West East

Interactive WiiArt

A thesis submitted to the Faculty of
The College of Imaging Arts and Sciences
in candidacy for the degree of
Master of Fine Arts

Rochester Institute of Technology
College of Imaging Arts and Sciences
School of Design
Graduate Computer Graphics Design MFA Program

Thesis Committee

Chris Jackson, Dan Deluna, Jason Arena

NaiChieh Yu
April 2010

Thesis for the Master of Fine Arts Degree

Rochester Institute of Technology
College of Imaging Arts and Sciences
School of Design
Computer Graphics Design

Title: Interactive WiiArt

Submitted by: NaiChieh Yu

Date: April 21, 2010

Thesis Committee Approval:

Chief Adviser: Associate Professor Chris Jackson, Computer Graphics Design

_____ _____

Signature of Chief Adviser Date

Associate Adviser: Assistant Professor Dan Deluna, Computer Graphics Design

_____ _____

Signature of Associate Adviser Date

Associate Adviser: Associate Professor Jason Arena, New Media

_____ _____

Signature of Associate Adviser Date

School of Design Chairperson Approval:

Chairperson, School of Design: Patti Lachance

_____ _____

Signature of Chairperson Date

Abstract

This interactive project allows users to explore modern graphics and traditional Chinese ink painting by using a Nintendo WiiRemote control. The purpose of this project is to provide an effortless platform for users to create an artwork with simple control. Moreover, the default patterns let users have multiple choices to paint the scene, and even though users do not have any art, design background, or experience, they can still create their own masterpiece easily.

The art styles chosen, Chinese ink painting and Vector Art, represent Eastern and Western cultures, respectively. There are several patterns in each painting board for users to choose. The patterns on the painting board for Chinese ink painting are basic elements for each Chinese ink painting; users can simply use different combinations of elements to create their own unique piece. In order to present the best result of painting, 2D and 3D are used, and it depends on which method is more suitable for each element. As for the patterns on the painting board for Vector Art, motions and many colors are used to generate a clear contrast between the two art styles. The patterns are drawn in Illustrator, and the motions are created in Flash.

The Nintendo WiiRemote control is chosen as the input device because of its popularity. Moreover, it is a novel input device that is used in an interactive drawing interface, and it creates a drawing experience that is similar to real drawing. When users hold the WiiRemote, they feel like they are holding the painting brush.

The project focuses on creating immersive and highly interactive environments as well as exciting, playful, and enjoyable experiences.

Project Definition

1.1 Introduction

An interactive drawing environment that offers users a new and fun drawing experience is created. Through this new interactive drawing interface, sequences of graphics are offered, and by simply playing with the WiiRemote, users can make their own piece in an easy and joyful way.

There are two different styles of painting boards for users to play with: one provides colorful modern graphic patterns, while the other provides traditional Chinese ink painting patterns. In each painting board, there are several sequences of graphics for users to choose. Instead of simple graphics used in interactive drawing tools today, those more complicated graphics in this project produce richer visuals and more magnificent results, and, therefore, provide users with more fun and a sense of accomplishment.

1.2 Inspiration

■ **Adobe CS3 Mural Installation Spring 2007***

This thesis is initially inspired by ”Adobe CS3 Mural Installation,” which is a project that uses people’s movements to interact with graphics.

The following description is quoted from the website of the project:

“In conjunction with the launch of Adobe’s CS3, this interactive mural installation was designed by Brand New School for Goodby, Silverstein & Partners of San Francisco. Using some fairly sophisticated programming and tracking hardware, peoples’ movements are recorded and translated into an animated mixed-media mural that reflects the creative license afforded by the new Adobe CS3 software package. From left to right, the mural evolves from simplicity to complexity as more elements are introduced.

The installation, which stands at 10’ X 25’, is set to tour around the world and appear at film festivals, media conferences, and wherever those seeking complete creative license are known to congregate.[1]

*Retrieved from Adobe CS3 Mural Installation:http://www.brandnewschool.com/project.php?id=410

Create a richer interactive drawing environment

Besides the project mentioned above, there are two additional thoughts regarding the inspiration of the thesis. First, users experience more depth of art in the interactive drawing environment. Traditionally, interactive drawing presents uncomplicated geometric shapes to users. Since most of the users do not have an art or design background, they would not be able to create a piece by using those simple interactive tools. With this thought in mind, the paint palette in this project provides various default patterns which are a sequence of graphics or Chinese ink painting instead of just a single color or a geometric shape. This interactive interface provides users with a visually richer experience.

Incorporate a novel input device

Meanwhile, drawing tools or platforms today are quite similar. In order to create an interactive drawing experience that is most similar to a real painting experience and being playful at the same time, the next task is to find something other than traditional devices such as a mouse or keypad.

In summary, the idea of this project is fairly straightforward: I want to set up an installation which is open and simple for users to play with. Moreover, this project would bring another dimension for users to explore the artwork and changes the way people use painting tools by adding interactive graphics and new input devices, which provide a whole new painting experience.

Keywords

WiiArt

Interactive drawing

Chinese ink painting

Vector art

WiiRemote

WiiFlash

Art experiences

Research

3.1 Art style

Before finalizing the art styles that would be used in this project, research is conducted to determine which art styles could create fantastic visuals in the easiest way. Among all the art styles researched, Chinese ink painting is first chosen for the thesis, since there are few projects about Chinese ink painting, and my background as a Chinese individual. The idea is to incorporate something from oriental culture into the thesis and create a contrast between Eastern and Western cultures visually. Therefore, I decided to put two art styles in the thesis, one represents Eastern culture, while the other represents Western culture. For the Western culture, Vector Art is chosen. This art style is fun and colorful and, therefore, provides a clear contrast to Chinese ink painting, which is purely black and white.

As a result, the chosen art styles present both Eastern and Western cultures with totally different art look.

3.2 Input Device

A novel input device, which is not a conventional device such as a mouse or keypad and one which is most similar to a real painting experience, is needed. Here is a list of input devices that were considered:

■ **Webcam**

Macromedia Flash Player 8, has an impressive and generous new feature - a new BitmapData ActionScript API, which gives developers the power to create and modify bitmaps at run time with ActionScript. The idea of using the webcam is to monitor the active webcam for movement and allow users to generate and interact with the graphics by moving around when in view.

■ **Microphone**

The ability to interact with a microphone is something that was introduced in recent versions of Flash. The idea of using a microphone is to create sequences of graphics which react to the activity level of the microphone, so if the user starts to blow into the microphone, the graphics will start to generate and randomly position on screen.

■ **WiiRemote + Wii Sensor Bar**

As of March 2009, Nintendo has sold nearly 50 million Wii game consoles. This makes WiiRemote one of the most common computer input devices in the world. A main feature of the WiiRemote is its motion sensing capability, which allows users to interact with and manipulate items on screen via gesture recognition and pointing through the use of accelerometer and optical sensor technology. By working with the Wii Sensor Bar, which is technically an array of infrared LED, the relative movement of the infrared LED will be used for cursor position updates. The idea of using WiiRemote is to use it as a paintbrush to draw the art on the screen.

Finally, WiiRemote is chosen to be the input device because of following reasons:

a. WiiRemote is a new technology/input device which recently has been used in many projects.

b. WiiRemote has become very popular, so people are already familiar with this input device/controller; therefore, no further education is needed about how to use it.

c. WiiRemote is an input device that is most similar to the paintbrush. When people hold the WiiRemote, it's just like they are holding the paintbrush, which is one of the ideas I had from the beginning of the project—to create an interactive drawing interface that is most similar to a real drawing experience.

Review of Literature

When I started researching this thesis project, the first thing was to search an appropriate input device to be the paint tool, and I found the Nintendo WiiRemote control which was originally designed for the Wii game. There are several online documentations about the Nintendo WiiRemote.

■ Wiimote Project

2008
http://www.wiimoteproject.com/
The Wiimote Project is the official discussion forum about WiiRemote, which posts many WiiRemote applications projects and innovative projects. It is a useful study of the innovative applications of WiiRemote and helps me know what already has been done.

