

OpenWebOS WebApp

May, 2016
Seungjae Baek

Dept. of software
Dankook University

<http://embedded.dankook.ac.kr/~baeksj>

Contents

- Drawing app design
- Create WebOS project
- HTML5 part implementation
- CSS part implementation
- JavaScript part implementation
- Result

Development Process

■ Drawing app design

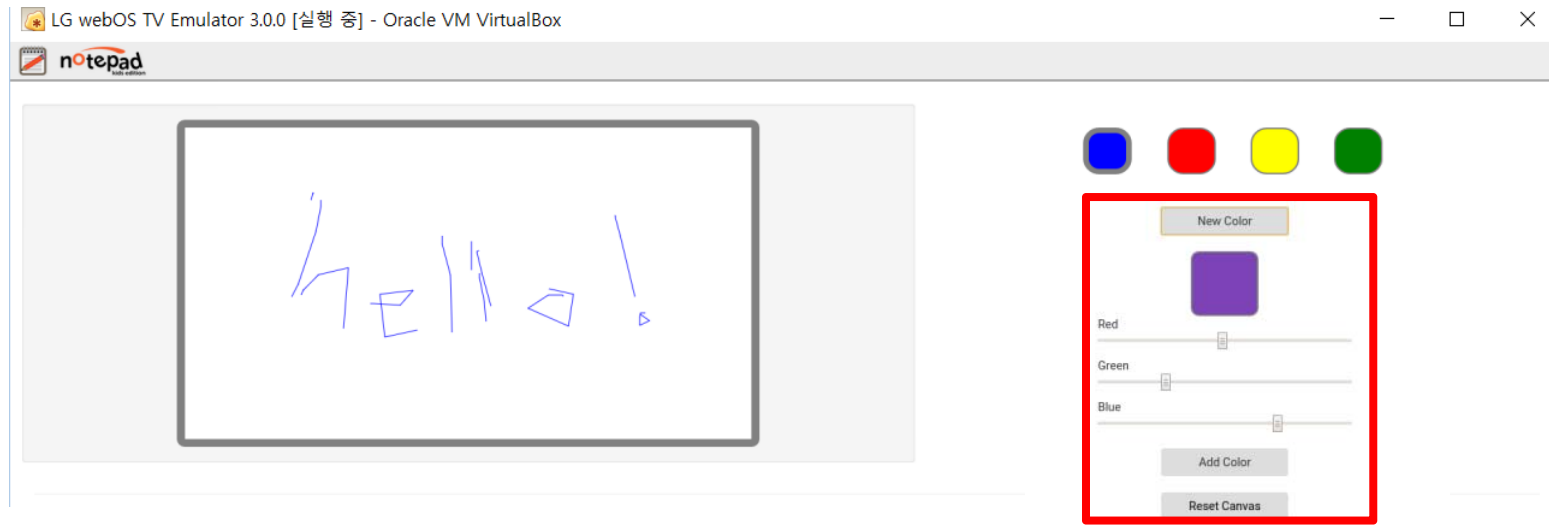


✓ Drawing application

- 색 선택 기능
- 색 추가 기능 - R / G / B 로 조절하여 색 추가
- 그림판 기능
- 그림판 초기화 기능

Development Process

