

# INTERNATIONAL JOURNAL OF GRAPHICS AND MULTIMEDIA (IJGM)

ISSN : 0976 – 6448 (Print)  
ISSN : 0976 – 6456 (Online)

Volume 2, Issue 1, January- December (2011), pp. 12-29  
© IAEME: [www.iaeme.com/ijgm.html](http://www.iaeme.com/ijgm.html)  
Journal Impact Factor (2011) - 0.1845 Calculated by GISI  
[www.jifactor.com](http://www.jifactor.com)



.....

## ANIMATION IN J2EE PROJECTS - AN OVERVIEW

*Mr. S. Manivannan<sup>1</sup>, Dr. S. Balasubramanian<sup>2</sup>*

### **Abstract:**

Programming in Java doesn't have to be dull and boring. In fact, it's possible to have a lot of fun while programming in Java. When it comes to having fun while programming, it's hard to beat a good old fashioned program that provides visual feedback and stimulation. And in that category, it's hard to beat an animation program. Many forms of animation are possible in Java. What all of them have in common is that they create some kind of motion on the screen by drawing successive frames at a relatively high speed (usually about 10-20 times per second). There are many varieties in animation like sprite animation, frame animation etc., For example, if you watch The Discovery Channel or The Learning Channel very much, you will already know that many sea creatures have the ability to change their color in very impressive ways. Sprite animation is used to cause the spherical creatures to swim, and will also use frame animation to cause them to change their color at the same time. This paper discusses about the various techniques, types of animations and example programs to illustrate the techniques.

<sup>1</sup> *Research Scholar, Anna University Coimbatore, Coimbatore - 641 046  
Email: [smanivannan2003@yahoo.com](mailto:smanivannan2003@yahoo.com)*

<sup>2</sup> *IPR Consultant & Research Supervisor, Anna University Coimbatore,  
Coimbatore - 641 046 Email: [s\\_balasubramanian@rediffmail.com](mailto:s_balasubramanian@rediffmail.com)*

## ANIMATION - DEFINITION

Animation is the rapid display of a sequence of images of 2-D or 3-D artwork or model positions in order to create an illusion of movement. It is an optical illusion of motion due to the phenomenon of persistence of vision, and can be created and demonstrated in a number of ways. The most common method of presenting animation is as a motion picture or video program, although several other forms of presenting animation also exist.

## Animation Techniques

- **Full animation** refers to the process of producing high-quality traditionally animated films, which regularly use detailed drawings and plausible movement. Fully animated films can be done in a variety of styles, from realistically designed works such as those produced by the Walt Disney studio, to the more "cartoony" styles of those produced by the Warner Bros. animation studio. Many of the Disney animated features are examples of full animation, as are non-Disney works such as *The Secret of NIMH* (US, 1982) and *The Iron Giant* (US, 1999), *Nocturna* (Spain, 2007)
- **Limited animation** involves the use of less detailed and/or more stylized drawings and methods of movement. Pioneered by the artists at the American studio United Productions of America, limited animation can be used as a method of stylized artistic expression, as in *Gerald McBoing Boing* (US, 1951), *Yellow Submarine* (UK, 1968), and much of the anime produced in Japan. Its primary use, however, has been in producing cost-effective animated content for media such as television (the work of Hanna-Barbera, Filmation, and other TV animation studios) and later the Internet (web cartoons). Some examples are; *Spongebob Squarepants* (USA, 1999-

present), The Fairy Oddparents (USA, 2001-present) and Invader Zim (USA, 2001-2006).

- **Rotoscoping** is a technique, patented by Max Fleischer in 1917, where animators trace live-action movement, frame by frame. The source film can be directly copied from actors' outlines into animated drawings, as in The Lord of the Rings (US, 1978), used as a basis and inspiration for character animation, as in most Disney films, or used in a stylized and expressive manner, as in Waking Life (US, 2001) and A Scanner Darkly (US, 2006). Some other examples are; Ralf Bakshi's The Lord of the Rings (USA, 1978) and Fire and Ice (USA, 1983), [[Heavy Metal (1981 film|Heavy Metal).
- **Live-action/animation** is a technique, when combining hand-drawn characters into live action shots. One of the earlier uses of it was Koko the Clown when Koko was drawn over live action footage. Other examples would include Who Framed Roger Rabbit? (USA, 1988), Space Jam (USA, 1996) and Osmosis Jones (USA, 2002).
- **Anime** is a technique primarily used in Japan but originated in USA. It usually consists of detailed characters but more of a stiff animation. mouth moments primarily use 2-3 frames, leg moments use about 6-10, etc. A lot of the time the eyes are very detailed, so sometimes instead of the animator drawing them over again in every frame, two eyes will be drawn in 5-6 angles and pasted on each frame(modern times uses computer for that). Some example of Anime films are; Spirited Away (Japan, 2001), Akira (Japan, 1988) and Princess Mononoke.

