

# TIZEN™ DEVELOPER CONFERENCE MAY 7-9, 2012



## HTML5 Canvas vs CSS3

Sirisha Muppavarapu & Brian Jones

# Scope

- Provide functional comparative analysis of both approaches to help you decide which best suits your needs (using WebKit browsers)
- Provide examples
- Demo

# HTML5

- Why are we interested in HTML5?
- Awesome features introduced in HTML5
- Power of CSS3 expands the scope and usability of HTML
- Earlier it was only webpages and now... apps?!!
- Ease of making HTML apps with the knowledge of making webpages

# CSS3

- Power/Features of CSS3
- Ease of use
  - Rotation: `-webkit-transform: rotate(45deg);`
  - Render images rather than fetch:

```
<div id="msBackground"> </div>
```

```
#msBackground {
```

```
  background-image: -webkit-gradient(linear, left top, right top, color-stop(.50, #abe0f9), color-stop(.5, #89d6f6));
```

```
  -webkit-background-size: 38px 42px;
```

```
  width: 100%;
```

```
  height: 100%;
```

```
  z:1;
```

```
}
```

- Animations using WebKit transitions, transforms, and animations

# Animation in CSS - example

```
<!DOCTYPE HTML>

<html>
  <head>
    <title>Tizen Conference Sample App</title>
    <link rel="stylesheet" type="text/css" href="sampleApp.css"></link>
  </head>

  <body>
    <div id="circle" class="circleNormal"> </div>
    <script src="jquery-1.6.2.min.js"></script>
    <script language="Javascript" src="sampleApp.js" type="text/javascript"></script>
  </body>
</html>
```

# Example (contd...)

## CSS Code:

```
#circle {
    height: 150px;
    width: 150px;
    position: absolute;
    top: 50px;
    -webkit-border-radius: 75px;
    -webkit-transition: all 500ms linear;
}

.circleNormal {
    background-color: red;
    left: 50px;
}

.circleClicked {
    background-color: blue;
    left: 500px;
}
```

## Java Script Code:

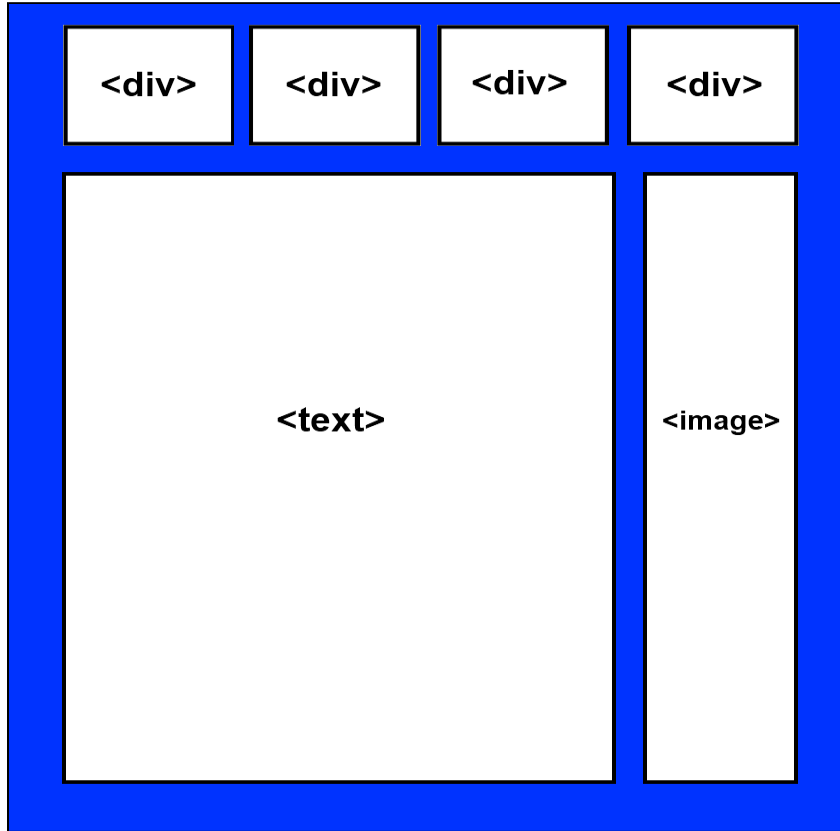
```
function handleCircleClick(event)
{
    $('#circle').removeClass('circleNormal').addClass('circleClicked');
}

$(document).ready(function()
{
    var elem = document.getElementById("circle");
    elem.addEventListener("click", handleCircleClick, false);
});
```

