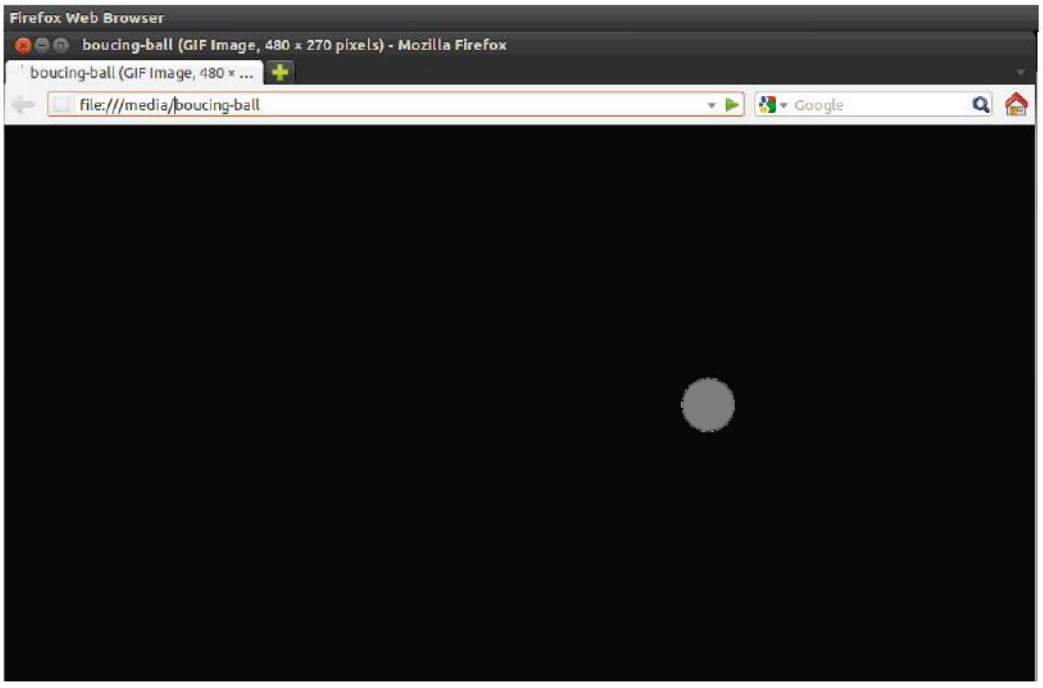


Figure 3.25 : Output displayed in web browser

Summary

In this chapter we learned how to create animation. Animate edit mode helps in recording the changes made to the object's position at different time using the timebar. When not in animate edit mode, any changes made to the object or its parameters will not have an effect on the animation. The concept of keyframe and waypoint was discussed. We also learned how to repeat an animation till the end time using the time loop layer. In the next chapter we will learn more about the concept of layers and using it to give effects to the objects.

EXERCISE

1. Define Animation and state its type.
2. Explain frame by frame and cel animation.
3. What is a keyframe?
4. What is tweening?
5. Explain animate edit mode.
6. State the purpose of timebar.
7. What is morphing?
8. What is kinematics? Give three examples.
9. **Choose the most appropriate option from those given below :**

   (1) Which of the following terms represent the objects to be displayed over a period of time?

   (a) frame (b) object
   (c) keyframe (d) panel

   (2) Each small change in the image is created and then displayed in sequence in which of the following techniques?

   (a) frame by frame animation (b) cel animation
   (c) tweening (d) morphing

   (3) Which of the following timings is default setting for the End time?

   (a) 2s (b) 5s
   (c) 3s (d) 6s

   (4) Which of the following terms represents a frame that defines the change to an object's properties?

   (a) timeline (b) layer
   (c) keyframe (d) panel

   (5) Which of the following represents an indicator used to represent waypoint?

   (a) circle (b) square
   (c) rectangle (d) diamond

(6) In animate edit mode the canvas outline colour changes to which of the following colours?

(a) orange (b) yellow

(c) red (d) green

(7) Which of the following terms defines the length of time in the document?

(a) frame (b) keyframe

(c) layer (d) tweening

(8) Which of the following is correct entry in the timeslider?

(a) 3f 9f (b) 3s 9f

(c) 3f 9s (d) 3s 9s

(9) Which of the following terms represent an effect wherein the image transforms into another?

(a) morphing (b) tweening

(c) kinematics (d) keyframe

(10) Which of the following is the default setting for frames per second?

(a) 15 (b) 30

(c) 24 (d) 28

(11) Which of the following terms represent the study of movement and motion of objects that have joints such as walking man or running leopard?

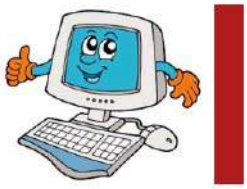
(a) animation (b) kinematics

(c) tweening (d) morphing

LABORATORY EXERCISES

1. Create an animation showing the star moving from top to bottom.
2. Create an animation showing a ball bouncing on the floor. Use time loop to continue the animation till the end time.
3. Create an animation showing a circle moving from right to left and at the same time star moving from left to right on the canvas.
4. Create an animation showing three balls bouncing at different speeds.

4 Introduction to Layers



A beautiful sight of a rising sun is always soothing to eyes. If this sunrise is observed in mountains, the sight becomes more pleasurable. This scene consists of many objects like mountains, sun, birds, trees etc. Assume that we want to create such an animation. Each of these objects when drawn on the canvas will be either above or below another object. For example, sun rises from behind the mountains, so the object 'sun' should be placed behind the object 'mountain'. Similarly, if the mountain is behind the hut, the object 'hut' will be placed above the object 'mountain'. Likewise we can place the other objects in our example like the trees and birds above or below the other objects. In this chapter we will learn about a concept named "layers" that will help us to create the scene mentioned here.

Layers

A "layer" in Synfig refers to different levels on which we can place our drawing objects or pictures, stacked above and below each other. Top layers will hide bottom layers. We can think of layers as sheets of paper which are cut into different shapes. On one layer we have a paper-sheet shaped like a star. On second layer we have a paper-sheet shaped like a square. The square might overlap and hide some part of the star.

Use of layers gives us the freedom to work with each object individually. When modified it does not disturb the other objects either above or below it. Layers can also be rearranged, allowing us to change the order in which objects are displayed, thus changing the appearance of our scene.

In Synfig, each individual element of the canvas is broken down into layers. For example, when we had drawn a circle it appeared on a different layer. The layers in Synfig are different from other animation software in the following two aspects:

1. Every object, element, and effect have their own layer. We do not have a layer with multiple effects. In other words a layer represents a single primitive such as object, outline or image.
2. The upper layer can change the behavior of the layers below it. It can distort or modify the layers beneath. This is done by using effect or filter layers.

There are different types of layers such as gradient, blur, distortion, filter and others.

Each layer has its own set of parameters. We can see the parameters of a layer by first selecting the layer from the layers panel and then looking into the params panel. Let us understand layers

concept by taking an example. Assume that, we want to give a gradient effect to a circle layer. Perform the steps mentioned to achieve gradient effect.

- Create a new file.
- Create two circles of different sizes using the circle tool. Keep the color of the circle as white so as to see the effect of gradient. Figure 4.1 shows these two circles.

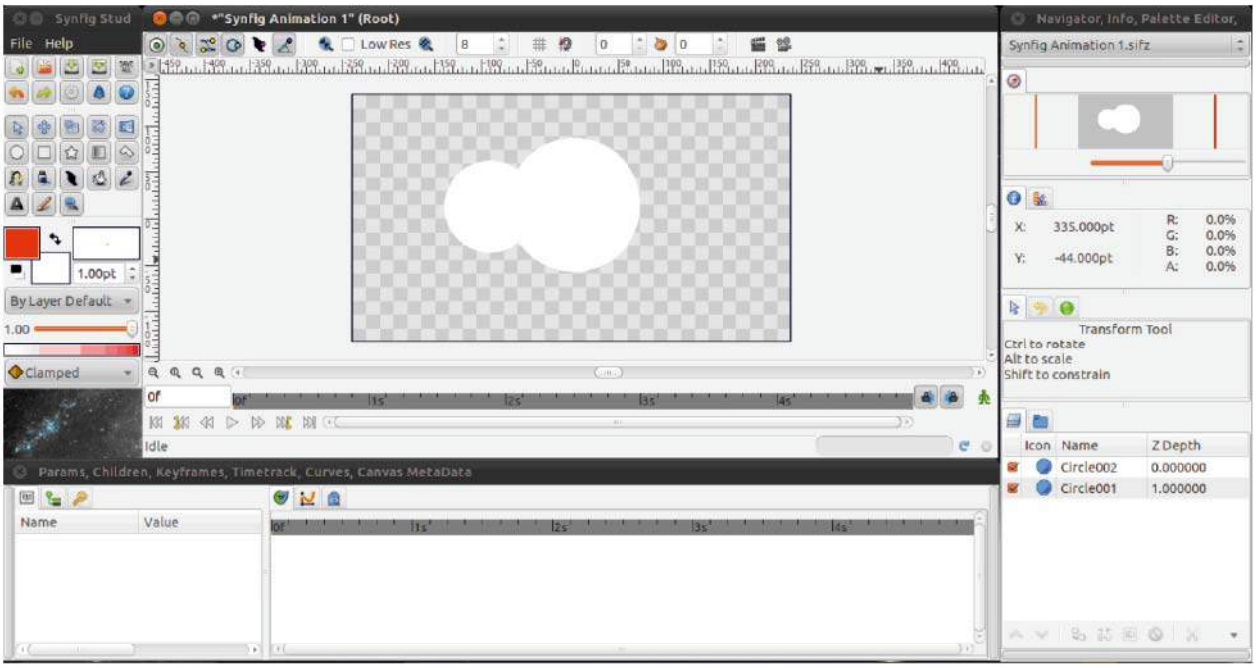


Figure 4.1 : Two circles of different size

- Select the gradient tool from toolbox. Select the outline color of your choice. Here we have selected the outline color as red. Now drag the gradient over the area as shown in the figure 4.2.

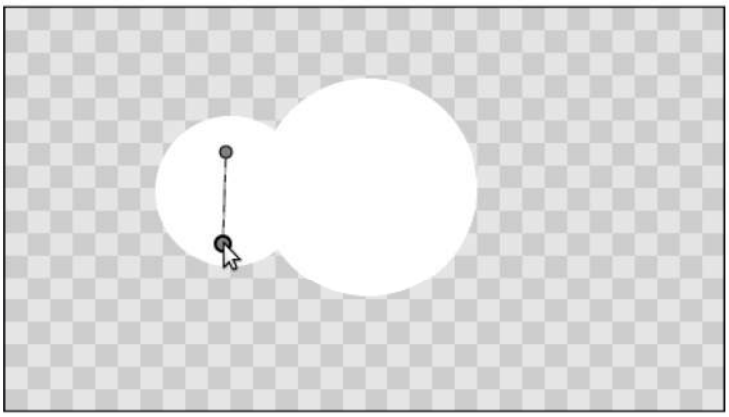


Figure 4.2 : Add gradient

- You see the entire canvas is covered by the gradient as in figure 4.3. This can be useful if you want to use the gradient as the background layer. But we want the gradient to be applied only onto the objects.

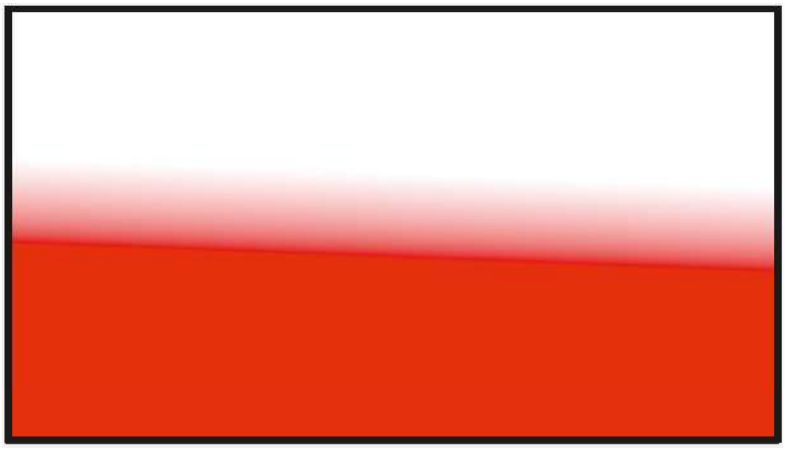


Figure 4.3 : Effect of Gradient on the canvas

- Select the gradient layer and go to the params panel. You can see "blend method" parameter. Blend mode defines how the image is layered on everything below it. By default, the setting is "Composite", which simply displays the content of the layer. In the blend method parameter change the setting to "straight onto" as shown in figure 4.4. Now you can see that the gradient is applied to the object layers below it. But we want the gradient to be applied only to a single object and not all the objects below the gradient layer. For this we will be using encapsulation.

  Encapsulation means grouping things together. If we want to apply gradient to a particular object or a group of objects then we need to encapsulate them into a layer. Thus, using encapsulation we can apply the gradient to one layer without applying it to the layers beneath it.

  So, in our example we need to select the gradient and the object (to which the gradient is to be applied) and encapsulate them together.

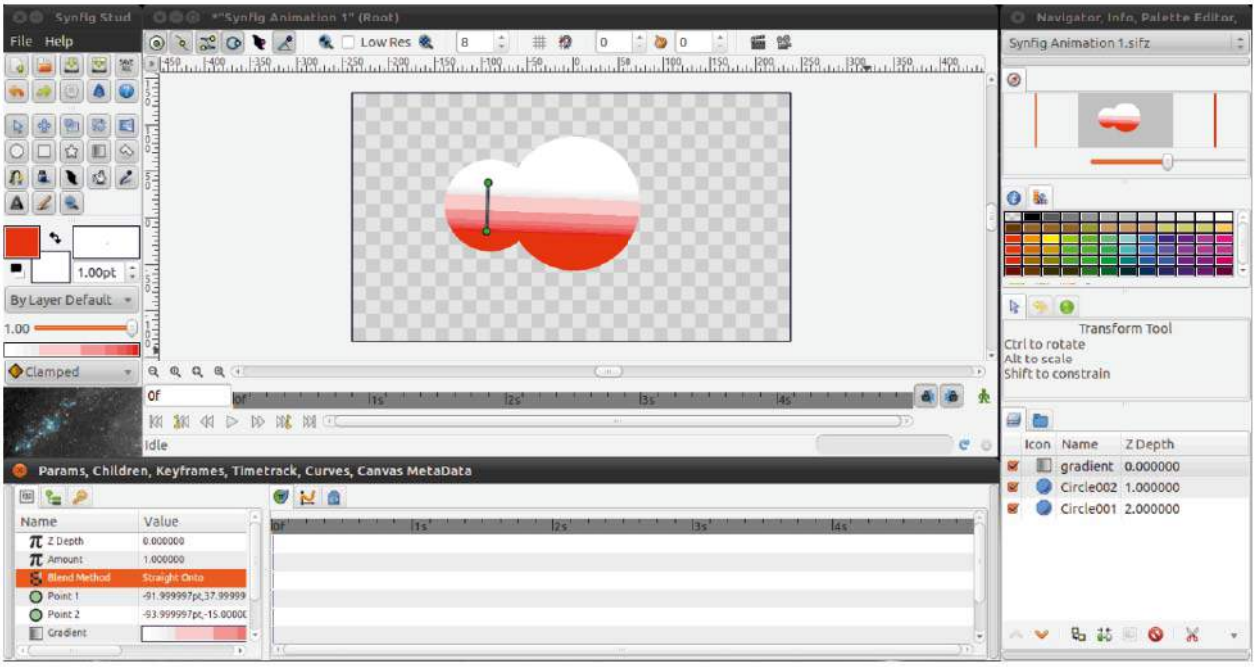


Figure 4.4 : Use of "straight onto" blend method

- Select the gradient layer and the circle layer (press and hold the CTRL key to select two layers) on which we want to apply the gradient. Right click → Encapsulate. This will create a new layer named "inline canvas" as shown in figure 4.5. You can see a small triangle near the inline canvas. Click on this and the inline canvas will expand to show you the two layers: gradient and circle as given in figure 4.6. Figure 4.7 shows the canvas with gradient applied only to the smaller circle.

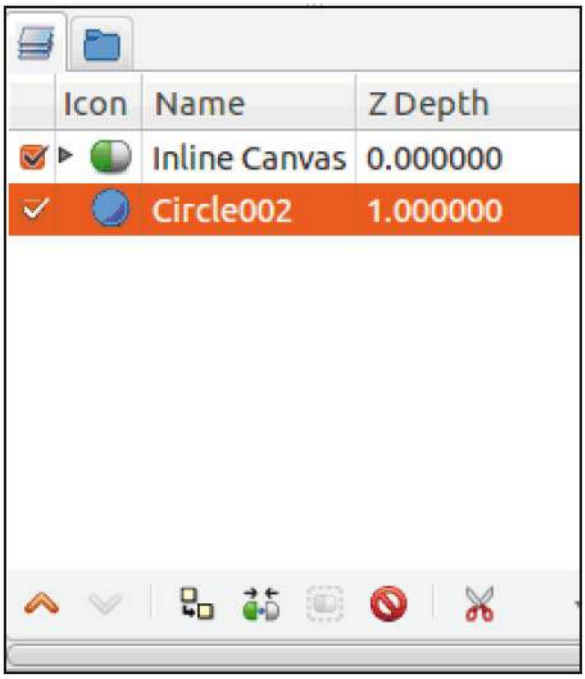


Figure 4.5 : Inline canvas layer created

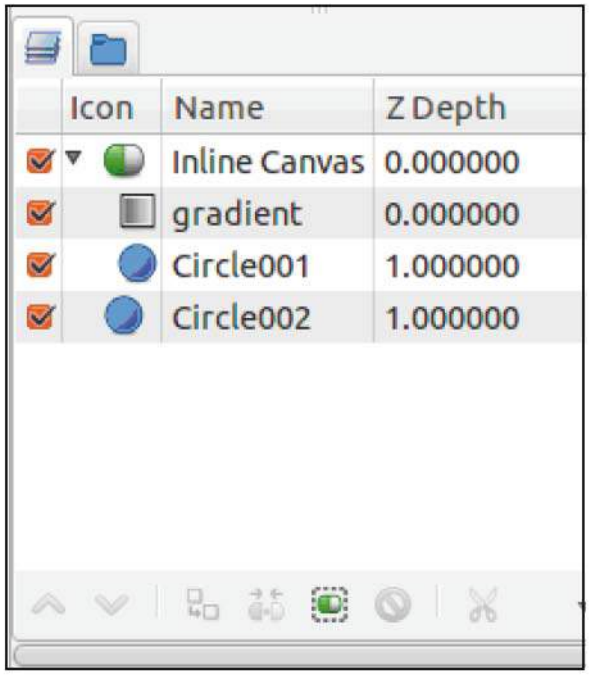


Figure 4.6 : Expanded Inline canvas layer

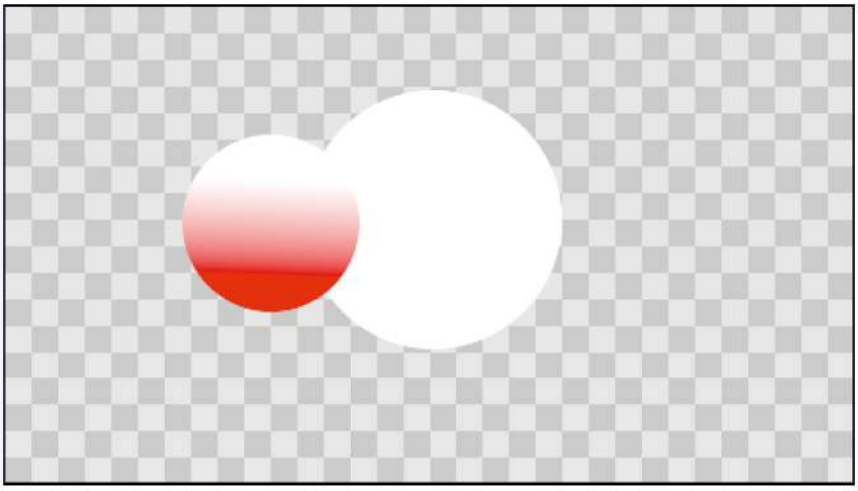


Figure 4.7 : Canvas showing the gradient effect after encapsulation

Thus we have seen how the gradient is applied to the objects using encapsulation. To make it more clear let us add one more circle using the circle tool as shown in figure 4.8.

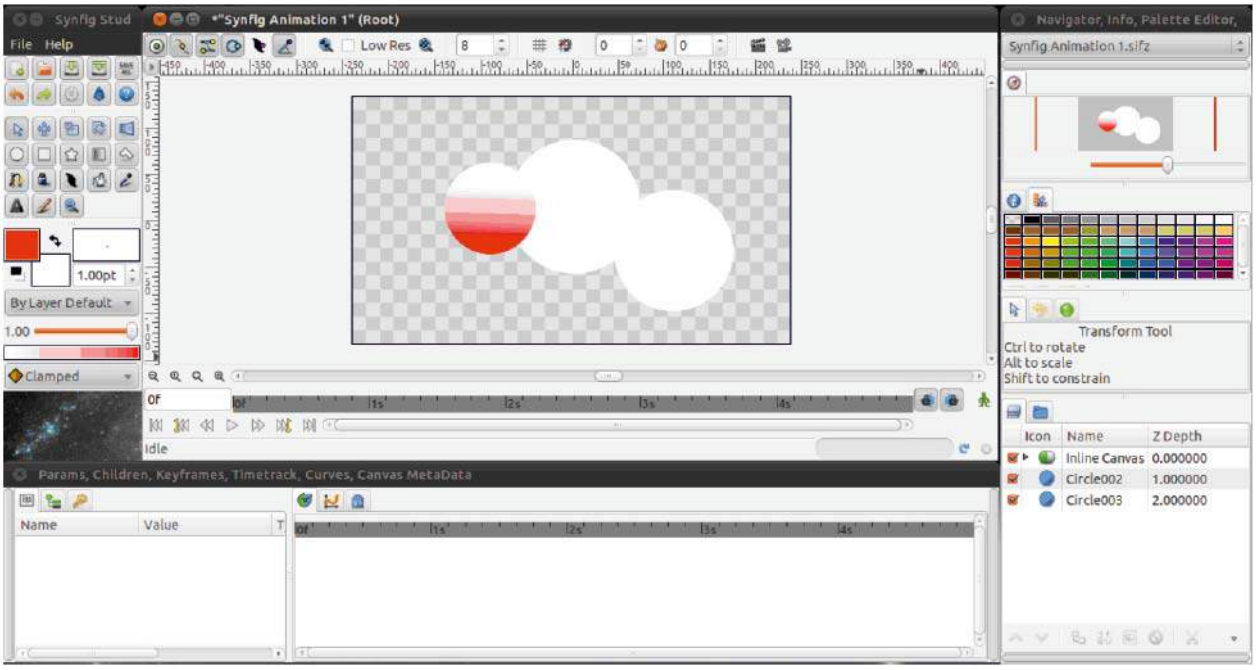


Figure 4.8 : Create a new circle layer

Now drag this new circle layer into the inline canvas layer. We place the circle layer just below the gradient layer and the effect will be as shown in figure 4.9. Here you can see the gradient is applied to all the objects inside the encapsulated area.

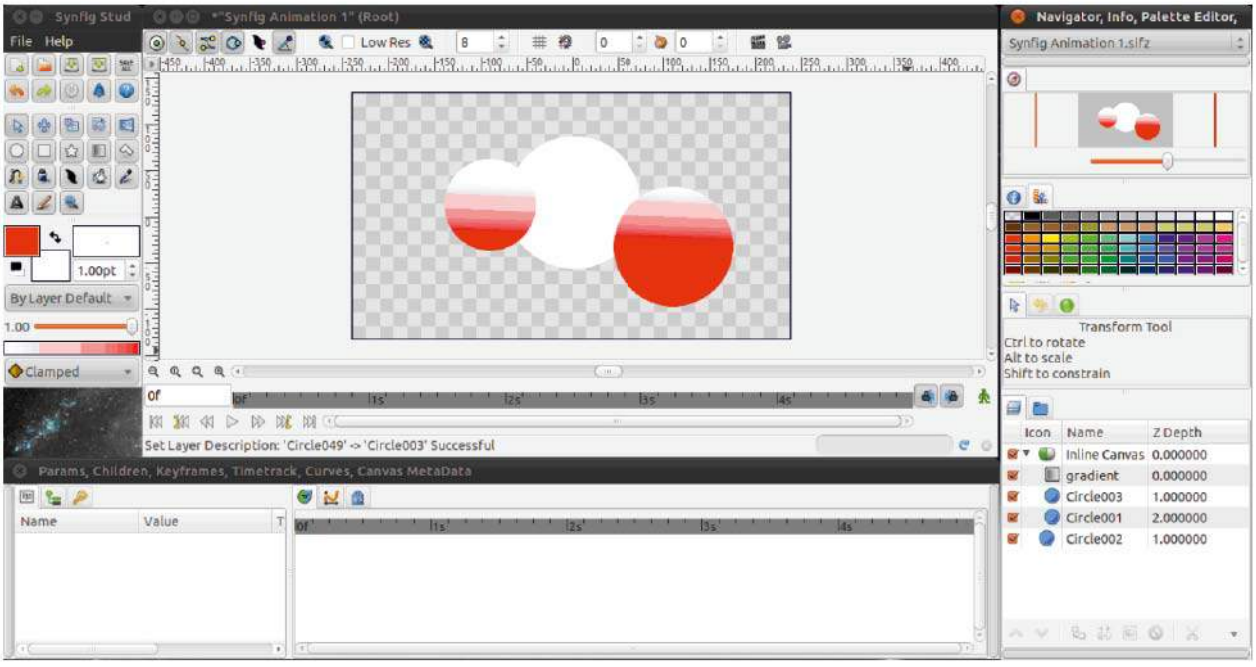


Figure 4.9 : New circle layer placed below the gradient layer

If we reposition the gradient layer using the lower layer button as shown in figure 4.10 (a), the gradient will now be applicable to only one of the circle layer. In other words, we can say that the gradient layer has an effect on only the layers that are below it in the encapsulated area. Figure 4.10 (b) shows the effect of this repositioning.

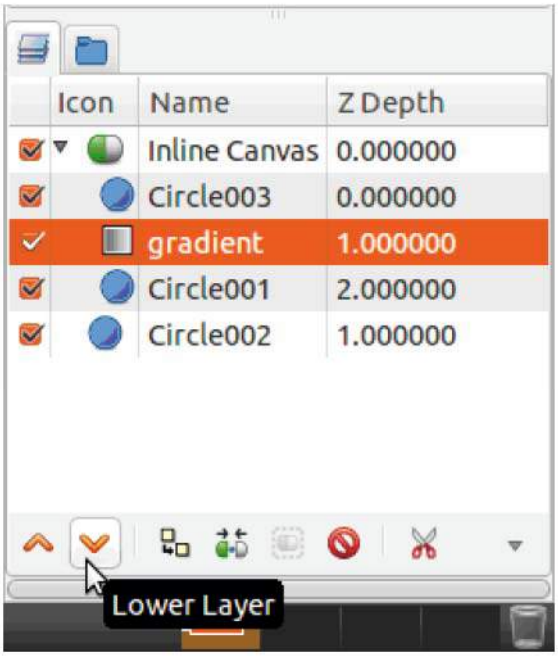


Figure 4.10(a) :
Reposition gradient layer

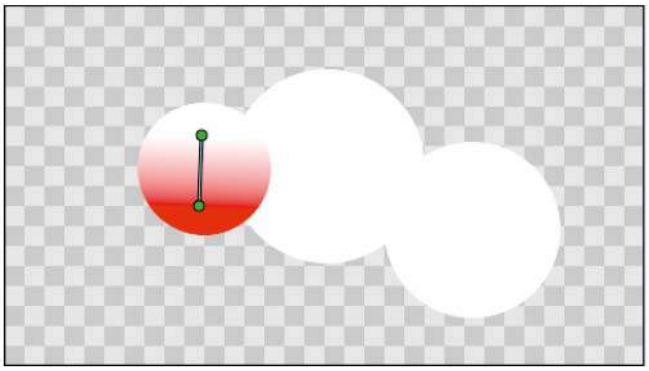


Figure 4.10(b) :
Effect of repositioning gradient layer

The important point to remember here is that a layer can only modify the data that it gets from the layer directly below it. For example if we used a Blur Layer on top of the layers inside the inline canvas, it would just blur the inline canvas layers and not every layer that lies under the inline canvas.

Rotation on a specified path

Using the concept of encapsulation, let us discuss another example of a ball rotating around a particular path. In our animation example of the circle moving from left to right described in previous chapter, we defined the starting point and the ending point. Here, Synfig automatically assumes the path to be straight. But in our example we want to move the object as per our defined path. Follow the steps mentioned to create user defined path.

- Create a new file.
- Let us first create our path for rotation using the Bline tool. Select the Bline tool and draw an oval shaped path as shown in the figure 4.11. In the Tool options panel select only the *Create Outline Bline* option.

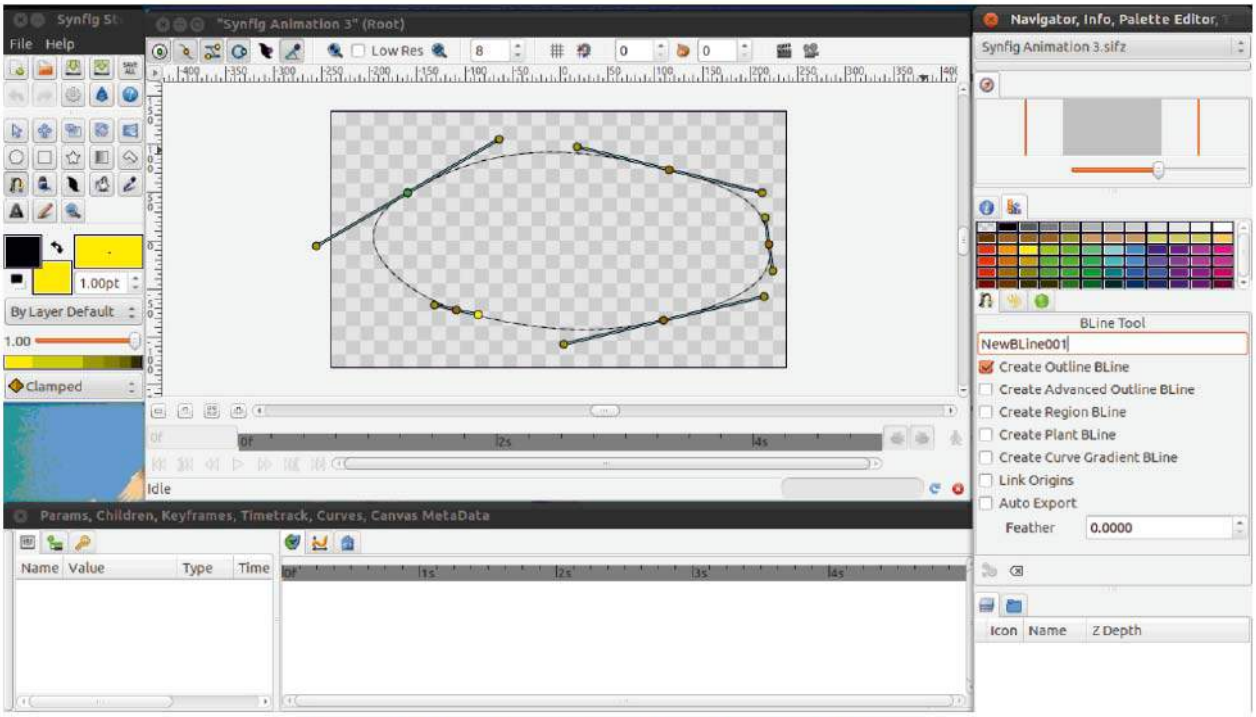


Figure 4.11 : Creation of path using Bline tool

- Select the transform tool to finish the Bline. (Pressing Esc will clear the Bline).
- Now select the circle tool. Using the palette editor select a color of your choice and draw a circle on the canvas as shown in figure 4.12.

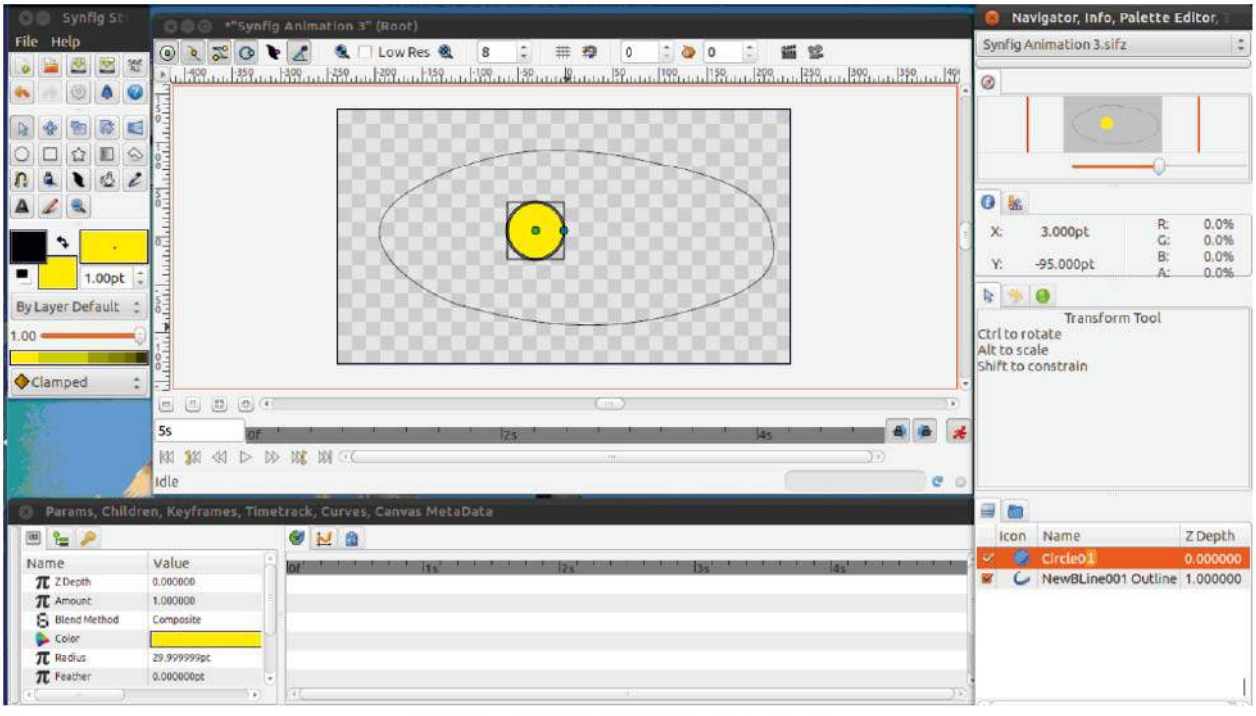


Figure 4.12 : Create the circle

- Let us add a new rotate layer above the circle layer. In the layers panel select the circle layer → Right click → New layer → Transform → Rotate. This will add a new layer named rotate above the circle layer as shown in figure 4.13 (a) and 4.13 (b).

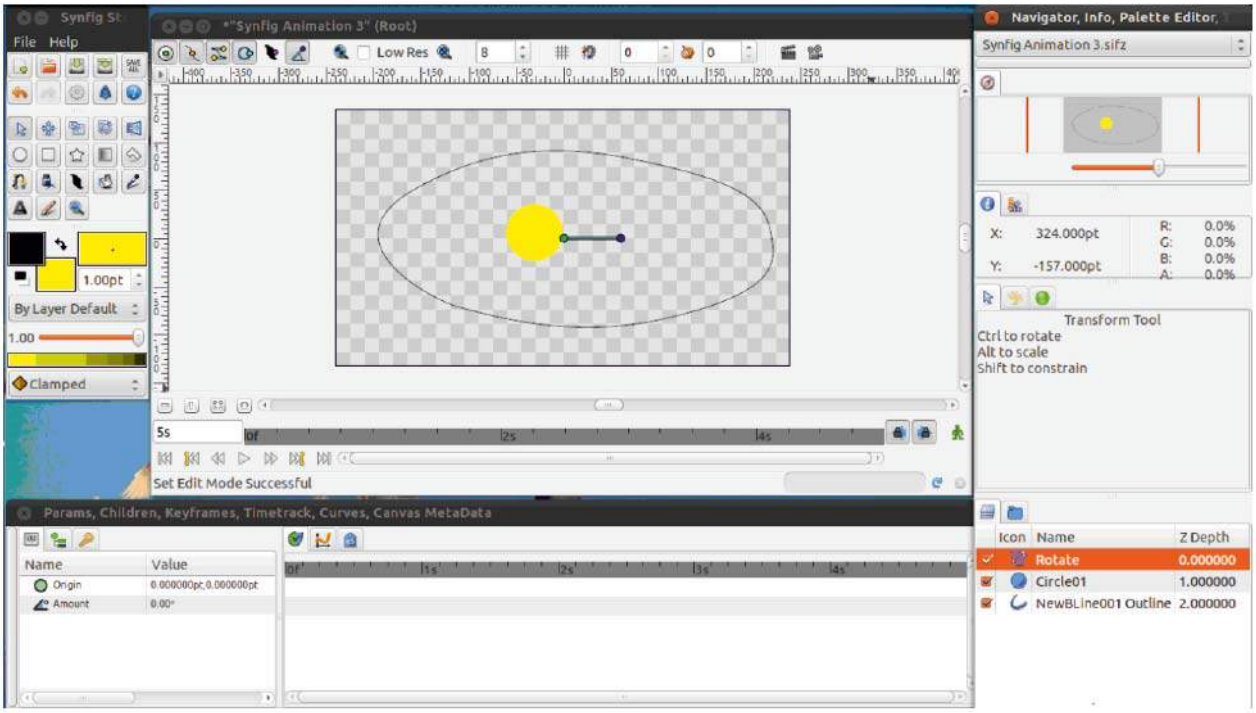


Figure 4.13(a) : New rotate layer above the circle layer

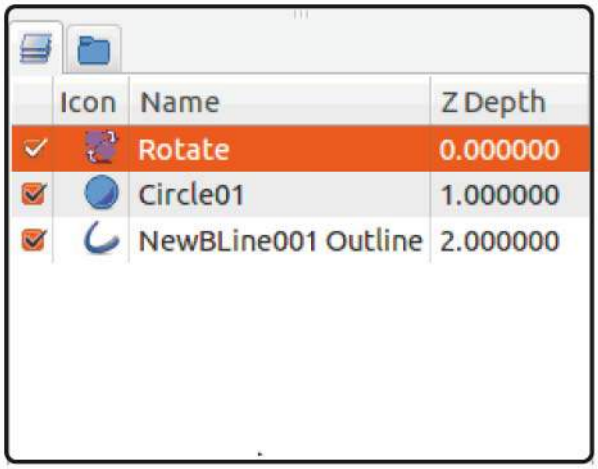


Figure 4.13(b) : Layers panel showing rotate layer above the circle layer

As we need to rotate the circle, we have added a rotate layer above the circle layer.