- **Drawn on film animation:** a technique where footage is produced by creating the images directly on film stock, for example by Norman McLaren, Len Lye and Stan Brakhage.
- **Paint-on-glass animation:** a technique for making animated films by manipulating slow drying oil paints on sheets of glass.
- **Pinscreen animation:** makes use of a screen filled with movable pins, which can be moved in or out by pressing an object onto the screen. The screen is lit from the side so that the pins cast shadows. The technique has been used to create animated films with a range of textural effects difficult to achieve with traditional cel animation.
- **Sand animation:** sand is moved around on a backlit or frontlit piece of glass to create each frame for an animated film. This creates an interesting effect when animated because of the light contrast.
- **Flip book:** A flip book (sometimes, especially in British English, flick book) is a book with a series of pictures that vary gradually from one page to the next, so that when the pages are turned rapidly, the pictures appear to animate by simulating motion or some other change. Flip books are often illustrated books for children, but may also be geared towards adults and employ a series of photographs rather than drawings. Flip books are not always separate books, but may appear as an added feature in ordinary books or magazines, often in the page corners. Software packages and websites are also available that convert digital video files into custom-made flip books.

## COMPUTER ANIMATION

Computer animation encompasses a variety of techniques, the unifying factor being that the animation is created digitally on a computer.

### 2D Animation

2D animation figures are created and/or edited on the computer using 2D bitmap graphics or created and edited using 2D vector graphics. This includes automated computerized versions of traditional animation techniques such as of tweening, morphing, onion skinning and interpolated rotoscoping.

**Examples:** Foster's Home for Imaginary Friends, SpongeBob SquarePants(certain sequences only), Danny Phantom(certain sequences only), The Fairly OddParents(certain sequences only), El Tigre: The Adventures of Manny Rivera

Analog computer animation

Flash animation

PowerPoint animation

### 3D Animation

3D animation digital models manipulated by an animator. In order to manipulate a mesh, it is given a digital skeletal structure that can be used to control the mesh. This process is called rigging. Various other techniques can be applied, such as mathematical functions (ex. gravity, particle simulations), simulated fur or hair, effects such as fire and water and the use of Motion capture to name but a few, these techniques fall under the category of 3d dynamics. Many 3D animations are very believable and are commonly used as Visual effects for recent movies.

**Examples:** Toy Story, Shrek, Pocoyo

## **ADVANTAGES**

1. They can graphically simplify complicated concepts and convey complex interrelationships, which are difficult to visualize. Concepts and ideas, which cannot easily be represented in words or even through illustrations, can be easily created and viewed from different angles.
2. Animation can combine vast amounts of scientific data into a compact package, which can then be presented simplistically.
3. Animation captures attention, and the information which is presented as a moving image is retained by the viewer for a longer time and with greater accuracy.
4. Animation can re-create an event, which is too expensive or too dangerous to reproduce, eg. an aircraft accident. Scenes which have been altered or which no longer exist, eg. demolished buildings or colonies to be constructed can be easily re-created through animation.
5. Virtual light sources from different angles are used with reflections, transparencies and shadows to optimize the photo-realistic effect.

## **Examples**

The methods `paint()` and `repaint()` which are the foundation of graphical display in Java. These methods are used to display images, shapes, colors etc. Now we proceed to learn about Animation using Java. Animation has many uses, from dramatic web pages to exciting games. Java animation is superior to things like animated GIF files because it provides dynamic interaction instead of just a static

performance. However a knowledge base of threads and exceptions is a must.

And a few more functions that you need to learn.

The following program introduces a new function called `getCodeBase`.

```
zzz.java
import java.applet.*;
public class zzz extends Applet
{
public void init()
{
System.out.println(getCodeBase());
}
}
```

The `init` function calls `System.out.println`. Here, instead of passing a string we are giving a function- `getCodeBase` as a parameter. `getCodeBase` already exists within the `Applet` class as a function. `getCodeBase` returns the location of the applet class i.e. the sub directory which contains the code of the applet.

```
C:\javaprg>javac zzz.java
```

```
C:\javaprg>appletviewer a.html
```

```
file:/C:/javaprg/
```

We once came across graffiti that said, "You can't fool me, I'm too stupid!!!" But we intend no folly here!

Let's understand the output.

Our applet is based on our local machine and hence it says file: We are working on the windows operating system and the current directory is the javaprg subdirectory in C drive. Had the applet been picked up from another site, getCodeBase would have returned a url (uniform resource locator). A url or a uri (uniform resource identifier) identifies some resource.

```
zzz.java
import java.applet.*;
public class zzz extends Applet
{
public void init()
{
java.util.Date d;
d = new java.util.Date();
System.out.println(d);
}
}
```

In the previous programs, when we declared a variable int ii, we were actually creating an object ii that looked like an int. In this program, java.util.Date d will create a variable called d that looks like java.util.Date. We decided not to have the import statement for Date, hence the longish name. You can rewrite the code with Date and the line import java.util.\* on the top.

