Colors of the music

Meng-Tieh Fang
Colors of the Music

by Meng-Tieh Fang

A Thesis submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Fine Arts in Computer Graphics Design

Department of Computer Graphics Design
School of Design
College of Imaging Arts and Sciences
Rochester Institute of Technology
Rochester, NY
September, 2009
Approvals

Chief Advisor: Chris Jackson, Associate Professor, Computer Graphics Design

Signature of Chair Advisor  Date

Associate Advisor: Dan DeLuna, Assistant Professor, Computer Graphics Design

Signature of Associate Advisor  Date

Associate Advisor: Nancy Doubleday, Associate Professor, Information Technology

Signature of Associate Advisor  Date

School of Design Chairperson:
Patti Lachance, Associate Professor, School of Design

Signature of Administrative Chairperson  Date

Reproduction Granted:
I, MENG-TIEH FANG, hereby grant permission to Rochester Institute of Technology to reproduce my thesis documentation in whole or part. Any reproduction will not be for commercial use or profit.

Signature of Author  Date

Inclusion in the RIT Digital Media Library Electronic Thesis and Dissertation (ETD) Archive:
I, MENG-TIEH FANG, additionally grant to Rochester Institute of Technology Digital Media Library the non-exclusive license to archive and provide electronic access to my thesis in whole or in part in all forms of media in perpetuity. I understand that my work, in addition to its bibliographic record and abstract, will be available to the worldwide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles and books) all or part of this thesis. I am aware that Rochester Institute of Technology does not require registration of copyright for ETDs. I hereby certify that, if appropriate, I have obtained and attached written permission statements from owners of each third party copyrighted matter to be included in my thesis. I certify that the version I submit is the same as that approved by my committee.

Signature of Author  Date
Abstract

Colors of the Music
Meng-Tieh Fang

It is a generation of high technology. Our daily life is full of computing in the aspects of work and leisure. Computing also influences on art, and now a computer can be a new form of representation. Artists use computers to create and show their art works, and viewers now not only view those art works but also involve in art works through the interactivity of a computer.

Motion graphics are attractive. Flash based programs are rich in interaction. This project combines the two features to create a user driven art that promotes classical music to children and of course to every one. In such an interactive and attractive way, users enjoy the music and listen to the three world-known songs from Saint Saëns’s Carnival of Animals- Royal March of Lion, Aquarium, and the Swan. When the music starts, by clicking the mouse, users can add motion graphics related to the music on a computer screen.

It is a problem to figure out a way to generate pixel-based motion graphics with Flash, which usually deals with vector images. First built are the codes, that give this project functions of interaction. Users can choose a song, add motion graphics on a computer screen, pause or play the music, change the volume of the sound, choose different color, and print out images they create. The rest are the motion graphics, that reflect the unique theme of each song. In order to display well, different methods are examined to generate motion graphics, including loading FLV files exported by After Effects, and adding movie clips built in Flash, and more.

The result shows that adding movie clips built in Flash is the best way. Other ways result in the pixilated edges of graphics, the difficulty of making alpha movies, or other technical problems. Having the capabilities of enabling users to control the program and displaying motion graphics in high quality, this project performs as a new medium introducing classical music to new generations.

Key words:
Flash, After Effect, Interaction, Motion graphics

The website address:
http://www.cheerevelyn.com/ColorsOfTheMusic.html
Content

Introduction ................................................................. 1
Review of Literature ....................................................... 2
Process ........................................................................... 3
   I.  Control the Program with Codes ................................ 3
       A. A Short Movie at the Beginning ......................... 3
           1. Loop the Short Movie .................................... 3
           2. Leave for Menu ............................................ 3
       B. Menu ............................................................... 5
           1. Three Choices ................................................ 5
           2. Information about the Music ........................... 5
           3. Guide of this Program ................................... 5
           4. Timer for Returning to the Beginning ................. 5
       C. Drawing Area and Tool Bar ............................... 6
           1. Area for Adding Motion Graphics ....................... 6
           2. Play the Music .............................................. 6
           3. Pause the Music ........................................... 6
           4. Control the Volume ........................................ 6
           5. Mute ............................................................ 7
           6. Rhythm Visualization Bar ................................ 7
           7. Choose Different Color .................................... 7
           8. Print ............................................................. 7
           9. Return to Menu ............................................... 7
          10. Timer for Counting the Length of a Song ............. 7
          11. Timer for Returning to Open ............................ 7
II. Display Motion Graphics .................................................. 8
   A. Add Pixel-Based Motion Graphics .................................... 8
      1. Load SWF files exported by After Effects ...................... 8
      2. Load FLV files exported by After Effects ...................... 8
      3. Add movie clips built in Flash .................................. 8
   B. Delete Motion Graphics ............................................... 9
   C. Style Design ............................................................. 9
      1. Royal March of Lion .............................................. 9
      2. Aquarium ......................................................... 9
      3. The Swan ......................................................... 9
Summary ............................................................................. 12
Conclusion ........................................................................ 14
Appendix ........................................................................... 15
Bibliography ...................................................................... 39
**Introduction**

Involving the concepts of rational, balance, equality, classical music is viewed as a beautiful art. However, for most children, classical music is very serious and abstract. In fact, classical music can contain some concrete meanings. Understanding the meaning in a song of classical music may increase children’s interest in classical music. Thus they get a chance to learn to admire and enjoy classical music.

This project uses the software, Flash, to design a user driven art combining motion graphics that visualize the meaning in classical music. To get to know classical music, it is better to start from some songs that are viewed as easy to be accepted by children or those who are not familiar with classical music. Therefore, from Saint Saëns’s Carnival of Animals, three songs are picked to be the subjects, and they are Royal March of Lion, Aquarium, and the Swan. They obviously describe concrete subjects. When listening to the music, users can add motion graphics related to the subjects on a computer screen by clicking the mouse.

Mainly, this project is comprised of two parts. One is the codes written in Flash ActionScript 3 for users to control the program. Users choose a subject a time, and click the mouse to add motion graphics on a computer screen. Users can also pause and play the music, change the volume, choose different color, and print image. The other part is the motion graphic that reflect the three subjects. The main point is to display the pixel-based motion graphics with Flash. To convey the concept of drawing, the texture of the motion graphics is crayon. Each graphic made with Photoshop is pixel-based. A method needs to be figured out to display the pixel-based motion graphic in high quality with Flash.

Users listen to Saint Saëns’s famous works by enjoying interaction and watching the colorful motion graphics. Not only children, but also every one can have fun with classical music, and understand and feel the meaning in it. This user driven art performs as a medium for classical music. It offers a chance for children and the crowd to easily enjoy classical music.
Review of Literature

Screen-Based Art
by Annette W. Balkema, Henk Slager.
Rodopi, 2000

With the development of technology, in the 21st century, the visual culture is occupied by the screen of many kinds, including the television screen, the video screen, the computer screen, etc. This book discusses the status of screen-based art, its difference from other art forms, and its physical and theoretical forms of representation. It is explored the connectedness of digital to the screen.

Computers and Art
by Stuart Mealing
Intellect Books, 1997

This book talks about using the computer as a tool in the field of art. Artists use the computer as a new form of representation to generate their art works. Besides, interactivity, a feature of the computer, changes the relation between artists and viewers from one way creation to various creation. Started from artists, an art work is continued by viewers. With the computer, viewers not only interpret but also participate in an art work.

Human-Computer Interaction: Theory and Practice
by Michael Smith, Constantine Stephanidis, Julie A. Jacko, Don Harris
Lawrence Erlbaum Associates, 2003

The increase of computing usage extended from workplace to home influences on the importance of human-computer interaction. This book contains various theoretical documents and practical projects displaying and exploring design of human-computer interaction. In one project, Developing Interactive Art Using Visual Programing, it is described how to use a visual programing environment called Max/MSP (Puckette, 2002) to build interactive art projects.
Process

I. Control the Program with Codes
At first, a short movie that loops shows three subjects- Royal March of Lion, Aquarium, and the Swan. Users move the cursor over this short movie and then enter the menu. In the menu, there are three buttons representing the three subjects. Users enter a subject by clicking one button. In each subject, there is a blank area for users to add motion graphics. Under this area, there is a tool bar for users to control this program, such as pause and play the music, change the volume, choose different color, and print images, etc. The whole program jumps back to the short movie at the very beginning, when staying in the menu for more than one minute, or five minutes later after the music ends.