- Now we will encapsulate the circle layer and rotate layer. To select both the layers, click on the rotate layer in the layers panel, press shift and select circle layer.
- Right click and select encapsulate. Thus the rotate and the circle layer are encapsulated into a layer

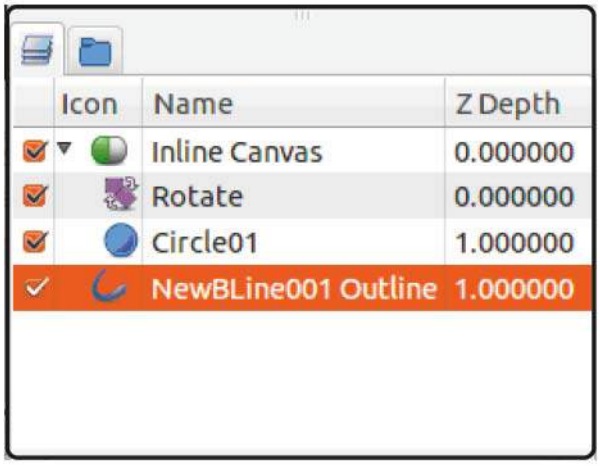


Figure 4.14 : Encapsulate rotate and circle layer

named inline canvas. You can change the name of the encapsulated layer. Figure 4.14 shows the encapsulated layer 'Inline Canvas'.

- Select the rotate and the circle layer and bring their positioning duck (green duck) near to one another as shown in figure 4.15.

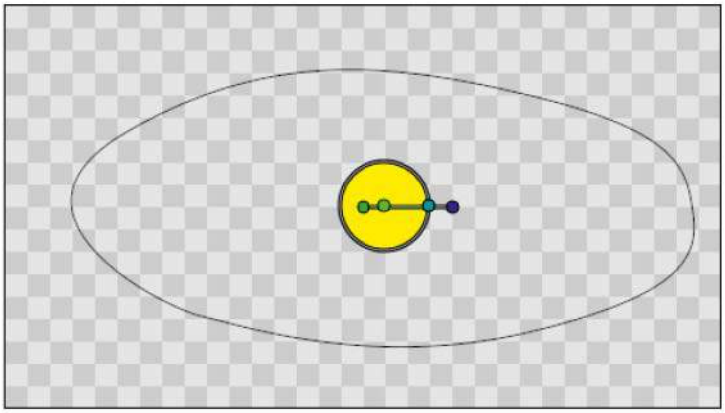


Figure 4.15 : Dragging the green ducks near to one another

- Select the inline canvas layer and select its positioning duck (green duck). Hold the CTRL key and choose the rotate layer. Continue to hold the CTRL key and select the blue duck on the rotation layer. Continue to hold the CTRL key and select the Bline layer. Now right click on the Bline in the canvas where you want the circle to start from. Select the option 'Link to Bline'.
- Link to Bline feature is used to attach a vertex of an object to the vertex of a Bline. The vertex gets stuck to the Bline vertex and can be moved along with it. Figure 4.16 shows the effect of using link to bline option.

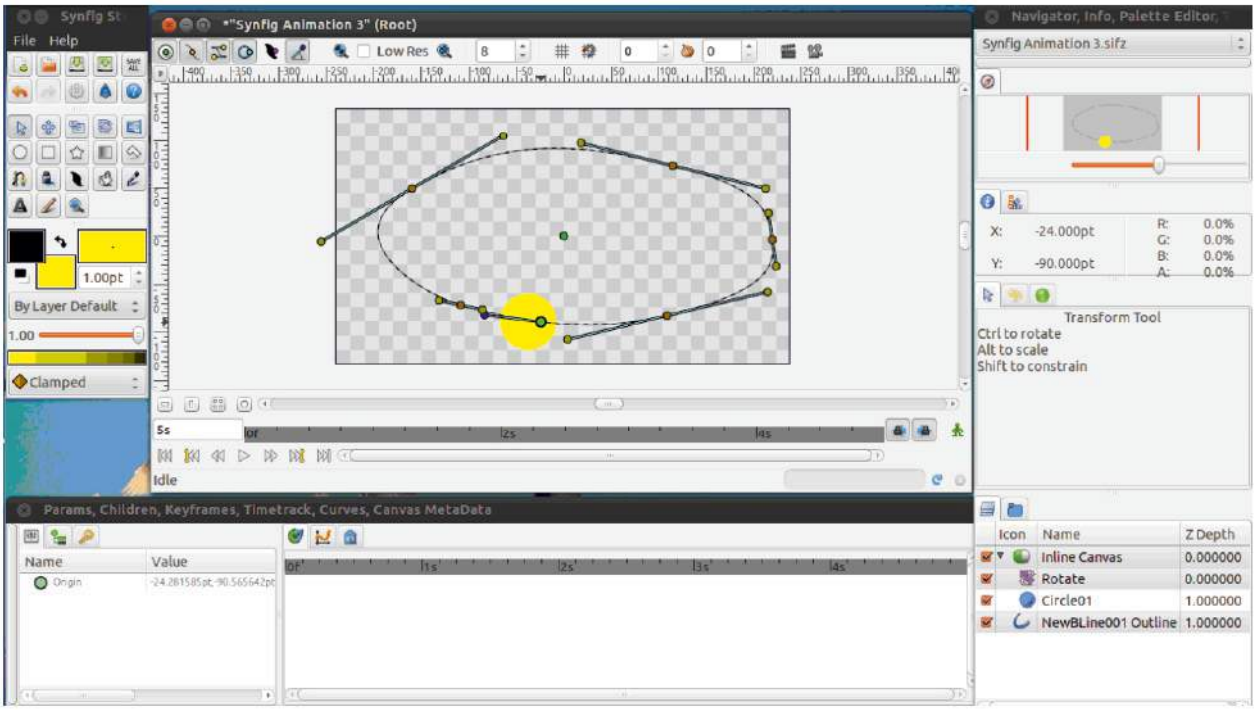


Figure 4.16 : Link to Bline option selected

- Click on the small triangle near the inline canvas layer to close it. Select the inline canvas layer as shown in figure 4.17.

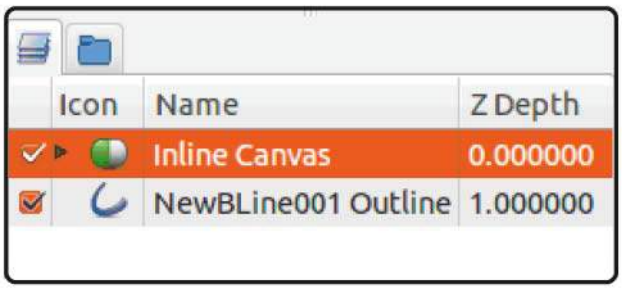


Figure 4.17 : Select Inline canvas layer

- Turn on the animation editing mode.
- In figure 4.18 you can see the red line around the canvas indicating that the animation mode is on. Drag the circle a little bit using the positioning duck (green) to record the starting position on the "0f" mark in the timeline as in figure 4.18. You can see a waypoint is added at "0f" mark.

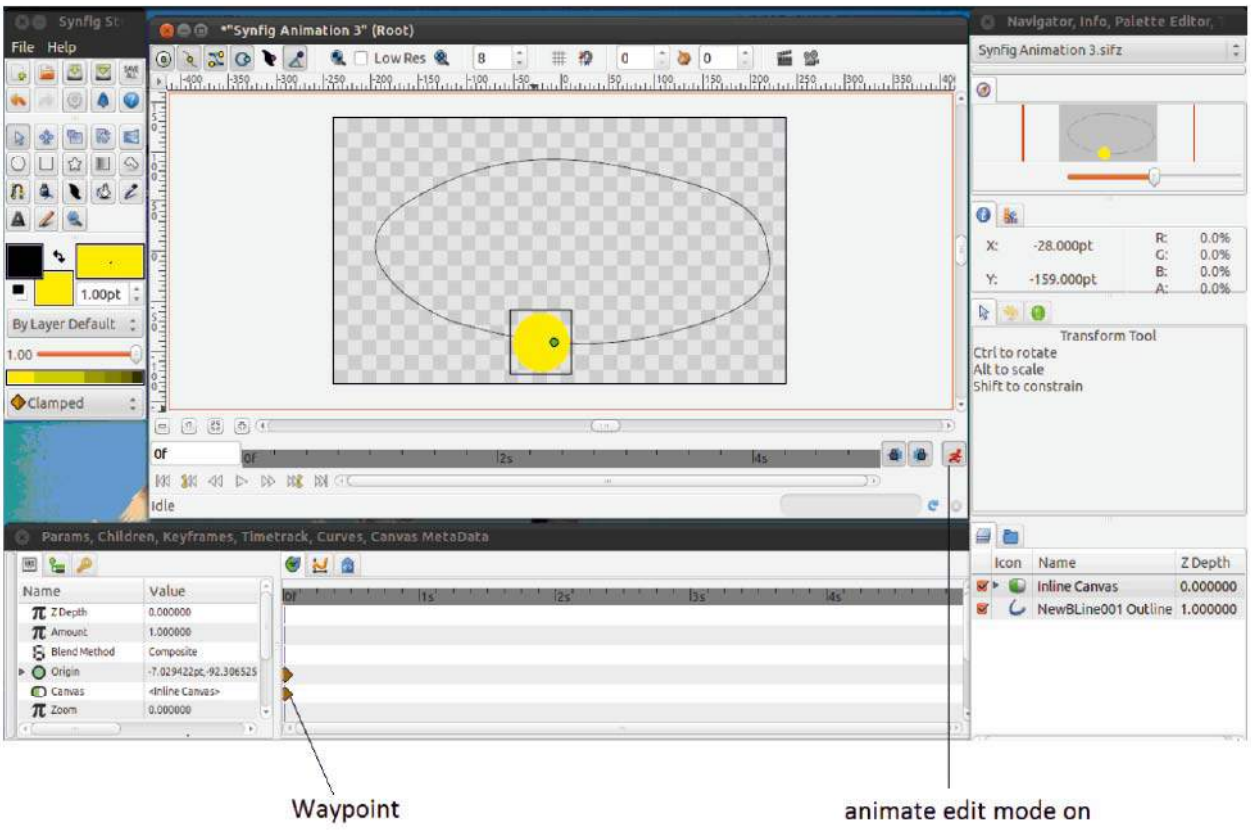


Figure 4.18 : Recording the position at 0s in the timeline

- Now click on 5s mark in the timeline and drag the circle in anticlockwise direction using the positioning duck (green duck) till you reach the ending point of the bline.

Figure 4.19 (a), (b) (c), (d), (e) and (f) shows the circle dragged from starting point to ending point. The timeline in each image shows the position of the circle at that particular time.

Note : Do not drag the circle directly to the end point by moving in clockwise direction. This will not record properly the position of the circle at different timing locations.

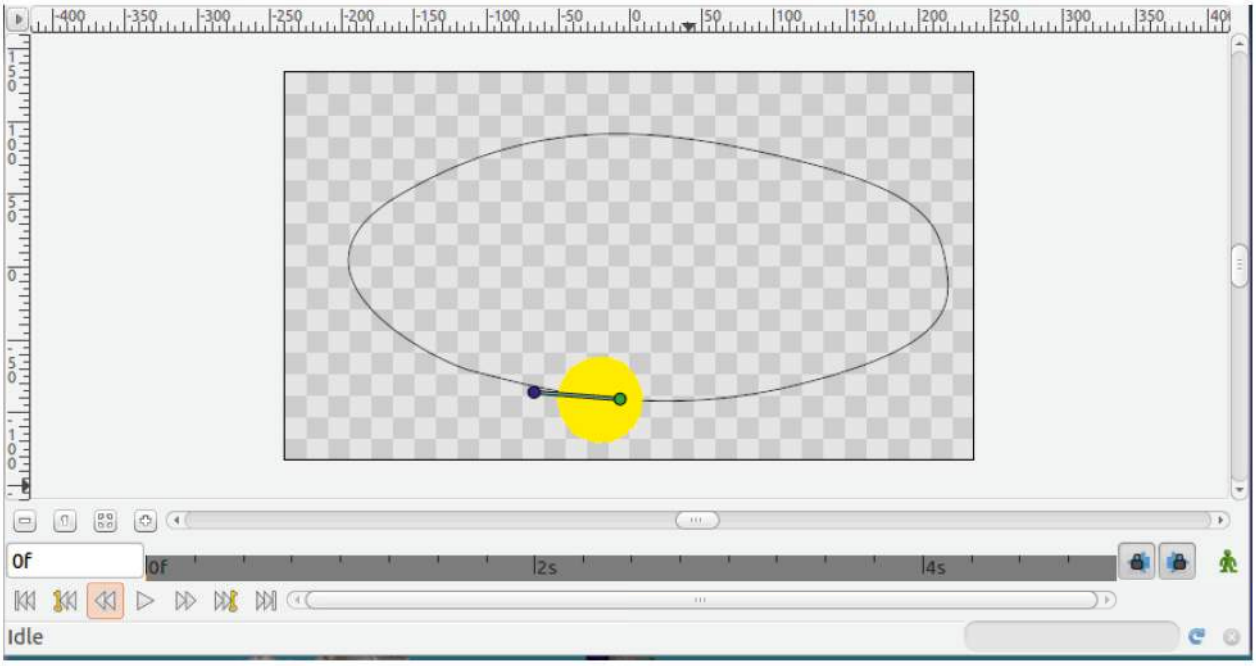


Figure 4.19(a) : Position of circle at 0f

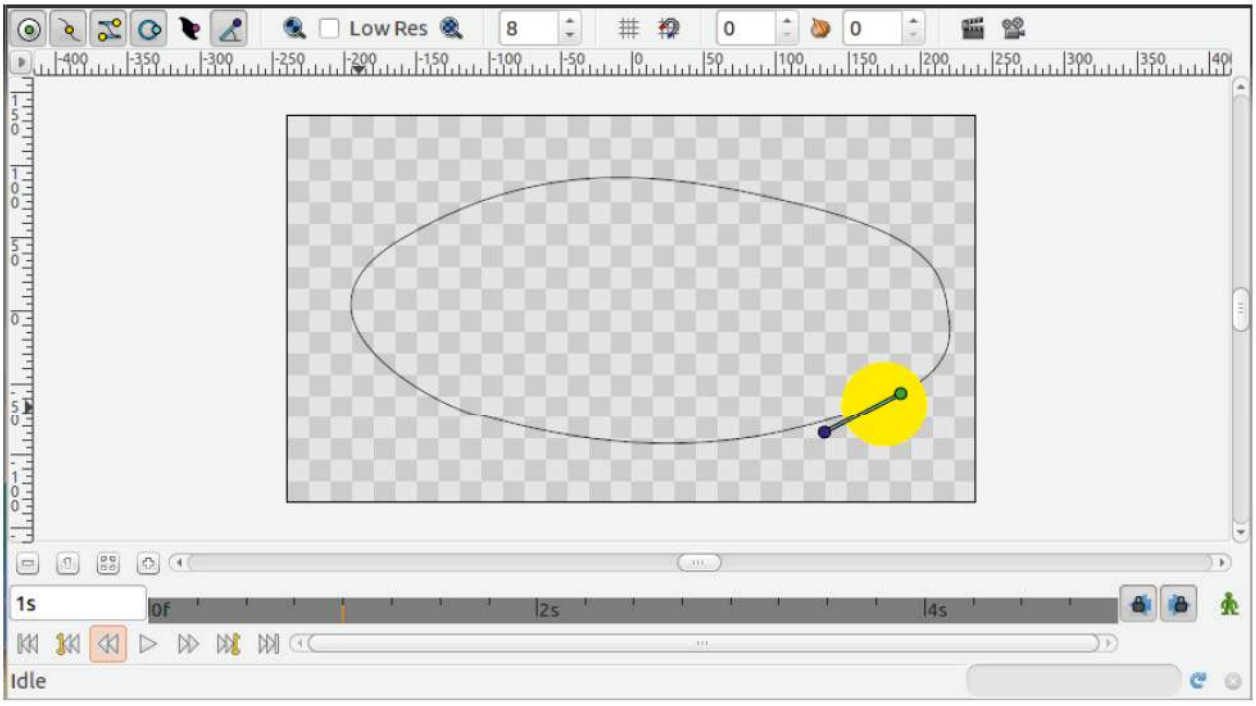


Figure 4.19(b) : Position of circle at 1s

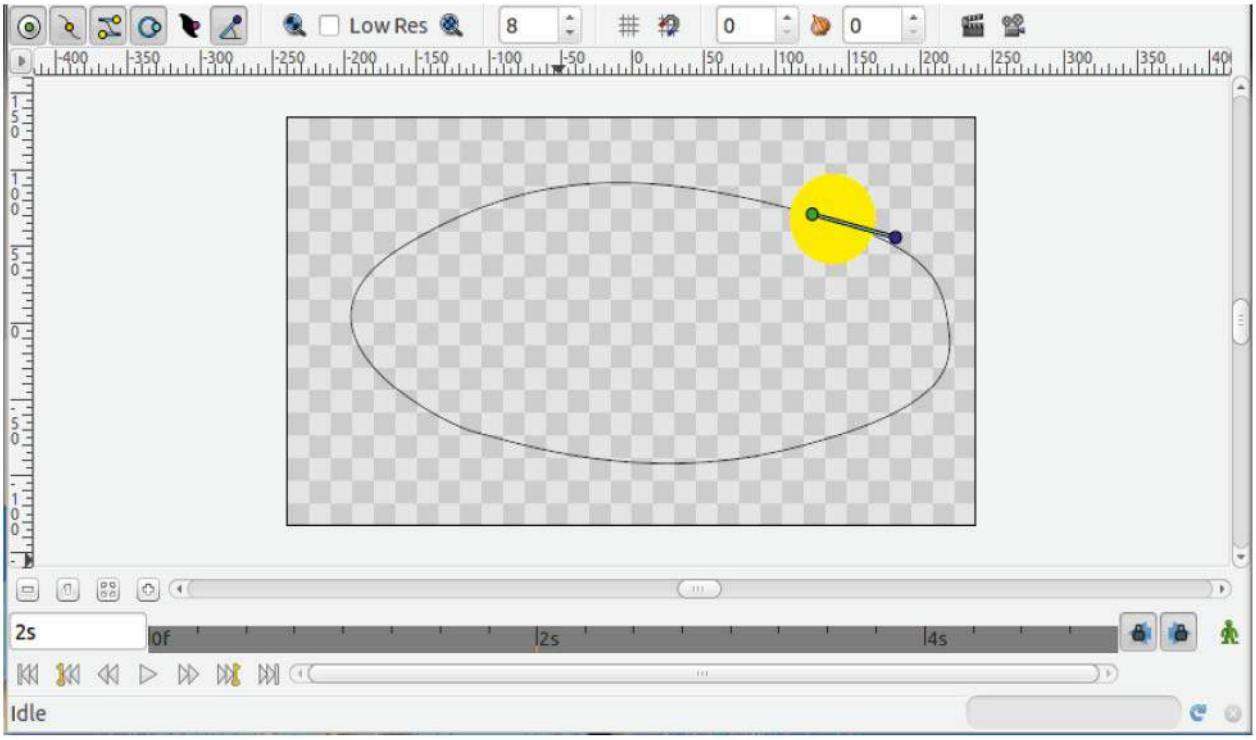


Figure 4.19(c) : Position of circle at 2s

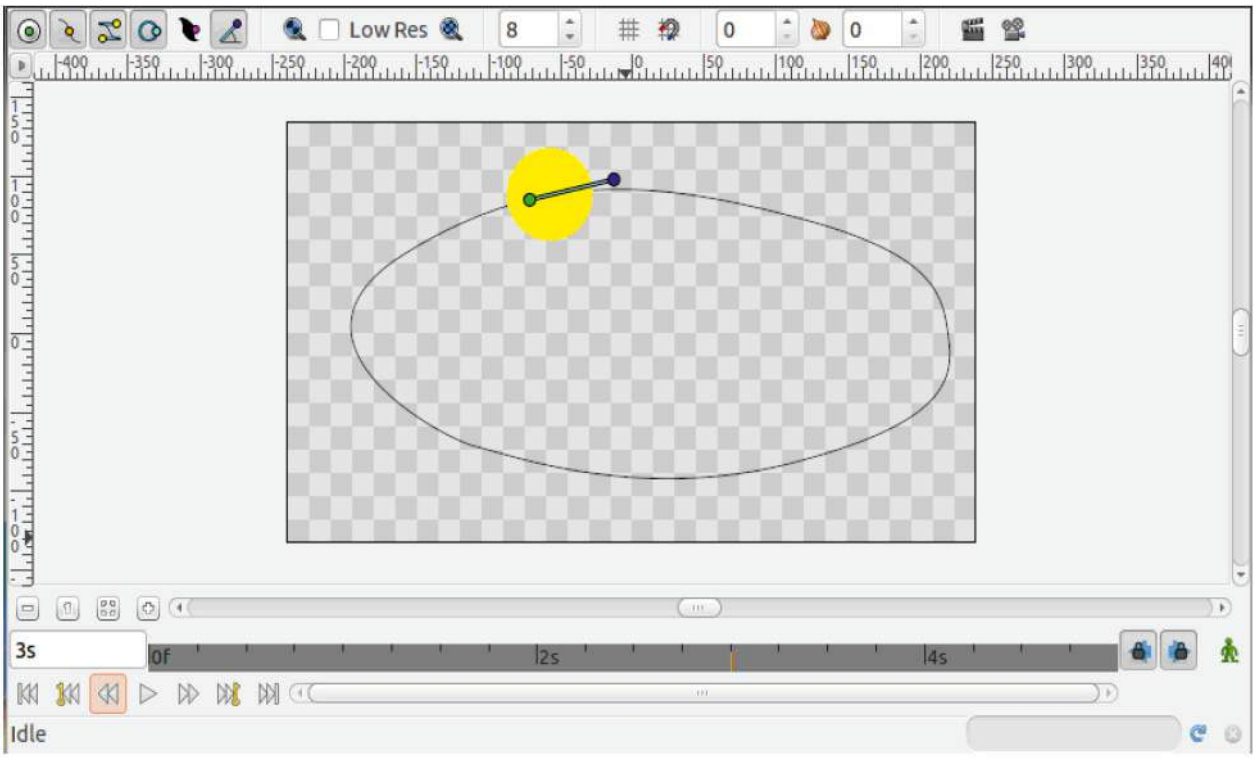


Figure 4.19(d) : Position of circle at 3s

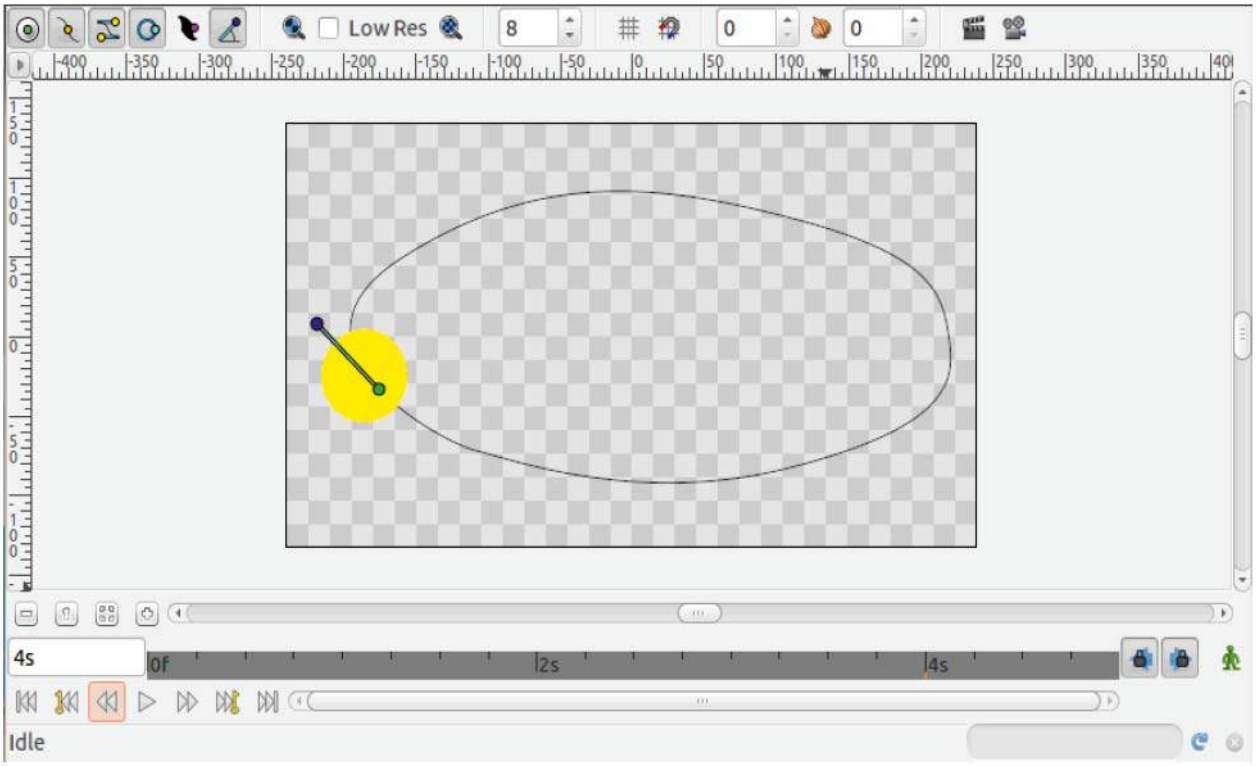


Figure 4.19(e) : Position of circle at 4s

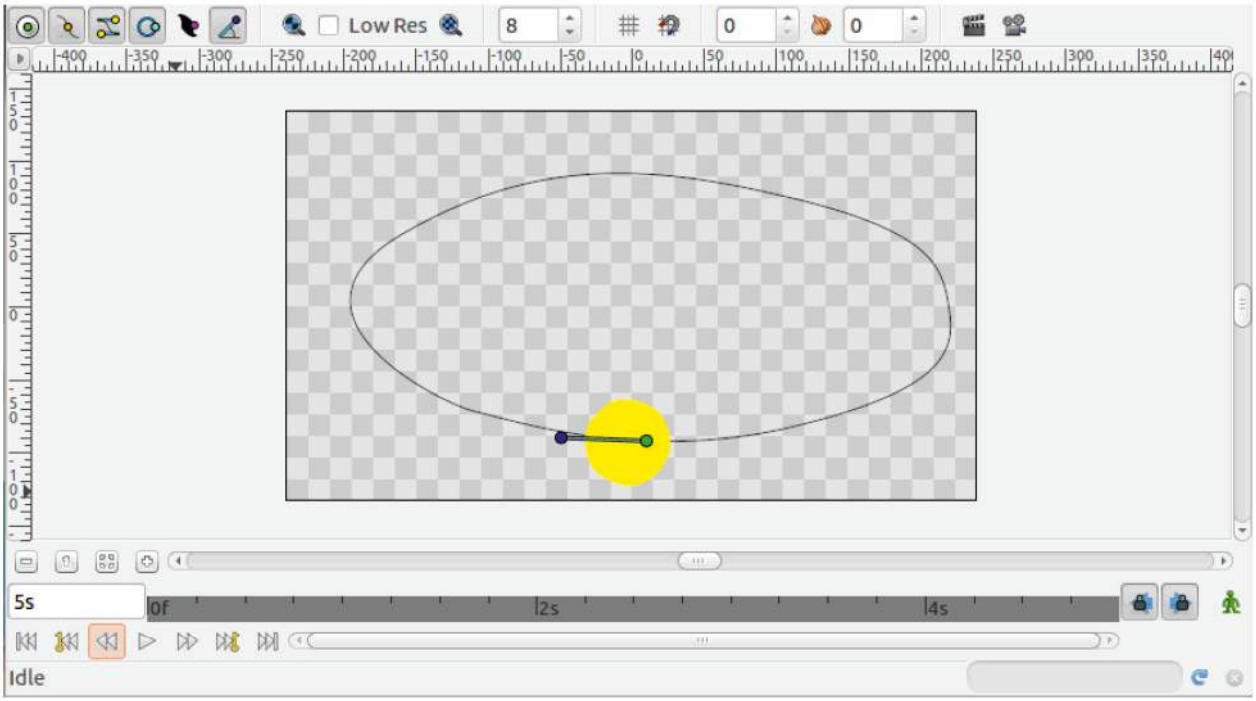


Figure 4.19(f) : Position of circle at 5s

- Click on play to preview the animation. You can see the circle moving in anticlockwise direction.
- Save and render the file. Figure 4.20 show the output as viewed using a web browser.

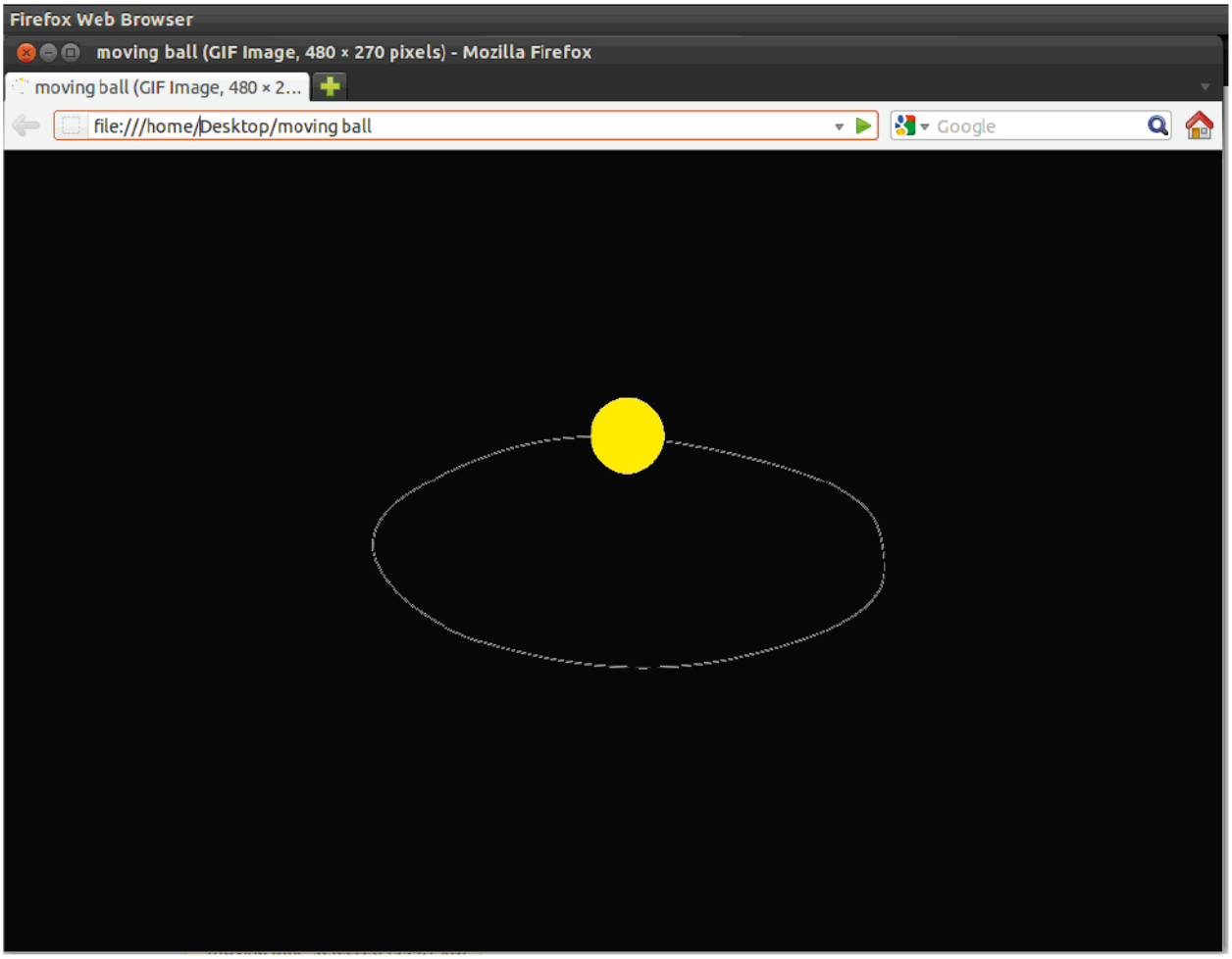


Figure 4.20 : Output displayed in web browser

Summary

In this chapter we introduced the concept of layers and how we can place our objects above or below another objects. Layers give us the freedom to work with each object individually. We introduced an important concept called "Encapsulation". Using encapsulation, we can give effects to various objects without changing other objects in the animation. We also discussed how to animate objects on a user defined path.

EXERCISE

1. Explain Layers. Give an example.
2. What is Encapsulation? Why do we need it?
3. Imagine a street view with objects such as building, car, sun, traffic light and person. State which layer will be above and below?
4. What is the purpose of Link to Bline option?

5. Choose the most appropriate option from those given below :

(1) Which one of the following is a type of layer?

(a) Timebar (b) Gradient

(c) Palette (d) Animate edit mode

(2) Which of the below have their own layer?

(a) object, element, effect (b) parameter, object, panel

(c) time, parameter, layer (d) effect, object, panel

(3) Which of the following panel shows the parameters of a layer?

(a) layer (b) keyframe

(c) params (d) timetrack

(4) Which of the following is the default setting for blend method?

(a) composite (b) alpha over

(c) straight (d) straight onto

(5) Using which of the following feature we can apply an effect to one layer without applying it to the layers beneath it?

(a) encapsulation (b) groups

(c) elements (d) parameters

(6) Each individual element of the canvas is broken down into which of the following components?

(a) frames (b) layers

(c) keyframes (d) panels

(7) Which of the following feature is used to attach a vertex of an object to the vertex of a Bline?

(a) link to object (b) link to line

(c) link to bline (d) link to object

(8) Which of the following represents a new layer created after encapsulation operation?

(a) inline layer (b) inline frame

(c) inline keyframe (d) inline canvas

(9) Each layer has its own set of which of the following components?

(a) frames (b) keyframes

(c) parameters (d) panels

(10) Which of the following terms represent grouping things together?

(a) cluster (b) inline

(c) parameters (d) encapsulation

(11) Which of the following terms represent the different levels on which we place our objects stacked above and below each other?

(a) layer (b) frame

(c) parameter (d) panels

(12) Gradient, blur, distortion and filter are types of which of the following components?

(a) frames (b) layer

(c) keyframes (d) panels

LABORATORY EXERCISES

1. Draw a rectangle, circle and square overlapping each other. Select different tool options and different color for each object. Also give a background color to the canvas.
2. Create a star and a circle layer. Apply gradient on the star layer. Create a square and apply blur effect on it. Now create a rectangle and also apply the blur effect. Note: the objects should overlap each other.
3. On the canvas using gradient show the sky and ground. Use different colored gradient for both. For sky use the shades of blue color as gradient and for ground use the shades of green color as gradient.
4. Draw two circles overlapping each other and apply blur effect to one of them.
5. Create an animation showing a star moving on the user defined path.
6. Create an animation showing a star rotating while moving from left to right.
7. Create an animation to show a kite flying in the sky.

5 Using Pictures in Synfig

Pictures when used in a presentation or animation improve the quality of reception. In Synfig, we can insert images to give a different look to our animation. We can also work with these images and adjust them as per our requirement just the same way as we work with the objects drawn using the tools. In this chapter, we will see how to work with pictures.

Inserting Image

Let us try to create a layer that contains image in it. To import an image onto a canvas perform the steps mentioned.

- Create a new file.
- Select File → Import. Alternatively, you can also press CTRL + i. This combination is the shortcut key to import image. This will open a dialog box showing the folders from where you want to import the image as shown in figure 5.1. Select the image that you would like to import and press open. The image will be imported on the canvas. Figure 5.2 shows the image imported to the canvas by us. Note that our screen may vary depending on the image that you choose to import.