■ Drawing app design



✓ Drawing application

- 색 선택 기능
- 색 추가 기능 – R / G / B 로 조절하여 색 추가
- 그림판 기능
- 그림판 초기화 기능

Development Process

■ Drawing app design

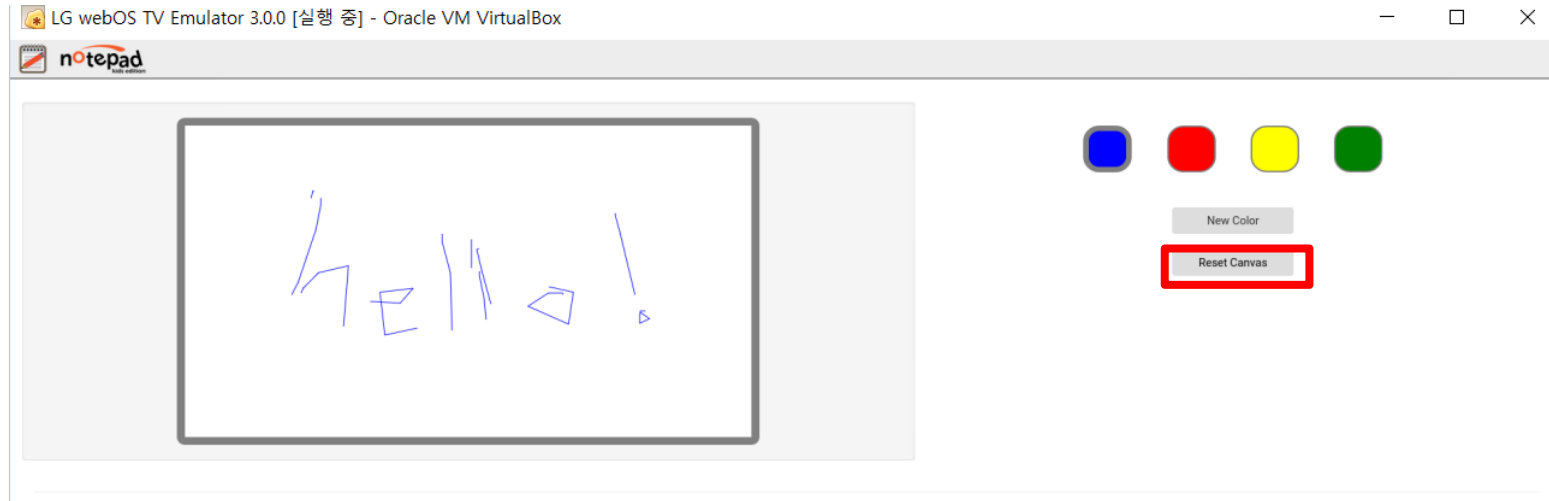


✓ Drawing application

- 색 선택 기능
- 색 추가 기능 – R /G /B 로 조절하여 색 추가
- **그림판 기능**
- 그림판 초기화 기능

Development Process

■ Drawing app design



✓ Drawing application

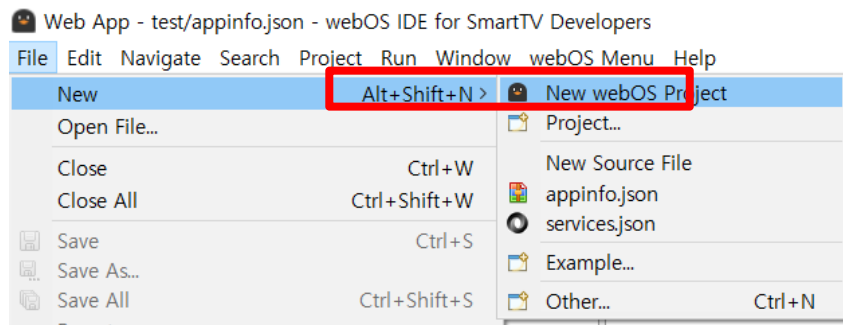
- 색 선택 기능
- 색 추가 기능 – R /G /B 로 조절하여 색 추가
- 그림판 기능
- **그림판 초기화 기능**

