



ICT in SES

Textures

Lesson №21

Introduction to textures



What is a texture?

- Structure (most often on the surface)
- Structure (felt when touching)

In computer graphics

- Image mapped onto object
- Retains the shape of the object
- A pixel of the texture is called *texel*

Using



Material structure

- Wooden floor, scratched metal plate, damaged skin, etc.

Repeated elements

- Brick wall, bathroom tiles, building windows, etc.

Background images

- Sky with clouds, stars, maps, etc.

Structure and coordinates



Shape

- Textures are always rectangular

Coordinates

- Own coordinate system
- The image in the own coordinate system is from $(0,0)$ to $(1,1)$



POT/NPOT texture

- For full functionality textures sizes in pixels must be $2^n \times 2^m$
- Otherwise only the base functionality is supported
- POT-texture (*power-of-two*) NPOT-texture (*non-POT*)



POT 512 x 256



NPOT 640 x 359



POT 256 x 256



Texture loading

Attempt at using a texture



Problem

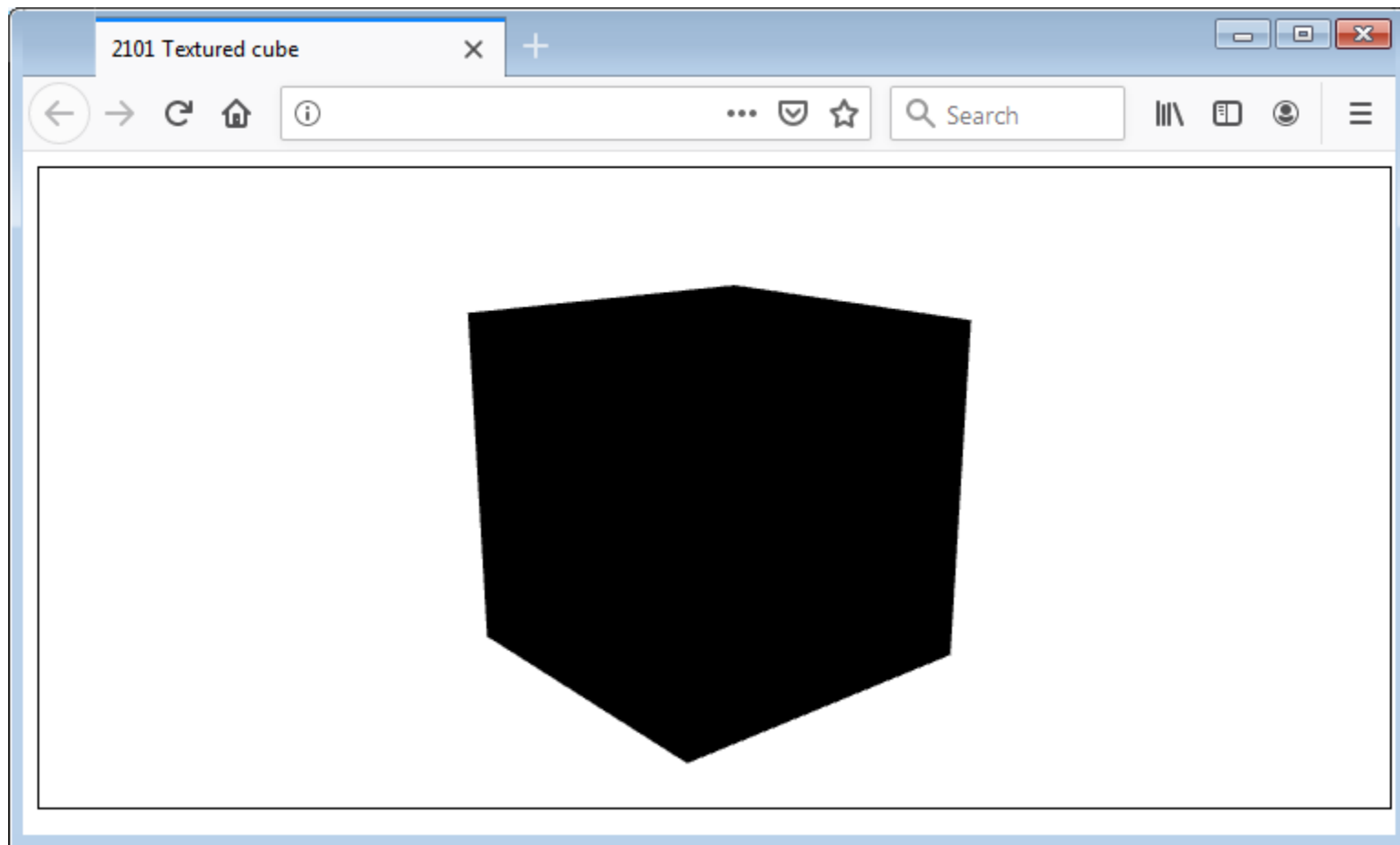
- Create a cube
- Add a texture to it
- Use the same image for a texture, but resized to 128x128 pixels



Solution

- 2D and 3D objects in Suica have property **image**
- It is object with the image of a texture and its properties
- Creating a texture object
- Using class **Image** with parameter the file name of the image

```
a = cube([0,0,0],30);  
a.image = new Suica.Image('texture.jpg');
```



TRY IT

Problem



Problem with texture

- Instead of an image, the cube is completely black
- This is browser protection for loading local files

CORS and SOP

- Cross-origin Resource Sharing
- Single-origin Policy
- Mechanism to control how resources from one location can be used in a page from another location

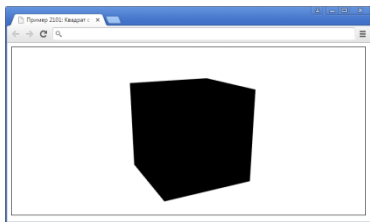
Explanation

- In some browsers loading local files as textures is considered as CORS violation
- Firefox error message in the JS console:
Error: WebGL warning: texImage: Cross-origin elements require CORS.
SecurityError: The operation is insecure.
- Chrome error message in the JS console:
Uncaught SecurityError: The cross-origin image may not be loaded.

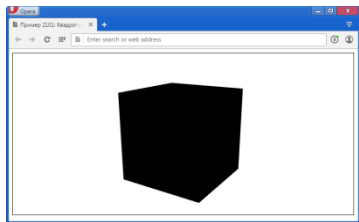
Other tests

- Different browsers, different setting, variable results

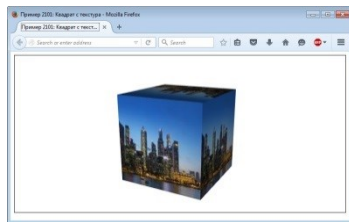
Chrome



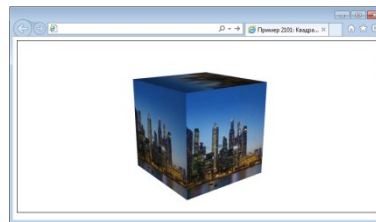
Opera



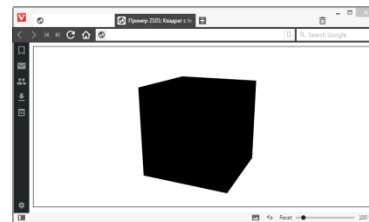
Firefox



IE



Vivaldi



Possible solutions



Another browser (not recommended)

- Using a browser without CORS protection
- No guarantee that browsers of clients computer will react the same way

Another configuration (not recommended)

- Some browsers allow configuration of CORS protection (including disabling it)
- This reduces the level of security, especially if the browser is left in non-CORS-ed state

Recommended solutions



Solution №1 – online

- The program and the image are online, in the same domain
- Advantages
 - This is the default state of any mobile application
- Disadvantage
 - Needs hosting

Solution №2 – embedded images

- Using Data URI
- A resource is embedded in a web address
- Advantages
 - CORS is not violated
 - Images are in the HTML file
- Disadvantages
 - There is a maximal supported length for URL
 - Images are not buffered / cached

