
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 Programming, it is described how to use a visual programming 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.

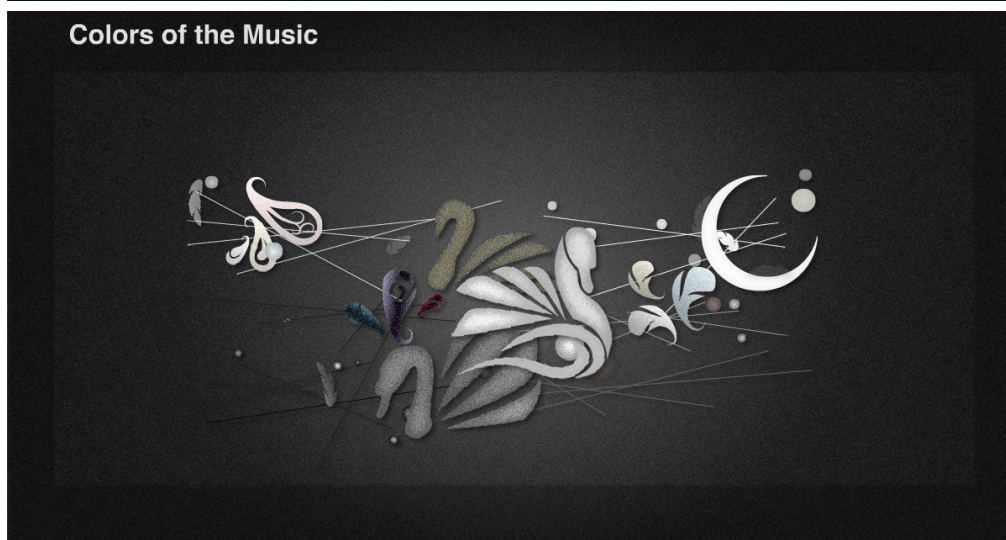
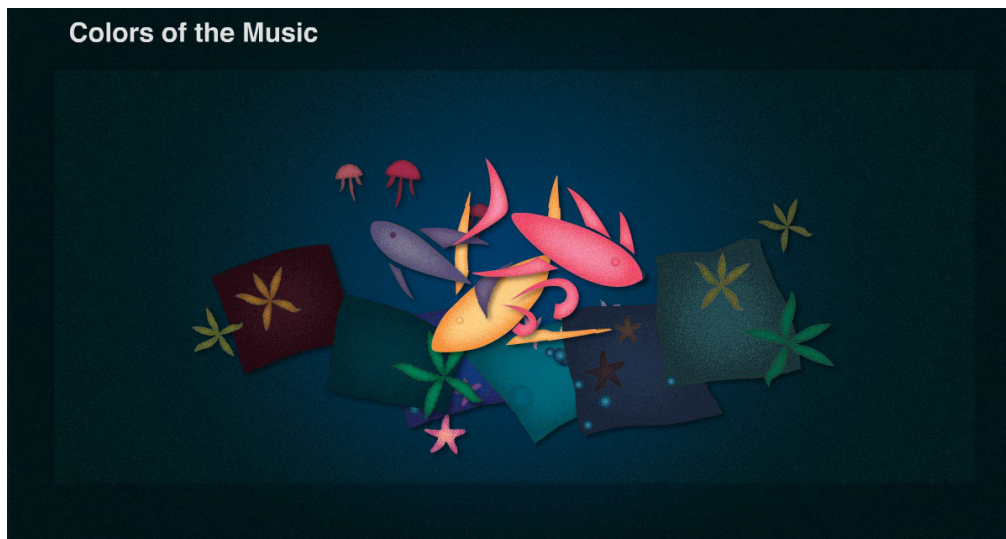
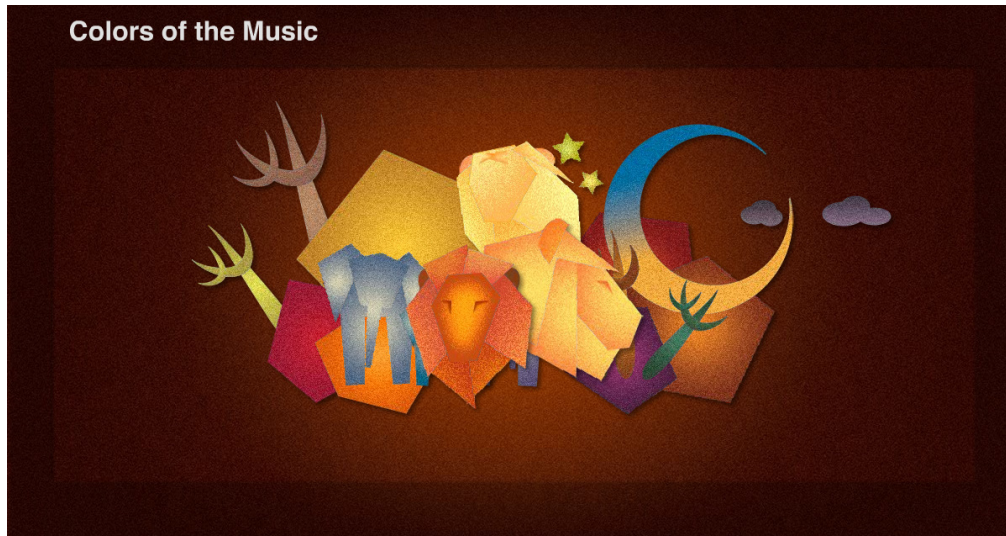
1. Loop the Short Movie

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 tool bar 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

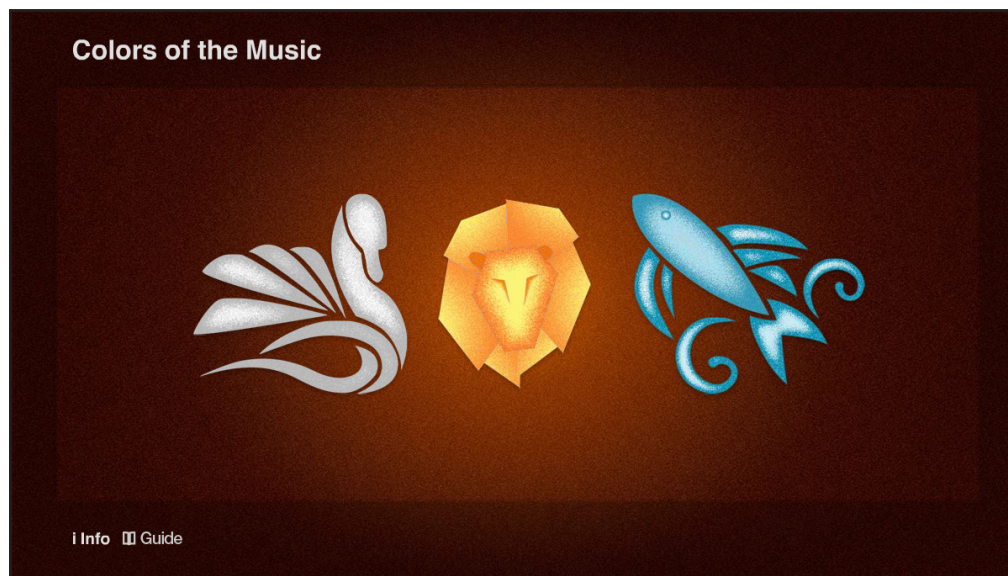
By clicking the button showing “info”, users can get brief information about the music and this project from a popping up window. Click the icon for closing to close the information window.

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.