Create webOS project

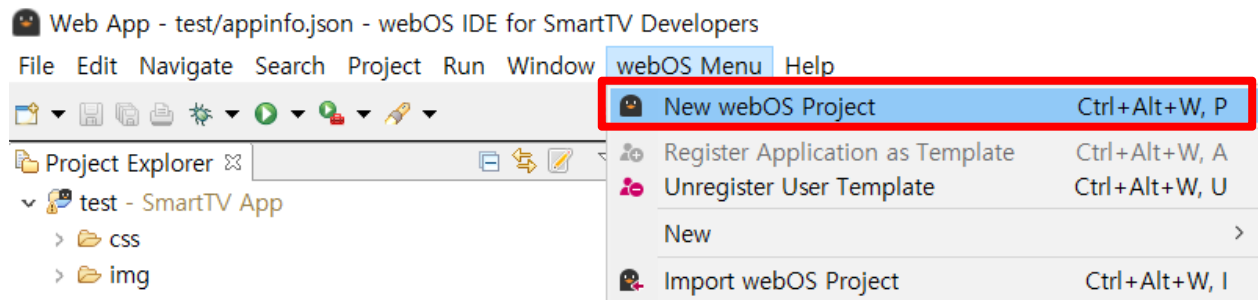
■ Create project

✓ 3 types of project creation

■ File -> New -> New webOS Project



■ webOS Menu -> New webOS Project

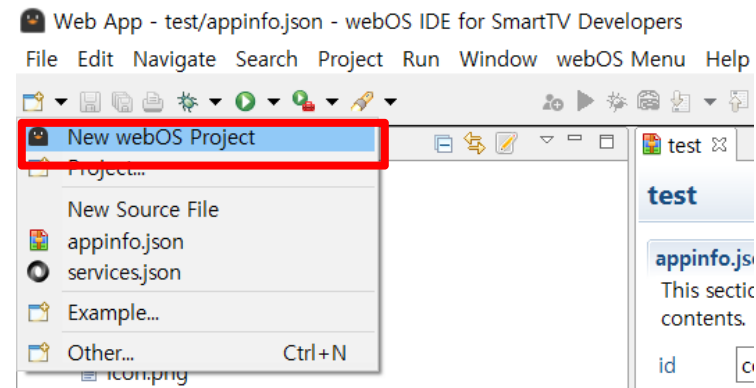


Create webOS project

- Create project

- ✓ 3 types of project creation

- Icon -> New webOS Project

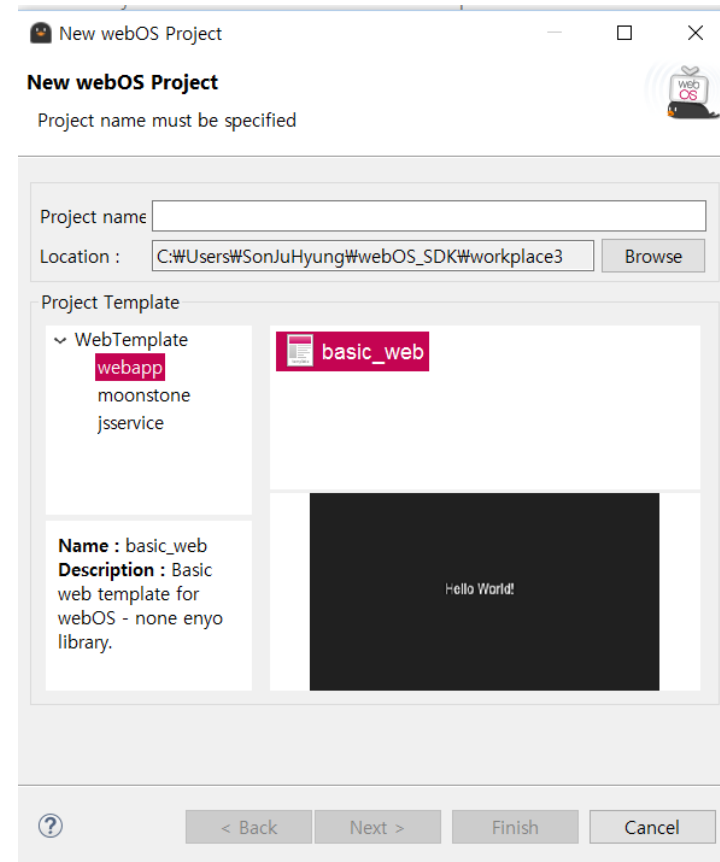


Development Purpose