A. A Short Movie at the Beginning
Looping at the beginning, the short movie shows three subjects, greets users, and gives an idea of what users are going to process and see. It works like a screen savor. When users touch the mouse and make the cursor move over this short movie, users are directed to the menu.

1. Loop the Short Movie
In Flash, to create a short looping movie, simply create a movie clip symbol on the stage. In this movie clip, short animations of the three subjects appear by turns. First showed is the subject of Royal March of Lion, and the following are the subjects of Aquarium and the Swan.

2. Leave for Menu
The codes, Array and Timer, are used to trace the position of the mouse. The short movie is a sensor. If, next second, the position of the mouse is different- users move the mouse, the short movie will be over, and users will enter the menu.
Animations in the short movie
B. Menu
Representing the three subjects, the buttons are designed in the figures of a lion, fish, and a swan. The color of the background changes when the mouse moves over different buttons. Users can click one button to enter one subject. At the bottom of the screen, there are two buttons for information about the music and the guide of the toolbar users are going to see after entering one subject. This whole program will return to the short looping movie at the beginning in one minute after entering the menu.

1. Three Choices
The color of the background changes to brown when the mouse moves over the button of a lion, changes to blue, when the mouse moves over the button of fish, and changes to gray when the mouse moves over the button of a swan. Users can enter a subject by clicking one of the three buttons.

2. Information about the Music
By clicking the button showing “info”, users can get brief information about the music and this project from a popping up window. Click the icon for closing to close the information window.

3. Guide of this Program
By clicking the button showing “guide”, users can get instruction of how to play. Click the icon for closing to close the guide window.

4. Timer for Returning to the Beginning
The code, Timer, is used to count one minute. When the time is up, the whole program returns to the short movie in the beginning.
C. Drawing Area and Tool Bar
After entering a subject, users see a tool bar at the bottom of the interface. By clicking the buttons on the tool bar, users can pause and play the music, control the volume, mute the sound, choose different color, print out images, and return to the menu. Besides, there is a bar visualizing the sound of the music, and a progress bar visualizing the current length of a song. Users cannot add motion graphics when the music ends. Five minutes after the music ends, this program jumps back to the short movie at the beginning.

1. Area for Adding Motion Graphics
In the center of the stage, there is transparent square sensing the position of the mouse. Motion graphics is added in the place where users click the mouse. When the music ends or pauses, this square stops sense as well; users cannot add motion graphics any more.

2. Play the Music
The music starts once users enter a subject. The button of playing the music works only when the music pauses.

3. Pause the Music
Users can pause the music any time. They can print images when they pause the music.

4. Control the Volume
Ten vertical bars shows the degree of the volume. Users can click the icons for plus and minor to increase or decrease the volume.
5. Mute
The button shows mute when the sound is on, and shows sound when the sound is off. Click the button to turn on or off the sound.

6. Rhythm Visualization Bar
This bar visualizing the sound and the rhythm of the music assists users to sense the beat in a song.

7. Choose Different Color
Each subject has its own tone of color. In the subject of Royal March of Lion, it is brown, in the subject of Aquarium, it is blue, and in the subject of the Swan, it is gray. In each subject, the difference of the color is the brightness. There are five levels for users to choose. The color in the same tone strengthens the style of a subject.

8. Print
When users pause the music or when the music ends, users can print out images by clicking the button of print. This button shows in low brightness and does not work, unless the button of pause is clicked or the music ends.

9. Return to Menu
The button showing menu is for users to click to return to the menu.

10. Timer for Counting the Length of a Song
Beneath all the buttons is the progress bar showing a concept of how long the song has been played. The timer in the codes counts the length of a song, and the progress bar grows steadily every certain period of time depending on the proportion of the length of a song to the width of the progress bar. Visualizing the length of a song helps users be aware of the time left for them to add motion graphics.

11. Timer for Returning to Open
In every subject, a timer is set in the ActionScript, and starts once users enter the subject. This timer counts five minutes more than the timer for counting the length of the song. When the time is up, the program returns to the short movie in the beginning.
II. Display Motion Graphics
Each subject has its own series of motion graphics designed according to the theme of each song. The tone of the color for the subject of Royal March of Lion is brown, for the subject of Aquarium is blue, and for the subject of the Swan is gray. Each series contains five groups that are different in the brightness from very dark to very bright. In each group, there are five motion graphics. One of the five motion graphics appears randomly in the place where users click the mouse. If users are not satisfied with what appears, they can click the delete button in the tool bar to delete the latest one until all the motion graphics are cleared.

A. Add Pixel-Based Motion Graphics
Conveying the idea of drawing, the texture of motion graphics is designed like crayon, and is made with Photoshop. Therefore, the graphics are pixel-based. The point is to show these pixel-based motion graphics with Flash. The following are some methods tested to make it work.

1. Load SWF files exported by After Effects
After Effect is a good software for making motion graphics, and it can export films to the form of looping SWF files. This way is acceptable though the special effects in After Effects is not allowed when exporting SWF files. However, loading SWF files is a large burden for Flash. It slows down the speed of the motion graphics because the loading SWF files demand much processing power. Therefore, loading SWF files was not the best solution.

2. Load FLV files exported by After Effects
Exporting FLV files from After Effect does not limit the usage of special effect. Besides, After Effects can export alpha movies, and this meets the need of a transparent background in each motion graphic. The quality of images in FLV files is very good. However, loading FLV files slows down the progress of Flash as well. With the increase of the motion graphics on the stage, motion graphics are delayed, and the cursor reacts slowly. Soon, this whole program breaks down. Besides, sometimes, the backgrounds of a few motion graphics becomes white and cover those motion graphics that appear earlier, though all the motion graphics are correctly exported to alpha movies. As a result, it cannot work to load FLV files exported by After Effects.

3. Add movie clips built in Flash
Finally, the idea of making motion graphics with After Effects is thrown away. All the motion graphics are made in Flash as many movie clip symbols. Photoshop files are accepted to be imported into Flash. Besides, when importing to the library, Flash offers an option that automatically
generates a movie clip symbol in which pieces of graphics are distributed to different layers according to their layers in the Photoshop file. It is very easy to edit. In each subject, there is a main movie clip symbol that contains all the motion graphics distributed in twenty-five frames. There are five different levels of brightness, so movie clip symbols in every five frames have similar brightness. When users click the mouse, the main movie clip symbol is added onto the screen and it stops at one frame judged according to the color users choose and a random number. Although motion graphics are pixel-based, the quality of displaying is quite good. Adding movie clips built in Flash does not cause the delay of showing motion graphics.

B. Delete Motion Graphics
An array is set to trace the number of the movie clips on the screen. When users click the delete button, the last movie clip is deleted. This delete button does not work when the screen is empty.

C. Style Design
To emphasize the theme, each subject has its own tone of color, such as brown in the subject of Royal March of Lion, blue in the subject of Aquarium, and Gray in the subject of the Swan. The figures are a little bit abstract to make a space for imagination.

1. Royal March of Lion
Generally speaking, lions are strong, wild, and bold. The figures designed for lions tend to be geometry images. Some African animals and plants presenting the environment of lions also appear in the motion graphics.

2. Aquarium
Performing as water’s waves, many blue tone squares are in the motion graphics for the subject of Aquarium. Fishes appear and vanish into the dark blue. There are also bubbles, for bubbles are usually linked with water and fish. Besides, jellyfishes, star fishes, and some other kinds of creatures are also in the motion graphics.

3. The Swan
Elegance, peace, slow are most people’s impression toward swans. Going with the sound of violins, thin, straight lines slowly grows and crosses each other. Designed according to the sound of a piano, colorful circles are like pearls or the light of the sunshine. Feathers fly and shuttle everywhere, and swans are surrounded by this dream-like environment.
Royal March of Lion

Aquarium
The Swan
Summary

To make children or anyone get close to and feel interested in classical music, in this project, I designed a Flash based user driven art in which the topic is classical music. This user driven art combines motion graphics that visualize the meaning of classical music. When listening to the music, users can click the mouse to add and watch motion graphics on a computer screen.