Example

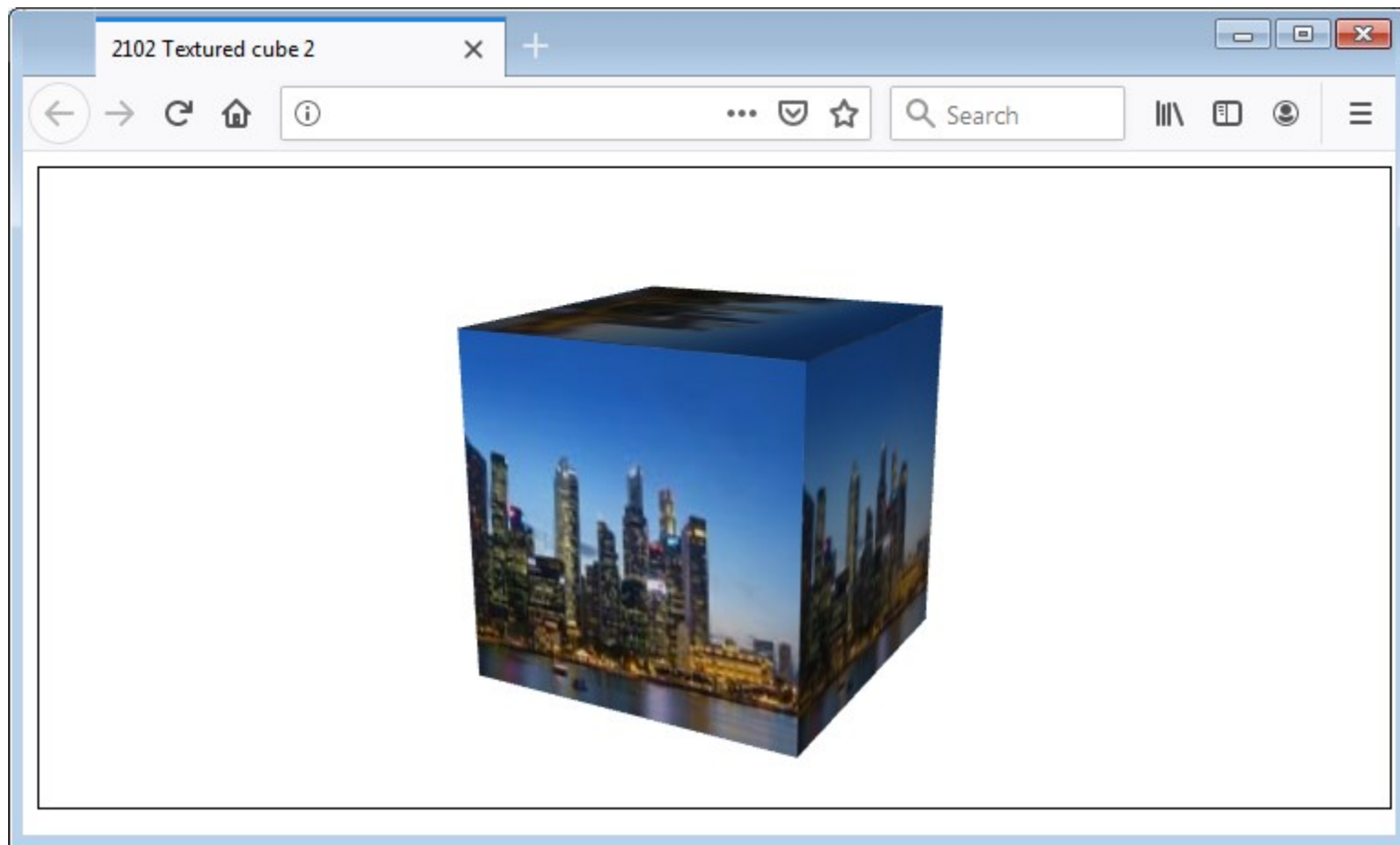
- Using online convertor to Data URI

This is one of the many convertors:

websemantics.co.uk/online_tools/image_to_data_uri_convertor

- The URL is used as image URL

```
a.image = new Suica.Image('data:image/jpeg;base64,
/9j/4AAQSkZJRgABAgEASABIAAD/4RIjRXhpZgAATU0AKgAAAA
gABwESAAMAAAABAAEAAAEaAAUAAAABAAAAYgEbAAUAAAABAAA
agEoAAMAAAABAAIAAAExAAIAAAAbAAAAcG EyAAIAAAAUAAAAjY
dpAAQAAAAABAAAApAAAAANAAAABIAAAAAQAAAEgAAAAABQWRvYmUg
UGhvdG9zaG9wIENTIFdpbmRvd3MAMjAxNToxMDoyOCANzozNz
:
CuV6Uwsb+6hw8PvpuHbS24cq4lv3Gu81NqPD7BXPt91f/9k='');
```

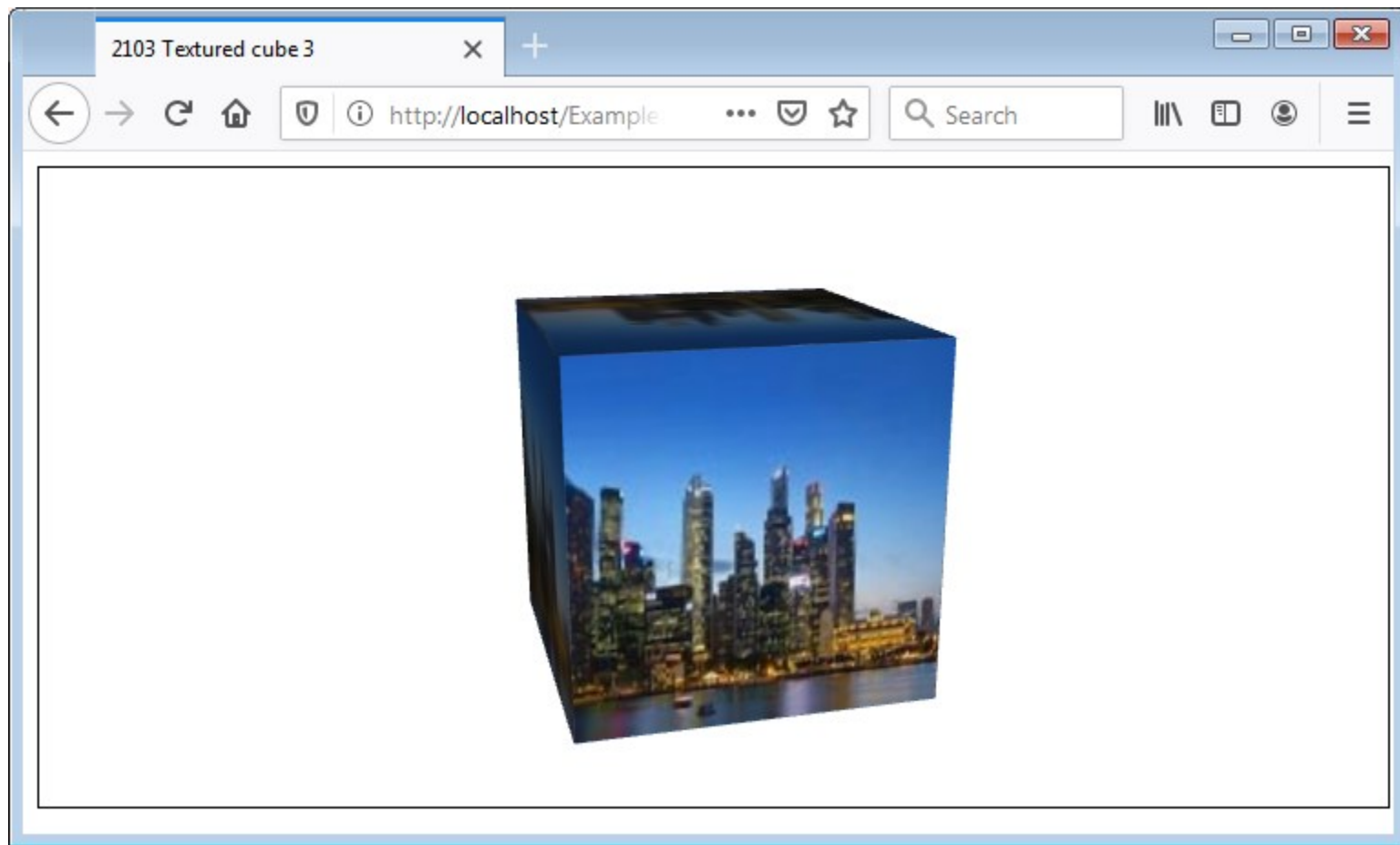


TRY IT

Solution N°3 – local web server

- Simulating online access (http or https)
- Many small servers like: QuickPHP, Mongoose and others
- Instead of adress `file://texture.jpg` the texture uses `http://localhost/texture.jpg` or just `texture.jpg`

```
a.image = new Suica.Image('http://localhost/Example-2103%20Textured%20cube%203/texture.jpg');
```