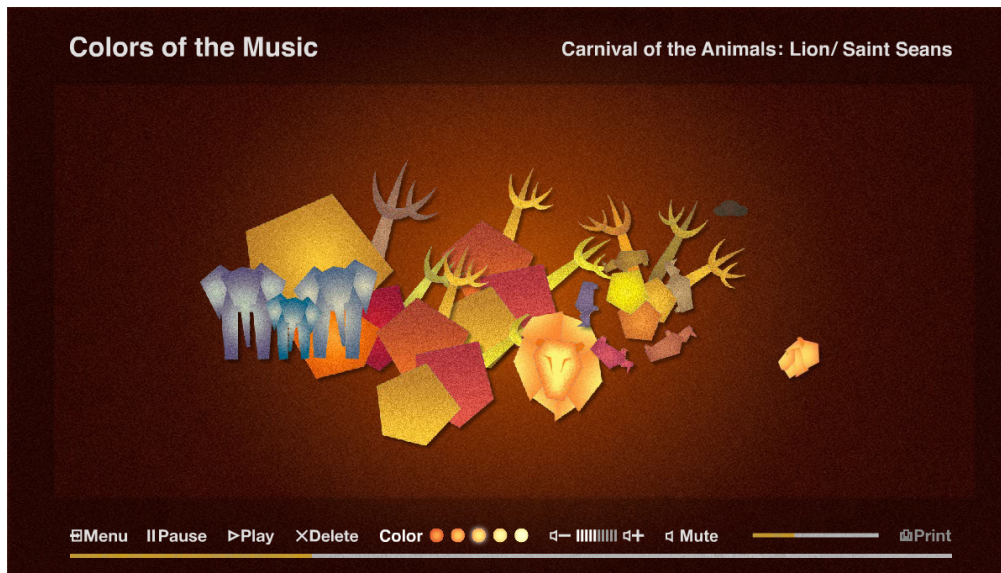
4. Timer for Returning to the Beginning

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.

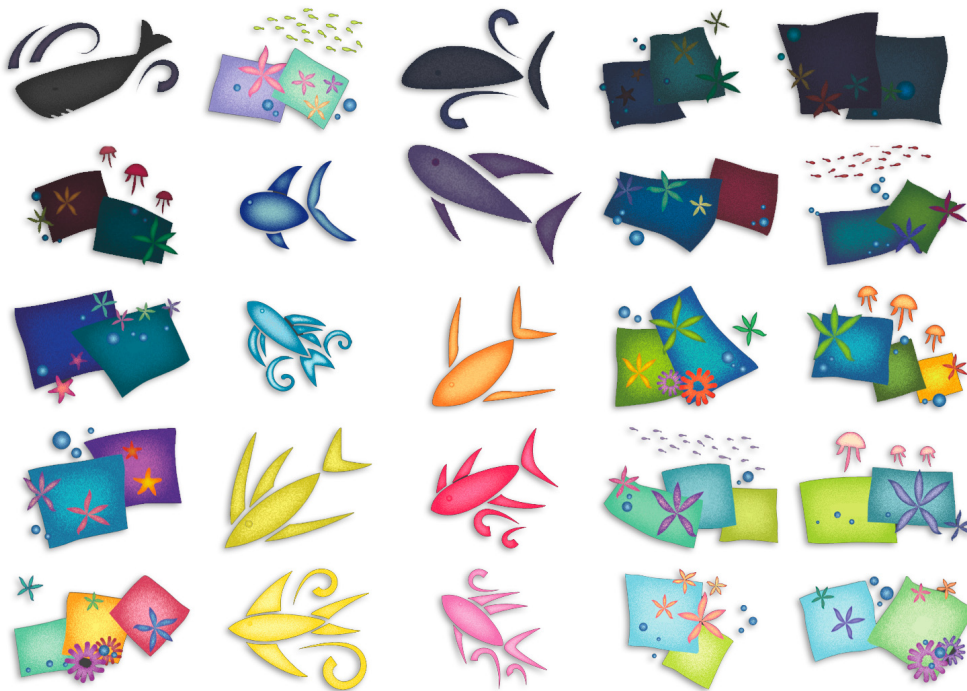
3. The Swan

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 interactivity 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

```
1  stop();
2
3  var loader:Loader = new Loader();
4  loader.load(new URLRequest("Open.swf"));
5  addChild(loader);
6
7  var positionX:Array = new Array;
8  var N:Number = new Number;
9  N = -1;
10
11 var timerOpen:Timer = new Timer(1000);
12 timerOpen.addEventListener(TimerEvent.TIMER, onTimerOpen);
13 timerOpen.start();
14
15 function onTimerOpen(evt:TimerEvent):void {
16     var MOUSEX:Number = new Number;
17     MOUSEX = mouseX;
18     N++;
19     positionX.push(MOUSEX);
20
21     if (positionX[N]>positionX[N-1]) {
22         gotoAndPlay(1, "Menu");
23         removeChild(loader);
24         timerOpen.reset();
25         timerOpen.removeEventListener(TimerEvent.TIMER, onTimerOpen);
26     }
27     if (positionX[N]<positionX[N-1]) {
28         gotoAndPlay(1, "Menu");
29         removeChild(loader);
30         timerOpen.reset();
31         timerOpen.removeEventListener(TimerEvent.TIMER, onTimerOpen);
32     }
33 }
```

Codes for the menu

```
1  stop();
2
3  //Timer
4  var timer:Timer = new Timer(60000);
5  timer.addEventListener(TimerEvent.TIMER, onTimer);
6  timer.start();
7
8  function onTimer(evt:TimerEvent):void {
9      gotoAndStop(1, "Open");
10     timer.reset();
11     timer.removeEventListener(TimerEvent.TIMER, onTimer);
12     removeChild(Intro);
13     removeChild(Close);
14 }
15
16 //Go To Drawing Stage
17 function DrawingLion(event:MouseEvent):void {
18     gotoAndPlay(1, "LionStage");
19     timer.reset();
20     timer.removeEventListener(TimerEvent.TIMER, onTimer);
21     removeChild(Intro);
22     removeChild(Close);
23 }
24
25 function DrawingSwan(event:MouseEvent):void {
26     gotoAndPlay(1, "SwanStage");
27     timer.reset();
28     timer.removeEventListener(TimerEvent.TIMER, onTimer);
29     removeChild(Intro);
30     removeChild(Close);
31 }
32
33 function DrawingFish(event:MouseEvent):void {
34     gotoAndPlay(1, "FishStage");
35     timer.reset();
36     timer.removeEventListener(TimerEvent.TIMER, onTimer);
37     removeChild(Intro);
38     removeChild(Close);
39 }
40
41 //Show info
42 var Intro:MC_intro = new MC_intro;
43 addChild(Intro);
44 Intro.x = 339.1;
45 Intro.y = 177.5;
46 Intro.visible = false;
47
48 var Close:BTN_close_btn = new BTN_close_btn;
49 addChild(Close);
50 Close.x = 567;
51 Close.y = 450;
52 Close.visible = false;
53
54 function ShowInfo(event:MouseEvent):void {
55     Intro.visible = true;
56     Close.visible = true;
57 }
58
```

```
59 function CloseInfo(event:MouseEvent):void {
60     Intro.visible = false;
61     Close.visible = false;
62 }
63
64 //Mouse Click
65 BTN_lion.addEventListener(MouseEvent.CLICK, DrawingLion);
66 BTN_swan.addEventListener(MouseEvent.CLICK, DrawingSwan);
67 BTN_fish.addEventListener(MouseEvent.CLICK, DrawingFish);
68 BTN_info.addEventListener(MouseEvent.CLICK, ShowInfo);
69 Close.addEventListener(MouseEvent.CLICK, CloseInfo);
70
71 //Background Change
72 var BG_Lion:MenuBG_Lion = new MenuBG_Lion;
73 BG_Lion.gotoAndPlay(1);
74 MenuBG.addChild(BG_Lion);
75
76 BTN_fish.addEventListener(MouseEvent.MOUSE_OVER, changeBGfish);
77 function changeBGfish(event:MouseEvent):void {
78     var BG_Fish:MenuBG_Fish = new MenuBG_Fish;
79     BG_Fish.gotoAndPlay(1);
80     MenuBG.addChild(BG_Fish);
81 }
82
83 BTN_lion.addEventListener(MouseEvent.MOUSE_OVER, changeBGLion);
84 function changeBGLion(event:MouseEvent):void {
85     var BG_Lion:MenuBG_Lion = new MenuBG_Lion;
86     BG_Lion.gotoAndPlay(1);
87     MenuBG.addChild(BG_Lion);
88 }
89
90 BTN_swan.addEventListener(MouseEvent.MOUSE_OVER, changeBGswan);
91 function changeBGswan(event:MouseEvent):void {
92     var BG_Swan:MenuBG_Swan = new MenuBG_Swan;
93     BG_Swan.gotoAndPlay(1);
94     MenuBG.addChild(BG_Swan);
95 }
```