■ Create project

✓ Edit webOS project

- 1) Project name 지정
- 2) Template 지정 -> webapp
- 3) Finish

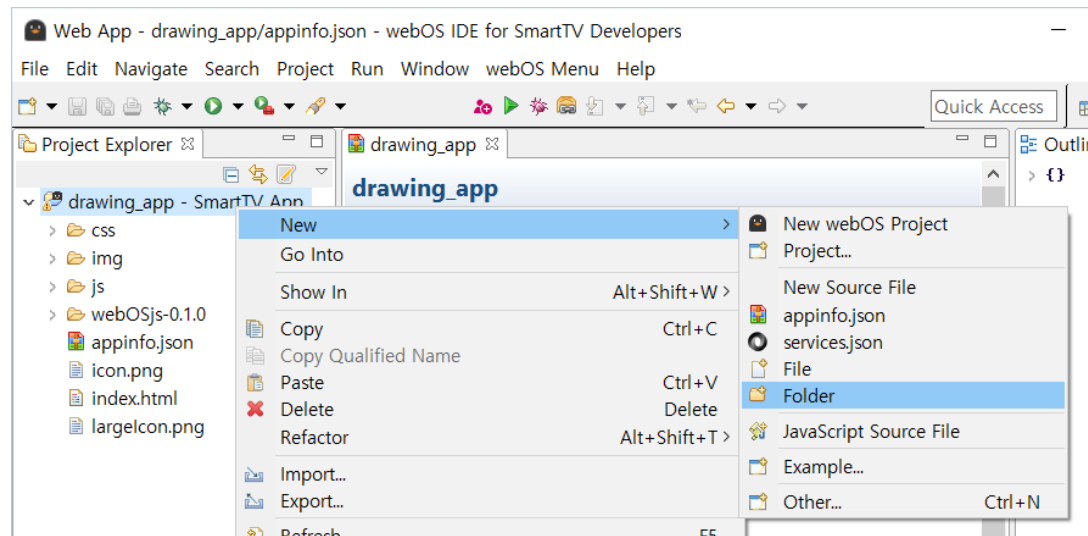


Create webOS project

■ Create project

✓ Project structure

- css, img, js 폴더 추가
 - css – HTML의 style 을 지정할 css 파일을 위한 폴더
 - js – 버튼 event등의 action을 handling하는 JavaScript를 위한 폴더
 - Img – 프로젝트에 사용할 image폴더
- Project name -> New -> Folder