TRY IT

Instructions



Using QuickPHP Web Server

- Start **QuickPHP.exe**
- In options set **binding address** to **127.0.0.1**, **port** to **80**, **allow directory listing** to be turned on, and **root folder** to the working folder
- Press [Start]
- The folder is now accessible at address **http://localhost**
- When testing is done, the server must be stopped

Using Mongoose Web Server

- Start `mongoose-free-5.6.exe`
- In options set `document_root` to the working folder, and `listening_port` to 80
- The folder is now accessible at address `http://localhost`
- When testing is done, the server must be stopped

Using another web server

- Follow its instructions

Seamless textures

Seamless



Seamless texture

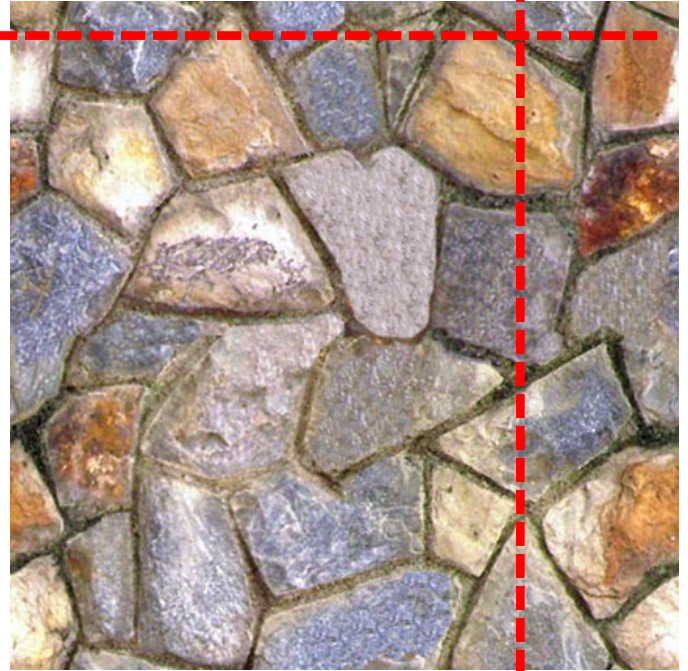
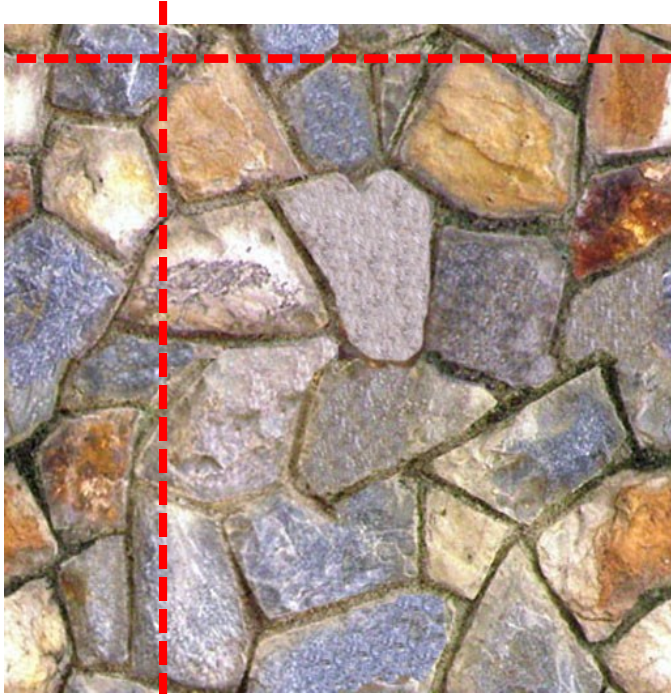
- Texture glued to itself, no visible seam
- Continuous pattern

Usage

- Large unipatterned surfaces
- Walls, driveways, ground, floor tiling, meadows, etc.

How it is done

- Textural continuity





Using textures

Examples



Example №1: Brick wall

- Texture featuring a fragment of a brick wall
- Seamless texture



Seamless Brick 09-28-1001

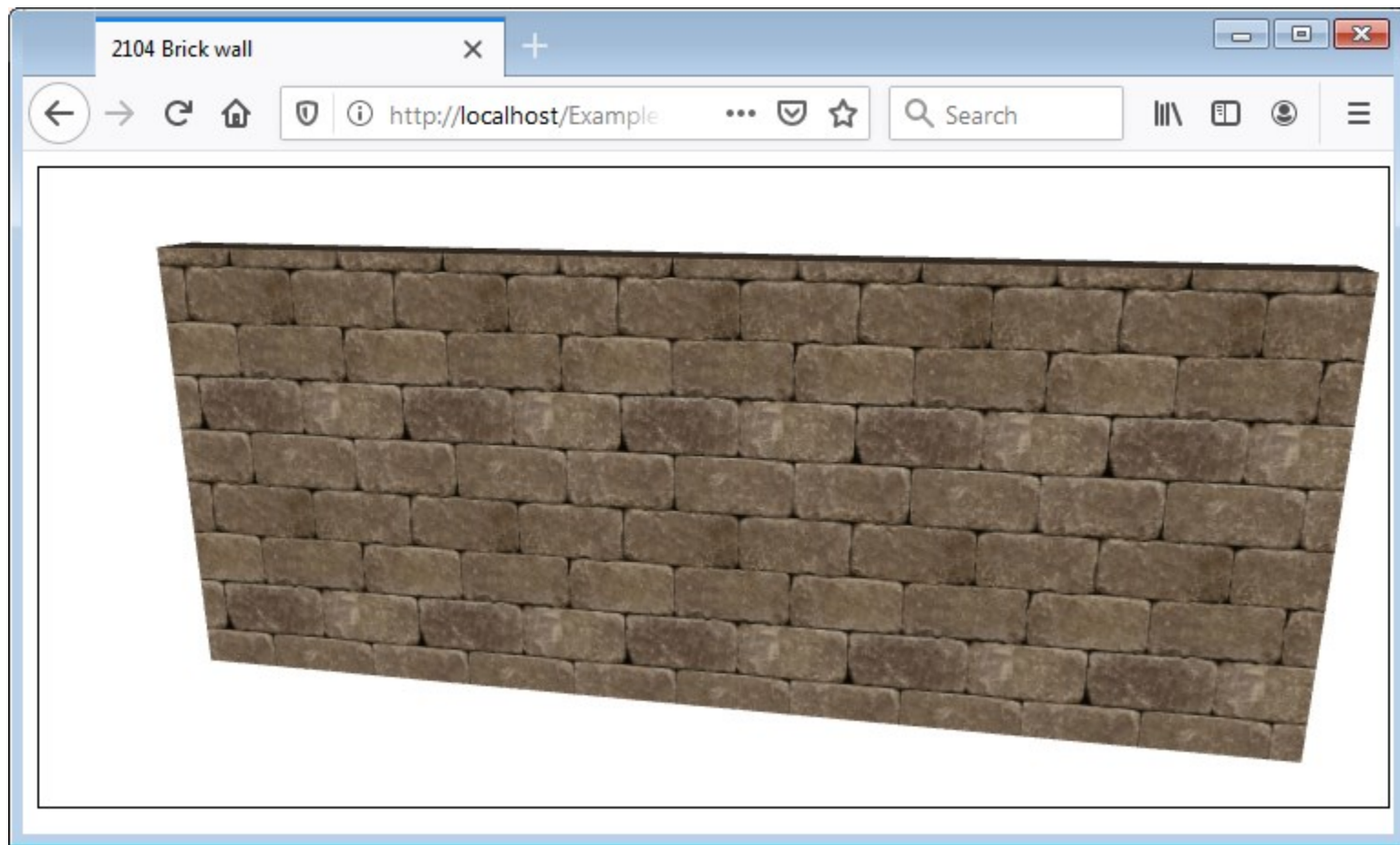
Author: texturemate, license: Public Domain