Codes for the subject of Royal March of Lion

```

1  stop();
2  BTN_print_lion.visible = false;
3  BTN_nomute_lion.visible = false;
4
5  //Add Print Background
6  var printBGLion:MC_printBG_lion = new MC_printBG_lion;
7  LionHolder.addChild(printBGLion);
8
9  //Time bar
10 var timebarTimerLion:Timer = new Timer(100);
11 timebarTimerLion.addEventListener(TimerEvent.TIMER, barAddLion);
12 timebarTimerLion.start();
13 MC_timebar_lion.width = 10;
14
15 var a:Number = new Number;
16 a = 1;
17
18 function barAddLion(evt:TimerEvent):void {
19     a +=20;
20     MC_timebar_lion.width = a*0.1;
21 }
22
23 //Music
24 var sndLion:Sound = new Sound();
25 sndLion.load(new URLRequest("SongLion.mp3"));
26
27 var channellion:SoundChannel;
28 channellion = sndLion.play();
29
30 //Sound Deta
31 var PeakLion:Number = new Number;
32 stage.addEventListener(Event.ENTER_FRAME, SoundDetaLion);
33 function SoundDetaLion(event:Event):void {
34     PeakLion = Math.floor((channellion.leftPeak+channellion.rightPeak)*50);
35     MC_peak_lion.scaleX = PeakLion*0.015;
36     if (MC_peak_lion.width > 150) {
37         MC_peak_lion.width = 150;
38     }
39     if (PeakLion == 0) {
40         PeakLion = Math.floor(Math.random()*10)+1;
41     }
42 }
43
44 //Music Stop
45 BTN_stop_lion.addEventListener(MouseEvent.CLICK, musicStopLion);
46
47 function musicStopLion(event:MouseEvent):void {
48     BTN_print_lion.visible = true;
49     var pausePosLion:Number = channellion.position;
50     channellion.stop();
51     transLion.volume = vLion;
52     channellion.soundTransform = transLion;
53     timerLion.stop();
54     timebarTimerLion.stop();
55     timerBackOpenLion.stop();
56     LionHolder.removeEventListener(MouseEvent.CLICK, addShapelionClick);
57     BTN_mute_lion.removeEventListener(MouseEvent.CLICK, mutelion);
58     BTN_nomute_lion.removeEventListener(MouseEvent.CLICK, nomuteLion);
59     BTN_lowVolume_lion.removeEventListener(MouseEvent.CLICK, lowVolumelion);
60     BTN_addVolume_lion.removeEventListener(MouseEvent.CLICK, addVolumelion);
61 }
62

```

```
63 //Music Play
64 BTN_play_lion.addEventListener(MouseEvent.CLICK, musicPlayLion);
65
66 function musicPlayLion(event:MouseEvent):void {
67     BTN_print_lion.visible = false;
68     var pausePosLion:Number = channellion.position;
69     channellion = sndLion.play(pausePosLion);
70     transLion.volume = vLion;
71     channellion.soundTransform = transLion;
72     timerLion.start();
73     timebarTimerLion.start();
74     timerBackOpenLion.start();
75     LionHolder.addEventListener(MouseEvent.CLICK, addShapeLionClick);
76     BTN_mute_lion.addEventListener(MouseEvent.CLICK, muteLion);
77     BTN_nomute_lion.addEventListener(MouseEvent.CLICK, nomuteLion);
78     BTN_lowVolume_lion.addEventListener(MouseEvent.CLICK, lowVolumeLion);
79     BTN_addVolume_lion.addEventListener(MouseEvent.CLICK, addVolumeLion);
80 }
81
82 //Volume
83 var transLion:SoundTransform = new SoundTransform();
84 var vLion:Number = 1.5;
85 var oSoundLion:Number = new Number;
86 var oVolumeBarLion:Number = new Number;
87 oSoundLion = vLion;
88 MC_volumeBar_lion.gotoAndStop(6);
89 oVolumeBarLion = MC_volumeBar_lion.currentFrame;
90 transLion.volume = vLion;
91 channellion.soundTransform = transLion;
92
93 //addVolume
94 BTN_addVolume_lion.addEventListener(MouseEvent.CLICK, addVolumeLion);
95
96 function addVolumeLion(event:MouseEvent):void {
97     MC_volumeBar_lion.nextFrame();
98     oVolumeBarLion = MC_volumeBar_lion.currentFrame;
99     vLion+=.3;
100    oSoundLion = vLion;
101    if (vLion > 5) {
102        vLion = 5;
103    }
104    transLion.volume = vLion;
105    channellion.soundTransform = transLion;
106 }
107
108 //lowVolume
109 BTN_lowVolume_lion.addEventListener(MouseEvent.CLICK, lowVolumeLion);
110
111 function lowVolumeLion(event:MouseEvent):void {
112     MC_volumeBar_lion.prevFrame();
113     oVolumeBarLion = MC_volumeBar_lion.currentFrame;
114     vLion-=.3;
115     oSoundLion = vLion;
116     if (vLion < 0) {
117         vLion = 0;
118     }
119     transLion.volume = vLion;
120     channellion.soundTransform = transLion;
121 }
122
```



```
123 //mute & no mute
124 BTN_mute_lion.addEventListener(MouseEvent.CLICK, muteLion);
125
126 function muteLion(event:MouseEvent):void {
127     BTN_lowVolume_lion.removeEventListener(MouseEvent.CLICK, lowVolumeLion);
128     BTN_addVolume_lion.removeEventListener(MouseEvent.CLICK, addVolumeLion);
129     MC_volumeBar_lion.gotoAndStop(1);
130     vLion = 0;
131     transLion.volume = vLion;
132     channellion.soundTransform = transLion;
133     BTN_mute_lion.visible = false;
134     BTN_nomute_lion.visible = true;
135 }
136
137 BTN_nomute_lion.addEventListener(MouseEvent.CLICK, nomuteLion);
138
139 function nomuteLion(event:MouseEvent):void {
140     BTN_lowVolume_lion.addEventListener(MouseEvent.CLICK, lowVolumeLion);
141     BTN_addVolume_lion.addEventListener(MouseEvent.CLICK, addVolumeLion);
142     MC_volumeBar_lion.gotoAndStop(oVolumeBarLion);
143     vLion = oSoundLion;
144     transLion.volume = vLion;
145     channellion.soundTransform = transLion;
146     BTN_nomute_lion.visible = false;
147     BTN_mute_lion.visible = true;
148 }
149
150 //Show Hide Mousev
151 LionHolder.addEventListener(MouseEvent.MOUSE_OVER, HideMouseLion);
152 LionHolder.addEventListener(MouseEvent.MOUSE_OUT, ShowMouseLion);
153 LionHolder.addEventListener(MouseEvent.MOUSE_DOWN, HideMouseLion);
154 LionHolder.addEventListener(MouseEvent.MOUSE_UP, HideMouseLion);
155 LionHolder.addEventListener(MouseEvent.CLICK, HideMouseLion);
156
157 var MC_crayon_lion:crayonLion = new crayonLion;
158 addChild(MC_crayon_lion);
159 MC_crayon_lion.gotoAndStop(3);
160
161 function followMouseLion(event:Event):void {
162     MC_crayon_lion.x = mouseX;
163     MC_crayon_lion.y = mouseY;
164 }
165
166 LionHolder.addEventListener(MouseEvent.ROLL_OVER, startMouseLion);
167 function startMouseLion(event:MouseEvent):void {
168     stage.addEventListener(Event.ENTER_FRAME, followMouseLion);
169 }
170
171 LionHolder.addEventListener(MouseEvent.ROLL_OUT, stopMouseLion);
172 function stopMouseLion(event:MouseEvent):void {
173     stage.removeEventListener(Event.ENTER_FRAME, followMouseLion);
174 }
175
176 function HideMouseLion(evt:MouseEvent) {
177     Mouse.hide();
178     MC_crayon_lion.mouseEnabled = false;
179 }
180 function ShowMouseLion(evt:MouseEvent) {
181     Mouse.show();
182 }
183
```

```

184 //Choose Color
185 var colorNumberLion:Number = 3;
186 MC_ColorChoice_Lion.gotoAndStop(3);
187
188 BTN_color_lion1.addEventListener(MouseEvent.CLICK, colorChangeLionA);
189 BTN_color_lion2.addEventListener(MouseEvent.CLICK, colorChangeLionB);
190 BTN_color_lion3.addEventListener(MouseEvent.CLICK, colorChangeLionC);
191 BTN_color_lion4.addEventListener(MouseEvent.CLICK, colorChangeLionD);
192 BTN_color_lion5.addEventListener(MouseEvent.CLICK, colorChangeLionE);
193
194 function colorChangeLionA(event:MouseEvent):void {
195     MC_crayon_lion.gotoAndStop(1);
196     MC_ColorChoice_Lion.gotoAndStop(1);
197     colorNumberLion = 1;
198 }
199
200 function colorChangeLionB(event:MouseEvent):void {
201     MC_crayon_lion.gotoAndStop(2);
202     MC_ColorChoice_Lion.gotoAndStop(2);
203     colorNumberLion = 2;
204 }
205
206 function colorChangeLionC(event:MouseEvent):void {
207     MC_crayon_lion.gotoAndStop(3);
208     MC_ColorChoice_Lion.gotoAndStop(3);
209     colorNumberLion = 3;
210 }
211
212 function colorChangeLionD(event:MouseEvent):void {
213     MC_crayon_lion.gotoAndStop(4);
214     MC_ColorChoice_Lion.gotoAndStop(4);
215     colorNumberLion = 4;
216 }
217
218 function colorChangeLionE(event:MouseEvent):void {
219     MC_crayon_lion.gotoAndStop(5);
220     MC_ColorChoice_Lion.gotoAndStop(5);
221     colorNumberLion = 5;
222 }
223
224 //Click on Stage
225 ///Number of lions for delete function
226 var N_Lion:Number = new Number;
227
228 ///Create an Array to trace the position of the mouse
229 var positionXLion:Array = new Array;
230 var NLion:Number = new Number;
231 NLion = -1;
232
233 LionHolder.addEventListener(MouseEvent.CLICK, addShapeLionClick);
234
235 function addShapeLionClick(evt:MouseEvent) {
236
237     MC_direction_lion.visible = false;
238
239     ///Random
240     var randomLion:Number;
241     if (colorNumberLion ==1) {
242         randomLion = Math.floor(Math.random()*5)+1;
243     }
244     if (colorNumberLion ==2) {
245         randomLion = Math.floor(Math.random()*5)+6;
246     }

