액션스크립트정석

클래스 관계보는것

<http://livedocs.adobe.com/flex/3/langref/index.html>

* [Flex 4 Documentation](http://www.adobe.com/devnet/flex/?view=documentation)
* [Flex 3 Documentation](http://livedocs.adobe.com/flex%20/3/langref/3/html/index.html)
* [Flex 2.0.1 Documentation](http://livedocs.adobe.com/flex%20/3/langref/201/)
* [Flex 2 Documentation](http://livedocs.adobe.com/flex%20/3/langref/2/)
* [Flex 1.5 Documentation](http://livedocs.adobe.com/flex%20/3/langref/15/)
* [Flex 1.0 Documentation](http://livedocs.adobe.com/flex%20/3/langref/1/flex_docs/index.html)

또는 Flex Builder -> [help] -> [Help Contents] ->좌측 Adobe Flex ? Language Reference

그래픽 Graphic

**package**

{

**import** flash.display.Sprite;

**public** **class** flexActionScriptTest **extends** Sprite

{

**public** **function** flexActionScriptTest()

{

*// 원의 테두리선 색상 지정*

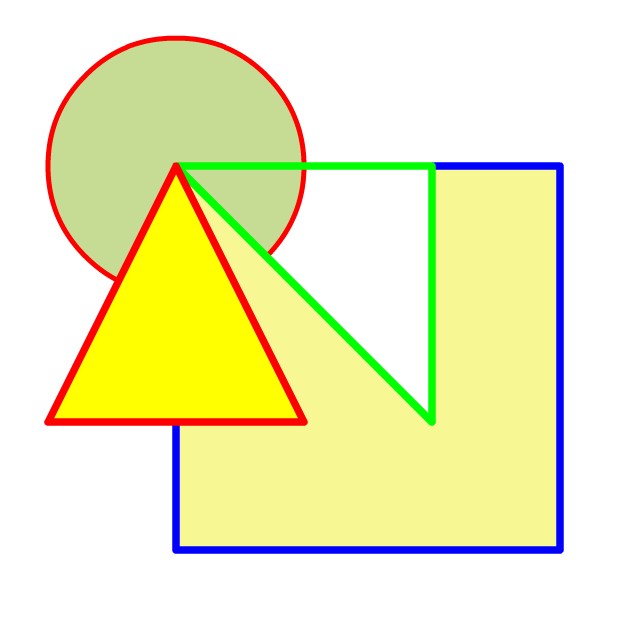
**this**.graphics.lineStyle(2, 0xFF0000);

*//원의 색상 지정*

**this**.graphics.beginFill(0xC6DB94);

*//원그리기 시작 좌표와 반지름 설정*

**this**.graphics.drawCircle(100, 100, 50);



*// 사각형의 테두리선 두께, 색상 지정*

**this**.graphics.lineStyle(3, 0x0000FF);

*//사각형의 색상 지정*

**this**.graphics.beginFill(0xF7F794);

*//사각형 그리기 시작 좌표와 가로, 세로 설정*

**this**.graphics.drawRect(100, 100, 150, 150);

*// graphics.clear();*

*// 직선의 색상과 두께 지정*

**this**.graphics.lineStyle(3, 0x00FF00);

*// 직선 그리기 시작점을 (100, 100)으로 이동*

**this**.graphics.moveTo(100, 100);

*// 좌표 (200, 200)까지 직선 그리기*

**this**.graphics.lineTo(120, 120);

*// 직선의 색상과 두께 지정*

**this**.graphics.lineStyle(3, 0x00ff00);

*// 직선 그리기 시작점을 (100, 100)으로 이동*

**this**.graphics.moveTo(100, 100);

*// 꼭지점 (200, 200), (200, 100), (100, 100)인 삼각형 그리기*

**this**.graphics.lineTo(200, 200);

**this**.graphics.lineTo(200, 100);

**this**.graphics.lineTo(100, 100);

**this**.graphics.lineStyle(3, 0xFF0000);

*// 채우기 색상 지정*

**this**.graphics.beginFill(0xFFFF00);

**this**.graphics.moveTo(100, 100);

*// 꼭지점 (50, 200), (150, 200), (100, 100)인 삼각형 그리기*

**this**.graphics.lineTo(50, 200);

**this**.graphics.lineTo(150, 200);

**this**.graphics.lineTo(100, 100);

}

}

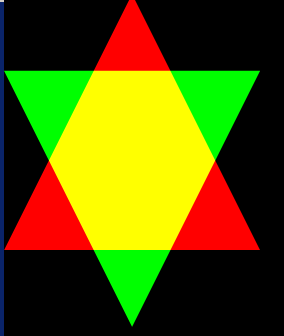
}

별그리기

**package** {

**import** flash.display.Sprite;

**import** flash.display.BlendMode;

 [**SWF**(backgroundColor = 0x000000)]

**public** **class** flexActionScriptTest **extends** Sprite

{

*// 삼각형의 높이 값*

**private** **var** triangleHeight:uint = 100;

*// 삼각형과 역삼각형간 거리*

**private** **var** distance:uint=30;

**public** **function** flexActionScriptTest()

{

*// 정삼각형 그리기*

addTriangle();

*// 역삼각형 그리기*

addRetroTriangle();

}

*// 삼각형 그리기*

**private** **function** addTriangle():**void**

{

*// 블랜딩을 표현하기위해 새로운 Sprite 객체를 만들어 화면에 추가한다.*

**var** triangle:Sprite = **new** Sprite();

triangle.graphics.beginFill(0xFF0000);

*// 꼭지점 1*

triangle.graphics.moveTo(triangleHeight/2, 0);

*// 꼭지점 2*

triangle.graphics.lineTo(triangleHeight, triangleHeight);

*// 꼭지점 3*

triangle.graphics.lineTo(0, triangleHeight);

*// 다시 꼭지점 1으로*

triangle.graphics.lineTo(triangleHeight/2, 0);

*// 그래픽이 겹칠 때 서로 밝게 비치도록 함*

triangle.blendMode = BlendMode.LIGHTEN;

**this**.addChild(triangle);

}

*// 역삼각형 그리기*

**private** **function** addRetroTriangle():**void**

{

*// 블랜딩을 표현하기위해 새로운 Sprite 객체를 만들어 화면에 추가한다.*

**var** triangle2:Sprite = **new** Sprite();

triangle2.graphics.beginFill(0x00FF00);

*// 꼭지점 1*

triangle2.graphics.moveTo(0, distance);

*// 꼭지점 2*

triangle2.graphics.lineTo(triangleHeight, distance);

*// 꼭지점 3*

triangle2.graphics.lineTo(triangleHeight/2, triangleHeight+distance);

*// 다시 꼭지점 1으로*

triangle2.graphics.lineTo(0, distance);

*// 그래픽이 겹칠 때 서로 밝게 비치도록 함*

triangle2.blendMode = BlendMode.LIGHTEN;

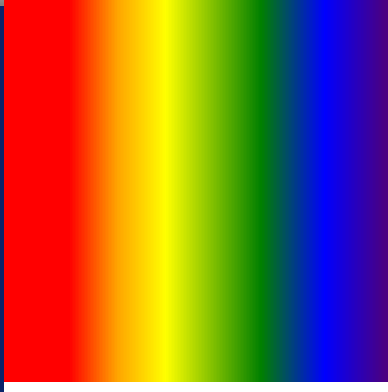
**this**.addChild(triangle2);

}

}

}

무지개

**package** {

**import** flash.display.Sprite;

*//그라데이션을 위한 클래스 import*

**import** flash.display.GradientType;

*//그라데이션 계산을 위한 행렬(Matrix)클래스 import*

**import** flash.geom.Matrix;

**public** **class** AsExam3\_2 **extends** Sprite

{

**public** **function** AsExam3\_2()

{

*//무지개를 그려주는 함수 호출*

drawRainbow();

}

**private** **function** drawRainbow():**void**

{

*//무지개를 그리기 위한 화면 객체 생성*

**var** myRainbow:Sprite = **new** Sprite();

*//행렬 객체 생성*

**var** gradientBoxMatrix:Matrix = **new** Matrix();

*//그라데이션 값 생성*

gradientBoxMatrix.createGradientBox(150, 0, 0, 25, 0);

*//myRainbow화면에 직선 방향의 사각형 그라데이션 그리기 시작*

myRainbow.graphics.beginGradientFill(GradientType.LINEAR,

[0xff0000, 0xffa500, 0xffff00, 0x008000, 0x0000ff, 0x4B0082, 0x800080],

[1, 1, 1, 1, 1, 1, 1],

[0, 32, 64, 128, 170, 210, 255],

gradientBoxMatrix);

*//설정한 그라데이션 정보로 가로 세로 150픽셀의 사각형 그리기*

myRainbow.graphics.drawRect(0, 0, 150, 150);

*//색상 채우기 종료*

myRainbow.graphics.endFill();

*//그라데이션 사각형을 화면에 보이도록 추가함*

**this**.addChild(myRainbow);

}

}

}

빛 삼원색 원3개 RGB

**package** {

*//화면의 겹치는 방법(BlendMode)을 설정하기 클래스 import*

**import** flash.display.BlendMode;

**import** flash.display.Sprite;

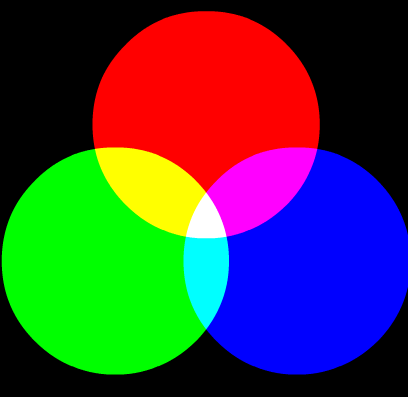
[**SWF**(backgroundColor = 0x000000)]

**public** **class** AsExam3\_3 **extends** Sprite

{

*// drawBackground에서 사용하는 클래스 변수*

**private** **var** sprObj:Sprite = **new** Sprite();

 **public** **function** AsExam3\_3()

{

*//반지름 100의 빨간색 원을 추가*

addCircle1();

*//같은 방식으로 초록색 원을 추가*

addCircle2();

*//같은 방식으로 파란색 원을 추가*

addCircle3();

}

**private** **function** addCircle1():**void**

{

*// addCircle1의 로컬 변수*

**var** sprObj:Sprite = **new** Sprite(); *// 주석처리하면 중첩된 색상 영역이 사라짐*

**this**.addChild(sprObj);

sprObj.graphics.beginFill(0xFF0000);

sprObj.graphics.drawCircle(200, 100, 100);

*//그려진 원의 겹쳐지는 부분이 밝아지도록 설정*

sprObj.blendMode = BlendMode.LIGHTEN;

}

**private** **function** addCircle2():**void**

{

*// addCircle2의 로컬 변수*

**var** sprObj:Sprite = **new** Sprite(); *// 주석처리하면 중첩된 색상 영역이 사라짐*

**this**.addChild(sprObj);

sprObj.graphics.beginFill(0x00FF00);

sprObj.graphics.drawCircle(120, 220, 100);

sprObj.blendMode = BlendMode.LIGHTEN;

}

**private** **function** addCircle3():**void**

{

*// addCircle3의 로컬 변수*

**var** sprObj:Sprite = **new** Sprite(); *// 주석처리하면 중첩된 색상 영역이 사라짐*

**this**.addChild(sprObj);

sprObj.graphics.beginFill(0x0000FF);

sprObj.graphics.drawCircle(280, 220, 100);

sprObj.blendMode = BlendMode.LIGHTEN;

}

}

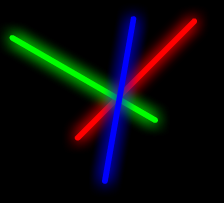
}

제다이 광성검

**package** {

**import** flash.display.Sprite;

**import** flash.filters.BitmapFilterQuality;

 **import** flash.filters.GlowFilter;

[**SWF**(backgroundColor = 0x000000)]

**public** **class** AsExam3\_4 **extends** Sprite

{

*// AsExam3\_4의 생성자 함수*

**public** **function** AsExam3\_4()

{

*// 적색 제다이 광선검 그리기*

drawSword(0, 200, -45, 0xFF0000);

*// 녹색 제다이 광선검 그리기*

drawSword(100, 0, 30, 0x00FF00);

*// 청색 제다이 광선검 그리기*

drawSword(0, -200, 100, 0x0000FF);

}

*// XY 좌표, 기울기, 색깔을 입력 받아 광선검을 그려주는 함수*

**private** **function** drawSword(pX:int, pY:int, r:int, c:int):**void**

{

*// 광선검을 위한 그래픽 객체 인스턴스화*

**var** swordSpr:Sprite = **new** Sprite();

*// 입력받은 색깔로 그리기 시작*

swordSpr.graphics.beginFill(c);

swordSpr.graphics.drawRoundRect(pX, pY, 150, 5, 5,5);

swordSpr.graphics.endFill();