The three songs of classical music are chosen from Saint Saëns’s Carnival of Animals, and they are Royal March of Lion, Aquarium, and the Swan. They apparently show some concrete meanings; they are viewed as easy to be accepted by children or those who are not familiar with classical music. Most people think that classical music is long and boring. The original length of the three chosen songs is not long, but in this project, to avoid feeling uninteresting, the length is cut even shorter to about one minute.

To contain the idea of drawing, the texture of the motion graphics is crayon. After testing with different methods, it is found that adding motion graphics built in Flash is the best solution. Motion graphics are made into movie clip symbols that do not affect the progress of Flash too much. Comparing to other methods such as loading FLV or SWF files exported by After Effects, this method does not cause unreasonable white background which should be transparent, and does not delay the movement of the cursor and of motion graphics.

The animation in detail in the motion graphics do appeal to users. Watching the change on a computer screen makes it natural to sit down to hear or listen to classical music. The slow motion works successfully in the subjects of Aquarium, and the Swan. It expresses the peace in the sea world and the elegance of swans. As for the subjects of Royal March of Lion in which the rhythm is strong and delighted, the motion needs to be faster and various.

The texture of the background is also crayon. The color of the background differs in each subject. For Royal March of Lion, it is brown, for Aquarium, it is blue, and for the Swan, it is gray. The background is static when the music is playing. Still, matching the theme, the background creates a precise atmosphere, and indeed involves users in the feeling of each subject.

The function of control for users works well. Users can add or delete motion graphics, pause or play the music on their decision, and change the volume of the music. They can choose different color to add and watch different motion graphics. When the music stops or pauses, users can print out images.
Appearing randomly, the motion graphics gives users expectation each time they click the mouse. Meanwhile, the delete function gives users the ability to make some arrangement toward the motion graphics according to their unique thought toward the music. Therefore, the images printed in the process or at the end shows different view or feeling from different user.

Thinking independently is the key. When a user tries to create his own image, he already starts to admire and make sense of classical music. In this project, it does not matter what the background knowledge of classical music is or how classical music is composed. Taking the chance to enjoy classical music, to get close to classical music, and to have fun with classical music is the substance of this project. There is no need to be afraid of classical music. Listening with some imagination, every one can enjoy it.
Conclusion

Being characterized by an emphasis on balance, clarity, and moderation, classical music is many people’s favor, and is played from century to century. Classical music is definitely worth introducing to new generations. Besides performing in a concert hall, in the present day, classical music needs a fresh way to be displayed to the crowd.

Since the rise of computing, computers are now applied in many areas. In the field of art, a computer becomes a new medium to display art works and in some cases increases the interaction between art works and viewers. With the software, Flash, this project creates a user driven environment that allows adding visual enjoyment to listening. Its goal is to make it more interesting to listen to and feel the classical music, and give classical music a chance to be known by children and every one.

Three songs are picked to be the subjects from Saint Saëns’s Carnival of Animals, and they are Royal March of Lion, Aquarium, and the Swan. Users choose a subject a time, and when listening to the music, they can add motion graphics about the subject on a computer screen. This project comprises two main parts. One is the functions of interaction built with Flash Action Script 3, giving users the ability of control. The other is the motion graphics that users are going to add.

In those motion graphics, conveying the concept of drawing, each graph’s texture is crayon and is a bitmap image. Therefore, the difficulty is to display the pixel-based motion graphics with Flash in high quality. To solve the problem, different methods are tested, including loading FLV files exported by After Effects, and adding movie clips built in Flash, and more. As a result, in a Flash based interactive program, to display pixel-based motion graphics in high quality, the best way is to add movie clips built in Flash by codes. Other ways may cause the pixilated edges of graphics, the difficulty of making alpha movies, or other technical problems.

This user driven art is completed, and it is time to do its duty. During the process of operating this program, the feedback that users see various motion graphics after clicking the mouse make users naturally stay in front of a computer listening to classical music. Even they may just hear the music, this user driven art is already a medium that successfully pass classical music to the crowd. The computer that features interactivty changes the way of enjoying classical music, and as well as makes a new approach to people for the classical music.
Appendix

Codes for the short movie at the beginning

```javascript
stop();

var loader:Loader = new Loader();
loader.load(new URLRequest("Open.swf"));
addChild(loader);

var positionX:Array = new Array;
var N:Number = new Number;
N = -1;

var timerOpen:Timer = new Timer(1000);
timerOpen.addEventListener(TimerEvent.TIMER, onTimerOpen);
timerOpen.start();

function onTimerOpen(evt:TimerEvent):void {
    var MOUSEX:Number = new Number;
    MOUSEX = mouseX;
    N++;
    positionX.push(MOUSEX);

    if (positionX[N]>positionX[N-1]) {
        gotoAndPlay(1, "Menu");
        removeChild(loader);
        timerOpen.reset();
        timerOpen.removeEventListener(TimerEvent.TIMER, onTimerOpen);
    } else {  
        gotoAndPlay(1, "Menu");
        removeChild(loader);
        timerOpen.reset();
        timerOpen.removeEventListener(TimerEvent.TIMER, onTimerOpen);
    }
}
```
Codes for the menu

stop();

//Timer
var timer:Timer = new Timer(60000);
timer.addEventListener(TimerEvent.TIMER, onTimer);
timer.start();

function onTimer(evt:TimerEvent):void {
    gotoAndStop(1, "Open");
timer.reset();
timer.removeEventListener(TimerEvent.TIMER, onTimer);
removeChild(Intro);
removeChild(Close);
}

//Go To Drawing Stage
function DrawingLion(event:MouseEvent):void {
    gotoAndPlay(1, "LionStage");
timer.reset();
timer.removeEventListener(TimerEvent.TIMER, onTimer);
removeChild(Intro);
removeChild(Close);
}

function DrawingSwan(event:MouseEvent):void {
    gotoAndPlay(1, "SwanStage");
timer.reset();
timer.removeEventListener(TimerEvent.TIMER, onTimer);
removeChild(Intro);
removeChild(Close);
}

function DrawingFish(event:MouseEvent):void {
    gotoAndPlay(1, "FishStage");
timer.reset();
timer.removeEventListener(TimerEvent.TIMER, onTimer);
removeChild(Intro);
removeChild(Close);
}

//Show info
var Intro:MC_intro = new MC_intro;
addChild(Intro);
Intro.x = 339.1;
Intro.y = 177.5;
Intro.visible = false;

var Close:BTN_close_btn = new BTN_close_btn;
addChild(Close);
Close.x = 567;
Close.y = 450;
Close.visible = false;

function ShowInfo(event:MouseEvent):void {
    Intro.visible = true;
    Close.visible = true;
}
function CloseInfo(event:MouseEvent):void {
    Intro.visible = false;
    Close.visible = false;
}

// Mouse Click
BTN_lion.addEventListener(MouseEvent.CLICK, DrawingLion);
BTN_swans.addEventListener(MouseEvent.CLICK, DrawingSwan);
BTN_fish.addEventListener(MouseEvent.CLICK, DrawingFish);
BTN_info.addEventListener(MouseEvent.CLICK, ShowInfo);
Close.addEventListener(MouseEvent.CLICK, CloseInfo);

// Background Change
var BG_Lion:MenuBG_Lion = new MenuBG_Lion;
BG_Lion.gotoAndPlay(1);
MenuBG.addChild(BG_Lion);

BTN_fish.addEventListener(MouseEvent.MOUSE_OVER, changeBGfish);
function changeBGfish(event:MouseEvent):void {
    var BG_Fish:MenuBG_Fish = new MenuBG_Fish;
    BG_Fish.gotoAndPlay(1);
    MenuBG.addChild(BG_Fish);
}

BTN_lion.addEventListener(MouseEvent.MOUSE_OVER, changeBGlion);
function changeBGlion(event:MouseEvent):void {
    var BG_Lion:MenuBG_Lion = new MenuBG_Lion;
    BG_Lion.gotoAndPlay(1);
    MenuBG.addChild(BG_Lion);
}