```

```

247     if (colorNumberLion ==3) {
248         randomLion = Math.floor(Math.random()*5)+11;
249     }
250     if (colorNumberLion ==4) {
251         randomLion = Math.floor(Math.random()*5)+16;
252     }
253     if (colorNumberLion ==5) {
254         randomLion = Math.floor(Math.random()*5)+21;
255     }
256     //// create a new object from the Library
257     var loaderLion:MG_lion = new MG_lion;
258     loaderLion.gotoAndStop(randomLion);
259
260     //// Loader Scale & Position
261     loaderLion.x = mouseX-50;
262     loaderLion.y = mouseY-70;
263
264     if (mouseX > 600) {
265         loaderLion.scaleX = 0.7;
266     }
267     if (mouseX <= 600) {
268         loaderLion.scaleX = -0.7;
269     }
270     loaderLion.scaleY = 0.7;
271     loaderLion.cacheAsBitmap = true;
272     LionHolder.addChild(loaderLion);
273     N_Lion = LionHolder.numChildren;
274 }
275
276 //Delete
277 BTN_delete_lion.addEventListener(MouseEvent.CLICK, deleteShapeLionClick);
278
279 function deleteShapeLionClick(event:MouseEvent):void {
280     if (N_Lion > 1) {
281         LionHolder.removeChildAt(LionHolder.numChildren -1);
282     }
283     N_Lion = LionHolder.numChildren;
284 }
285
286 //Print
287 BTN_print_lion.addEventListener(MouseEvent.CLICK, PrintLion);
288
289 function PrintLion(event:MouseEvent):void {
290     var printTitleLion:MC_printTitle_lion = new MC_printTitle_lion;
291     var LionPrintJob:PrintJob = new PrintJob();
292     if (LionPrintJob.start()) {
293         try {
294             LionHolder.addChild(printTitleLion);
295             printTitleLion.x = 25;
296             printTitleLion.y = 20;
297             LionHolder.scaleX = .5;
298             LionHolder.scaleY = .5;
299             LionPrintJob.addPage(LionHolder,new Rectangle(7, 7, 1093, 493));
300         } catch (error:Error) {
301             // Handle error,
302         }
303         LionPrintJob.send();
304         LionHolder.removeChild(printTitleLion);
305         LionHolder.scaleX = 1;
306         LionHolder.scaleY = 1;
307     } else {
308         //trace("Print job canceled");
309     }
310 }
311

```

```
312 //Quit
313 BTN_quit_lion.addEventListener(MouseEvent.CLICK, LionGotoMenu);
314
315 function LionGotoMenu(event:MouseEvent):void {
316     gotoAndStop(1, "Menu");
317     channellion.stop();
318     removeChild(MC_crayon_lion);
319     timerLion.reset();
320     timerLion.removeEventListener(TimerEvent.TIMER, onTimerLion);
321     timerBackOpenLion.reset();
322     timerBackOpenLion.removeEventListener(TimerEvent.TIMER, onTimerBackOpenLion);
323     timebarTimerLion.reset();
324     timebarTimerLion.removeEventListener(TimerEvent.TIMER, barAddLion);
325 }
326
327 //Timer
328 var timerLion:Timer = new Timer(85000);
329 timerLion.addEventListener(TimerEvent.TIMER, onTimerLion);
330 timerLion.start();
331
332 function onTimerLion(evt:TimerEvent):void {
333     LionHolder.removeEventListener(MouseEvent.CLICK, addShapeLionClick);
334     BTN_stop_lion.removeEventListener(MouseEvent.CLICK, musicStopLion);
335     BTN_play_lion.removeEventListener(MouseEvent.CLICK, musicPlayLion);
336     BTN_delete_lion.removeEventListener(MouseEvent.CLICK, deleteShapeLionClick);
337     BTN_print_lion.visible = true;
338     timerLion.reset();
339     timerLion.removeEventListener(TimerEvent.TIMER, onTimerLion);
340     timebarTimerLion.reset();
341     timebarTimerLion.removeEventListener(TimerEvent.TIMER, barAddLion);
342     MC_timebar_lion.width = 1050;
343 }
344
345 //Timer for going back to Open
346 var timerBackOpenLion:Timer = new Timer(385000);
347 timerBackOpenLion.addEventListener(TimerEvent.TIMER, onTimerBackOpenLion);
348 timerBackOpenLion.start();
349
350 function onTimerBackOpenLion(evt:TimerEvent):void {
351     gotoAndStop(1, "Open");
352     timerBackOpenLion.reset();
353     timerBackOpenLion.removeEventListener(TimerEvent.TIMER, onTimerBackOpenLion);
354     channellion.stop();
355     removeChild(MC_crayon_lion);
356 }
```