*// 광선검을 일정 각도로 기울여 줌*

swordSpr.rotation = r;

*// 광선검을 화면에 추가*

addChild(swordSpr);

*// 광선검 효과를 주기 위한 필터 객체 생성*

**var** glow:GlowFilter = **new** GlowFilter();

glow.color = c;

glow.alpha = 10;

glow.blurX = 15;

glow.blurY = 15;

glow.quality = BitmapFilterQuality.MEDIUM;

*// 광선검 효과 필터 적용*

swordSpr.filters = [glow];

}

}

}

촛불 candles

**package** {

*//flash.display의 모든 클래스 import*

**import** flash.display.\*;

*//그라데이션 계산을 위한 행렬(Matrix)클래스 import*

**import** flash.geom.Matrix;

**public** **class** AsExam3\_5 **extends** Sprite

{

 *//촛불을 위한 그래픽 객체 선언*

**private** **var** candle:Sprite;

*//행렬 객체 생성*

**private** **var** matrix:Matrix;

*//AsExam3\_4의 생성자 함수*

**public** **function** AsExam3\_5()

{

*//촛불 객체사용을 위한 초기화*

candle=**new** Sprite();

*//행렬 객체 사용을 위한 초기화*

matrix= **new** Matrix();

*//촛불을 그려주는 함수 호출*

drawCandle();

}

*//촛불을 그려주는 함수 선언*

**private** **function** drawCandle():**void**

{

*//촛불의 색: 노란색, 주황색, 어두운남색 배경*

**var** colors:Array = [0xffff00, 0xffa500, 0x171079];

*//각 색상에 대한 투명도*

**var** alphas:Array = [1, 1, 1];

*//색상의 비율*

**var** ratios:Array = [0, 64, 255];

*//행렬객체 생성*

matrix.createGradientBox(25, 150, 0, 60, 0);

*//candle화면에 방사형(RAIAL) 그라데이션 그리기 시작*

candle.graphics.beginGradientFill(GradientType.RADIAL, colors, alphas, ratios, matrix, SpreadMethod.PAD, InterpolationMethod.LINEAR\_RGB, 0);

*//설정한 그라데이션 정보로 가로 세로 150픽셀의 사각형 그리기*

candle.graphics.drawRect(0, 0, 150, 150);

*//그라데이션을 화면에 보이도록 추가함*

addChild(candle);

}

}

}

랜덤 원그리기

**package** {

**import** flash.display.Sprite;

**public** **class** AsExam3\_6 **extends** Sprite

{

*// static으로 선언한 색상값 변수*

**public** **static** **var** randomColor:uint;

*// 인스턴스 변수*

**public** **var** radius:uint;

*// AsExam3\_6의 생성자 함수, 파라미터 값이 없으면 100, 100이 초기값*

**public** **function** AsExam3\_6(px:int=100, py:int=100)

{

*// Math함수를 이용하여 랜덤한 색상값을 만들어 낸다*

randomColor = Math.random() \* 0x1000000;

*// 반지름은 100보다 적은 랜덤값으로 설정*

radius = 100 \* Math.random();

**this**.graphics.beginFill(randomColor);

*// 파라미터로 받은 좌표값에 랜덤한 원을 그려줌*

**this**.graphics.drawCircle(px, py, radius);

}

}

}

클릭한 부분에 랜덤한 원그리기

**package** {

**import** flash.display.Sprite;

*// 마우스 이벤트 처리를 위한 클래스 import*

**import** flash.events.MouseEvent;

*// 그래픽 객체간 겹치는 방법 설정을 위한 클래스 import*

**import** flash.display.BlendMode;

 [**SWF**(backgroundColor=0x000000)]

**public** **class** AsExam3\_7 **extends** Sprite

{

*// 그래픽 객체 선언*

**private** **var** circleSpr:Sprite;

**public** **function** AsExam3\_7() {

displayMain();

addEventHandlers();

}

**private** **function** displayMain():**void**

{

*//그래픽 객체 인스턴스화*

circleSpr = **new** Sprite();

*//화면에 보이도록 그래픽 객체 추가*

**this**.addChild(circleSpr);

circleSpr.graphics.beginFill(0x000000);

circleSpr.graphics.drawRect(0, 0, 500, 500);

circleSpr.blendMode= BlendMode.LIGHTEN;

}

**private** **function** addEventHandlers():**void**

{

*//마우스를 클릭할 때 실행될 함수(mouseClickHandler)지정*

stage.addEventListener(MouseEvent.CLICK, mouseClickHandler);

}

*// 화면을 클릭할 때 마다 AsExam3\_6의 인스턴스를 만들어 화면에 추가*

**private** **function** mouseClickHandler(event:MouseEvent):**void** {

**var** sp:AsExam3\_6 = **new** AsExam3\_6(**this**.mouseX, **this**.mouseY);

**//var** sp:AsExam3\_6 = **new** AsExam3\_6(event.stageX, event.stageY);

sp.blendMode = BlendMode.LIGHTEN;

circleSpr.addChild(sp);

*// 인스턴스 변수의 호출 방법*

**trace**(**"sp.radius: "** + sp.radius);

*// static 변수의 호출 방법*

**trace**(**"AsExam3\_6.randomColor: "** + AsExam3\_6.randomColor);

*// radius는 인스턴스변수이므로 static으로는 호출 안됨*

*// trace("AsExam3\_6.radius: " + AsExam3\_6.radius);*

*// randomColor는 static이므로 인스턴스 변수 방식으로는 호출 안됨*

*// trace("sp.randomColor: " + sp.randomColor);*

}

}

}

단항연산자를 이용한 막대그래프

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

[**SWF**(backgroundColor=0x000000)]

**public** **class** AsExam4\_1 **extends** Sprite

{

*// 막대 그래프의 길이*

**private** **var** barHeight:int = 100;

*// 막대 그래프를 그리기 위한 디스플레이 오브젝트*

**private** **var** sp:Sprite = **new** Sprite();

 **public** **function** AsExam4\_1()

{

displayMain();

drawBar();

addEventHandlers();

}

**private** **function** displayMain():**void**

{

*// 버튼 모드로 설정*

sp.buttonMode = **true**;

*// 화면에 디스플레이 오브젝트 추가*

**this**.addChild(sp);

}

**private** **function** addEventHandlers():**void**

{

*// 디스플레이 오브젝트를 클릭했을 때 실행될 함수 지정*

sp.addEventListener(MouseEvent.CLICK, clickHandler);

}

**private** **function** clickHandler(event:MouseEvent):**void**

{

*// 막대그래프의 길이를 1씩 증가시킴*

barHeight++;

drawBar();

}

*// 흰색 테두리에 랜덤으로 색깔이 바뀌는 사각형을 그려주는 함수*

**private** **function** drawBar():**void**

{

sp.graphics.clear();

sp.graphics.lineStyle(2, 0xCCCCCC);

sp.graphics.drawRect(50, 100, 200, 50);

sp.graphics.beginFill(Math.random()\*0x1000000, 0.5);

sp.graphics.drawRect(50, 100, barHeight, 50);

}

}

}

페이징 알고리즘 사각형 정렬

**package**

{

**import** flash.display.Shape;

**import** flash.display.Sprite;

[**SWF**(backgroundColor=0x000000)]

**public** **class** AsExam4\_2 **extends** Sprite

{

*// 전체 건수*

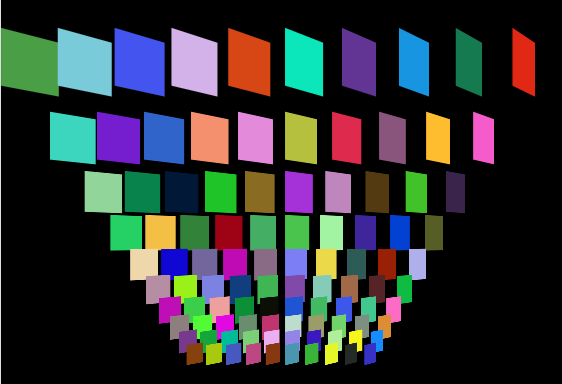
**private** **var** total\_records:uint = 100;

*// 페이지당 보여줄 건수*

**private** **var** num\_of\_row\_per\_page:uint= 10;

*// 페이지 인덱스 갯수*

**private** **var** num\_of\_index\_per\_page:uint=0;

 *// Shape 객체를 담을 배열*

**private** **var** arr:Array = **new** Array();

**public** **function** AsExam4\_2()

{

displayMain();

calculatePage();

layoutPage();

}

*// 화면에 Shape 추가*

**private** **function** displayMain():**void**

{

**for**(**var** idx:int=0; idx<total\_records; idx++)

{

**var** sh:Shape = **new** Shape();

sh.graphics.beginFill(Math.random()\*0x1000000);

sh.graphics.drawRect(0, 0, 50, 50);

**this**.addChild(sh);

arr.push(sh);

}

}

*// Paging 계산*

**private** **function** calculatePage():**void**

{

**if**(total\_records%num\_of\_row\_per\_page>0)

{

num\_of\_index\_per\_page = total\_records/num\_of\_row\_per\_page+1;

}

**else**

{

num\_of\_index\_per\_page = total\_records/num\_of\_row\_per\_page;

}

**trace**(**'num\_of\_index\_per\_page:'**+ num\_of\_index\_per\_page);

}

*// 계산된 페이지 정보에 따라 레이아웃*

**private** **function** layoutPage():**void**

{

**for**(**var** idx:int=0; idx<arr.length; idx++)

{

**var** sh:Shape = arr[idx];

sh.x = (idx % num\_of\_row\_per\_page) \* 50;

sh.y = int(idx / num\_of\_row\_per\_page) \* 50;

sh.z = int(idx / num\_of\_row\_per\_page) \* 100;

sh.rotationY = -45;

}

}

}

}

논리연산

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

[**SWF**(backgroundColor=0xFFFFFF)]

**public** **class** AsExam4\_3 **extends** Sprite

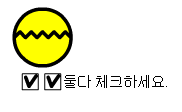
{

*// 체크박스 2개 선언*

**private** **var** cb1:AsExam8\_CheckBox = **new** AsExam8\_CheckBox(**""**, **"cb1"**);

**private** **var** cb2:AsExam8\_CheckBox = **new** AsExam8\_CheckBox(**"둘다 체크하세요."**, **"cb2"**);

**public** **function** AsExam4\_3()

 {

*// 꺼진 전구 그리기*

turnOff();

displayMain();

addEventHandlers();

}

*// 체크박스를 화면에 추가*

**private** **function** displayMain():**void**

{

**this**.addChild(cb1);

**this**.addChild(cb2);

cb1.x = 80;

cb2.x = 100;

cb1.y = cb2.y = 70;

}

*// 체크박스를 체크했을 때 clickHandler 함수가 실행*

**private** **function** addEventHandlers():**void**

{

cb1.addEventListener(**"checkboxChecked"**, clickHandler);

cb2.addEventListener(**"checkboxChecked"**, clickHandler);

}

*// && 연산으로 두 체크박스의 값에 따라 전구를 켜고 끔*

**private** **function** clickHandler(event:MouseEvent):**void**

{

**if**(cb1.checkboxEnabled && cb2.checkboxEnabled){

turnOn();

}

**else**{

turnOff();

}

}

*// 불 켜진 전구 그리기*

**private** **function** turnOn():**void**

{

**this**.graphics.clear();

**this**.graphics.lineStyle(3, 0x000000);

**this**.graphics.beginFill(0xFFFF00);

**this**.graphics.drawCircle(100, 40, 25);

**this**.graphics.endFill();

drawFilament();

}

*// 불 꺼진 전구 그리기*

**private** **function** turnOff():**void**

{

**this**.graphics.clear();

**this**.graphics.lineStyle(3, 0x000000);

**this**.graphics.drawCircle(100, 40, 25);

drawFilament();

}

*// 지그재그로 필라멘트 모양 그리기*

**private** **function** drawFilament():**void**

{

**this**.graphics.moveTo(75, 40);

**this**.graphics.lineTo(80, 35);

**this**.graphics.lineTo(85, 40);

**this**.graphics.lineTo(90, 35);

**this**.graphics.lineTo(95, 40);

**this**.graphics.lineTo(100, 35);

**this**.graphics.lineTo(105, 40);

**this**.graphics.lineTo(110, 35);

**this**.graphics.lineTo(115, 40);

**this**.graphics.lineTo(120, 35);

**this**.graphics.lineTo(125, 40);

}

}

}

논리연산에 사용된 체크박스

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

**import** flash.text.TextField;

*// 마우스 클릭시 checkboxChecked를 발생한다는 선언*

[**Event**(name=**"checkboxChecked"**, type=**"flash.events.MouseEvent"**)]