BTN_swans.addEventListener(MouseEvent.MOUSE_OVER, changeBGswan);
function changeBGswan(event:MouseEvent):void {
    var BG_Swan:MenuBG_Swan = new MenuBG_Swan;
    BG_Swan.gotoAndPlay(1);
    MenuBG.addChild(BG_Swan);
}
//Add Print Background
var printBGlion:MC_printBG_lion = new MC_printBG_lion;
LionHolder.addChild(printBGlion);

//Time bar
var timebarTimerLion:Timer = new Timer(100);
timebarTimerLion.addEventListener(TimerEvent.TIMER, barAddLion);
timebarTimerLion.start();
MC_timebar_lion.width = 10;

var a:Number = new Number;
a = 1;
function barAddLion(evt:TimerEvent):void {
  a +=20;
  MC_timebar_lion.width = a*0.1;
}

//Music
var sndLion:Sound = new Sound();
sndLion.load(new URLRequest("SongLion.mp3"));
var channelLion:SoundChannel;
channelLion = sndLion.play();

//Sound Data
var PeakLion:Number = new Number;
function SoundDetalLion(event:Event):void {
  PeakLion = Math.floor((channelLion.leftPeak+channelLion.rightPeak)*50);
  MC_peak_lion.scaleX = PeakLion*0.015;
  if (MC_peak_lion.width > 150) {
    MC_peak_lion.width = 150;
  }
  if (PeakLion == 0) {
    PeakLion = Math.floor(Math.random()*10)+1;
  }
}

//Music Stop
BTN_stop_lion.addEventListener(MouseEvent.CLICK, musicStopLion);
function musicStopLion(event:MouseEvent):void {
  BTN_print_lion.visible = true;
  var pausePosition:Number = channelLion.position;
  channelLion.stop();
  transLion.volume = vLion;
  channelLion.soundTransform = transLion;
  timerLion.stop();
timebarTimerLion.stop();
timerBackOpenLion.stop();
LionHolder.removeEventListener(MouseEvent.CLICK, addShapeLionClick);
BTN_mute_lion.removeEventListener(MouseEvent.CLICK, muteLion);
BTN_nomute_lion.removeEventListener(MouseEvent.CLICK, nomuteLion);
BTN_lowVolume_lion.removeEventListener(MouseEvent.CLICK, lowVolumeLion);
BTN_addVolume_lion.removeEventListener(MouseEvent.CLICK, addVolumeLion);
}
//Music Play
BTN_play_lion.addEventListener(MouseEvent.CLICK, musicPlayLion);

function musicPlayLion(event:MouseEvent):void {
    BTN_print_lion.visible = false;
    var pausePosition:Number = channelLion.position;
    channelLion = sndLion.play(pausePosition);
    transLion.volume = vLion;
    channelLion.soundTransform = transLion;
    timerLion.start();
    timerBarTimerLion.start();
    LionHolder.addEventListener(MouseEvent.CLICK, addShapeLionClick);
    BTN_mute_lion.addEventListener(MouseEvent.CLICK, muteLion);
    BTN_nomute_lion.addEventListener(MouseEvent.CLICK, nomuteLion);
    BTN_lowVolume_lion.addEventListener(MouseEvent.CLICK, lowVolumeLion);
    BTN_addVolume_lion.addEventListener(MouseEvent.CLICK, addVolumeLion);
}

//Volume
var transLion:SoundTransform = new SoundTransform();
vLion = 1.5;
var oSoundLion:Number = new Number;
var oVolumeBarLion:Number = new Number;
oSoundLion = vLion;
MC_volumeBar_lion.gotoAndStop(6);
oVolumeBarLion = MC_volumeBar_lion.currentFrame;
transLion.volume = vLion;
channelLion.soundTransform = transLion;

//addVolume
BTN_addVolume_lion.addEventListener(MouseEvent.CLICK, addVolumeLion);

function addVolumeLion(event:MouseEvent):void {
    MC_volumeBar_lion.nextFrame();
oVolumeBarLion = MC_volumeBar_lion.currentFrame;
vLion+=.3;
oSoundLion = vLion;
    if (vLion > 5) {
        vLion = 5;
    }
    transLion.volume = vLion;
    channelLion.soundTransform = transLion;
}

//lowVolume
BTN_lowVolume_lion.addEventListener(MouseEvent.CLICK, lowVolumeLion);

function lowVolumeLion(event:MouseEvent):void {
    MC_volumeBar_lion.prevFrame();
oVolumeBarLion = MC_volumeBar_lion.currentFrame;
vLion-=.3;
oSoundLion = vLion;
    if (vLion < 0) {
        vLion = 0;
    }
    transLion.volume = vLion;
    channelLion.soundTransform = transLion;
//mute & no mute
BTN_mute_lion.addEventListener(MouseEvent.CLICK, muteLion);

function muteLion(event:MouseEvent):void {
  BTN_lowVolume_lion.removeEventListener(MouseEvent.CLICK, lowVolumeLion);
  BTN_addVolume_lion.removeEventListener(MouseEvent.CLICK, addVolumeLion);
  MC_volumeBar_lion.gotoAndStop(1);
  vLion = 0;
  transLion.volume = vLion;
  channelLion.soundTransform = transLion;
  BTN_mute_lion.visible = false;
  BTN_nomute_lion.visible = true;
}

BTN_nomute_lion.addEventListener(MouseEvent.CLICK, nomuteLion);

function nomuteLion(event:MouseEvent):void {
  BTN_lowVolume_lion.addEventListener(MouseEvent.CLICK, lowVolumeLion);
  BTN_addVolume_lion.addEventListener(MouseEvent.CLICK, addVolumeLion);
  MC_volumeBar_lion.gotoAndStop(oVolumeBarLion);
  vLion = oSoundLion;
  transLion.volume = vLion;
  channelLion.soundTransform = transLion;
  BTN_nomute_lion.visible = false;
  BTN_mute_lion.visible = true;
}

//Show Hide Mousev
LionHolder.addEventListener(MouseEvent.MOUSE_OVER, HideMouseLion);
LionHolder.addEventListener(MouseEvent.MOUSE_OUT, ShowMouseLion);
LionHolder.addEventListener(MouseEvent.MOUSE_DOWN, HideMouseLion);
LionHolder.addEventListener(MouseEvent.MOUSE_UP, HideMouseLion);

var MC_crayon_lion:crayonLion = new crayonLion;
addChild(MC_crayon_lion);
MC_crayon_lion.gotoAndStop(3);

function followMouseLion(event:Event):void {
  MC_crayon_lion.x = mouseX;
  MC_crayon_lion.y = mouseY;
}

LionHolder.addEventListener(MouseEvent.ROLL_OVER, startMouseLion);
function startMouseLion(event:MouseEvent):void {
  stage.addEventListener(Event.ENTER_FRAME, followMouseLion);
}

LionHolder.addEventListener(MouseEvent.ROLL_OUT, stopMouseLion);
function stopMouseLion(event:MouseEvent):void {
  stage.removeEventListener(Event.ENTER_FRAME, followMouseLion);
}

function HideMouseLion(evt:MouseEvent) {
  Mouse.hide();
  MC_crayon_lion.mouseEnabled = false;
}

function ShowMouseLion(evt:MouseEvent) {
  Mouse.show();
}
//Choose Color
var colorNumberLion: Number = 3;
MC_ColorChoice_Lion.gotoAndStop(3);

BTN_color_lion1.addEventListener(MouseEvent.CLICK, colorChangeLionA);
BTN_color_lion2.addEventListener(MouseEvent.CLICK, colorChangeLionB);
BTN_color_lion3.addEventListener(MouseEvent.CLICK, colorChangeLionC);
BTN_color_lion4.addEventListener(MouseEvent.CLICK, colorChangeLionD);
BTN_color_lion5.addEventListener(MouseEvent.CLICK, colorChangeLionE);

function colorChangeLionA(event:MouseEvent): void {
    MC_crayon_lion.gotoAndStop(1);
    MC_ColorChoice_Lion.gotoAndStop(1);
    colorNumberLion = 1;
}

function colorChangeLionB(event:MouseEvent): void {
    MC_crayon_lion.gotoAndStop(2);
    MC_ColorChoice_Lion.gotoAndStop(2);
    colorNumberLion = 2;
}

function colorChangeLionC(event:MouseEvent): void {
    MC_crayon_lion.gotoAndStop(3);
    MC_ColorChoice_Lion.gotoAndStop(3);
    colorNumberLion = 3;
}

function colorChangeLionD(event:MouseEvent): void {
    MC_crayon_lion.gotoAndStop(4);
    MC_ColorChoice_Lion.gotoAndStop(4);
    colorNumberLion = 4;
}

function colorChangeLionE(event:MouseEvent): void {
    MC_crayon_lion.gotoAndStop(5);
    MC_ColorChoice_Lion.gotoAndStop(5);
    colorNumberLion = 5;
}