There is a difference between saying Date d and int i. A variable gets created as an object if its data type is int, char or long. Since these data types are built into programming languages like C and C++, the object is created at the time of its declaration. Anything

other than the basic data types must be explicitly created. Java implements the same rules.

Thus when we say `d` looks like `Date`, an object that looks like `Date` is still not created. Here the object is declared to be of `Date` type and will be created in the future.

Let's understand how we can create objects ourselves.

To create an object other than the basic types, languages like C++ utilized the word `new`. Java follows the same route and uses the `new` keyword as well. The next line in our program uses `new` to create the object `d`. This keyword requires the name of the class which in this case is `java.util.Date`.

`d` now becomes an instance or an occurrence of the class `Date`. `Date` is given with the round brackets because that is part of Java's syntax. The `println` function with `d` as a parameter will display the current date and time when the applet is executed.

```
C:\javaprg>javac zzz.java
```

```
C:\javaprg>appletviewer a.html
```

```
Tue Aug 31 14:31:07 GMT+05:30 2004
```

In the previous chapter, we used the `paint` function extensively. We also had `g` that looked like `Graphics` and if you recollect we did not use `new Graphics()` anywhere. The reason for this is that whoever called `paint` had already created an instance of

`Graphics` and passed it as a parameter. Hence we were spared the effort of doing so

ourselves.

```
zzz.java
import java.awt.*;
import java.applet.*;
public class zzz extends Applet
{
Image n;
public void init()
{
n = getImage(getCodeBase(),"T1.gif");
}
public void paint(Graphics g)
{
g.drawImage(n,10,40,this);
}
}
```

This program introduces an additional class called Image. n looks like Image. Every class has code and so does Image, which has its own code and its own set of variables. This class handles images. getImage is a function that requires a location from where the image file can be picked up. So the first parameter is the location and the second parameter is the name of the graphics file. The function getCodeBase returns the location of the applet.

For this program, you need a gif file called T1.gif which must be placed in the same directory as that of the applet. T1.gif comes with the Java development kit.

One of the subdirectories in the Java subdirectory is called images, which has beans and within beans lies T1.gif;

To be precise the gif files are in,

C:\Sun\AppServer\jdk\demo\plugin\applets\Animator\images\Beans.

The T series starts from T1 and goes up to T10. We recommend you copy all of them to your current directory, which in our case is c:\javaprg.

getImage returns an image which is stored in n. That means this function internally performs a new Image and returns an object that looks like Image. In our current program, n represents an image T1.gif. Just as drawLine and drawString are functions in the Graphics class and to call them we say g.drawString and g.drawLine, the same applies to the drawImage function. It is also to be found within the Graphics class. drawImage takes 4 parameters. The dot separates the name of the object from the name of the function.

The first parameter to the drawImage function is an image, the second is the x coordinate, the third is the y coordinate and the last one is an instance of an object. n in our program represents the image T1.gif to be displayed. The next two are the locations on the screen where we want the image to appear and finally we have the word 'this'.

'this' is a reserved word; it is a part of the Java programming language. In our code it stands for two things, either zzz or applet. 'this' stands for the current class or the class that you are extending from. So if Applet has been extended from two or three or more classes then it stands for them too. drawImage requires an instance of a class i.e. an object and hence we provide it as the last parameter by saying 'this'; which means

zzz or Applet.

After running the applet you will see the image T1.gif displayed in the appletviewer window. Maximize the window to view it properly.

The next program is similar to the earlier one. This program proves that strings and images are all treated in the same manner. This may seem strange at first, but you'll soon get used to the idea.

```
zzz.java
import java.awt.*;
import java.applet.*;
public class zzz extends Applet
{
Image n;int x = 19,y=76;
public void init()
{
n = getImage(getCodeBase(),"T1.gif");
}
public void paint(Graphics g)
{
g.drawImage(n,x,y,this);
}
public boolean mouseDown(Event e , int x1 ,int y1)
{
x = x1; y = y1;
repaint();
return true;
}
```

```
}  
}
```

In this program, we have used `drawImage` instead of `drawString`. Whenever the mouse is clicked in the window, `x` and `y` get initialized to the click position. `repaint()` calls the `paint` function which in turn draws the image where the mouse is clicked. Thus the image follows the mouse click.

## Loops

The `if` statement is the bedrock of programming because it gives intelligence and decision power to a language. The second major part of any programming language is a looping construct. In Java, the `for` statement allows you to repeat computer programming statements. Needless to say, you already know that we can enclose statements within a block, thus it also allows repetition of multiple statements.

