Multimedia-Programmierung
Übung 5

Ludwig-Maximilians-Universität München
Sommersemester 2014
Today

- Recap on javascript
- CreateJS, “a suite of javascript libraries & tools for building rich, interactive experiences with HTML5.”
JavaScript - Introduction

- scripting language
- often used for dynamic web applications → client-side scripts
- can be combined with HTML and CSS
- asynchronous communication with web server possible
- many tools and libraries available for JavaScript e.g. jQuery or CreateJS

Useful Links:
- JavaScript API
- W3-Schools Tutorials
- SELFHTML Tutorials
JavaScript - JavaScript and HTML

- embedding JavaScript in HTML directly

```html
<html>
  <head>
    <script>
      // Do something
    </script>
  </head>
  <body>...
</html>
```

- embedding external JavaScript

```html
<script src="myScript.js"></script>
```
JavaScript - Variables

- Declaring variables
  
  ```javascript
  var a, b, c;
  var d = 0;
  ```

- Using variables
  
  ```javascript
  var myString = "text";
  myString = 'some other text'
  myString = 42;
  ```
  
  The type of a variable changes dynamically at runtime depending on its content.
JavaScript - Arrays and Objects

● Arrays

```javascript
var array = [0, 1, "many"];
var first = array[0];
```

● Objects

```javascript
var person = {
  firstName: "MMP",
  lastName: "rocks",
  id: 42
};

var firstName = person["firstName"];  
var lastName = person.lastName;
```

both notations are equivalent
JavaScript - Events and DOM Manipulation

```html
<script>
    function onClickFunction() {
        var par=document.getElementById("p1");
        par.innerHTML="Hello World";
    }
</script>

<button onclick="onClickFunction()">Klick mich</button>
<p id="p1"> </p>
```

- get DOM elements
- manipulate DOM elements
- event handling
JavaScript - Debugging

Use web console in your browser to debug your javascript code:

- Web Console in Firefox (Ctrl + Shift + K)
- JavaScript Console in Chrome (Ctrl + Shift + J)
- Safari (Ctrl + Alt + I)
- Opera (Ctrl + Shift + I)

Use `console.log("my log message")` to output text to the console
JavaScript - Libraries

Popular JavaScript Libraries:

- jQuery
- Modernizr
- D3
- Ext JS
- MooTools
- Prototype
- ASP.NET Ajax
- AngularJS
- YUI Library
- JQuery Mobile
- ...
- EaselJS, TweenJS, SoundJS, PreloadJS

More on the following slides!
CreateJS - Introduction

- “A suite of Javascript libraries & tools for building rich, interactive experiences with HTML5”

- Download, Docs and Demos:
  [http://www.createjs.com](http://www.createjs.com)
CreateJS - Modules

EaselJS  TweenJS  SoundJS  PreloadJS

graphics from www.createjs.com
CreateJS - Getting Started

```html
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>
  <body onload="init()">
    <canvas width="500" height="800" id="canvas">
      Your Browser does not support Canvas</canvas>
  </body>
</html>
```
EaselJS - The canvas is your stage

```javascript
var stage, text;

function init() {
    stage = new createjs.Stage("canvas");
    text = new createjs.Text("Hello World", "50px Arial", "black");
    stage.addChild(text);
    stage.update();
}
```

Output:

```
Hello World
```
EaselJS - Shapes

- Drawing shapes with EaselJS

```javascript
var shape = new createjs.Shape();
shape.graphics.beginFill("green").drawRect(0,0,100,100);
stage.addChild(shape);

var graphics = new createjs.Graphics();
graphics.beginFill("blue").drawRect(50,50,100,100);
var blueRect = new createjs.Shape(graphics);
stage.addChild(blueRect);

stage.update();
```

use shape objects to display and transform graphics

use this approach to share the same graphics
EaselJS - Shape transformations

- Transforming Shapes

```
shape.x = 50;
shape.scaleX = 1;
shape.regX = 50;
shape.regY = 50;
shape.rotation = 180;
```

- use negative scaling values to flip an object
- use the reg attribute to set an object's center (i.e. for later rotation)
EaselJS - Drawing and Images

● Method chaining with graphics

```javascript
var shape = new createjs.Shape();
shape.graphics.beginFill("black").moveTo(50,50)
  .lineTo(100,100).lineTo(100,50).lineTo(50,100);
shape.graphics.closePath();
```

● Working with images

```javascript
var bitmap = new createjs.Bitmap("filename.jpg");
var bitmap_copy=bitmap.clone();
var blurFilter = new createjs.BlurFilter(32, 16, 2);
bitmap.filters=[blurFilter];
bitmap.cache(0,0,100,100);
```
EaselJS - Event Handling