//Click on Stage
///Number of lions for delete function
var N_Lion: Number = new Number;

///Create an Array to trace the position of the mouse
var positionXLion: Array = new Array;
var NLion: Number = new Number;
NLion = -1;

LionHolder.addEventListener(MouseEvent.CLICK, addShapeLionClick);

function addShapeLionClick(evt: MouseEvent) {
    MC_direction_lion.visible = false;

    ///Random
    var randomLion: Number;
    if (colorNumberLion ==1) {
        randomLion = Math.floor(Math.random()*5)+1;
    }
    if (colorNumberLion ==2) {
        randomLion = Math.floor(Math.random()*5)+6;
    }
}
if (colorNumberLion == 3) {
    randomLion = Math.floor(Math.random() * 5) + 11;
}
if (colorNumberLion == 4) {
    randomLion = Math.floor(Math.random() * 5) + 16;
}
if (colorNumberLion == 5) {
    randomLion = Math.floor(Math.random() * 5) + 21;
}

// create a new object from the Library
var loaderLion:MG_lion = new MG_lion;
loaderLion.gotoAndStop(randomLion);

// Loader Scale & Position
loaderLion.x = mouseX - 50;
loaderLion.y = mouseY - 70;
if (mouseX > 600) {
    loaderLion.scaleX = 0.7;
}
if (mouseX <= 600) {
    loaderLion.scaleX = -0.7;
}
loaderLion.scaleY = 0.7;
loaderLion.cacheAsBitmap = true;
LionHolder.addChild(loaderLion);
N_Lion = LionHolder.numChildren;

//Delete
BTN_delete_lion.addEventListener(MouseEvent.CLICK, deleteShapeLionClick);

function deleteShapeLionClick(event:MouseEvent):void {
    if (N_Lion > 1) {
        LionHolder.removeChildAt(LionHolder.numChildren - 1);
    }
    N_Lion = LionHolder.numChildren;
}

//Print
BTN_print_lion.addEventListener(MouseEvent.CLICK, PrintLion);

function PrintLion(event:MouseEvent):void {
    var printTitleLion:MC_printTitle_lion = new MC_printTitle_lion;
    var LionPrintJob:PrintJob = new PrintJob();
    if (LionPrintJob.start()) {
        try {
            LionHolder.addChild(printTitleLion);
            printTitleLion.x = 25;
            printTitleLion.y = 20;
            LionHolder.scaleX = .5;
            LionHolder.scaleY = .5;
            LionPrintJob.addPage(LionHolder, new Rectangle(7, 7, 1093, 493));
        } catch (error:Error) {
            // Handle error,
        } finally {
            LionPrintJob.send();
            LionHolder.removeChild(printTitleLion);
            LionHolder.scaleX = 1;
            LionHolder.scaleY = 1;
        }
    } else {
        //trace("Print job canceled");
    }
}
// Quit
BTN_quit_lion.addEventListener(MouseEvent.CLICK, LionGotoMenu);

function LionGotoMenu(event:MouseEvent):void {
  gotoAndStop(1, "Menu");
  channelLion.stop();
  removeChild(MC_crayon_lion);
  timerLion.reset();
  timerLion.removeEventListener(TimerEvent.TIMER, onTimerLion);
  timerBackOpenLion.reset();
  timerBackOpenLion.removeEventListener(TimerEvent.TIMER, onTimerBackOpenLion);
  timebarTimerLion.reset();
  timebarTimerLion.removeEventListener(TimerEvent.TIMER, barAddLion);
}

// Timer
var timerLion:Timer = new Timer(85000); timerLion.addEventListener(TimerEvent.TIMER, onTimerLion);
timerLion.start();

function onTimerLion(evt:TimerEvent):void {
  LionHolder.removeEventListener(MouseEvent.CLICK, addShapeLionClick);
  BTN_stop_lion.removeEventListener(MouseEvent.CLICK, musicStopLion);
  BTN_play_lion.removeEventListener(MouseEvent.CLICK, musicPlayLion);
  BTN_delete_lion.removeEventListener(MouseEvent.CLICK, deleteShapeLionClick);
  BTN_print_lion.visible = true;
  timerLion.reset();
  timerLion.dispatchEvent(TimerEvent.TIMER, onTimerLion);
  timerLion.dispatchEvent(TimerEvent.TIMER, onTimerBackOpenLion);
  channelLion.stop();
  MC_timebar_lion.width = 1050;
}

// Timer for going back to Open
var timerBackOpenLion:Timer = new Timer(385000); timerBackOpenLion.addEventListener(TimerEvent.TIMER, onTimerBackOpenLion);
timerBackOpenLion.start();

function onTimerBackOpenLion(evt:TimerEvent):void {
  gotoAndStop(1, "Open");
  timerBackOpenLion.reset();
  timerBackOpenLion.removeEventListener(TimerEvent.TIMER, onTimerBackOpenLion);
  channelLion.stop();
  removeChild(MC_crayon_lion);
Codes for the subject of Aquarium

```javascript
stop();
BTN_print_fish.visible = false;
BTN_nomute_fish.visible = false;

// Add Print Background
var printBGFish:MC_printBG_fish = new MC_printBG_fish;
FishHolder.addChild(printBGFish);

// Time bar
var timebarTimerFish:Timer = new Timer(100);
timebarTimerFish.addEventListener(TimerEvent.TIMER, barAddFish);
timebarTimerFish.start();
MC_timebar_fish.width = 10;

var c:Number = new Number;
c = 1;

function barAddFish(evt:TimerEvent):void {
  c += 20;
  MC_timebar_fish.width = c*0.1;
}

// Music
var sndFish:Sound = new Sound();
sndFish.load(new URLRequest("SongFish.mp3"));
var channelFish:SoundChannel;
channelFish = sndFish.play();

// Sound Deta
var PeakFish:Number = new Number;
stage.addEventListener(Event.ENTER_FRAME, SoundDetaFish);
function SoundDetaFish(event:Event):void {
  PeakFish = Math.floor((channelFish.leftPeak+channelFish.rightPeak)*50);
  MC_peak_fish.scaleX = PeakFish*0.015;
  if (MC_peak_fish.width > 150) {
    MC_peak_fish.width = 150;
  }
  if (PeakFish == 0) {
    PeakFish = Math.floor(Math.random()*10)+1;
  }
}

// Music Stop
BTN_stop_fish.addEventListener(MouseEvent.CLICK, musicStopFish);
function musicStopFish(event:MouseEvent):void {
  BTN_print_fish.visible = true;
  var pausePosFish:Number = channelFish.position;
  channelFish.stop();
  transFish.volume = vFish;
  channelFish.soundTransform = transFish;
  timerFish.stop();
timebarTimerFish.stop();
timerBackOpenFish.stop();
FishHolder.removeEventListener(MouseEvent.CLICK, addShapeFishClick);
BTN_mute_fish.removeEventListener(MouseEvent.CLICK, muteFish);
BTN_nomute_fish.removeEventListener(MouseEvent.CLICK, nomuteFish);
BTN_lowVolume_fish.removeEventListener(MouseEvent.CLICK, lowVolumeFish);
BTN_addVolume_fish.removeEventListener(MouseEvent.CLICK, addVolumeFish);
```
//Music Play
BTN_play_fish.addEventListener(MouseEvent.CLICK, musicPlayFish);

function musicPlayFish(event:MouseEvent):void {
    BTN_print_fish.visible = false;
    var pausePosFish:Number = channelFish.position;
    channelFish = sndFish.play(pausePosFish);
    transFish.volume = vFish;
    channelFish.soundTransform = transFish;
    timerFish.start();
    timebarTimerFish.start();
    FishHolder.addEventListener(MouseEvent.CLICK, addShapeFishClick);
    BTN_mute_fish.addEventListener(MouseEvent.CLICK, muteFish);
    BTN_nomute_fish.addEventListener(MouseEvent.CLICK, nomuteFish);
    BTN_lowVolume_fish.addEventListener(MouseEvent.CLICK, lowVolumeFish);
    BTN_addVolume_fish.addEventListener(MouseEvent.CLICK, addVolumeFish);
}