■ Johnny Chung Lee

2008
http://johnnylee.net/projects/wii/
Johnny Chung Lee is a PhD graduate student at Carnegie Mellon University in Human-Computer Interaction. He shows us the currently unexploited potential of the Nintendo's WiiRemote technology and provides plenty of new directions for game designers to consider.

■ WiiRemote and Flash

2008
http://www.wiiflash.org/
WiiFlash is a project dedicated to the WiiRemote and Flash® applications, and wiiflash.org is the WiiFlash project home. This site provides WiiFlash API documentation that helps me establish a WiiFlash Server and learn the WiiFlash ActionScript API in order to control the WiiRemote interactive with Flash objects.

Process

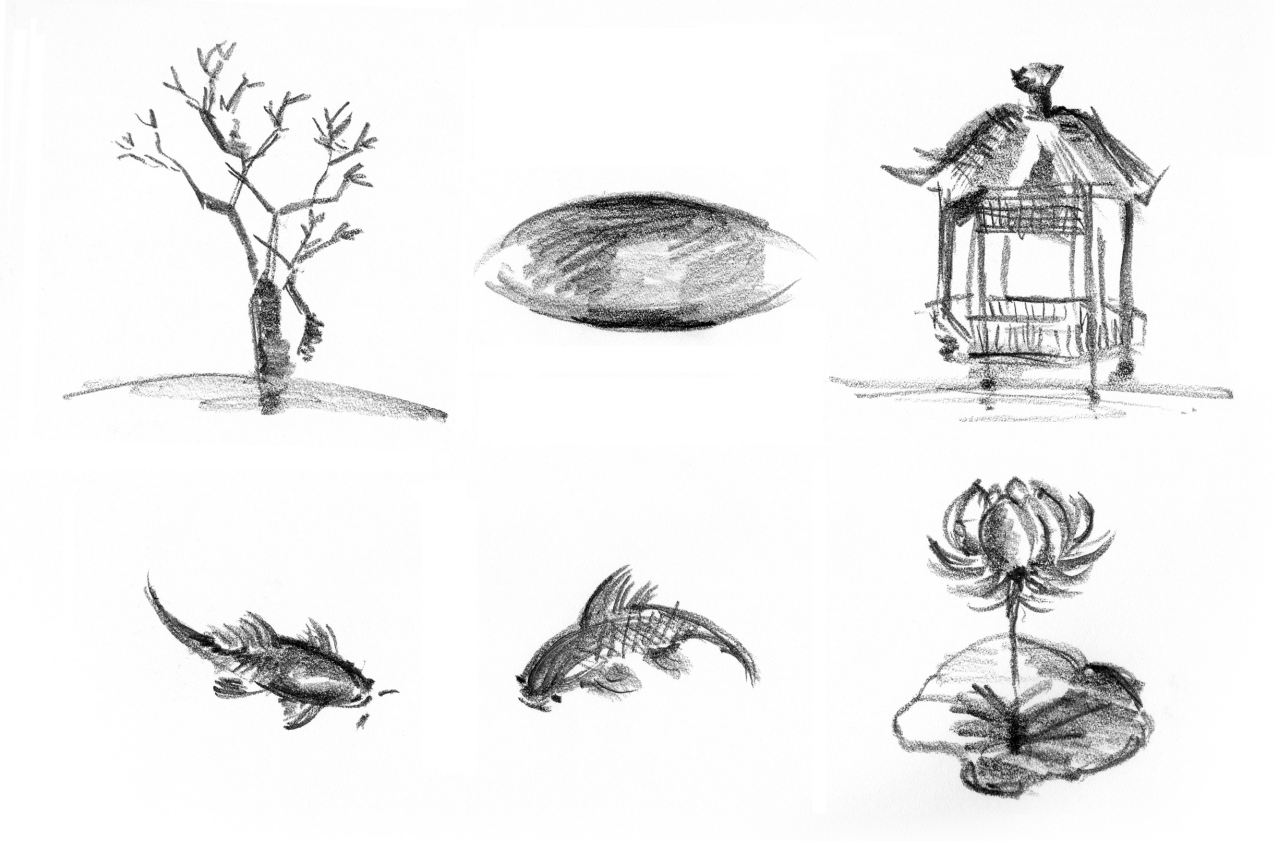
The working process for this thesis could be divided into two parts: art styles and WiiRemote. First, I will explain how the patterns are created for each art style and then explain how to connect the WiiRemote to the painting board.

5.1 Art Style Design: Chinese Ink Painting

Color is important in Chinese ink painting, or rather, the lack of color is important. The Chinese ink painting uses only black and white and is a shade of black ink. There are ten patterns in this painting board for users to choose. Those patterns are basic elements for Chinese ink painting, so users can create different pieces by using different combinations of those elements. The elements are created by using both 2D and 3D, in order to have the best outcome of the painting.

■ **Idea sketch**

■ Idea sketch



■ 2D / Photoshop Effects

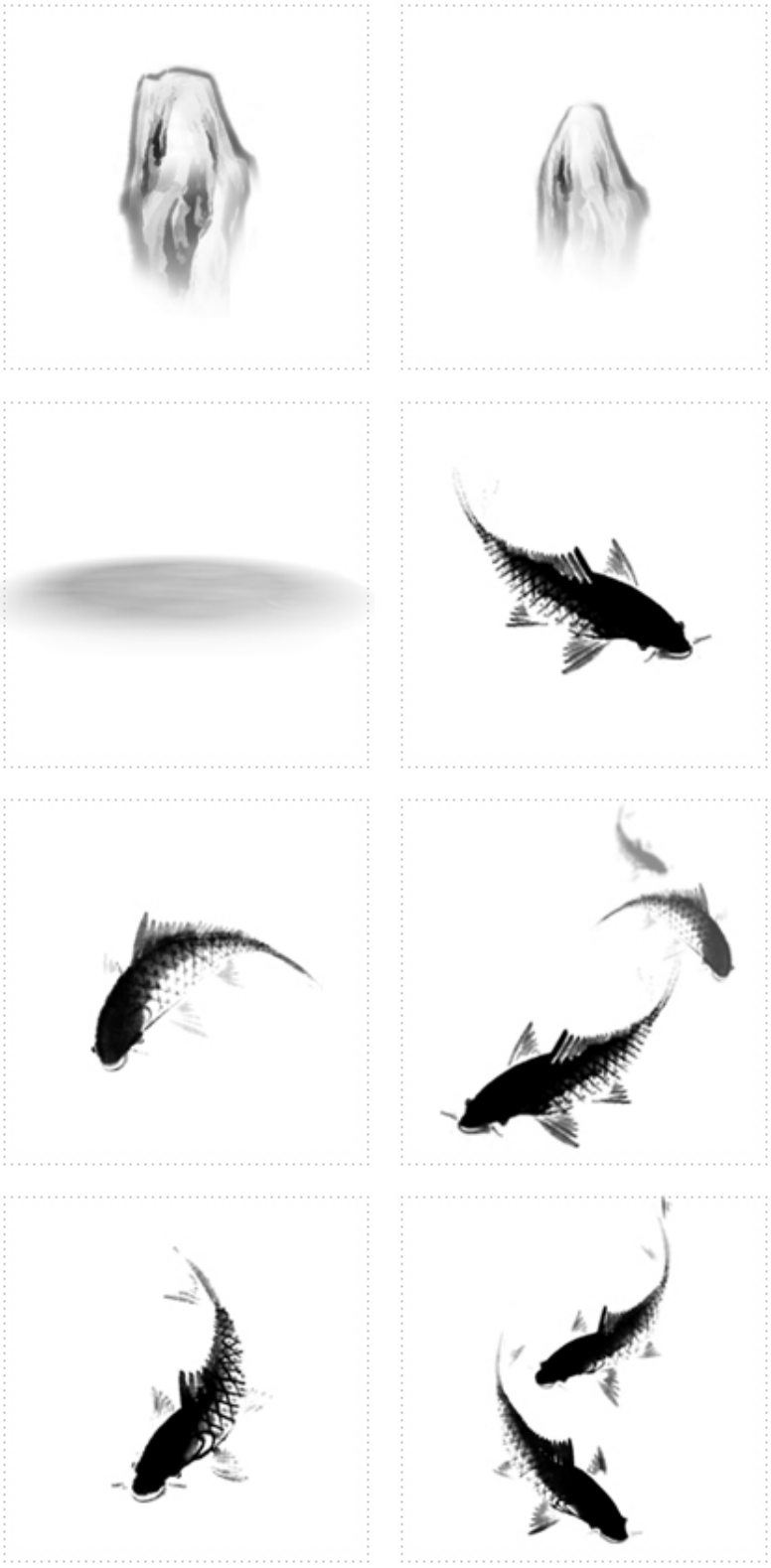
The following elements are created in 2D / Photoshop Effects:

› Landscape

› Rock

› Water

› Fish



■ 3D / Maya

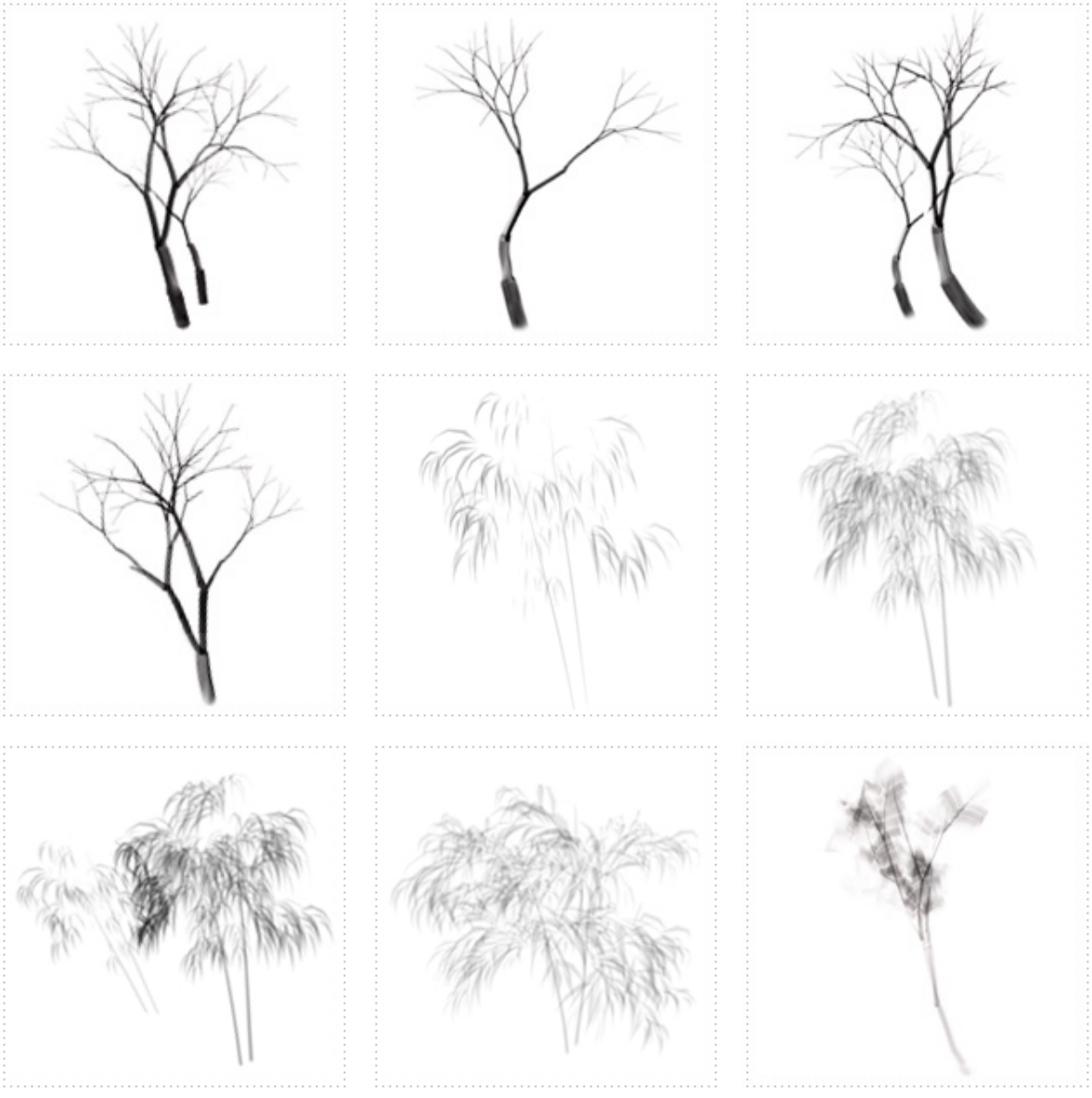
The following elements are created in 3D:

› Trees

› Bamboo

› Pavilion

› Lotus



■ **3D / Maya**

■ 3D / Maya

■ Seal and Calligraphy

One of the distinctive characteristics of Chinese ink painting is the use of inscriptions in poetry of calligraphy and of special seals as part of the painting. The calligraphy in this thesis is written by a Taiwanese calligrapher, Chia Ling Lu, with her personal seal.

Process

5.2 Art Style Design: Vector Art

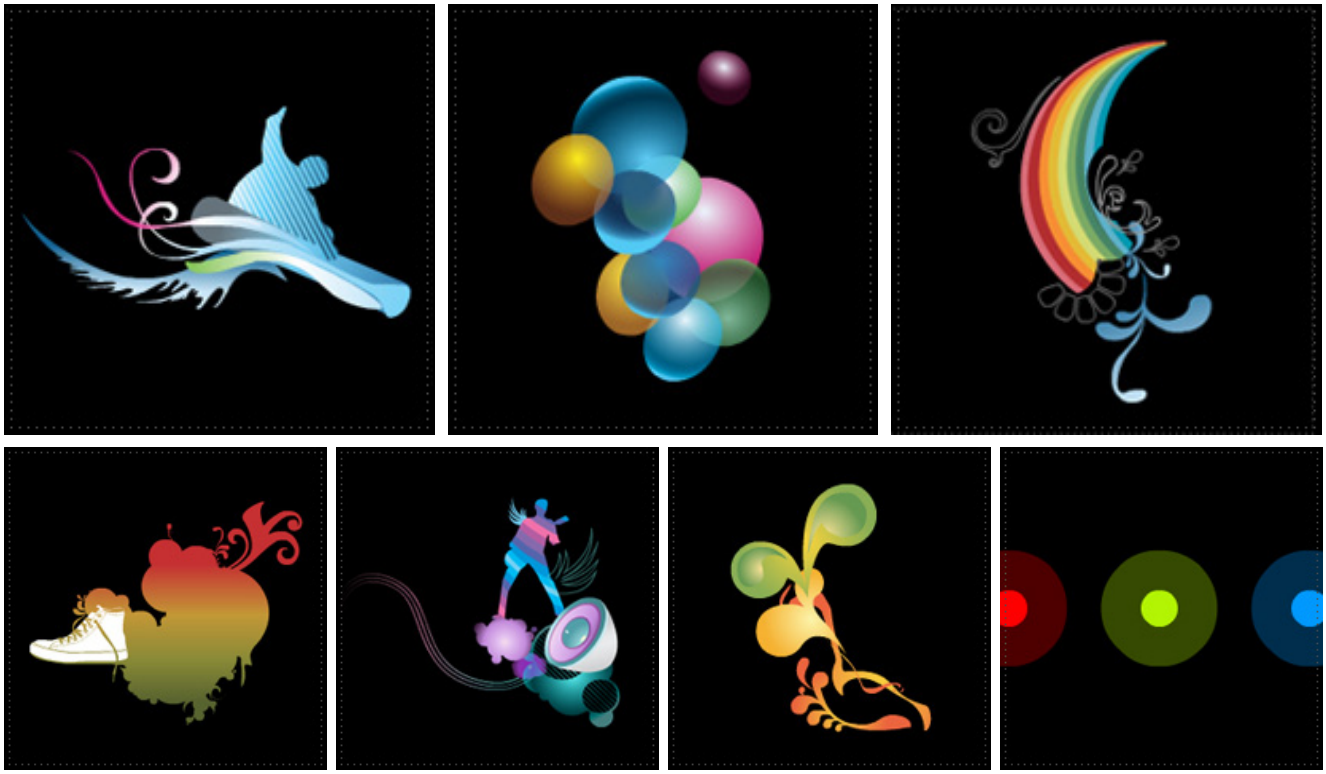
In contrast to Chinese ink painting, Vector Art is colorful. This art style consists of vivid graphics including many current life style elements, such as music and sports.