- Shape events
  - `mousedown, mouseup, mouseout, mousein, pressmove, pressup, click, dblclick, rollout, rollover, tick...`

- Stage events
  - `stagemousedown, stagemousemove, stagemouseup, drawnd, drawstart...`

Get mouse position for stage event with `event.stageX` and `event.stageY`
PreloadJS - Introduction & LoadQueue

● Used for preloading images and get real-time progress information

● Preloading resources

```javascript
var manifest = [
    {id: "id_image_1", src: "loremipsum.png" },
    {id: "id_sound_1", src: "music.mp3" }
];
var queue = new createjs.LoadQueue(false);
queue.installPlugin(createjs.Sound);
queue.loadManifest(manifest);
```

 Plugins need to be installed before loading resources that need them

list of resources to preload
PreloadJS - LoadQueue

- Queue Events

```javascript
queue.addEventListener("complete", handleComplete);
queue.addEventListener("error", handleError);
...
```

- Retrieving Resources

```javascript
var image = queue.getResult("id_image_1");
var bitmap = new createjs.Bitmap(image);
```

start your main application here

create a bitmap from your image to e.g. draw it on the stage
Simple Animations with EaselJS

How can we move the red circle from left to right?

- Increase x-position of circle continuously until the circle hits the wall

```javascript
var circle, stage, radius=42;
...

function moveCircleToRight(){
    circle.x += 1;
    if (circle.x > stage.canvas.width - radius)
        { circle.x = 0; }
}
```

reset x position when circle hits wall
Simple Animations with EaselJS II

- How often should `moveCircleToRight` be called?
- Use a ticker to call a function with each tick

```javascript
createjs.Ticker.addEventListener("tick", moveCircleToRight);
createjs.Ticker.setFPS(40); // set max frames per second
createjs.Ticker.setInterval(25); // OR set min ms between frames
```
Advanced Animation with EaselJS

How can we move the circle in any direction?

- Use a vector to determine the direction

  ```javascript
  var start = {x:0,y:0};
  var end = {x:50,y:100};
  var tempDir = {x:(end.x - start.x),
                  y:(end.y - start.y)};
  ```

- Normalize the direction vector

  ```javascript
  var magnitude = Math.sqrt(tempDir.x * tempDir.x +
                             tempDir.y * tempDir.y);
  var dir = {x:(tempDir.x/magnitude),
             y:(tempDir.y/magnitude)};
  ```
Advanced Animation with EaselJS II

- Define a speed for your circle to move with
  
  ```javascript
  var speed = 42;
  ```

- Calculate the new position
  
  ```javascript
  var newX = circle.x + dir.x * speed;
  var newY = circle.y + dir.y * speed;
  
circle.x = Math.round(newX);
  circle.y = Math.round(newY);
  ```

  - exact values for further calculation
  - rounded values for the screen output
Advanced Animation with EaselJS III

- Problem: the ticker is not absolutely reliable
- Take the time between two ticks in account to get realistic behaviour

```javascript
var moved_distance = event.delta/1000*speed;

var newX = circle.x + dir.x * moved_distance;
var newY = circle.y + dir.y * moved_distance;

circle.x = Math.round(newX);
circle.y = Math.round(newY);
```

moved_distance returns the millis since the last tick

the measurement unit of speed changes from px/tick to px/s!
Animation with TweenJS

- Use tweens to animate properties

```javascript
createjs.Ticker.addEventListener("tick",stage);
createjs.Tween.get(circle)
  .wait(500)
  .to({x:50,y:100},1000)
  .call(handleComplete);
```

- Loop and ease your animation

```javascript
createjs.Tween.get(circle,{loop:true})
  .wait(500)
  .to({x:50,y:100},1000,createjs.Ease.cubicInOut);
```

More: [http://www.createjs.com/#!/TweenJS/demos/sparkTable](http://www.createjs.com/#!/TweenJS/demos/sparkTable)
Animation with TweenJS II

- Further Properties to add to your tween
  
  *loop, useTicks, override, paused …*

- Combine multiple tweens

  ```javascript
  var tween1 = createjs.Tween.get(circle,{paused:true})
  .to({alpha:0},1000);  
  
  createjs.Tween.get(circle,{useTicks:true})
  .to({x:50,y:100},25)
  .play(tween1);
  ```

  **create a paused tween**
  **use ticks instead of milliseconds**
  **play the tween specified before**
Useful Links

- CreateJS API: http://www.createjs.com/Docs/


- Tutorial Easel JS and Mouse Interaction: http://www.createjs.com/tutorials/Mouse%20Interaction/


- CreateJS on Github (including sources for all examples, tutorials and demos): https://github.com/CreateJS