//Volume
var transFish:SoundTransform = new SoundTransform();
var vFish:Number = 1.5;
var oSoundFish:Number = new Number;
var oVolumeBarFish:Number = new Number;
oSoundFish = vFish;
MC_volumeBar_fish.gotoAndStop(6);
oVolumeBarFish = MC_volumeBar_fish.currentFrame;
transFish.volume = vFish;
channelFish.soundTransform = transFish;

//addVolume
BTN_addVolume_fish.addEventListener(MouseEvent.CLICK, addVolumeFish);

function addVolumeFish(event:MouseEvent):void {
    MC_volumeBar_fish.nextFrame();
oVolumeBarFish = MC_volumeBar_fish.currentFrame;
vFish+=.3;
oSoundFish = vFish;
    if (vFish > 5) { vFish = 5; }
    transFish.volume = vFish;
    channelFish.soundTransform = transFish;
}

//lowVolume
BTN_lowVolume_fish.addEventListener(MouseEvent.CLICK, lowVolumeFish);

function lowVolumeFish(event:MouseEvent):void {
    MC_volumeBar_fish.prevFrame();
oVolumeBarFish = MC_volumeBar_fish.currentFrame;
vFish-=.3;
oSoundFish = vFish;
    if (vFish < 0) { vFish = 0; }
transFish.volume = vFish;
    channelFish.soundTransform = transFish;
function muteFish(event:MouseEvent):void {
    BTN_lowVolume_fish.removeEventListener(MouseEvent.CLICK, lowVolumeFish);
    BTN_addVolume_fish.addEventListener(MouseEvent.CLICK, addVolumeFish);
    MC_volumeBar_fish.gotoAndStop(1);
    vFish = 0;
    transFish.volume = vFish;
    channelFish.soundTransform = transFish;
    BTN_mute_fish.visible = false;
    BTN_nomute_fish.visible = true;
}

function nomuteFish(event:MouseEvent):void {
    BTN_lowVolume_fish.addEventListener(MouseEvent.CLICK, lowVolumeFish);
    BTN_addVolume_fish.addEventListener(MouseEvent.CLICK, addVolumeFish);
    MC_volumeBar_fish.gotoAndStop(oVolumeBarFish);
    vFish = oSoundFish;
    transFish.volume = vFish;
    channelFish.soundTransform = transFish;
    BTN_nomute_fish.visible = false;
    BTN_mute_fish.visible = true;
}

var MC_crayon_fish:crayonFish = new crayonFish;
addChild(MC_crayon_fish);
MC_crayon_fish.gotoAndStop(3);

function followMouseFish(event:Event):void {
    MC_crayon_fish.x = mouseX;
    MC_crayon_fish.y = mouseY;
}

FishHolder.addEventListener(MouseEvent.MOUSE_OVER, startMouseFish);
function startMouseFish(event:MouseEvent):void {
    stage.addEventListener(Event.ENTER_FRAME, followMouseFish);
}

FishHolder.addEventListener(MouseEvent.MOUSE_OUT, stopMouseFish);
function stopMouseFish(event:MouseEvent):void {
    stage.removeEventListener(Event.ENTER_FRAME, followMouseFish);
}

function HideMouseFish(evt:MouseEvent) {
    Mouse.hide();
    MC_crayon_fish.mouseEnabled = false;
}

function ShowMouseFish(evt:MouseEvent) {
    Mouse.show();
}
//Choose Color

var colorNumberFish:Number = 3;
MC_ColorChoice_Fish.gotoAndStop(3);
BTN_color_fish1.addEventListener(MouseEvent.CLICK, colorChangeFishA);
BTN_color_fish2.addEventListener(MouseEvent.CLICK, colorChangeFishB);
BTN_color_fish3.addEventListener(MouseEvent.CLICK, colorChangeFishC);
BTN_color_fish4.addEventListener(MouseEvent.CLICK, colorChangeFishD);
BTN_color_fish5.addEventListener(MouseEvent.CLICK, colorChangeFishE);

function colorChangeFishA(event:MouseEvent):void {
    MC_crayon_fish.gotoAndStop(1);
    MC_ColorChoice_Fish.gotoAndStop(1);
    colorNumberFish = 1;
}

function colorChangeFishB(event:MouseEvent):void {
    MC_crayon_fish.gotoAndStop(2);
    MC_ColorChoice_Fish.gotoAndStop(2);
    colorNumberFish = 2;
}

function colorChangeFishC(event:MouseEvent):void {
    MC_crayon_fish.gotoAndStop(3);
    MC_ColorChoice_Fish.gotoAndStop(3);
    colorNumberFish = 3;
}

function colorChangeFishD(event:MouseEvent):void {
    MC_crayon_fish.gotoAndStop(4);
    MC_ColorChoice_Fish.gotoAndStop(4);
    colorNumberFish = 4;
}

function colorChangeFishE(event:MouseEvent):void {
    MC_crayon_fish.gotoAndStop(5);
    MC_ColorChoice_Fish.gotoAndStop(5);
    colorNumberFish = 5;
}

//Click on Stage

//Number of fishs for delete function
var N_Fish:Number = new Number;

//Create an Array to trace the position of the mouse
var positionXFish:Array = new Array;
var Nfish:Number = new Number;
Nfish = -1;
FishHolder.addEventListener(MouseEvent.CLICK, addShapeFishClick);

function addShapeFishClick(evt:MouseEvent) {
    MC_direction_fish.visible = false;

    //Random
    var randomFish:Number;
    if (colorNumberFish ==1) {
        randomFish = Math.floor(Math.random()*5)+1;
    }
    if (colorNumberFish ==2) {
        randomFish = Math.floor(Math.random()*5)+6;
    }
}
if (colorNumberFish == 3) {
    randomFish = Math.floor(Math.random()*5)+11;
}
if (colorNumberFish == 4) {
    randomFish = Math.floor(Math.random()*5)+16;
}
if (colorNumberFish == 5) {
    randomFish = Math.floor(Math.random()*5)+21;
}

// create a new object from the Library
var loaderFish: MG_fish = new MG_fish;
loaderFish.gotoAndStop(randomFish);

// Loader Scale & Position
loaderFish.x = mouseX-50;
loaderFish.y = mouseY-70;
if (mouseX > 600) {
    loaderFish.scaleX = 0.7;
}
if (mouseX <= 600) {
    loaderFish.scaleX = -0.7;
}
loaderFish.scaleY = 0.7;
loaderFish.cacheAsBitmap = true;
FishHolder.addChild(loaderFish);
N_Fish = FishHolder.numChildren;

//Delete
BTN_delete_fish.addEventListener(MouseEvent.CLICK, deleteShapeFishClick);

function deleteShapeFishClick(event:MouseEvent): void {
    if (N_Fish > 1) {
        FishHolder.removeChildAt(FishHolder.numChildren -1);
    }
    N_Fish = FishHolder.numChildren;
}

//Print
BTN_print_fish.addEventListener(MouseEvent.CLICK, PrintFish);

function PrintFish(event:MouseEvent): void {
    var printTitleFish: MC_printTitle_fish = new MC_printTitle_fish;
    var FishPrintJob: PrintJob = new PrintJob();
    if (FishPrintJob.start()) {
        try {
            FishHolder.addChild(printTitleFish);
            printTitleFish.x = 25;
            printTitleFish.y = 20;
            FishHolder.scaleX = .5;
            FishHolder.scaleY = .5;
            FishPrintJob.addPage(FishHolder, new Rectangle(7, 7, 1093, 493));
        } catch (error: Error) {
            // Handle error,
        }
        FishPrintJob.send();
    }
    else {
        //trace("Print job canceled");
    }
}
// Quit
BTN_quit_fish.addEventListener(MouseEvent.CLICK, FishGotoMenu);

function FishGotoMenu(event:MouseEvent):void {
    gotoAndStop(1, "Menu");
    channelFish.stop();
    removeChild(MC_crayon_fish);
    timerFish.reset();
timerFish.removeEventListener(TimerEvent.TIMER, onTimerFish);
timerBackOpenFish.reset();
timerBackOpenFish.removeEventListener(TimerEvent.TIMER, onTimerBackOpenFish);
timebarTimerFish.reset();
timebarTimerFish.removeEventListener(TimerEvent.TIMER, barAddFish);
}