Codes for the subject of Aquarium

```

1  stop();
2  BTN_print_fish.visible = false;
3  BTN_nomute_fish.visible = false;
4
5  //Add Print Background
6  var printBGFish:MC_printBG_fish = new MC_printBG_fish;
7  FishHolder.addChild(printBGFish);
8
9  //Time bar
10 var timebarTimerFish:Timer = new Timer(100);
11 timebarTimerFish.addEventListener(TimerEvent.TIMER, barAddFish);
12 timebarTimerFish.start();
13 MC_timebar_fish.width = 10;
14
15 var c:Number = new Number;
16 c = 1;
17
18 function barAddFish(evt:TimerEvent):void {
19     c +=20;
20     MC_timebar_fish.width = c*0.1;
21 }
22
23 //Music
24 var sndFish:Sound = new Sound();
25 sndFish.load(new URLRequest("SongFish.mp3"));
26
27 var channelFish:SoundChannel;
28 channelFish = sndFish.play();
29
30 //Sound Deta
31 var PeakFish:Number = new Number;
32 stage.addEventListener(Event.ENTER_FRAME, SoundDetaFish);
33 function SoundDetaFish(event:Event):void {
34     PeakFish = Math.floor((channelFish.leftPeak+channelFish.rightPeak)*50);
35     MC_peak_fish.scaleX = PeakFish*0.015;
36     if (MC_peak_fish.width > 150) {
37         MC_peak_fish.width = 150;
38     }
39     if (PeakFish == 0) {
40         PeakFish = Math.floor(Math.random()*10)+1;
41     }
42 }
43
44 //Music Stop
45 BTN_stop_fish.addEventListener(MouseEvent.CLICK, musicStopFish);
46
47 function musicStopFish(event:MouseEvent):void {
48     BTN_print_fish.visible = true;
49     var pausePosFish:Number = channelFish.position;
50     channelFish.stop();
51     transFish.volume = vFish;
52     channelFish.soundTransform = transFish;
53     timerFish.stop();
54     timebarTimerFish.stop();
55     timerBackOpenFish.stop();
56     FishHolder.removeEventListener(MouseEvent.CLICK, addShapeFishClick);
57     BTN_mute_fish.removeEventListener(MouseEvent.CLICK, muteFish);
58     BTN_nomute_fish.removeEventListener(MouseEvent.CLICK, nomuteFish);
59     BTN_lowVolume_fish.removeEventListener(MouseEvent.CLICK, lowVolumeFish);
60     BTN_addVolume_fish.removeEventListener(MouseEvent.CLICK, addVolumeFish);
61 }
62

```

```
63 //Music Play
64 BTN_play_fish.addEventListener(MouseEvent.CLICK, musicPlayFish);
65
66 function musicPlayFish(event:MouseEvent):void {
67     BTN_print_fish.visible = false;
68     var pausePosFish:Number = channelFish.position;
69     channelFish = sndFish.play(pausePosFish);
70     transFish.volume = vFish;
71     channelFish.soundTransform = transFish;
72     timerFish.start();
73     timebarTimerFish.start();
74     timerBackOpenFish.start();
75     FishHolder.addEventListener(MouseEvent.CLICK, addShapeFishClick);
76     BTN_mute_fish.addEventListener(MouseEvent.CLICK, muteFish);
77     BTN_nomute_fish.addEventListener(MouseEvent.CLICK, nomuteFish);
78     BTN_lowVolume_fish.addEventListener(MouseEvent.CLICK, lowVolumeFish);
79     BTN_addVolume_fish.addEventListener(MouseEvent.CLICK, addVolumeFish);
80 }
81
82 //Volume
83 var transFish:SoundTransform = new SoundTransform();
84 var vFish:Number = 1.5;
85 var oSoundFish:Number = new Number;
86 var oVolumeBarFish:Number = new Number;
87 oSoundFish = vFish;
88 MC_volumeBar_fish.gotoAndStop(6);
89 oVolumeBarFish = MC_volumeBar_fish.currentFrame;
90 transFish.volume = vFish;
91 channelFish.soundTransform = transFish;
92
93 //addVolume
94 BTN_addVolume_fish.addEventListener(MouseEvent.CLICK, addVolumeFish);
95
96 function addVolumeFish(event:MouseEvent):void {
97     MC_volumeBar_fish.nextFrame();
98     oVolumeBarFish = MC_volumeBar_fish.currentFrame;
99     vFish+=.3;
100    oSoundFish = vFish;
101    if (vFish > 5) {
102        vFish = 5;
103    }
104    transFish.volume = vFish;
105    channelFish.soundTransform = transFish;
106 }
107
108 //lowVolume
109 BTN_lowVolume_fish.addEventListener(MouseEvent.CLICK, lowVolumeFish);
110
111 function lowVolumeFish(event:MouseEvent):void {
112     MC_volumeBar_fish.prevFrame();
113     oVolumeBarFish = MC_volumeBar_fish.currentFrame;
114     vFish-=.3;
115     oSoundFish = vFish;
116     if (vFish < 0) {
117         vFish = 0;
118     }
119     transFish.volume = vFish;
120     channelFish.soundTransform = transFish;
121 }
122
```

```

123 //mute & no mute
124 BTN_mute_fish.addEventListener(MouseEvent.CLICK, muteFish);
125
126 function muteFish(event:MouseEvent):void {
127     BTN_lowVolume_fish.removeEventListener(MouseEvent.CLICK, lowVolumeFish);
128     BTN_addVolume_fish.removeEventListener(MouseEvent.CLICK, addVolumeFish);
129     MC_volumeBar_fish.gotoAndStop(1);
130     vFish = 0;
131     transFish.volume = vFish;
132     channelFish.soundTransform = transFish;
133     BTN_mute_fish.visible = false;
134     BTN_nomute_fish.visible = true;
135 }
136
137 BTN_nomute_fish.addEventListener(MouseEvent.CLICK, nomuteFish);
138
139 function nomuteFish(event:MouseEvent):void {
140     BTN_lowVolume_fish.addEventListener(MouseEvent.CLICK, lowVolumeFish);
141     BTN_addVolume_fish.addEventListener(MouseEvent.CLICK, addVolumeFish);
142     MC_volumeBar_fish.gotoAndStop(oVolumeBarFish);
143     vFish = oSoundFish;
144     transFish.volume = vFish;
145     channelFish.soundTransform = transFish;
146     BTN_nomute_fish.visible = false;
147     BTN_mute_fish.visible = true;
148 }
149
150 //Show Hide Mousev
151 FishHolder.addEventListener(MouseEvent.MOUSE_OVER, HideMouseFish);
152 FishHolder.addEventListener(MouseEvent.MOUSE_OUT, ShowMouseFish);
153 FishHolder.addEventListener(MouseEvent.MOUSE_DOWN, HideMouseFish);
154 FishHolder.addEventListener(MouseEvent.MOUSE_UP, HideMouseFish);
155 FishHolder.addEventListener(MouseEvent.CLICK, HideMouseFish);
156
157 var MC_crayon_fish:crayonFish = new crayonFish;
158 addChild(MC_crayon_fish);
159 MC_crayon_fish.gotoAndStop(3);
160
161 function followMouseFish(event:Event):void {
162     MC_crayon_fish.x = mouseX;
163     MC_crayon_fish.y = mouseY;
164 }
165
166 FishHolder.addEventListener(MouseEvent.ROLL_OVER, startMouseFish);
167 function startMouseFish(event:MouseEvent):void {
168     stage.addEventListener(Event.ENTER_FRAME, followMouseFish);
169 }
170
171 FishHolder.addEventListener(MouseEvent.ROLL_OUT, stopMouseFish);
172 function stopMouseFish(event:MouseEvent):void {
173     stage.removeEventListener(Event.ENTER_FRAME, followMouseFish);
174 }
175
176 function HideMouseFish(evt:MouseEvent) {
177     Mouse.hide();
178     MC_crayon_fish.mouseEnabled = false;
179 }
180 function ShowMouseFish(evt:MouseEvent) {
181     Mouse.show();
182 }
183