<http://texturemate.com/content/free-texture-seamless-brick-09-28-1001>

Implementation

- Instead of constructing many textured cubes, there is one textured cuboid of size **100 x 40 x 5**
- Property **image** has subproperty **scale**, defining how many copies of the texture to make; in this case – **5** horizontal and **2** vertical copies
- No need to write **localhost**, the image is in the same web folder as the HTML file, there is no CORS

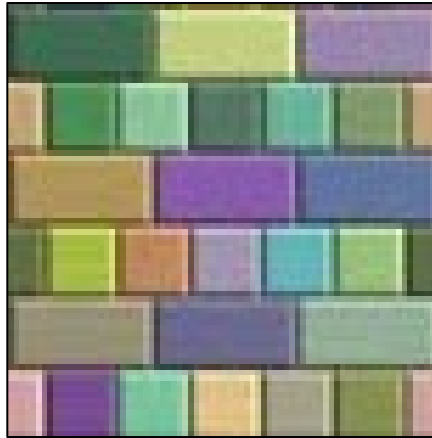
```
a = cuboid([0,0,0],[100,5,40]);  
a.image = new Suica.Image('bricks.jpg');  
a.image.scale = [5,2];
```



TRY IT

Example Nº2: Floor tiling

- Texture of floor tiles
- Horizontal floor extending to the horizon
- At the pattern is small, the texture may appear blurred



Colourful Wooden Blocks Background Texture

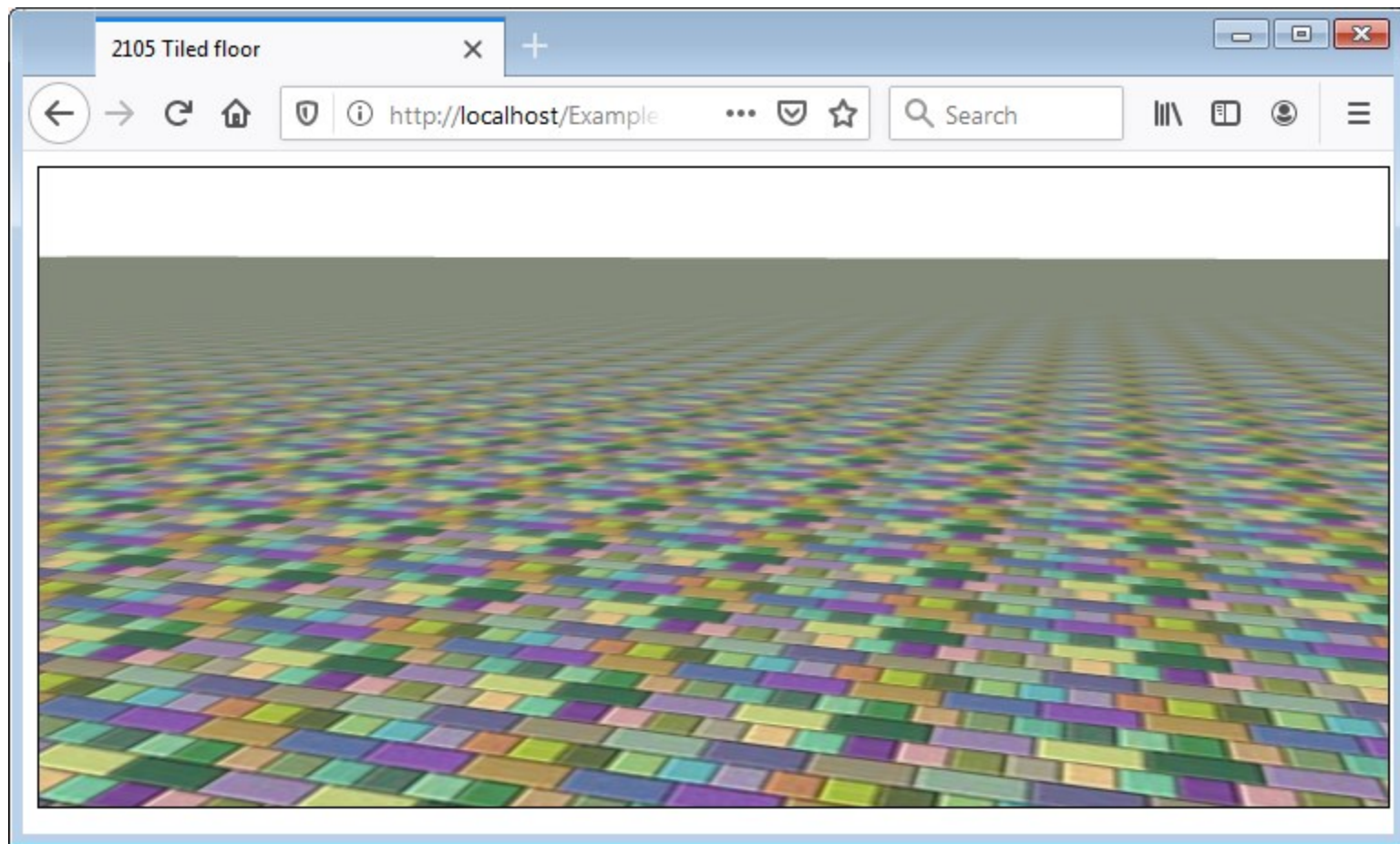
Author: www.myfreetextures.com, license: CC-BY

<http://www.myfreetextures.com/colourful-wooden-blocks-background-texture/>

Implementation

- Creating very large floor 5000 x 5000
- Repeating texture 400 time in each direction
- Because of low view point the floor is poorly lit and too dark
- Recovering brightness by turning the lighting off

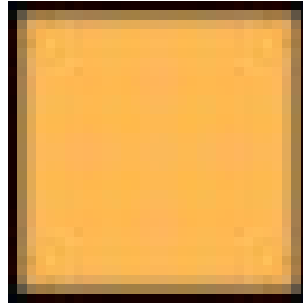
```
a = cuboid([0,0,0],[5000,5000,1]);  
a.light = false;  
a.image = new Suica.Image('tiles.jpg');  
a.image.scale = [400,400];
```



TRY IT

Example №3: Cube of cubes

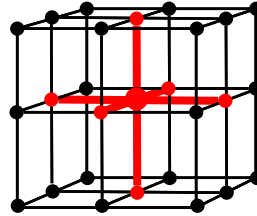
- Making a punched cube with textured walls
- Cube made of standard cubes
- Changing the visible count of cube without changing the number of objects