**public** **class** AsExam8\_CheckBox **extends** Sprite

{

*// 체크 박스의 상태값 저장*

**public** **var** checkboxEnabled:Boolean;

*// 체크박스의 라벨 텍스트 값*

**public** **var** checkboxText:String;

*// 체크박스 라벨 텍스트를 표시하는 텍스트필드*

**private** **var** checkboxTextField:TextField=**new** TextField();

*// 체크박스를 그리기 위한 Sprite*

**private** **var** checkBoxArea:Sprite = **new** Sprite();

*// 체크박스의 클릭 영역을 만들어주기 위한 Sprite*

**private** **var** checkBoxHitArea:Sprite = **new** Sprite();

*// 체크박스텍스트와 인스턴스 이름을 입력 받는 생성자 함수*

**public** **function** AsExam8\_CheckBox(checkboxText:String, insName:String)

{

**this**.name = insName;

**this**.checkboxText = checkboxText;

initCheckBox();

}

**private** **function** initCheckBox():**void**

{

*// 체크되지 않은 상태의 체크박스를 그림*

drawUnchecked();

*// 체크박스의 마우스 클릭 영역을 그림*

drawHitarea();

*// 체크박스 라벨을 위한 텍스트 필드를 추가*

drawTextField();

*// 체크박스의 이벤트 핸들러 함수 추가*

checkBoxArea.addEventListener(MouseEvent.CLICK, clickHandler);

**this**.addChild(checkBoxArea);

}

*// 체크박스 라벨 변경을 위한 함수로 외부에서 사용가능하도록 하기 위해 punblic으로 선언*

**public** **function** setCheckBoxText(txt:String):**void**

{

checkboxTextField.text = txt;

}

*// 텍스트필드를 화면에 추가함*

**private** **function** drawTextField():**void**

{

checkboxTextField.x = 16;

checkboxTextField.y = 0;

checkboxTextField.text = checkboxText;

checkboxTextField.autoSize = **"left"**;

**this**.addChild(checkboxTextField);

}

*// checkBoxHitArea를 화면에 그린 다음 checkBoxHitArea를 checkBoxArea의 히트영역으로 설정*

**private** **function** drawHitarea():**void**

{

checkBoxHitArea.graphics.beginFill(0xFFFF00, 0);

checkBoxHitArea.graphics.drawRect(2, 2, 14, 14);

checkBoxHitArea.graphics.endFill();

checkBoxArea.buttonMode = **true**;

checkBoxArea.hitArea = checkBoxHitArea;

checkBoxHitArea.mouseEnabled = **false**;

**this**.addChild(checkBoxHitArea);

}

*// V자 표시로 체크된 상태를 그림*

**private** **function** drawChecked():**void**

{

checkBoxArea.graphics.clear();

checkBoxArea.graphics.lineStyle(1, 0x000000);

checkBoxArea.graphics.drawRect(2, 2, 14, 14);

checkBoxArea.graphics.lineStyle(3, 0x000000);

*// V자 라인을 그림*

checkBoxArea.graphics.moveTo(5, 5);

checkBoxArea.graphics.lineTo(8, 13);

checkBoxArea.graphics.lineTo(14, 4);

checkBoxArea.graphics.endFill();

}

*// V자 표시 없이 체크되지 않은 상태를 그림*

**private** **function** drawUnchecked():**void**

{

checkBoxArea.graphics.clear();

checkBoxArea.graphics.lineStyle(1, 0x000000);

checkBoxArea.graphics.drawRect(2, 2, 14, 14);

checkBoxArea.graphics.endFill();

}

*// 체크가 되면 화면 상태를 바꾸고 checkboxChecked 이벤트 발생*

**private** **function** clickHandler(event:MouseEvent):**void**

{

**if**(!checkboxEnabled)

{

drawChecked();

checkboxEnabled = **true**;

}

**else**

{

drawUnchecked();

checkboxEnabled = **false**;

}

event.currentTarget.dispatchEvent(**new** MouseEvent(**"checkboxChecked"**));

}

}

}

숫자맞추는 게임 textfield

**package**

{

**import** flash.display.Sprite;

**import** flash.events.KeyboardEvent;

**import** flash.text.TextField;

[**SWF**(backgroundColor=0xFFFFFF)]

**public** **class** AsExam4\_4 **extends** Sprite

{

*// 정답이될 숫자변수*

**private** **var** quizNumber:int;

*// 답변 시도 횟수*

**private** **var** guessCount:int;

*// 숫자를 입력받는 텍스트필드*

**private** **var** ansTF:TextField = **new** TextField();

**private** **var** label:TextField = **new** TextField();

**public** **function** AsExam4\_4()

{

displayMain();

addEventHandlers();

}

*// 안내문과 숫자를 입력받을 텍스트필드를 화면에 추가한다.*

**private** **function** displayMain():**void**

{

quizNumber = Math.random() \* 100;

**this**.addChild(label);

label.text = **'0~99까지의 숫자를 입력한 후 엔터키를 치시오.'**;

label.autoSize = **"left"**;

label.x = 40;

label.y = 10;

**this**.addChild(ansTF);

ansTF.type = **"input"**;

ansTF.border = **true**;

ansTF.background = **true**;

 ansTF.width = 30;

ansTF.height = 20;

*// 글자수 입력은 3자 이내*

ansTF.maxChars = 3;

ansTF.y = 10;

*// 숫자만 입력되도록 제한*

ansTF.restrict = **"0-9"**;

}

*// 숫자 입력창에 키보드 이벤트 핸들러 함수를 추가함*

**private** **function** addEventHandlers():**void**

{

ansTF.addEventListener(KeyboardEvent.KEY\_DOWN, keyDownHandler);

}

*// 엔터키 입력이 들어오면 숫자값을 비교해서 메시지를 보낸다.*

**private** **function** keyDownHandler(event:KeyboardEvent):**void**

{

**if**(event.keyCode == 13)

{

guessCount++;

**trace**(ansTF.text, quizNumber);

label.text = **"입력한 숫자는 "** + ansTF.text + **"이며"**;

**var** num:int = int(ansTF.text);

**if**(num > quizNumber)

{

label.text = **"정답보다 크구료. ["** + guessCount + **"]번째 도전"** ;

}

**else** **if**(num < quizNumber)

{

label.text = **"정답보다 적구료. ["** + guessCount + **"]번째 도전"** ;

}

**else** **if**(num == quizNumber)

{

label.text = **"["** + guessCount + **"]번째 도전만에"** + quizNumber + **"를 맞췄소."**;

*// 정답을 맞추면 숫자를 다시 랜덤으로 생성한다.*

quizNumber = Math.random() \* 100;

guessCount = 0;

}

}

}

}

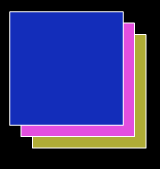
}

할당 사각형 겹치기

**package**

{

**import** flash.display.Shape;

 **import** flash.display.Sprite;

[**SWF**(backgroundColor=0x000000)]

**public** **class** AsExam4\_5 **extends** Sprite

{

**public** **function** AsExam4\_5()

{

displayMain();

}

**private** **function** displayMain():**void**

{

**var** sh1:AsExam4\_5\_1 = **new** AsExam4\_5\_1();

**var** sh2:AsExam4\_5\_1 = **new** AsExam4\_5\_1();

**var** sh3:AsExam4\_5\_1 = **new** AsExam4\_5\_1();

**this**.addChild(sh1);

**this**.addChild(sh2);

**this**.addChild(sh3);

sh1.x += (sh2.x += (sh3.x += 10));

**trace**(sh1.x, sh2.x, sh3.x); *// 40, 30, 20*

sh1.y += (sh2.y += (sh3.y += 10));

**trace**(sh1.y, sh2.y, sh3.y); *// 40, 30, 20*

}

}

}

**import** flash.display.Shape;

**class** AsExam4\_5\_1 **extends** Shape

{

**public** **function** AsExam4\_5\_1()

{

**this**.graphics.lineStyle(1, 0xFFFFFF);

**this**.graphics.beginFill(Math.random()\*0x1000000);

**this**.graphics.drawRect(0, 0, 100, 100);

**this**.x = **this**.y = 10;

}

}비트연산 RGB 표현

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

**import** flash.text.TextField;

[**SWF**(backgroundColor=0xFFFFFF)]

**public** **class** AsExam4\_6 **extends** Sprite

{

*// 각 색상별 값을 변화시키기 위한 컨트롤*

**private** **var** nsRed:AsExam8\_NumericStepper = **new** AsExam8\_NumericStepper(0, 255, 10);

**private** **var** nsGreen:AsExam8\_NumericStepper = **new** AsExam8\_NumericStepper(0, 255, 10);

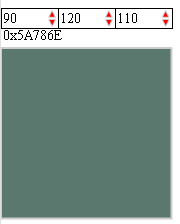
**private** **var** nsBlue:AsExam8\_NumericStepper = **new** AsExam8\_NumericStepper(0, 255, 10);

*// RGB값을 보여주는 텍스트필드*

**private** **var** tf:TextField = **new** TextField();

**public** **function** AsExam4\_6()

{

 displayMain();

drawBox(0);

addEventHandlers();

}

*// 컨트롤과 텍스트필드를 화면에 배치*

**private** **function** displayMain():**void**

{

**this**.addChild(nsRed);

**this**.addChild(nsGreen);

**this**.addChild(nsBlue);

nsGreen.x = 50;

nsBlue.x = 100;

nsRed.y = nsGreen.y = nsBlue.y = 10;

**this**.addChild(tf);

tf.y = 25;

}

**private** **function** addEventHandlers():**void**

{

nsRed.addEventListener(**"numberChanged"**, numberChangedHandler);

nsGreen.addEventListener(**"numberChanged"**, numberChangedHandler);

nsBlue.addEventListener(**"numberChanged"**, numberChangedHandler);

}

*// 컨트롤의 값이 바뀌었을 때 실행될 함수*

**private** **function** numberChangedHandler(event:MouseEvent):**void**

{

*// 16진수 값 계산*

**var** rVal:int = nsRed.selectedNumber << 16;

**var** gVal:int = nsGreen.selectedNumber << 8;

**var** bVal:int = nsBlue.selectedNumber;

**var** rgb:int = rVal | gVal | bVal;

**trace**(rgb.toString(16).toUpperCase());

drawBox(rgb);

tf.text = **"0x"** + nsRed.selectedNumber.toString(16).toUpperCase()

+ nsGreen.selectedNumber.toString(16).toUpperCase() + nsBlue.selectedNumber.toString(16).toUpperCase();

}

*// 지정된 색상으로 박스를 그림*

**private** **function** drawBox(rgb:int):**void**

{

**this**.graphics.clear();

**this**.graphics.lineStyle(2, 0xCCCCCC);

**this**.graphics.beginFill(rgb);

**this**.graphics.drawRect(0, 45, 150, 150);

}

}

}

RGB 표현 쪽에서쓴 조정 박스

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

**import** flash.text.TextField;

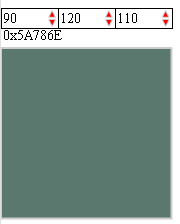
*// 숫자 입력기를 선택하면 numberChanged이벤트가 발생하도록 함*

[**Event**(name=**"numberChanged"**, type=**"flash.events.MouseEvent"**)]

**public** **class** AsExam8\_NumericStepper **extends** Sprite

{

*// 숫자값을 보여주기 위한 텍스트필드*

 **private** **var** tf:TextField = **new** TextField();

*// 현재 선택한 값*

**public** **var** selectedNumber:Number=0;

*// 숫자값 변화 간격*

**public** **var** interval:Number=1;

*// 시작값*

**public** **var** fromNum:Number=0;

*// 최고값*

**public** **var** toNum:Number=0;

*// 라벨값*

**private** **var** nsLabel:String;

*// 라벨 텍스트필드*

**private** **var** tfLabel:TextField = **new** TextField();

*// 상하 버튼의 누름 여부*

**public** **var** upDownYN:int = 1;

*// 시작값, 최고값, 간격, 라벨, 인스턴스명, 선택값을 입력 받는 생성자 함수*

**public** **function** AsExam8\_NumericStepper(fromNum:Number=0, toNum:Number=0, interval:Number=1, nsLabel:String=**''**, insName:String=**''**, selectedNum:Number=0)

{

**this**.fromNum = fromNum;

**this**.toNum = toNum;

**this**.interval = interval;

**this**.nsLabel = nsLabel;

**this**.name = insName;

**this**.selectedNumber = selectedNum;

drawNumericStepper();

}

*// 숫자 입력기를 화면에 그린다.*

**private** **function** drawNumericStepper():**void**

{

*// 텍스트필드를 화면에 추가한다.*

**this**.addChild(tf);

tf.background = **true**;

tf.border = **true**;

tf.height = 18;

tf.width = 50;

tf.text = **""** + selectedNumber;

*// 라벨을 화면에 추가한다.*

**this**.addChild(tfLabel);

tfLabel.x = 50;

tfLabel.text = **this**.nsLabel;

*// 위쪽 버튼을 추가한다.*

**var** upBtn:AsExam8\_TriangleButton = **new** AsExam8\_TriangleButton(6);

*// 버튼 방향이 위로 가도록 180도 회전한다.*

upBtn.rotation = 180;

upBtn.x = 48;

upBtn.y = 8;