In order to create stronger contrast between the two art styles - Chinese ink painting and Vector art—bright colors and fluid lines are used often when designing patterns for Vector art. Also, motions are used to enhance the visual effect. The graphics are drawn in Illustrator first, and then Flash is used to animate the motion.

■ **Idea sketch**

■ **Motion frames**

■ **Final illustration**



Process

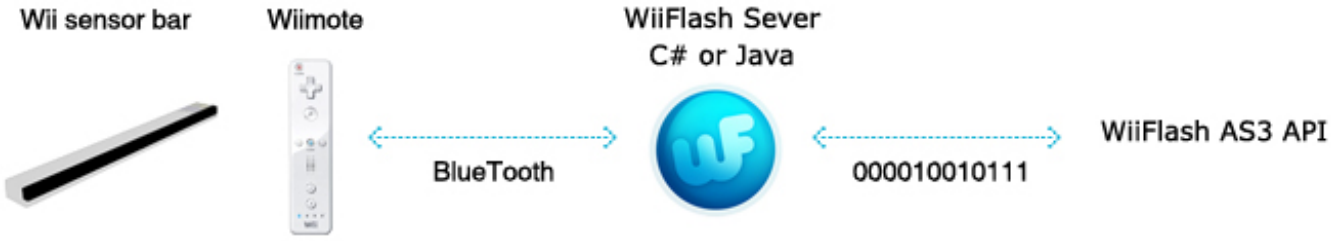
5.3 WiiRemote Connection

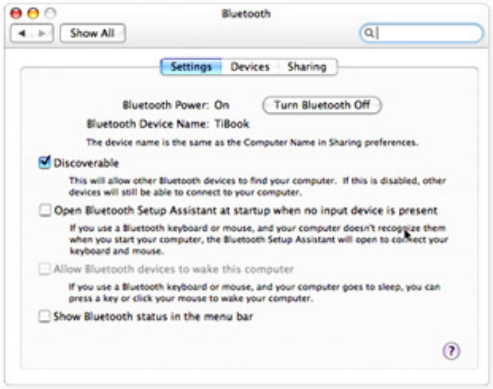
■ **What do I need?**

WiiFlash , a WiiRemote, and a Bluetooth dongle for PC

■ **How does it work?**

The connection process is shown in the following steps:





Step 1

Step 2

Step 1: Set up Bluetooth

Step 2: Connect to Nintendo WiiRemote

Click on the new connection button to search for any available devices. As soon as it starts browsing, hold buttons 1 and 2 at the same time on the WiiRemote to make it detectable. Release the buttons when you see the WiiRemote in the devices list. It should be detected as an HID device called “Nintendo RVL-CNT-01.″

Step 3: Getting the WiiFlash Server

Start the WiiFlash Server; it will tell you how many WiiRemotes are connected. If the Wii Controller is connected, the WiiFlash Server window will be blue. If not, it will be red, and there might be some problems with the Bluetooth connection.

› Connected

› Not connected

■ **Online test application**

To make sure the connection is secure, check the online test application*, which provides the readout of WiiRemote’s variables and indicates which buttons you are pressing.

WiiRemote Info:

› Sensor X: Value of the x acceleration sensor.

› Sensor Y: Value of the y acceleration sensor.

› Pitch: Pitch angle of the WiiRemote in radians.

› Roll: Roll angle of the WiiRemote in radians.

› Yaw: Yaw angle of the WiiRemote in radians.

› Battery level: WiiRemote battery level from 0 to 1 (full batteries).

*The online test application is provided by WiiFlash, http://wiiflash.bytearray.org/online-demo/

Process

5.4 WiiFlash Coding: WiiRemote Functions ActionScript

■ **Virtual Mouse - Using the WiiRemote as a mouse in Flash**

In order to use the WiiRemote as a mouse and execute events in Flash, it needs to create a virtual mouse in Flash.

› Solution: Senocular's VirtualMouse class*

This mouse, when moved around the screen, will trigger mouse events that would otherwise originate from the actual mouse controlled by the user.

Set the virtual mouse instances' X and Y values to the same as the pointer sprite. Trigger the WiiRemote button press event to fire off a virtual mouse click event.

*This is an open source which is created by Trevor McCauley

The following code demonstrates how to use WiiRemote to control virtual mouse:

› Import virtual mouse class and create a new virtual mouse

```
1  import org.wiiflash.Wiimote;
2  import org.wiiflash.IR;
3  import org.wiiflash.events.ButtonEvent;
4  import org.wiiflash.events.WiimoteEvent;
5  import com.senocular.ui.*;
6
7  // ----- create a new Virtual Mouse ---------------------------------
8  var mouse:VirtualMouse = new VirtualMouse(stage, cursor.x, cursor.y);
9
10 // ----- don't let it interact with the cursor ----------------------
11 mouse.ignore(cursor);
```

› Connect WiiRemote

```
13  // ----- connect Wiimote ---------------------------------------------
14  var wii:Wiimote = new Wiimote();
15  wii.connect();
16  wii.addEventListener(Event.CONNECT, connectWiimote);
17  wii.addEventListener(Event.CLOSE, closeConnection);
18  wii.addEventListener(WiimoteEvent.UPDATE, onRefresh);
19
20  // ----- connect Wiimote functions -----------------------------------
21  function connectWiimote(e:Event):void {
22      wii.leds = 1;
23      if (e.target.leds == 1) {// if is paint brush wiimote
24      }
25  }
26  function closeConnection(e:Event):void {
27      if (e.target.leds == 1) {// if is paint brush wiimote
28      }
29  }
```

› Update cursor position

```
30  function onRefresh(e:WiimoteEvent):void {
31      // ----- get IR position ----------------------------------------
32      // ----- x goes from 1-0 ----------------------------------------
33      var wiix = 1 - wii.ir.x1;
34
35      // ----- y goes from 0-1 ----------------------------------------
36      var wiiy = wii.ir.y1;
37
38      // ----- update cursor position ---------------------------------
39      if (wiix >= 0 && wiix <= 1024 && wiiy >= 0 && wiiy <= 768) {
40          mouse.x = wiix * 1024;
41          mouse.y = wiiy * 768;
42      } else {
43          if (mouse.x < 0 && wiix < 0) {
44              mouse.x == 0;
45          }
46          if (wiix > 1024) {
47              mouse.x == 1024;
48          }
49          if (wiiy < 0) {
50              mouse.y == 0;
51          }
52          if (wiiy > 768) {
53              mouse.y == 768;
54          }
55      }
56  }
```