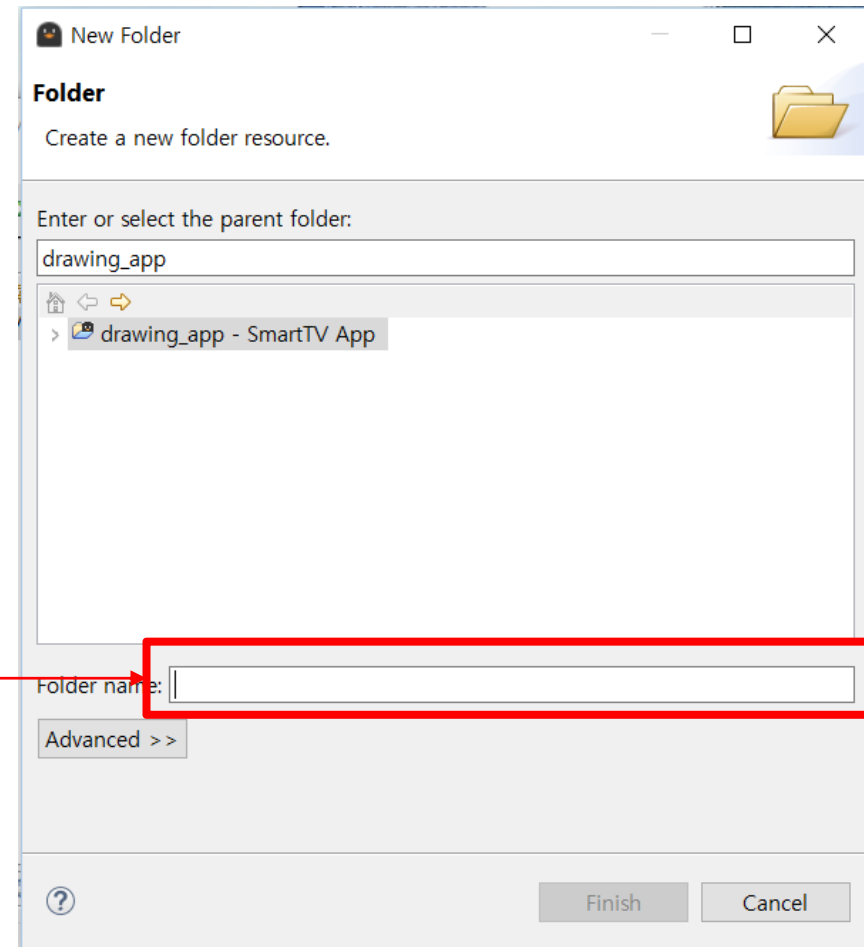
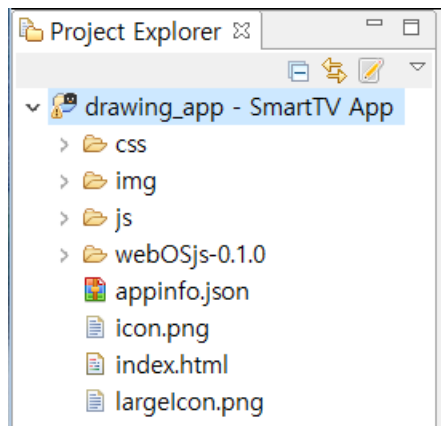
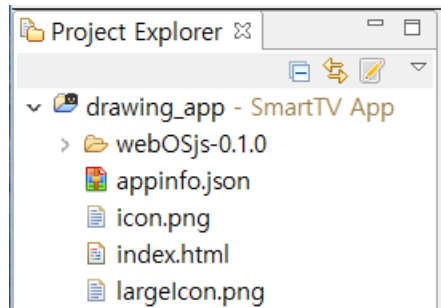


Create webOS project

■ Create project

✓ Project structure

- 폴더 이름 지정



HTML5 implementation

■ index.html

- ✓ Emulator에서 application 실행 시 보여질 첫 화면 설계
- ✓ Header implementation

```
<!DOCTYPE html>
<html>
<head>
  <title>Palette drawing Application</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
  <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css">

  <link href='http://fonts.googleapis.com/css?family=Roboto' rel='stylesheet' type='text/css'>
  <link rel="stylesheet" href="css/style.css" type="text/css" media="screen" title="no title" charset="utf-8">
  <link href='http://fonts.googleapis.com/css?family=Droid+Sans' rel='stylesheet' type='text/css'>
  <link rel="icon" href="img/pencil.ico">
</head>
```

- Stylesheet 로 css를 HTML과 연결
- Google API 의 폰트 사용
 - <http://fonts.googleapis.com/css?family=Roboto>
 - <http://fonts.googleapis.com/css?family=Droid+Sans>
- MaxCDN에서 제공하는 BootstrapCDN의 css를 사용
 - <https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css>
 - <http://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css>
- Image 폴더의 image 사용 – 현재 프로젝트에서는 pencil.ico라는 파일 사용

HTML5 implementation

■ index.html

✓ Body implementation

```
<body>
<div class="wrapper">
  <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      
    </div>
  </nav>