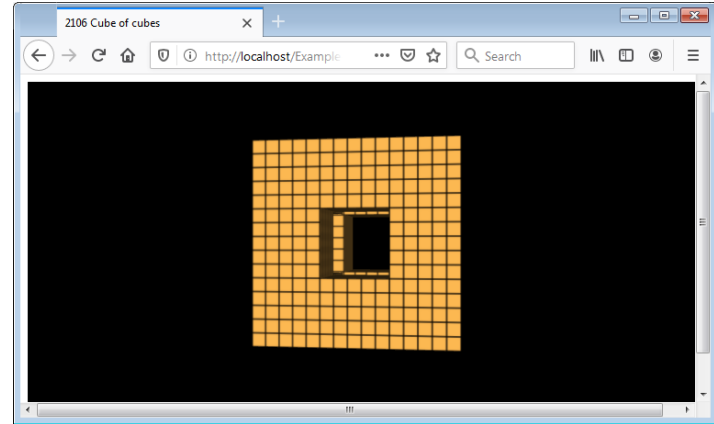
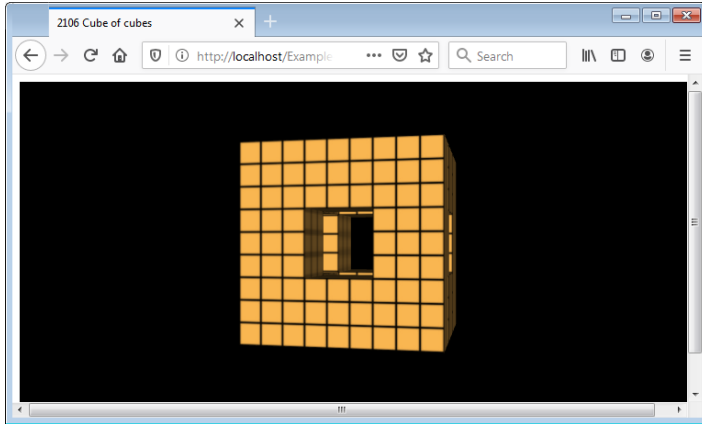
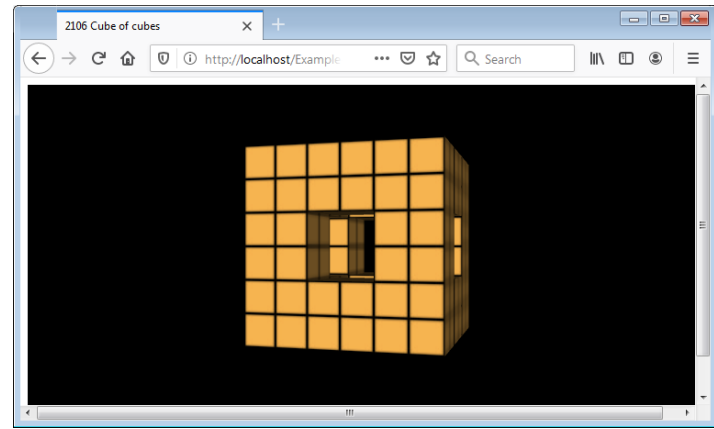
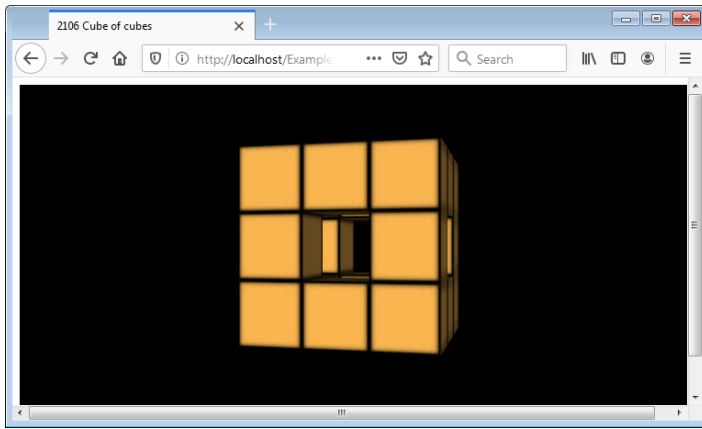
Texture used for
each cube side

Implementation

- Drawing cubes in 3x3x3 cube, without the central cubes
- Skipping the 7 cubes at a distance 1 or less to the center
- Texture loaded once



```
style = {image: new Suica.Image('block.jpg')};  
for (var x=-1; x<2; x++)  
  for (var y=-1; y<2; y++)  
    for (var z=-1; z<2; z++)  
      if (x*x+y*y+z*z>1.1)  
        cube([x,y,z],1).custom(style);
```



TRY IT

Multiple textures

Multiple textures



Example

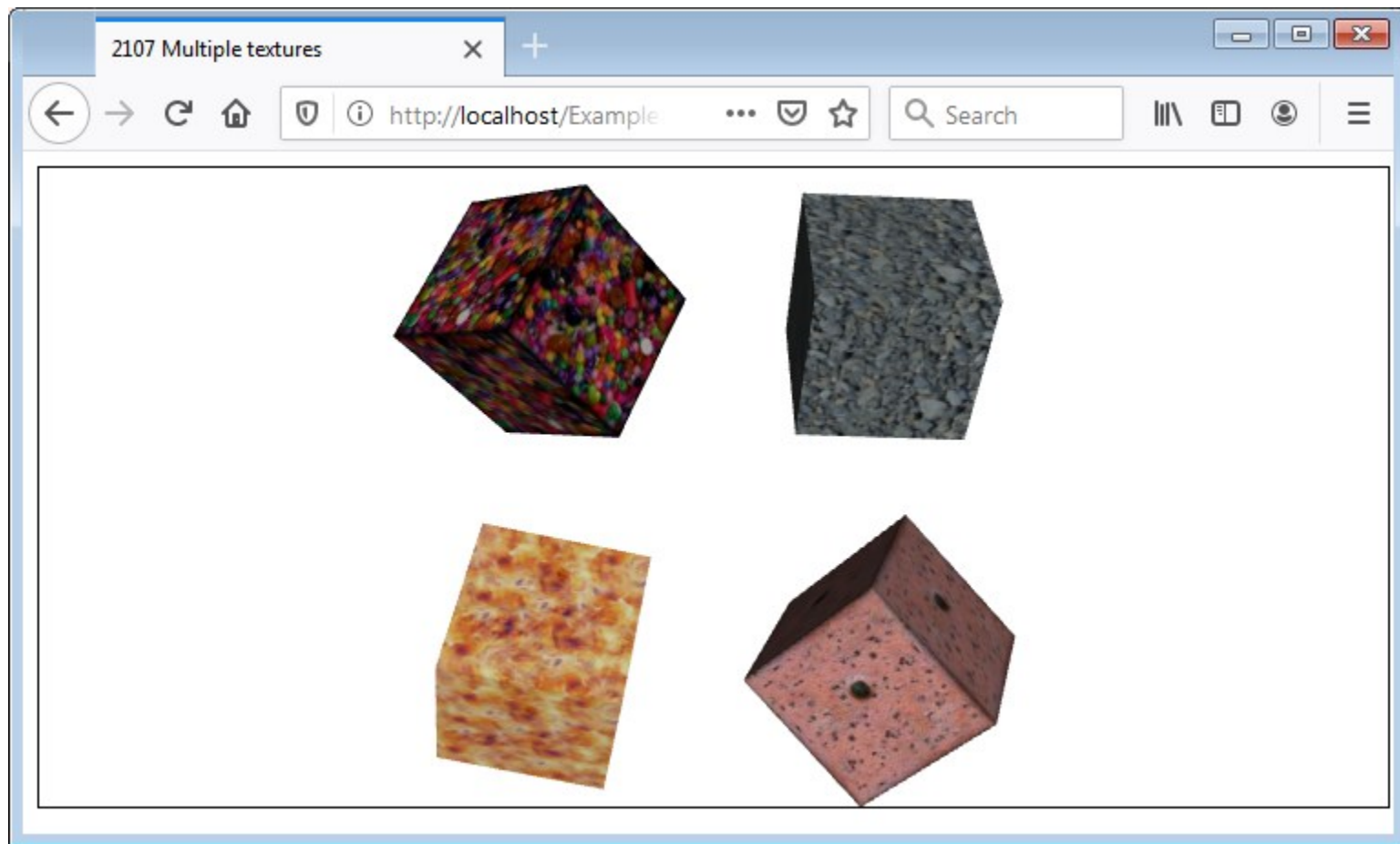
- Drawing several cubes
- Each with its own texture



Implementation

- Every cube as own instance of **Image**

```
a = [];  
  
a[0] = cube([+20,+20,0],20).custom(  
    {image: new Suica.Image('texture1.jpg')});  
a[1] = cube([+20,-20,0],20).custom(  
    {image: new Suica.Image('texture2.jpg')});  
a[2] = cube([-20,+20,0],20).custom(  
    {image: new Suica.Image('texture3.jpg')});  
a[3] = cube([-20,-20,0],20).custom(  
    {image: new Suica.Image('texture4.jpg')});
```