› Move the WiiRemote pointer with mouse movements

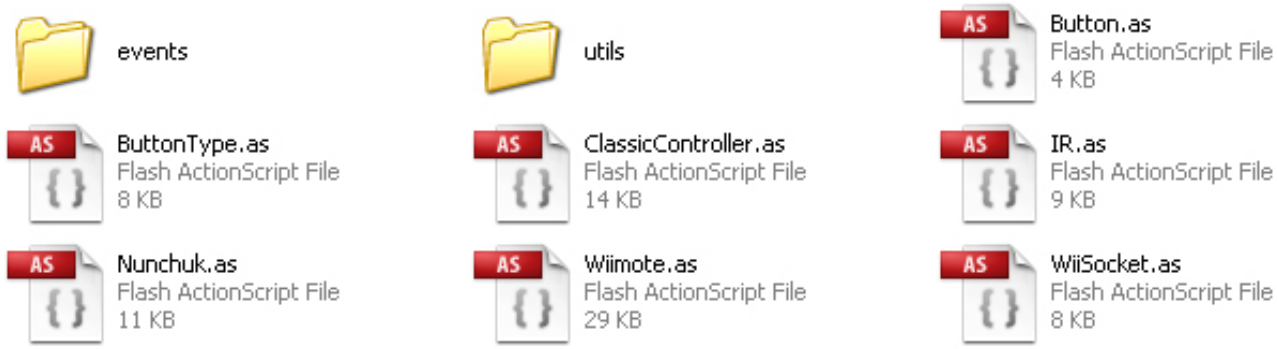
```
// ----- move the cursor with mouse movements ------------------------
stage.addEventListener(MouseEvent.MOUSE_MOVE, move);

function move(event:MouseEvent):void {
    // if event is of the type IVirtualMouseEvent
    // the event is from the virtual mouse
    // ----- move the cursor with mouse movements --------------------
    if (event is IVirtualMouseEvent){

        // ----- hide indicator -------------------------------------
        if (mouseIndicator.visible) mouseIndicator.visible = false;

        // ----- cursor is the virtual mouse ------------------------
        cursor.x = mouse.x;
        cursor.y = mouse.y;
    }
}
```

› Trigger the mouse event

```
// ----- play the click animation when the virtual mouse is pressed --
stage.addEventListener(MouseEvent.MOUSE_DOWN, down);
function down(event:MouseEvent):void {

    // ----- event is from the virtual mouse ------------------------
    if (event is IVirtualMouseEvent) {
        cursor.play();
    }
}

// ----- set up event handlers for interacting with the image --------
rotateLeftButton.addEventListener(MouseEvent.CLICK, rotateHandler);
rotateRightButton.addEventListener(MouseEvent.CLICK, rotateHandler);

// ----- rotate the image based on which button was pressed ----------
function rotateHandler(event:MouseEvent):void {
    image.rotation += (event.currentTarget == rotateLeftButton) ? -10 : 10;
}

// ----- set up handlers for press/click ----------------------------
wii.addEventListener(ButtonEvent.A_PRESS, doAction);
function doAction(e:ButtonEvent):void {
    mouse.click();
}
```

Final ActionScripts:

› Import WiiFlash class.



› Set up timer for Vector Art brush

```
18  // ----- set up timer for Vector brush ---------------------------------
19  function initVector() {
20      stop();
21
22      addChild(container);
23
24      timer = new Timer(100);
25      timer.addEventListener(TimerEvent.TIMER, drawDots);
26      timer.start();
27  }
```

› Draw Vector Art base on timer set up

```
29  // ----- Vector Art Drawing ----------------------------------------------
30  function drawDots(e:TimerEvent):void {
31      if (cursor.x>120 && btnClick == 1 && !stopDrawing && stageDrawing) {
32          var flower:flowers = new flowers();
33          container.addChild(flower);
34          bigArray.push(flower);
35          flowerArray.push(flower);
36          flower.x = mouse.x;
37          flower.y = mouse.y;
38          flower.scaleX = flower.scaleY = Math.random()*1.2;
39          flower.alpha = Math.random()*1.2;
40          flower.rotation = Math.random()*360;
41      }
42
43      if (cursor.x>120 && btnClick == 2 && !stopDrawing && stageDrawing) {
44          var circle:circles = new circles();
45          container.addChild(circle);
46          bigArray.push(circle);
47          circlesArray.push(circle);
```

```
48          circle.x = mouse.x;
49          circle.y = mouse.y;
50          circle.scaleX = circle.scaleY = Math.random()*1.2;
51          circle.alpha = Math.random()*1.2;
52          circle.rotation = Math.random()*360;
53      }
54      if (cursor.x>120 && btnClick == 3 && !stopDrawing && stageDrawing) {
55          var rainbow:rainbows = new rainbows();
56          container.addChild(rainbow);
57          bigArray.push(rainbow);
58          rainbowArray.push(rainbow);
59          rainbow.x = mouse.x;
60          rainbow.y = mouse.y;
61          rainbow.scaleX = rainbow.scaleY = Math.random()*1.2;
62          rainbow.alpha = Math.random()*1.5;
63          rainbow.rotation = Math.random()*360;
64      }
65      if (cursor.x>120 && btnClick == 4 && !stopDrawing && stageDrawing) {
66          var sports:Sports = new Sports();
67          container.addChild(sports);
68          bigArray.push(sports);
69          sportsArray.push(sports);
70          sports.x = mouse.x;
71          sports.y = mouse.y;
72          sports.scaleX = sports.scaleY = Math.random()*1.2;
73          sports.alpha = Math.random()*1.5;
74          sports.rotation = Math.random()*360;
75      }
76      if (cursor.x>120 && btnClick == 5 && !stopDrawing && stageDrawing) {
77          var musicwing:Musicwing = new Musicwing();
78          container.addChild(musicwing);
79          bigArray.push(musicwing);
80          musicwingArray.push(musicwing);
81          musicwing.x = mouse.x;
82          musicwing.y = mouse.y;
83          musicwing.scaleX = musicwing.scaleY = Math.random()*1.5;
84          musicwing.alpha = Math.random()*1.5;
85          musicwing.rotation = Math.random()*180;
86      }
87      if (cursor.x>120 && btnClick == 6 && !stopDrawing && stageDrawing) {
88          var splashcloud:Splashcloud = new Splashcloud();
89          container.addChild(splashcloud);
90          bigArray.push(splashcloud);
91          splashcloudArray.push(splashcloud);
92          splashcloud.x = mouse.x;
93          splashcloud.y = mouse.y;
94          splashcloud.scaleX = splashcloud.scaleY = Math.random()*1.5;
95          splashcloud.alpha = Math.random()*1.5;
96          splashcloud.rotation = Math.random()*180;
97      }
```

```
98      if (cursor.x>120 && btnClick == 7 && !stopDrawing && stageDrawing) {
99          var dot:Dot = new Dot();
100         container.addChild(dot);
101         bigArray.push(dot);
102         dotArray.push(dot);
103         dot.x = mouse.x;
104         dot.y = mouse.y;
105         dot.gotoAndStop(Math.ceil(Math.random()*4));
106         dot.scaleX = dot.scaleY = Math.random()*2 + 0.3;
107         dot.alpha = Math.random()*1.5;
108     }
109 }
110
```

› Switch to Vector Art pattern

```
111 // ----- Vector Art Chosen ---------------------------------------------
112 for (var i = 1; i <= 7; i++) {
113     this["artBtn" + i].addEventListener(MouseEvent.CLICK, changePattern);
114 }
115
116 function changePattern(e:MouseEvent) {
117     switch (e.target.name) {
118         case "artBtn1" :
119             stopDrawing = false;
120             stageDrawing = false;
121             btnClick = 1;
122             break;
123         case "artBtn2" :
124             stopDrawing = false;
125             stageDrawing = false;
126             btnClick = 2;
127             break;
128         case "artBtn3" :
129             stopDrawing = false;
130             stageDrawing = false;
131             btnClick = 3;
132             break;
133         case "artBtn4" :
134             stopDrawing = false;
135             stageDrawing = false;
136             btnClick = 4;
137             break;
138         case "artBtn5" :
139             stopDrawing = false;
140             stageDrawing = false;
141             btnClick = 5;
142             break;
143         case "artBtn6" :
144             stopDrawing = false;
145             stageDrawing = false;
```

```
146             btnClick = 6;
147             break;
148         case "artBtn7" :
149             stopDrawing = false;
150             stageDrawing = false;
151             btnClick = 7;
152             break;
153     }
154 }
```

