LABORATORIO DI REALTÀ AUMENTATA

Claudio Piciarelli

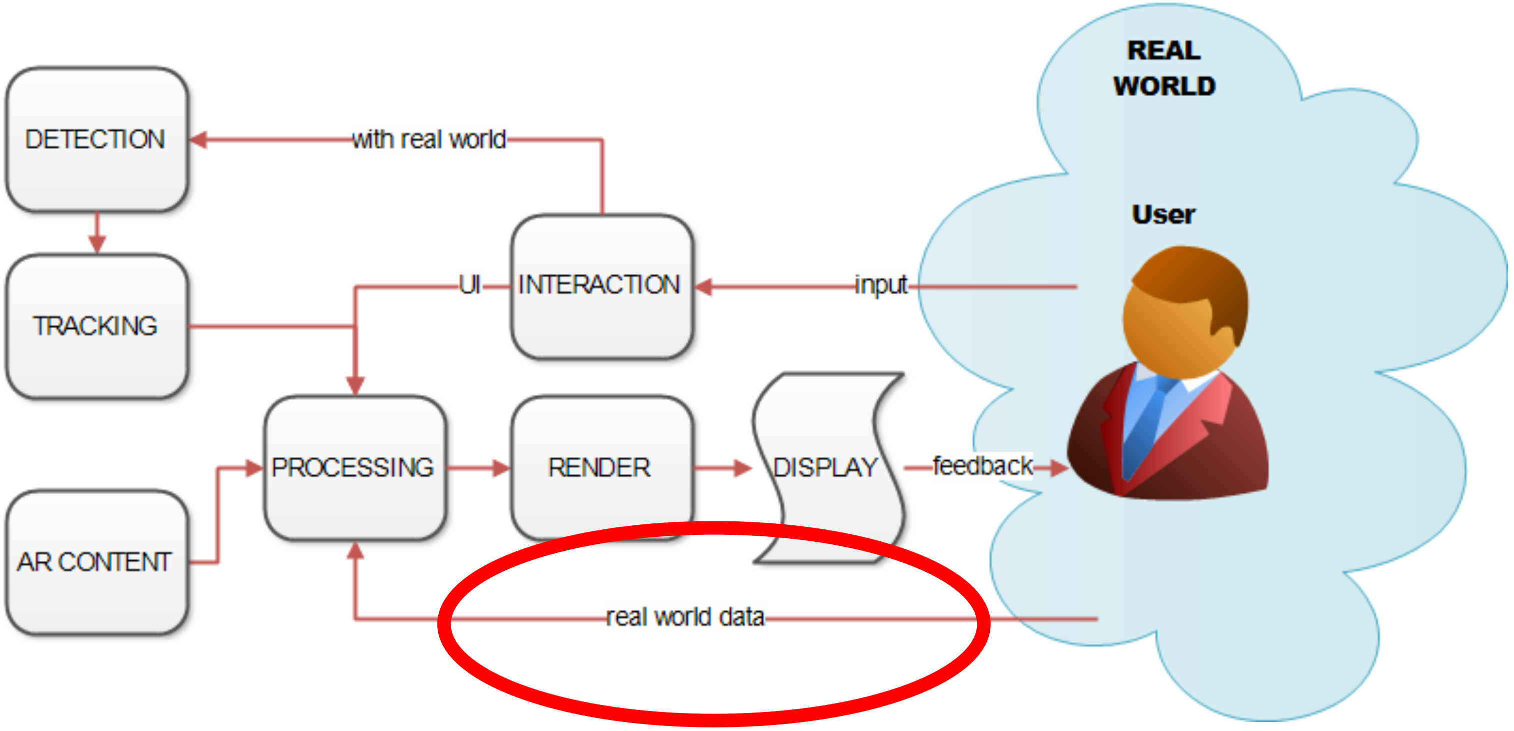
Università degli Studi di Udine
Corso di Laurea in Scienze e Tecnologie Multimediali