*// 아래쪽 버튼을 추가한다.*

**var** downBtn:AsExam8\_TriangleButton = **new** AsExam8\_TriangleButton(6);

**this**.addChild(upBtn);

**this**.addChild(downBtn);

downBtn.y = 10;

downBtn.x = 42;

*// 버튼에 클릭이벤트 핸들러 함수를 등록한다.*

upBtn.addEventListener(MouseEvent.CLICK, upBtnClickHandler);

downBtn.addEventListener(MouseEvent.CLICK, downBtnClickHandler);

}

*// 업버튼을 누르면 선택값을 증가시킨다.*

**private** **function** upBtnClickHandler(event:MouseEvent):**void**

{

selectedNumber+= interval;

*// 최고값을 넘지 않도록*

**if**(selectedNumber > toNum) selectedNumber = toNum;

selectedNumber = Math.round(selectedNumber\*10)/10;

tf.text = **""** + selectedNumber;

upDownYN = 1;

*// numberChanged 이벤트를 디스패치한다.*

event.currentTarget.dispatchEvent(**new** MouseEvent(**"numberChanged"**));

}

*// 다운버튼을 누르면 선택값을 감소시킨다.*

**private** **function** downBtnClickHandler(event:MouseEvent):**void**

{

selectedNumber-= interval;

*// 최저값을 넘지 않도록*

**if**(selectedNumber < fromNum) selectedNumber = fromNum;

selectedNumber = Math.round(selectedNumber\*10)/10;

tf.text = **""** + selectedNumber;

upDownYN = -1;

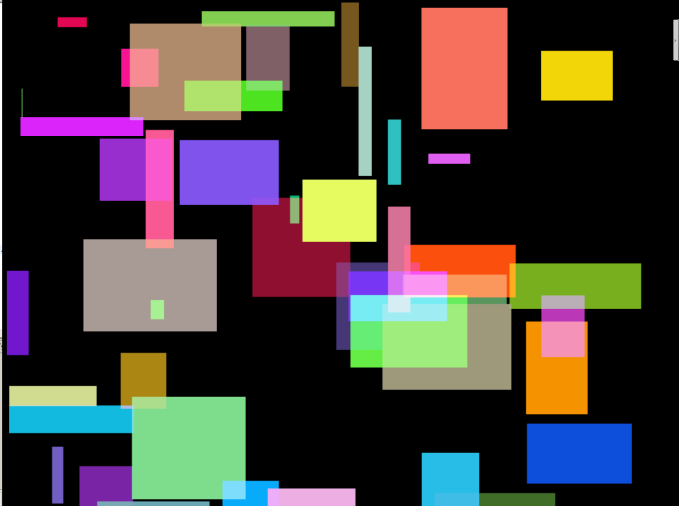
*// numberChanged 이벤트를 디스패치한다.*

event.currentTarget.dispatchEvent(**new** MouseEvent(**"numberChanged"**));

}

}

}랜덤 사각형 전체 뿌리기

**package** {

**import** flash.display.BlendMode;

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

[**SWF**(backgroundColor=0x000000)]

**public** **class** AsExam4\_7 **extends** Sprite

{

*// 화면에 추가되는 Sprite 객체를 담을 배열*

**private** **var** arr:Array = **new** Array();

**public** **function** AsExam4\_7()

{

displayMain();

addEventHandlers();

}

*// 화면을 클릭했을 때 clickHandler 함수가 실행되도록 함*

**private** **function** addEventHandlers():**void**

{

stage.addEventListener(MouseEvent.CLICK, clickHandler);

}

*// 화면에 랜덤한 곳에 랜덤한 크기로 50개의 사각형을 그린다음 배열에 추가*

**private** **function** displayMain():**void**

{

**for**(**var** i:int=0; i<50; i++)

{

**var** sp:Sprite = **new** Sprite();

sp.blendMode = BlendMode.LIGHTEN;

sp.graphics.beginFill(Math.random()\*0x1000000);

sp.graphics.drawRect(Math.random()\*400, Math.random()\*400, Math.random()\*100, Math.random()\*100);

**this**.addChild(sp);

arr.push(sp);

}

}

*// 화면을 클릭할 때 마다 배열에 있는 Sprite를 캐스팅하여 다시 화면을 그린다.*

**private** **function** clickHandler(event:MouseEvent):**void**

{

**for**(**var** i:int=0; i<arr.length; i++)

{

**var** sp:Sprite = arr[i] **as** Sprite;

sp.graphics.clear();

sp.graphics.beginFill(Math.random()\*0x1000000);

sp.graphics.drawRect(Math.random()\*400, Math.random()\*400, Math.random()\*100, Math.random()\*100);

}

}

}

}

문자열 랜덤 반복 및 화면 뿌리기

**package**

{

**import** flash.display.Sprite;

**import** flash.text.TextField;

**import** flash.text.TextFormat;

[**SWF**(width=800, height=800, backgroundColor=0xffffff)]

**public** **class** AsExam4\_8 **extends** Sprite

{

*// 텍스트필드들을 담을 Sprite 객체*

**private** **var** sp:Sprite = **new** Sprite();

 **public** **function** AsExam4\_8()

{

displayMain();

addChars();

}

*// Sprite를 화면에 추가*

**private** **function** displayMain():**void**

{

sp.x = 500;

**this**.addChild(sp);

}

*// 텍스트필드 200개를 Sprite에 추가*

**private** **function** addChars():**void**

{

**for**(**var** i:int = 0; i < 200; i++)

{

**var** tf:TextField = **new** TextField();

tf.defaultTextFormat = **new** TextFormat(**"Arial"**, 35, Math.random()\*0x1000000);

*// 한글 유니코드 범위에 있는 값을 무작위로 추출*

tf.text = String.fromCharCode(Math.random() \* 5000 + 45000);

*//tf.text = String.fromCharCode(65 + Math.random() \* 25); // 알파벳*

tf.x = Math.random() \* 500 - 250;

tf.y = Math.random() \* 500 - 250;

*// 입체감을 주기 위해 z 좌표 추가*

tf.z = Math.random() \* 1000;

sp.addChild(tf);

}

}

}

}

정규식표현식과 replace함수 이용 문자열 검색

**package** {

**import** flash.display.Sprite;

**import** flash.events.TextEvent;

**import** flash.text.TextField;

[**SWF**(width=800,height=600,backgroundColor=0xFFFFFF)]

**public** **class** AsExam4\_9 **extends** Sprite

{

*// 검색어를 입력받을 TextField*

**private** **var** textInputTxt:TextField;

*// 컨텐츠를 표시할 TextField*

**private** **var** textArea:TextField;

**private** **var** msg:String = **' 코드의 로직은 머리에서 코드의 입력은 타이핑에서 완성된다.'** +

**' 코드의 결과는 항상 정직하다. '**+

**' 사용자에게 코드는 트릭이다.'**+

**' 사소한 속성이나 함수의 존재 이유는 분명 어딘가에 쓰이기 때문이다.'**+

**' 코딩의 지름길은 단축키다.'**+

**' 고기를 잡는 법을 배워야 하듯이 코딩 결과보다는 코딩하는 과정을 배워라.'**+

**' 조각은 불필요한 부분을 제거하는 것. 코딩최적화는 불필요한 로직을 제거하는 것.'**+

**' 매트릭스의 네오의 눈으로 실세계를 코드화해서 보는 눈을 만들라.'**+

**' 물체의 회전후 모양을 예상하는 것은 3D의 기본이다.'**;

**public** **function** AsExam4\_9()

{

displayMain();

addEventHandlers();

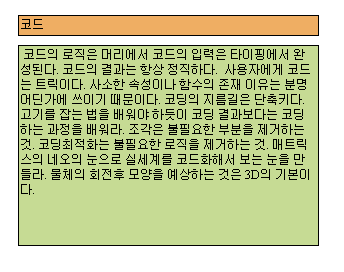
}

**private** **function** displayMain():**void**

{

textInputTxt = **new** TextField();

textInputTxt.type = **"input"**;

 textInputTxt.background = **true**;

textInputTxt.border = **true**;

textInputTxt.height = 20;

textInputTxt.width = 300;

textInputTxt.backgroundColor = 0xEFAE63;

textInputTxt.x = textInputTxt.y = 30;

**this**.addChild(textInputTxt);

textInputTxt.text = **"검색할 키워드를 입력하세요."**;

textArea = **new** TextField();

textArea.height = 200;

textArea.width = 300;

textArea.multiline = **true**;

textArea.x = 30;

textArea.y = 60;

textArea.type = **"input"**;

textArea.backgroundColor = 0xC6DB94;

textArea.background = **true**;

textArea.border = **true**;

textArea.wordWrap = **true**;

textArea.htmlText = msg;

**this**.addChild(textArea);

}

*// 검색어 창에 텍스트를 입력할 때 실행할 함수 지정*

**private** **function** addEventHandlers():**void**

{

textInputTxt.addEventListener(TextEvent.TEXT\_INPUT, textInputHandler);

}

*// 텍스트가 입력되면 검색어를 마킹하는 함수 호출*

**private** **function** textInputHandler(event:TextEvent):**void**

{

**var** key:String = textInputTxt.text + event.text;

markString(key);

}

*// 정규표현식을 이용하여 모든 일치하는 문자열을 검색한 다음 html로 마킹함*

**private** **function** markString(key:String):**void**

{

**var** reg:RegExp = **new** RegExp(key, **"gi"**);

**var** res:String = msg.replace(reg, **"<b><u><font color='#FF0000'>"** + key + **"</font></u></b>"**);

textArea.htmlText = res;

}

}

}

클릭스 이미지 바뀌는 가위바위보

**package** {

**import** flash.display.Loader;

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

**import** flash.net.URLRequest;

[**SWF**(backgroundColor=0x000000)]

**public** **class** AsExam5\_1 **extends** Sprite

 {

*// 이미지를 로드하는 클래스*

**private** **var** ld:Loader = **new** Loader();

**public** **function** AsExam5\_1()

{

displayMain();

addEventHandlers();

}

**private** **function** displayMain():**void**

{

**this**.addChild(ld);

ld.x = ld.y = 100;

*// assets/2.jpg를 로드한다.*

ld.load(**new** URLRequest(**"assets/2.jpg"**));

}

*// 화면을 클릭하면 clickHandler 함수가 실행된다.*

**private** **function** addEventHandlers():**void**

{

stage.addEventListener(MouseEvent.CLICK, clickHanlder);

}

**private** **function** clickHanlder(event:MouseEvent):**void**

{

*// 1 ~ 3 까지의 숫자를 랜덤으로 생성*

**var** hand:uint = Math.random() \* 3 + 1;

**if**(hand == 1)

{

ld.load(**new** URLRequest(**"assets/"** + hand + **".jpg"**));*//1이면 가위*

}

**else** **if**(hand == 2)

{

ld.load(**new** URLRequest(**"assets/"** + hand + **".jpg"**));*//2면 바위*

}

**else** **if**(hand == 3)

{

ld.load(**new** URLRequest(**"assets/"** + hand + **".jpg"**));*//3이면 보*

}

**else**

{

**trace**(**'error'**);

}

}

}

}

주사위 랜덤함수 및 흐름제어

**package** {

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

*// switch 문을 이용한 주사위 그리기*

**public** **class** AsExam5\_2 **extends** Sprite

{

*// 주사위 면을 위한 Sprite 클래스*

**private** **var** diceSpr:Sprite = **new** Sprite();

**public** **function** AsExam5\_2()

{

displayMain();

addEventHandlers();

}

**private** **function** displayMain():**void**

{

**this**.addChild(diceSpr);

*// 주사위 모양을 그려주는 함수 호출*

drawDice();

}

**private** **function** addEventHandlers():**void**

{

*// 마우스를 클릭하면 주사위 모양을 그려주는 함수가 호출되도록 함*

diceSpr.addEventListener(MouseEvent.CLICK, drawDice);

}

*// 입력 파라미터가 없으면 null값으로 입력되도록 파라미터를 세팅*

**private** **function** drawDice(event:MouseEvent=**null**):**void**

{

*// 1 ~ 6 까지의 난수 발생*

**var** diceNumber:uint = Math.random()\*6 + 1;

*// 이전의 주사위 그림을 지워줌*

diceSpr.graphics.clear();

diceSpr.graphics.lineStyle();

*// 하얀 면을 그려줌*

diceSpr.graphics.beginFill(0xFFFFFF);

diceSpr.graphics.drawRect(25, 25, 70, 70);

diceSpr.graphics.endFill();

diceSpr.graphics.beginFill(0x0000FF);

*// 주사위 값에 따라 점을 그려 줌*

**switch**(diceNumber) {

**case** 1:

diceSpr.graphics.drawCircle(60, 60, 5);

**break**;

**case** 2:

diceSpr.graphics.drawCircle(40, 40, 5);

diceSpr.graphics.drawCircle(80, 80, 5);