// Timer
var timerFish:Timer = new Timer(85000);
timerFish.addEventListener(TimerEvent.TIMER, onTimerFish);
timerFish.start();

function onTimerFish(evt:TimerEvent):void {
    FishHolder.removeEventListener(MouseEvent.CLICK, addShapeFishClick);
    BTN_stop_fish.removeEventListener(MouseEvent.CLICK, musicStopFish);
    BTN_play_fish.removeEventListener(MouseEvent.CLICK, musicPlayFish);
    BTN_delete_fish.removeEventListener(MouseEvent.CLICK, deleteShapeFishClick);
    BTN_print_fish.visible = true;
timerFish.reset();
timerFish.removeEventListener(TimerEvent.TIMER, onTimerFish);
timebarTimerFish.reset();
timebarTimerFish.removeEventListener(TimerEvent.TIMER, barAddFish);
MC_timebar_fish.width = 1050;
}

// Timer for going back to Open
var timerBackOpenFish:Timer = new Timer(385000);
timerBackOpenFish.addEventListener(TimerEvent.TIMER, onTimerBackOpenFish);
timerBackOpenFish.start();

function onTimerBackOpenFish(evt:TimerEvent):void {
    gotoAndStop(1, "Open");
timerBackOpenFish.reset();
timerBackOpenFish.removeEventListener(TimerEvent.TIMER, onTimerBackOpenFish);
channelFish.stop();
removeChild(MC_crayon_fish);
Codes for the subject of the Swan

```javascript
stop();
BTN_print_swan.visible = false;
BTN_nomute_swan.visible = false;

// Add Print Background
var printBGSwan:MC_printBG_swan = new MC_printBG_swan;
SwanHolder.addChild(printBGSwan);

// Time bar
var timebarTimerSwan:Timer = new Timer(100);
timebarTimerSwan.addEventListener(TimerEvent.TIMER, barAddSwan);
timebarTimerSwan.start();
MC_timebar_swan.width = 10;
var b:Number = new Number();
b = 1;

function barAddSwan(evt:TimerEvent):void {
    b += 20;
    MC_timebar_swan.width = b*0.1;
}

// Music
var sndSwan:Sound = new Sound();
sndSwan.load(new URLRequest("SongSwan.mp3"));
var channelSwan:SoundChannel;
channelSwan = sndSwan.play();

// Sound Data
var PeakSwan:Number = new Number();
stage.addEventListener(Event.ENTER_FRAME, SoundDetaSwan);
function SoundDetaSwan(event:Event):void {
    PeakSwan = Math.floor(channelSwan.leftPeak+channelSwan.rightPeak)*50;
    MC_peak_swan.scaleX = PeakSwan*0.015;
    if (MC_peak_swan.width > 150) {
        MC_peak_swan.width = 150;
    }
    if (PeakSwan == 0) {
        PeakSwan = Math.floor(Math.random()*10)+1;
    }
}

// Music Stop
BTN_stop_swan.addEventListener(MouseEvent.CLICK, musicStopSwan);
function musicStopSwan(event:MouseEvent):void {
    BTN_print_swan.visible = true;
    var pausePosSwan:Number = channelSwan.position;
    channelSwan.stop();
    transSwan volume = vSwan;
    channelSwan.soundTransform = transSwan;
    timerSwan.stop();
timebarTimerSwan.stop();
timerBackOpenSwan.stop();
SwanHolder.removeEventListener(MouseEvent.CLICK, addShapeSwanClick);
BTN_mute_swan.removeEventListener(MouseEvent.CLICK, muteSwan);
BTN_nomute_swan.removeEventListener(MouseEvent.CLICK, nomuteSwan);
BTN_lowVolume_swan.removeEventListener(MouseEvent.CLICK, lowVolumeSwan);
BTN_addVolume_swan.removeEventListener(MouseEvent.CLICK, addVolumeSwan);
```
//Music Play
BTN_play_swan.addEventListener(MouseEvent.CLICK, musicPlaySwan);

function musicPlaySwan(event:MouseEvent):void {
    BTN_print_swan.visible = false;
    var pausePosSwan:Number = channelSwan.position;
    channelSwan = sndSwan.play(pausePosSwan);
    transSwan.volume = vSwan;
    channelSwan.soundTransform = transSwan;
    timerSwan.start();
    timerBarTimerSwan.start();
    SwanHolder.addEventListener(MouseEvent.CLICK, addShapeSwanClick);
    BTN_mute_swan.addEventListener(MouseEvent.CLICK, muteSwan);
    BTN_nomute_swan.addEventListener(MouseEvent.CLICK, nomuteSwan);
    BTN_lowVolume_swan.addEventListener(MouseEvent.CLICK, lowVolumeSwan);
    BTN_addVolume_swan.addEventListener(MouseEvent.CLICK, addVolumeSwan);
}

//Volume
var transSwan:SoundTransform = new SoundTransform();
var vSwan:Number = 1.5;
var oSoundSwan:Number = new Number;
var oVolumeBarSwan:Number = new Number;
oSoundSwan = vSwan;
MC_volumeBar_swan.gotoAndStop(6);
oVolumeBarSwan = MC_volumeBar_swan.currentFrame;
transSwan.volume = vSwan;
channelSwan.soundTransform = transSwan;

//addVolume
BTN_addVolume_swan.addEventListener(MouseEvent.CLICK, addVolumeSwan);

function addVolumeSwan(event:MouseEvent):void {
    MC_volumeBar_swan.nextFrame();
oVolumeBarSwan = MC_volumeBar_swan.currentFrame;
vSwan+=.3;
oSoundSwan = vSwan;
if (vSwan > 5) {
    vSwan = 5;
}
transSwan.volume = vSwan;
channelSwan.soundTransform = transSwan;
}

//lowVolume
BTN_lowVolume_swan.addEventListener(MouseEvent.CLICK, lowVolumeSwan);

function lowVolumeSwan(event:MouseEvent):void {
    MC_volumeBar_swan.prevFrame();
oVolumeBarSwan = MC_volumeBar_swan.currentFrame;
vSwan-=.3;
oSoundSwan = vSwan;
if (vSwan < 0) {
    vSwan = 0;
}
transSwan.volume = vSwan;
channelSwan.soundTransform = transSwan;
}
//mute & no mute
BTN_mute_swan.addEventListener(MouseEvent.CLICK, muteSwan);

function muteSwan(event:MouseEvent):void {
    BTN_lowVolume_swan.removeEventListener(MouseEvent.CLICK, lowVolumeSwan);
    BTN_addVolume_swan.removeEventListener(MouseEvent.CLICK, addVolumeSwan);
    MC_volumeBar_swan.gotoAndStop(1);
    vSwan = 0;
    transSwan.volume = vSwan;
    channelSwan.soundTransform = transSwan;
    BTN_mute_swan.visible = false;
    BTN_nomute_swan.visible = true;
}

 BTN_nomute_swan.addEventListener(MouseEvent.CLICK, nomuteSwan);

function nomuteSwan(event:MouseEvent):void {
    BTN_lowVolume_swan.addEventListener(MouseEvent.CLICK, lowVolumeSwan);
    BTN_addVolume_swan.addEventListener(MouseEvent.CLICK, addVolumeSwan);
    MC_volumeBar_swan.gotoAndStop(oVolumeBarSwan);
    vSwan = oSoundSwan;
    transSwan.volume = vSwan;
    channelSwan.soundTransform = transSwan;
    BTN_nomute_swan.visible = false;
    BTN_mute_swan.visible = true;
}