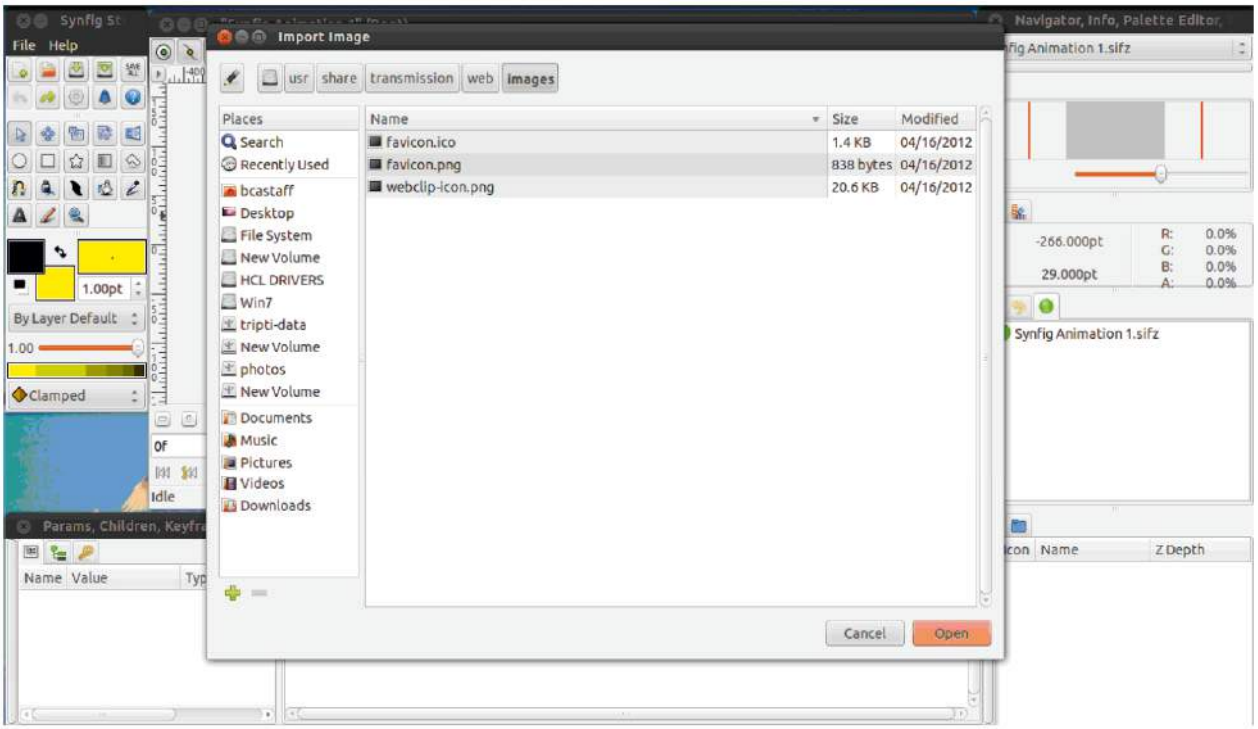


Figure 5.1 : Import image dialog box

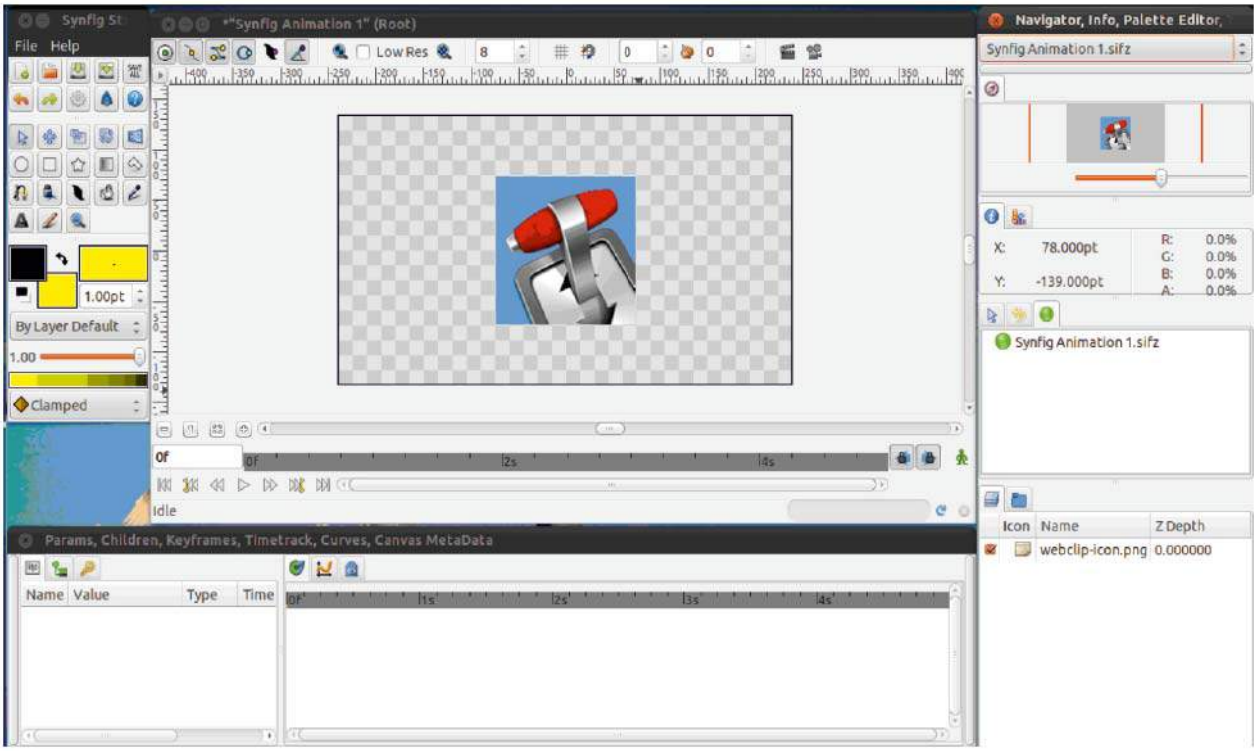


Figure 5.2 : Image inserted on the canvas

- To resize the imported image, select the image layer in the layers panel. As shown in figure 5.3 you can see two green points on the image. These can be now used to change the size of the image. Figure 5.4 shows the resized image when the green dots are moved.

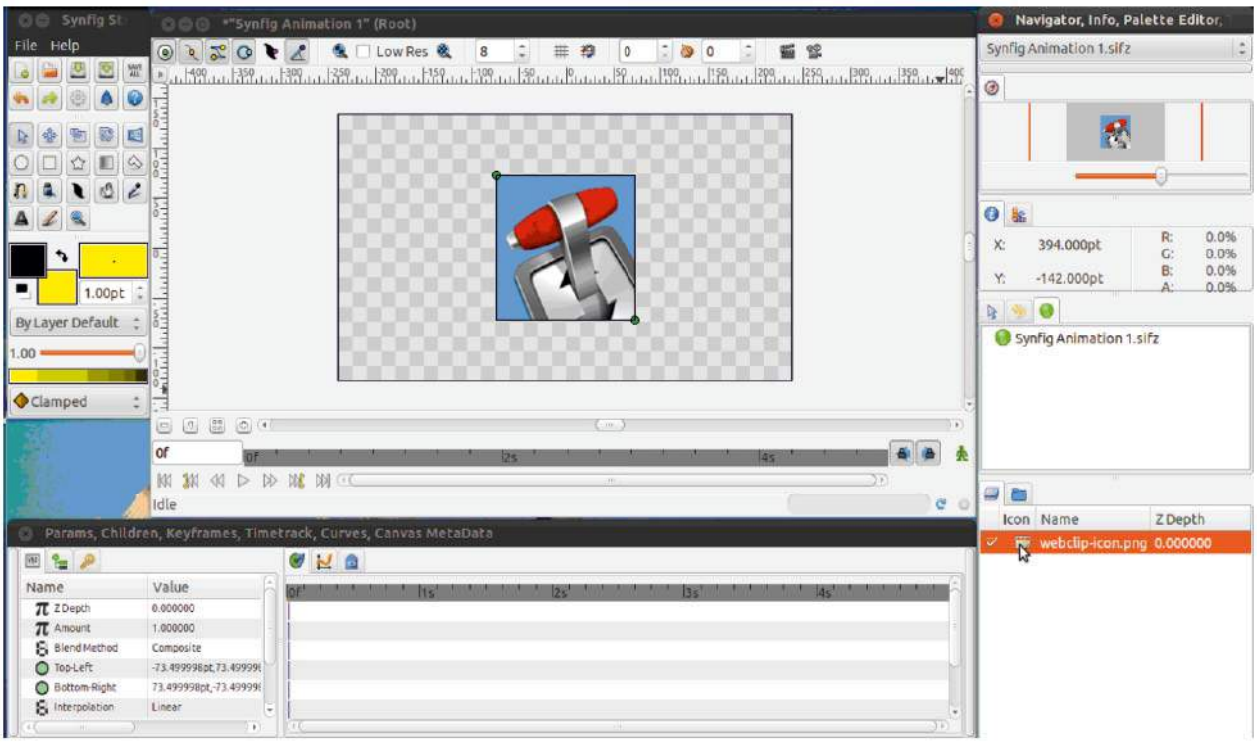


Figure 5.3 : Green points to resize the image

Figure 5.4 : Image resized using green points

- Observe that whenever we resize, the image tends to get distorted. This happens as we have not maintained the aspect ratio. If we want to resize the image and keep its aspect ratio we need to encapsulate the layer.

- Right click on the image layer and select encapsulate. This will add an inline canvas layer. Open up the encapsulated layer using the small triangle and you can see the image layer inside the inline canvas as shown in figure 5.5.

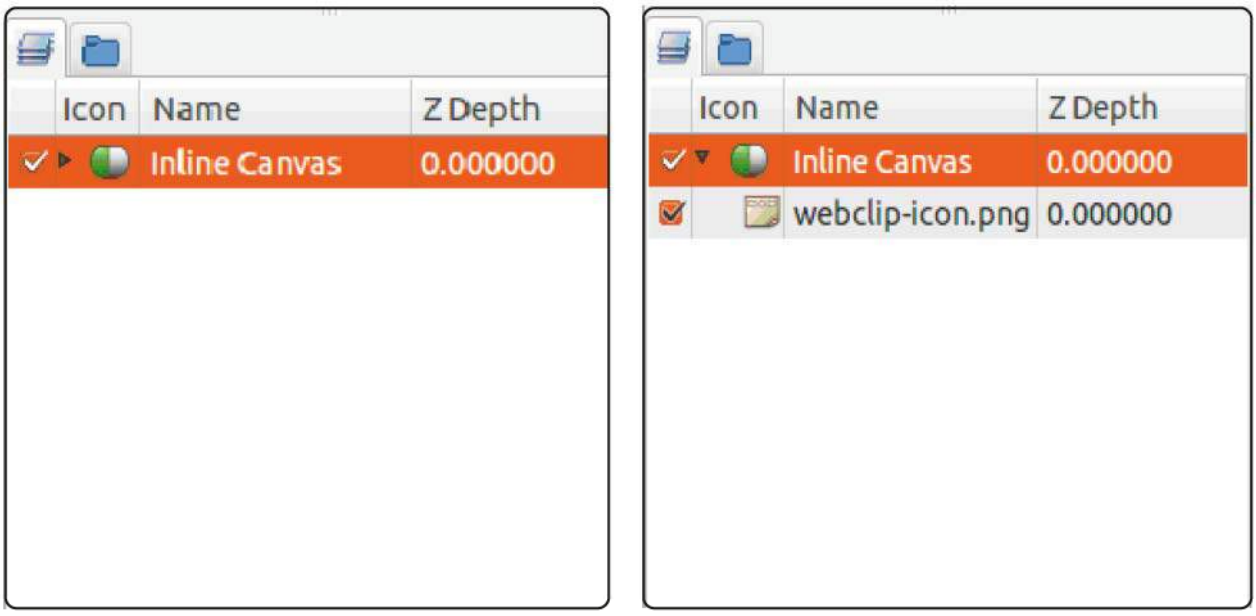


Figure 5.5 : Encapsulated image layer

- Now we need to add a new scale layer above the image layer. Select the image layer → Right click → New Layer → Transform → Scale. You can see a scale layer is added to top of the image layer as shown in figure 5.6.

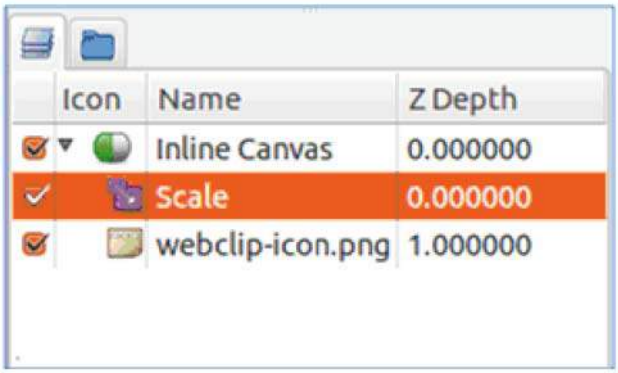


Figure 5.6 : Scale layer added above image layer

- Select the scale layer and in the parameters panel change the amount setting from 0 to -1 as shown in figure 5.7. In the canvas you see the change in the image size as shown in figure 5.8.

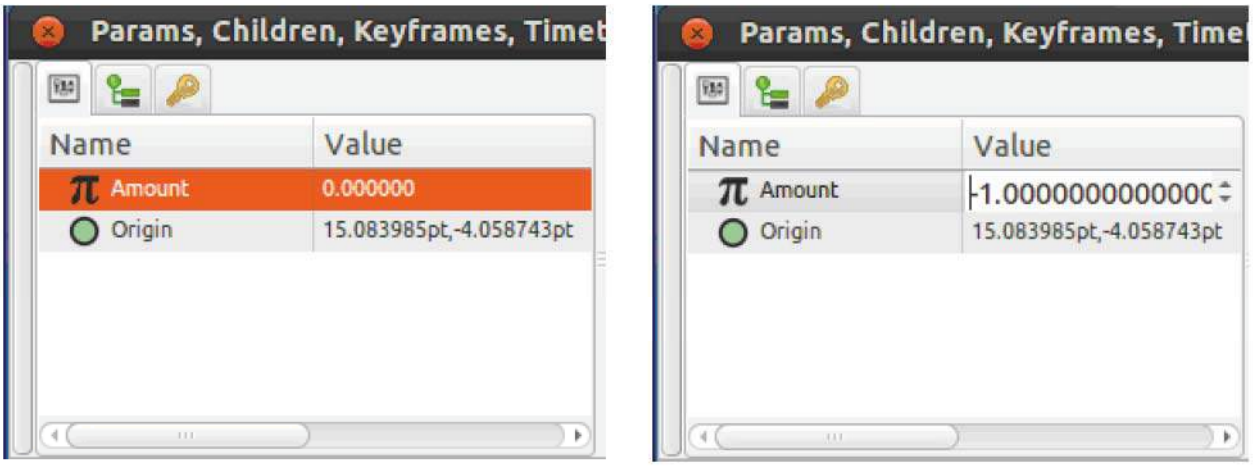


Figure 5.7 : Scale layer amount setting changed from 0 to -1

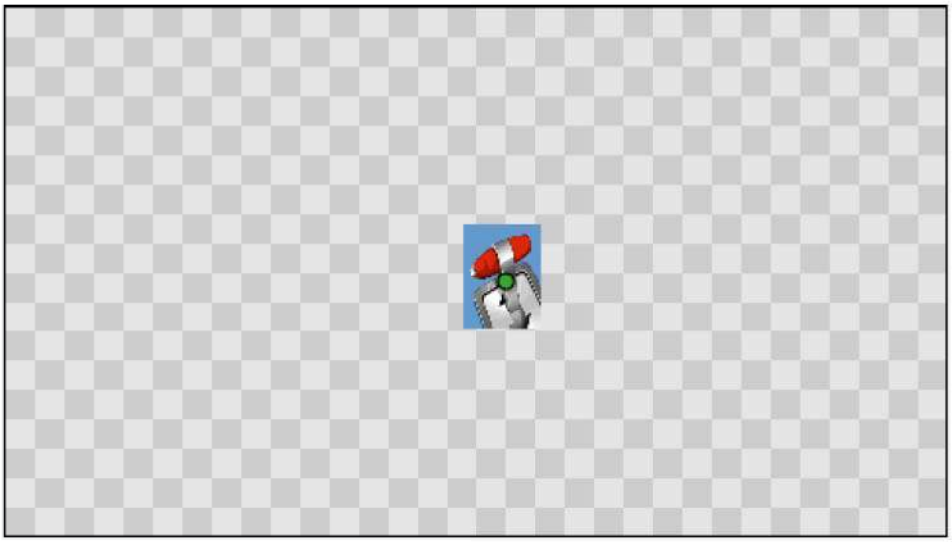


Figure 5.8 : Scaled image

- Now if you want to rotate the image then add a new rotate layer above the scale layer. Select the scale layer → Right click → New Layer → Transform → Rotate. You can see a rotate layer is added to top of the scale and image layer as shown in figure 5.9.

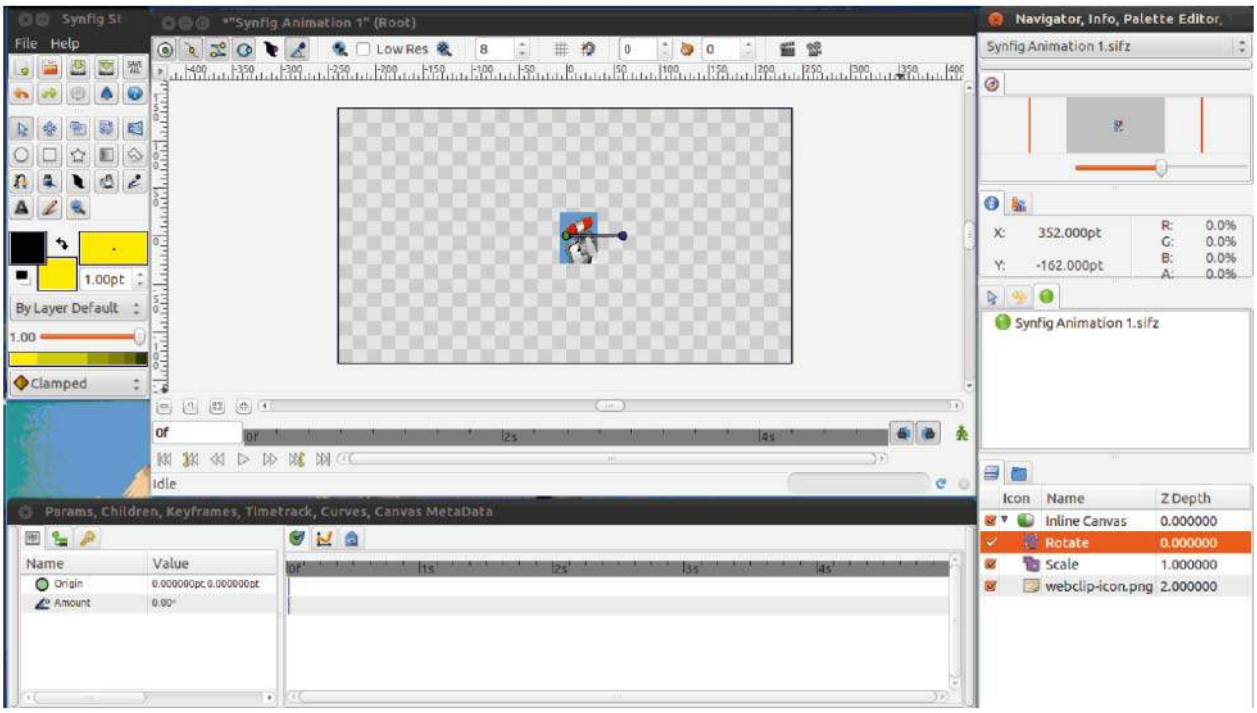


Figure 5.9 : Rotate layer added above scale layer

- By using the rotation duck (blue duck) you can rotate the image as you want. Figure 5.10 shows the rotated image.

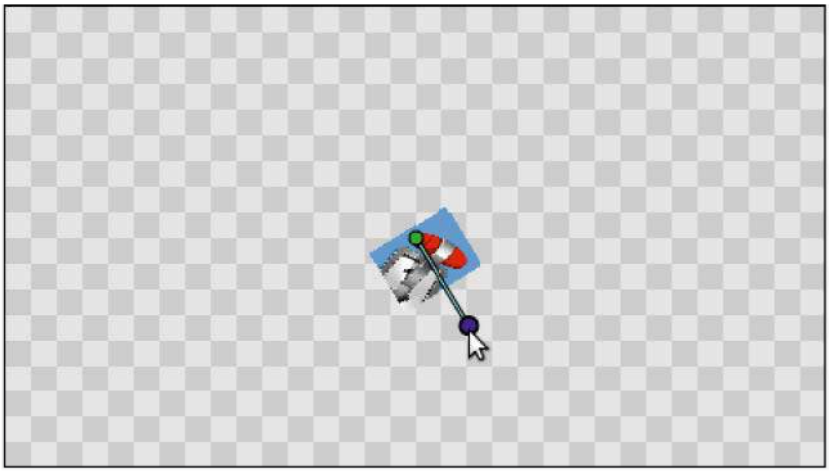


Figure 5.10 : Image rotated using the duck

Observe that once an image has been inserted we can scale, rotate and perform different operations on it. Images when used always enhance the animation.

Masking

Suppose, in your animation you want to show the movement of an object say clouds moving through the window. If the window is drawn using Synfig then you can place the object behind the window and can move the cloud object. But if the window is an image then it becomes difficult to animate the cloud.

Masking is a feature by which you can hide or reveal areas of a layer. Basically, the shape used as a mask acts like a window with the help of which you can see the objects beneath it. Alternatively, it can be used as patch to hide the objects beneath it.

Let us understand by an example and try to show a cloud moving through the window using the concept called masking.

- Create a new file.
- Now import an image on the canvas using CTRL + i. Select the picture. And resize it as per the requirement. Figure 5.11 shows the imported image. Note that your screen may look different based on the image you select. You can see the image layer added in the layers panel.

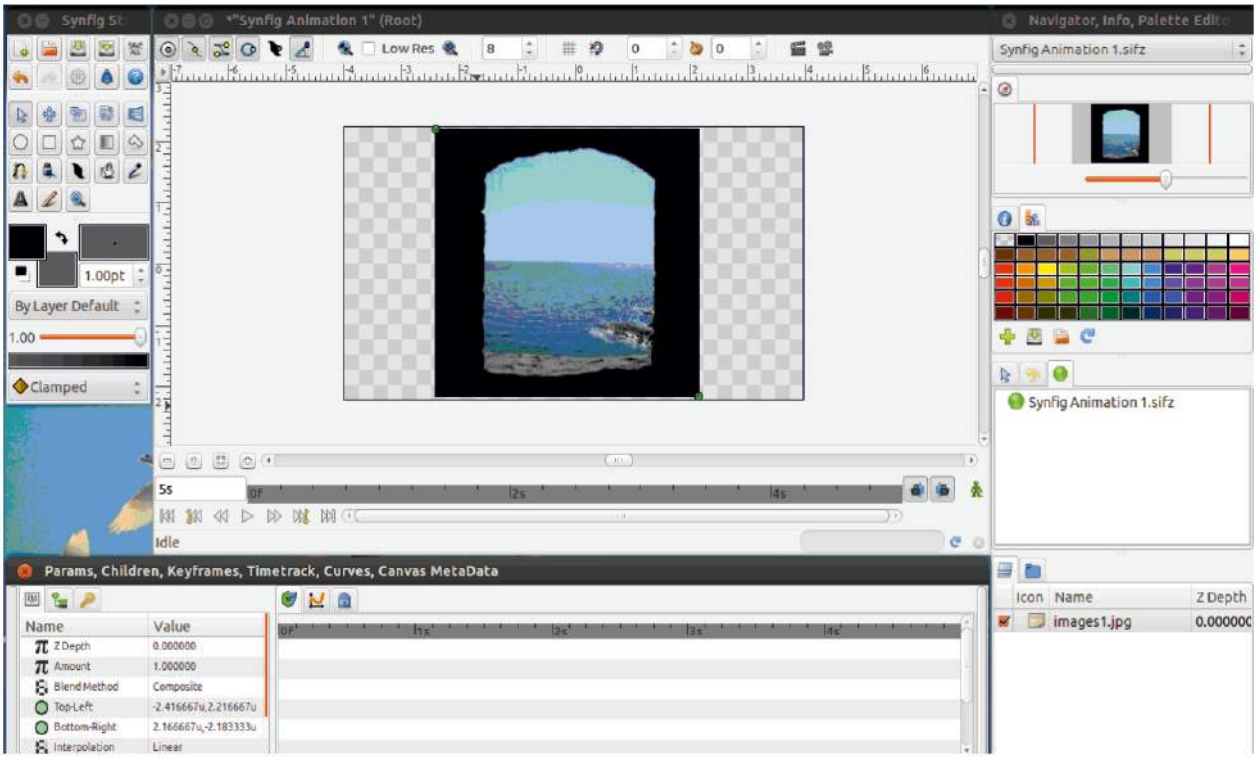


Figure 5.11 : Image imported on canvas

- Now using Bline tool draw a cloud on the left side of the canvas as shown in figure 5.12. You can see the Bline layer on top of the image layer.

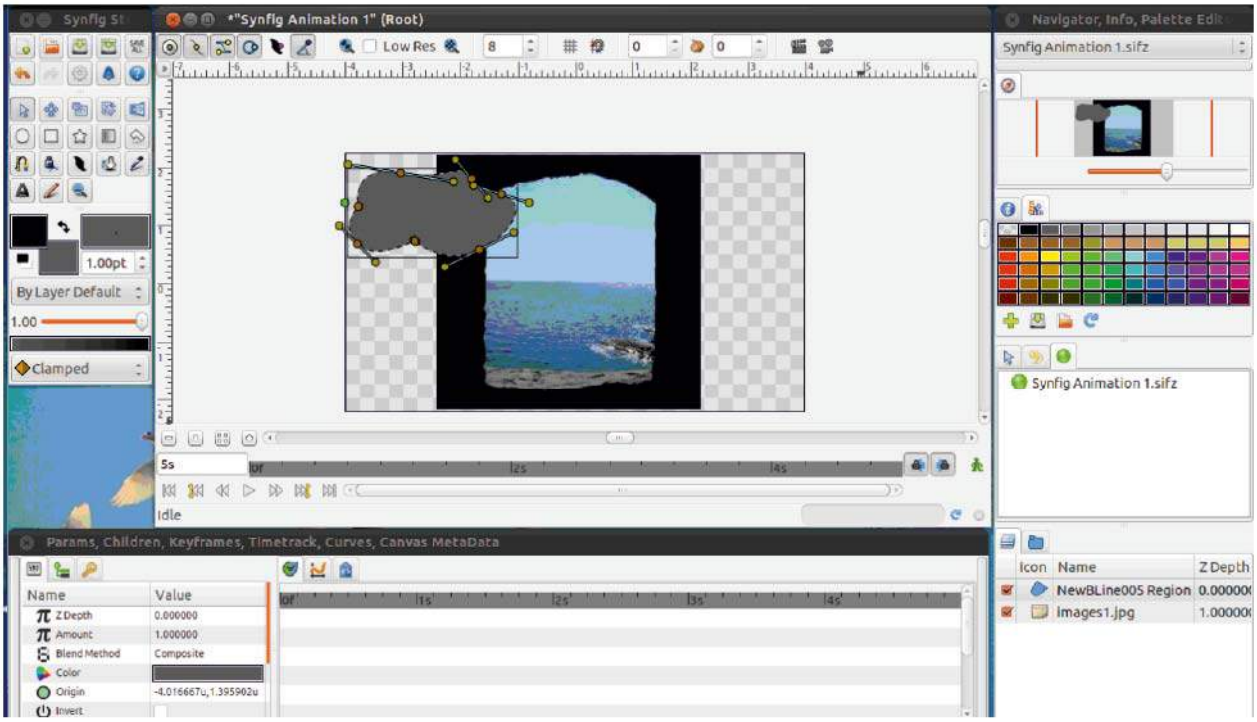


Figure 5.12 : Cloud drawn using Bline tool

- Now turn on the animate edit mode.
- We will create a small animation of cloud moving from left to right of the canvas. Move the cloud a little bit to record the position at 0f in the timeline. Now click on 5s mark on the timeline and drag the cloud towards the right. Figure 5.13 (a) and (b) shows the status of animation at "0f" and "5s" respectively.

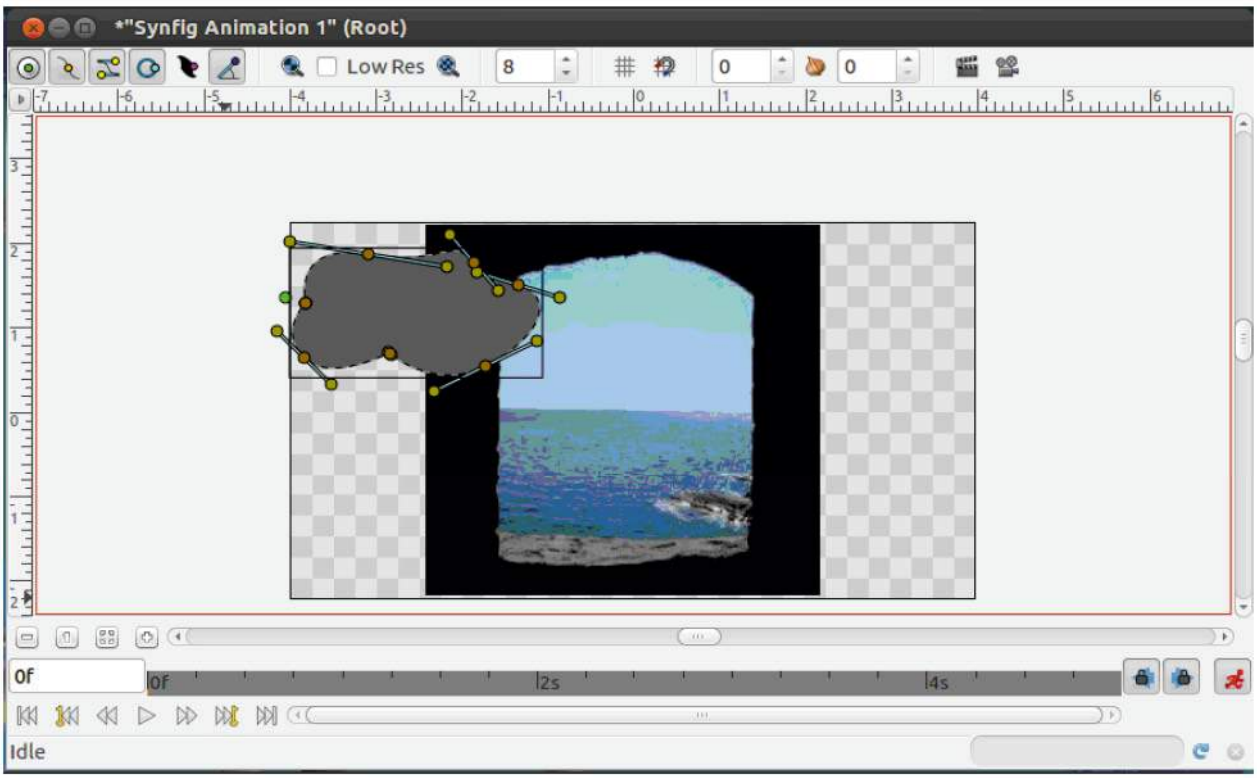


Figure 5.13(a) : Animation of cloud at 0f

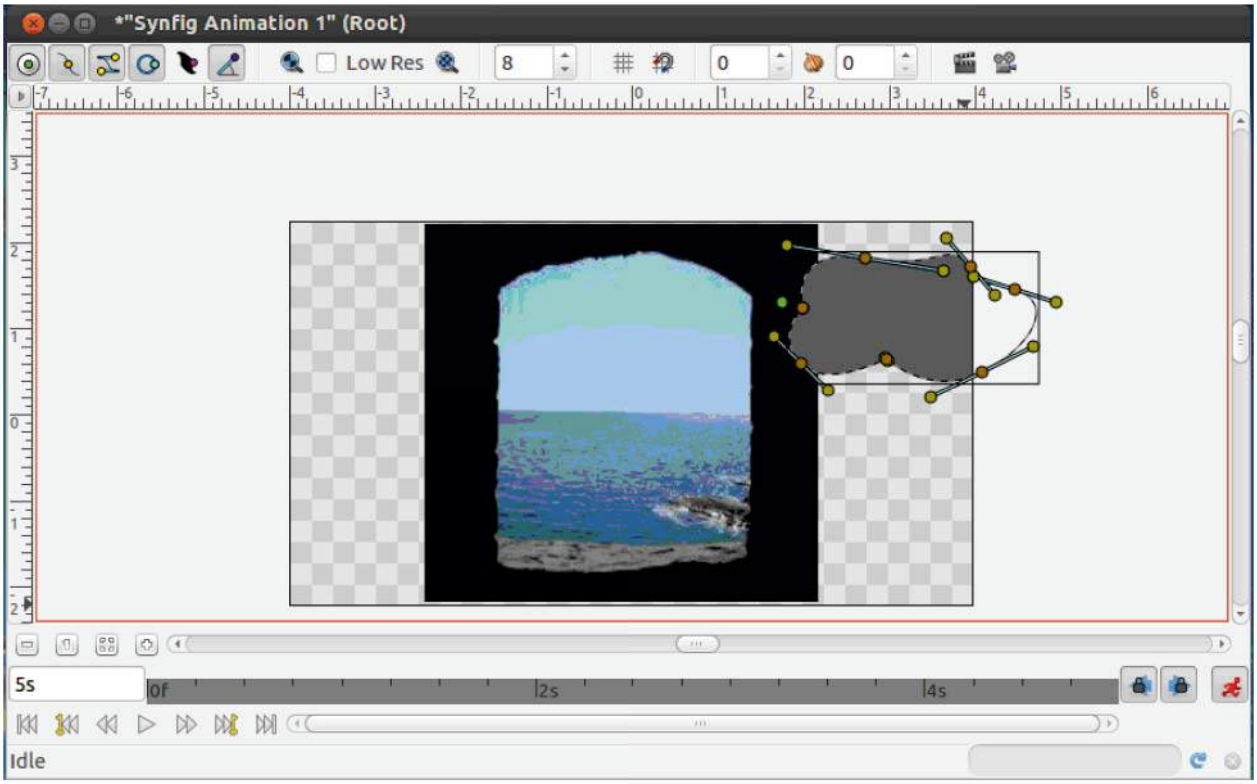


Figure 5.13(b) : Animation of cloud at 5s

- Turn off the animate mode and click on the play button to see the animation.
- You can see the cloud moving in front of image, as the cloud layer (or NewBline005 layer) is above the image layer shown in figure 5.14(a).
- To change the animation, in the layers panel, drag the cloud layer below the image layer as shown in figure 5.14(b). Click on the play button to see the animation. You can see the difference in the animation. Now the cloud is moving from behind the image.

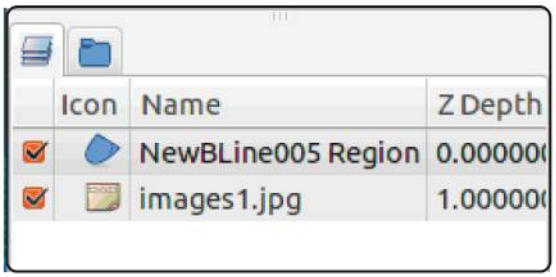


Figure 5.14(a) : Actual position of layers

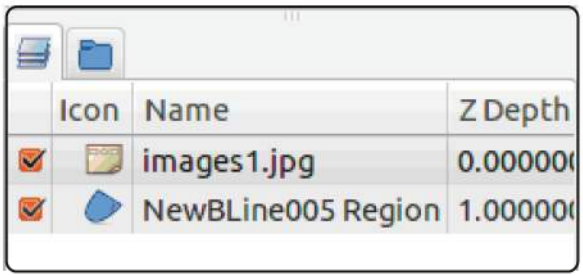


Figure 5.14(b) : Modified position of layers

But we want the cloud to move through the window image only. We will do this by using masking. Masking is used to hide or reveal an object.

Let us first use masking for hiding the objects. Here we want the cloud object to be hidden at both the ends of the image.

- First, we need to create a mask shape to hide the cloud on both the sides of the canvas. We will use the Bline tool to create the mask shape. Select the Bline tool and from the tool options select only 'Create Region Bline' option. Draw a rectangular shaped Bline as shown in figure 5.15. To end the Bline, at the last vertex right click and select 'Loop Bline'.

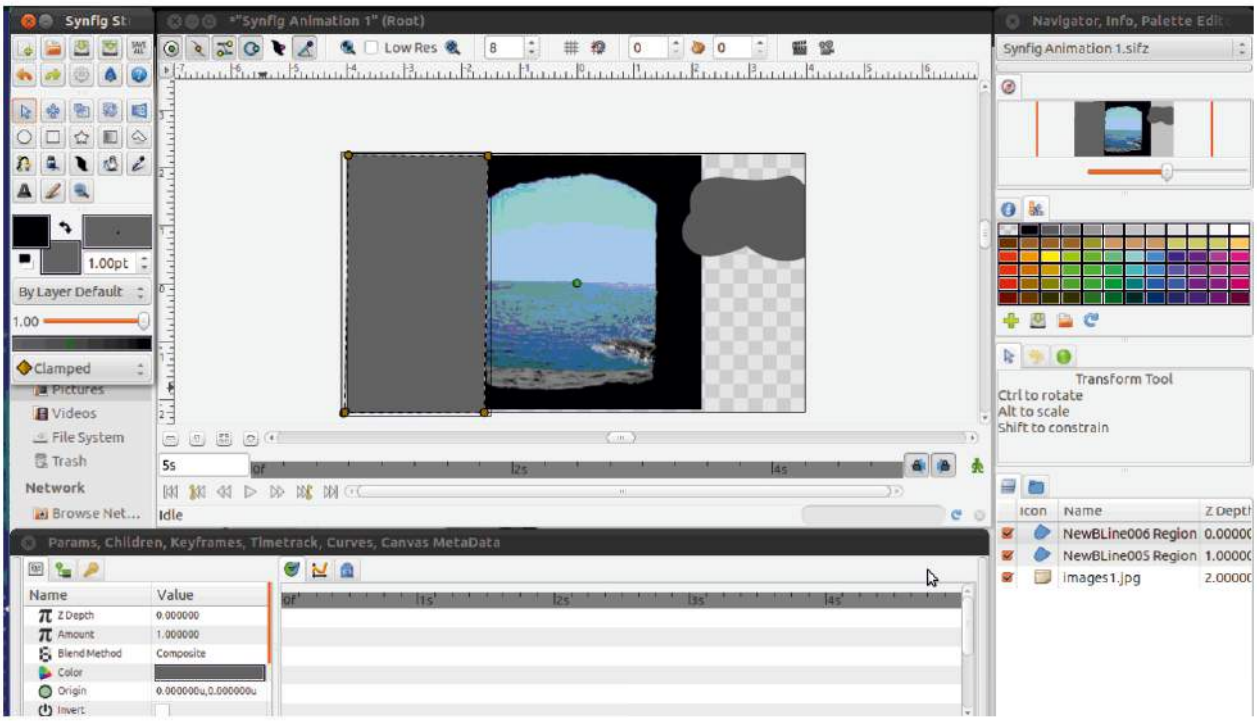


Figure 5.15 : Mask shape created using Bline tool

- Place the mask shape above the elements you want to mask. In our example we will place the mask shape above the cloud layer. Also, as we want to apply mask only on the cloud layer, we will encapsulate the mask shape and the cloud layer. Figure 5.16 shows the encapsulated layer 'inline canvas' and also shows the contents of inline canvas layer. We have renamed the layers NewBline005 as cloud layer and NewBline006 as mask layer. Click on the layer name which will allow you to change the name.

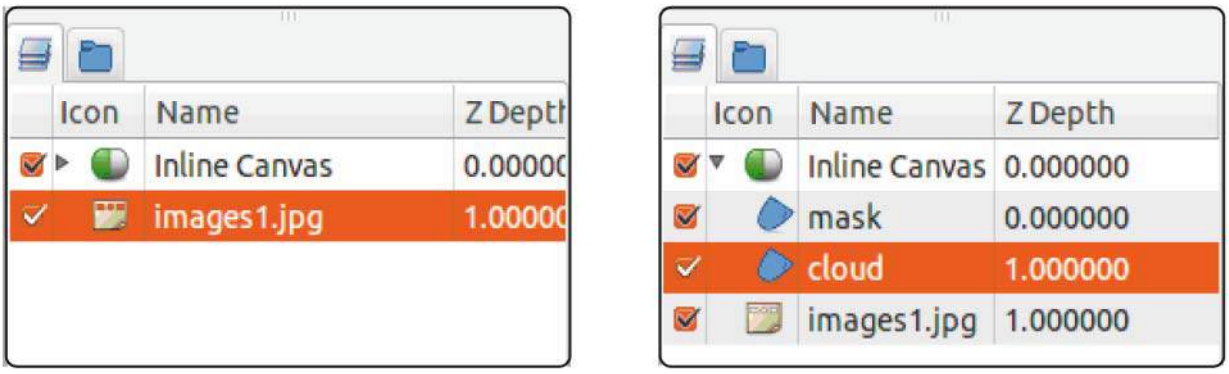


Figure 5.16 : Mask shape and cloud layer encapsulated

- Select the mask layer from the layers panel and in the parameters panel set the blend method to 'Alpha over'. Figure 5.17 shows the blend method changed to 'Alpha Over'.