```

```

184 //Choose Color
185 var colorNumberFish:Number = 3;
186 MC_ColorChoice_Fish.gotoAndStop(3);
187
188 BTN_color_fish1.addEventListener(MouseEvent.CLICK, colorChangeFishA);
189 BTN_color_fish2.addEventListener(MouseEvent.CLICK, colorChangeFishB);
190 BTN_color_fish3.addEventListener(MouseEvent.CLICK, colorChangeFishC);
191 BTN_color_fish4.addEventListener(MouseEvent.CLICK, colorChangeFishD);
192 BTN_color_fish5.addEventListener(MouseEvent.CLICK, colorChangeFishE);
193
194 function colorChangeFishA(event:MouseEvent):void {
195     MC_crayon_fish.gotoAndStop(1);
196     MC_ColorChoice_Fish.gotoAndStop(1);
197     colorNumberFish = 1;
198 }
199
200 function colorChangeFishB(event:MouseEvent):void {
201     MC_crayon_fish.gotoAndStop(2);
202     MC_ColorChoice_Fish.gotoAndStop(2);
203     colorNumberFish = 2;
204 }
205
206 function colorChangeFishC(event:MouseEvent):void {
207     MC_crayon_fish.gotoAndStop(3);
208     MC_ColorChoice_Fish.gotoAndStop(3);
209     colorNumberFish = 3;
210 }
211
212 function colorChangeFishD(event:MouseEvent):void {
213     MC_crayon_fish.gotoAndStop(4);
214     MC_ColorChoice_Fish.gotoAndStop(4);
215     colorNumberFish = 4;
216 }
217
218 function colorChangeFishE(event:MouseEvent):void {
219     MC_crayon_fish.gotoAndStop(5);
220     MC_ColorChoice_Fish.gotoAndStop(5);
221     colorNumberFish = 5;
222 }
223
224 //Click on Stage
225 ///Number of fishs for delete function
226 var N_Fish:Number = new Number;
227
228 ///Create an Array to trace the position of the mouse
229 var positionXFish:Array = new Array;
230 var NFish:Number = new Number;
231 NFish = -1;
232
233 FishHolder.addEventListener(MouseEvent.CLICK, addShapeFishClick);
234
235 function addShapeFishClick(evt:MouseEvent) {
236
237     MC_direction_fish.visible = false;
238
239     ///Random
240     var randomFish:Number;
241     if (colorNumberFish ==1) {
242         randomFish = Math.floor(Math.random()*5)+1;
243     }
244     if (colorNumberFish ==2) {
245         randomFish = Math.floor(Math.random()*5)+6;
246     }

```



```

247     if (colorNumberFish ==3) {
248         randomFish = Math.floor(Math.random()*5)+11;
249     }
250     if (colorNumberFish ==4) {
251         randomFish = Math.floor(Math.random()*5)+16;
252     }
253     if (colorNumberFish ==5) {
254         randomFish = Math.floor(Math.random()*5)+21;
255     }
256     //// create a new object from the Library
257     var loaderFish:MG_fish = new MG_fish;
258     loaderFish.gotoAndStop(randomFish);
259
260     //// Loader Scale & Position
261     loaderFish.x = mouseX-50;
262     loaderFish.y = mouseY-70;
263
264     if (mouseX > 600) {
265         loaderFish.scaleX = 0.7;
266     }
267     if (mouseX <= 600) {
268         loaderFish.scaleX = -0.7;
269     }
270     loaderFish.scaleY = 0.7;
271     loaderFish.cacheAsBitmap = true;
272     FishHolder.addChild(loaderFish);
273     N_Fish = FishHolder.numChildren;
274 }
275
276 //Delete
277 BTN_delete_fish.addEventListener(MouseEvent.CLICK, deleteShapeFishClick);
278
279 function deleteShapeFishClick(event:MouseEvent):void {
280     if (N_Fish > 1) {
281         FishHolder.removeChildAt(FishHolder.numChildren -1);
282     }
283     N_Fish = FishHolder.numChildren;
284 }
285
286 //Print
287 BTN_print_fish.addEventListener(MouseEvent.CLICK, PrintFish);
288
289 function PrintFish(event:MouseEvent):void {
290     var printTitleFish:MC_printTitle_fish = new MC_printTitle_fish;
291     var FishPrintJob:PrintJob = new PrintJob();
292     if (FishPrintJob.start()) {
293         try {
294             FishHolder.addChild(printTitleFish);
295             printTitleFish.x = 25;
296             printTitleFish.y = 20;
297             FishHolder.scaleX = .5;
298             FishHolder.scaleY = .5;
299             FishPrintJob.addPage(FishHolder,new Rectangle(7, 7, 1093, 493));
300         } catch (error:Error) {
301             // Handle error,
302         }
303         FishPrintJob.send();
304         FishHolder.removeChild(printTitleFish);
305         FishHolder.scaleX = 1;
306         FishHolder.scaleY = 1;
307     } else {
308         //trace("Print job canceled");
309     }
310 }
311

```

```
312 //Quit
313 BTN_quit_fish.addEventListener(MouseEvent.CLICK, FishGotoMenu);
314
315 function FishGotoMenu(event:MouseEvent):void {
316     gotoAndStop(1, "Menu");
317     channelFish.stop();
318     removeChild(MC_crayon_fish);
319     timerFish.reset();
320     timerFish.removeEventListener(TimerEvent.TIMER, onTimerFish);
321     timerBackOpenFish.reset();
322     timerBackOpenFish.removeEventListener(TimerEvent.TIMER, onTimerBackOpenFish);
323     timebarTimerFish.reset();
324     timebarTimerFish.removeEventListener(TimerEvent.TIMER, barAddFish);
325 }
326
327 //Timer
328 var timerFish:Timer = new Timer(85000);
329 timerFish.addEventListener(TimerEvent.TIMER, onTimerFish);
330 timerFish.start();
331
332 function onTimerFish(evt:TimerEvent):void {
333     FishHolder.removeEventListener(MouseEvent.CLICK, addShapeFishClick);
334     BTN_stop_fish.removeEventListener(MouseEvent.CLICK, musicStopFish);
335     BTN_play_fish.removeEventListener(MouseEvent.CLICK, musicPlayFish);
336     BTN_delete_fish.removeEventListener(MouseEvent.CLICK, deleteShapeFishClick);
337     BTN_print_fish.visible = true;
338     timerFish.reset();
339     timerFish.removeEventListener(TimerEvent.TIMER, onTimerFish);
340     timebarTimerFish.reset();
341     timebarTimerFish.removeEventListener(TimerEvent.TIMER, barAddFish);
342     MC_timebar_fish.width = 1050;
343 }
344
345 //Timer for going back to Open
346 var timerBackOpenFish:Timer = new Timer(385000);
347 timerBackOpenFish.addEventListener(TimerEvent.TIMER, onTimerBackOpenFish);
348 timerBackOpenFish.start();
349
350 function onTimerBackOpenFish(evt:TimerEvent):void {
351     gotoAndStop(1, "Open");
352     timerBackOpenFish.reset();
353     timerBackOpenFish.removeEventListener(TimerEvent.TIMER, onTimerBackOpenFish);
354     channelFish.stop();
355     removeChild(MC_crayon_fish);
356 }
```