# HTML5 Canvas

- Its Conception and Why was this introduced
- Provides new possibilities for media without the need for plugins
- Many new apps are being made with focus a on canvas
- Allows for different approach for app development

## CSS3



## Canvas





# Canvas Animation Code

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="javascript/animationDemo.js"></script>
    <title>Canvas Animation Demo</title>
  </head>
  <body>

    <canvas id="board" position:absolute; left:0px; top:0px;" width="800" height="600"></canvas>

  </body>
</html>
```

## JavaScript Code

```
var boardCanvas, boardCtx;
var xCoord = 50, yCoord, radius = 50, red = 225, blue = 0;

window.onload = function ()
{
    boardCanvas = document.getElementById("board");
    boardCtx = boardCanvas.getContext("2d");
    yCoord = boardCanvas.height / 2
    boardCanvas.addEventListener('mousedown', onMouseDown);
    drawBall();
}

function onMouseDown(event)
{
    var xDiff = event.clientX - xCoord;
    var yDiff = event.clientY - yCoord;
    if ( (xDiff * xDiff) + (yDiff * yDiff) < (radius * radius) )
        moveBall();
}
```

## JavaScript Code

```
function drawBall()
{
    red -= 1;
    blue += 2;
    boardCtx.clearRect(xCoord - radius - 3, yCoord - radius - 3,
        (radius * 2) + 6, (radius * 2) + 6);
    boardCtx.fillStyle= "rgba(" + red + ", 0 ," + blue + ", 100)";
    boardCtx.beginPath();
    boardCtx.arc(xCoord, yCoord, radius, 0, Math.PI*2, true);
    boardCtx.closePath();
    boardCtx.fill();
}