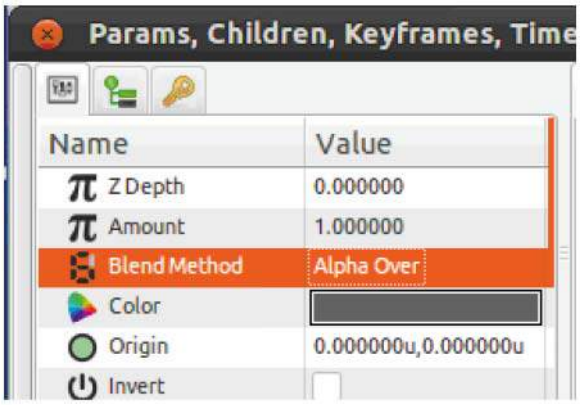


Figure 5.17 : Change the Blend method to 'Alpha over'

- Play the animation and you will see that the cloud is now not visible on the left side of the canvas.

- We need to do the same thing on the right side of the canvas also. So draw a rectangular mask shape using Bline tool on the right side of the canvas as we did earlier on the left side. Figure 5.18 shows the mask shape on the right side of the canvas.

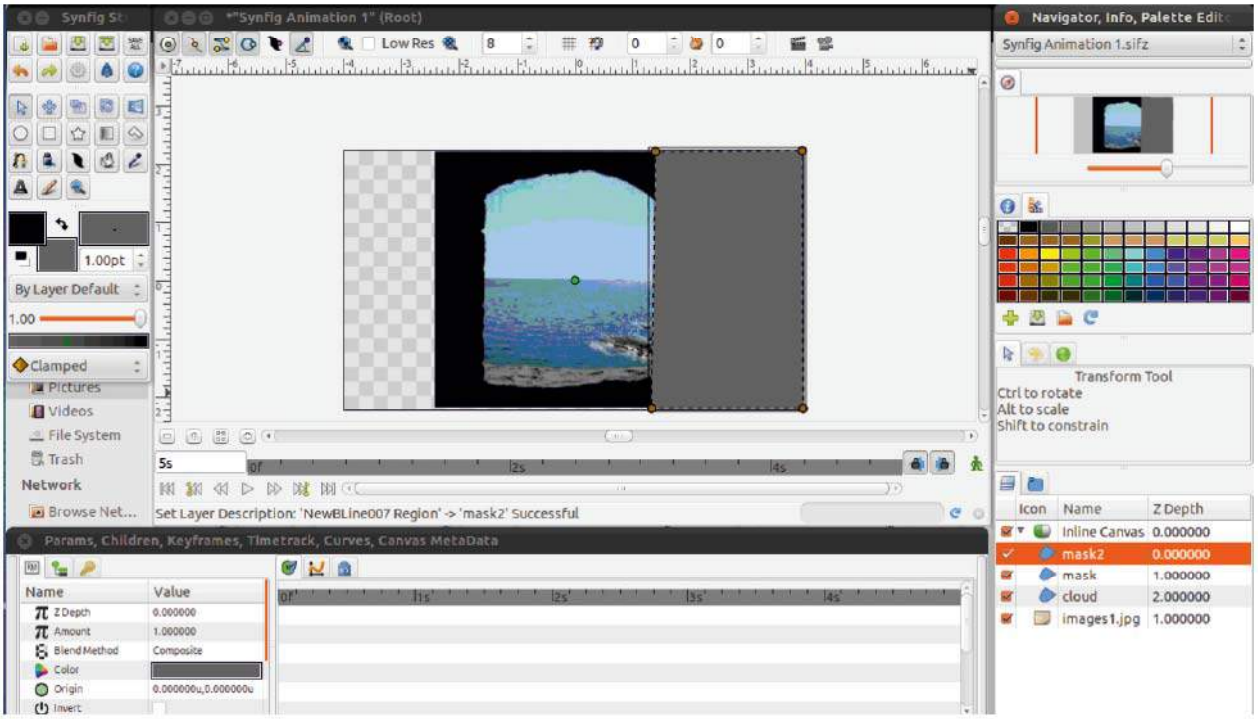


Figure 5.18 : Mask shape created on the right side of the canvas

- Rename the layer as mask2 and drag the mask2 shape layer into the encapsulated layer above the mask layer as shown in figure 5.19.

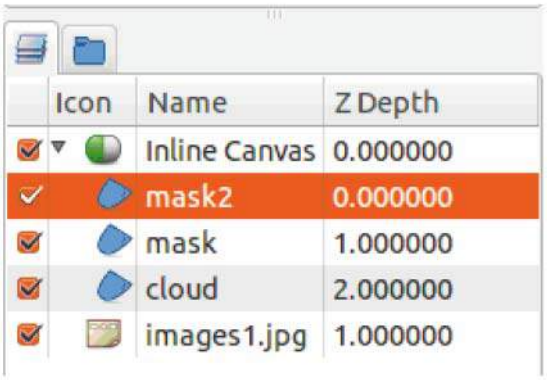


Figure 5.19 : Repositioning mask2 layer

- Select the mask2 layer from the layers panel and in the parameters panel set its blend method to 'Alpha over'.
- Play the animation and you can see that now the cloud moves through the window image.
- Save and render the file. Figure 5.20 shows the output on web browser.

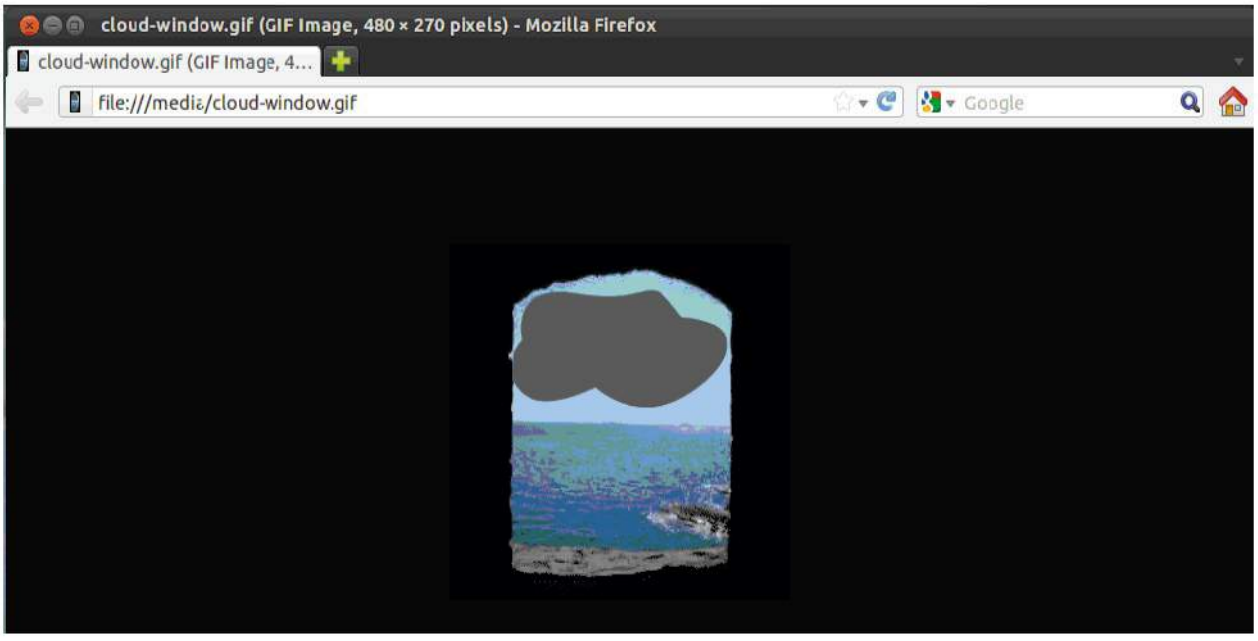


Figure 5.20 : Output displayed in web browser

Thus, using masking we can hide the objects. Masking can also be used to reveal the objects. Let us say, in the image we want to show the sun. Only some part of the sun is visible in the image. Here we use masking to reveal the objects.

- Create a new file.
- Import an image on the canvas using CTRL + i. Select the picture. Figure 5.21 shows the imported image and the image layer added in the layers panel.

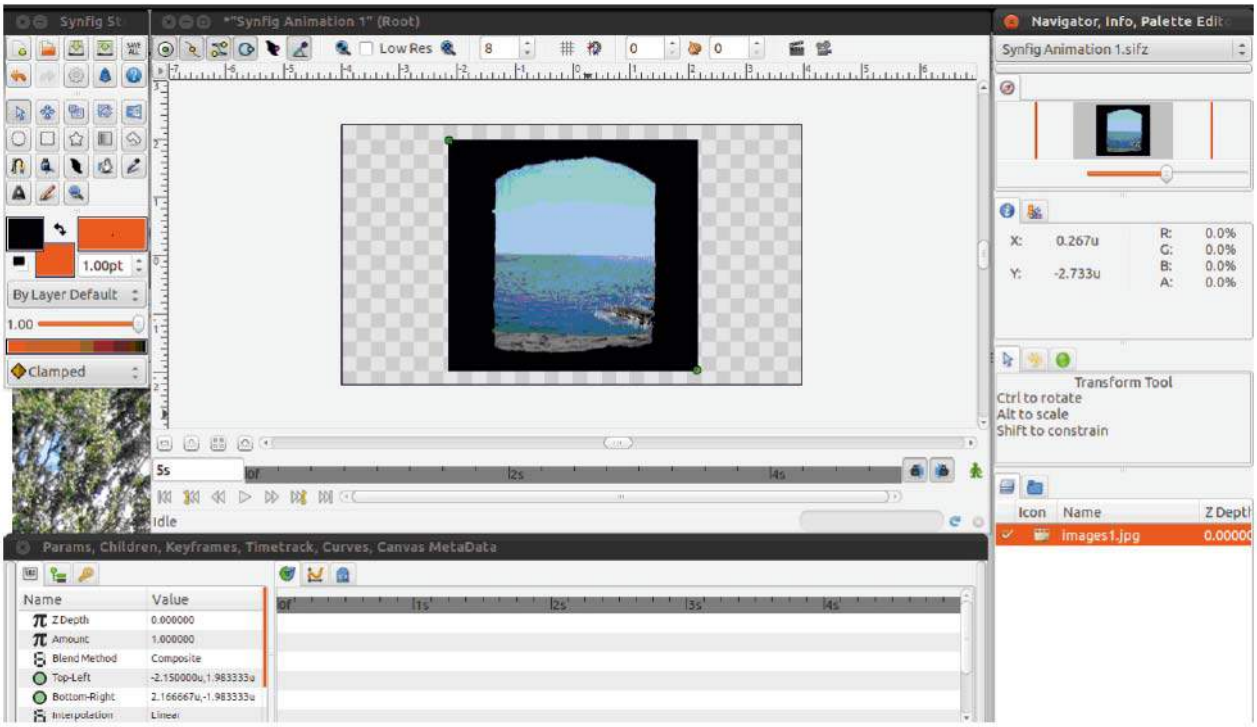


Figure 5.21 : Import the image

- Select circle tool and the color from the palette. Draw an orange coloured circle as shown in the figure 5.22. Rename the layer as 'sun'.

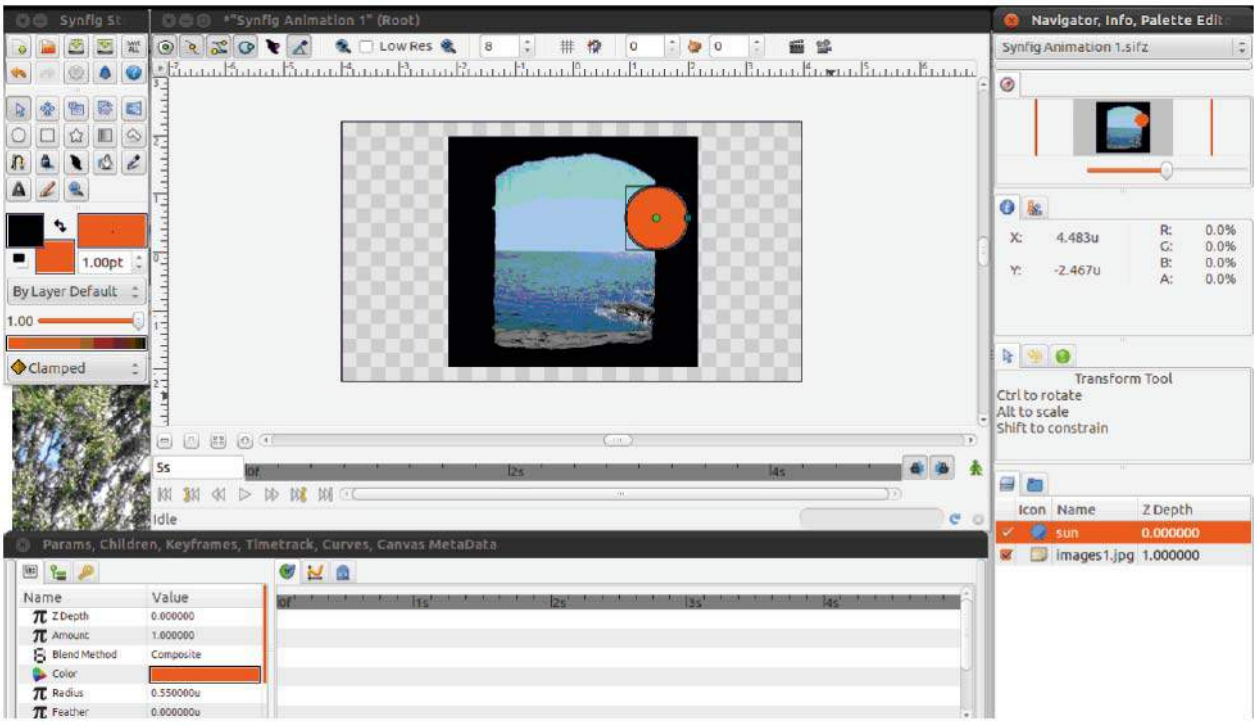


Figure 5.22 : Draw a circle

- Draw a mask shape using Bline tool. Here we need to draw the mask on the part of the sun that will be revealed. Figure 5.23 shows the mask. Rename the layer as 'mask' as shown in figure 5.24.

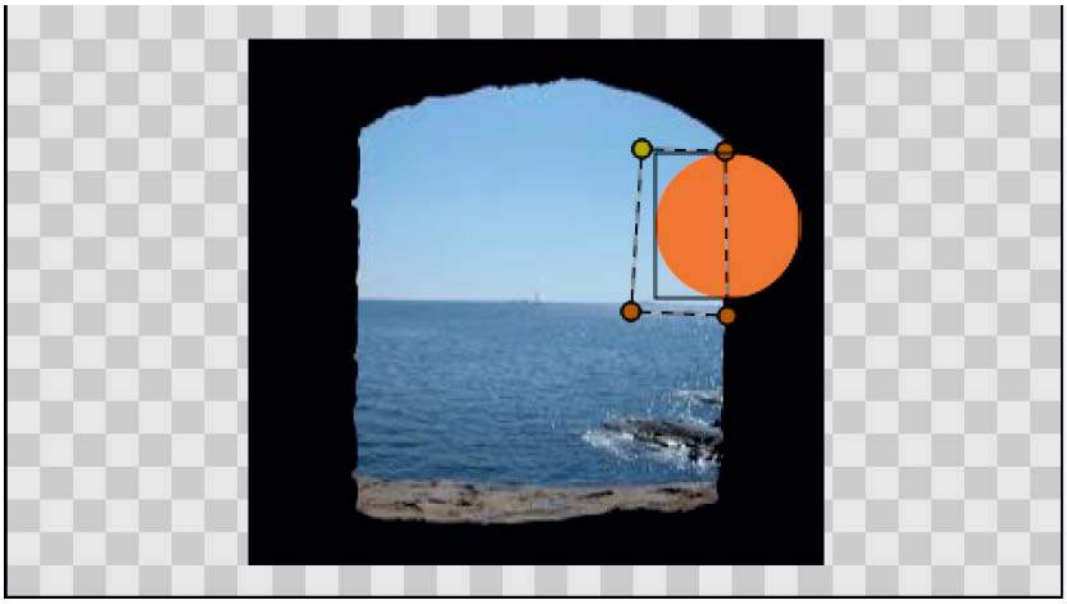


Figure 5.23 : Draw a mask shape

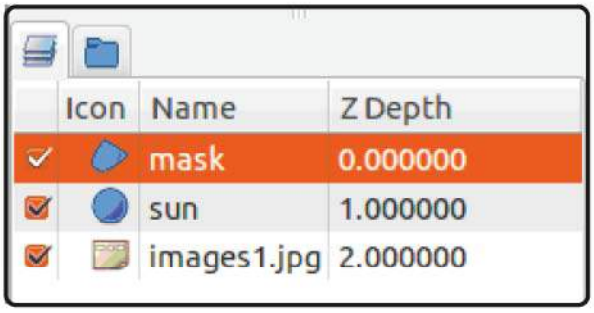


Figure 5.24 : Renaming the new Bline layer

- Select the mask layer and in the parameters panel select the 'invert' option as shown in figure 5.25. Figure 5.26 shows the effect on the canvas after selecting invert option.

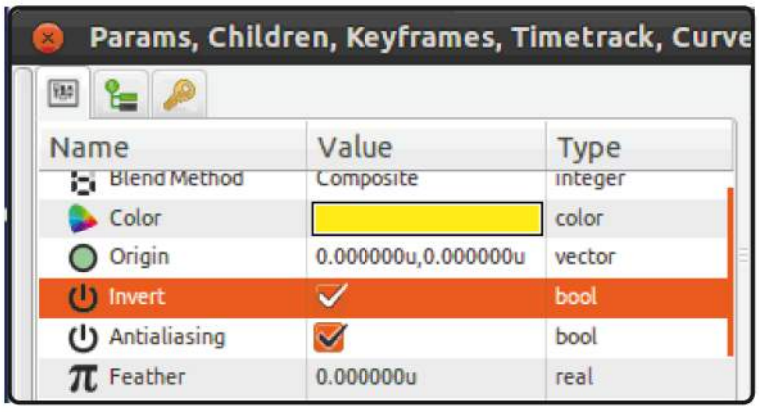


Figure 5.25 : Invert option selected in parameters panel

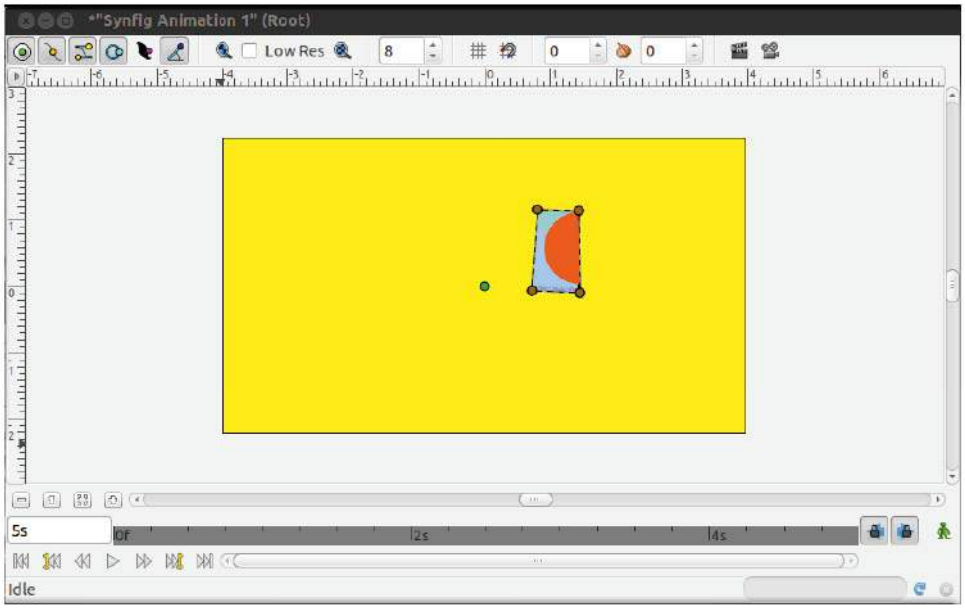


Figure 5.26 : Effect on the canvas after selecting invert option

- Change the blend method to 'Alpha over' as shown in the figure 5.27. The canvas will now appear as shown in figure 5.28.

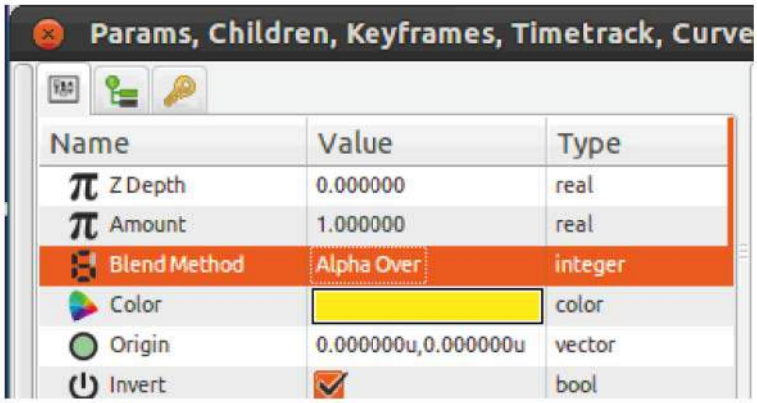


Figure 5.27 : Change the blend method to Alpha over

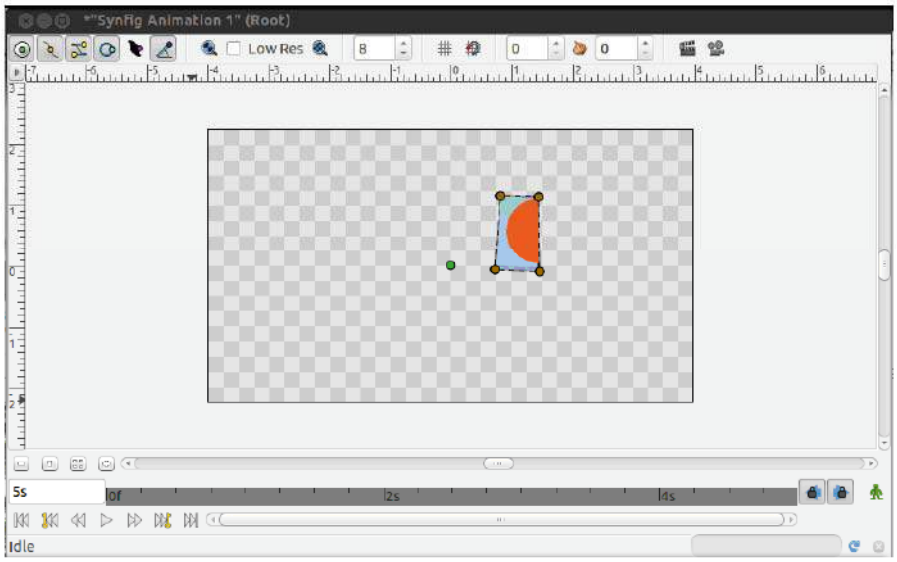


Figure 5.28 : Effect on the canvas after changing blend method

- As the image layer is placed below the mask layer, only the objects that are below the mask layer are visible. But we want the mask effect on only the sun layer and not the image. So we need to encapsulate both of them as shown in figure 5.29.

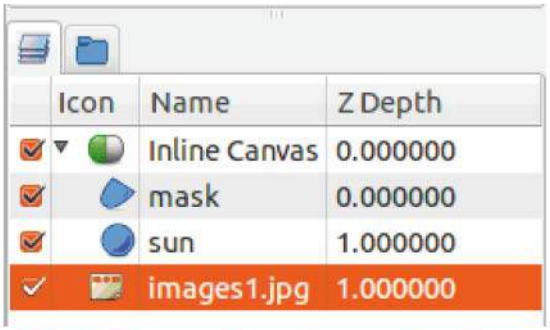


Figure 5.2 9: Encapsulate the mask and sun layers

- We can see the effect of masking in figure 5.30. We can put any number of objects below the mask layer which will be revealed. The objects on which we do not want to have masking effect should be placed out of the encapsulated layer.

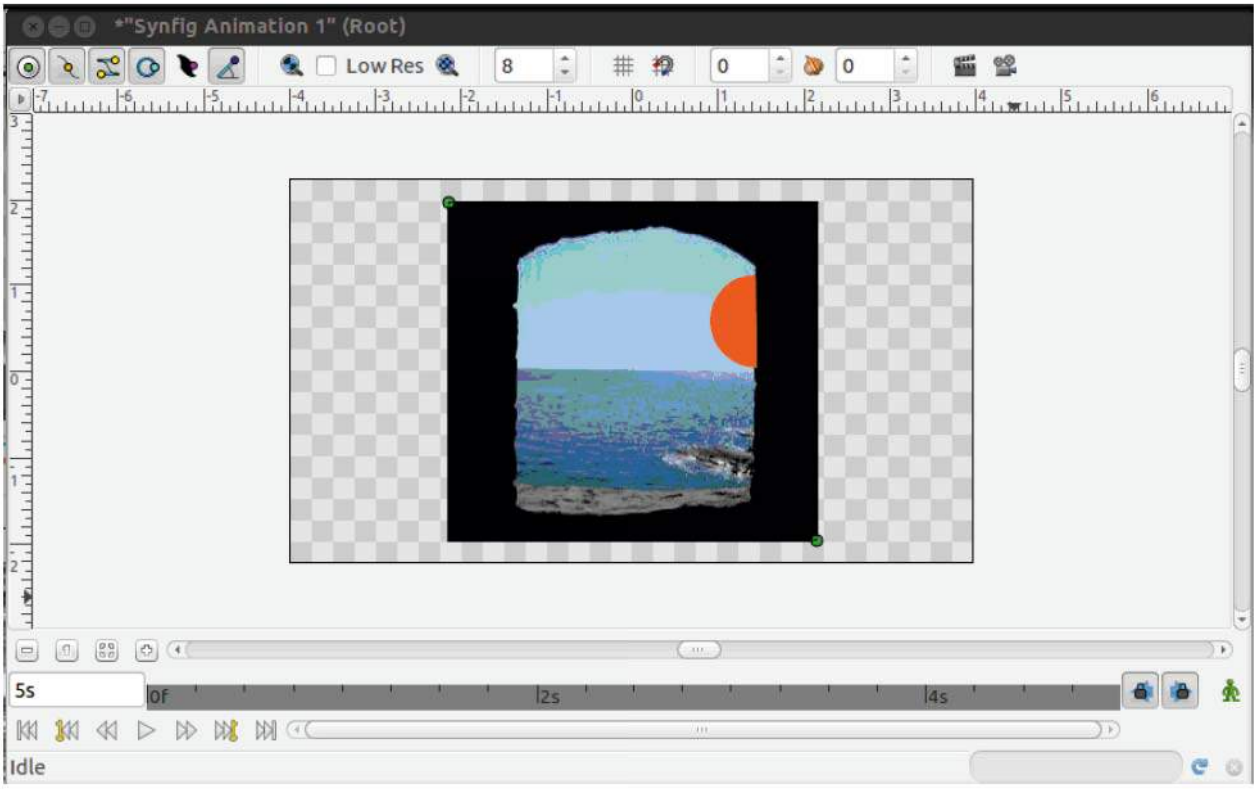


Figure 5.30 : Effect of masking to reveal an object

Create slideshow using Synfig

Assume that you want to create a slideshow on "Gujarat Tourism" wherein you want to show the viewer glimpse of Gujarat. To show the glimpse we will need to display images one after the other. Also at a time, only one image is visible. As we need to work with individual image they should be placed on different layers. To handle the visibility of the images we will use the parameters 'blend method' and 'amount' property of each layer. We have earlier used the blend method property which

defines the layering of the image on everything below it. The amount property is used for the visibility of the image just like the alpha value. Amount value of 1 signifies the layer is fully visible and 0 means the layer is fully transparent. Follow the steps given to create the slideshow:

- Create a new file.
- Press CTRL + i to import the image. If you do not have images then download some from the Internet. Import the images one by one on the canvas. Place each image in a different layer. Figure 5.31(a), (b), (c), (d), (e) and (f) shows the six imported images in the layers panel. Resize the images to the size of the canvas.