Codes for the subject of the Swan

```

1  stop();
2  BTN_print_swan.visible = false;
3  BTN_nomute_swan.visible = false;
4
5  //Add Print Background
6  var printBGSwan:MC_printBG_swan = new MC_printBG_swan;
7  SwanHolder.addChild(printBGSwan);
8
9  //Time bar
10 var timebarTimerSwan:Timer = new Timer(100);
11 timebarTimerSwan.addEventListener(TimerEvent.TIMER, barAddSwan);
12 timebarTimerSwan.start();
13 MC_timebar_swan.width = 10;
14
15 var b:Number = new Number;
16 b = 1;
17
18 function barAddSwan(evt:TimerEvent):void {
19     b +=20;
20     MC_timebar_swan.width = b*0.1;
21 }
22
23 //Music
24 var sndSwan:Sound = new Sound();
25 sndSwan.load(new URLRequest("SongSwan.mp3"));
26
27 var channelSwan:SoundChannel;
28 channelSwan = sndSwan.play();
29
30 //Sound Deta
31 var PeakSwan:Number = new Number;
32 stage.addEventListener(Event.ENTER_FRAME, SoundDetaSwan);
33 function SoundDetaSwan(event:Event):void {
34     PeakSwan = Math.floor((channelSwan.leftPeak+channelSwan.rightPeak)*50);
35     MC_peak_swan.scaleX = PeakSwan*0.015;
36     if (MC_peak_swan.width > 150) {
37         MC_peak_swan.width = 150;
38     }
39     if (PeakSwan == 0) {
40         PeakSwan = Math.floor(Math.random()*10)+1;
41     }
42 }
43
44 //Music Stop
45 BTN_stop_swan.addEventListener(MouseEvent.CLICK, musicStopSwan);
46
47 function musicStopSwan(event:MouseEvent):void {
48     BTN_print_swan.visible = true;
49     var pausePosSwan:Number = channelSwan.position;
50     channelSwan.stop();
51     transSwan.volume = vSwan;
52     channelSwan.soundTransform = transSwan;
53     timerSwan.stop();
54     timebarTimerSwan.stop();
55     timerBackOpenSwan.stop();
56     SwanHolder.removeEventListener(MouseEvent.CLICK, addShapeSwanClick);
57     BTN_mute_swan.removeEventListener(MouseEvent.CLICK, muteSwan);
58     BTN_nomute_swan.removeEventListener(MouseEvent.CLICK, nomuteSwan);
59     BTN_lowVolume_swan.removeEventListener(MouseEvent.CLICK, lowVolumeSwan);
60     BTN_addVolume_swan.removeEventListener(MouseEvent.CLICK, addVolumeSwan);
61 }
62

```

```
63 //Music Play
64 BTN_play_swan.addEventListener(MouseEvent.CLICK, musicPlaySwan);
65
66 function musicPlaySwan(event:MouseEvent):void {
67     BTN_print_swan.visible = false;
68     var pausePosSwan:Number = channelSwan.position;
69     channelSwan = sndSwan.play(pausePosSwan);
70     transSwan.volume = vSwan;
71     channelSwan.soundTransform = transSwan;
72     timerSwan.start();
73     timebarTimerSwan.start();
74     timerBackOpenSwan.start();
75     SwanHolder.addEventListener(MouseEvent.CLICK, addShapeSwanClick);
76     BTN_mute_swan.addEventListener(MouseEvent.CLICK, muteSwan);
77     BTN_nomute_swan.addEventListener(MouseEvent.CLICK, nomuteSwan);
78     BTN_lowVolume_swan.addEventListener(MouseEvent.CLICK, lowVolumeSwan);
79     BTN_addVolume_swan.addEventListener(MouseEvent.CLICK, addVolumeSwan);
80 }
81
82 //Volume
83 var transSwan:SoundTransform = new SoundTransform();
84 var vSwan:Number = 1.5;
85 var oSoundSwan:Number = new Number;
86 var oVolumeBarSwan:Number = new Number;
87 oSoundSwan = vSwan;
88 MC_volumeBar_swan.gotoAndStop(6);
89 oVolumeBarSwan = MC_volumeBar_swan.currentFrame;
90 transSwan.volume = vSwan;
91 channelSwan.soundTransform = transSwan;
92
93 //addVolume
94 BTN_addVolume_swan.addEventListener(MouseEvent.CLICK, addVolumeSwan);
95
96 function addVolumeSwan(event:MouseEvent):void {
97     MC_volumeBar_swan.nextFrame();
98     oVolumeBarSwan = MC_volumeBar_swan.currentFrame;
99     vSwan+=.3;
100    oSoundSwan = vSwan;
101    if (vSwan > 5) {
102        vSwan = 5;
103    }
104    transSwan.volume = vSwan;
105    channelSwan.soundTransform = transSwan;
106 }
107
108 //lowVolume
109 BTN_lowVolume_swan.addEventListener(MouseEvent.CLICK, lowVolumeSwan);
110
111 function lowVolumeSwan(event:MouseEvent):void {
112     MC_volumeBar_swan.prevFrame();
113     oVolumeBarSwan = MC_volumeBar_swan.currentFrame;
114     vSwan-=.3;
115     oSoundSwan = vSwan;
116     if (vSwan < 0) {
117         vSwan = 0;
118     }
119     transSwan.volume = vSwan;
120     channelSwan.soundTransform = transSwan;
121 }
122
```

```

123 //mute & no mute
124 BTN_mute_swan.addEventListener(MouseEvent.CLICK, muteSwan);
125
126 function muteSwan(event:MouseEvent):void {
127     BTN_lowVolume_swan.removeEventListener(MouseEvent.CLICK, lowVolumeSwan);
128     BTN_addVolume_swan.removeEventListener(MouseEvent.CLICK, addVolumeSwan);
129     MC_volumeBar_swan.gotoAndStop(1);
130     vSwan = 0;
131     transSwan.volume = vSwan;
132     channelSwan.soundTransform = transSwan;
133     BTN_mute_swan.visible = false;
134     BTN_nomute_swan.visible = true;
135 }
136
137 BTN_nomute_swan.addEventListener(MouseEvent.CLICK, nomuteSwan);
138
139 function nomuteSwan(event:MouseEvent):void {
140     BTN_lowVolume_swan.addEventListener(MouseEvent.CLICK, lowVolumeSwan);
141     BTN_addVolume_swan.addEventListener(MouseEvent.CLICK, addVolumeSwan);
142     MC_volumeBar_swan.gotoAndStop(oVolumeBarSwan);
143     vSwan = oSoundSwan;
144     transSwan.volume = vSwan;
145     channelSwan.soundTransform = transSwan;
146     BTN_nomute_swan.visible = false;
147     BTN_mute_swan.visible = true;
148 }
149
150 //Show Hide Mousev
151 SwanHolder.addEventListener(MouseEvent.MOUSE_OVER, HideMouseSwan);
152 SwanHolder.addEventListener(MouseEvent.MOUSE_OUT, ShowMouseSwan);
153 SwanHolder.addEventListener(MouseEvent.MOUSE_DOWN, HideMouseSwan);
154 SwanHolder.addEventListener(MouseEvent.MOUSE_UP, HideMouseSwan);
155 SwanHolder.addEventListener(MouseEvent.CLICK, HideMouseSwan);
156
157 var MC_crayon_swan:crayonSwan = new crayonSwan;
158 addChild(MC_crayon_swan);
159 MC_crayon_swan.gotoAndStop(3);
160
161 function followMouseSwan(event:Event):void {
162     MC_crayon_swan.x = mouseX;
163     MC_crayon_swan.y = mouseY;
164 }
165
166 SwanHolder.addEventListener(MouseEvent.ROLL_OVER, startMouseSwan);
167 function startMouseSwan(event:MouseEvent):void {
168     stage.addEventListener(Event.ENTER_FRAME, followMouseSwan);
169 }
170
171 SwanHolder.addEventListener(MouseEvent.ROLL_OUT, stopMouseSwan);
172 function stopMouseSwan(event:MouseEvent):void {
173     stage.removeEventListener(Event.ENTER_FRAME, followMouseSwan);
174 }
175
176 function HideMouseSwan(evt:MouseEvent) {
177     Mouse.hide();
178     MC_crayon_swan.mouseEnabled = false;
179 }
180 function ShowMouseSwan(evt:MouseEvent) {
181     Mouse.show();
182 }
183