Project: useful HTML5 tags

Prerequisites

- What do you need:
  - A WebRTC-enabled browser (I will use Firefox)
    - http://iswebrtcreadyyet.com/
  - Mac users: Safari support still limited
  - A text editor, e.g. Notepad++ (has syntax highlighting and a very basic javascript completion)
  - You can use your preferred editor, though

Architecture of an AR system



STM AR: video input

- Idea: acquire a video stream from a webcam using a standard browser
- If you don't have a webcam, you can work on videos
- Alternative solution: use a webcam simulator software (e.g. manycam)

- Before we start, let's introduce HTML5's **`<canvas>`** and **`<video>`** tags

HTML 5

→ rispetto ad html 4 c'è supporto ai contenuti mediali, nello specifico noi abbiamo usato i tag <canvas>, <video>

- "HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It was finalized, and published, on 28 October 2014 by the World Wide Web Consortium (W3C). This is the fifth revision of the HTML standard since the inception of the World Wide Web. The previous version, HTML 4, was standardized in 1997." – source: wikipedia

- Most relevant innovations versus HTML 4
  - Focuses on semantic content of documents (**`<main>`**, **`<section>`**, **`<article>`**, **`<header>`**, **`<footer>`**, **`<aside>`** …)
  - Javascript APIs for web applications (web storage, web workers…)
  - **Multimedia support (`<canvas>`, `<audio>`, `<video>`, `<svg>`…)**

"tela", immagine su cui posso disegnarci dinamicamente sopra tramite codice JS

Basic structure of an HTML 5 document

No more long, unreadable preambles

```
<!doctype html>
<html>
<head> → robe che non vengono visualizzate
        <meta charset="UTF-8"> → set di caratteri usato nella pagina
        <title>Title</title> → titolo della pagina web
</head>
<body>
        <!-- This is a comment and will be ignored -->
        Hello world!
</body>
</html>
```

The <canvas> tag

- A generic drawing area, with specific APIs for drawing geometric figures, images, etc.
- Just a grid of 4-channels pixels
  - Red, Green, Blue, Alpha
    ↳ trasparenza
- With HTML 5 and canvas, we can get rid of obsolete technologies such as Adobe Flash

CANVAS ci permette di disegnare i contenuti aumentati sopra l'immagine reale

<canvas> definition

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>canvas test</title>
</head>
<body>
    <canvas width="200px" height="200px" id="demo_canvas">
        Display this content if canvas not supported
    </canvas>
</body>
</html>
```

Canvas-specific attributes

Common to all html5 tags

Canvas context

- The context is the js element used to interact with the canvas. Currently only the “2d” context is supported (we will later see how to draw 3d images)
- Javascript code:

```
var canvas = document.getElementById("demo_canvas");
var context = canvas.getContext("2d");
```

↳ il CONTEXT è la modalità con cui accediamo al CANVAS

Where to put javascript code

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script>
        window.onload = function(){
            var canvas = document.getElementById("mycanvas");
            var context = canvas.getContext("2d");
        }
    </script>
</head>
…
```

- This way the code is executed after the whole page has been loaded

Canvas Paths

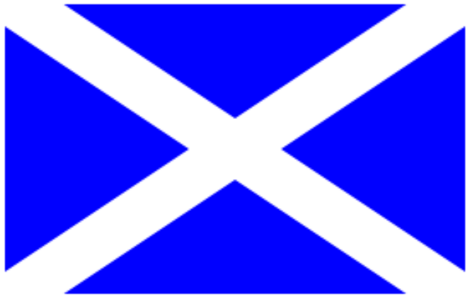
- A path is a **sequence of drawing operations**. Rendering must be explicitly executed (unrendered drawings are lost when a new path starts)

Path start

Drawing commands

Rendering commands

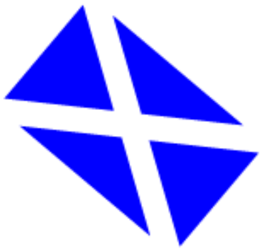
```
context.beginPath();
context.fillStyle = "#0000ff";
context.rect(0,0,300,200); // x,y,width,height
context.strokeStyle = "#ffffff";
context.moveTo(0,0);
context.lineWidth="30";
context.lineTo(300,200);
context.moveTo(300,0);
context.lineTo(0,200);
context.fill();
context.stroke();
```