```

- nav 태그를 통해 주요 링크들을 그룹 지정
- navbar-toogle 버튼 생성

HTML5 implementation

■ index.html

✓ Body implementation

```
<div class="row">
  <div class="col-md-7 well">
    <canvas width="700" height="400"></canvas>
  </div>
  <div class="col-md-5">
    <div class="buttons">
      <ul>
        <li class="blue selected"></li>
        <li class="red"></li>
        <li class="yellow"></li>
        <li class="green"></li>
      </ul>
    </div>
    <button class="btn" id="revealColorSelector">New Color</button>
    <div id="colorSelector">
      <span id="newColor"></span>
      <p>
        <label for="red">Red</label>
        <input id="red" name="red" type="range" min="0" max="255" value="0">
      </p>
      <p>
        <label for="green">Green</label>
        <input id="green" name="green" type="range" min="0" max="255" value="0">
      </p>
      <p>
        <label for="blue">Blue</label>
        <input id="blue" name="blue" type="range" min="0" max="255" value="0">
      </p>
    </div>
  </div>
</div>
```

- Canvas 태그를 통해 브라우저 위에 동작하는 그림판 생성
- 각 색들의 목록을 li 태그를 통해 그룹
- 새로운 색을 추가하는 버튼 생성
- 버튼 클릭 시 Red / Green / Blue 의 각 값을 입력 값으로 받을 수 있도록 지정 input 태그 지정

HTML5 implementation

■ index.html

✓ body implementation

```
        <button class="btn" id="addNewColor">Add Color</button>
    </div>
    <a href=""><button class="btn reset" id="revealColorSelector">Reset Canvas</button></a>
</div>
</div>
</div>
<hr>
</div>

</div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script src="js/app.js" type="text/javascript" charset="utf-8"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
</body>
</html>
```

- 색 추가 버튼(Add Color) 생성 및 그림판 초기화 버튼(Reset Canvas)생성
- MaxCDN에서 제공하는 BootstrapCDN의 JavaScript를 사용
 - <https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js>
- Ajax로 jquery를 가져오기 위해 사용
 - <https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js>

CSS implementation

■ style.css

```
body{
  font-family: 'Droid Sans', sans-serif;
  counter-reset: section;
  left: 0;
  margin: 80px 30px;
  position: relative;
  font-family: 'Roboto';
}

.navbar {
  background-color: #EDED;
}

.toplogo {
  margin-top: 5px;
  margin-left: 10px;
  max-height: 40px;
}

.top-nav {
  padding: 0 15px;
  background-color: #EDED;
}

canvas {
  display: block;
  max-width: 100%;
  max-height: 100%;
  background: white;
  border: 10px solid gray;
  border-radius: 10px;
  margin: auto;
  cursor: url(../img/pencil.png), crosshair;
}

canvas:hover {
  border: 10px solid #F15A29;
}
```

- ✓ body
 - 글꼴 지정(사용자 지정)
 - Droid Sans, sans-serif, Roboto
 - counter 사용
 - 바깥 여백 지정, 태그 위치 지정
- ✓ navbar
 - 배경색 지정
- ✓ toplogo
 - 위쪽, 왼쪽의 여백 지정 및 최대 크기 지정
- ✓ top-nav
 - 배경색, 내부 여백 지정
- ✓ canvas
 - layout 요소를 block으로 지정하여 박스로 생성
 - 최대 크기 지정, 배경색 지정, 여백 지정

CSS implementation

■ style.css

```
ul {
  list-style:none;
  margin: 0;
  float: left;
  padding: 10px 0 20px;
  width: 100%;
  text-align: center;
}

ul li {
  height: 60px;
  width: 60px;
  border-radius: 20px;
  cursor: pointer;
  display: inline-block;
  margin: 20px;
  border: 2px solid gray;
}

ul li:hover {
  border: 2px solid black;
}

.selected {
  border: 7px solid gray;
  height: 60px;
  width: 60px;
}

.selected:hover{
  border: 7px solid black;
  height: 60px;
  width: 60px;
}

.blue{
  background: blue;
}
```