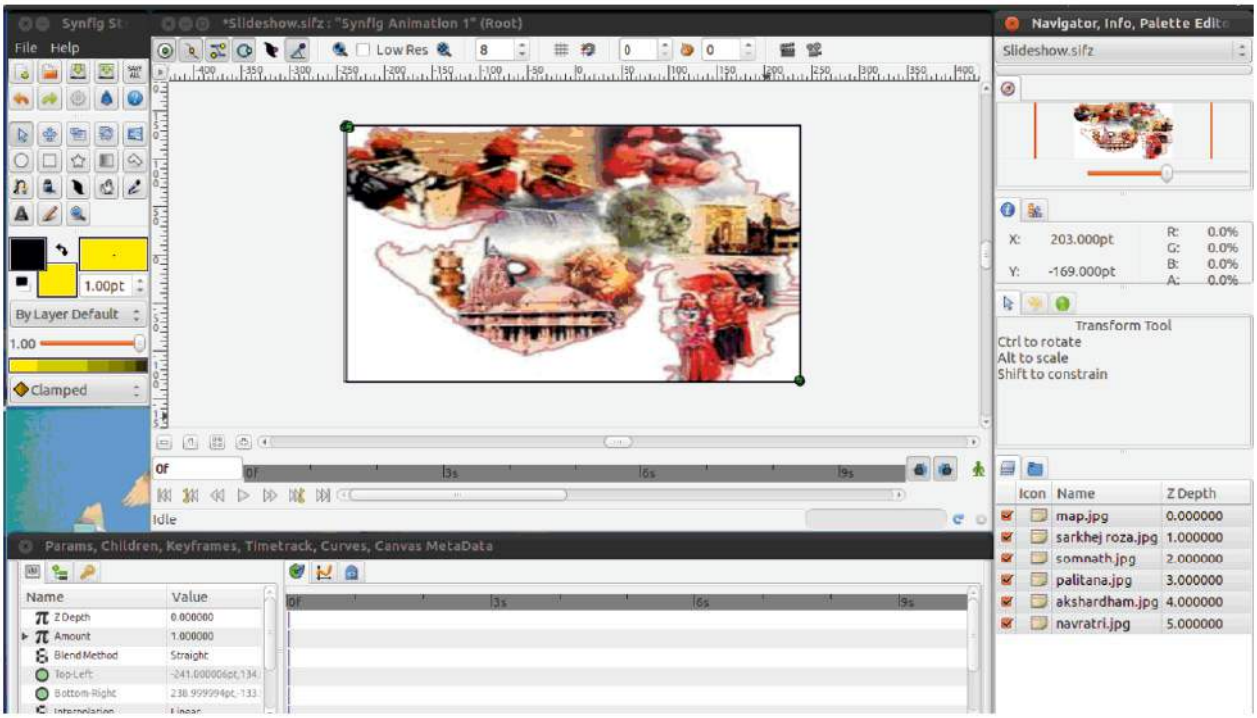


Figure 5.31(a) : Imported Gujarat Map image

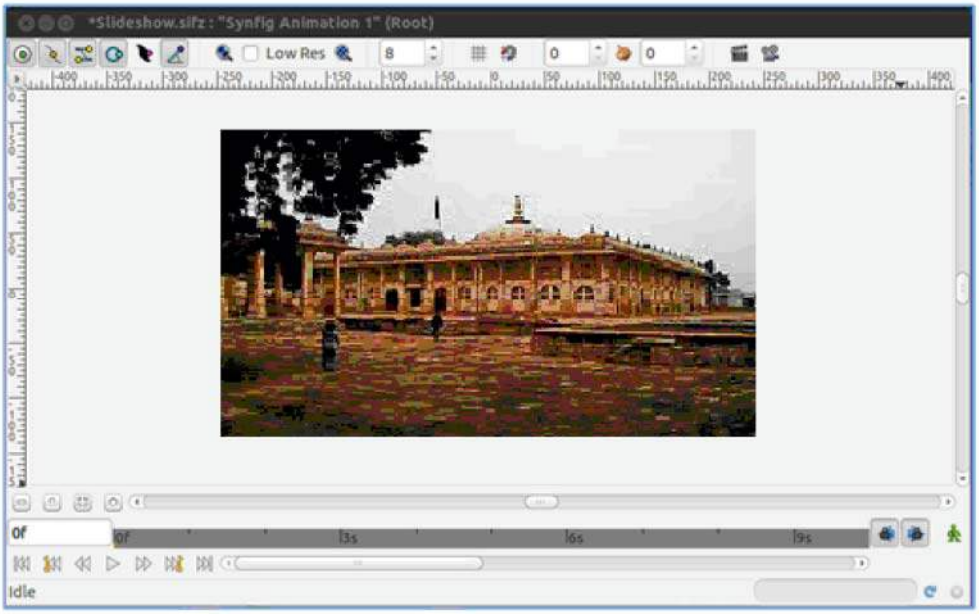


Figure 5.31(b) : Imported Sarkhej Roza image

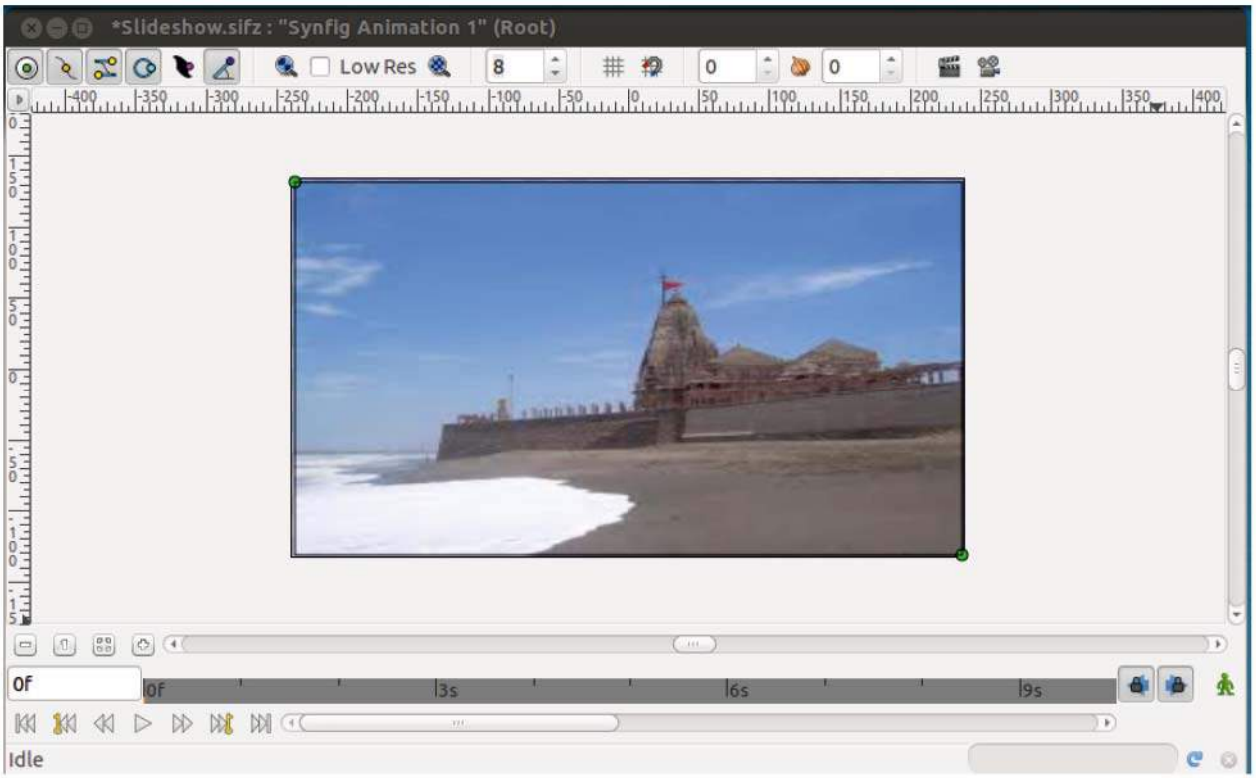


Figure 5.31(c) : Imported Somnath image

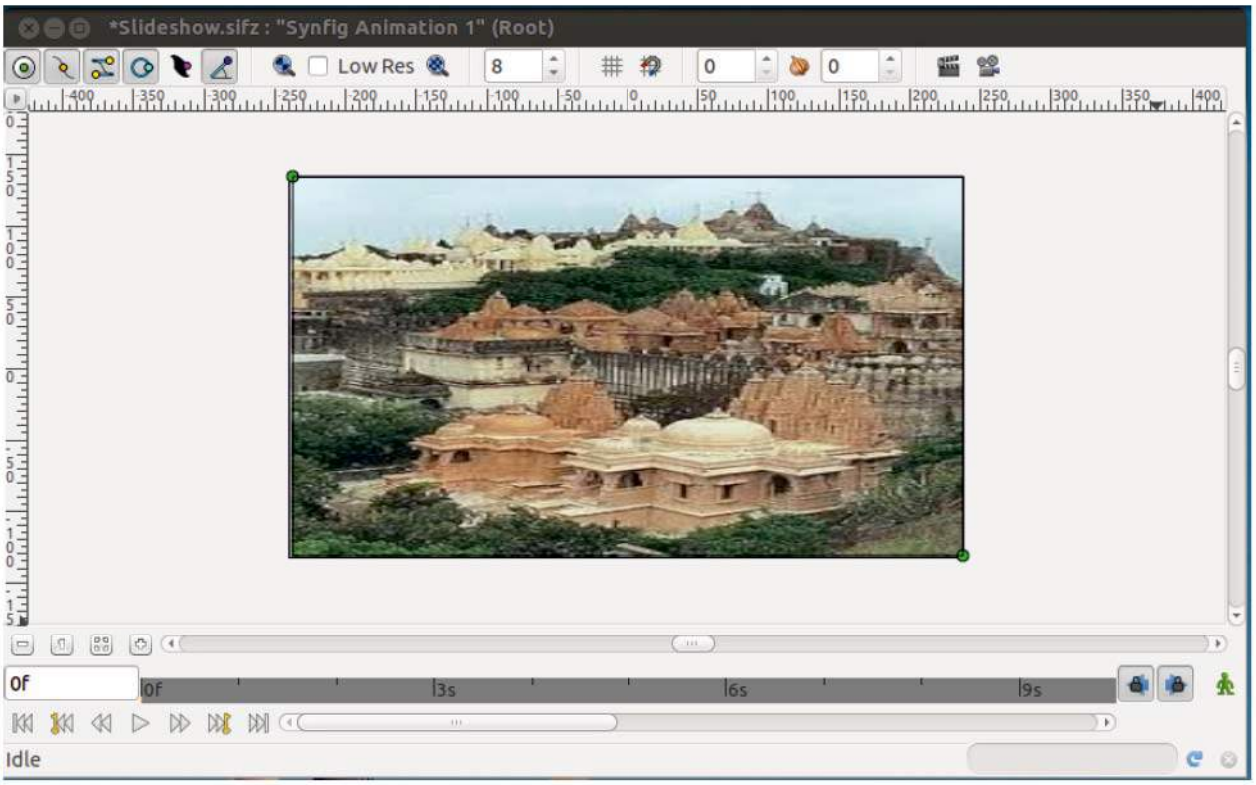


Figure 5.31(d) : Imported Palitana image

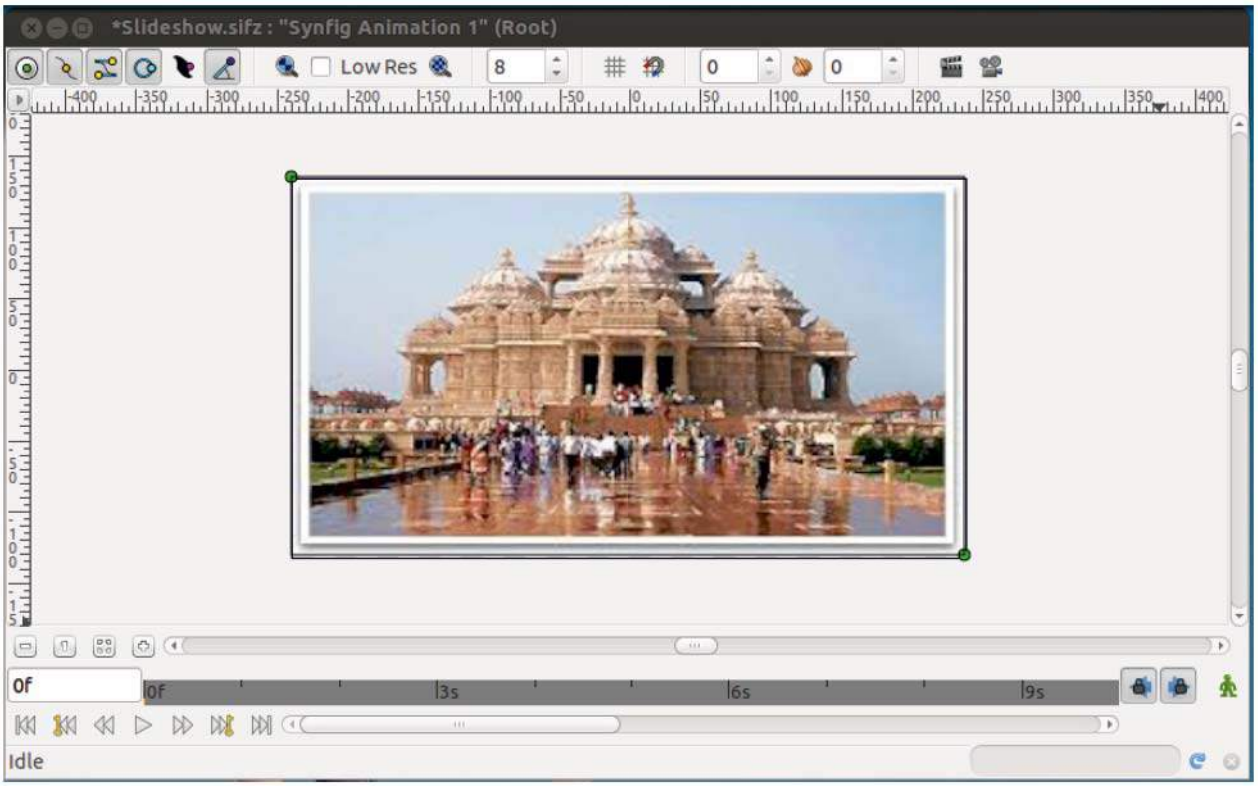


Figure 5.31(e) : Imported Akshardham image

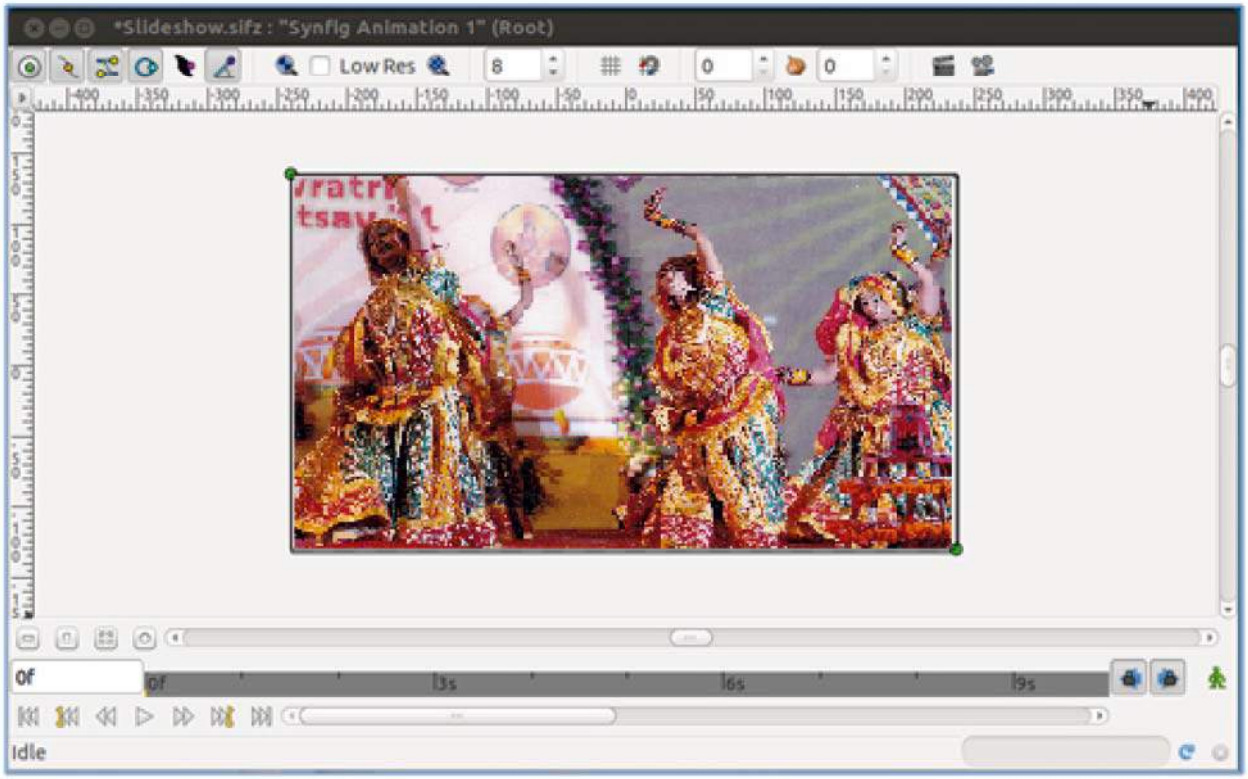


Figure 5.31(f): Imported Navratri image

- Press shift key and select all the images in the layers panel. Go to parameters panel and change the blend method to 'straight' for all the images as shown in figure 5.32.

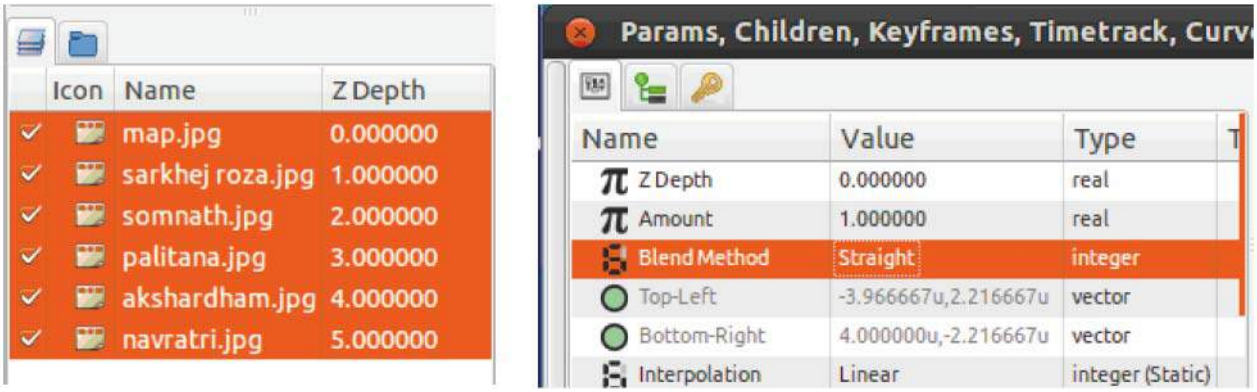


Figure 5.32 : Select the images and change blend method to straight

- After setting the blend method property we now need to work with the amount parameter. But like blend method, the amount parameter cannot be set collectively for all the layers. We need to work with amount property for each layer individually. Here we need to convert the amount property to 'Timed swap'.
- Select the topmost map layer as shown in figure 5.33 and in the parameters panel select the amount parameter and right click. Right clicking will open a context menu. Select Convert → Timed swap. Convert specifies that the parameter is controlled automatically in different ways. Each parameter can be converted and contains different options. Figure 5.34 shows the amount property Timed swap. This will add new sub parameters: Before, After, Time and Length to the amount property as shown in figure 5.35. These options specify the visibility of the image, swapping it from 'before' to 'after' parameter using 'length' and finishing the swap at the specified 'time'.

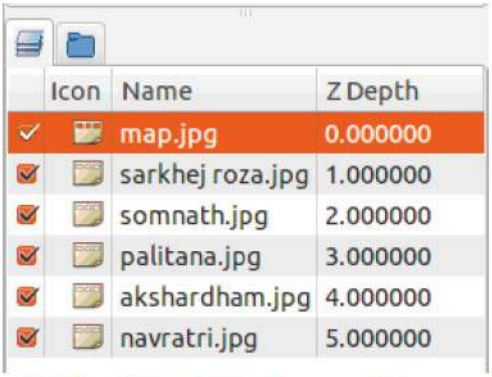


Figure 5.33 : Select the topmost image layer

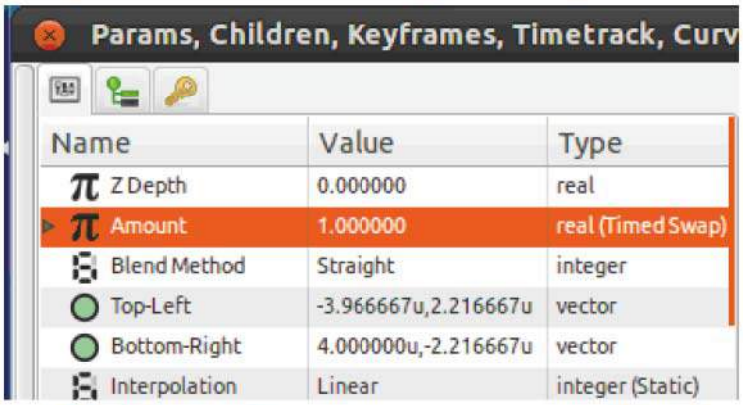


Figure 5.34 : Change the amount parameter to Timed swap

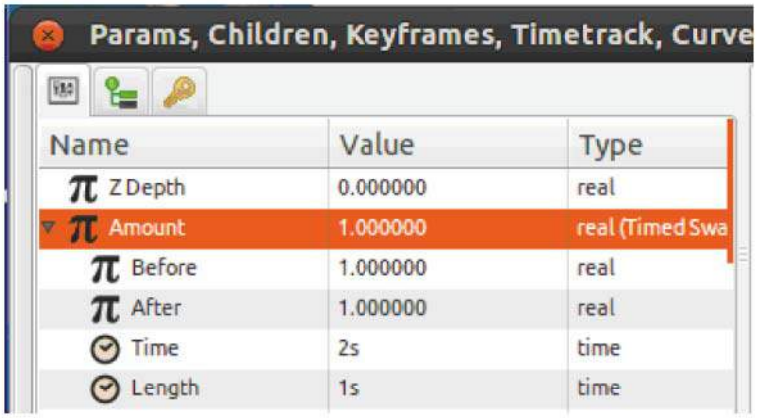


Figure 5.35 : Newly added parameters to the amount parameter

- Set the parameter 'before' to 1.0 and 'after' to 0.0. The parameter before set to 1.0 indicates the layer will be visible before the 'time' specified and after set to 0.0 indicates that the layer will be invisible after the specified 'time'. If you want the images to display for 5 seconds using 1 second transition between each image then change the length to '1s' and time to '5s'. After 5 seconds the image will be invisible. Figure 5.36 shows the settings applied to amount parameter.

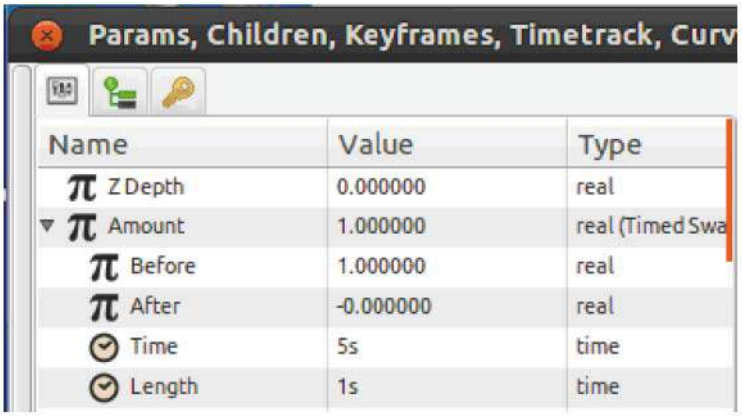


Figure 5.36 : Map layer Sub parameters settings

- Select the next image layer. Make it timed swap just like we did in earlier layer. Set the parameter 'before' to 1.0, 'after' to 0.0 and length to 1s. Set time to 10s. The swapping will be done at 10s. And image will be invisible after 10s. Likewise for the next image

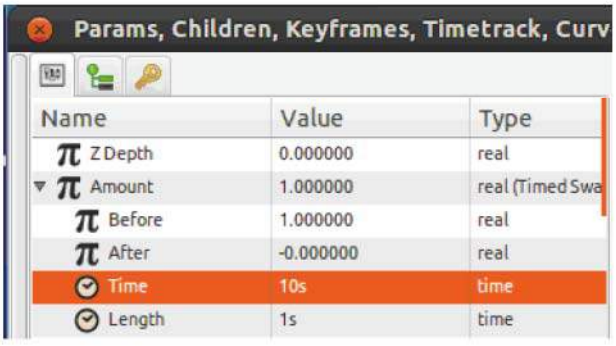


Figure 5.37(a) :
Time setting for Sarkhej Roza layer

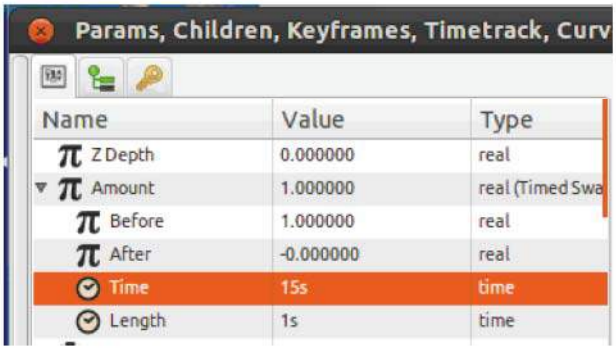


Figure 5.37(b) :
Time setting for Somnath layer

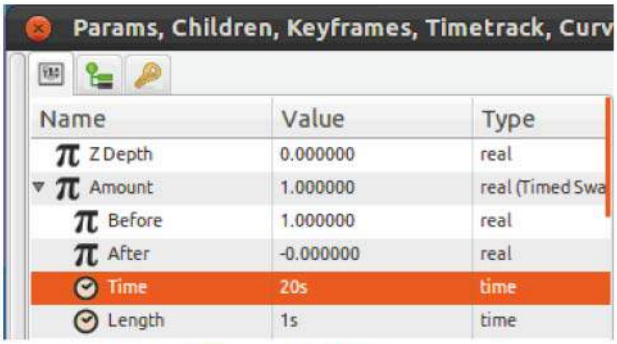


Figure 5.37(c) :
Time setting for Palitana layer

Figure 5.37(d) :
Time setting for Akshardham layer

layers change the time to 15s, 20s, 25s and so on. Figure 5.37 (a), (b), (c), (d) and (e) shows the settings for all the image layers.

Figure 5.37(e) : Time setting for Navratri layer

- Click on Caret → Edit → Properties. Select the time tab and set the end time equal to the last image layers time value. In our case the last image layer time is 30s. You can set the end time a little more than 30s if you want a blank screen in the end. Figure 5.38 shows the end time setting.

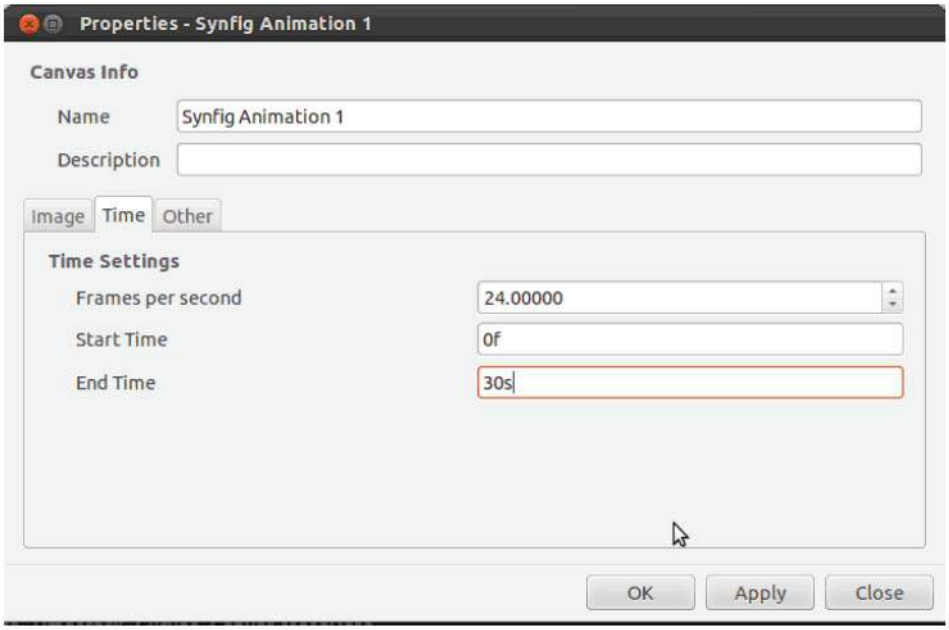


Figure 5.38 : End time set to 30s

- Click the play button to preview the presentation.
- Save and render the file. Figure 5.39 shows the output in web browser.

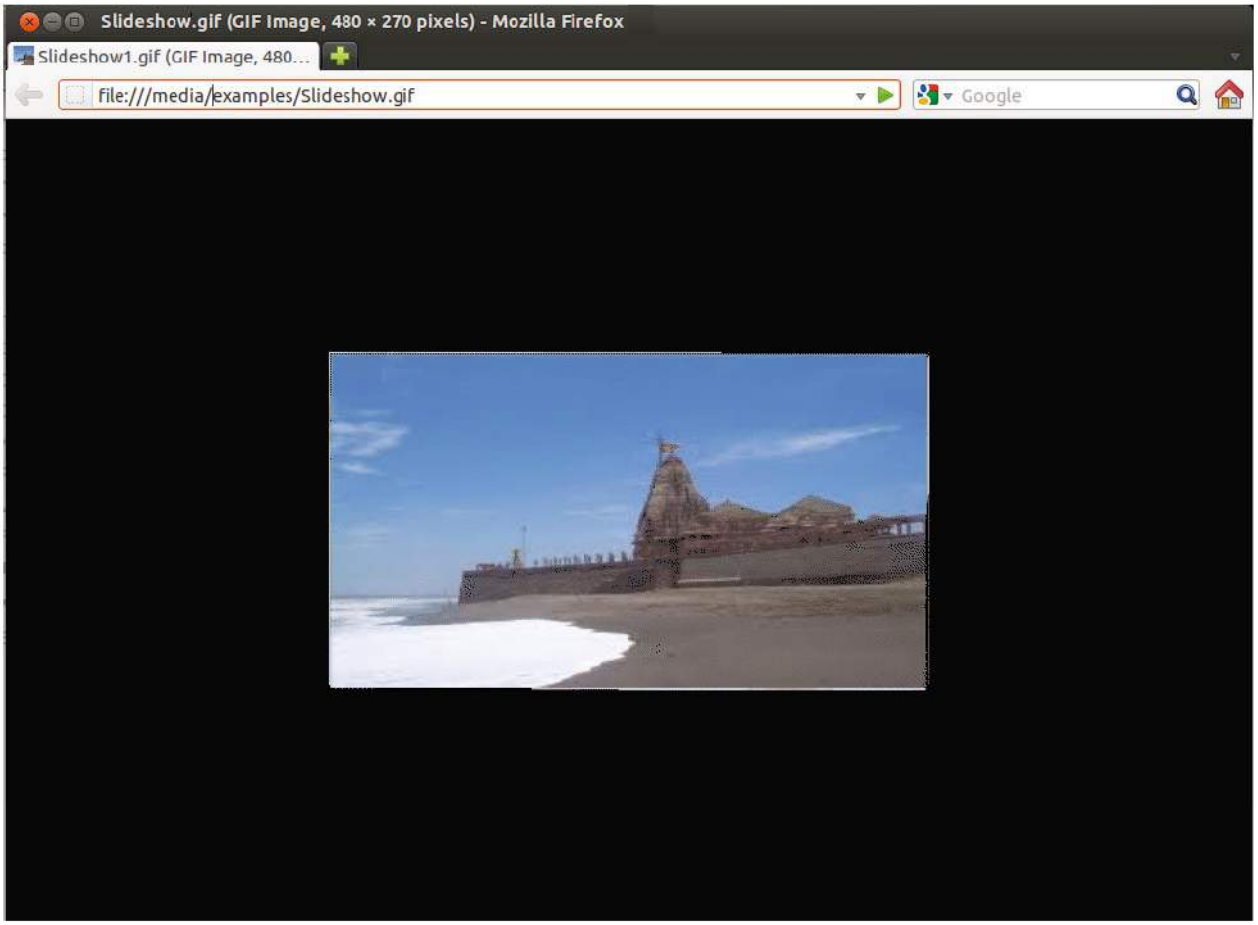


Figure 5.39 : Output displayed in web browser

If you want to place a background in the slideshow then encapsulate all the image layers. Place the background as the bottommost layer. The blend and amount parameters will be applied only in the encapsulated layer. We can also add text to the images and display the text during the slideshow.

Interactivity

Sometimes in animation, you need to move according to the choice of the user. Also the user might want to skip a particular portion of the animation and move further. These actions used for interactivity in animation software are known as action scripts. Action scripts are used to add flexibility to the animation programs. The term action script signifies the use of scripting language to add flexibility to the actions performed during the animation. It helps the user to either play the animation on a particular action or move as per his choice during the animation.

As of now, Synfig studio does not support action scripting directly in the animation software. But we can use javascript or java to add interactivity to our animation. It may not be possible here to discuss in detail the use of javascript or java to our animation. But let us discuss an example wherein we have used javascript to add interactivity to our animation.

Assume that you want to play the animation only when the user takes the mouse over the image. To achieve this, first we take two images. One of the images is the static image. In our example the static image is static-ball.gif. The second image is the animated image. In our example the animated image is animated-ball.gif. The animated image is the rendered file from Synfig in a format like gif, jpeg or bmp. The listed options are available when we render the files in Synfig as discussed in the earlier chapters. When the user opens the web browser, static image will be displayed. After he/she takes the mouse over the image, the animated image will be displayed using the javascript code. When the user moves the mouse away from the image the animation will stop. Follow the steps as mentioned:

- Create a new file in Gedit and write the code given:

```
<html>
<head>
<title>Rollover with a Mouse Events</title>
<script type="text/javascript">
if( document.images ){
    var image1 = new Image();       // Preload an image
    image1.src = "static-ball.gif";
    var image2 = new Image();       // Preload second image
    image2.src = "animated-ball.gif";
}
</script>
</head>
<body>
<a href="#" onMouseOver="document.myImage.src=image2.src" onMouseOut="document.myImage.src=image1.src">
<img name="myImage" src="static-ball.gif" />
</a>
</body>
</html>
```

- Save the file with extension .html.
- Open the file in a web browser to see the output. Figure 5.40 shows the output when the page loads. Taking the mouse over the image will animate it. Figure 5.41 shows the animated image.

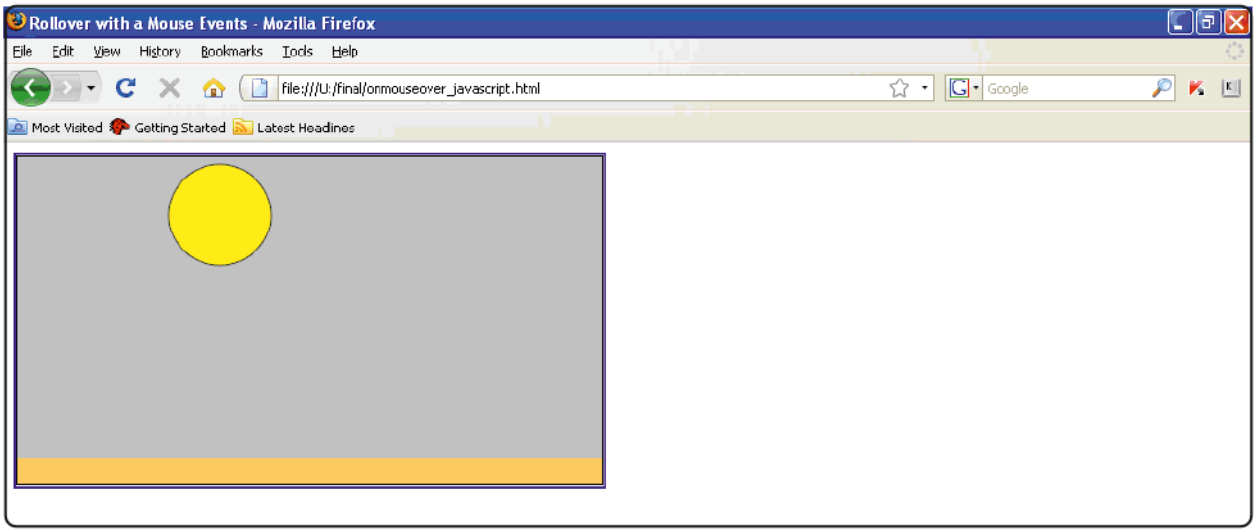


Figure 5.40 : Image displayed when the page loads.

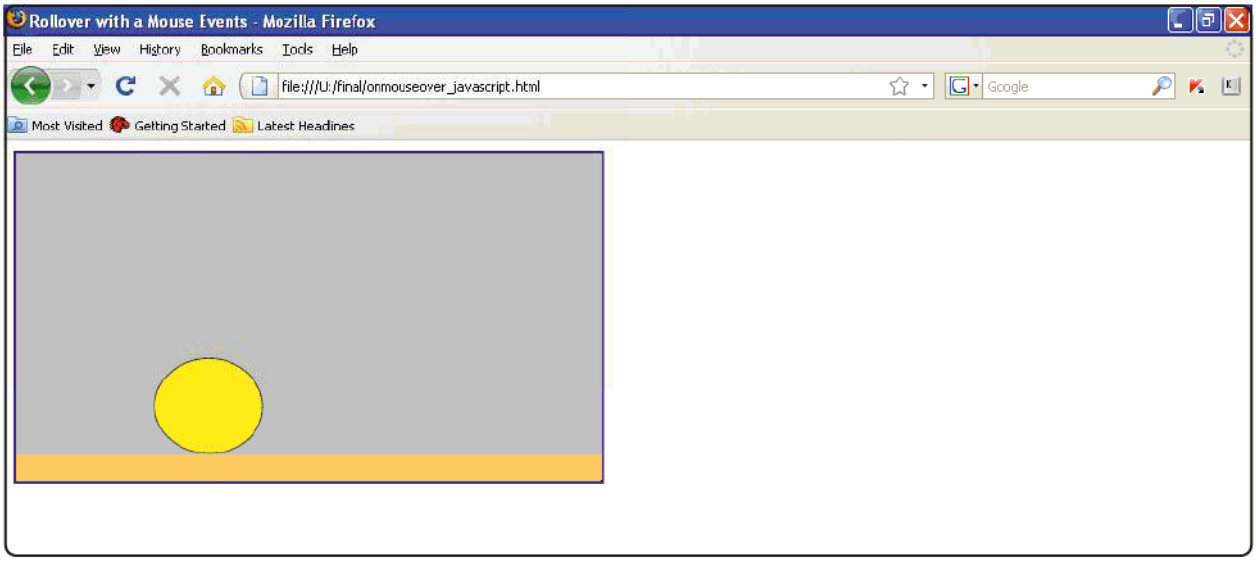


Figure 5.41 : Animated image displayed when the mouse is over the image.

Thus, using javascript we added interactivity to the animation.

Portability of files

The animation files created using particular animation software can be converted into Adobe Flash files (SWF files) for further manipulation. SWFTOOLS is an open source software tool suite which provides the conversion of various file formats into swf files. SWF is an Adobe flash file format used for vector graphics animation. After converting the files into swf format they can be further manipulated or created using Adobe Flash animation software. The tool includes programs for reading SWF files, combining them, and creating them from other content like images, sound or video file. Table 5.1 shows some of the programs used for conversion of files into swf format.

Program	Description
GIF2SWF	Converts gif files into swf files
PNG2SWF	Converts png files into swf files
JPEG2SWF	Converts jpeg files into swf files
PDF2SWF	Converts pdf files into swf files
WAV2SWF	Converts wav audio files into swf files
AVI2SWF	Converts avi animation files into swf files
FONT2SWF	Converts font files to swf files
SWFExtract	Extracts images, sounds and movie clips from swf files

Table 5.1 : List of programs supported in SWFTOOLS

Using the programs listed in the table 5.1, we can also convert the gif or jpeg files created using Synfig to Adobe Flash files (swf) for further manipulation.

Summary

In this chapter we learned how to insert pictures and work with them. Pictures like any other object created in Synfig can be rotated, transformed or scaled. We also discussed to hide or reveal an object using the concept of masking. We can also create a slideshow of images by working with the blend method and timed swap parameter. Adding interactivity to the animation can be provided by using javascript or java.

EXERCISE

1. What is masking?
2. Explain the use of timed swap option.
3. What is action script? Why do we need it?
4. **Choose the most appropriate option from those given below :**

   (1) Which of the following commands is used as a shortcut key to import image?

   (a) CTRL + a (b) CTRL + m

   (c) CTRL + i (d) CTRL + n

   (2) Which of the following colour represents the points used to change the size of the image?

   (a) green (b) yellow

   (c) blue (d) red

   (3) Which of the following amount value signifies that the layer is fully visible?

   (a) 1 (b) 0

   (c) 2 (d) -1

   (4) Which of the following is used to add actions as a part of interactivity in animation software?

   (a) action script (b) java script

   (c) action programs (d) java actions

   (5) Which of the following is a feature used to hide or reveal areas of a layer?

   (a) tweening (b) masking

   (c) morphing (d) encapsulation

   (6) The visibility of the images is handled by which of the following parameters?

   (a) blend, amount (b) mask, amount

   (c) blend, import (d) mask, blend

(7) Which of the following blend method is used to reveal or hide the object?

(a) composite (b) alpha over

(c) straight onto (d) straight

(8) Timed swap property adds which of the following sub parameters?

(a) up, down, time, length (b) before, after, size, shape

(c) before, after, time, length (d) top, bottoms, time, length

LABORATORIES EXERCISES

1. Create a slideshow on the theme of "Khushboo Gujarat ki".
2. Create a slideshow on the theme of "Festivals of India". Use text to display name of each festival.
3. Insert an image showing scenery with sky and mountains. Using masking show the sunrise in the image.

6 Basic Ubuntu Linux Commands

In standard 9 we have studied in depth the working of an operating system called Ubuntu. During the study we learned that once a user logs into the computer system having Ubuntu Linux, he/she can interact with the computer using command line interface or graphical interface. Both these interfaces are important and have their own uses. In this chapter we will learn how to use the command line interface in detail. The command line interface allows us to access the real power of Linux with greater efficiency. It is the most influential parts of any Ubuntu system.

Starting Up the Terminal

To open a command-line console in a graphical interface, a window named terminal window is provided in Linux. To open a terminal window, click on Applications àAccessories à Terminal alternatively you can use CTRL+ ALT + t keys together. A terminal window similar to the one shown in figure 6.1 will be seen on the screen. Note that the look of your actual window may be different from the one shown in figure 6.1 as the user might be different.

Figure 6.1 : Terminal Window

Once the window is clearly visible, you will see a blinking cursor preceded by some letters, and perhaps numbers and symbols, ending with a $. The first word in that string of characters is username, followed by the symbol @. The symbol @ is followed by the name of the computer that is being used. Finally you will find a colon and the name of the directory you are working in (Generally you start working in your home directory, which is represented by a ~ symbol).

The command prompt indicates that the interface is ready to interact with the user in the form of commands. A command is basically a program that accomplishes certain task. Once the prompt is displayed we can issue commands as described in this chapter. Before discussing various commands, let us revise our knowledge about the term shell. A shell is the command-line interface. Shell is a user program or an environment provided for user interaction. It is a command language interpreter that accepts or issues commands, understands it, interacts with the kernel to execute it and displays results as per given instruction. Numerous shells are available to work on the Ubuntu Linux systems, but the shells available on a particular system may vary.

Some popular shells provided with Linux are Bourne shell (sh), C shell (csh and tcsh), Korn shell (ksh) and bash (sh) shell. Bourne shell with sh as its acronym is the earliest Unix shell used as command line interface. Bourne shell provides basic mechanisms for shell script programming, which allows us to write a program based solely on commands. C shell identified as csh is another shell commonly found on Linux systems. Shell programming can be done using C programming syntax in this shell. The newer version of csh is tcsh. It provides additional shell script programming features to address the limitations of csh. The Korn shell or ksh was developed to combine the features of both sh and csh. Bash shell is a newer version of Bourne shell. Thus it contains same syntax and functions as sh. Nowadays bash is considered standard shell for Linux systems and is thus commonly used and available on all Linux operating systems.

Listing the shells available on the system

To find all available shells in your system you can use the cat command. Type the command as shown in figure 6.2 on the command prompt. You will get list of all the available shells in your computer system. The list of shell that you get as an output may be different from the one shown in figure 6.2 depending on your system configurations. The cat command is discussed later in this chapter.

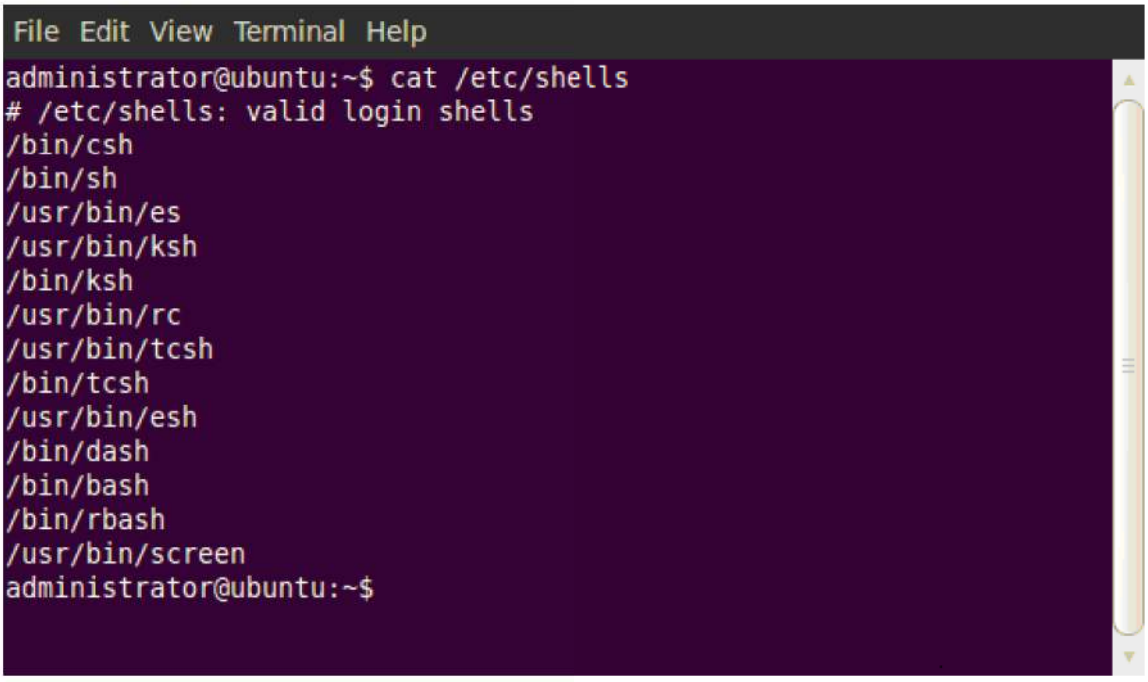


Figure 6.2 : Different available shells

Determining the default shell

Each Ubuntu Linux account is configured with a certain shell as its default command line interface. Each time you log on, this default shell is utilized for working within the system. Linux operating system comes with certain variables that contain current environment settings as its values and thus are known as environment variables.

The value of default shell is stored in one such environment variable named SHELL. Thus by displaying the value of the variable SHELL, we can come to know which our default shell is. To display the value of any variable, *echo* command can be used. Type echo $SHELL on the command prompt and press Enter key. Default shell will be displayed on the screen as shown in figure 6.3. Note that Linux commands are case sensitive hence SHELL, Shell and shell are not same.

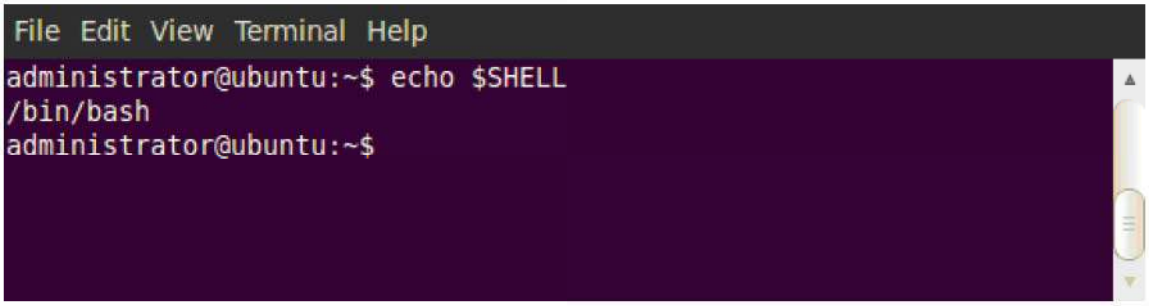


Figure 6.3 : Default Shell

Changing the current shell

As discussed earlier we have different shells available with Linux. To change your default shell, type the name of the shell you want to use on the command line. For example, if you want to use the C shell (provided it is available on the system), type csh at a command prompt. Then the command prompt will provide a csh interface.

Note :
A shell change is temporary and will last only as long as you are logged on that command line.

To return to default shell, type exit or press CTRL + d at the command prompt of the new shell.