**break**;

**case** 3:

diceSpr.graphics.drawCircle(40, 40, 5);

diceSpr.graphics.drawCircle(60, 60, 5);

diceSpr.graphics.drawCircle(80, 80, 5);

**break**;

**case** 4:

diceSpr.graphics.drawCircle(40, 40, 5);

diceSpr.graphics.drawCircle(40, 80, 5);

diceSpr.graphics.drawCircle(80, 40, 5);

diceSpr.graphics.drawCircle(80, 80, 5);

**break**;

**case** 5:

diceSpr.graphics.drawCircle(40, 40, 5);

diceSpr.graphics.drawCircle(80, 40, 5);

diceSpr.graphics.drawCircle(60, 60, 5);

diceSpr.graphics.drawCircle(40, 80, 5);

diceSpr.graphics.drawCircle(80, 80, 5);

**break**;

**case** 6:

diceSpr.graphics.drawCircle(40, 40, 5);

diceSpr.graphics.drawCircle(40, 60, 5);

diceSpr.graphics.drawCircle(40, 80, 5);

diceSpr.graphics.drawCircle(80, 40, 5);

diceSpr.graphics.drawCircle(80, 60, 5);

diceSpr.graphics.drawCircle(80, 80, 5);

**break**;

}

}

}

}

클릭과 삼단항연산자 이용 하이로우 게임 카드게임

**package** {

**import** flash.display.Sprite;

**import** flash.events.\*;

**import** flash.text.TextField;

*//?: 문을 이용한 하이로우 게임*

**public** **class** AsExam5\_3 **extends** Sprite

{

**private** **var** tfMain:TextField = **new** TextField();*//결과 메시지 텍스트필드*

**private** **var** cardHighNum:uint;*// High 카드 숫자*

**private** **var** cardLowNum:uint; *// Low 카드 숫자*

*// 카드를 그려주는 내부 클래스를 생성*

**private** **var** cardHigh:AsExam5\_3\_Card = **new** AsExam5\_3\_Card(5, 25, 100, 170, **"HIGH"**, 0xEFAE63);

**private** **var** cardLow:AsExam5\_3\_Card = **new** AsExam5\_3\_Card(120, 25, 100, 170, **"LOW"**, 0xC6DB94);

**public** **function** AsExam5\_3()

{

displayMain();

addEventHandlers();

}

*// 카드를 2장 추가하고 메시지를 보여주는 텍스트 필드를 추가한다.*

**private** **function** displayMain():**void**

{

**this**.addChild(cardHigh);

**this**.addChild(cardLow);

**this**.addChild(tfMain);

tfMain.x = 5;

tfMain.y = 200;

tfMain.width = 300;

tfMain.text = **"왼쪽은 High, 오른쪽은 Low 클릭하세요."**;

}

*// 카드를 클릭했을 때 실행될 함수를 지정한다.*

**private** **function** addEventHandlers():**void**

{

cardHigh.addEventListener(MouseEvent.CLICK, cardHighHandler);

cardLow.addEventListener(MouseEvent.CLICK, cardLowHandler);

}

*// High 패를 선택했을 때 실행*

**private** **function** cardHighHandler(event:MouseEvent):**void**

{

shuffleCards();

winCheck(1, cardHighNum, cardLowNum);

}

*// Low 패를 선택했을 때 실행*

**private** **function** cardLowHandler(event:MouseEvent):**void**

{

shuffleCards();

winCheck(2, cardLowNum, cardHighNum);

}

*// 카드 패 번호를 섞어서 각 카드 1 ~ 10까지 번호를 배정*

**private** **function** shuffleCards():**void**

{

cardHighNum = Math.random() \* 10 + 1;

cardLowNum = Math.random() \* 10 + 1;

cardHigh.setMsg(**"High : ["** + cardHighNum + **"]"**);

cardLow.setMsg(**"Low : ["** + cardLowNum + **"]"**);

}

*// High냐 Low냐에 따라서 카드패 번호를 ? 연산자로 비교하여 결과 메시지 생성*

**private** **function** winCheck(card:uint, yourNum:uint, comNum:uint):**void**

{

**var** resMsg:String; *//결과 메시지*

**switch**(card) {

**case** 1: *// high*

resMsg = yourNum > comNum ? **"You win! "** : **"You lose"**;

**break**;

**case** 2: *// low*

resMsg = yourNum < comNum ? **"You win! "** : **"You lose"**;

**break**;

}

tfMain.text = resMsg;

}

}

}

**import** flash.display.Sprite;

**import** flash.text.TextField;

*// 카드를 그려주는 내부 클래스*

**class** AsExam5\_3\_Card **extends** Sprite

{

**private** **var** tf:TextField = **new** TextField();

*// 카드의 좌표, 가로세로 크기, 메시지, 카드 색상을 입력받음*

**public** **function** AsExam5\_3\_Card(px:Number, py:Number, w:Number, h:Number, label:String, c:uint)

{

 *// 카드의 모양(사각형)을 그려줌*

**this**.graphics.beginFill(0xFFFFFF);

**this**.graphics.drawRect(px, py, w, h);

**this**.graphics.beginFill(c);

**this**.graphics.drawRect(px+2, py+2, w-4, h-4);

**this**.graphics.endFill();

tf.text = label;

tf.x = px + 30;

tf.y = **this**.height/2;

**this**.addChild(tf);

}

*// 카드에 표기될 메시지를 설정*

**public** **function** setMsg(msg:String):**void**

{

tf.text = msg;

}

}

플래그를 이용한 그리드색상 on/off

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

*// Boolean 플래그를 이용한 그리드 컬러 On/Off*

**public** **class** AsExam5\_7 **extends** Sprite

{

*// 그리드 셀의 On/Off를 위한 Boolean 플래그 변수*

**private** **var** myFlag:Boolean = **true**;

*// 그리드 셀 Sprite객체를 담는 배열*

**private** **var** arrGridSpr:Array;

**public** **function** AsExam5\_7()

{

displayMain();

drawBoxes();

addEventHandlers();

}

*// 배열에 Sprite를 가로 세로 크기 (20\*20=400)만큼 생성하여 담아둠*

**private** **function** displayMain():**void**

{

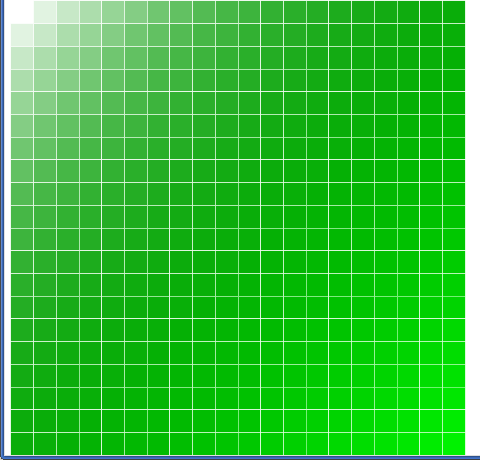
arrGridSpr = **new** Array();

**for**(**var** j:Number=0; j<400; j++)

{

**var** sp:Sprite = **new** Sprite();

arrGridSpr.push(sp);

 **this**.addChild(sp);

}

}

*// 가로 20, 세로 20개의 그리드 셀을 이중 반복문으로 그려줌*

**private** **function** drawBoxes():**void**

{

*// 그리드 픽셀의 크기*

**var** recSize:Number = 20;

*// 각 그리드 셀의 좌표 계산을 위한 변수*

**var** posX:Number = 0;

**var** posY:Number = 0;

**var** pixelColor:uint=0x000000;

*// Sprite 카운트 변수*

**var** cnt:int= 0;

**for**(**var** i:Number=0; i<recSize; i++)

{

**for**(**var** j:Number=0; j<recSize; j++)

{

**var** sp:Sprite = arrGridSpr[cnt];

cnt++;

sp.graphics.lineStyle(.1, 0xFFFFFF, 0.4);

**if**(myFlag) {

pixelColor = 0x00FF00;

} **else** {

pixelColor = 0x000000;

}

sp.graphics.beginFill(pixelColor, (i\*0.025 + j\*0.025));

sp.graphics.drawRect(5, 5, recSize, recSize);

sp.x = posX;

sp.y = posY;

posY += recSize;

}

posX += recSize;

posY = 0;

}

}

**private** **function** addEventHandlers():**void**

{

stage.addEventListener(MouseEvent.CLICK, clickHandler);

}

*// 그리드를 클릭할 때 마다 플래그가 On/Off되면서 화면을 다시 그려줌*

**private** **function** clickHandler(event:MouseEvent):**void**

{

myFlag = !myFlag;

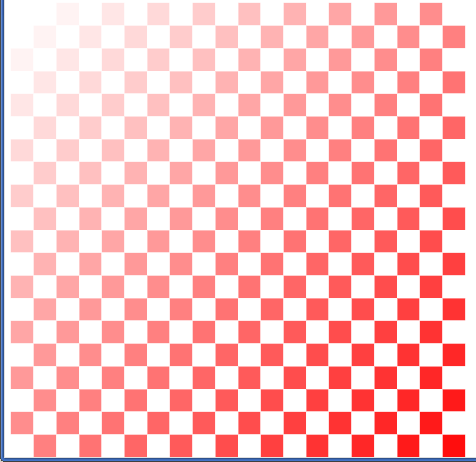
drawBoxes();

}

}

}

체스판 만들기

**package** {

**import** flash.display.Sprite;

**import** flash.events.\*;

*//홀짝수 로직을 이용한 체스판 만들기*

**public** **class** AsExam5\_8 **extends** Sprite

{

*// 그리드 셀의 On/Off를 위한 Boolean 플래그 변수*

**private** **var** myFlag:Boolean = **true**;

*// 그리드 셀 Sprite객체를 담는 배열*

**private** **var** arrGridSpr:Array;

**public** **function** AsExam5\_8()

{

displayMain();

drawBoxes();

}

*// 배열에 Sprite를 가로 세로 크기 (20\*20=400)만큼 생성하여 담아둠*

**private** **function** displayMain():**void**

{

arrGridSpr = **new** Array();

**for**(**var** j:Number=0; j<400; j++)

{

**var** sp:Sprite = **new** Sprite();

arrGridSpr.push(sp);

**this**.addChild(sp);

}

}

*// 홀수 번째와 짝수 번째를 번갈아 가면서 서로 다른 색깔로 사각형을 그려줌*

**private** **function** drawBoxes():**void**

{

*// 그리드 픽셀의 크기*

**var** recSize:Number = 20;

*// 각 그리드 셀의 좌표 계산을 위한 변수*

**var** posX:Number = 0;

**var** posY:Number = 0;

**var** pixelColor:uint=0x000000;

*// Sprite 카운트 변수*

**var** cnt:int= 0;

**for**(**var** i:Number=0; i<recSize; i++)

{

**for**(**var** j:Number=0; j<recSize; j++)

{

**var** sp:Sprite = arrGridSpr [cnt];

cnt++;

sp.graphics.lineStyle(.1, 0xFFFFFF, 0.4);

**if**(i%2 != 0) { *// 가로행에서 홀수*

**if**(j%2!=0) {*// 세로열에서 홀수*

pixelColor = 0xFF0000;

} **else** {

pixelColor = 0xFFFFFF;

}

} **else** {

**if**(j%2==0) {*// 세로열에서 짝수*

pixelColor = 0xFF0000;

} **else** {

pixelColor = 0xFFFFFF;

}

}

sp.graphics.beginFill(pixelColor, (i\*0.025 + j\*0.025));

sp.graphics.drawRect(5, 5, recSize, recSize);

sp.x = posX;

sp.y = posY;

posY += recSize;

}

posX += recSize;

posY = 0;

}

}

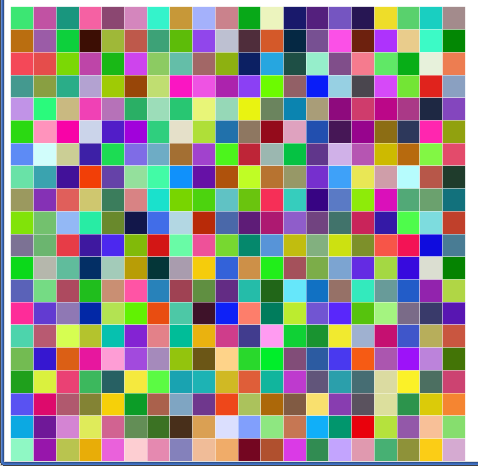
}

}

랜덤색상. 타일만들기

**package** {

**import** flash.display.Sprite;

 **import** flash.events.MouseEvent;

*// 임시 변수를 활용한 객체 이동*

**public** **class** AsExam5\_9 **extends** Sprite

{

**private** **var** arrGridSpr:Array;

**public** **function** AsExam5\_9()

{

displayMain();

addEventHandlers();

}

**private** **function** displayMain():**void**

{

**var** recSize:Number = 20;

**var** posX:Number = 0;

**var** posY:Number = 0;

arrGridSpr = **new** Array();