//Show Hide Mousev
SwanHolder.addEventListener(MouseEvent.MOUSE_OVER, HideMouseSwan);
SwanHolder.addEventListener(MouseEvent.MOUSE_OUT, ShowMouseSwan);
SwanHolder.addEventListener(MouseEvent.MOUSE_DOWN, HideMouseSwan);
SwanHolder.addEventListener(MouseEvent.MOUSE_UP, HideMouseSwan);

var MC_crayon_swan:crayonSwan = new crayonSwan;
addChild(MC_crayon_swan);
MC_crayon_swan.gotoAndStop(3);

function followMouseSwan(event:Event):void {
    MC_crayon_swan.x = mouseX;
    MC_crayon_swan.y = mouseY;
}

SwanHolder.addEventListener(MouseEvent.ROLL_OVER, startMouseSwan);
function startMouseSwan(event:MouseEvent):void {
    stage.addEventListener(Event.ENTER_FRAME, followMouseSwan);
}

SwanHolder.addEventListener(MouseEvent.ROLL_OUT, stopMouseSwan);
function stopMouseSwan(event:MouseEvent):void {
    stage.removeEventListener(Event.ENTER_FRAME, followMouseSwan);
}

function HideMouseSwan(evt:MouseEvent) {
    Mouse.hide();
    MC_crayon_swan.mouseEnabled = false;
}

function ShowMouseSwan(evt:MouseEvent) {
    Mouse.show();
}
//Choose Color
var colorNumberSwan: Number = 3;
MC_ColorChoice_Swan.gotoAndStop(3);

BTN_color_swan1.addEventListener(MouseEvent.CLICK, colorChangeSwanA);
BTN_color_swan2.addEventListener(MouseEvent.CLICK, colorChangeSwanB);
BTN_color_swan3.addEventListener(MouseEvent.CLICK, colorChangeSwanC);
BTN_color_swan4.addEventListener(MouseEvent.CLICK, colorChangeSwanD);
BTN_color_swan5.addEventListener(MouseEvent.CLICK, colorChangeSwanE);

function colorChangeSwanA(event: MouseEvent): void {
    MC_crayon_swan.gotoAndStop(1);
    MC_ColorChoice_Swan.gotoAndStop(1);
    colorNumberSwan = 1;
}

function colorChangeSwanB(event: MouseEvent): void {
    MC_crayon_swan.gotoAndStop(2);
    MC_ColorChoice_Swan.gotoAndStop(2);
    colorNumberSwan = 2;
}

function colorChangeSwanC(event: MouseEvent): void {
    MC_crayon_swan.gotoAndStop(3);
    MC_ColorChoice_Swan.gotoAndStop(3);
    colorNumberSwan = 3;
}

function colorChangeSwanD(event: MouseEvent): void {
    MC_crayon_swan.gotoAndStop(4);
    MC_ColorChoice_Swan.gotoAndStop(4);
    colorNumberSwan = 4;
}

function colorChangeSwanE(event: MouseEvent): void {
    MC_crayon_swan.gotoAndStop(5);
    MC_ColorChoice_Swan.gotoAndStop(5);
    colorNumberSwan = 5;
}

//Click on Stage
//Number of swans for delete function
var N_Swan: Number = new Number;

//Create a Array to trace the position of the mouse
var positionXSwan: Array = new Array;
var NSwan: Number = new Number;
NSwan = -1;
SwanHolder.addEventListener(MouseEvent.CLICK, addShapeSwanClick);

function addShapeSwanClick(evt: MouseEvent) {
    MC_direction_swan.visible = false;

    //Random
    var randomSwan: Number;
    if (colorNumberSwan == 1) {
        randomSwan = Math.floor(Math.random() * 5) + 1;
    }
    if (colorNumberSwan == 2) {
        randomSwan = Math.floor(Math.random() * 5) + 6;
    }
if (colorNumberSwan == 3) {
    randomSwan = Math.floor(Math.random() * 5) + 11;
}
if (colorNumberSwan == 4) {
    randomSwan = Math.floor(Math.random() * 5) + 16;
}
if (colorNumberSwan == 5) {
    randomSwan = Math.floor(Math.random() * 5) + 21;
}

// create a new object from the Library
var loaderSwan: MG_swan = new MG_swan;
loaderSwan.gotoAndStop(randomSwan);

// Loader Scale & Position
loaderSwan.x = mouseX - 50;
loaderSwan.y = mouseY - 70;

if (mouseX > 600) {
    loaderSwan.scaleX = 0.7;
}
if (mouseX <= 600) {
    loaderSwan.scaleX = -0.7;
}
loaderSwan.scaleY = 0.7;
loaderSwan.cacheAsBitmap = true;
SwanHolder.addChild(loaderSwan);
N_Swan = SwanHolder.numChildren;

//Delete
BTN_delete_swan.addEventListener(MouseEvent.CLICK, deleteShapeSwanClick);

function deleteShapeSwanClick(event: MouseEvent): void {
    if (N_Swan > 1) {
        SwanHolder.removeChild(SwanHolder.numChildren - 1);
    }
    N_Swan = SwanHolder.numChildren;
}

//Print
BTN_print_swan.addEventListener(MouseEvent.CLICK, PrintSwan);

function PrintSwan(event: MouseEvent): void {
    var printTitleSwan: MC_printTitle_swan = new MC_printTitle_swan;
    var SwanPrintJob: PrintJob = new PrintJob();
    if (SwanPrintJob.start()) {
        try {
            SwanHolder.addChild(printTitleSwan);
            printTitleSwan.x = 25;
            printTitleSwan.y = 20;
            SwanHolder.scaleX = .5;
            SwanHolder.scaleY = .5;
            SwanPrintJob.addPage(SwanHolder, new Rectangle(7, 7, 1093, 493));
        } catch (error: Error) {
            // Handle error,
        }
        SwanPrintJob.send();
        SwanHolder.removeChild(printTitleSwan);
        SwanHolder.scaleX = 1;
        SwanHolder.scaleY = 1;
    } else {
        //trace("Print job canceled");
    }
}
BTN_quit_swan.addEventListener(MouseEvent.CLICK, SwanGotoMenu);

function SwanGotoMenu(event:MouseEvent):void {
  gotoAndStop(1, "Menu");
  channelSwan.stop();
  removeChild(MC_crayon_swan);
  timerSwan.reset();
  timerSwan.removeEventListener(TimerEvent.TIMER, onTimerSwan);
  timerBackOpenSwan.reset();
  timerBackOpenSwan.removeEventListener(TimerEvent.TIMER, onTimerBackOpenSwan);
  timebarTimerSwan.reset();
  timebarTimerSwan.removeEventListener(TimerEvent.TIMER, barAddSwan);
}

//Timer
var timerSwan:Timer = new Timer(85000);
timerSwan.addEventListener(TimerEvent.TIMER, onTimerSwan);
timerSwan.start();

function onTimerSwan(evt:TimerEvent):void {
  SwanHolder.removeEventListener(MouseEvent.CLICK, addShapeSwanClick);
  BTN_stop_swan.removeEventListener(MouseEvent.CLICK, musicStopSwan);
  BTN_play_swan.removeEventListener(MouseEvent.CLICK, musicPlaySwan);
  BTN_delete_swan.removeEventListener(MouseEvent.CLICK, deleteShapeSwanClick);
  BTN_print_swan.visible = true;
  timerSwan.reset();
  timerSwan.removeEventListener(TimerEvent.TIMER, onTimerSwan);
  timebarTimerSwan.reset();
  timebarTimerSwan.removeEventListener(TimerEvent.TIMER, barAddSwan);
  MC_timebar_swan.width = 1050;
}

//Timer for going back to Open
var timerBackOpenSwan:Timer = new Timer(385000);
timerBackOpenSwan.addEventListener(TimerEvent.TIMER, onTimerBackOpenSwan);
timerBackOpenSwan.start();

function onTimerBackOpenSwan(evt:TimerEvent):void {
  gotoAndStop(1, "Open");
  timerBackOpenSwan.reset();
  timerBackOpenSwan.removeEventListener(TimerEvent.TIMER, onTimerBackOpenSwan);
  channelSwan.stop();
  removeChild(MC_crayon_swan);
Bibliography

Annette W. Balkema, Henk Slager. Screen-Based Art. Rodopi, 2000

Gary Rosenzweig. ActionScript 3.0 Game Programming University. QUE CORP, 2007

Green, Tom and Dias, Tiago. From After Effects to Flash: Poetry in Motion Graphics. Friends of ED, 2006