Command Syntax

The syntax of Linux commands is uniform. It consists of three parts, in the order specified below :

- **Name :** It is the name of the command, for example ls, echo etc.
- **Options :** It is possible to alter the behavior of the commands by specifying additional options. A command may have zero or more options. Options when present starts with a hyphen symbol (-) and are usually a single letter or a digit. Some commands may have options with double hyphen and/or sequence of letters or digits. Depending on the command, the number and meaning of the options will vary.
- **Arguments :** Along with options user can also provide arguments. A command may take zero or more arguments to do its work. The number and expected meaning of the arguments vary from command to command. Some commands may take no arguments; others may take an exact number, while other commands may take any number of arguments.

Linux commands can be classified as internal or external based on whether its binary file exists or not. The commands that have a binary file explicitly stored in either /sbin, /usr/sbin, /usr/bin, /bin, or /usr/local/bin directories are called external commands. They are generally executed by the kernel and will generate a process id at the time of execution. Most of the commands that we use in

Linux are external commands. On the other hand the commands directly executed by the shell are called internal commands. Internal commands do not generate a new process.

To know whether a command is internal or external we can use the *type* command. The syntax of type command is shown below :

$type command

For example if we execute a command

$type info

we will get the output as shown below :

info is /usr/bin/info

As can be observed this refers to a binary file *info* stored in /usr/bin/, this indicates that info is an external command.

Issuing General Purpose Commands

As you are now familiar with the syntax of commands, let us see how to issue commands through command line interface. The best way to start with learning Linux commands is to try working with some general purpose Linux commands. You can issue a command by typing a command name followed by necessary options and arguments. Other way is you can type in first few letters of a command, press the tab key and the shell automatically provides the remaining information. For example, to display a calendar, type ca on the prompt and press tab key. Linux shell will automatically display list of all commands starting with alphabets ca including the calendar command *cal* on the screen. If you get more than one command in the list, then type the desired command on the prompt and press Enter key to execute the command.

Calendar (*cal*)

The *cal* command is used to display a calendar of any specific month or entire year. The default output of *cal* command is calendar of the current month. See figure 6.4.

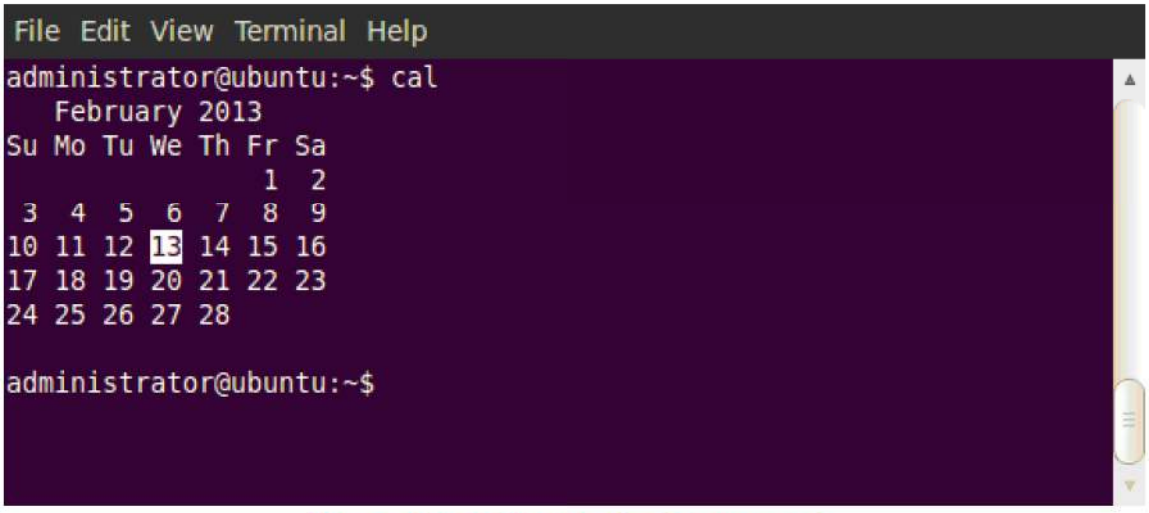


Figure 6.4 : Output of cal command

We can change the calendar as per our requirement, for example to display the calendar of January, 2013 type the following command on the prompt and press Enter key.

$cal 01 2013

The output of this command will be similar to the one shown in figure 6.4 except that the month would be January 2013. Similarly if we want to display the calendar of the entire year 2013, we simply have to type cal followed by the year as shown below :

$cal 2013

Note that the calendar of entire year may not be displayed on the entire monitor screen; hence we will have to use a pipe operator as shown below :

$cal 2013 | more

In the above command we have concatenated two commands, here *more* is also a command which takes input from the *cal* command. The pipe (|) symbol used in between the two commands is discussed in detail later in this chapter.

Date (*date*)

Another utility command is *date;* it is used to display the system date.

$date

The output of the command is shown in figure 6.5.

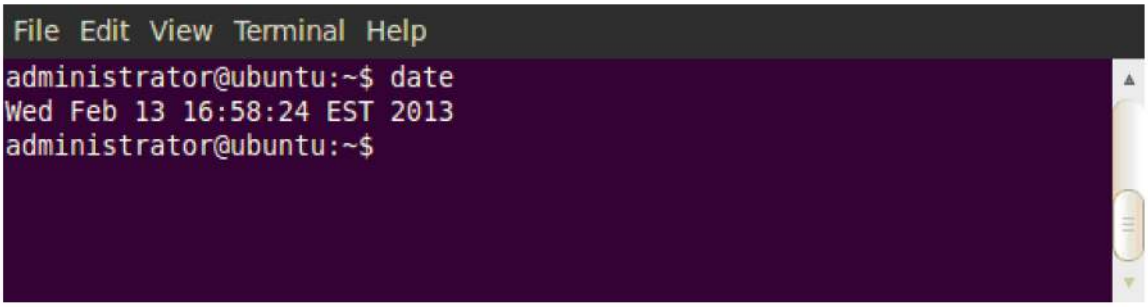


Figure 6.5 : Output of date command

Observe that the output displays both date as well as time. The date command can also be used with suitable format specification as arguments. Each format is preceded by + symbol, followed by % operator and a single character describing the format. For example, to display only the current date in mm/dd/yy format use the command shown below :

$date +%D

Figure 6.6 shows the output of the command.

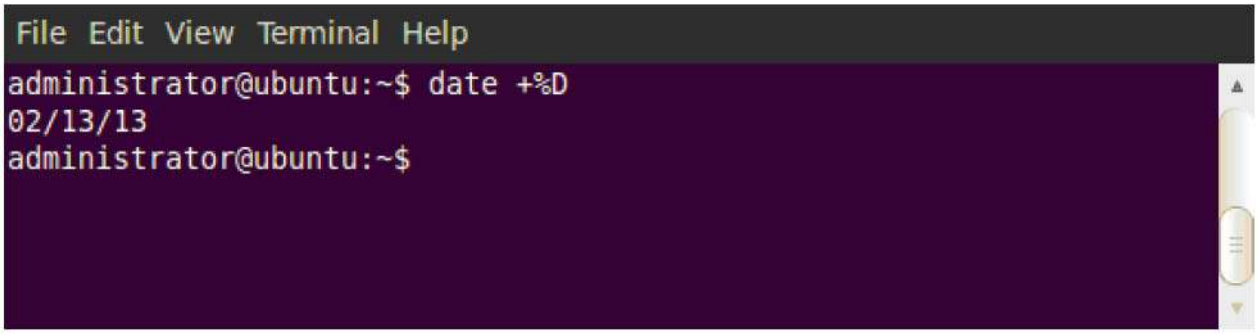


Figure 6.6 : Output of formatted date command

The command line calculator (*bc*)

The *bc* command in Linux is a command line calculator. In addition to performing simple mathematical functions, it can also perform conversions between different number systems, as well as allows us to use some scientific functions. To work with this command use the syntax given below :

$bc -l

A screen similar to the one shown in figure 6.7 will be displayed. Notice that the dollar prompt is not visible on the screen; this indicates that the bc command is now ready to take input from you. The **-**l switch is used to include the standard math library.

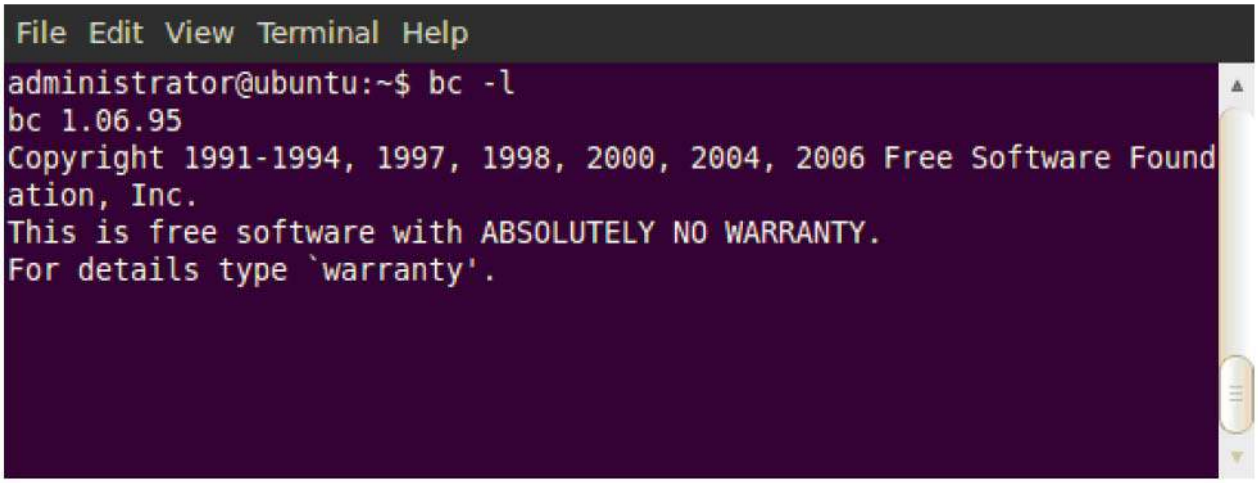


Figure 6.7 : Initiation of bc command

Now, just type the formula that you want to evaluate at the blinking cursor and the press Enter key. You can type a simple expression like 5 * 5, or you may type a complex expression with grouped operators. Let us type the expression **((99.1 / 5.15) * 99.9)**, now press the Enter key. The command will display the output in the next line as shown in figure 6.8.

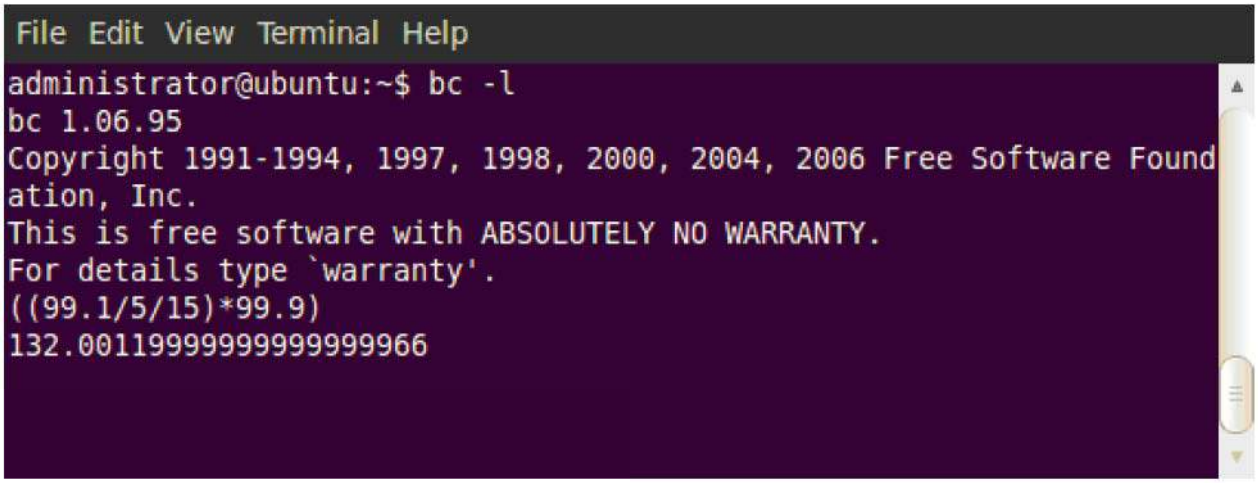


Figure 6.8 : Output of bc command

Observe that we still are not able to see the prompt; this simply means that we can continue working in command line calculator mode.

In addition to the normal mathematical functions like addition, subtraction, multiplication, division, modulus, and exponents, we can also use trigonometric or logarithmic functions like sine, cosine, arctangent, and log. For example, if you need to find the natural logarithm of value 2013, use the command **l(2013)** and you will get output **7.607381425639791484 20.**

The *ibase* function allows us to set the numbering system that we want to use for input. Similarly the *obase* function allows us to set what numbering system to use for output. Let us try to convert numbers from decimal number system to hexadecimal number system. First, we need to set *obase* as shown below :

obase = 16

Now, type the number you want to convert to hexadecimal as shown in the example and press the Enter key.

256

100

Here 100 is the hexadecimal equivalent of decimal number 256.

Similarly if you want convert this result to binary number system then just change the obase again as shown here.

obase=2

Now type **100** and the output will be 1100100, observe that this is binary equivalent of decimal 100 and not 256. The reason for this is very simple, we have not changed ibase, and hence all entries are considered to be decimal entries.

To convert hexadecimal value 100 to its binary equivalent, set ibase and obase as shown below :

ibase=16

obase=2

Now type a hexadecimal number that you need to convert to binary. For example, type **100** and you will get **100000000** as a result.

To return back to decimal mode set ibase to 10. Execute the following to find out the square root of a number using sqrt function available in math library.

sqrt(256)

16.00000000000000000000

As compared to graphical calculator the command line bc calculator is more faster and flexible. To return to the Linux command prompt, press CTRL+ d.

Displaying a message (*echo*)

We need to display message very frequently when using command prompt. The *echo* command is used to display a message on the terminal. For example, type the following command and press the Enter key. The string written after echo will be displayed on your monitor screen.

$echo Hi, I am learning Ubuntu Linux

Hi, I am learning Ubuntu Linux

It is also possible to enclose the string within double quotes. The output will not contain the double quotes. The echo command can also be used to display values of variable. For example, define a variable named cost and assign it value 10 as shown below :

$cost=10

Once you press the Enter key you will be returned to prompt. Now type the command given below :

$echo The cost of product is Rs. $cost

The cost of product is Rs. 10

To display the value of cost on the screen it is passed to the echo command. Notice that in the string we have written cost twice. The one that is prefixed with $ symbol represents a variable, while the other is a normal string. When the echo command finds any string prefixed with a $ symbol will consider it to be a variable. It will then try to print the value of variable.

The echo command can be used along with other commands to give meaningful output. For example we may combine the echo and the date command in the manner shown below :

$echo Current time is ‘ date +%T ‘

Current time is 14 :55 :04

Observe that the command to be executed is placed within back quotes (quotes available on key with ~ sign), thus first the date command will be executed and then its result will be displayed using echo command.

Changing password (*passwd*)

A user needs to change password very often due to various reasons. The *passwd* command helps us perform this operation. It is used to change the password of the current login account by default. The following command allows you to change the password.

$passwd

Once you press the Enter key a message similar to the one below will be shown, along with the blinking cursor.

Changing password for administrator

(current) UNIX password :

Type your current password and press the Enter key, Linux will check whether you have entered a valid password, and if you have then it will prompt for the new password. You will be asked to enter new password and retype it again as shown below :

Enter new UNIX password :

Retype new UNIX password :

passwd : password updated successfully

If you have typed the new password correctly and it does not conflict with any guidelines decided for password, your new password will be registered by the system. In case of any problem you may get an error message. It is also possible to change the password of other user of the system by specifying the username after the passwd command. For example, if we have a user named harshal, to change its password we may type the following command.

$passwd harshal

In case of genuine user you will be allowed to change its password.

Clearing the Screen (*clear*)

While working on command prompt you must have observed that the screen often gets full. At times it also becomes difficult to see the output clearly, we have one simple solution for this problem, use the *clear* command to remove data on the screen.

$clear

Getting Help on the Linux Commands

Before looking at any other commands first let us learn how to get help when using commands on Linux platform. Linux provides two inbuilt commands namely *help* and *man* to assist the user while working on the command line interface. All the commands that we use in Linux supports the -h (or -help) option. This option generates a small description of how to use the command. Figure 6.9 shows the use of help command.

```
File  Edit  View  Terminal  Help
administrator@ubuntu:~$ mv --help
Usage: mv [OPTION]... [-T] SOURCE DEST
  or:  mv [OPTION]... SOURCE... DIRECTORY
  or:  mv [OPTION]... -t DIRECTORY SOURCE...
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options t
oo.
      --backup[=CONTROL]       make a backup of each existing destina
tion file
  -b                           like --backup but does not accept an a
rgument
  -f, --force                  do not prompt before overwriting
  -i, --interactive            prompt before overwrite
  -n, --no-clobber             do not overwrite an existing file
If you specify more than one of -i, -f, -n, only the final one takes
effect.
      --strip-trailing-slashes  remove any trailing slashes from each
 SOURCE
```

Figure 6.9 : Use of help command

Observe that the command used to display the help in figure 6.9 is *mv - - help*. The alternate mechanism to get help on the command is to use Linux online manuals. The man command activates a manual corresponding to a specific command that we need to look at. For example, the command *man mv* will show us the manual for the move command. Figure 6.10 shows the output of man command.

Observe that man command gives us exhaustive information of a command and may run into multiple screens. It generally displays one page at a time, as we press the Enter key the contents scroll down. To come out of the manual screen type alphabet 'q', this will take you back to the command prompt.

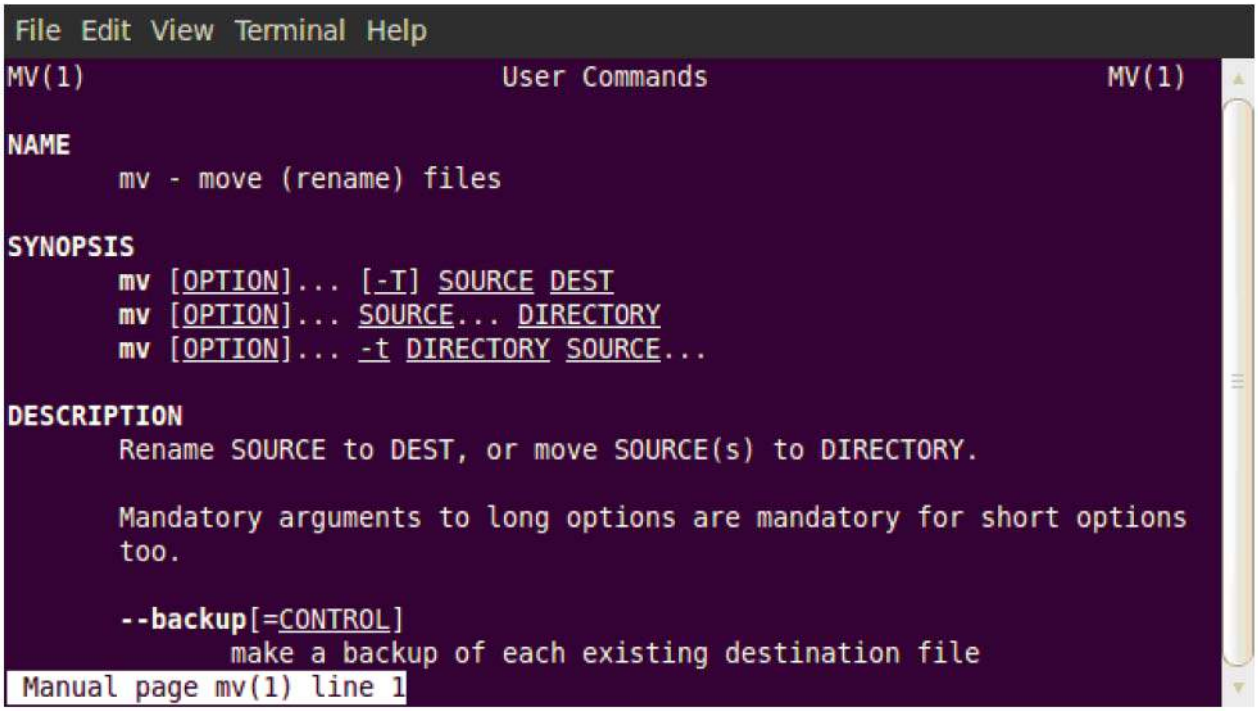


Figure 6.10 : Use of man command

In case we want only small description of a command then we may use the *whatis* command. It gives us one line explanation of the command, but omits any additional information about options. Figure 6.11 shows the sample output of whatis command.

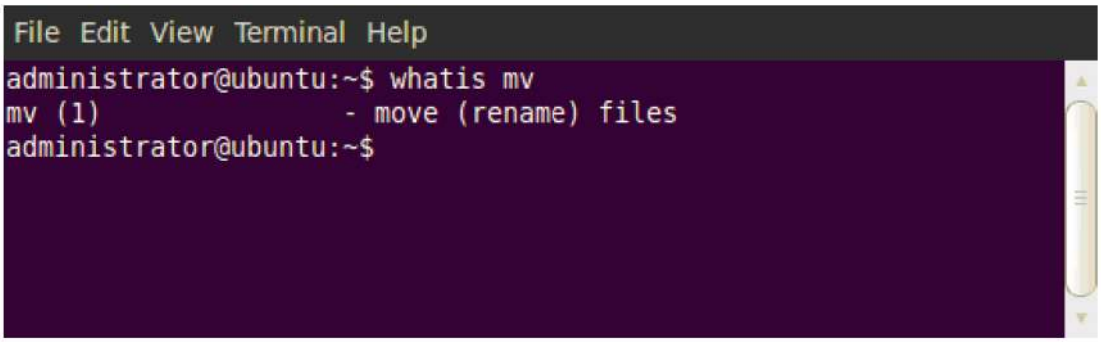


Figure 6.11 : Sample output of whatis command

Many times it may happen that we may not know which command to look for exactly. In such situations we may use the *apropos* command. The syntax of the command is mentioned below :

$apropos string

When we execute this command we will get a list of all the commands that has the string as the part of the command or command description. For example if we type *apropos copy* on the command prompt and try to execute it, then we may get screen full of commands that have copy as a string within the command or its description. A user must take caution while using this command. Sample output of apropos command is shown in figure 6.12.

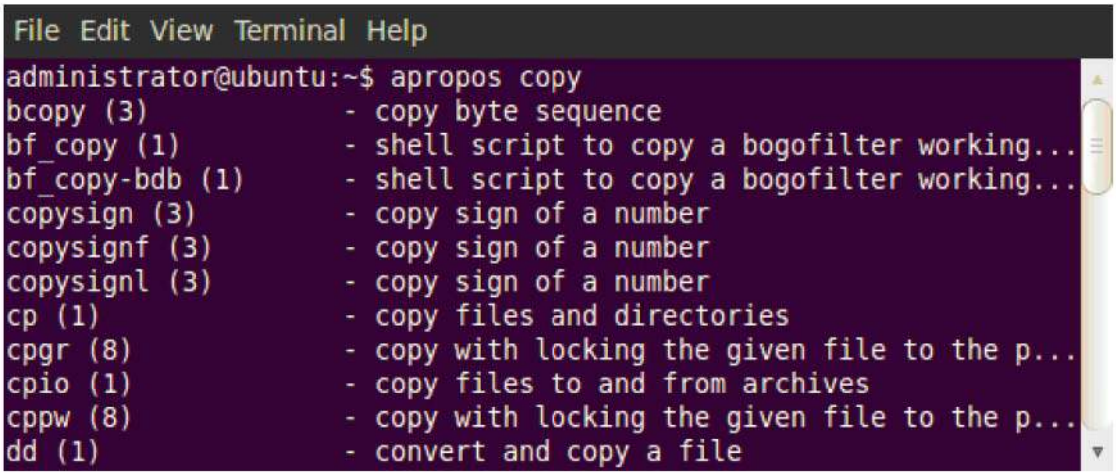


Figure 6.12 : Sample output of apropos command

Working with Directories

In Linux a directory is a special type of file that is used to store files and other directories. Here ' / ' symbol represents the root directory. All other directories come under root directory. Let us learn how to work with directories using terminal window.

Home directory

When a user logs on to the system, Linux automatically places the user in the directory called the home directory. It is created by the system at the time when a user account is created and generally will have a path /home/username. Here the username refers to the login name. For example if the username is harshal, then the home directory will be /home/harshal. It is possible to change this path if needed. The default working directory path is stored in system variable named HOME. We can cross check the directory by using the echo command.

$echo $HOME

/home/harshal

Note that the path displayed using this command is an absolute path name. Absolute path name is a sequence of directory names separated by / (slashes). An absolute path name shows a location in reference to the root directory. The first slash (/) is synonymous to root directory while the other slashes act as delimiters to other directory names. Thus the directory harshal is located within directory home which further is located in the root directory. Similarly the directory **/home/administrator** refers to home directory of username administrator.

It is possible to change the default path of the home directory. Suppose that you are able to see the output as shown below :

/home/its/ug1/svics

Here *svics* (home directory of user svics) is within the sub directory *ug1* (a directory that represents a sub group), which further is within a directory *its* (a directory that represents a group). Directory home and '/' have their default meaning.

Present Working Directory (*pwd*)

After we log into a system we can move around from one directory to another. But at any given point of time we will be located only in one directory. The directory where we are located at that moment is known as *current directory* or *present working directory*. To know the current directory that we are working in we can use the *pwd* command.

$pwd

Creating a Directory (*mkdir*)

A directory in Linux can be created using the *mkdir* command. The command takes the name of the directory to be created as its argument. Let us create a directory named *subject*.

$mkdir subject

The power of command line over GUI lies in its flexibility. If we create a directory using GUI we will be able to create one directory at a time, while it is possible to create multiple directories using a single mkdir command. The following command syntax illustrates the same.

$mkdir animals birds vehicles plants

The command when executed will create four directories named animals, birds, vehicles and plants in the current directory.

Change Directory (*cd*)

In the case when we need to store any data within a directory, first we need to make it our current directory. We can change (go within) a specific directory using the *cd* command. Let us try to create a directory named *math, science* and *economics* within a directory *subject.* To create these directories first we need to be in the *subject* directory. The command sequence shown below allows us to perform the said operation.

$cd subject

$pwd

/home/administrator/subject

$mkdir math science economics

Observe that the user name here is administrator. To again come back to the *administrator* directory, simply type the command below :

$cd ..

In the above command double dots (..) refer to the parent directory. Note that there should be one space between the cd command and the double dot.

Assume that you are in some internal directory that has path /home/administrator/subject/economics and you need to come back to the users home directory then again the cd command comes in handy. To come back to home directory we can issue the cd command twice as shown below :

$cd ..

$cd ..

The command sequence here will take us back one level at time. If we are say *M* levels down within the home directory, then we will have to execute the cd command *M* times. An alternative approach is to use a single cd command as shown below :

$cd ../../

Some example usage of the cd command along with its description is given in table 6.1.

Command Issued	Action Performed
cd ~/Desktop	Changes directory to /home/username/Desktop, from any current path. Here the symbol ~ refers to home directory of the user.
cd /	Changes directory to the root directory from any current path.
cd	Changes directory to the home directory from any current path.
cd -	Changes directory to the previously changed directory.
cd /var/www	Changes directory directly to the www sub-directory with directory var. It is useful when we know the path explicitly.

Table 6.1 : Sample cd commands

Remove Directory (*rmdir*)

An empty directory can be deleted by using the command *rmdir*.

$rmdir science

Here science is a directory name, and it will be removed using the above command only if it is empty. In case it is not empty we will get a message 'rmdir : failed to remove 'science' : Directory not empty'. In this case we have to first delete all the contents within the directory and then reissue this command. Note that it is also possible to delete multiple empty directories in the same way we created them.

To delete a non-empty directory with all its contents we can use the *rm* command as shown below :

$rm -r science

The rm command is discussed in detail in the later part of chapter.

Naming Conventions in Linux

We have created directories with different names using the mkdir command. While creating any object like a directory or a file in Linux we have to follow certain rules. On most of the Linux systems today, a name (of directory or file) can consist of up to 255 characters. Unlike Windows OS, names in Linux can practically consist of any ASCII character except for the slash (/) and the NULL character. Any other control characters or nonprintable characters are permitted. Examples of some valid names are .name, ^myname^-++, -{}(), test$#, xy.ab.ef etc.

However, it is recommended that a name should be relevant and contain only alphabetic characters, numerals, period (.), hyphen (-) and underscore (_). Linux strictly adheres to case sensitivity, thus math, Math and MATH are three different names. It is possible for the directories with these names to coexist at same level. If these names refer to a file then they can again coexist in the same directory.

Working with Files

Directory generally works as a container. The data is normally stored in files, which further may be stored in directory for proper arrangement and easy access. Text editors like nano, pico, vi or vim, ed and others are generally best suited for creating text file. However, many times the user needs to create a file quickly. The *cat* command comes in handy in such scenario. It is mainly used to display the contents of a small file on the terminal. But it can also be used to create a file, concatenate two files and append contents into a file.

Create a file using *cat* command

Let us create a file named *introduction* within the directory *economics*. To create this file you will have to first make the directory *economics* as your current directory. You can use the cd command here. Now type the command cat followed by a greater than (>) symbol and name of the file as shown below :

$cat > introduction

When we execute this command the cursor will be positioned in the next line, waiting for us to type the contents of the file. Type the text that you want to store in file and press CTRL + d. This will take you back to the command prompt. The combination CTRL + d in Linux indicate the end of file character. The greater than (>) symbol used in the above command is known as

redirection operator. It is used to instruct the shell that a redirection is required i.e., input should go to the specified file. The cat command when used without the greater than (>) symbol, displays the contents of the filename specified in argument. Figure 6.13 shows the process of creating and then displaying a file.

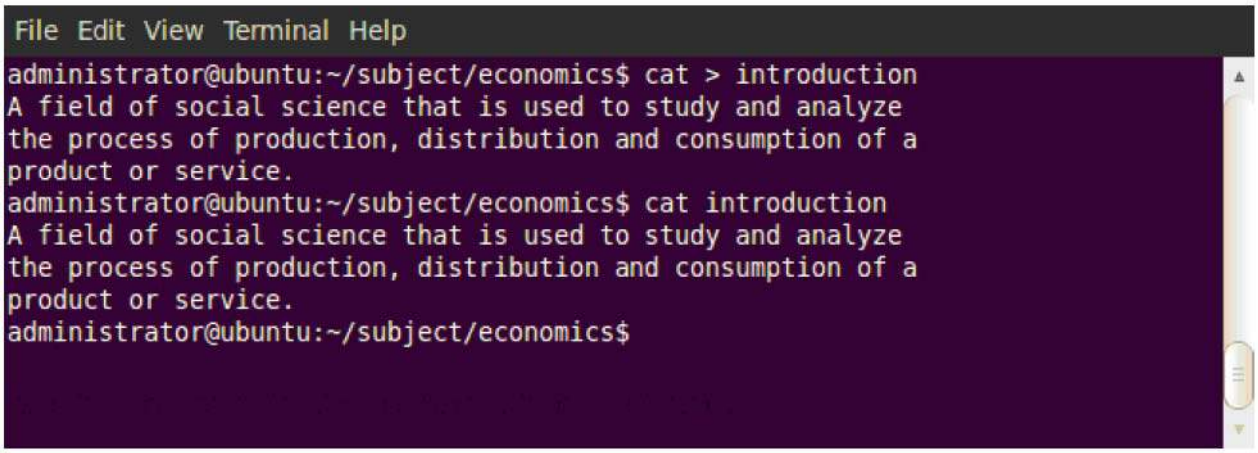


Figure 6.13 : Creating and displaying file using cat command

Appending contents using cat command

Assume that you have an existing file and you want to add some more content in the file. The cat command can be used again here with one simple change. The redirection operator used previously is to be replaced by append output (>>) redirection operator. The command to append data in the *introduction* file is shown below :

$cat >> introduction

An alternate definition states that Economics is a science which studies human behaviour as a relationship between ends and scarce means which have alternative uses.

[CTRL+d]

Note that if the file already exists and we use the command *cat > filename* then, the existing contents will be overwritten with the new one. So it is necessary to be careful while opening a file that already has some contents.

Concatenating multiple files using cat command

The cat command can also be used to concatenate the contents of multiple files and store it in another file. The syntax of using the concatenation is shown below :

$cat file1 file2 > file3

The above command will create *file3* that contains the text of both the files, namely file1 and file2. The new file created will have contents based on the sequence of filenames. Here the intial contents will be of file1, after which the contents of file2 will be appended.

Deleting a File (*rm*)

The *rm* command is used to delete/remove one or more files. For example to delete the file *introduction*, execute the following command :

$rm introduction

We can also delete multiple files using a single rm command. For instance the command

$rm file1 file2 file3

will delete all the three files supplied to it as argument. Table 6.2 gives some options that can be used along with the rm command.

Option	Usage
rm -i Filename	Deletes file using interactive mode. The user will be asked to verify the delete operation.
rm -r Directoryname	Deletes directory and along with all its contents.
rm -r *	Deletes all contents (file and/or directory) within the current directory. Here the symbol * is known as wildcard character. This is a very dangerous command as it will delete all the files and directories within the current directory, hence such command should be used only if you certain about the action and the result.
rm -rf *	Same as rm -r, but also deletes the contents even if it is write protected.

Table 6.2: Some options of rm command

Moving around the File system

So far we have learned how to create and delete a directory or file. Now let us see how to view the contents that are part of our file system. The *ls* command gives us the list of the contents in the current or a specified directory. The ls command can be used with different options to change the output. Let us begin with the plain and simple ls command without any options. Figure 6.14 shows the output of simple ls command. Note that the output on your computer may vary.

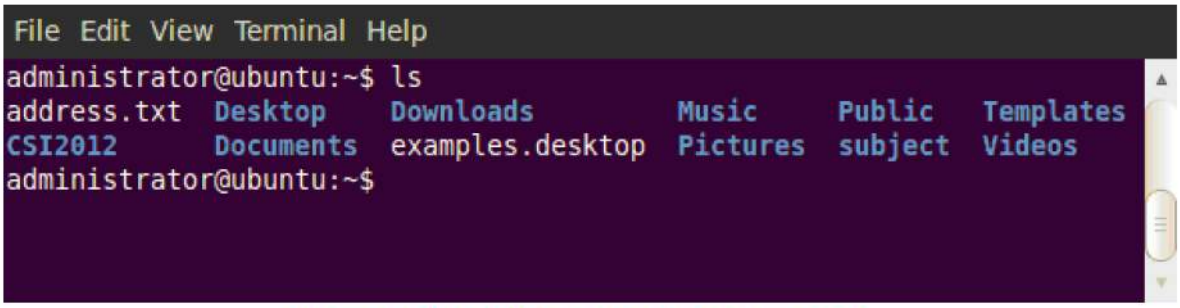


Figure 6.14 : Output of the ls command

Let us now create a file named *.introduction* and then check whether we are able to list it or not. Type the command shown below to create the file.

$cat > .introduction

Learning Ubuntu Linux is fun….

[CTRL + d]

Now use the ls command again to list the contents of a file. You must have observed that the file recently created is not visible on the screen. Note that in Linux any filename that is preceded by a '.' is treated as a hidden file. To list hidden files in the current directory we need to use -a option of the ls command. Figure 6.15 shows how to list hidden files.

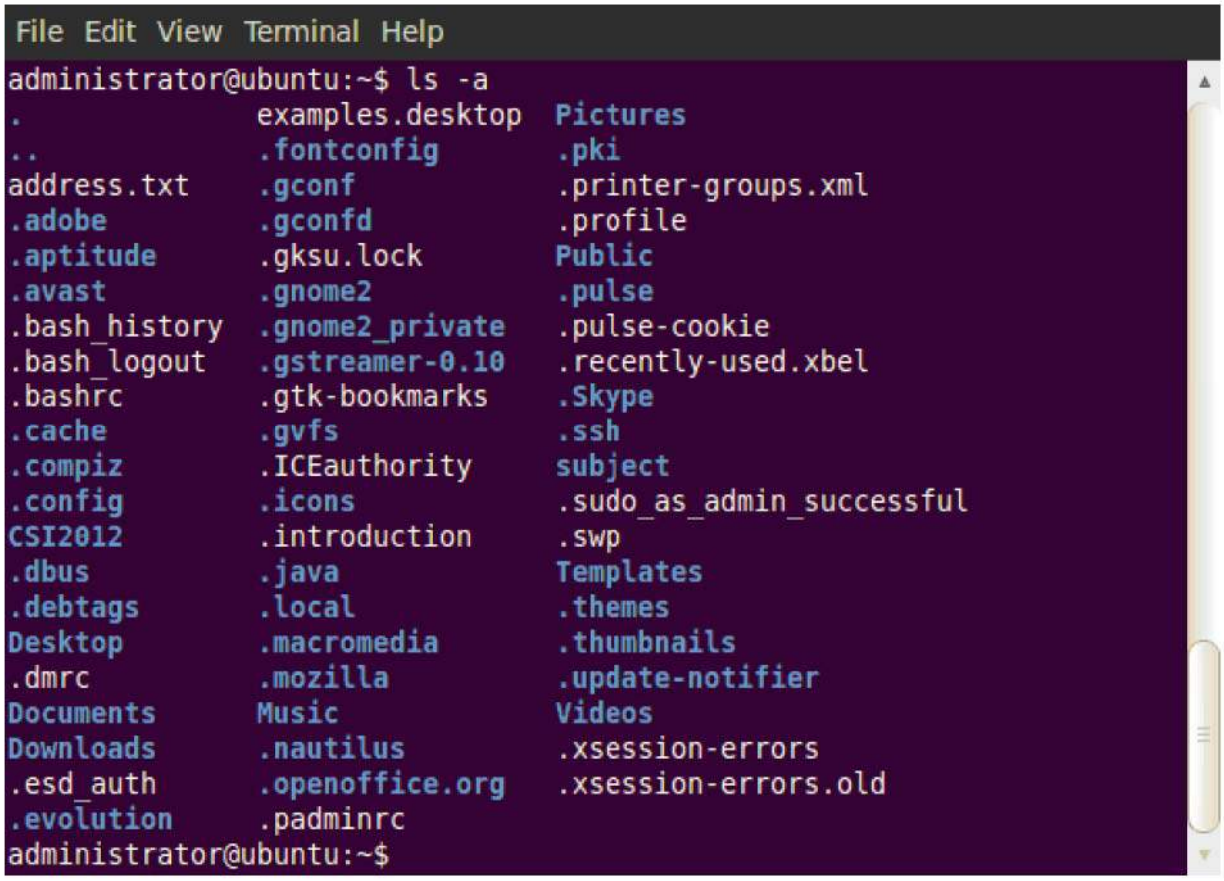


Figure 6.15: Listing hidden files

Observe the difference in the output of figure 6.14 and figure 6.15. Note that figure 6.15 shows more number of files than shown in figure 6.14. The two entries '.' and '..' visible in figure 6.15 are of importance, these two entries are automatically created in the directory whenever the directory is created. Table 6.3 gives some options that can be used along with the ls command.

