



[1] Pixi.js 로 텍스트 움직여보기

1. Pixi.js란?

The HTML5 Creation Engine Create beautiful digital content with the fastest, most flexible 2D **WebGL** **renderer**.

HTML5 생성 엔진 가장 빠르고 유연한 2D WebGL 렌더링을 통해 아름다운 디지털 콘텐츠를 만드십시오.

WebGL이란?

- 웹 브라우저에서 아무런 플러그인 없이 3D 그래픽을 그릴 수 있는 사실 상의 표준 3D 그래픽 라이브러리
- W3C에서 관리하지 않기 때문에 W3C HTML5 스펙에 WebGL 없음. 그러나 사용률 기준 글로벌 브라우저에서 거의 다 WebGL 지원한다.
- 그림을 그리는 성능이 캔버스보다 뛰어나고, CPU에게 더 많은 여유를 준다.
비교 : [Canvas2D](#) vs [WebGL](#)
- 비동기 (non-blocking) : JavaScript는 WebGL API인 gl.dorw***() 호출로 그리기를 그래픽 처리 장치 (Graphics Processing Unit)에 위임하고, 그리기 결과 대기없이 바로 다음 라인의 JavaScript를 수행한다.

rendering이란?

- 평면인 그림에 형태·위치·조명 등 외부의 정보에 따라 다르게 나타나는 [그림자·색상·농도](#) 등을 고려하면서 실감나는 3차원 화상을 만들어내는 과정 또는 그러한 기법을 일컫는다.

이 라이브러리를 사용해 만든 애니메이션은, WebGL을 자동적으로 사용해 주고 있다.

pixi.js의 examples 페이지에 여러가지 예제가 소개되어 있다. <http://www.pixijs.com/>

*Star warp - sprites <https://pixijs.io/examples/#/demos/star-warp-sprites.js> *Mouse blending <https://pixijs.io/examples/#/filters/filter-mouse.js> *Slots demo <https://pixijs.io/examples/#/demos/slots-demo.js>

2. Pixi.js 설치하기

<https://github.com/pixijs/pixi.js> <https://github.com/pixijs/pixi.js/releases>

[pixi.js](#) or [pixi.min.js](#) 를 다운로드 받아 파일을 로드하면, pixi.js 기능을 이용하는것이 가능해진다.

```
<script src="pixi.js"></script>
```

3. 텍스트 움직여 보기

```

<!DOCTYPE html>
<html lang="ko">
<head>
  <title>sample1</title>
  <meta charset="UTF-8">
  <meta name="description" content="">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="viewport" content="width=1920, user-scalable=no">
  <script src="js/pixi.js"></script>
</head>
<body>
  <div id="pixiview"></div>
  <script>
    var width = 600;
    var height = 400;

    //스테이지를 만든다
    var stage = new PIXI.Stage(000);

    //렌더링을 만든다
    var renderer = PIXI.autoDetectRenderer(width, height);

    //렌더링의 view를 html의 요소에 추가한다
    document.getElementById('pixiview').appendChild(renderer.view);

    //텍스트 오브젝트를 만든다
    var text = 'Hello World!';
    var style = new PIXI.TextStyle({
      fontFamily:'Arial',
      fontSize:90,
      fontWeight:'bold',
      fill:'#fff'
    });
    var textobj = new PIXI.Text(text, style);
    textobj.anchor.x = 0.5;
    textobj.anchor.y = 0.5;
    textobj.position.x = width/2;
    textobj.position.y = height/2;

    var blurFilter = new PIXI.filters.BlurFilter();
    textobj.filters = [blurFilter];

    //텍스트 오브젝트를 스테이지에 싣는다
    stage.addChild(textobj);

    var count = 0;
    //애니메이션 함수를 정의한다
    function animate(){
      count += 0.01;
      requestAnimationFrame(animate);
      var blurAmount = Math.cos(count);
      blurFilter.blur = 20 * (blurAmount);
      textobj.rotation += 0.03;
    }
  </script>
</body>
</html>

```

```

        renderer.render(stage);
    }

    //다음의 애니메이션 프레임으로 animate()를 불러온다
    requestAnimationFrame(animate);
</script>
</body>
</html>

```

전체적인 흐름

pixi.js 애니메이션은 기본적으로 아래와 같은 흐름으로 이루어진다.

1. 오브젝트 (화상 스프라이트나 텍스트등), 스테이지, 렌더링을 만든다.

```

var width = 600;
var height = 400;

var stage = new PIXI.Stage(000);
var renderer = PIXI.autoDetectRenderer(width, height);

```

PIXI.STAGE() 를 생성한다. 파라미터에 있는것은 배경색이다. PIXI.autoDetectRenderer() 를 생성한다. 지정하고 있는것은, 렌더링이 취급하는 화면영역의 높이, 가로사이즈 이다.