TRY IT

Changing texture



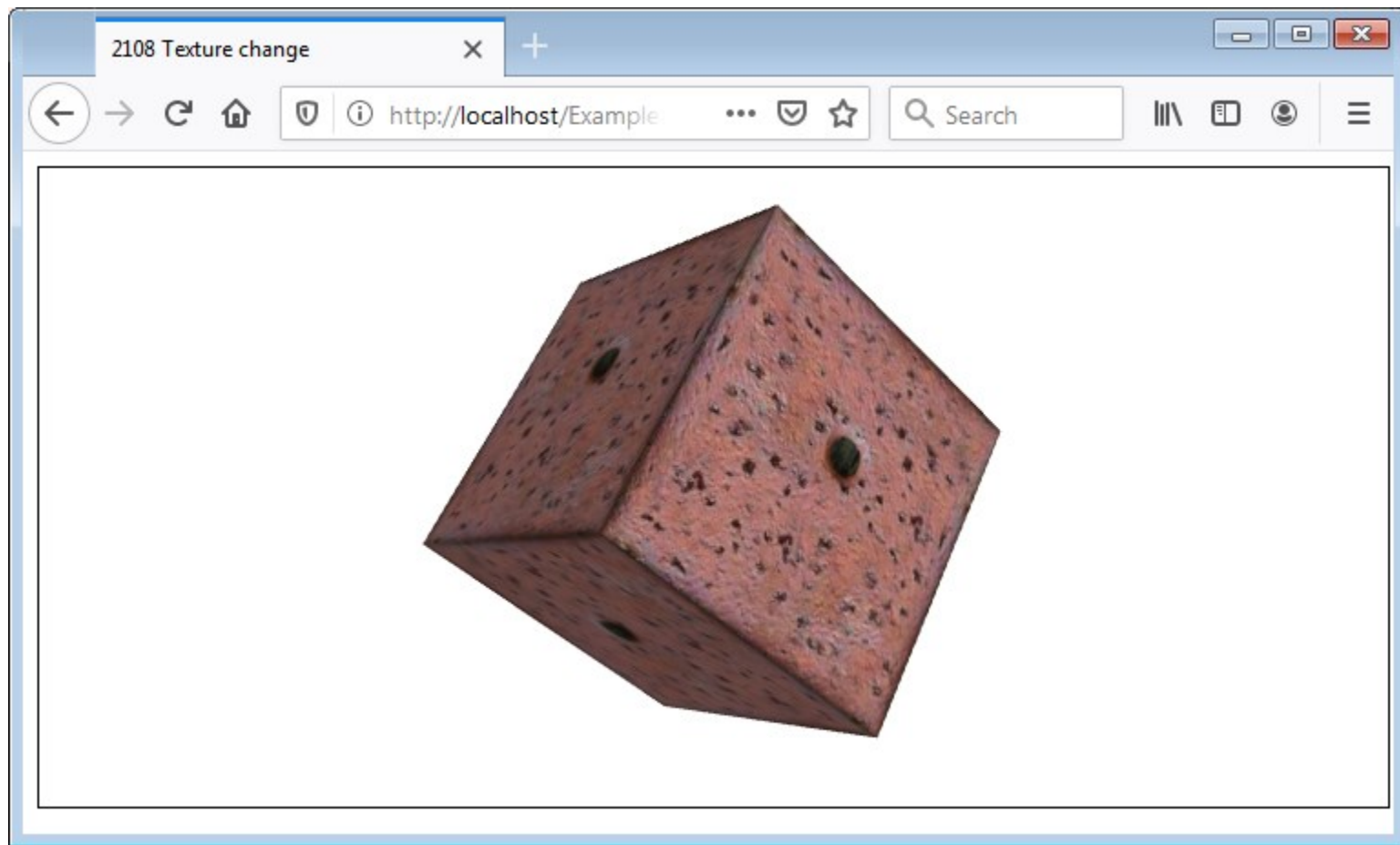
New problem

- Several texture and only one cube
- The cube changes its texture

Implementation

- Loading 4 texture in array
- Changing cube's **image** property in the animation loop

```
image = [];  
for (var i=0; i<4; i++)  
    image[i]=new Suica.Image('texture'+(i+1)+'.jpg');  
:  
function rotate()  
{  
    a.spin = Suica.time;  
    a.focus = [Math.sin(Suica.time),...];  
    a.image = image[Math.floor(Suica.time)%4];  
}
```



TRY IT

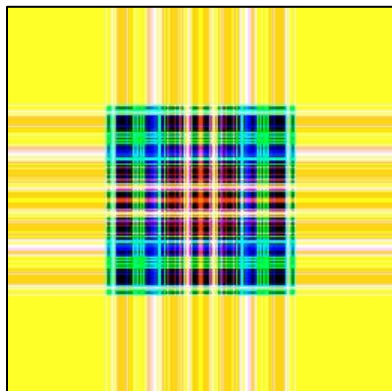
Other properties

Texture shifting



Property **offset**

- Defines texture shifting
- Together with **scale** define what part of the texture is mapped

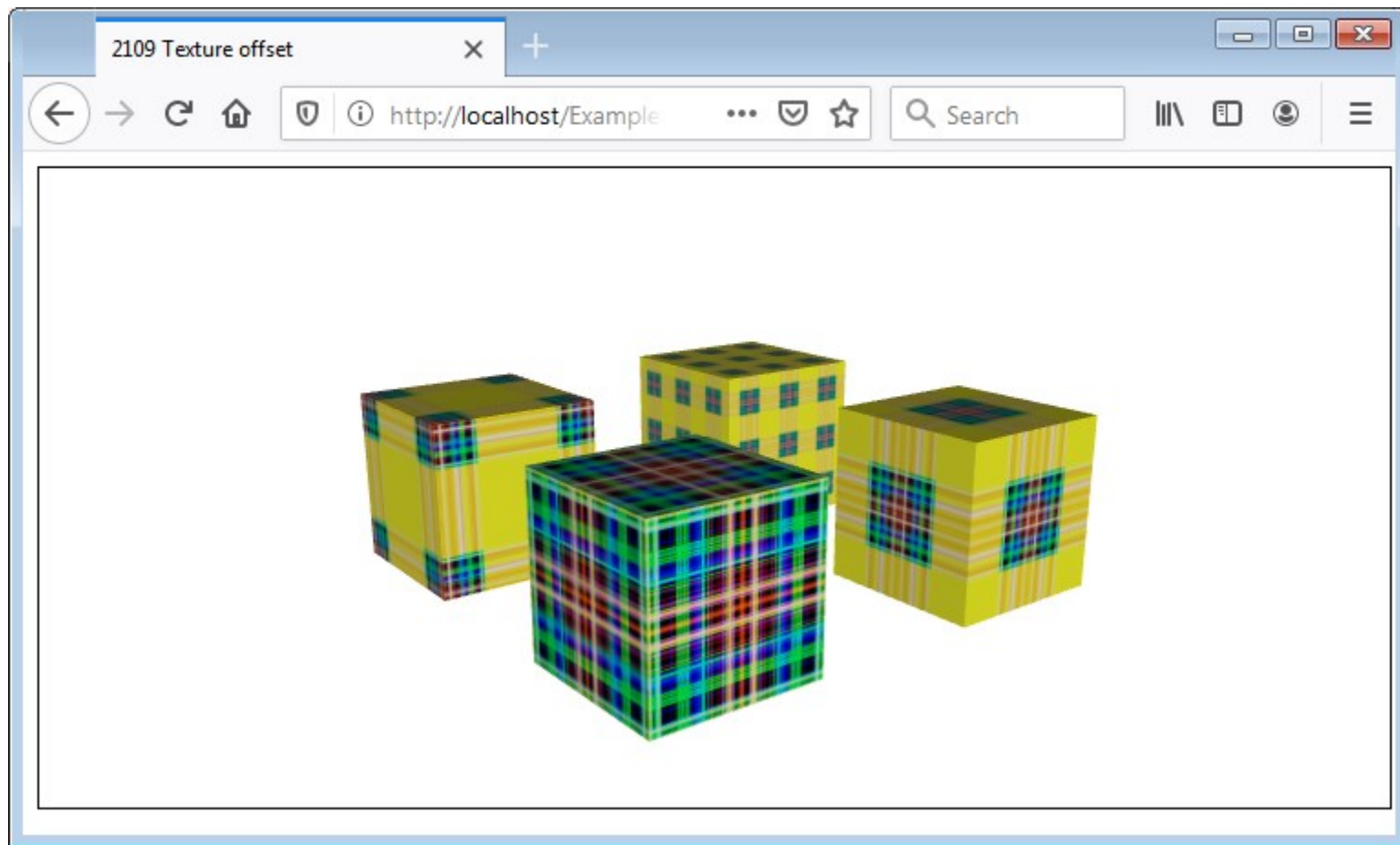


Examples

- Four combinations of **scale** and **offset** (source shows two)
- For each combination there is a new texture instance

```
a = cube([-15,15,0],15);  
a.image = new Suica.Image('crosshatch.jpg');  
a.image.scale = [3,3];  
a.image.offset = [0,0];
```

```
a = cube([15,-15,0],15);  
a.image = new Suica.Image('crosshatch.jpg');  
a.image.scale = [1/2,1/2];  
a.image.offset = [0.75,0.75];
```