function moveBall()
{
    drawBall();
    xCoord += 3;
    if (xCoord < 600)
        cancelAnimation =
            window.webkitRequestAnimationFrame(moveBall);
}
```

# Sample WebApps

Currently available as part of Tizen SDK:

<https://developer.tizen.org/resources/sample-web-applications>

- Hanganman

- A word guessing game. Guess the word correctly and save the man!

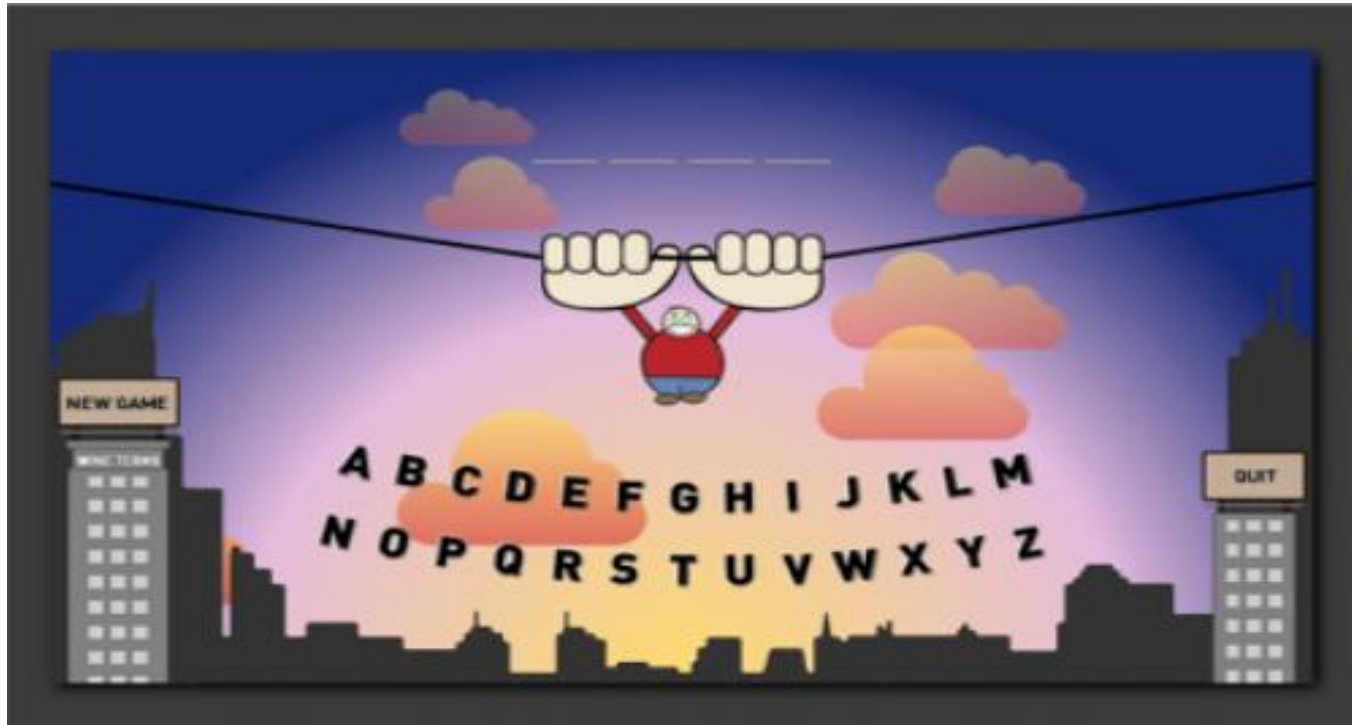
- Annex

- A tile board game. Match your wits against the computer for board domination.

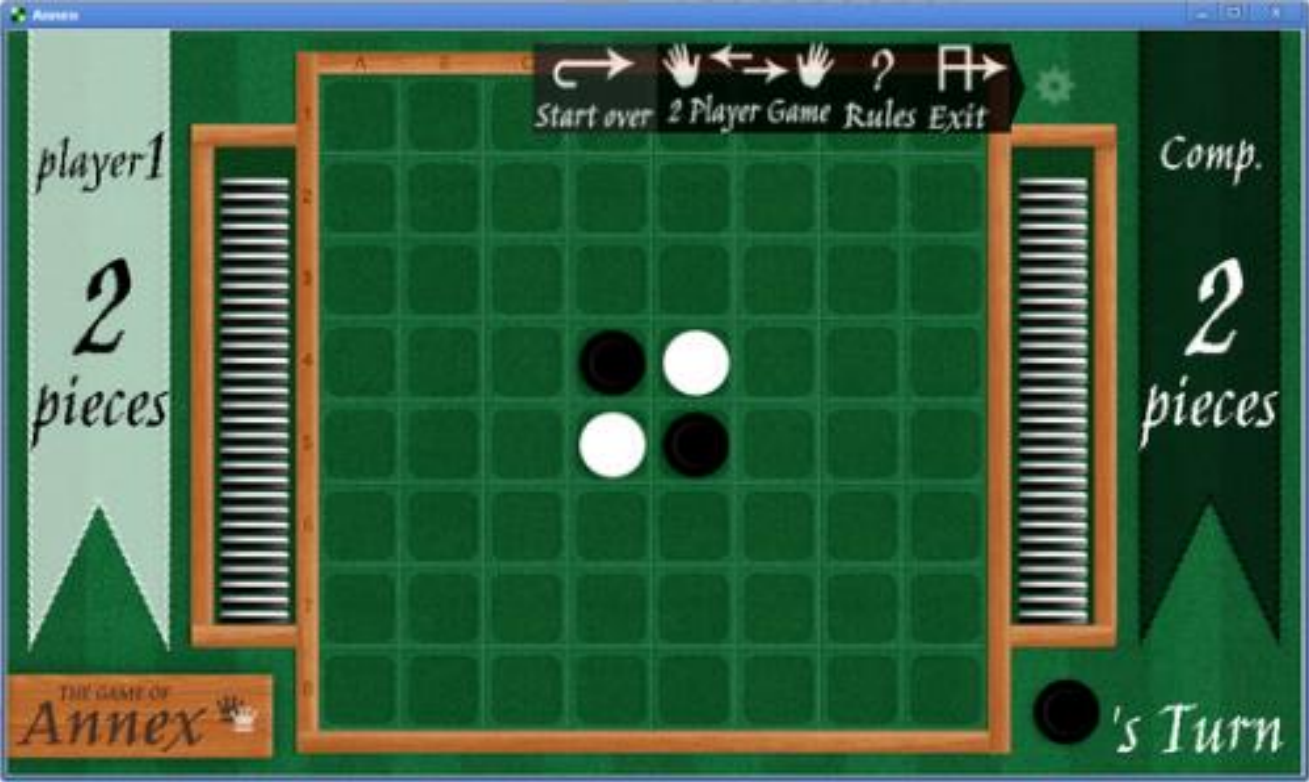
- Countingbeads

- A simple counting game. See if your young kids can count to 50.

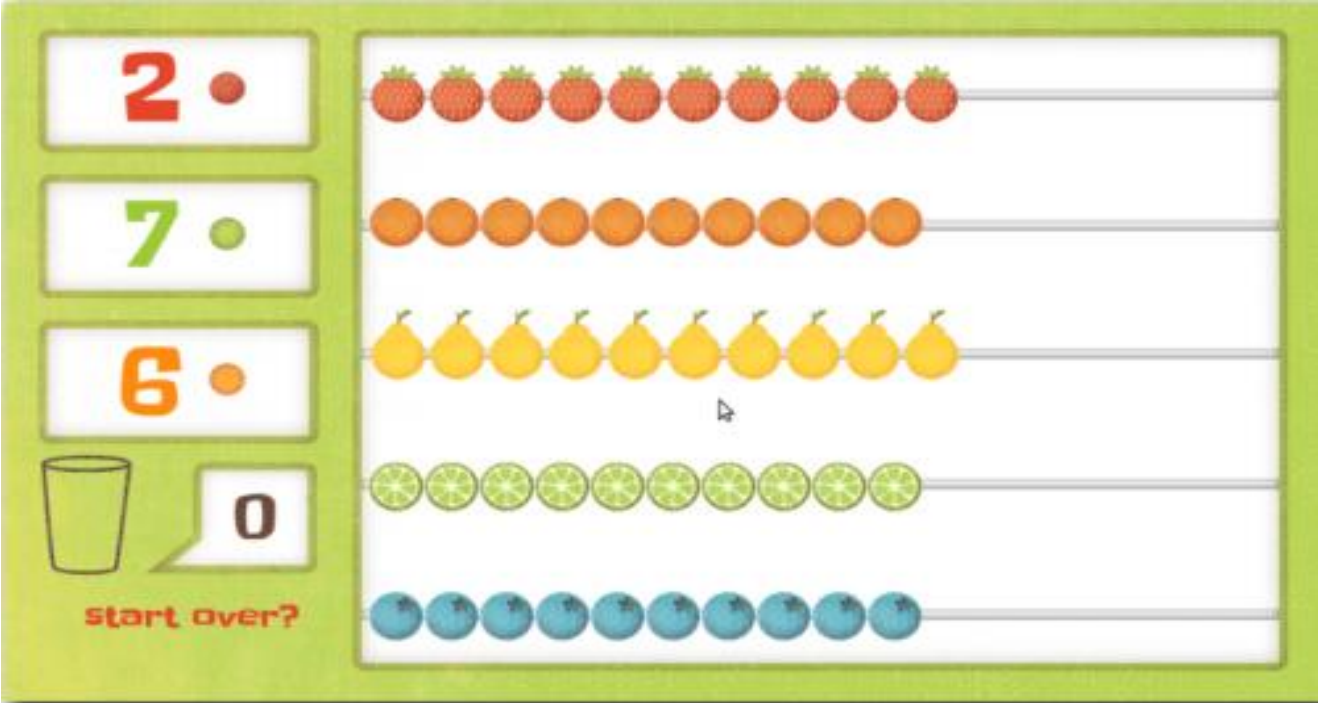
# Hangonman



# Annex



# Countingbeads



# Code complexity by Methodology

App Name	Category	HTML	CSS	JS
Synthesizer	Canvas	47	61	5724
Countingbeads	CSS3	133	710	635
Hanganman	CSS3	160	1350	997
Numeroo	Hybrid	119	235	2410

# Pros & Cons - Canvas

## Pros:

- Simple Dom
- Consistent on all browsers, removing the need for checking what browser is being used.
- Can confine work to a single place (Javascript)
- Multiple canvas objects can be stacked to create depth and layering effects
- Easy to draw lines, curves, and shapes