**for**(**var** i:Number=0; i<recSize; i++)

{

**for**(**var** j:Number=0; j<recSize; j++)

{

**var** sp:Sprite = **new** Sprite();

arrGridSpr.push(sp);

sp.graphics.lineStyle(.1, 0xFFFFFF, 0.4);

sp.graphics.beginFill(Math.random() \* 0x1000000);

sp.graphics.drawRect(5, 5, recSize, recSize);

sp.x = posX;

sp.y = posY;

**this**.addChild(sp);

posY += recSize;

}

posX += recSize;

posY = 0;

}

}

**private** **function** addEventHandlers():**void**

{

**this**.addEventListener(MouseEvent.CLICK, clickHandler);

}

*// 여기서는 마우스가 클릭될 때 마다 다음의 셀과 좌표값을 바꾸어 준다.*

**private** **function** clickHandler(event:MouseEvent):**void**

{

**for**(**var** i:Number=0; i<arrGridSpr.length; i++)

{

**var** tempX:Number = arrGridSpr[i].x;

**var** tempY:Number = arrGridSpr[i].y;

**if**((i+1) >= arrGridSpr.length) **break**;

arrGridSpr[i].x = arrGridSpr[i+1].x;

arrGridSpr[i].y = arrGridSpr[i+1].y;

arrGridSpr[i+1].x = tempX;

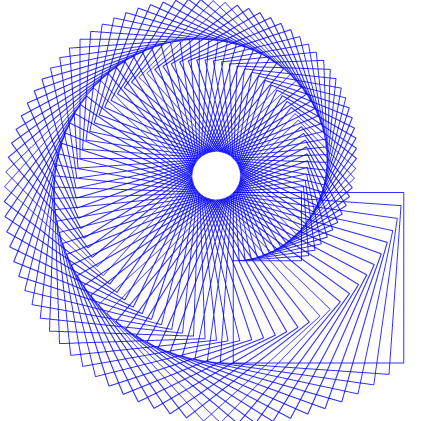
arrGridSpr[i+1].y = tempY;

}

}

}

}

재귀 호출하여 암모나이트 그리기

**package** {

**import** flash.display.Sprite;

*// 재귀 패턴으로 그린 암모나이트 문양*

**public** **class** AsExam5\_10 **extends** Sprite

{

*// 각 사각형의 각도를 저장하는 변수*

**private** **var** degree:Number = 0;

*// 사각형의 크기*

**private** **var** size:Number = 150;

**public** **function** AsExam5\_10()

{

displayMain();

}

*// 재귀 패턴으로 각도가 360도가 되면 재귀를 멈춘다*

**private** **function** displayMain():**void**

{

**if**(degree>360) **return**; *//재귀패턴이 끝나는 로직*

**var** sp:Sprite = **new** Sprite();

sp.graphics.lineStyle(1, 0x0000FF, .8);

sp.graphics.drawRect(15, 15, size, size);

sp.rotation = degree;

sp.x = 200;

sp.y = 150;

degree += 4; *//사각형을 4도씩 기울여서 그린다*

size--;*// 사각형 사이즈는 1씩 감소 시킨다*

**this**.addChild(sp);

displayMain();*//함수의 끝에서 자신을 다시 호출한다*

}

}

}

내부 class 선언방법 // 클릭과 삼단항연산자 이용 하이로우 게임 카드게임 에서볼수있다.

Pakcage{

import

Class{

....

}

}

Import..

Internal class ....{

}

내부클래스 참조.

그냥 가져다쓰면된다. 내부에서만 가능.

클래스 개념

오바리이딩 사용가능 override 라고 앞부분에 붙쳐줘야함

Public//모두허용

Private//나만허용

Protected//자식.들과 같은패키지

Internal//현패기지에만 허용

인터페이스 //자바와동일.

Package{

Public interface ID{

Function a(ints:int):void;

}

}

추상클래스

Abstract class..

예외처리

자바동일 Excetion->Error

**package**

{

**import** flash.display.Sprite;

**public** **class** AsExam6\_9 **extends** Sprite

{

**public** **function** AsExam6\_9()

{

**try**

{

*// 에러가 예상 되거나 에러를 발생시키는 코드*

**throw** **new** Error(**'okgosu.tistory.com error'**);

}

**catch** (err:Error)

{

*// catch ( )에 명시한 에러가 발생했을 때 실행할 문장*

**trace**(**'Error message: '** + err.message);

}

**finally**

{

*// 에러 발생 여부와 상관없이 무조건 실행할 문장*

**trace**(**'This finally block is always executed.'**);

}

}

}

}

클래스는 is-a와 ha-a가 있다. 상속? 포함?

Is // 변수가 특정클래스 데이터 타입 인터페이스와 호환가능한지?

Instanceof 변수가 특정 클래스의 인스턴지인지 확인 (is 권장)

In 변수가 특정객체의 멤버인지.

**package**

{

**import** flash.display.Sprite;

**public** **class** AsExam6\_12 **extends** Sprite

{

**public** **function** AsExam6\_12()

{

**var** sp:AsExam6\_8\_Circle = **new** AsExam6\_8\_Circle();

**trace**(sp **is** AsExam6\_8\_Circle);*// true*

**trace**(sp **instanceof** AsExam6\_8\_Circle);*// true*

**trace**(sp **is** Sprite);*// true*

**trace**(sp **instanceof** Sprite);*// true*

**trace**(sp **is** Object); *// true*

**trace**(sp **instanceof** Object);*// true*

**trace**(sp **is** IDrawable); *// true*

**trace**(sp **instanceof** IDrawable);*// instanceof는 interface에 대해 false를 리턴*

**trace**(sp **is** Boolean); *// false*

**trace**(**false** **is** Boolean); *// true*

}

}

}

//in 연산자.

**package**

{

**import** flash.display.Sprite;

**public** **class** AsExam6\_13 **extends** Sprite

{

**public** **function** AsExam6\_13()

{

**var** sp:Sprite = **new** Sprite();

**trace**(**"x"** **in** sp); *// true*

**trace**(**"y"** **in** sp); *// true*

**trace**(**"z"** **in** sp); *// false*

**var** okgosuArr:Array = [**"okgosu.net"**, **"okgosu.tistory.com"**];

**trace**(0 **in** okgosuArr); *// true*

**trace**(1 **in** okgosuArr); *// true*

**trace**(2 **in** okgosuArr); *// false*

*// okgosuArr의 모든 원소 출력*

**for each** (**var** k:String **in** okgosuArr)

{

**trace**(k);

}

}

}

}

클래스 타입호환성

**package**

{

**import** flash.display.Sprite;

**public** **class** AsExam6\_14 **extends** Sprite

{

**public** **function** AsExam6\_14()

{

**trace**(**this**.stage.frameRate);

**var** sp1:AsExam6\_8\_Circle = **new** AsExam6\_8\_Circle();

classTypeCheck1(sp1); *// object*

classTypeCheck2(sp1); *// object*

**var** sp2:Sprite = **new** Sprite();

classTypeCheck1(sp2); *// object*

*// classTypeCheck2(sp2); //*

*// 에러 발생 : 정적 유형 flash.display:Sprite의 값을 관련되지 않은 유형 OkGosuCircle\_3(으)로 암시적으로 강제 변환하려고 했습니다.*

}

**static** **private** **function** classTypeCheck1(p:Object):**void**

{

**trace**(**typeof** p);

}

**static** **private** **function** classTypeCheck2(p:AsExam6\_8\_Circle):**void**

{

**trace**(**typeof** p);

}

}

}**이벤트**

|  |  |
| --- | --- |
| **시스템이벤트** | **플래스 애플리케이션이 작동하는 동안 사용자가 아무 조작을하지 않더라도 내부적으로 발생하는 이벤트입니다.**   * **enterFrame 기본적으로 1초에 24번** * **TimerEvent 일정시간 간격을두고 발생시킬수있는이벤트** * **StatusEvent 네트워크나 디바이스 상태를 나타내는이벤트** * **progressEvent 데이터 로드할때로드 데이터크기와 전체크기를 나타내는 이벤** |
| **사용자이벤트** | **사용자의 조작으로 인해 발생하는 이벤트**   * **mouseEvent마우스 위치 마우스 클릭여부** * **keyboardEvent 키보드누름 여부 누린키보드값 입력한 텍스트값등 해당** * **FocusEvent,** * **MoveEvent** * **TextEvent** * **ProgressEvent** * **DataEvent** * **ListEvent** * **NetStatusEvent** * **FlexEvent** |
| **일반이벤트** | **Flash.event.\* 패키지에 기본적으로 들어있는 이벤트로. 모든 이벤트들의 부모이다.**  [**http://livedocs.adobe.com/flex/3/langref/flash/events/package-detail.html**](http://livedocs.adobe.com/flex/3/langref/flash/events/package-detail.html) |
| **사용자정의 이벤트** | **사용자 정의 이벤트는 EventDispatcher 클래스의 DispatchEvent() 함수를 이용해 그이벤트를 발생시켜야한다.**  **Flash.events.\*패키지에 들어있느 이벤트들을 상속해 만든 이벤트이다.** |

**Event 클래스의 주요 중요 속성**

|  |  |  |
| --- | --- | --- |
| **속성** | **데이터 타입** | **설명** |
| **type** | **String** | **이벤트 이름** |
| **target** | **EventDispatcher** | **이벤트를 발생시킨 객체 참조** |
| **currentTarger** | **EventDispatcher** | **현재 이벤트를 처리되고 있는 객체참조** |
| **eventPhase** | **Uint** | **현재 이벤트의 처리단계를 나타내는 상수값**  **EventPhase.CAPTURING\_PHASE:캡쳐 단계**  **EventPhase.AT\_TARGET:타깃 단계**  **EventPhase.BUBBLING\_PHASE: 버블단계** |
| **bubbles** | **Boolean** | **이벤트가 버블(진원지로 재전파) 단계를 거칠 것인지 설정(기본값 false)** |
| **Cancelable** | **Boolean** | **이벤트의 취소가능여부(기본 false)** |

**이벤트가 발생한 객체와 이벤트 리스너가 처리되고 있는 객체가 다른 경우가 있기때문에 아래 속성이 필요하다**

**target , currentTarger**

**Event클래스의 주요 함수**

**Event 클래스의 함수들은 이벤트 처리를 멈추거나 커스텀 이벤트를 마들 때 사용된다 Event()**

**함수는 생성자 함수로**

**이벤트명,**

**버플여부,**

**취소가능 여부 설정한다**

**clone() 함수는 커스텀 이벤트를 전파하기 위해서 사용한다**

**preventDefault()**

**stopImmediatePropagation();**

**stopPropagation()**

**은 이벤트 처리를 멈추게 할때 사용한다.**

|  |  |
| --- | --- |
| **함수정의** | **설명** |
| **Event(type:Strimg.bubbles:Boolean=false,cancelable:Boolean=false)** | **이벤트의 생성자 함수로 이벤트명,버블여부,취소가능여부를 매개변수로 넘김 , 버블 여부와 취소 가능 여부는 기본 false** |
| **Clone():Event** | **이벤트의 인스턴스를 복제한다 이는 사용자 정의 이벤트에서 이벤트를 복제할때 사용** |
| **formatToString(classname:String,... arguments):String** | **이벤트 객체의 정보를 문자열로 만들어 리턴한다.** |
| **isDefaultPrevented():Boolean** | **preventDefault() 함수가 호출되는지 리턴한다** |
| **preventDefault():void** | **이벤트가 취소 가능할 경우 기본적인 작동을 멈추게한다** |
| **stopImmediatePropagation():void** | **Guswo 노드를 포함하여 이하 모든 노드의 이벤트 처리를 멈추게한다.** |
| **stopPropagatio():void** | **Guswo 노드의 다음 노드의 이벤트 처리를 멈추게한다** |
| **toString():String** | **이벤트 객체의 속성값을 문자여로 리턴한다** |

**기본 이벤트**

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

**public** **class** AsExam6\_16 **extends** Sprite

{

**public** **function** AsExam6\_16()

{

**var** circleSpr:Sprite = **new** Sprite();

**this**.addChild(circleSpr);

circleSpr.graphics.beginFill(0xFF0000);

circleSpr.graphics.drawCircle(100, 100, 100);

*// circleSpr에 대해 마우스 클릭 이벤트 핸들러 함수를 등록한다.*

circleSpr.addEventListener(**MouseEvent.CLICK**, clickHandler);

}

**private** **function** clickHandler(event:MouseEvent):**void**

{

**trace**(event.type);

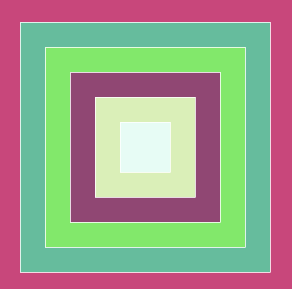
**trace**(event.target);

**trace**(event.stageX);

}

}

}

**이벤트 클래스 속성파악**

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

[**SWF**(width=800,height=800,backgroundColor=0xFFFFFF)]