› Set up WiiRemote button event

```
155 // ----- set up Wiimote buttons ----------------------------------------
156 wii.addEventListener(ButtonEvent.A_PRESS, doAction);
157 wii.addEventListener(ButtonEvent.A_PRESS, pressDrawing);
158 wii.addEventListener(ButtonEvent.A_RELEASE, releaseDrawing);
159 wii.addEventListener(ButtonEvent.B_PRESS, doClear);
160
161 function pressDrawing(e:ButtonEvent):void {
162     stageDrawing = true;
163     stopDrawing = false;
164 }
165 function releaseDrawing(e:ButtonEvent):void {
166     stopDrawing = true;
167 }
168 function doAction(e:ButtonEvent):void {
169     mouse.click();
170 }
171 function doClear(e:ButtonEvent):void {
172     stopDrawing = true;
173     for (var i:int=0; i<bigArray.length; i++) {
174         container.removeChild(bigArray[i]);
175     }
176     bigArray.splice(0, bigArray.length);
177 }
```

› Clean the stage and go back to Main stage

```
179 // ----- back to Main Stage --------------------------------------
180 artbackBtn.addEventListener(MouseEvent.CLICK, easeBackBase);
181
182 function easeBackBase(e:MouseEvent) {
183     stopDrawing = true;
184     stageDrawing = false;
185     for (var i:int=0; i<bigArray.length; i++) {
186         container.removeChild(bigArray[i]);
187     }
188     bigArray.splice(0, bigArray.length);
189     timer.removeEventListener(TimerEvent.TIMER, drawDots);
190     var tweenbase:Tween = new Tween(selectStyle, "x", Regular.easeOut, selectStyle
```

```
191     gotoAndStop("base");
192     var timerFinish:Timer = new Timer(1100,1);
193     timerFinish.start();
194     timerFinish.addEventListener(TimerEvent.TIMER_COMPLETE, easeFinish);
195     function easeFinish(e:TimerEvent) {
196         selectStyle.x = -1024;
197     }
198 }
```

› Set up timer for Chinese ink pinking brush

```
18  // ----- set up timer for Ink Painting brush -------------------------------
19  function initInk() {
20      stop();
21
22      addChild(inkContainer);
23
24      timerInk = new Timer(100);
25      timerInk.addEventListener(TimerEvent.TIMER, startInkPainting);
26      timerInk.start();
27  }
```

› Draw Chinese ink painting base on timer set up

```
29  // ----- Ink Painting Drawing ----------------------------------------------
30  function startInkPainting(e:TimerEvent):void {
31      if (cursor.x<900 && inkbtnClick == 1 && !stopPainting && stagePainting) {
32          var water:Water = new Water();
33          inkContainer.addChild(water);
34          inkArray.push(water);
35          waterArray.push(water);
36          water.x = mouse.x;
37          water.y = mouse.y;
38          water.scaleX = water.scaleY = Math.random()*.7 + .3;
39          water.alpha = Math.random()*.8 + .2;
40      }
41
42      if (cursor.x<900 && inkbtnClick == 2 && !stopPainting && stagePainting) {
43          var pavilion:Pavilion = new Pavilion();
44          inkContainer.addChild(pavilion);
45          inkArray.push(pavilion);
46          pavilionArray.push(pavilion);
47          pavilion.x = mouse.x;
48          pavilion.y = mouse.y;
49          pavilion.gotoAndStop(Math.ceil(Math.random()*7));
50          pavilion.scaleX = pavilion.scaleY = Math.random()*.6 + .4;
51          pavilion.alpha = Math.random()*.9 + .1;
52      }
53
```

```
54      if (cursor.x<900 && inkbtnClick == 3 && !stopPainting && stagePainting) {
55          var treea:TreeA = new TreeA();
56          inkContainer.addChild(treea);
57          inkArray.push(treea);
58          treeaArray.push(treea);
59          treea.x = mouse.x;
60          treea.y = mouse.y;
61          treea.gotoAndStop(Math.ceil(Math.random()*4));
62          treea.scaleX = treea.scaleY = Math.random()*.6;
63          treea.alpha = Math.random()*1;
64          treea.rotation = Math.random()*30;
65      }
66      if (cursor.x<900 && inkbtnClick == 4 && !stopPainting && stagePainting) {
67          var lotus:Lotus = new Lotus();
68          inkContainer.addChild(lotus);
69          inkArray.push(lotus);
70          lotusArray.push(lotus);
71          lotus.x = mouse.x;
72          lotus.y = mouse.y;
73          lotus.gotoAndStop(Math.ceil(Math.random()*6));
74          lotus.scaleX = lotus.scaleY = Math.random()*.4 + .6;
75          lotus.alpha = Math.random()*.8 + .2;
76      }
77      if (cursor.x<900 && inkbtnClick == 5 && !stopPainting && stagePainting) {
78          var treec:TreeC = new TreeC();
79          inkContainer.addChild(treec);
80          inkArray.push(treec);
81          treecArray.push(treec);
82          treec.x = mouse.x;
83          treec.y = mouse.y;
84          treec.gotoAndStop(Math.ceil(Math.random()*4));
85          treec.scaleX = treec.scaleY = Math.random()*1;
86          treec.alpha = Math.random()*.8 + .2;
87          treec.rotation = Math.random()*20;
88      }
89      if (cursor.x<900 && inkbtnClick == 6 && !stopPainting && stagePainting) {
90          var mountainc:MountainC = new MountainC();
91          inkContainer.addChild(mountainc);
92          inkArray.push(mountainc);
93          mountaincArray.push(mountainc);
94          mountainc.x = mouse.x;
95          mountainc.y = mouse.y;
96          mountainc.gotoAndStop(Math.ceil(Math.random()*2));
97          mountainc.scaleX = mountainc.scaleY = Math.random()*.7 + .3;
98          mountainc.alpha = Math.random()*.9 + .1;
99      }
100     if (cursor.x<900 && inkbtnClick == 7 && !stopPainting && stagePainting) {
101         var mountainm:MountainM = new MountainM();
102         inkContainer.addChild(mountainm);
103         inkArray.push(mountainm);
104         mountainmArray.push(mountainm);
```

```
105             mountainm.x = mouse.x;
106             mountainm.y = mouse.y;
107             mountainm.gotoAndStop(Math.ceil(Math.random()*3));
108             mountainm.scaleX = mountainm.scaleY = Math.random()*1;
109             mountainm.alpha = Math.random()*1;
110         }
111         if (cursor.x<900 && inkbtnClick == 8 && !stopPainting && stagePainting) {
112             var catfish:Catfish = new Catfish();
113             inkContainer.addChild(catfish);
114             inkArray.push(catfish);
115             catfishArray.push(catfish);
116             catfish.x = mouse.x;
117             catfish.y = mouse.y;
118             catfish.gotoAndStop(Math.ceil(Math.random()*5));
119             catfish.scaleX = catfish.scaleY = Math.random()*.8 + .2;
120             catfish.alpha = Math.random()*.8 + .2;
121         }
122         if (cursor.x<900 && inkbtnClick == 9 && !stopPainting && stagePainting) {
123             var calligraphy:Calligraphy = new Calligraphy();
124             inkContainer.addChild(calligraphy);
125             inkArray.push(calligraphy);
126             calligraphyArray.push(calligraphy);
127             calligraphy.x = mouse.x;
128             calligraphy.y = mouse.y;
129             calligraphy.gotoAndStop(Math.ceil(Math.random()*3));
130         }
131         if (cursor.x<900 && inkbtnClick == 10 && !stopPainting && stagePainting) {
132             var treeb:TreeB = new TreeB();
133             inkContainer.addChild(treeb);
134             inkArray.push(treeb);
135             treebArray.push(treeb);
136             treeb.x = mouse.x;
137             treeb.y = mouse.y;
138             treeb.gotoAndStop(Math.ceil(Math.random()*4));
139             treeb.scaleX = treeb.scaleY = Math.random()*.8 + .2;
140             treeb.alpha = Math.random()*1;
141             treeb.rotation = Math.random()*30;
142         }
```