- ✓ u1
 - list의 블릿 숫자 변경
 - block 인 displa의 위치 설정
 - 경계, 내부 여백 지정, text 정렬 지정
- ✓ u1 li
 - 크기속성, 테두리 속성 지정
 - 커서 모양 내부 여백, 태그 테두리 지정
- ✓ u1 li hover
 - 경계 지정
- ✓ selected , selected hover
 - 경계, 크기 지정, 마우스 오버 시 지정

CSS implementation

■ style.css

```
.red {  
  background: red;  
}  
  
.yellow {  
  background: yellow;  
}  
  
.green {  
  background: green;  
}  
  
.btn {  
  display: block;  
  margin: auto;  
  width: 150px;  
}  
  
.btn:hover{  
  background-color: #F15A29;  
}  
  
#colorSelector {  
  display:none;  
}  
  
#newColor {  
  display: block;  
  width: 80px;  
  height: 80px;  
  border-radius: 10px;  
  margin: auto;  
  border: 2px solid gray;  
  margin-top: 20px;  
}
```

- ✓ blue, yellow, green
 - 색 지정
- ✓ button
 - 버튼 크기 및 모양 지정
- ✓ button
 - 경계, 크기 지정
- ✓ button hover
 - 마우스 오버 시 색 지정
- ✓ color selector
 - layout을 보이지 않도록 지정
- ✓ new color
 - 추가될 색의 크기, 위치 등 속성 지정

CSS implementation

■ style.css

```
p {
  max-width: 300px;
  display: block;
  margin: auto;
  padding-bottom: 10px;
}

label {
  display: block;
  margin: auto;
}

#addNewColor {
  margin-top: 10px;
}

.reset {
  color: black;
  margin-top: 20px;
}

a {
  color: black;
}

a:hover {
  color: black;
  text-decoration: none;
}
```

- ✓ p
 - layout 을 블록 박스로 지정, 크기 지정 등
- ✓ label
 - 라벨 텍스트의 블록 박스 지정 및 여백 지정
- ✓ addNewColor
 - 여백 지정
- ✓ reset
 - color reset버튼의 여백 및 색 지정
- ✓ a , a hover
 - a 태그의 색 및 a 태그 오버 효과 시 색 지정

JavaScript implementation

■ app.js

✓ 변수 지정

```
$(document).ready(function(){  
  
  var color = $(".selected").css("background-color");  
  var $canvas = $("canvas")  
  var context = $canvas[0].getContext("2d");  
  var lastEvent;  
  var mouseDown = false;
```

- 선택한 색을 저장할 변수
- canvas의 상태를 저장할 변수
- 마지막 실행된 event를 임시적으로 저장할 변수
- mouse Event를 관리하는 변수

JavaScript implementation

■ app.js

```
$('.buttons').on('click', "li", function(){
    $(this).siblings().removeClass('selected');
    $(this).addClass('selected');

    color = $(this).css('background-color');
})

$('#revealColorSelector').click(function(){
    changeColor();
    $('#colorSelector').toggle();
});
```

- 선택된 색깔의 class를 추가 하며 canvas의 배경으로 설정한다.
- 색깔이 선택되었을 시 toggle되도록 한다.

JavaScript implementation

■ app.js

✓ 함수 구현

```
function changeColor(){
  var r = $("#red").val();
  var g = $("#green").val();
  var b = $("#blue").val();

  $("#newColor").css("background-color", "rgb(" + r + ", " + g + ", " + b + ")");
}
```

- RED, GREEN, BLUE의 바에 값을 받아 각각 r, g, b변수에 저장하고 설정한 색을 새로운 색으로 추가한다.