**public** **class** AsExam6\_17 **extends** Sprite

{

**public** **function** A sExam6\_17() {

*// AsExam6\_17에 square\_0 추가*

**var** square\_0:AsExam6\_17\_Rect = **new** AsExam6\_17\_Rect(300);

**this**.addChild(square\_0);

*// square\_0에 square\_1 추가*

**var** square\_1:AsExam6\_17\_Rect = **new** AsExam6\_17\_Rect(250);

square\_0.addChild(square\_1);

*// square\_1에 square\_2 추가*

**var** square\_2:AsExam6\_17\_Rect = **new** AsExam6\_17\_Rect(200);

square\_1.addChild(square\_2);

*// square\_2에 square\_3 추가*

**var** square\_3:AsExam6\_17\_Rect = **new** AsExam6\_17\_Rect(150);

square\_2.addChild(square\_3);

*// square\_3에 square\_4 추가*

**var** square\_4:AsExam6\_17\_Rect = **new** AsExam6\_17\_Rect(100);

square\_3.addChild(square\_4);

*// square\_4에 square\_5 추가*

**var** square\_5:AsExam6\_17\_Rect = **new** AsExam6\_17\_Rect(50);

square\_4.addChild(square\_5);

클릭 이벤트전파

target

각각 리스너있어서 실행되면 각각 currentTarget

}

}

}

**import** flash.display.Sprite;

**import** flash.events.Event;

**import** flash.events.MouseEvent;

**class** AsExam6\_17\_Rect **extends** Sprite

{

**private** **var** sideLen:int;

**public** **function** AsExam6\_17\_Rect(pLength:int)

{

**this**.sideLen = pLength;

**this**.x = **this**.y = 25;

**this**.buttonMode = **true**;

**this**.addEventListener(MouseEvent.CLICK, clickHandler);

draw();

}

**private** **function** draw():**void**

{

**this**.graphics.clear();

**this**.graphics.lineStyle(.5, 0xFFFFFF, .8);

**this**.graphics.beginFill(Math.random() \* 0x1000000);

**this**.graphics.drawRect(0, 0, sideLen, sideLen);

}

**private** **function** clickHandler(e:Event):**void**

{

**trace**(**"eventPhase:"** + e.eventPhase + **" type:"** + e.type + **" target: "** + e.target.name +

**" currentTarget: "** + e.currentTarget.name + **" bubbles:"** + e.bubbles + **" cancelable:"** + e.cancelable);

draw();

}

}

**이벤트 전파 중지 함수**

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

[**SWF**(width=800,height=600,backgroundColor=0xFFFFFF)]

**public** **class** AsExam6\_18 **extends** Sprite

{

**public** **function** AsExam6\_18()

{

drawRects();

}

**private** **function** drawRects():**void**

{

**var** temp:Sprite = **null**;

**for**(**var** i:uint=0; i<10; i++)

{

**var** sp:AsExam6\_18\_Rect = **new** AsExam6\_18\_Rect(200);

*// 5번째 사각형에서 blockEvent를 true로 설정한다.*

**if**(i==5) sp.blockEvent = **true**;

**if**(temp == **null**)

{

**this**.addChild(sp);

} **else** {

temp.addChild(sp);

}

temp = sp;

}

}

}

}

**import** flash.display.GradientType;

**import** flash.display.Sprite;

**import** flash.events.Event;

**import** flash.events.MouseEvent;

**import** flash.geom.Matrix;

**class** AsExam6\_18\_Rect **extends** Sprite

{

**private** **var** sideLen:int;

**public** **var** blockEvent:Boolean;

**private** **var** colorFlag:Boolean = **false**;

**public** **function** AsExam6\_18\_Rect(pLength:int, blockEvent:Boolean=**false**)

{ **this**.sideLen = pLength;

**this**.blockEvent = blockEvent;

**this**.x = **this**.y = 10;

**this**.buttonMode = **true**;

*// 클릭 이벤트 핸들러에 대해 이벤트 핸들러 함수를 2개 등록한다.*

**this**.addEventListener(MouseEvent.CLICK, clickHandler1);

**this**.addEventListener(MouseEvent.CLICK, clickHandler2);

drawRainbow();

} *// 무지개빛 그라데이션으로 사각형을 그린다.*

**private** **function** drawRainbow():**void**

{

**this**.graphics.clear();

**this**.graphics.lineStyle(.5, 0xFFFFFF, .8);

**var** gradientBoxMatrix:Matrix = **new** Matrix();

gradientBoxMatrix.createGradientBox(150, 0, 0, 25, 0);

**var** colorSet:Array = **null**;

**if**(colorFlag) {

colorSet = [0x800080, 0x4B0082, 0x0000ff, 0x008000, 0xffff00, 0xffa500, 0xff0000];

colorFlag = **false**;

} **else** {

colorSet = [0xff0000, 0xffa500, 0xffff00, 0x008000, 0x0000ff, 0x4B0082, 0x800080];

colorFlag = **true**;

}

**this**.graphics.beginGradientFill(GradientType.LINEAR,

colorSet,

[1, 1, 1, 1, 1, 1, 1],

[0, 32, 64, 128, 170, 210, 255],

gradientBoxMatrix);

**this**.graphics.drawRect(0, 0, sideLen, sideLen);

**this**.graphics.endFill();

전파를 막았다.

5번째부터는 true 로 생성했기때문

}

**private** **function** clickHandler1(e:Event):**void**

{

**trace**(e.currentTarget.name + **' clickHandler1 was called'**);

*// blockEvent가 true면 이벤트 전파를 막는다*

**if**(blockEvent)

{

**e.stopPropagation();**

*//e.stopImmediatePropagation();*

}

}

**private** **function** clickHandler2(e:Event):**void**

{

**trace**(e.currentTarget.name + **' clickHandler2 was called'**);

drawRainbow();

}

}

**클릭시 생기고 또클릭시 없어지는 트리**

**package**

{

**import** flash.display.Sprite;

**import** flash.events.\*;

[**SWF**(width=800,height=800,backgroundColor=0xFFFFFF)]

**public** **class** AsExam6\_19 **extends** Sprite

{

**private** **var** childSpr:Sprite;

**public** **function** AsExam6\_19():**void**

{

childSpr = **new** Sprite();

**this**.addChild(childSpr);

childSpr.graphics.beginFill(0x0FF0FF);

childSpr.graphics.drawRect(0, 0, 800, 600);

childSpr.addEventListener(MouseEvent.CLICK, clickHandler);

}

*// 마우스를 클릭하면 AsExam6\_19\_SubTriangle을 생성하여 화면에 추가한다.*

**private** **function** clickHandler(event:MouseEvent):**void**

{

**var** sp:AsExam6\_19\_SubTriangle = **new** AsExam6\_19\_SubTriangle(Math.random() \* 0x1000000);

sp.x = event.stageX;

sp.y = event.stageY;

sp.addEventListener(Event.ADDED, spAdded);

sp.addEventListener(Event.ADDED\_TO\_STAGE, spAddedToStage);

sp.addEventListener(Event.REMOVED, spRemoved);

sp.addEventListener(Event.REMOVED\_FROM\_STAGE, spRemovedFromStage);

childSpr.addChild(sp);

}

**private** **function** spRemovedFromStage(event:Event):**void**

{

**trace**(**"스테이지에서 Sprite 제거"**);

}

**private** **function** spRemoved(event:Event):**void**

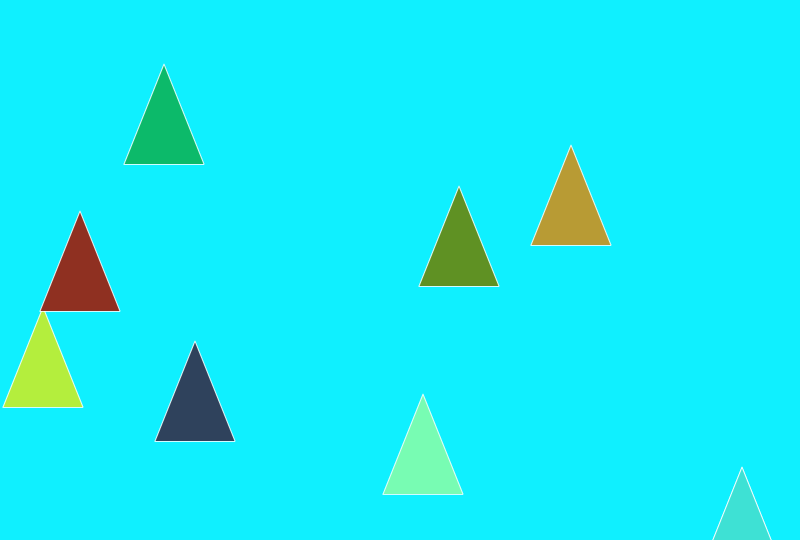
{

**trace**(**"Sprite 제거"**);

}

**private** **function** spAddedToStage(event:Event):**void**

{

 **trace**(**"스테이지에 Sprite 추가"**);

}

**private** **function** spAdded(event:Event):**void**

{

**trace**(**"Sprite 추가"**);

}

}

}

트리

**package**

{

**import** flash.display.Sprite;

**import** flash.events.\*;

[**SWF**(width=100,height=100,backgroundColor=0x00FFFF)]

**public** **class** AsExam6\_19\_SubTriangle **extends** Sprite

{

**public** **function** AsExam6\_19\_SubTriangle(inColor:uint)

{

**var** triangleHeight:Number = 100;

**var** triangleWidth:Number = 80;

*// 화면에 삼각형을 그린다.*

**this**.graphics.beginFill(inColor);

**this**.graphics.lineStyle(.5, 0xFFFFFF, .8);

**this**.graphics.moveTo(triangleWidth/2,0); *// 꼭지점 1*

**this**.graphics.lineTo(triangleWidth, triangleHeight);*// 꼭지점 2*

**this**.graphics.lineTo(0, triangleHeight);*// 꼭지점 3*

**this**.graphics.lineTo(triangleWidth/2, 0);*//다시 꼭지점 1으로*

**this**.addEventListener(MouseEvent.CLICK, remove);

}

*// 자신을 클릭했을 때 화면에서 제거하는 함수*

**private** **function** remove(event:Event):**void**

{

**if**(event.target == event.currentTarget)

{

**var** target:Sprite = Sprite(event.target);

target.parent.removeChild(target);

}

event.stopImmediatePropagation();

}

}}

**사각형 돌리는거..이벤트**

**package**

{

**import** flash.display.Sprite;

**import** flash.events.MouseEvent;

[**SWF**(width=800,height=600,backgroundColor=0xFFFFFF)]

**public** **class** AsExam6\_20 **extends** Sprite

{

**public** **function** AsExam6\_20()

{

drawOkGosuSquares();

}

*// 24개의 AsExam6\_20\_SubSquare를 서로 꼬리에 꼬리를 물게끔 추가한다.*

**private** **function** drawOkGosuSquares():**void**

{

**var** temp:Sprite = **null**;

**for**(**var** i:uint=0; i<24; i++)

{

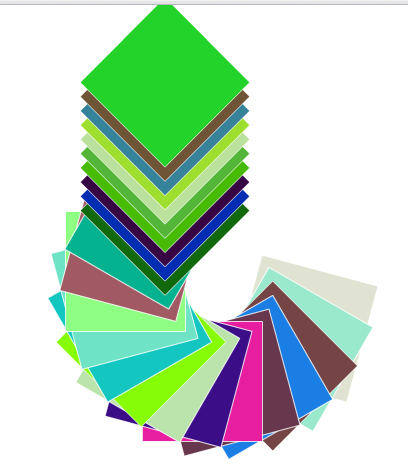
**var** sp:AsExam6\_20\_SubSquare = **new** AsExam6\_20\_SubSquare(120);

sp.x = sp.y = 10;

*// 임시 변수를 이용하여 클래스가 부모 자식관계로 추가되도록 한다.*

**if**(temp == **null**)

{

 **this**.addChild(sp);

sp.x = sp.y = 250;

} **else** {

temp.addChild(sp);

}

temp = sp;

}

}

}

}

//위에서쓰는 사각형

**package**

{

**import** flash.display.Sprite;

**import** flash.events.Event;

**import** flash.events.MouseEvent;

**public** **class** AsExam6\_20\_SubSquare **extends** Sprite

{

**private** **var** sideLen:int;

**public** **function** AsExam6\_20\_SubSquare(pLength:int)

{

**this**.sideLen = pLength;

**this**.x = **this**.y = 25;

**this**.buttonMode = **true**;

**this**.addEventListener(MouseEvent.CLICK, clickHandler);

draw();

}

*// 사각형을 그린다.*

**private** **function** draw():**void**

{

**this**.graphics.clear();

**this**.graphics.lineStyle(.5, 0xFFFFFF, .8);

**this**.graphics.beginFill(Math.random() \* 0x1000000);

**this**.graphics.drawRect(0, 0, sideLen, sideLen);

}

*// 클릭하면 15도씩 회전한다.*

**private** **function** clickHandler(e:Event):**void**

{

**this**.rotation += 15;

}

}

}

**타이머 시작 일시중지**

**package**

{

**import** flash.display.\*;

**import** flash.events.MouseEvent;

**import** flash.events.TimerEvent;

**import** flash.text.TextField;

**import** flash.utils.Timer;

[**SWF**(width=800,height=600,backgroundColor=0xFFFFFF)]