Option	Usage
ls ~	Lists the files that are in user's home directory.
ls [svics]*	Lists all the files in which the first character of the filename matches with any of the given alphabets within the square brackets. The remaining part of filename can contain and valid ASCII character.
ls [n-s][5-7]??	Lists all files with 4 character filename. With the condition that the first character is in the range n to s, second character is in the range of 5 to 7, whereas the third and fourth characters are any valid ASCII character.
ls -r	List the files by sorting them in reverse order.
ls -t	List the files by sorting them based on their modification time.
ls -F	List the files and mark all executable files with * and directories with / symbol.
ls -1	List one file per line.

Table 6.3: Some options of ls command

Pattern Matching – The wildcards

In the above discussion you have already seen the usage of characters asterisk (*) and question mark (?). These characters are known as wildcard characters used for matching a pattern as required by the user. Table 6.4 summarizes the working of wildcards used by shell.

Wildcard	Pattern to be matched
*	Any number of characters including none
?	A single character
[abc]	A single character – either a, b or c (user can use other characters also).
[!abc]	A single character *other than* a, b or c (user can use other characters also).
[p-s]	A single character within the ASCII range of the characters p to s (user can use other characters also).
[!p-s]	A single character that is not within the ASCII range of the characters p to s (user can use other characters also).

Table 6.4: The wildcard characters

Manipulating Files and Directories

In the previous section we learned how to create and delete a file or a directory. Let us now see how to perform operations like copy, move, and assign permission on them.

Copying a file (*cp*)

Very often we need to create a replica of the data that we have generated, the *cp* command copies a file or group of files specified as an argument to it. It creates an exact replica of a file on the disk at the location specified by the user. The cp command needs at least two arguments. The first argument refers to a source file while the second argument refers to a destination file. Let us create a copy of file *introduction* using the following command :

$cp introduction new_introduction

After execution of the above command, an exact copy of file *introduction* will be created with the name *new_introduction*. If a file with the name *new_introduction* already exists, it will simply be overwritten without any warning from the system. In case no such file exists a new file will be first created and then the contents of the file *introduction* will be copied in it.

The cp command can also be used to copy more than one file into a specified directory. For instance, the following command :

$cp file1 file2 my_dir

will copy two files named *file1* and *file2* in a directory named *my_dir*. It is necessary that the directory *my_dir* already exists, or else we will get an error message. Table 6.5 gives some example usage of the cp command.

Command	Description
cp /vol/examples/tutorial/science.txt .	Copies the file *science.txt* to current directory. The dot (.) at the end refers to the current directory.
cp chap01 progs/unit1	A file named chap01 is copied within the directory progs with name unit1. (no directory with name unti1 should exist in progs directory)
cp chap01 progs	A file named chap01 is copied within the directory progs with same name. (because progs is a directory).
cp -r progs newprogs	The directory named progs along with all its contents is copied and stored as a directory newprogs.

Table 6.5 : Sample cp commands

Renaming files and/or moving files (*mv*)

Changing the name of a file or directory is another operation that user performs regularly. The *mv* command is used for renaming a file or directory. For example, to rename the file *introduction* to *introduction.txt*, execute the following command :

$mv introduction introduction.txt

The command will rename the file and store it at the same location as that of the file *introduction*. Thus no additional space is consumed on disk during renaming.

The *mv* command can also be used to move a file or group of files to a different directory. For example, the command

$mv file1 file2 my_dir

will move the files named *file1*and *file2* to the directory named *my_dir*.

This command is also used to rename a directory. For instance, the execution of the following command :

$mv math mathematics

will rename an existing directory *math* to *mathematics*.

Paging output (*more*)

The *more* command is used to view one page of content on the screen at a time. For instance if the file introduction.txt contains text that can not fit in a single screen, then reading its contents would become difficult. The *more* command when used displays the contents of file one page at a time. To view the next page we may press any key. Generally we may press character 'b' to view previous page and character 'f' to view next page. The sample usage of the command is shown below :

$more introduction.txt

Compare two files (*cmp*)

The *cmp* command compares two files of any type and writes the results to the standard output. If the two files compared differ in contents then the byte and line number at which the first dissimilarity occurred is reported. In case there is no difference between the contents of the files we simply see the command prompt again. The sample usage of the command is shown below :

$cmp introduction introduction.txt

Difference (*Diff*)

An extension of the *cmp* command is the *diff* command. The diff command compares two files and displays the contents of both files indicating where the difference lies. To understand the working of the *diff* command we have created a copy of the file introduction.txt and named it new_intro.txt. We have also removed some lines from the new file. Figure 6.16 shows the output of the command shown below :

$diff introduction.txt new_intro.txt

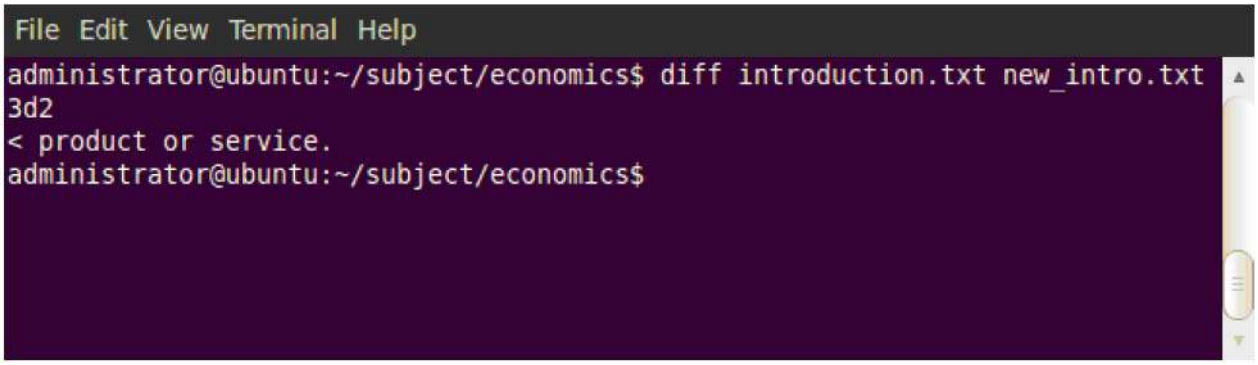


Figure 6.16 : Output of diff command

In figure 6.16 the lines beginning with a < indicates that file introduction.txt contains the text shown but file new_intro.txt does not contain the line. Any changes in the file new_intro.txt would be shown with the lines beginning with a > sign.

Counting lines, words and characters in a file (*wc*)

The *wc* command is used to count the number of lines, words, and characters in the specified file or files. The *wc* command can be used along with three options -l, -w and –c for counting lines, words and characters respectively. For instance, execute the following command to count the number of lines in the file introduction.txt.

$wc -l introduction.txt

4 introduction.txt

Similarly the commands **wc –w introduction.txt** and **wc –c introduction.txt** will give us the count of number of words and characters in the file. To get all the information together we can use the command as shown below :

$wc -l -w -c introduction.txt

4 49 307 introduction.txt

File permissions

In the earlier section, we have seen options which can be used with ls command. The ls command has several other options also. For example the following command when executed may result in the output similar to the one shown here.

$ls -l

total 6

-rw-r--r– 1 administrator administrator 313 2013-02-15 18 :04 about_Gandhiji.txt

-rw-r--r– 1 administrator administrator 444 2013-02-15 18 :19 introduction.txt

-rw-r--r– 1 administrator administrator 401 2013-02-20 16 :43 address.txt

drwxr-xr-x 1 administrator administrator 4096 2013-02-21 18 :15 backup

-rw-r--r– 1 administrator administrator 144 2013-02-13 18 :49 city.txt

-rw-r--r– 1 administrator administrator 226 2013-02-20 14 :11 script10.sh

Observe the output; it gives us a clear idea about an object in our file system. An object can be categorized as a regular file, a directory or a process. It shows us the owner of the object, size of the object, date and time on which the object was created along with the name of the object. Let us try to understand the file permission in detail. Figure 6.17 shows the relation of different permissions w.r.t. owner, group and other users.

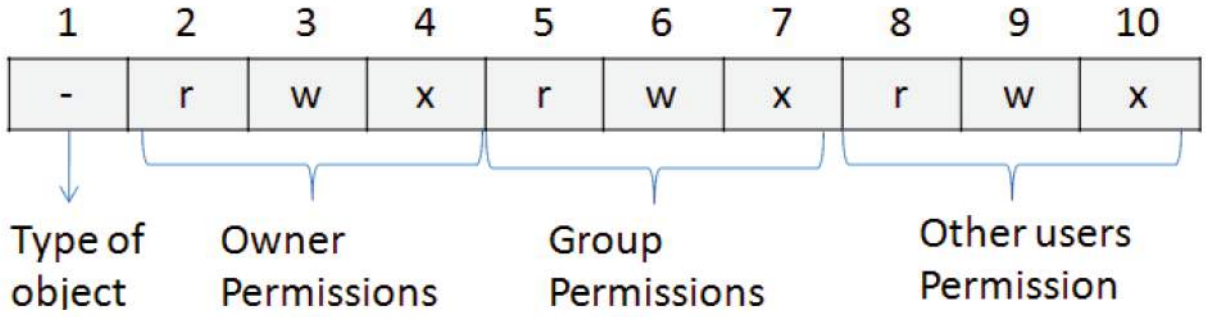


Figure 6.17 : File permissions

As can be observed in figure 6.17 the first column refers to type of object. The character '–' in the first column refers to a file, character 'd' refers to a directory and character 'p' refers to a process. The next nine characters tell the system what access is permitted for this object; hence the name "permissions". An object in Linux has three permissions namely, read (r), write (w) and execute (x).

The set of three characters after the file type shown in column 2 to 4 tells which permission the owner of the file has. Owner of the file is the user who created the file (administrator in our case). The character **r** in the first position means you are permitted to read the file. A **w** in the second position means you may write in the file. This includes the ability to delete a file. An **x** in the third position means you may execute the file. A hyphen '–' in any position means that you don't have that particular permission. As you can in the output of the ls **-l** command, administrator the user who owns the file can read and write files like about_Gandhiji.txt, introduction.txt and address.txt. The characters in column 5 to 7 denotes permission given to the group, to which user belongs. Similarly the characters in column 8 to 10 denote permission given to any other user or group. Generally the term others refer to the users of the system that do not belong to group to which the owner of the file belongs. A user if wishes can change the permission of an object that he/ she owns.

The file permissions can also be given as numeric representation. We use octal (base of 8) number system for representing permissions as numeric values. Every octal digit combines read, write and execute permissions together. For example, in permission 644, "6" refers to the rights of the owner, "4" refers to rights of the group and "4" refers to rights of others. The permission 0644 when assigned is interpreted as read and write permission to owner, only read permission to group and others. Table 6.6 shows the interpretation of the octal numbers when used as permission.

Permission in text mode	Permission in octal mode	Meaning
---	0	No permission assigned
--x	1	Only execute access is allowed
-w-	2	Only write access is allowed
-wx	3	Write and execute access are allowed
r--	4	Only read access is allowed
r-x	5	Read and execute access are allowed
rw-	6	Read and write access are allowed
rwx	7	Everything is allowed

Table 6.6 : Octal numbers and permission

Changing Permissions (*chmod*)

To change the permission we use the *chmod* command. The operation of changing the permission is also known as change mode operation. For instance, in the above example we have seen that user (owner) has read and write permissions on the file. To make the file read only file the following command can be used :

$chmod ugo-w introduction.txt

The character 'u' in the above chmod command stands for user, 'g' for group and 'o' for other. After executing the command if we again list the file, then the output will be similar to the one shown below.

$ls -l introduction.txt

-r--r--r-- 1 administrator administrator 307 2013-02-11 14 :19 introduction.txt

Please note that write operation will not be permitted on this file. Additionally it also prevents user from deleting the file intentionally or unintentionally.

To assign write and execute permission to the owner of a file, execute the following command.

$chmod u+wx script10.sh

$ls -l script10.sh

-rwxr--r-- 1 administrator administrator 226 2013-02-20 16 :05 script10.sh

Here the file script10.sh is known as a script file. We are going to learn about shell scripting in the next chapters.

Table 6.7 shows some abbreviations and its meaning when used with the chmod command.

Category	Operation	Permission
u-user	+ assign permission	r- read permission
g-group	– remove permission	w - write permission
o-other	= assign absolute permission	x - execute permission
a-all		

Table 6.7 : Abbreviation used by chmod

I/O Redirection

A user interacts with the Operating System using a standard input device (keyboard). The Operating system displays the output on standard output device (monitor). Thus if any command is executed its input will be taken from the keyboard and output will be displayed on the monitor.

Sometimes it is useful to redirect the input or output to a file or a printer. Linux provides redirection symbols to change the standard input flow. The greater than symbol ' > ' implies redirection of output. It instructs the OS to put the output in the destination (file) specified by the user instead of displaying on the monitor screen. Similarly the less than symbol ' < ' implies redirection of input, it instructs the OS to accept the input from the specified source (file) instead of keyboard.

Assume that we issue a command **wc –l < introduction.txt**, here we are instructing OS to accept the input from a file named introduction.txt instead of the keyboard. Similarly the command **ls > list.txt** when executed will transfer the output of ls command to a file named list.txt instead of the monitor. When output redirection is used the output will not be displayed on the monitor hence to see the output we will have to use the command **cat list.txt**.

Piping

Redirection facility discussed above helps in associating the Linux commands to files. Many times we need to use multiple commands to perform a single operation. The piping facility of Linux helps in such cases. The pipe symbol (|) is used to provide the output of one command as an input to another command. The process of converting output of one command into input of another command is known as piping. Let us see an example of piping.

$ls | wc -l

When we execute the above command, the output of *ls* command becomes the input to the *wc* command. Thus we will get information of total number of files in a current directory. Real power of pipe facility can be availed when we use it along with filters. Filters have been discussed in the next section.

Filters

Filters are commands that accept data from the standard input, process or manipulate it and then write the results to the standard output. Various filters like head, tail, cut, paste, sort and uniq are available in Ubuntu Linux. Let us see the working of these filters.

Displaying lines from top of the file (*head*)

The head command is used to display the required number of lines at the beginning of the file based on user's requirement. When used without any option it displays first 10 lines of the file. To display the lines as per users requirement we need to pass an argument to the head command. For example to display first 2 lines of the file *introduction.txt*, execute the following command :

$head -2 introduction.txt

Displaying lines from bottom of the file (*tail*)

The tail command works exactly opposite of the head command. It displays specified number of lines from the end of the file. To display last 2 lines of the file *introduction.txt*, execute the following command :

$tail -2 introduction.txt

We can use the tail command to display lines from n^{th} line within the files. For example if we execute the following command :

$tail +5 introduction.txt will display lines from 5^{th} line onwards from the file introduction.txt.

Slicing a file vertically (*cut*)

The head and tail command discussed in the above sections are used to slice the file horizontally. We can slice the data within the file vertically using the *cut* command. The cut command gives exact and precise outputs if the file has specific delimiters. Let us create one such delimited file to understand the working of the cut command. Create a file named address.txt using cat command that stores the data as shown :

$cat address.txt

20013, Vaidehi, Sanjay, Shah, Sector-23, GH-6, Gandhinagar, 382023

20014, Dhrumil, Ajay, Patel, Yesh Enclave, Mota Bazar, Vidyanagar, 388120

20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058

20016, Abdul, Shamsher, Khan, Khan Villa, M G Road, Nadiad, 388011

20017, Nirav, Jose, Mackwan, Jose House, M G Road, Nadiad, 388011

20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058

Let us see how to use the cut command along with its various options.

Cutting Characters (-c)

To extract specific characters from each line of the file, cut command with -c option is used. For instance, to extract the roll numbers and first names from the file address.txt execute the following command :

$cut -c 1-15 address.txt
20013, Vaidehi,
20014, Dhrumil,
20015, Harshit,
20016, Abdul, S
20017, Nirav, J
20018, Vidita,

Though the output looks fine to certain extent, it is not exactly the same as we would have expected. Look at the data of Abdul and Nirav it has additional characters. The -c option is useful for fixed length fields, we had problem in the output as the first names were not stored using a fixed length.

Cutting fields (-f)

To overcome the problem mentioned in the -c option we may use a delimiter. The cut command can treat values separated by the delimiter as separate field values. Observe that we have used comma ',' as a delimiter in the address.txt file. Thus, to extract roll number and first name only we may execute the following command :

$cut -d "," -f 1,2 address.txt
20013, Vaidehi
20014, Dhrumil
20015, Harshit
20016, Abdul
20017, Nirav
20018, Vidita

Observe that we have got the desired output now. In this command the –d is option used to specify delimiter appearing in a file (comma in our case) and –f option is used to specify field numbers to be displayed (roll number (1) and first name (2) in our case).

It is also possible to slice the file vertically from fields in between. For example assume that we want to display the first name, city and pin-code then we need to cut field numbered 2 and 7 onwards. The said operation can be performed by executing the following command :

$cut -d "," -f 2,7- address.txt
Vaidehi, Gandhinagar, 382023
Dhrumil, Vidyanagar, 388120
Harshit, Ahmedabad, 380058
Abdul, Nadiad, 388011
Nirav, Nadiad, 388011
Vidita, Ahmedabad, 380058

Here 7- in the command imply that we need to display all fields after field number seven (including seven) from the file address.txt. It is also possible to redirect the output to a file. For example if we execute the following command :

$cut -d "," -f 2,7- address.txt > output_cut.txt

the output instead of being displayed on monitor will be transferred to the file output_cut.txt.

Joining Contents (*paste*)

Two files can be pasted together using the *paste* command. For the paste command to work properly we need to ensure that both the files have exactly the same number of lines. If the number of lines is not same then the command may not result into expected output as it pastes from top of the files.

We will first create two different files named part1.txt and part2.txt using cut command and then join them using the paste command. Figure 6.18 shows the process of performing this operation.

```
File  Edit  View  Terminal  Help
administrator@ubuntu:~$ cut -d "," -f 1-4 address.txt > part1.txt
administrator@ubuntu:~$ cut -d "," -f 5-8 address.txt > part2.txt
administrator@ubuntu:~$ cat part1.txt
20013, Vaidehi, Sanjay, Shah
20014, Dhrumil, Ajay, Patel
20015, Harshit, Amit, Jain
20016, Abdul, Shamsher, Khan
20017, Nirav, Jose, Mackwan
20018, Vidita, Harshal, Arolkar

administrator@ubuntu:~$ cat part2.txt
 Sector-23, GH-6, Gandhinagar, 382023
 Yesh Enclave, Mota Bazar, Vidyanagar, 388120
 58, Jaldeep I, Ahmedabad, 380058
 Khan Villa, M G Road, Nadiad, 388011
Jose House, M G Road, Nadiad, 388011
 17, Jaldeep I, Ahmedabad, 380058

administrator@ubuntu:~$ paste part1.txt part2.txt
20013, Vaidehi, Sanjay, Shah     Sector-23, GH-6, Gandhinagar, 382023
20014, Dhrumil, Ajay, Patel      Yesh Enclave, Mota Bazar, Vidyanagar, 388120
20015, Harshit, Amit, Jain       58, Jaldeep I, Ahmedabad, 380058
20016, Abdul, Shamsher, Khan     Khan Villa, M G Road, Nadiad, 388011
20017, Nirav, Jose, Mackwan     Jose House, M G Road, Nadiad, 388011
20018, Vidita, Harshal, Arolkar  17, Jaldeep I, Ahmedabad, 380058

administrator@ubuntu:~$
```

Figure 6.18 : Example of paste command

Ordering Output (*sort*)

The *sort* command is used to order the data stored within a file in ascending or descending sequence at the time of display. Like the cut command, it also identifies fields and can sort on specified fields. When the sort command is used without any options, it sorts the file based on entire line. It reorders the lines based on ASCII sequence. The sorting is first applied on white spaces, followed by numerals, uppercase letters and finally lowercase letters.

Let us try to arrange the file address.txt in descending order of roll numbers (you must have observed that it is already arranged in ascending order). To display contents of file sorted in reverse order, execute the following command :

$sort -r address.txt

20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058

20017, Nirav, Jose, Mackwan,Jose House, M G Road, Nadiad, 388011

20016, Abdul, Shamsher, Khan, Khan Villa, M G Road, Nadiad, 388011

20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058

20014, Dhrumil, Ajay, Patel, Yesh Enclave, Mota Bazar, Vidyanagar, 388120

20013, Vaidehi, Sanjay, Shah, Sector-23, GH-6, Gandhinagar, 382023

Note :

The execution of sort command does not modify the actual file. The records in the actual file will remain in the same position. The sort order is applied only at the time of displaying the output.

The sort command is mostly used in conjunction with other commands. For example, we can use it with the cut command to order the output of the cut command. Execute the following command to see the output of both the cut and sort commands when combined.

$cut -d "," -f 2-4 address.txt | sort

Abdul, Shamsher, Khan

Dhrumil, Ajay, Patel

Harshit, Amit, Jain

Nirav, Jose, Mackwan

Vaidehi, Sanjay, Shah

Vidita, Harshal, Arolkar

The cut command in the above example extracts second, third and fourth column from the file address.txt. The extracted output is then given as input to the sort command. The sort command then sorts the contents and displays it on the screen.

Character Conversion (*tr*)

The command used as filters work with a line or column. The *tr* (translate) command allows us to work with individual characters within a line. It is used to translate (convert) strings or patterns from one set of characters to another.

Working with the address.txt file you must have observed that it contains "," as delimiter. Assume that we do not want to show the delimiter at the time of the display, instead we would like to show a blank space. The tr command allows us to perform this operation. Execute the command shown below :

$cat address.txt | tr -s '[,]' '[ ]'

20013 Vaidehi Sanjay Shah Sector-23 GH-6 Gandhinagar 382023

20014 Dhrumil Ajay Patel Yesh Enclave Mota Bazar Vidyanagar 388120

20015 Harshit Amit Jain 58 Jaldeep I Ahmedabad 380058

20016 Abdul Shamsher Khan Khan Villa M G Road Nadiad 388011

20017 Nirav Jose Mackwan Jose House M G Road Nadiad 388011

20018 Vidita Harshal Arolkar 17 Jaldeep I Ahmedabad 380058

The translation is only applicable at the display it will not permanently replace the actual delimiter. The -s option squeezes the additional space visible in the actual file. In case we need to save the translations visible we can redirect the output to a new file.

Pattern matching (*grep*)

The find operation is one of the most widely used operation in GUI applications. We must have used the CTRL + f keys to find a keyword within files. The *grep* command performs similar operation from the command line interface. The command is based on a fundamental idea of search globally for a regular expression and display lines where instances are found (g/re/p).

Let us make use of the grep command to find a name within the file address.txt and display its record. Figure 6.19 shows the working of the grep command.

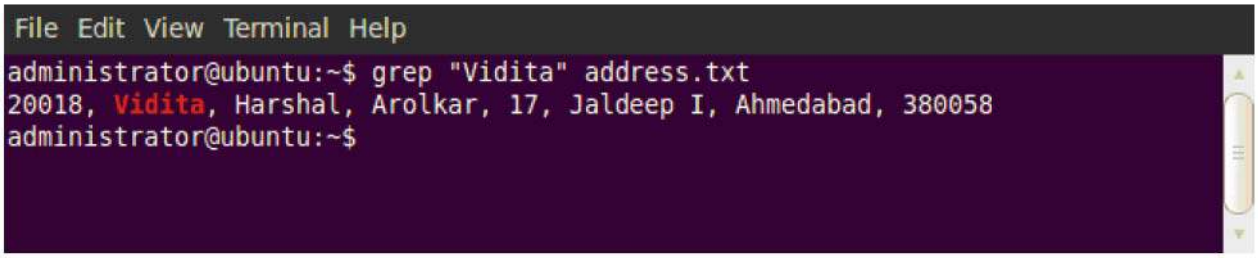


Figure 6.19 : Working of grep command

Observe that the string that we are looking for is shown in red colour, also we have enclosed the keyword in double quote. It is not compulsory to enclose the keyword in double quotes hence the command **grep Vidita address.txt** will also give same output. The string used in the grep command is case sensitive hence the strings "Vidita" and "vidita" are different. We can use different options along with grep command that will help us refine our search in a better way. Table 6.8 lists the options and their usage.

Option	Usage
-c	Return only the number of matches, without quoting the text
-i	Ignore case while searching
-l	Return only file names containing a match, without quoting the text.
-n	Return the line number of matched text, as well as the text itself.
-v	Returns all the lines that do not match the text.
-w	Return lines which display only whole words
-o	Shows only the matched string

Table 6.8: Options of grep command

One very powerful feature of grep command is to use a regular expression as a keyword. Say for example we want details about the persons who are staying in society that has 'Jal' as it's starting and 'I' as its end, then regular expression can be used. Let us execute the command shown here :

$grep "Jal.*I" address.txt

The output of this command is shown in figure 6.20. Here "Jal.*I" is a regular expression.

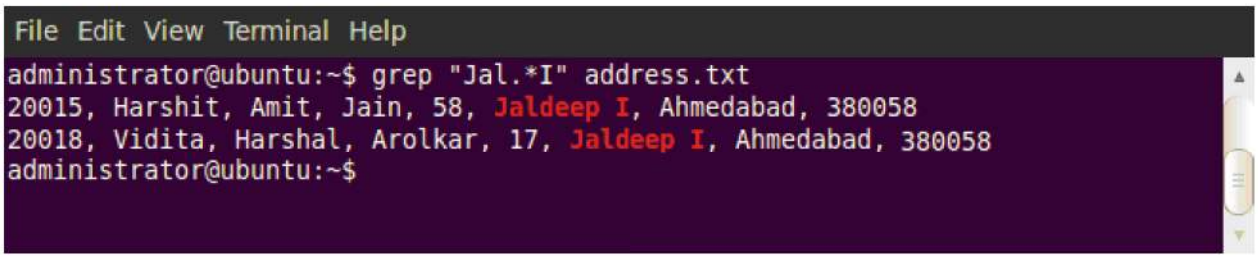


Figure 6.20 : Using regular expressions in grep command

A regular expression is normally followed by one of several repetition operators shown in table 6.9.

Repetition Operator	Meaning
?	The preceding item is optional and matched at most once.
*	The preceding item will be matched zero or more times.
+	The preceding item will be matched one or more times.
{n}	The preceding item is matched exactly n number of times.
{n,}	The preceding item is matched n or more number of times.
{,m}	The preceding item is matched at most m number of times.
{n,m}	The preceding item is matched at least n number of times, but not more than m number of times.

Table 6.9 : Repetition operator

Searching a file or Directory (*find*)

Many times we forget the location of the file or a directory that we have created. The *find* command helps us look for such forgotten objects. The find command looks for the search criteria (file or directory or both) that you have specified starting from the directory you specify within all its sub directories. We can also search for the object based on its name, owner, group, type, permissions, date, and other criteria. Note that the find command when used without any other arguments displays the pathname of all the files and directories in the present directory and all its subdirectories.

Assume that we want to look for the location of file introduction.txt that we had created earlier, the command would be

$find -name introduction.txt

If the file exists then its path will be given as an output. Otherwise we may either get an error or a prompt will be visible if no such file exists. Note that we know the name of the file here, what if we only remembered first few characters of the file name. The wildcard characters can be used in such cases. The example shown below helps in finding all files that start with string "intro".

$find -name intro*
./subject/economics/my1_dir/intro1
./subject/economics/my1_dir/intro
./subject/economics/introduction.txt
./subject/economics/introduction
./subject/economics/my_dir/intro1
./subject/economics/my_dir/intro
./subject/economics/intro1

The output of this command may vary on your screens depending on the number of files that you have which start with string "intro". Also in Linux we may not assign a file extension, hence we may not be able to know whether intro1, intro and introduction in the above outputs are files or directories. We may refine the search if needed by using the *type* option as shown below :

$find -name intro* -type f

Table 6.10 shows some example of find and its expected output description.

Command	Description
find / -type d	Search all directory and sub directory available on root only.
find . -mtime -1	Search objects modified within the past 24 hours.
find . -mtime +1	Search objects modified more than 48 hours ago.
find ./dir1 ./dir2 -name script.sh	Search directories "./dir1" and "./dir2" for a file "script.sh".
find -size 0 -delete	Search for files of zero byte and delete them from the disk.
find -executable	Search for the executable file in current directory.
find /home -user jagat	Search for the object whose owner is jagat within the home directory and its sub directory.
find . -perm 664	Search for object that has read and write authorization for their owner, and group but which other users can only read.

Table 6.10: Example of find command

Running Commands as the Superuser

When you log in to your computer, the account you use is a regular user account. This account has a limited right. The security model of Ubuntu generally allows you to work as a normal user. Not providing administrative rights prevents any accidental changes or installation of malicious programs that may disturb functioning of the system. But many times the user may need to have administrative privileges. The administrative privileges are available to only a user known as superuser. To use the superuser account when using the terminal, we need to add *sudo* as a prefix to the commands that we want to execute. For example, execute the following command to install a new program called skype from command line.

$sudo apt-get install skype

When you execute the command it will ask for password, provide the password of superuser (generally it will be different from the normal user). This password is the password of the first user that you

added when you installed Ubuntu Linux on the computer. Once the user is authenticated as sudo by means of the terminal, the software will start installing. Once the installation is over we can start using the software.

The super user can also perform the operations of adding, deleting or updating a user, group or object in the system. Some of the commands listed below are used for such purposes.

adduser : The *adduser* command creates a new user on the system. Figure 6.20 shows the process of creating a user. The command when executed will ask for password and some additional details as shown in figure 6.21. Once all the details are provided a new user along with its home directory will be created in the system.

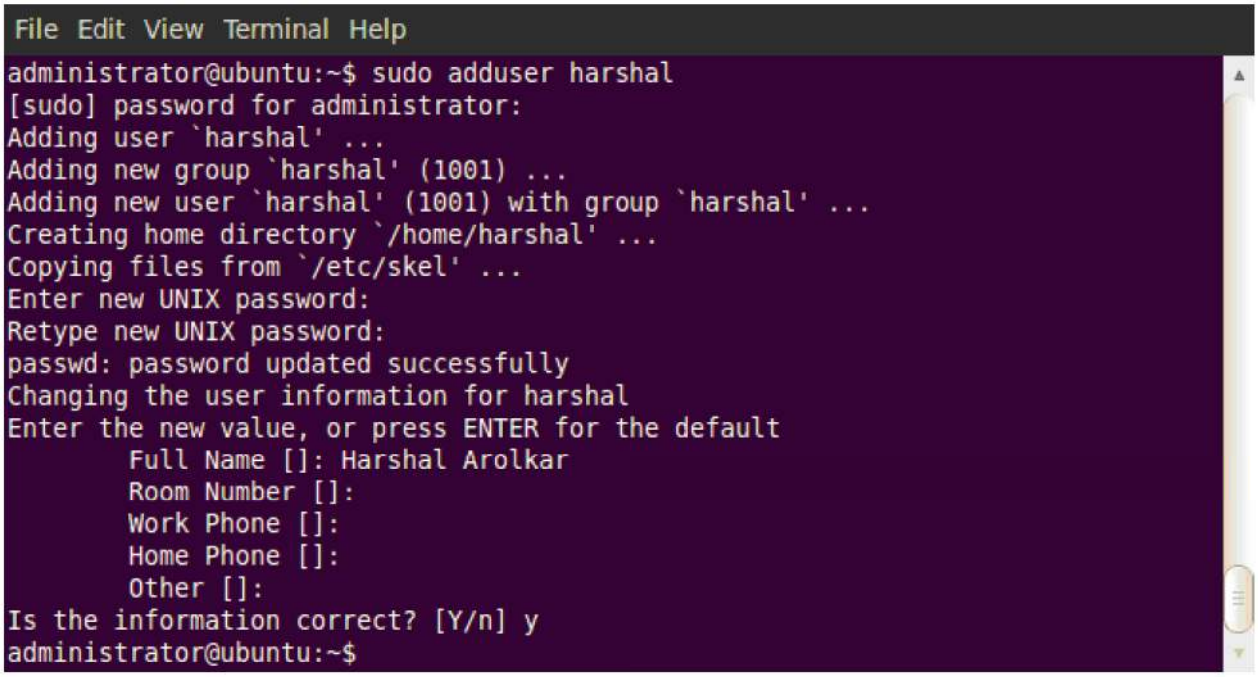


Figure 6.21 : Adding a user

passwd : The *passwd* command when executed as a super user do, allows us to change the password of any valid user of the system.

who : The *who* command when executed displays the list of all the users that are presently logged into the machine.

addgroup : The *addgroup* command adds a new group. The users are normally divided into groups so that they can be better controlled.

deluser : The *deluser* command is used to delete a user from the system. Note that we need to explicitly remove the user's files and home directory, by using the -remove -home option.

delgroup : The *delgroup* command deletes a group from the system. To perform this operation we must first make sure that no user is associated with the group that we are going to delete.

Summary

In this chapter we learned how to use the Ubuntu Linux command line interface. The CLI (Command Line Interface) when used allows us to perform all operations that we perform using the GUI in efficient and fast manner. We saw how to initiate the CLI using the Linux terminal. We also learned how to create, rename and delete a file or a directory, find out the directory that we are working in, change the directory if required. Later we saw how to create a copy of the file as well as directory. An access right is one of the ways to make sure that our data is not misused; we saw how to assign or change access rights. We saw how to increase the effectiveness of commands by joining multiple commands using the pipes. Further we looked at some features like counting the words or lines within the file, slicing the file horizontally and vertically, joining the file, searching for a file, directory or a string pattern within the files, arranging the display in ascending or descending order. Finally we saw how normal user can perform administrative tasks of installing new software, adding or deleting a user or group, checking who is using the system or change the password of some user.

EXERCISE

1. What is command prompt?

2. Describe shell. Name any three Linux shells.

3. How shells interpret command? Explain with suitable figure.

4. Write the steps used to start a terminal in Ubuntu Linux.

5. Explain following command in detail :

ls, cat, wc, chmod

6. What is the meaning of internal commands in Ubuntu Linux?

7. How can you find help for any command in Ubuntu Linux?

8. Explain different wildcard characters, giving suitable example.

9. Explain the use of pipes, giving suitable example.

10. What is redirection? Explain giving suitable example.

11. List the filter commands used in Linux.

12. **Choose the most appropriate option from those given below :**

(1) Which of the following command is used to count the total number of lines, words, and characters contained in a file?

(a) countw (b) wcount

(c) wc (d) wordcount

(2) Which of the following command is used to remove files?

(a) dm (b) rm

(c) delete (d) erase

(3) Which of the following command is used to remove the directory?

(a) rdir (b) remove

(c) rd (d) rmdir

(4) Which of the following command is used to count just the number of lines contained in a file?

(a) wc - r (b) wc - w

(c) wc - c (d) wc - l

(5) The command chmod 761 letter is equivalent to which of the following access rights?

(a) chmod u=7, g = 6, o = 1

(b) chmod a = 761

(c) schmod u = rws, g = rw, o = x

(d) chmod 167

(6) Which of the following refers to the maximum length of a filename in Linux?

(a) 8 (b) 10

(c) 200 (d) 255

(7) The hierarchy of a series of directories branching in a user system starts from which of the following directories?

(a) \home (b) \root

(c) /home (d) /root

(8) Which of the following command is used to copy a file?

(a) tar (b) cpio

(c) cp (d) copy

(9) Which of the following command is used to display your current working directory?

(a) path (b) pwd

(c) prompt pg (d) dir

(10) Which of the following command is used for searching a pattern in a file?

(a) grep (b) find

(c) lookup (d) All of the above

(11) Which of the following is not a redirection symbol?

(a) > (b) <

(c) * (d) >>

(12) Which of the following syntax is correct to assign a read permission on a user file?

(a) chmod r filename
(b) chmod u+r filename
(c) chmod filename r
(d) chmod filename u+r

(13) Which of the following refers to the minimum arguments of cp command?

(a) One
(b) Two
(c) Three
(d) None

(14) The mv command in Linux is used for which of the following purpose?

(a) To rename a directory
(b) To move a file
(c) To copy a file
(d) All of the above

(15) Which of the following command is used to view one page content on the screen at a time?

(a) More
(b) more
(c) PAGE
(d) page

Laboratory Exercises

1. Perform the following using Linux commands :

(a) Print the calendar of December 2012.

(b) Execute the command which displays login name, the name of your terminal and date and time since user logged in.

(c) List all files starting with character 'n' or 'N'.

(d) Display the current working directory.

(e) Prepare two files named class11_A.txt and class11_B.txt containing details of students of eleventh standard. The file should contain names of the students. Now merge these two files in a single file and name it class11_.txt. (Use cat command).

(f) Hide the three files created in question 'e'.

(g) List only directories.

(h) Get help on the use of the cat command.

(i) List all files whose fourth character is 'g' and sixth character is digit.

(j) Using terminal calculator perform the following operation :

(1) Calculate 2500/7

(2) Convert decimal 50 to its binary equivalent

(3) Convert decimal 25 to its hexadecimal equivalent

(4) Find square root of 36

(5) Convert hexadecimal 25 to its decimal equivalent

2. Show the output of following Linux commands :

(a) cat f1 >> f2

(b) echo $SHELL

(c) mkdir d1 d2 d3

(d) ls s*t

(e) ls [a-f]*

(f) ls -R

(g) ls -xR

(h) cp f1 f2

(i) ls | sort

(j) ls | tr -s " " | cut -d " " -f 5 | sort

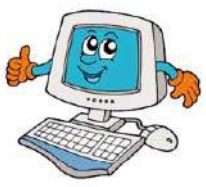
(k) ls -l | grep -c "address.txt"

(l) grep "Harshit Jain" address.txt

(m) chmod u-w address.txt

(n) wc -l address.txt > totalstudents

7 Vim Editor and Basic Scripting



In the previous chapter, we have discussed commands that can be used to work with Ubuntu Linux. The commands were executed one at a time. Though we could execute multiple commands by using the pipes, the process would become tedious as number of commands increases. A better way of executing multiple commands at one go is to type the sequence of commands in a text file. Then give this file to Linux shell for execution. The Linux shell will execute all the commands available within the text file in the specified sequence. This text file is known as shell script. A shell script can be defined as series of commands written in a plain text file. The shell scripts are commonly used by the users to perform routine individual tasks and system administration. In this chapter we will look at an editor that assists us in writing the shell script along with some sample shell scripts.

Working with Vim Editor

We have learned how to create a file using the cat command. The cat command although it allows us to create a file is not a good option to use when creating a shell script. We need a good text editor to perform such operations. Text editors like nano, pico, vi or Vim, ed and others are generally best suited for creating text file. Gedit is a graphical editor available with GNOME desktop environment. Kwrite is a graphical editor available with KDE desktop environment. We will use the Vim editor which is a visual display editor to write the shell script. This editor is available with almost all Unix and Linux flavors.

The Vim (Vi Improved) is a text editor written by Bram Moolenaar and first released publicly in 1991. It is an enhanced version of the vi editor distributed with most UNIX systems. Vim is a highly configurable text editor built to enable efficient text editing. The Vim editor can be used from both a command line interface and as a standalone application in a graphical user interface.