This next program should clarify this.

```
zzz.java  
import java.applet.*;  
public class zzz extends Applet  
{  
public void init()  
{  
int i;  
for ( i = 1; i<= 10; i = i + 1)  
{  
System.out.println("Hi "+i);
```

```
}  
}  
}
```

The for statement has 2 semicolons. The statement upto the first semicolon is executed only once. For the first time i.e. only once is i initialized to 1. The statement enclosed within the first and second semicolon is a condition. The condition checks whether  $i \leq 10$  evaluates to true. Since this condition evaluates to true, the statements within the open and the close brackets are executed. If the condition evaluates to false, these statements are skipped and the loop terminates.

Here, since the for loop is required to execute only a single statement, the '{}' brackets are optional. This rule is applicable here too. The variable i has a value 1 which is less than 10, so System.out.println will be called and it will display hi 1. After all the statements within the block are executed, the last part of the for is executed.  $i=i+1$  will increase the value of i by 1, making its new value 2. The condition is checked again, is  $2 \leq 10$ ? The answer here is true so hi 2 is displayed. Now i is incremented once more and it becomes 3.

The condition  $3 \leq 10$  is again evaluated to true or false and this goes on till the condition is false. When i has the value 11, the condition checked is,  $11 \leq 10$ , which is false. The for terminates and the remaining lines of the program after the for block are executed. In this manner, the for statement enables the repetition of code. When we leave the for statement the value of i will be 11.

The while loop is similar to the for loop.

```
zzz.java
import java.awt.*;
import java.applet.*;
public class zzz extends Applet
{
public void init()

{
int i;
i = 1;
while ( i<= 10 )
{
System.out.println("Hi "+i); i =
i + 1;
}
}
}
```

The while loop only takes a condition, hence the variable i is initialized to 1 before entering the loop. The condition checks if i <= 10 evaluates to true. Currently the value of i is 1. The condition evaluates to true and the statements within the curly braces are executed. System.out.println is called with Hi and i. The next most important thing to do is increment the value of i or else it will have the same value and the loop will go on indefinitely.

You may ponder over the question, "Should I use for or while?" After all, the for loop is similar to the while loop. To answer your question is "On Mondays, Wednesdays, Fridays use for and on Tuesdays, Thursdays, Saturdays use while. Take a break on Sundays; nobody works on Sundays". Alternatively "Toss a coin. If it's heads use while and if it's tails, don't use for" ;-)

This is a prevalent problem with computer programming. There are multiple ways of doing the same thing. Both while and for do the same thing. The middle parameter of the for statement and the condition in while loop are basically the same, so it is entirely up to you to choose one.

The next program creates an infinite loop using the while statement.

The while condition simply says true. The condition will never evaluate to false and hence the loop will go on indefinitely.

```
zzz.java
import java.awt.*;
import java.applet.*;
public class zzz extends Applet
{
public void init()
{
int i;
i = 1;
while ( true )
{
```

```
System.out.println("Hi "+i); i =  
i + 1;  
}  
}  
}
```

Press Ctrl-C in the dos console to end of the program

## **CONCLUSION**

Thus animations play a vital and critical role in communications...

It is proved that the animations can combine vast amounts of scientific data into a compact package, which can then be presented simplistically, can capture attention, and the information which is presented as a moving image is retained by the viewer for a longer time and with greater accuracy and can re-create an event, which is too expensive or too dangerous to reproduce, eg. an aircraft accident and All these can be achieved by using J2EE technology which is very popular across the world.

## **REFERENCES**

1. Culhane, Shamus, Animation Script to Screen
2. Laybourne, Kit, The Animation Book
3. Ledoux, Trish, Ranney, Doug, & Patten, Fred (Ed.), Complete Anime Guide: Japanese Animation Film Directory and Resource Guide, Tiger Mountain Press 1997
4. Lowe, Richard & Schnotz, Wolfgang (Eds) Learning with Animation. Research implications for design Cambridge University Press, 2008

5. Masson, Terrence, CG101: A Computer Graphics Industry Reference  
Unique and personal histories of early computer animation production,  
plus a comprehensive foundation of the industry for all reading levels.  
ISBN 0-9778710-0-2  
Animation Art: From Pencil to Pixel, the History of  
Cartoon, Anime & CGI. Fulhamm London.: Flame Tree Publishing. ISBN  
1-84451-140-5
  
6. Crafton, Donald (1982). Before Mickey. Cambridge, Massachusetts.: The  
MIT Press. ISBN 0-262-03083-7
  
7. Solomon, Charles (1989). Enchanted Drawings: The History of Animation.  
New York.: Random House, Inc. ISBN 0-394-54684-9
  
8. Java 2: The Complete Reference, Fifth Edition