TRY IT

Scale of bases



Property **baseScale**

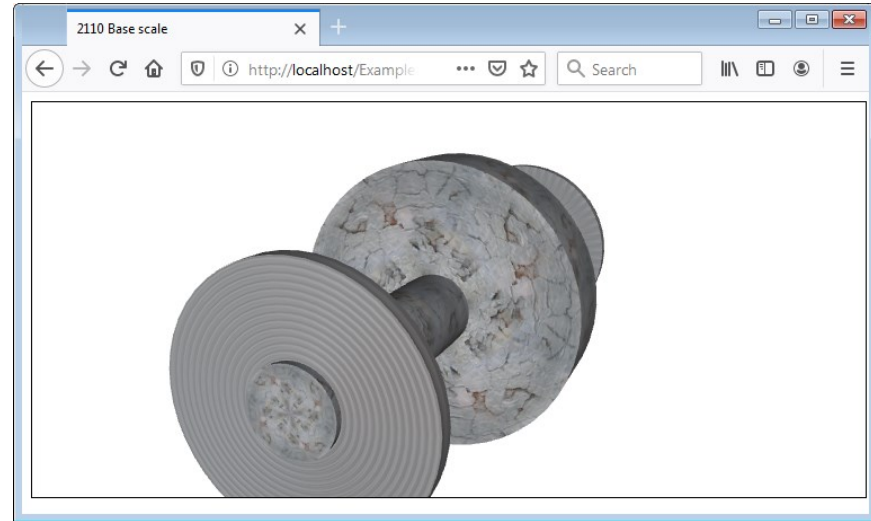
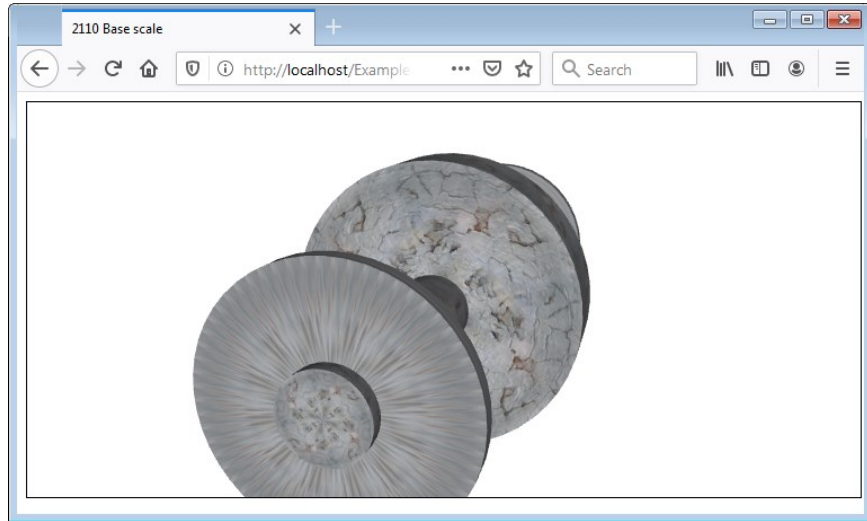
- Objects may have separate texture scaling for their bases
- Used for various purposes

Different size of patterns

Concentric patterns

Radial patterns





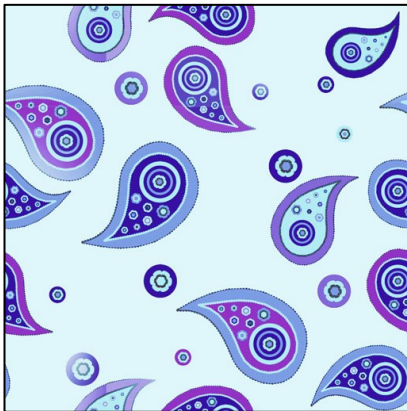
TRY IT

Texture on a sphere



General problem

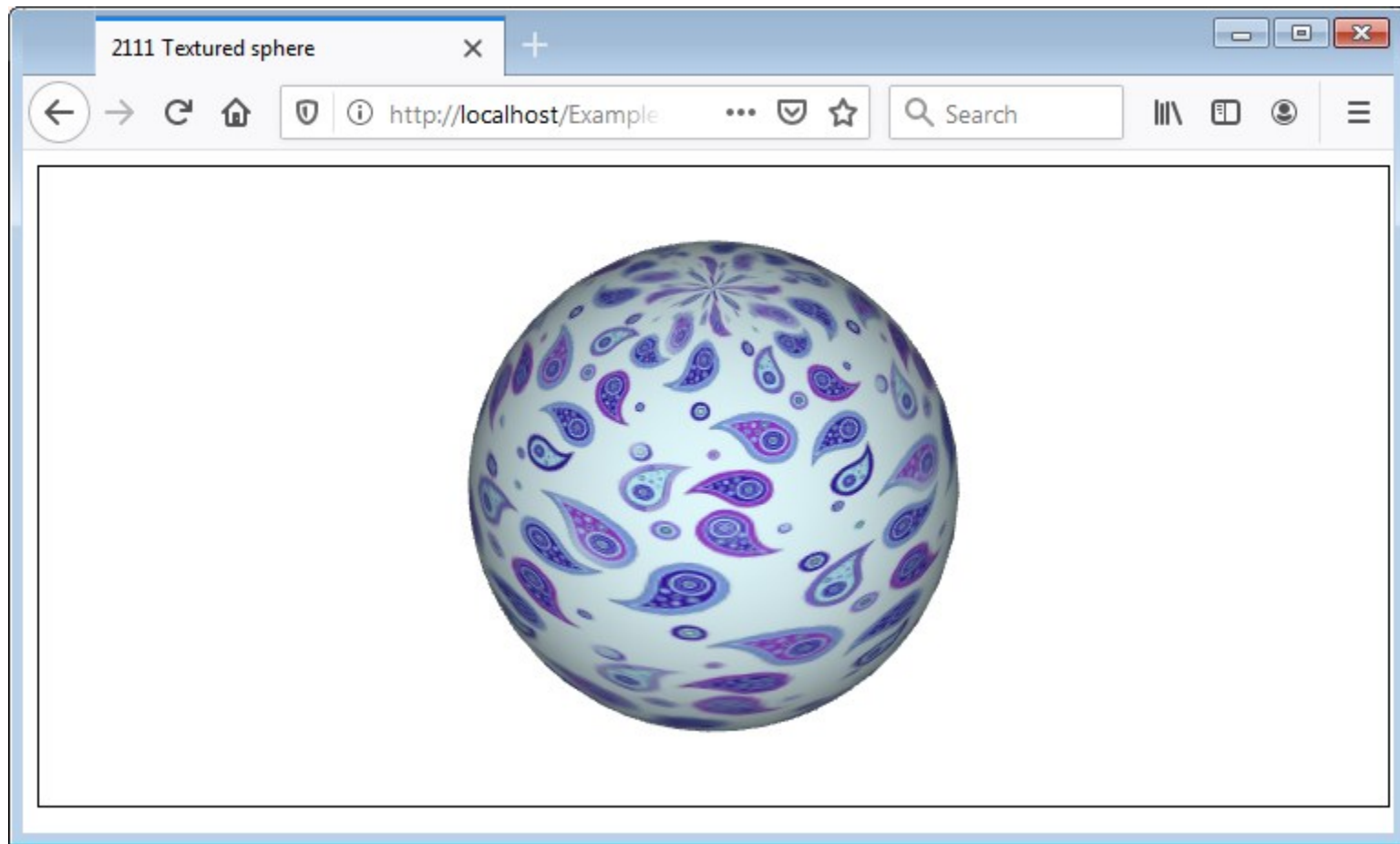
- Impossible to map rectangular texture to spherical surface
- Texture is collapsing near the poles