To work with the Vim editor we will have to initiate it first. Open a new Terminal Window. We can open the Vim editor using two ways, first type ***vi*** at the prompt and press Enter key or type ***vi*** followed by a file name and press Enter key. Figure 7.1 shows the Vim editor interface when we don't specify a file name.

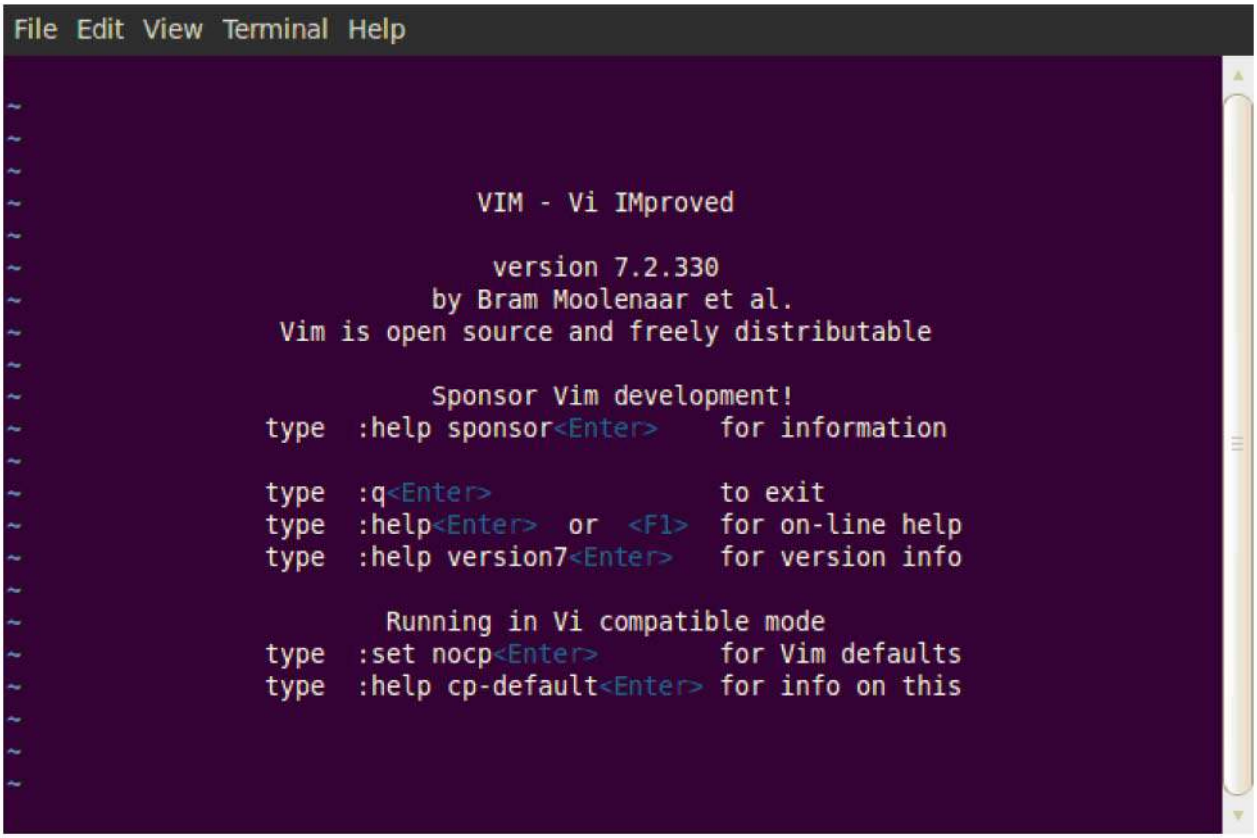


Figure 7.1 : Vim editor interface

It is a good option to start the Vim editor by specifying a file name. Type the command given below:

$vi first_vim_file

and press Enter key, the command when executed will open the editor as shown in figure 7.2.

Figure 7.2 : Creating file using Vim editor

Observe that the screen in figure 7.2 is filled with tildes (~) on the left side of the screen. The tilde (~) symbol indicates that the lines are yet to be used by the editor. Notice that the cursor appears in the top left corner of the screen and some text is visible on the last line. The last line is known as command line and it displays the name of the file along with the information about the total number of lines and columns within the file.

Vim Modes

The Vim editor functions in three different modes namely, a command mode, an insert mode and a last line mode.

The command mode

When we first start editing a file using the Vim editor, the editor will be opened in a command mode. We can issue many commands that allow us to insert, append, delete text, or search and navigate within our file. Note that when using command mode we can't insert text immediately. We first need to issue an insert (i), append (a), or open (o) command to insert the text in the file.

An extension to the command mode is a visual mode. Visual mode is a flexible and easy way to select a piece of text from the file. It is the only way to select a block of text that needs to be modified. Table 7.1 shows us the characters that assist us in the visual mode.

Command	Usage
v	Switch to the visual mode (allows us to manipulate characters)
V	Switch to the visual mode (allows us to manipulate lines)
CTRL + v	Switch to the block-visual mode (allows us to manipulate rectangular blocks of text)

Table 7.1 : Characters that are used in visual mode

The insert mode

When we issue an insert, append, or open command, we will be in the insert mode. The current text editors show us the current mode of operation. Once in an insert mode we can type text into our file or navigate within the file.

We can toggle between the command mode and the insert mode by pressing the ESC key. This operation will be performed often when using the Vim editor. Table 7.2 shows us the characters that assist us in the insert mode.

Command	Usage
a	To insert text after the current cursor position.
i	To insert text before the current cursor position.
A	To append text at the end of the current line.
I	To insert text from the beginning of a line.
O	To insert in a new line above the current cursor position.
o	To insert in a new line below the current cursor position.

Table 7.2: Characters that are used in insert mode

The last line mode

The last line mode normally is used to perform operations like quitting the Vim session or saving a file. To go to the last line mode we first need to be in the command mode. From command mode we can go to last line mode by pressing the colon (:) key. After pressing this key, we will see a colon character at the beginning of the last line of our editor window with a cursor blinking near it. This indicates that the editor is ready for accepting a "last line command".

It is possible to toggle back to the command mode from the last line mode by pressing the ESC key twice or by pressing the [Backspace] key until the initial ":" character is gone along with all the characters that we had typed or by simply pressing the ENTER key.

Creating a file in Vim

Let us now learn how to create a simple text file using the Vim editor. Execute the command given below to open the editor interface.

$vi about_Gandhiji

To enter text within the file the editor needs to be in the insert mode. By default, the editor will start in the command mode. As seen in table 7.2 there are several commands that put the editor into the insert mode. The most commonly used commands to get into insert mode are 'a' and 'i'.

Press 'i' and the editor will now be in the insert mode. Now type the contents as given in the box below.

> Mahatma Gandhi was born on 2nd October 1869 in Porbandar, Gujarat.
> His name was Mohandas Karamchand Gandhi.

Saving the file

Once we have written the contents shown we need to save this file. To save the file we need to switch to the last line mode from the insert mode. Press the ESC key and type colon (**:**), you will notice that colon is displayed in the bottom of the screen. Type ***wq*** (write and quit) as shown in figure 7.3 and press the Enter key. The Vim editor will now be closed and we will be able to see the shell prompt. To see the contents of the file we can use the cat command.

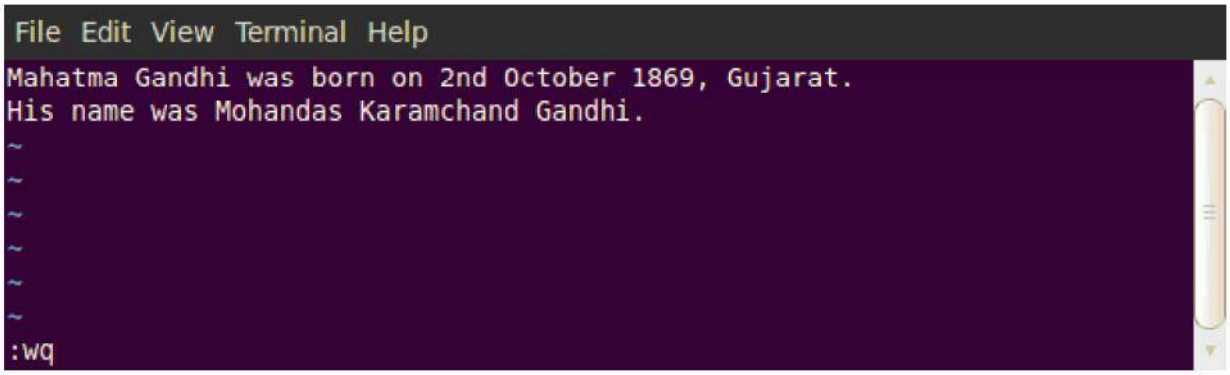


Figure 7.3: Saving a file and quitting the editor

There are several other commands available to save a file depending on the current status and usage. Table 7.3 shows commands that can be used in the last line mode along with their usage.

Command	Usage
:w	To save file and remain in editing mode
:wq	To save file and quit editing mode
x	To save file and quit editing mode (same as above)
:q	To quit editing mode when no changes are made
:q!	To quit editing mode without saving changes made in the file
:saveas FILENAME	To save existing file with new name and continue editing it under the new file name.

Table 7.3 : The last line mode commands to save the file

Note that if we open the Vim editor without typing file name initially the text will be directly stored in the system buffer (main memory). To transfer the contents from the buffer to a hard disk we need to type the file name along with the *wq* command.

Moving around in the document

You must have observed when creating the file about_Gandhiji the arrow keys (by any chance if you used them) were not working as per our expectations. Normally the arrow keys are used to move the cursor in up, down, left and right directions. In insert mode we cannot do anything except for typing the text. When using the Vim editor we need to use some special keystrokes to move within the document after going to the command mode. Once in command mode we can also use the arrow keys to move within the document. Table 7.4 lists the keystrokes that are used to navigate within the documents.

Command	Usage
h	Moves cursor left
l	Moves cursor right
j	Moves cursor down
k	Moves cursor up
Spacebar	Move cursor right one space

-/+ Keys	Move cursor down/up in first column
CTRL + d	Scroll down one half of a page
CTRL + u	Scroll up one half of a page
CTRL + f	Scroll forward one page
CTRL + b	Scroll back one page
M	Move the cursor middle of the page
H	Move the cursor to top of the page
L	Move the cursor to bottom of page
$	Move the cursor to end of line
)	Move the cursor to beginning of next sentence
(	Move the cursor to beginning of current sentence
G	Move the cursor to end of file
W	Move the cursor one word at a time
Nw	Move the cursor ahead by N number of words
B	Move the cursor back a word at a time.
b	Move the cursor back a word at a time.
Nb	Move the cursor back by N number of words
e	Move the cursor to end of word
gg	Move to first line of file
0	Move to the beginning of the line.

Table 7.4 : Keys to navigate in file

Try and work on these keystrokes to become familiar with them.

Editing the Document

Editing the document is one of the most common operations that a user would perform once the document is created. It is possible to insert or delete any data at a specific position as per our

needs. We can also replace contents or change the case of individual characters if required. A user needs to toggle between the command and the insert mode when we edit a document. The characters in table 7.2 showed us how to initiate different insert options. Let us try to add the contents given in the box below to the file about_Gandhiji.

> His mother, Putlibai, was a very religious lady and used to tell him stories from the scriptures and mythology.
>
> Little Gandhi grew up to be an honest and a decent student. At the age of 13 he was married to Kasturba.

To edit the document we need to again open it using the vi command, hence execute the command **vi about_Gandhiji** again. You will observe that the blinking cursor is visible on the first character at top left. In normal cases we would have used the 'I' option to enter the insert mode, but we need to append the contents at the end.

Type G and you will see that the cursor gets positioned at last line. The position of the cursor will depend upon how the file was initially saved. Typing G may place the cursor in a new line after the last line (case when Enter key was pressed before saving the file) or the cursor will be placed on the first character of the last line (case when Enter key was not pressed before saving the file).

If the cursor is placed at new line we press ESC and then 'I' and start typing the contents. In case we are at the first character of the last line, then press 'o'. This action will take you to a new line. Now start typing the contents, in case of any errors in typing we may use the Backspace key and go back one cursor position at a time to correct the error. Once the editing is over press ESC key and type *:wq* to go the last line mode, save the file and end the Vim session. Figure 7.4 shows the look of the editor after the new contents have been added.

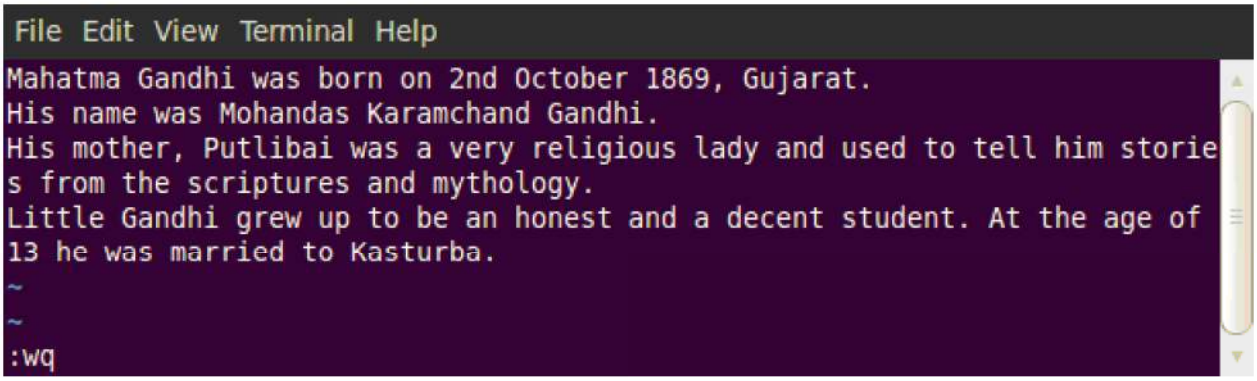


Figure 7.4 : Appending data in existing file

We may use the commands shown in table 7.5 to perform various editing operations on any document.

Command	Usage
u	Undo last change
U	Undo all changes to entire line
dd	Delete single line
Ndd	Delete N number of lines
D	Delete contents of line after cursor
C	Delete contents of line after cursor and insert new text. Press ESC key to end the insertion
dw	Delete one word
Ndw	Delete N number of words
cw	Change word
x	Delete the character under the cursor.
X	Delete the character before the cursor (Backspace).
r	Replace single character
R	Overwrite characters from cursor onward
s	Substitute one character under cursor and continue to insert
S	Substitute entire line and begin to insert at beginning of line
~	Change case of individual character
.	Repeat last command action.

Table 7.5 : Commands to perform editing

In addition to the commands given in table 7.5 the Vim editor also allows us to copy text from our file into temporary buffers and vice-versa. Each buffer acts like temporary memory, more commonly known as "clipboard". Table 7.6 lists some of the commands that are used for capturing and pasting data.

Command	Usage
yy	Copy single line (defined by current cursor position) into the buffer
Nyy	Copy N lines from current cursor position into the buffer
p	Place (paste) contents of buffer after current line defined by current cursor position.

Table 7.6 : Commands to capture and paste

Searching and replacing text

Searching for content and replacing it is another common operation performed by users. The Vim editor allows us to use special commands to search text or a regular expression within the file. We can also substitute a word in place of another using command. Table 7.7 lists various commands that are useful for performing search or replace operation within a file.

Command	Usage
/	Search for text in a forward direction
?	Search for text in a backwards direction
n	Search again in the same direction
SHIFT + n	Search again in the opposite direction
f	Press *f* and type the character to be searched. The cursor will move to that character on the current line.
SHIFT + f	Similar to *f* but searches in backward direction
t	Similar to *f* except that it moves the cursor one character before the specified character.
SHIFT + t	Similar to *t* but searches in backward direction
:s/old_string/new_string	Substitutes *new_string* for the first occurrence of the old_string in the current line
:s/old_string/new_string/g	Substitutes *new_string* for all the occurrences of the old_string in the current line
:%s/old_string/new_string/g	Substitutes *new_string* for all the occurrences of the old_string in the whole file
:%s/old_string/new_string/gc	Substitutes *new_string* for all the occurrences of the old_string in the file, but asks for confirmation before substituting the new_string

Table 7.7 : Commands to perform search and replace operation

Let us try to use some of these commands in the file about_Gandhiji. Assume that we wanted to replace all the occurrence of the word "Gandhi" with word "Gandhiji". To perform this operation we need to first open the file using the command *vi about_Gandhiji*, then go to the last line mode by pressing the *ESC* key and execute the command given below:

:%s/Gandhi/Gandhiji/g

Here %s indicates we are trying to replace a string. The term "Gandhi" refers to the old string that is to be replaced while the term "Gandhiji" refers to the new string. The option "g" indicates that we have to substitute all the occurrences of the term "Gandhi" with the term "Gandhiji" in the whole file. The output of this command is shown in figure 7.5. Observe that it also shows how many occurrences have been replaced. We need to save the file if we need this change to be reflected in it. If we quit without saving then the changes will not be reflected in the file.

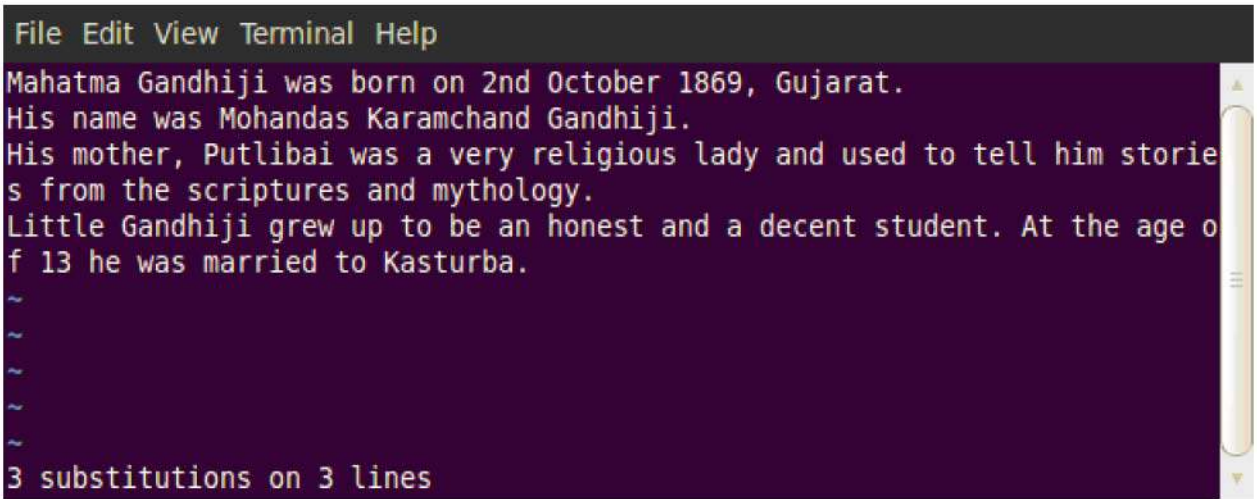


Figure 7.5 : Search and replace operation

Executing Linux commands through Vim

It is also possible to execute the Linux commands from within the Vim editor. To execute any Linux command we need to type the exclamation (!) symbol before the command.

For example if we want to see the current working directory then perform the following steps :

- Open the Vim editor by typing vi on the command prompt.
- Go to the last line mode by pressing 'ESC :'
- Type !pwd
- Press Enter key.

We will be able to see the current working directory. Figure 7.6 shows the operation and its output.

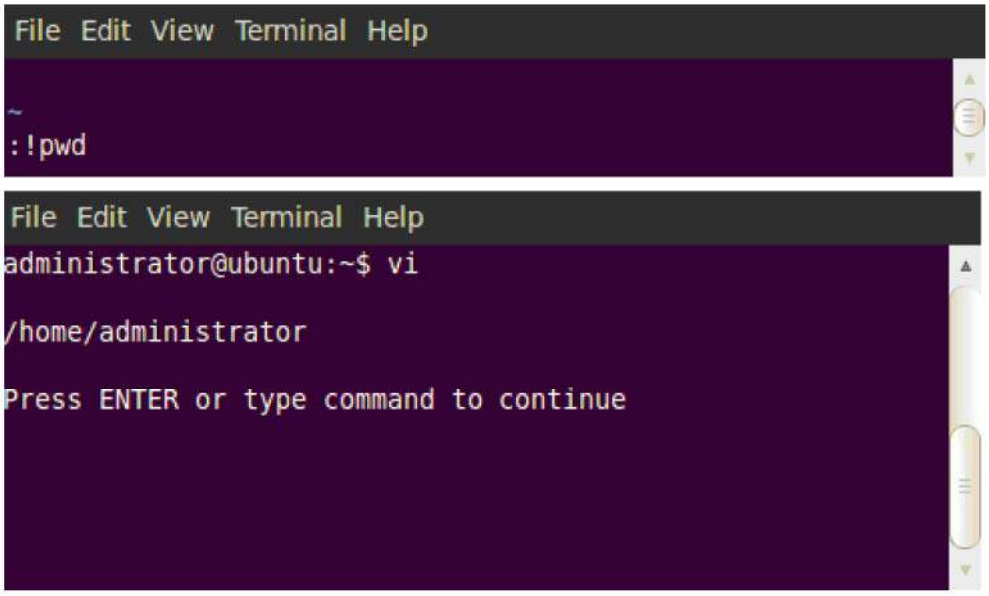


Figure 7.6 : Linux command in Vim editor

Similarly if we want to add the current date in a new line from the current cursor position within a file we may execute the following steps:

- Open the file using Vim editor.
- Go to the last line mode by pressing 'ESC :'
- Type r !date
- Press Enter key.

The r option allows us to insert data in the file or buffer.

Shell Script

We saw how a command can be executed from a command prompt as well as using the Vim editor. Both these mechanisms allow us to execute commands one at a time. A shell script allows us to execute more than one command at one go in a better way. Thus instead of spending time in typing the commands on the prompt every time a task needs to be performed, we can create a shell script and execute the given set of commands by typing a single line command.

Shell script can be defined as "Set of commands written in plain text file that performs a designated task in a controlled order." Shell scripts can be designed to be interactive; such a script may accept input from the user and perform different tasks based on the input provided.

Creating and Executing a Shell Script

We can create the shell script using any text editor. As we have learned to work with the Vim editor we will create scripts in it. Let us create a small shell script that welcomes a user. Type the script shown in the box below in Vim editor and save it with the name *script1.sh*. The extension

"sh" is basically used to specify that the file is a shell script. Note that it is not mandatory to create a file with an extension "sh", a file without extension can also be used as a shell script. But it is always a good practice to give extension as it will help us differentiate between normal files and shell script files.

```
#Script 1: Script to welcome the user who has logged into the system
clear
echo Hello
who am i
echo  Welcome to Ubuntu Linux
```

Observe the first line in the script; it begins with symbol '#'. Any line preceded by the '#' symbol is considered as a comment. The comments when part of the script are not executed; they are messages that help user understand the usage or meaning of the script. The second line is a command that clears the screen contents before giving the output of the script. The third line displays a message "Hello", the fourth line executes a command "*who am i*" that gives us the name and some additional details of the user currently logged into the system. The last line again displays a message "Welcome to Ubuntu Linux".

To execute the script we need to use *sh* or *bash* command. If the script is stored in current directory then type the command mentioned below:

$bash script1.sh or **$sh script1.sh**

Sometimes we might come across issues related to file privileges. For a script to be executed it needs to have execute permission explicitly set. By any chance if such a problem occurs we will have to use the chmod command to set the desired privileges. For Example issuing the command ***chmod + x script1.sh*** will make the file executable. If everything goes fine we will get the output similar to the one shown in figure 7.7.

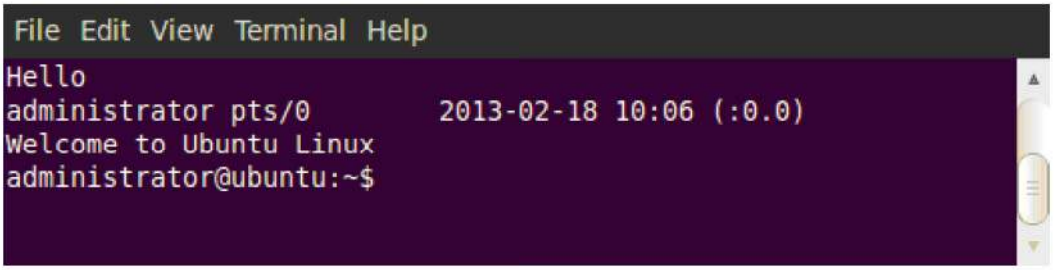


Figure 7.7 : Output of Script 1

Observe that in figure 7.7 we are getting some additional contents along with the user name. Let us use our knowledge of filters and try to remove the additional contents. The modified script is shown below:

```
#Script 2: Modified script to welcome the user who has logged into the system
clear
echo Hello
echo "`who am i | cut -d " " -f 1`"
echo Welcome to Ubuntu Linux
```

Observe that we have used the filter *cut* along with the command *who am i*. We further have joined the two commands using the pipe. To make sure that the contents within the double quotes after the echo command are not treated as message we enclose them in back quotes (‘ ‘). The back quotes are printed on the key with ~ sign on the keyboard. Type the modified script using Vim editor and save it as *script2.sh*. Execute this script and you will observe that we are only able to see the contents that we want. Figure 7.8 shows the output of the modified script.

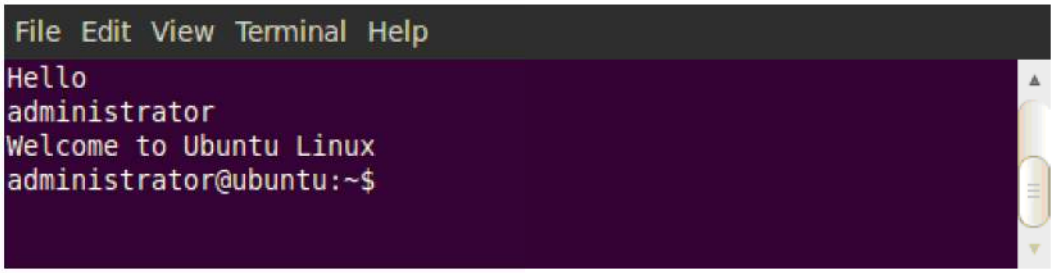


Figure 7.8 : Output of Script 2

Let us further modify the script to display current date and time. The script is given in the box below .

```
#Script 3: Script to welcome the user and display login date and time
clear
echo Hello
echo "`who am i | cut -d " " -f 1`"
echo Welcome to Ubuntu Linux
echo The current date and time is
date
```

Create a file named *script3.sh* and type the contents of the script 3 in it. Execute it and observe the output. Once we are comfortable using the shell scripts we will find them very helpful in performing repetitive tasks.

Shell Script Variables

The process of shell scripting is almost similar to the process of writing programs in a higher level language. One of the most common features of higher level programming is provision of variables.

As the name indicates variables are entities wherein we can store or edit a value. The value stored in the variable can also be reused or changed as per users need. Shell script variables like any other programming language variables are integral part of shell scripting. A variable when used in a shell script allows us to assign a value to it or accept its value from the user. We can also display the value assigned to this on the screen using echo command. Let us write a small script that shows the use of variable. The code of the script is given in the box below:

```
# Script 4: Shell script to show use of variables
clear
subject="Computer Science"
echo $subject is easy to learn.
```

Type the contents of the script 4 in a file named *script4.sh*. Let us try to understand the script just saved. Similar to all the other scripts the first line indicates a comment. The second line clears the screen of any previous contents. In the third statement we have defined a variable called subject and assigned it value "Computer Science" using a simple assignment operator. As the string contains white space in between two words we need to enclose it within double quotes. The fourth line displays the message on the screen. The '$' symbol preceding the variable name *subject* instructs the shell to extract the value stored or assigned in the variable. Figure 7.9 displays the output of the script when it is executed.

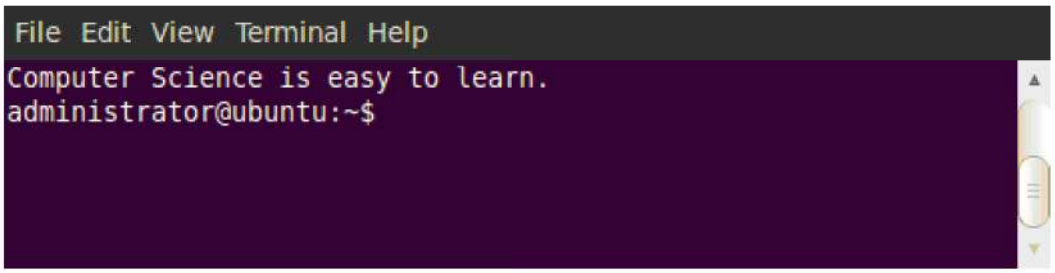


Figure 7.9 : Output of Script 4

A user needs to take care that there should not be any space on either side of the equal to (=) symbol at the time of assigning a value. If due to some reasons a space occurs at either side, the shell will interpret the string after the space as a command. This may give unexpected outputs. The statement subject="Computer Science" first creates a variable named *subject* and then assigns it a value "Computer Science". If we reuse this variable again, the old value stored in it will be overwritten. Let us try to understand the last statement by writing a simple script given in the box:

```
# Script 5: Shell script to show use of variables
clear
subject="Computer Science"
echo $subject is easy to learn.
subject="Economics"
echo $subject is easy to learn.
```

Save the script as script5.sh and observe its output after executing it.

We need to follow the rules mentioned below when defining a variable in a shell script.

- A variable name can consist of alphabets, digits or an underscore (_).
- No special character other than underscore allowed as part of variable name.
- The first character of a variable name must either be an alphabet or an underscore.

Note :

If the shell is unable to understand a word as a variable it will interpret it as a Linux command.

User Interaction and Shell Script

You must have observed in the previous example that we have assigned a value to the variable within the script itself. A variable when used in such a manner does not have much significance. A variable is generally used so that it can be further used in some operations with change in its value if needed. This property can be achieved only if we are able to accept the value of the variable from the user. It is possible to assign value to variables defined in the shell script using the *read* command. The *read* command expects the user to key in the data on the standard input device, it then takes all the contents that we type and stores it in the variable name supplied to it as an argument. Let us rewrite a script 5 to accept the subject names from the user. Save the code in the box with a file name *script6.sh*.

```
# Script 6: Shell script to accept value of variable from user
clear
echo -n "Enter your name: "
read name
echo -n "Enter name of a subject: "
read subject
echo Hello I am $name and I am studying $subject.
```

When we execute the script, the first executable statement will first clear the screen contents. Then it will display a message “Enter your name:”, the next statement waits for the user to enter its name. Pressing of Enter key indicates end of entry, so be careful that it is pressed only after the name has been typed. In case we press Enter key without typing anything the script will assign NULL value to the variable and move to next command. The third and fourth

statement also does the same task of displaying message and waiting for the user to key in a value for subject. The last statement displays the values of both the variables along with an appropriate message.

Observe that we have used -n option along with the echo command. This option instructs the echo command not to print a new line after the message is displayed. The echo command by default inserts a new line after displaying the message passed to it as an argument. Figure 7.10 shows output of the script.

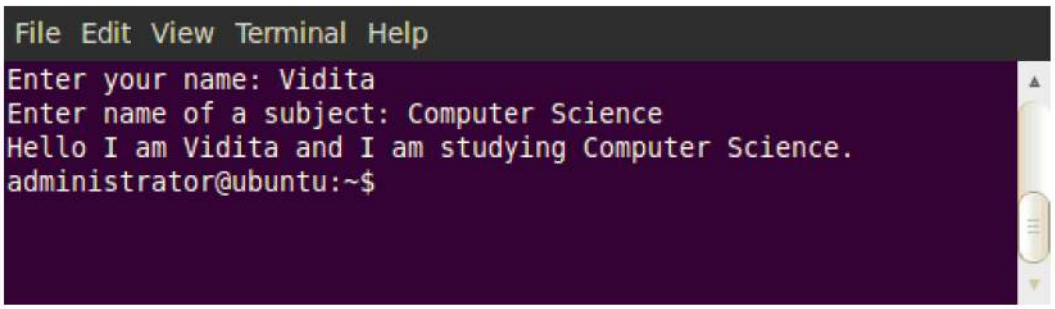


Figure 7.10 : Output of Script 6

As we are accepting the values of both the variables from the user, the output will change according to what user enters every time we execute the script. Let us write one more script that accepts a file name from the user and display the total number of lines in that file. The code of the script is shown herewith save it as *script7.sh*.

```
#Script 7: Shell script to display total number of lines in a file
clear
echo  -n "Enter a file name:  "
read fname
echo "There are `cat $fname | wc -l ` lines in a file $fname"
```

Figure 7.11 shows the output of the script when it is executed.

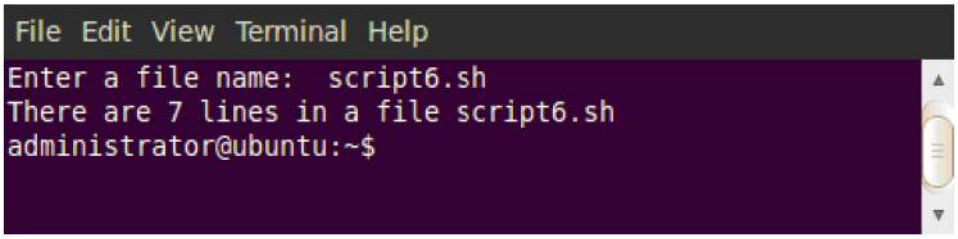


Figure 7.11 : Output of Script 7

Shell Arithmetic

In the previous section we saw how to define a variable, assign value to it and how to retrieve the stored value. The value assigned so far were all strings (set of alphabet, digit or special characters). We can also assign only numeric values to the variable and perform operation with them. Let us write a shell script that accept two numbers and perform addition of these numbers. The code of the script is shown in the box below:

```
# Script 8: Script to add two numbers
echo -n "Enter first number: "
read num1
echo -n "Enter second number: "
read num2
sum=`expr $num1 + $num2`
echo "The addition of $num1 and $num2 is $sum"
```

Type the script and save it as *script8.sh*. In this script the term *expr* means expression, the contents written after expr are assumed to be operands and operators of an expression. Note that there should be one space between operator (+) and operands ($num1, $num2). Additionally, there should be no space before or after the assignment operator (=).

Figure 7.12 shows the different output of the script when it is executed twice. We are able to see the different outputs in one screen as we have not used the clear command in this script.

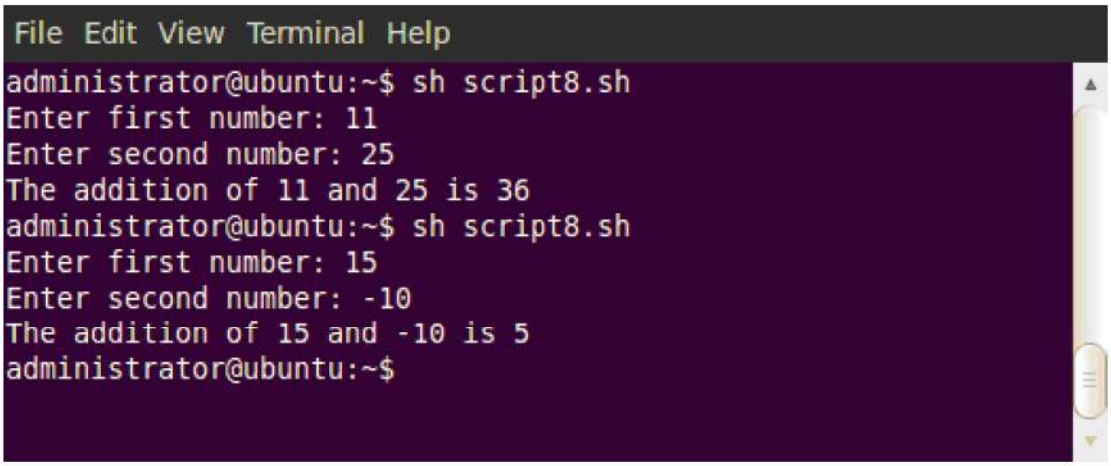


Figure 7.12 : Output of Script 8

We can also perform subtraction, multiplication, division and modular division by using the **-**, *, / and % operators respectively. The expressions are evaluated as per the general norms of mathematics.

In case of tie between operators of same priority, preference is given to the operator which occurs first. To force one operation to be performed earlier than the other, we can enclose the operation in parenthesis.

For example, in the expression **$num1 * \($num2 + $num3 \) / $num4**, the operation **$num2 + $num3** will be evaluated first as it is enclosed within parentheses. Observe that we have preceded the '*' symbol as well as the left and right parentheses by a back slash character (\).

Note :
We need to prefix the multiplication (*) symbol with backslash (\) character when finding product of two numbers. Otherwise the shell will treat the (*) symbol as a wildcard character.

Let us create one more script that will accept a birth year from the user and display users current age in years. The code of the script is shown in the box below.

```
# Script 9: Script to calculate age of user in years
echo -n "Enter year of your birth: "
read byear
cyear=`date | tr -s ' ' | cut -d " " -f 6`
age=`expr $cyear - $byear`
echo "You are $age years old as of today."
```

Observe that in script 9 to make sure that all the multiple spaces in output of date command are squeezed to single space we have used the *tr* command with *–s* option. As we want only the year value which appears in the 6th column of the output when the date command is executed we have used the *cut* filter. Figure 7.13 shows the different output of the script when it is executed.

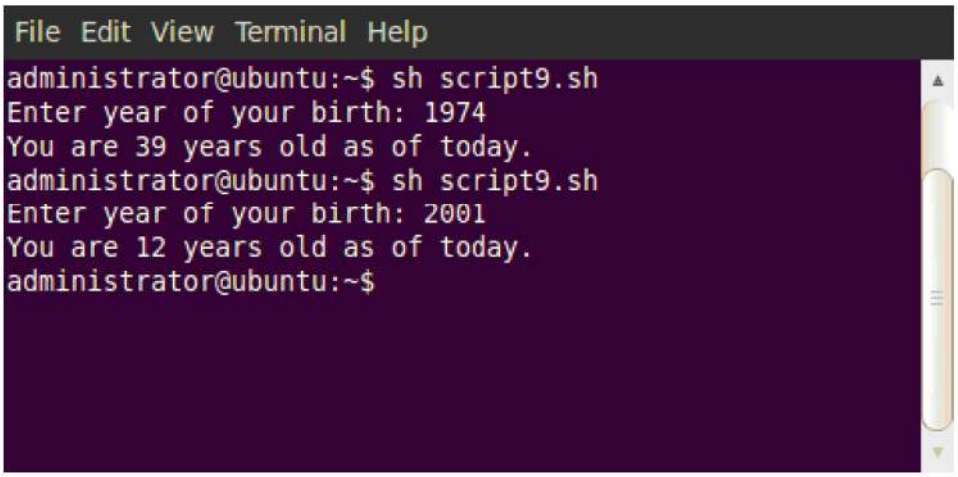


Figure 7.13 : Output of Script 9