2. 렌더링의 View를 html의 요소에 추가한다.

```

document.getElementById('pixiview').appendChild(renderer.view);

```

3. 애니메이션 오브젝트를 만든다.

```

var text = 'Hello world!';
var style = new PIXI.TextStyle({
    fontFamily:'Arial',
    fontSize:90,
    fontWeight:'bold',
    fill:'#fff'
});

var blurFilter = new PIXI.filters.BlurFilter();
textobj.filters = [blurFilter];

var textobj = new PIXI.Text(text, style);
textobj.anchor.x = 0.5;
textobj.anchor.y = 0.5;
textobj.position.x = width/2;
textobj.position.y = height/2;

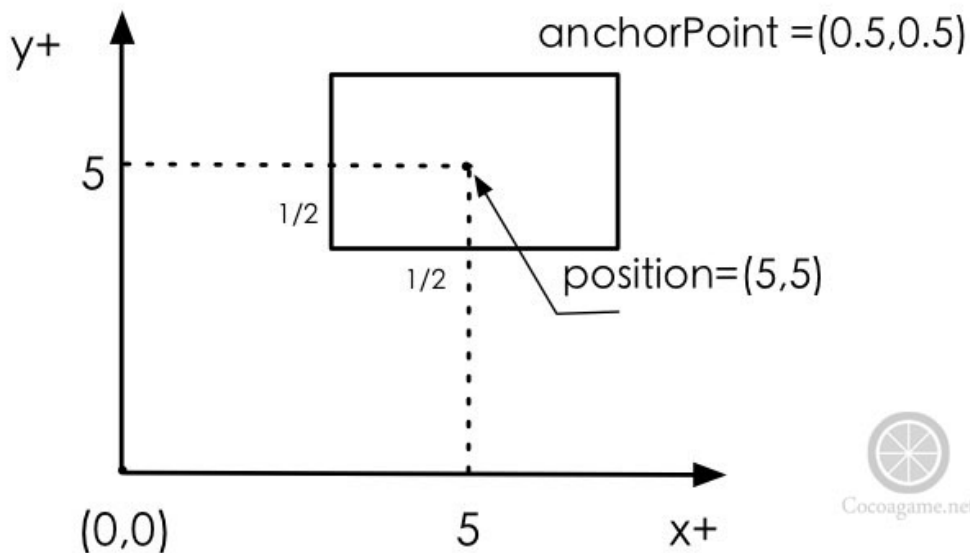
```

먼저 변수 text에 글자를 넣어주고, new PIXI.TextStyle를 생성하여 글자 사이즈랑 색깔등을 지정해 주고, new PIXI.Text(text, style) 를 생성하여 텍스트의 위치와 값을 정해주었다. anchor와 position 프로퍼티로 객체의 위치를 지정해 주고, blur 효과도 추가했다.

노드에는 두 가지 중요한 속성 인 position과 anchorPoint가 있습니다.

position 속성은 Node 객체의 실제 위치 (위치)입니다. position 속성은 종종 anchorPoint 특성과 함께 사용되어 화면의 특정 위치에 정확하게 배치 된 Node 객체 (표준 직사각형 그래픽)를 변환하고 직사각형 앵커를 설정해야하며 anchorPoint는 위치 비율에 상대적으로, 기본값은 (0.5,0.5)입니다. 다음과 같은 몇 가지 조건을 살펴 봅시다.

예를 들어 anchorPoint (0.5,0.5)를 기준으로 기본값입니다.



아래는 anchorPoint (0.0,0.0) 상황입니다.

3. 애니메이션 오브젝트를 스테이지에 실는다.

```
stage.addChild(textobj);
```

오브젝트 스테이지 등록은 스테이지 오브젝트의 addChild()메서드로 행한다.

4. 애니메이션 함수를 정의한다.

```
var count = 0;
//애니메이션 함수를 정의한다
function animate(){
    count += 0.01;
    var blurAmount = Math.cos(count);
    blurFilter.blur = 20 * (blurAmount); //블러효과
    textobj.rotation += 0.03; //텍스트 회전
    renderer.render(stage); //그림을 그린다
    requestAnimationFrame(animate); // 다음 타이밍에서 애니메이션을 부른다
}
```

5. 다음의 애니메이션 프레임으로 animate()를 불러온다.

```
requestAnimationFrame(animate);
```

참고 사이트

- <http://www.pixijs.com>
- <https://qiita.com/tadfmac>
- <https://www.slideshare.net/hanmomhanda/web-gl-42962918>
- <https://terms.naver.com/entry.nhn?docId=1231713&cid=40942&categoryId=32828>
- <https://www.programering.com/a/MDOxUzMwATA.html>