- Full screen effects and filters
- Absolute pixel manipulation possible, including video
- Almost anything is possible

## Cons:

- It is pixel based and not object based (must manually keep track of object location and events)
- Developer must provide their own animations and effects, which is time intensive
- Smaller user base since it is a newer tech
- Not as optimized as CSS for most browsers



# Pros & Cons – CSS3

## Pros:

- Ease of use, wide user base, heavily documented
- Separates presentation from DOM content
- Code reusability
- Allows the content to be presented in more than one style (ex: portrait, landscape)
  - <link rel="stylesheet" media="all and (orientation:portrait)" href="portrait.css">
  - <link rel="stylesheet" media="all and (orientation:landscape)" href="landscape.css">
- Inbuilt support for animations (Browser Optimized)

# Pros & Cons – CSS3 (continued)

## Cons:

- Dependent on HW acceleration
- Large DOMs will affect readability
- Difficult / impossible to write your own animations
- Limited Image manipulation (using WebKit filters)
- Lack of expression based computations
- CSS3 is not yet a W3C standard though many browsers support many of the new features (CSS3 browser support reference:  
[http://www.w3schools.com/cssref/css3\\_browsersupport.asp](http://www.w3schools.com/cssref/css3_browsersupport.asp))

# In Summary

## Canvas:

- Pixel based api
- Drawing graphs, need total control on your image manipulation.
- Great for large number of objects

## CSS3:

- Quick turnaround
- Ease of use, lots of options inbuilt
- Control at the object level

## Hybrid:

- Leverage power of both. Great if you have a drawing area that changes quickly, but want to quickly throw in the buttons, CSS3 elements.

# Coming soon...

- 11 more apps will be available soon and more pipelined
  - Bubblewrap
  - Flashcards
  - Go
  - Mancala
  - Memory game
  - Memory game (Older kids)
  - Numeroo
  - Run rabbit run
  - Slider puzzle
  - Synthesizer
  - Sweetspot

# Demos

# Backup

# Benchmarks

- Reference:

- <http://blog.frontendforce.com/2010/03/games-development-in-javascript-canvas-vs-dom-benchmark/>
- <http://www.html5rocks.com/en/tutorials/speed/html5/>

- Outcome:

- In Chrome: a lot of elements — DOM loses, large canvas size — Canvas loses.

**TIZEN™** DEVELOPER  
CONFERENCE  
MAY 7-9, 2012