JavaScript implementation

■ app.js

```
$("#input[type=range]").change(changeColor);
$("#addNewColor").click(function(){
    var $newColor = $("- </li>");
    $newColor.css("background-color", $("#newColor").css("background-color"));
    $(".buttons ul").append($newColor);
    $newColor.click();
});

```

- 색이 변경되었을 때 배경색을 변경하며, 색이 추가되었을 시 색 목록에 추가가 되도록 설정한다.

JavaScript implementation

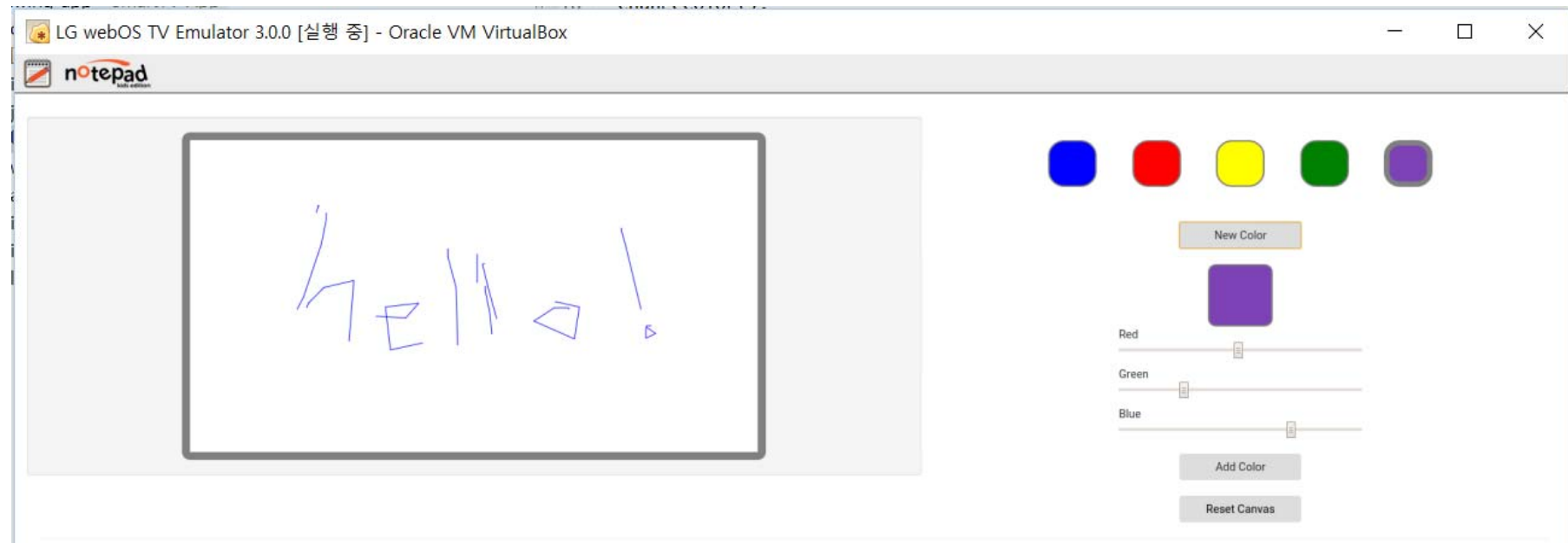
- app.js

- 마우스 처리

```
$canvas.mousedown(function(e){
  lastEvent = e;
  mouseDown = true;
}).mousemove(function(e){
  if(mouseDown) {
    context.beginPath();
    context.moveTo(lastEvent.offsetX, lastEvent.offsetY);
    context.lineTo(e.offsetX, e.offsetY);
    context.strokeStyle = color;
    console.log(color);
    context.stroke();
    lastEvent = e;
  }
}).mouseup(function(){
  mouseDown = false;
}).mouseleave(function(){
  $canvas.mouseup();
});
})
```


Result

■ Project result





Thank you