```

```
184 //Choose Color
185 var colorNumberSwan:Number = 3;
186 MC_ColorChoice_Swan.gotoAndStop(3);
187
188 BTN_color_swan1.addEventListener(MouseEvent.CLICK, colorChangeSwanA);
189 BTN_color_swan2.addEventListener(MouseEvent.CLICK, colorChangeSwanB);
190 BTN_color_swan3.addEventListener(MouseEvent.CLICK, colorChangeSwanC);
191 BTN_color_swan4.addEventListener(MouseEvent.CLICK, colorChangeSwanD);
192 BTN_color_swan5.addEventListener(MouseEvent.CLICK, colorChangeSwanE);
193
194 function colorChangeSwanA(event:MouseEvent):void {
195     MC_crayon_swan.gotoAndStop(1);
196     MC_ColorChoice_Swan.gotoAndStop(1);
197     colorNumberSwan = 1;
198 }
199
200 function colorChangeSwanB(event:MouseEvent):void {
201     MC_crayon_swan.gotoAndStop(2);
202     MC_ColorChoice_Swan.gotoAndStop(2);
203     colorNumberSwan = 2;
204 }
205
206 function colorChangeSwanC(event:MouseEvent):void {
207     MC_crayon_swan.gotoAndStop(3);
208     MC_ColorChoice_Swan.gotoAndStop(3);
209     colorNumberSwan = 3;
210 }
211
212 function colorChangeSwanD(event:MouseEvent):void {
213     MC_crayon_swan.gotoAndStop(4);
214     MC_ColorChoice_Swan.gotoAndStop(4);
215     colorNumberSwan = 4;
216 }
217
218 function colorChangeSwanE(event:MouseEvent):void {
219     MC_crayon_swan.gotoAndStop(5);
220     MC_ColorChoice_Swan.gotoAndStop(5);
221     colorNumberSwan = 5;
222 }
223
224 //Click on Stage
225 ///Number of swans for delete function
226 var N_Swan:Number = new Number;
227
228 ///Create an Array to trace the position of the mouse
229 var positionXSwan:Array = new Array;
230 var NSwan:Number = new Number;
231 NSwan = -1;
232
233 SwanHolder.addEventListener(MouseEvent.CLICK, addShapeSwanClick);
234
235 function addShapeSwanClick(evt:MouseEvent) {
236
237     MC_direction_swan.visible = false;
238
239     ///Random
240     var randomSwan:Number;
241     if (colorNumberSwan ==1) {
242         randomSwan = Math.floor(Math.random()*5)+1;
243     }
244     if (colorNumberSwan ==2) {
245         randomSwan = Math.floor(Math.random()*5)+6;
246     }
```

```

247     if (colorNumberSwan ==3) {
248         randomSwan = Math.floor(Math.random()*5)+11;
249     }
250     if (colorNumberSwan ==4) {
251         randomSwan = Math.floor(Math.random()*5)+16;
252     }
253     if (colorNumberSwan ==5) {
254         randomSwan = Math.floor(Math.random()*5)+21;
255     }
256     //// create a new object from the Library
257     var loaderSwan:MG_swan = new MG_swan;
258     loaderSwan.gotoAndStop(randomSwan);
259
260     //// Loader Scale & Position
261     loaderSwan.x = mouseX-50;
262     loaderSwan.y = mouseY-70;
263
264     if (mouseX > 600) {
265         loaderSwan.scaleX = 0.7;
266     }
267     if (mouseX <= 600) {
268         loaderSwan.scaleX = -0.7;
269     }
270     loaderSwan.scaleY = 0.7;
271     loaderSwan.cacheAsBitmap = true;
272     SwanHolder.addChild(loaderSwan);
273     N_Swan = SwanHolder.numChildren;
274 }
275
276 //Delete
277 BTN_delete_swan.addEventListener(MouseEvent.CLICK, deleteShapeSwanClick);
278
279 function deleteShapeSwanClick(event:MouseEvent):void {
280     if (N_Swan > 1) {
281         SwanHolder.removeChildAt(SwanHolder.numChildren -1);
282     }
283     N_Swan = SwanHolder.numChildren;
284 }
285
286 //Print
287 BTN_print_swan.addEventListener(MouseEvent.CLICK, PrintSwan);
288
289 function PrintSwan(event:MouseEvent):void {
290     var printTitleSwan:MC_printTitle_swan = new MC_printTitle_swan;
291     var SwanPrintJob:PrintJob = new PrintJob();
292     if (SwanPrintJob.start()) {
293         try {
294             SwanHolder.addChild(printTitleSwan);
295             printTitleSwan.x = 25;
296             printTitleSwan.y = 20;
297             SwanHolder.scaleX = .5;
298             SwanHolder.scaleY = .5;
299             SwanPrintJob.addPage(SwanHolder,new Rectangle(7, 7, 1093, 493));
300         } catch (error:Error) {
301             // Handle error,
302         }
303         SwanPrintJob.send();
304         SwanHolder.removeChild(printTitleSwan);
305         SwanHolder.scaleX = 1;
306         SwanHolder.scaleY = 1;
307     } else {
308         //trace("Print job canceled");
309     }
310 }
311

```

```
312 //Quit
313 BTN_quit_swan.addEventListener(MouseEvent.CLICK, SwanGotoMenu);
314
315 function SwanGotoMenu(event:MouseEvent):void {
316     gotoAndStop(1, "Menu");
317     channelSwan.stop();
318     removeChild(MC_crayon_swan);
319     timerSwan.reset();
320     timerSwan.removeEventListener(TimerEvent.TIMER, onTimerSwan);
321     timerBackOpenSwan.reset();
322     timerBackOpenSwan.removeEventListener(TimerEvent.TIMER, onTimerBackOpenSwan);
323     timebarTimerSwan.reset();
324     timebarTimerSwan.removeEventListener(TimerEvent.TIMER, barAddSwan);
325 }
326
327 //Timer
328 var timerSwan:Timer = new Timer(85000);
329 timerSwan.addEventListener(TimerEvent.TIMER, onTimerSwan);
330 timerSwan.start();
331
332 function onTimerSwan(evt:TimerEvent):void {
333     SwanHolder.removeEventListener(MouseEvent.CLICK, addShapeSwanClick);
334     BTN_stop_swan.removeEventListener(MouseEvent.CLICK, musicStopSwan);
335     BTN_play_swan.removeEventListener(MouseEvent.CLICK, musicPlaySwan);
336     BTN_delete_swan.removeEventListener(MouseEvent.CLICK, deleteShapeSwanClick);
337     BTN_print_swan.visible = true;
338     timerSwan.reset();
339     timerSwan.removeEventListener(TimerEvent.TIMER, onTimerSwan);
340     timebarTimerSwan.reset();
341     timebarTimerSwan.removeEventListener(TimerEvent.TIMER, barAddSwan);
342     MC_timebar_swan.width = 1050;
343 }
344
345 //Timer for going back to Open
346 var timerBackOpenSwan:Timer = new Timer(385000);
347 timerBackOpenSwan.addEventListener(TimerEvent.TIMER, onTimerBackOpenSwan);
348 timerBackOpenSwan.start();
349
350 function onTimerBackOpenSwan(evt:TimerEvent):void {
351     gotoAndStop(1, "Open");
352     timerBackOpenSwan.reset();
353     timerBackOpenSwan.removeEventListener(TimerEvent.TIMER, onTimerBackOpenSwan);
354     channelSwan.stop();
355     removeChild(MC_crayon_swan);
356 }
```

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

Rich Shupe, Zevan Rosser. Learning ActionScript 3.0: A Beginner's Guide. O'Reilly, 2007

Stuart Mealing. Computers and Art. Intellect Books, 1997

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

Yang, Dong-Yu. Flash ActionScript 3.0. Flag Publishing Co., Ltd., 2008