Canvas transformations

- Modify the way a path is drawn. Influence the next paths

```
context.scale(0.5,0.5);
context.translate(100,50);
context.rotate(0.7);
context.beginPath();
...
```



Understanding BeginPath()

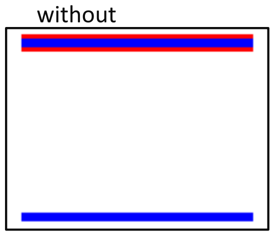
- What does this code do?

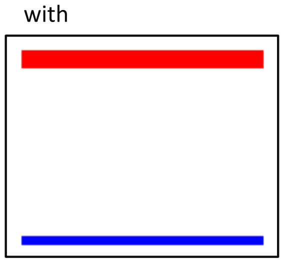
```
context.beginPath();
context.moveTo(100, 100);
context.lineTo(500, 100);
context.lineWidth = 30;
context.strokeStyle = "red";
context.stroke();

context.moveTo(100, 400);
context.lineTo(500, 400);
context.lineWidth = 15;
context.strokeStyle = "blue";
context.stroke();
```

Understanding BeginPath()

- Check the difference with and without BeginPath()

```
context.beginPath();
context.moveTo(100, 100);
context.lineTo(500, 100);
context.lineWidth = 30;
context.strokeStyle = "red";
context.stroke();
context.beginPath();
context.moveTo(100, 400);
context.lineTo(500, 400);
context.lineWidth = 15;
context.strokeStyle = "blue";
context.stroke();
```





Images in canvas

- An image can be easily loaded into a canvas

```
var img = new Image();
img.src = "flower.jpg";
img.onload = function() {
        context.drawImage(img,0,0);
}
```

- (be sure the image is loaded first! Otherwise it will show nothing)

Pixel-wise manipulation

Local: does not work on many browsers for security reasons. You must place your code online.

- Can modify each single pixel

```
img.onload = function() {
    context.drawImage(img,0,0);
    var imgdata = context.getImageData(0,0,img.width,img.height);
    var imgarr = imgdata.data;
    for( var x = 0; x < img.width / 2; x++){
        for( var y = 0; y < img.height; y++){
            var pos = (img.width * y + x) * 4;
            var g = (imgarr[pos]+imgarr[pos+1]+imgarr[pos+2]) / 3;
            imgarr[pos] = imgarr[pos+1] = imgarr[pos+2] = g;
        }
    }
    imgdata.data = imgarr;
    context.putImageData(imgdata,0,0);
}
```

https://avires.dimi.uniud.it/claudio/teach/labar2020/drawcanvas.html

More on canvas

- Canvas API has many more properties and methods
- For a full list, see:
  - http://www.w3schools.com/tags/ref_canvas.asp

HTML5 and videos

- Before HTML5 there were no standards defining how to play videos in web pages
- Many incompatibility problems (e.g. Flash vs. iOS)
- HTML5 offers the **`<video>`** tag
- Standard, and it adds the possibility to modify the video and the control interface, e.g. through CSS

Which video format?

- Many video formats and codecs exist…
- Unfortunately, no single format is supported by all browsers
- The best choice nowadays is the WebM format sponsored by Google
- …Unless you're using Safari (MacOS, iOS), in this case fall back on .mp4 with h264 codec

WebM

- Specifically developed for the web (e.g. YouTube)
- Royalty-free & open source
- Fast decoding for low-power devices
- Based on a subset of the Matroska container
  - VP8/VP9 video codecs
  - Vorbis/Opus audio codecs

- free video converters available online

Using the <video> tag

Adds video controllers (play/stop, mute etc.)

```
<video controls>
      <source src="marker.webm" type="video/webm">
      <source src="marker.mp4" type="video/mp4">
</video>
```

Need to specify file types

Several sources. The browser will choose the best one

<video> properties

- poster (image to show at the beginning)
- autoplay
- loop
- preload = none | metadata | auto
- Muted = true | false