Paisley pattern blue

Author: Пол Шермън, license: Public Domain

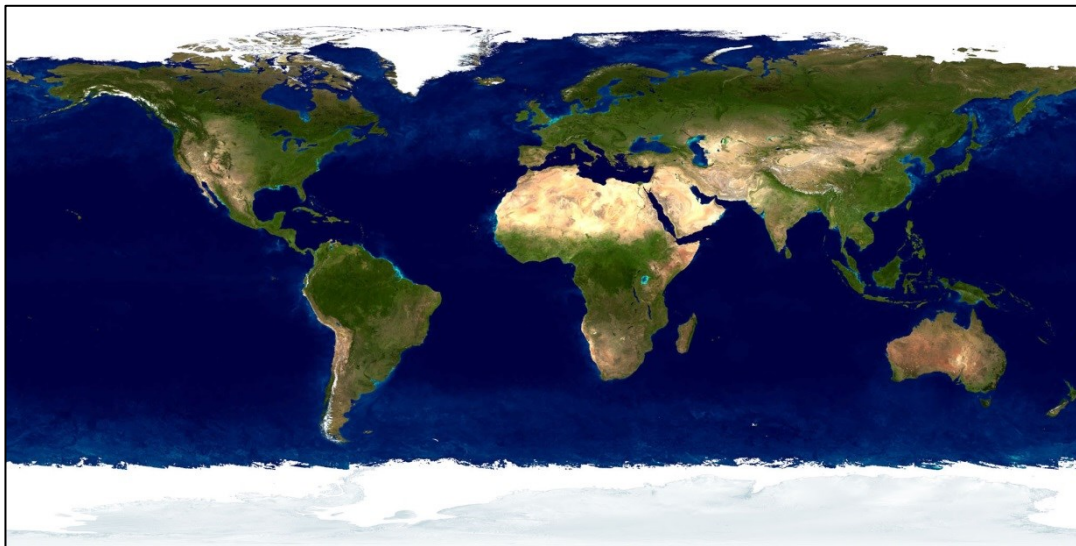
http://www.wpclipart.com/textures/paisley/paisley_pattern_blue.png.html



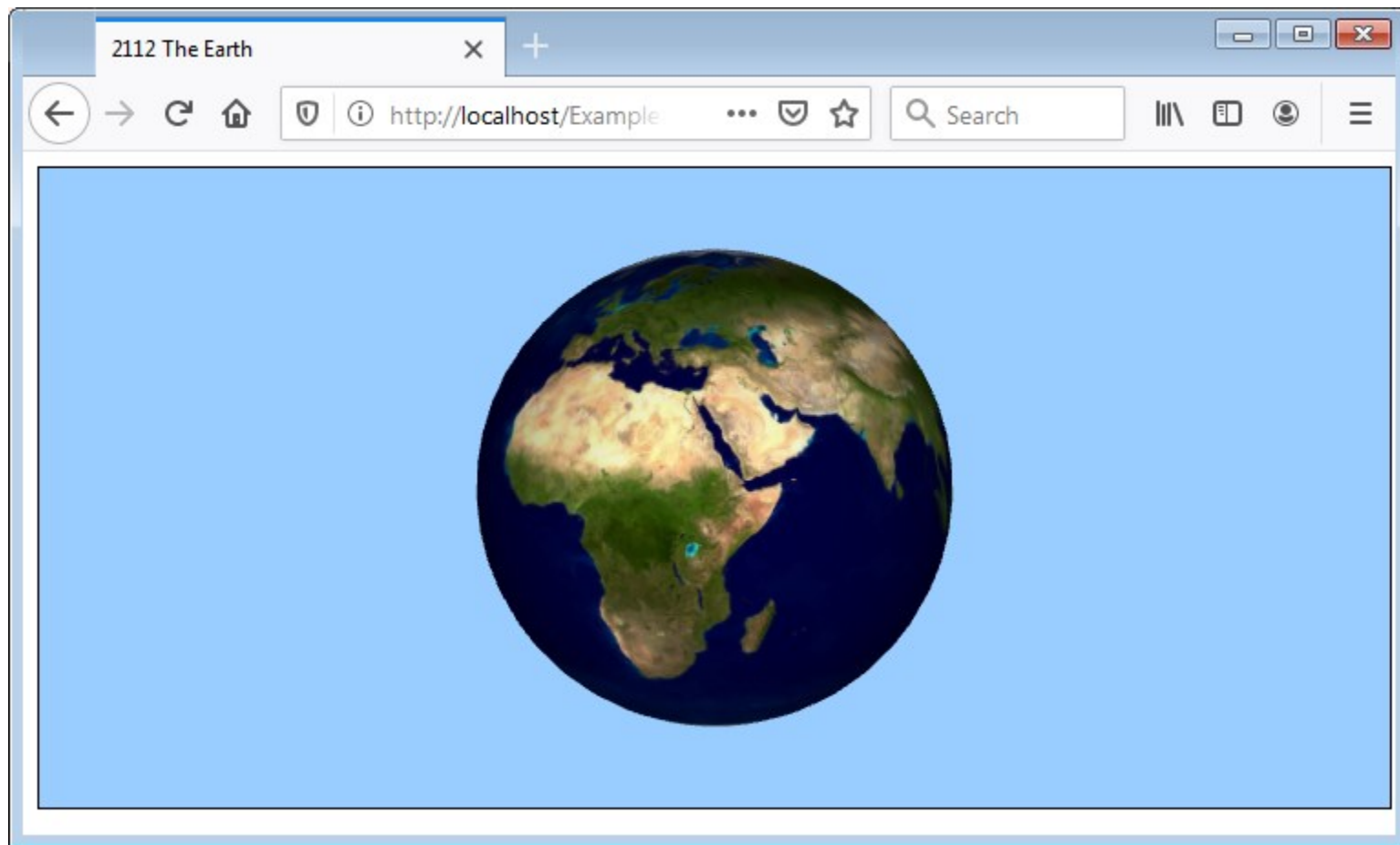
TRY IT

Earth model

- Using a special texture
- Expanded near the poles
- Compensating collapsing when mapped onto a sphere



The Blue Marble: Land surface, ocean color and sea ice
Authors: Reto Stöckli, Robert Simmon, NASA Goddard Space Flight Center, license: Public Domain
<http://visibleearth.nasa.gov/view.php?id=57730>



TRY IT



Summary

Textures



Textures in computer graphics

- Rectangular images, preferably with size $2^n \times 2^m$
- Mapped onto graphical objects
- Seamless textures can be glued together

Source

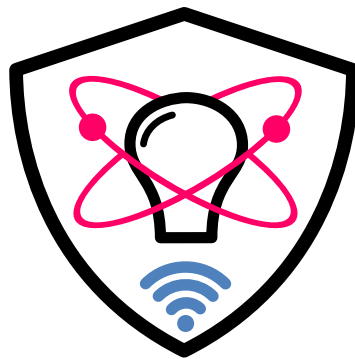
- Read from an image file
- If file is in another domain, there might be CORS issue
- Texture data could be embedded in Data URI
- Local test may use local mini web server

Creating a texture

- With class `Image`
- Stored as property `image`

Properties of textures

- Size in `scale`
- Size of the base in `baseScale`
- Shifting in `offset`
- Address of the image in `url`



ICT in SES

The end

Comments, questions