› Switch to Chinese ink painting pattern

```
145 // ----- Ink Brush Chosen ------------------------------------------------------
146 for (var n = 1; n <= 10; n++) {
147     this["inkBtn" + n].addEventListener(MouseEvent.CLICK, changeInkBrushes);
148 }
149
150 function changeInkBrushes(e:MouseEvent) {
151     switch (e.target.name) {
152         case "inkBtn1" :
153             stopPainting = false;
```

```
154             stagePainting = false;
155             inkbtnClick = 1;
156             break;
157         case "inkBtn2" :
158             stopPainting = false;
159             stagePainting = false;
160             inkbtnClick = 2;
161             break;
162         case "inkBtn3" :
163             stopPainting = false;
164             stagePainting = false;
165             inkbtnClick = 3;
166             break;
167         case "inkBtn4" :
168             stopPainting = false;
169             stagePainting = false;
170             inkbtnClick = 4;
171             break;
172         case "inkBtn5" :
173             stopPainting = false;
174             stagePainting = false;
175             inkbtnClick = 5;
176             break;
177         case "inkBtn6" :
178             stopPainting = false;
179             stagePainting = false;
180             inkbtnClick = 6;
181             break;
182         case "inkBtn7" :
183             stopPainting = false;
184             stagePainting = false;
185             inkbtnClick = 7;
186             break;
187         case "inkBtn8" :
188             stopPainting = false;
189             stagePainting = false;
190             inkbtnClick = 8;
191             break;
192         case "inkBtn9" :
193             stopPainting = false;
194             stagePainting = false;
195             inkbtnClick = 9;
196             break;
197         case "inkBtn10" :
198             stopPainting = false;
199             stagePainting = false;
200             inkbtnClick = 10;
201             break;
202     }
203 }
```

› Switch to Chinese ink painting pattern

```
// ----- set up Wiimote buttons ---------------------------------------
wii.addEventListener(ButtonEvent.A_PRESS, doInkAction);
wii.addEventListener(ButtonEvent.A_PRESS, pressPainting);
wii.addEventListener(ButtonEvent.A_RELEASE, releasePainting);
wii.addEventListener(ButtonEvent.B_PRESS, doInkClear);

function pressPainting(e:ButtonEvent):void {
    stagePainting = true;
    stopPainting = false;
}
function releasePainting(e:ButtonEvent):void {
    stopPainting = true;
}
function doInkAction(e:ButtonEvent):void {
    mouse.click();
}
function doInkClear(e:ButtonEvent):void {
    stopPainting = true;
    for (var i:int=0; i<inkArray.length; i++) {
        inkContainer.removeChild(inkArray[i]);
    }
    inkArray.splice(0, inkArray.length);

}
```

› Clean the stage and go back to Main stage

```
// ----- back to Main Stage -------------------------------------------
inkbackBtn.addEventListener(MouseEvent.CLICK, inkeaseBackBase);

function inkeaseBackBase(e:MouseEvent) {
    stopPainting = true;
    for (var i:int=0; i<inkArray.length; i++) {
        inkContainer.removeChild(inkArray[i]);
    }
    inkArray.splice(0, inkArray.length);
    gotoAndStop("base");
    var tweeninkbase:Tween = new Tween(selectStyle, "x", Regular.easeOut,
    timerInk.removeEventListener(TimerEvent.TIMER, startInkPainting);
    var timerInkFinish:Timer = new Timer(1100,1);
    timerInkFinish.start();
    timerInkFinish.addEventListener(TimerEvent.TIMER_COMPLETE, easeInkFinish);
    function easeInkFinish(e:TimerEvent) {
        selectStyle.x = -1024;
    }
}
```

Process

5.5 Interface Design

Main stage

Chinese Ink Painting Scene

Vector Art Scene

■ Main stage

The two art styles represent Eastern and Western cultures, respectively. In order to create a strong contrast between two art styles, the interface is divided into two parts with white and black backgrounds, respectively. On the right side, the white background is used to mimic the characteristic of Chinese ink painting, making the painting with ink on white papers; while on the left side, the black background is used to make the colors more outstanding. Therefore, the main stage is presented as black-and-white versus colors, Chinese ink painting versus Vector Art, and Eastern versus Western cultures. Meanwhile, the icon of each art style is a half circle, and two half circles become a circle in the middle which conveys the meaning of culture fusion.

■ Chinese Ink Painting Scene

Because of the characteristic of Chinese ink painting—making the painting with ink on white papers—use a white background to mimic the paper.

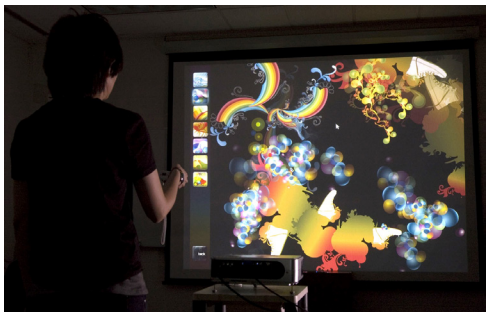
■ Vector Art Scene

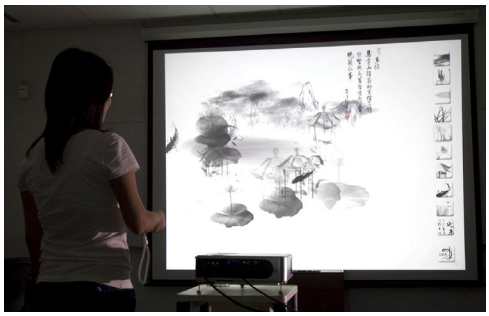
To create the strongest contrast with the white background, the black background is used for the Vector Art painting board. Also, the black background makes graphics colors stand out more boldly.

Demonstration

Imagine RIT Exhibit / May 2, 2009

The Final Project:
The different pieces of the installation included a wall as a screen, a projector, a laptop, a Bluetooth dongle, a Wii Sensor Bar and a WiiRemote (all sitting on their respective pedestals).





■ **User feedback**

› The sequences of graphics come out too fast. Sometimes the graphics pile up together, and it makes it hard to see how the graphics look like.

› After drawing awhile, the movement becomes slower. The drawing is not as smooth as it is in the beginning.

› Interface is clear and easy-to-use.

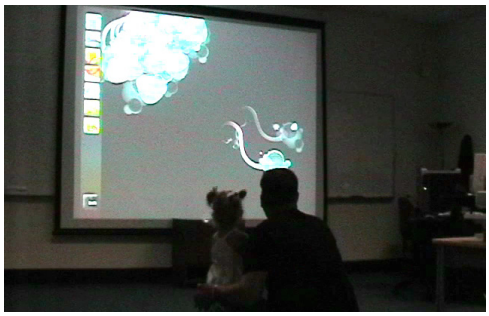
› The idea of having both east and west styles is fun, because users can experience both art styles.

› Users are really familiar with Wii Remote Controller, even kids. They know how to control the Remote instantly.

› It's an interesting experience for users to create their own Chinese ink paintings in such a fun and easy way.

› Parents think that it's an educational tool for kids to boost their inspiration and creativity.

Process



■ **Improvement**

› Adjust the timer set up in ActionScript; decrease the frequency, to make the graphics come out one after another, so it will appear more smoothly for users.

› Try to fix the codes. After adding graphics to the stage, make sure when users click clear the stage, all the graphics will be removed from the stage. Also, for the coding aspect, remove all the functions in ActionScript so it won’t keep adding the loading.

Conclusion

This project is about creating an interactive installation piece which demonstrates the interactions between computers and humans and conveys a sense of fun, simplicity, and openness. Through the combination of artistic concepts and programming language, users will be taken into the fun world of digital art.