**public** **class** AsExam6\_22 **extends** Sprite

{

**private** **var** minuteTimer:Timer;

**private** **var** msTimerTF:TextField;

**private** **var** msCount:uint;

**private** **var** flag:Boolean;

**public** **function** AsExam6\_22()

{

layoutMyTimer();

*// 1/100초에 한번씩 타이머 이벤트가 발생하며, 6000번 작동한다.*

minuteTimer = **new** Timer(10, 6000);

*// 타이머 이벤트마다 실행될 이벤트 핸들러를 지정함*

minuteTimer.addEventListener(TimerEvent.TIMER, timerPlay);

*// 타이머 이벤트가 종료될 때 실행될 이벤트 핸들러를 지정함*

minuteTimer.addEventListener(TimerEvent.TIMER\_COMPLETE, timesUpHandler);

*// 타이머 이벤트를 시작함*

minuteTimer.start();

}

**private** **function** layoutMyTimer():**void**

{

*// 화면에 텍스트 필드를 추가한다.*

msTimerTF = **new** TextField();

msTimerTF.x = msTimerTF.y = 50;

stage.addEventListener(MouseEvent.CLICK, cickHandler);

**this**.addChild(msTimerTF);

}

**public** **function** timerPlay(event:TimerEvent):**void**

{

msCount++;

msTimerTF.text = **""** + msCount;

**trace**(**'msCount'**, msCount);

}

**public** **function** timesUpHandler(event:TimerEvent):**void**

{

**trace**(**"Time's Up!"**);

}

**private** **function** cickHandler(event:MouseEvent):**void**

{

**if**(flag) {

minuteTimer.start();

} **else** {

*// 타이머 이벤트를 중지함*

minuteTimer.stop();

}

flag = !flag;

}

}

}

**이벤트 사용 시계**

**package**

{

**import** flash.display.Sprite;

**import** flash.events.Event;

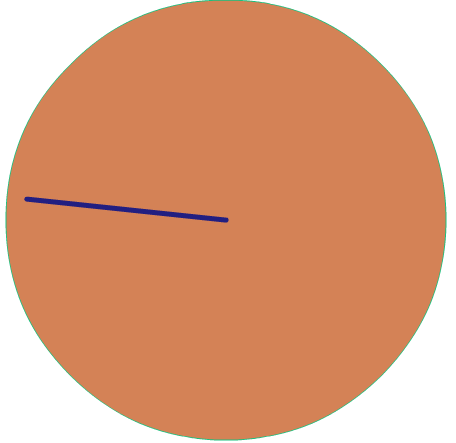
**import** flash.events.MouseEvent;

[**SWF**(width=800,height=600,frameRate=20,backgroundColor=0xFFFFFF)]

**public** **class** AsExam6\_15 **extends** Sprite

{

**private** **var** needle:Sprite;

 **private** **var** currentDegree:uint = 0;

**private** **var** needleMoveFlag:Boolean = **false**;

**public** **function** AsExam6\_15()

{

drawWatch();

drawNeedle();

}

*// 시계의 둥근 원을 그리는 함수*

**private** **function** drawWatch():**void**

{

**var** circleSp:Sprite = **new** Sprite();

circleSp.graphics.beginFill(Math.random() \* 0xffffff);

circleSp.graphics.lineStyle(1, Math.random() \* 0xffffff);

circleSp.graphics.drawCircle(250, 250, 220);

**this**.addChild(circleSp);

circleSp.addEventListener(MouseEvent.CLICK, clickHandler);

}

*// 시계 바늘을 그린다.*

**private** **function** drawNeedle():**void**

{

needle = **new** Sprite();

**this**.addChild(needle);

needle.x = 250;

needle.y = 250;

needle.graphics.beginFill(Math.random() \* 0xffffff);

needle.graphics.lineStyle(5, Math.random() \* 0xffffff);

needle.graphics.moveTo(0, 0);

needle.graphics.lineTo(0, -200);

needle.addEventListener(Event.ENTER\_FRAME, moveNeedle);

}

*// 클릭 이벤트가 발행할 때마다 시계바늘을 움직이거나 멈추게 한다.*

**private** **function** clickHandler(event:Event):**void**

{

**if**(!needleMoveFlag)

{

needle.removeEventListener(Event.ENTER\_FRAME, moveNeedle);

}

**else**

{

needle.addEventListener(Event.ENTER\_FRAME, moveNeedle);

}

needleMoveFlag = !needleMoveFlag;

}

*// 엔터프레임 이벤트가 발생할 때 마다 시계 바늘을 6도씩 회전한다.*

**private** **function** moveNeedle(event:Event):**void**

{

needle.rotation = currentDegree;

currentDegree += 6;

}

}

}

커스텀 이벤트

**package**

{

**import** flash.display.Sprite;

*// 사용자 정의 이벤트 클래스 import*

**import** GosuEvent;

**import** flash.text.TextField;

**public** **class** AsExam6\_23 **extends** Sprite

{

**private** **var** tf:TextField;

**private** **var** sHand:HandSprite;

**private** **var** rHand:HandSprite;

**private** **var** pHand:HandSprite;

**public** **function** AsExam6\_23()

{

sHand = **new** HandSprite(**'1'**); *//가위*

**this**.addChild(sHand);

sHand.x = 50;

rHand = **new** HandSprite(**'2'**);*//바위*

**this**.addChild(rHand);

rHand.x = 120;

pHand = **new** HandSprite(**'3'**);*//보*

**this**.addChild(pHand);

pHand.x = 190;

tf = **new** TextField();

tf.x = 90;

tf.y = 100;

tf.width = 200;

**this**.addChild(tf);

**this**.addEventListener(**'okgosu'**, actionHandler);

}

*// 가위바위보 이미지를 선택했을 때 실행되는 함수*

**private** **function** actionHandler(event:GosuEvent):**void**

{

**if**(event.hand == **"1"**)

{

tf.text = **'가위를 선택하셨습니다.'**;

rHand.restore();

pHand.restore();

} **else** **if**(event.hand == **"2"**) {

tf.text = **'바위를 선택하셨습니다.'**;

sHand.restore();

pHand.restore();

} **else** **if**(event.hand == **"3"**) {

tf.text = **'보를 선택하셨습니다.'**;

sHand.restore();

rHand.restore();

} **else** {

**trace**(**'error'**);

}

}

}

}

////가위바위보 컨포넌트 디스페치할곳

**package**

{

**import** flash.display.Loader;

**import** flash.display.Sprite;

**import** flash.events.\*;

**import** flash.net.URLRequest;

*// 가위바위보 이미지를 로드하여 보여주는 클래스*

**public** **class** HandSprite **extends** Sprite

{

*// 선택한 가위바위보값을 저장*

**public** **var** hand:String;

*// 가위바위보 이미지를 로드하기 위한 Loader*

**private** **var** ld:Loader;

**public** **function** HandSprite(imgName:String)

{

**this**.hand = imgName;

**this**.buttonMode = **true**;

loadHand();

}

*// assets폴더에 있는 가위바위보 이미지를 로드함*

**private** **function** loadHand():**void**

{

ld = **new** Loader();

**this**.addChild(ld);

ld.load(**new** URLRequest(**"assets/"** + hand + **".jpg"**));

ld.scaleX = 0.7;

ld.scaleY = 0.7;

**this**.addEventListener(MouseEvent.CLICK, clickHandler);

}

*// 로드한 이미지를 0.7배로 축소함*

**public** **function** restore():**void**

{

ld.scaleX = 0.7;

ld.scaleY = 0.7;

}

*// 가위바위보 이미지를 클릭하면 GosuEvent가 발생하면서 이미지를 1.1배 확대함*

**private** **function** clickHandler(event:MouseEvent):**void**

{

**this**.dispatchEvent(**new** GosuEvent(**"okgosu"**, hand));

ld.scaleX = 1.1;

ld.scaleY = 1.1;

}

}

}

디스플레이 리스트 아키텍처

액션스크립트의 모든 그래픽 요소는 flash.display.\* 패키지 안에있는 클래스들이다...

|  |  |
| --- | --- |
| 구성요소 | 설명 |
| 스테이지 Stage | 디스플레이 리스트의 베이스로서 앺,ㄹ리케이션의 바탕화면 즉 플래시 플레이어가 그리는영역 |
| 디스플레이 오브젝트 | 화면에 그려지는 그래픽 요소. bitmap,shap,textfield등.. |
| 디스플레이 오브젝ㅌ크 컨테이너 | 디스플레이 오브젝트나 디스플레이 오브젝트 컨테이너를 담아 배치하는 그래픽 요소;  Loader sprit movieClip |

로딩과정

1. 플래스 플레이어가 swf로드
2. swf안에 포함된 메인액션클래스찾음
3. 메인 액션스크립트 클래스 생서앚 함수 호출 인스턴스 생성
4. 생성된 인스턴스를 스테이즈 객체의 자식으로 추가

**package**

{

**import** flash.display.Sprite;

**import** flash.events.Event;

**import** flash.events.MouseEvent;

**import** flash.text.TextField;

[**SWF**(width=400,height=400,backgroundColor=0xCCCCCC)]

**public** **class** AsExam7\_1 **extends** Sprite

{

*// 프레임레이트를 위한 변수*

**private** **var** frNum:int=2;

*// 디스플레이 모드를 표시하는 텍스트*

**private** **var** tf:TextField = **new** TextField();

*// 회전할 날개 생성*

**private** **var** wing1:AsExam7\_1\_Wing = **new** AsExam7\_1\_Wing(0xFF0000);

**private** **var** wing2:AsExam7\_1\_Wing = **new** AsExam7\_1\_Wing(0x0000FF);

**public** **function** AsExam7\_1()

{

displayMain();

addEventHandlers();

}

**private** **function** displayMain():**void**

{

**this**.stage.frameRate = frNum;

**this**.addChild(wing1);

**this**.addChild(wing2);

**this**.addChild(tf);

**trace**(**'frameRate: '** + **this**.stage.frameRate);

wing1.x = 200;

wing1.y = 150;

wing2.x = 200;

wing2.y = 150;

wing2.rotation = 180;

tf.text = **this**.stage.scaleMode;

tf.x = 200;

tf.y = 150;

}

**private** **function** addEventHandlers():**void**

{

*// 빨간 날개를 클릭하면 프레임레이트를 올려서 회전 속도를 증가시킴*

wing1.addEventListener(MouseEvent.CLICK, clickHandler);

wing2.addEventListener(MouseEvent.CLICK, fullSizeHandler);

stage.addEventListener(Event.ENTER\_FRAME, rotateWings);

}

*// 엔터프레임 이벤트가 발생하면 날개를 회전시킨다.*

**private** **function** rotateWings(evt:Event):**void**

{

wing1.rotation += 5;

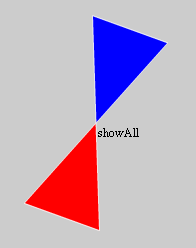
wing2.rotation += 5;

}

*// 파란 날개를 클릭하면 디스플레이 모드를 바꿈*

**private** **function** fullSizeHandler(evt:MouseEvent):**void**

{

 **switch**(stage.displayState)

{

**case** **"normal"**:

stage.displayState = **"fullScreen"**;

**break**;

**case** **"fullScreen"**:

stage.displayState = **"normal"**;

**break**;

}

**var** rNum:int = Math.random() \* 4;

**switch**(rNum)

{

**case** 0:

stage.scaleMode = **"exactFit"**;

**break**;

**case** 1:

stage.scaleMode = **"showAll"**;

**break**;

**case** 2:

stage.scaleMode = **"noScale"**;

**break**;

**case** 3:

stage.scaleMode = **"noBorder"**;

**break**;

}

tf.text = **this**.stage.scaleMode;

**trace**(**'stage.scaleMode: '** + stage.scaleMode);

**trace**(**'stage.displayState: '** + stage.displayState);

}

*// 회전 속도 증가*

**private** **function** clickHandler(evt:MouseEvent):**void**

{

**trace**(**'frameRate: '** + **this**.stage.frameRate);

frNum+= 5;

**this**.stage.frameRate = frNum;

**trace**(**'frameRate: '** + **this**.stage.frameRate);

}

}

}

XML 사용하기.

<fx:Script>

<![CDATA[

**import** mx.events.FlexEvent;

**var** xml:String=**""** +

**"<w>"** +

**"<a id='good'>"** +

**"babo"** +

**"</a>"**+

**"</w>"**;

**protected** **function** creationComplete(event:FlexEvent):**void**

{

**var** xmlo:XML = **new** XML(xml);

**trace**(xmlo.toXMLString());

**trace**(xmlo.a);

}

]]>

</fx:Script>