Javascript methods

In the javascript part:

```
<script>
    var video;
    window.onload = function(){
        video = document.getElementById("myvideo");
    }
    myplay = function(){ video.play(); }
    mypause = function(){ video.pause(); }
</script>
```

In the HTML part:

```
<video controls id="myvideo">
    <source src="marker.webm" type="video/webm">
    <source src="marker.mp4" type="video/mp4">
</video>
<menu>
    <button type="button" onclick="myplay()">Play</button>
    <button type="button" onclick="mypause()">Pause</button>
</menu>
```

Javascript properties

Control position and speed

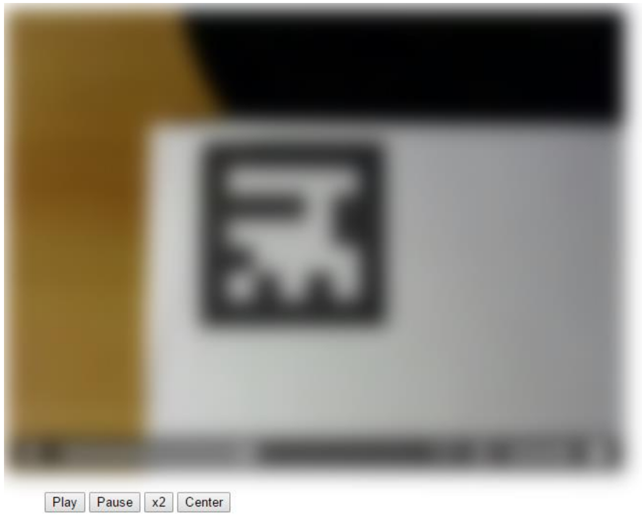
```
mydoublespeed = function(){ video.playbackRate = 2.0; }
mysetcenter = function(){ video.currentTime =
                             Math.round(video.duration / 2); }
```

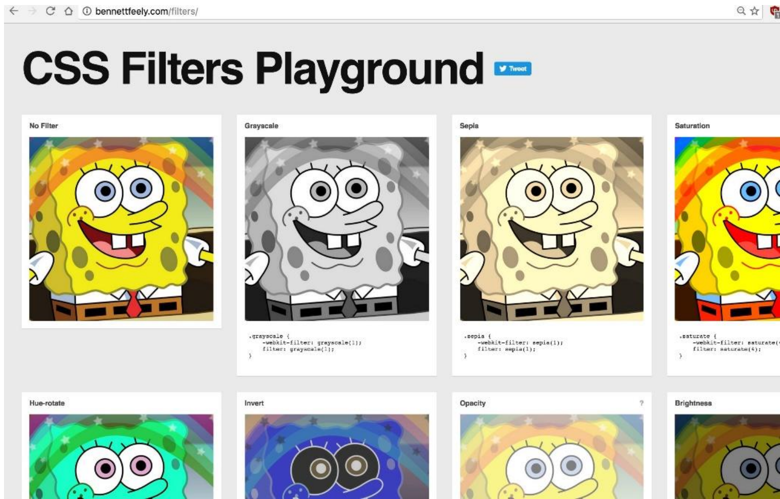
Full list of javascript properties and events (lots of them!):
http://www.w3schools.com/tags/ref_av_dom.asp

CSS filters

- It is even possible to apply CSS filters

```
<style>
        video{
                filter: blur(10px);
        }
</style>
```





```
.grayscale {
    -webkit-filter: grayscale(1);
    filter: grayscale(1);
}
```

```
.sepia {
    -webkit-filter: sepia(1);
    filter: sepia(1);
}
```

```
.saturate {
    -webkit-filter: saturate(
    filter: saturate(4);
}
```

Videos and canvas

```
var video;
window.onload = function(){
    video = document.getElementById("myvideo");
    video.onclick = function(){
        var context =
document.getElementById("mycanvas").getContext("2d");
        context.canvas.width = video.clientWidth;
        context.canvas.height = video.clientHeight;
        context.drawImage(video,0,0);
    }
}
```