At the conception stage of this project, more functions were considered for users, such as enlarging the graphics or using the yaw function of WiiRemote to control the graphics. However, at the development stage, I found that if users had too many functions, this would create confusion or inconvenience. Also, the loading for hardware would be too heavy and it would take too much processing time. Therefore, a simple interface with easy-to-use functions was created. Furthermore, under the current specification, it would not provide more patterns for users because of the processing time. If I put more patterns on the painting board, users would experience a lag when they paint the scene.

The most breakthrough point of this thesis project is the incorporation of WiiRemote. WiiRemote used in computer graphic design is not popular, or is still at the experiment stage. Before WiiFlash was released, it was difficult to connect WiiRemote and Flash, but with WiiFlash it has become possible. Therefore, when the connection of WiiRemote and Flash was made, I was able to do the consequential work. During the development stage, researching and testing how to build up the relationship between WiiRemote, Wii Sensor Bar, and Flash took a long time, as did figuring out how to use WiiRemote to trigger a mouse event in Flash so users could control graphics.

As a CGD-major student, my idea for this thesis project is that it should be related to both design and programming, and this thesis managed both well. For the design part, I researched many art styles, and I designed some essential graphics for the art styles chosen, Vector Art and Chinese ink painting. For the programming part, I studied WiiFlash language and the functions of the up-to-date technology, WiiRemote.

The result of the project came out as expected. At the Imagine RIT Exhibit, people thought the installation piece was fun and easy to use, even for kids. Also, the project was submitted to the 2009 Adobe Design Achievement Awards since Adobe software was used to finish the project, and it received the honor of Semifinalist.

Bibliography

Jean, L. (1984). *Chinese Ink Painting: Techniques in Shades of Black.*

Keith P. (2007). *Foundation ActionScript 3.0 Animation.*

Lott, J. (2007). *ActionScript 3.0 cookbook*. CA: O'Reilly.

Moock, C. (2007). *Essential ActionScript 3.0.* CA: O'Reilly.

Yangmu, W. (2000). *The Techniques of Chinese Painting.*

Mai Mai, S. (1959). *The Way of Chinese Painting: Its Ideas and Technique.*

Appendices

Thesis Proposal

Thesis Proposal

Interactive WiiArt

Abstract

This project is an interactive project that allows users to explore graphics and painting by using a mouse or other input device. The project focuses on creating immersive and highly interactive environments as well as exciting, playful, and enjoyable experiences.

The purpose of this project is to provide a 3D platform for users to easily create an art work with depth. Moreover, the default patterns let users have multiple choices to paint the scene, and even though the users don't have any art, design background, or experience, they can still create their own masterpiece easily.

Problem Statement

Traditionally, interactive drawing presents a 2D visual interface to users, and drawing tools or platforms today are quite similar. This interactive interface will provide users with a visually rich experience. Furthermore, users will experience the depth of the art in this 3D environment.

Since users are in a 3D environment, by using a mouse or wiimote to zoom in or zoom out, the scene can provide them a virtual camera control experience, yet they can then quickly and easily create 3D-like art piece. Most of the users will not have an art or design background, so the paint palette provides various default patterns which are a sequence of graphics instead of a single color or shape.

This project brings another dimension to exploring the art work and changes the way people use the paint tool by adding interactive graphics and other input devices which provide different ways to experience the painting.

Scope

The visual elements of the project include graphics and traditional Chinese painting that require illustration and painting skill. The area of Computer Graphics Design will be user interaction that incorporates Flash and input device integration.

Literature Survey

Introduction

When I first started researching my thesis project, I was searching different input devices to be my paint tool, and I found the Wiimote which was originally designed for the Wii game.

In order to implement this thesis project, I have to learn Papervision3D, which is an open source realtime 3D engine for the Flash platform. Also, connecting wiimote with Flash requires establishing a WiiFlash Server on PC and learning WiiFlash ActionScript API.

Online Documentation

Papervision3D
2007
http://blog.papervision3d.org/
This is the official developers' blog of the Papervision3D, which posts many Papervision3D applications projects and innovative projects. It is a useful study of the innovative applications of Papervision3D and will help me know what already has been done.

Papervision3D - official resource site
2007
http://www.pv3d.org/
pv3d.org is the sister site for papervision3d.org. It's the official resource site for Papervision3D instruction. This site provides examples and tutorials that can guide me from zero with a structured walk-through of the Papervision3D workflow.

Wiimote and Flash
2008
http://www.wiiflash.org/
WiiFlash is a project dedicated to the Wiimote and Flash® applications, and wiiflash.org is the WiiFlash project home. This site provides WiiFlash API ocumentation that will help me establish a WiiFlash Server and learn the WiiFlash ActionScript API in order to control the Wiimote interactive with Flash objects.

Project Description / Methodology

Design - Choose two different styles of painting boards. One provides colorful modern graphics patterns, and the other provides traditional Chinese painting patterns. Subjects or participants –

The target audience will be all users, and approximately 20 participants will be my usability testers.

Instruments - The questionnaires will focus on attitude questions, and the biggest advantage is that a usability questionnaire gives feedback from the point of view of the user. There's no right or wrong answer, and the results can help me improve the usability of my project.

Procedure - In order to implement this thesis project, I have to learn Papervision3D, which is an open source realtime 3D engine for the Flash platform. Also, connecting Wiimote with Flash requires establishing a WiiFlash Server on PC and learning WiiFlash ActionScript API.

Limitations

Using Papervision3D to create a 3D environment may not be as flexible as 3D software, and integration with input device Wiimote may encounter some programming problems.

Implications

Physical computing - The idea is controlling the painting tools from the physical world. Use implications from physical computing for understanding human beings' relationship to the digital world. In this project, the Wiimote control is used to translate the information input to a software system which is Flash.

Ink Painting Simulation - Present a method for simulating ink dispersion in absorbent paper for art creation. These implications include: effect techniques of Chinese ink painting style; dynamic cloud, fog, tree, water, and growth principles. It helps to explore a new joint point between traditional painting and the digital world.project.

Marketing Plan

Reviewing this project requires Wiimote and Bluetooth, posting the project to the online WiiFlash community where users can meet the requirements and where some of them are experts in Wii in conjunction with Flash field. Also, it invites the graphic designers and artists to review whether the project delivers its educational purpose.

Competition: 2008 Adobe Design Achievement Awards

Contest begins on November 1, 2007 and ends on May 2, 2008. All entries must be received by Adobe no later than 5 p.m. Pacific time on May 2, 2008.

Budget

$200

Target Audience

Age	Any
Gender	Male and Female
Interests	Art, Design, Drawing, Painting
Education Level	Any
Occupation	Any
Computer Experience	Intermediate

Scenario

Scenario 1 - James is a twenty-one-year-old college student who is majoring in engineering. He likes art and likes to discover new things. This project will provide him with an interactive painting experience and the ability to creative his own art piece easily.

Scenario 2 - Jessica is twenty-nine years old and is a secretary of the Chi Omega sisterhood. She likes Eastern culture, and this project will give her an experience with Chinese painting without knowing Chinese painting tools and palette.

Technical requirements

Reviewing this project requires Wiimote and Bluetooth, posting the project to the online WiiFlash community where users can meet the requirements and where some of them are experts in Wii in conjunction with Flash field. Also, it invites the graphic designers and artists to review whether the project delivers its educational purpose.

Competition: 2008 Adobe Design Achievement Awards

Contest begins on November 1, 2007 and ends on May 2, 2008. All entries must be received by Adobe no later than 5 p.m. Pacific time on May 2, 2008.

Budget

$200

Target Audience

Age	Any
Gender	Male and Female
Interests	Art, Design, Drawing, Painting
Education Level	Any
Occupation	Any
Computer Experience	Intermediate

Technical requirements

Software

Adobe Flash professional
Papervision3D
WiiFlash ActionScript API

Hardware

Macintosh G5
Windows VISTA or Windows XP Professional
1.00GB of RAM
50GB free disk space
One projector or output devices or monitor
Bluetooth dongle
Wiimote