

P.PORTO

POLITÉCNICO
DO PORTO
ESMAD

COMPUTAÇÃO GRÁFICA
TSIW

Teacher

TERESA TERROSO

- responsible of the Curricular Unit
- teacher of both TP and PL classes
- Email: teresaterroso@esmad.ipp.pt
- Office: C305
- Office hours (for students):
5^{as} – 14h to 18h

Learning outputs

- Understand the basic concepts of **Web Graphic Animation (2D)**, **Computer Graphics (3D)** and **Web interaction**
- Know and distinguish different tools for Animation and Computer Graphics
- Render dynamic graphics on the Web using the HTML5 element **CANVAS**, **CSS** and **JavaScript**
- Create and manipulate simple shapes, text, and images on a web page and animate objects
- Implement basic interactions with the user
- Understand the 3D graphic pipeline
- Implementation of graphic systems using graphics engines (**Three.js**)

Syllabus

Module 1. Web Graphic Animation

Module 2. Animation using JavaScript

Module 3. 2D graphics on Canvas

Module 4. Math, physics, and animation

Module 5. CSS animations and transitions

Module 6. 3D graphic pipeline

Module 7. Three.js graphic library

Teaching methodologies

- All classes in **informatic labs**
- **TP classes**
 - use of formal presentations, providing several examples and showing practical applications
- **PL classes**
 - in-class exercises, with presentation, discussion and immediate evaluation of the student work

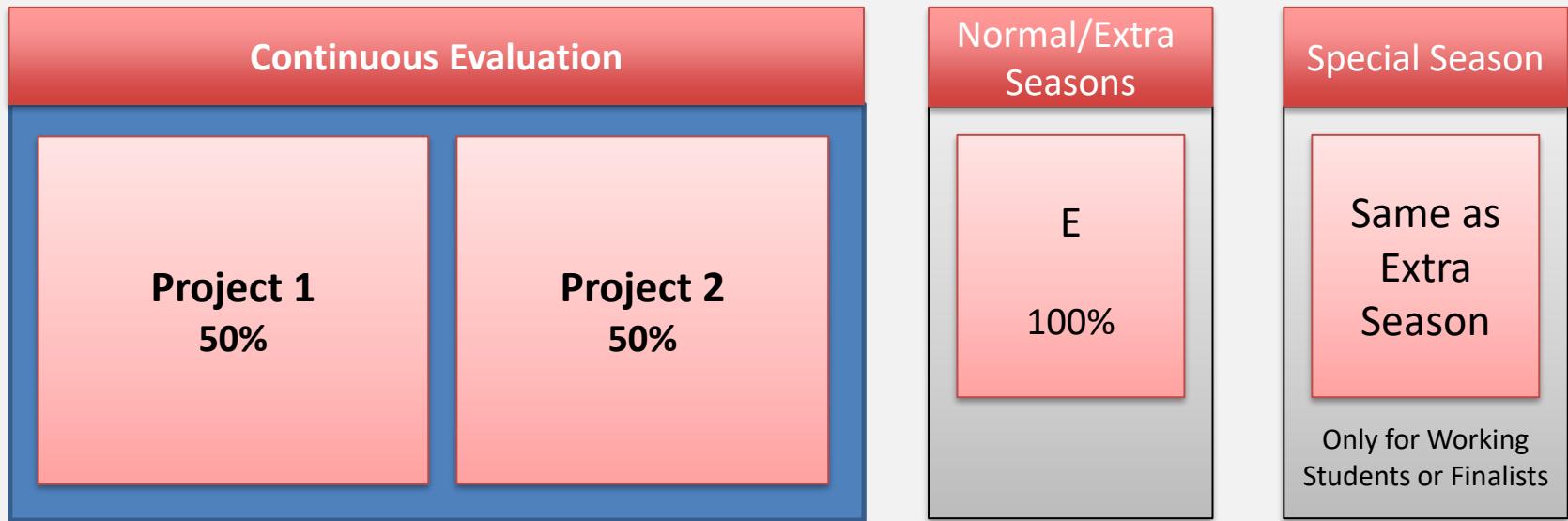
Teaching methodologies

- **Moodle**
 - General informations
 - Learning materials, assignments and assessment
 - URL: <https://moodle.esmad.ipp.pt/course/view.php?id=177>
- **Teams**
 - Faster communications
 - Group projects
 - URL: <https://tinyurl.com/LTSIWCOMPGRAF>



Do you receive / read the messages from Moodle forums or from chats within Teams channels?
Check your institutional e-mail on a daily basis!

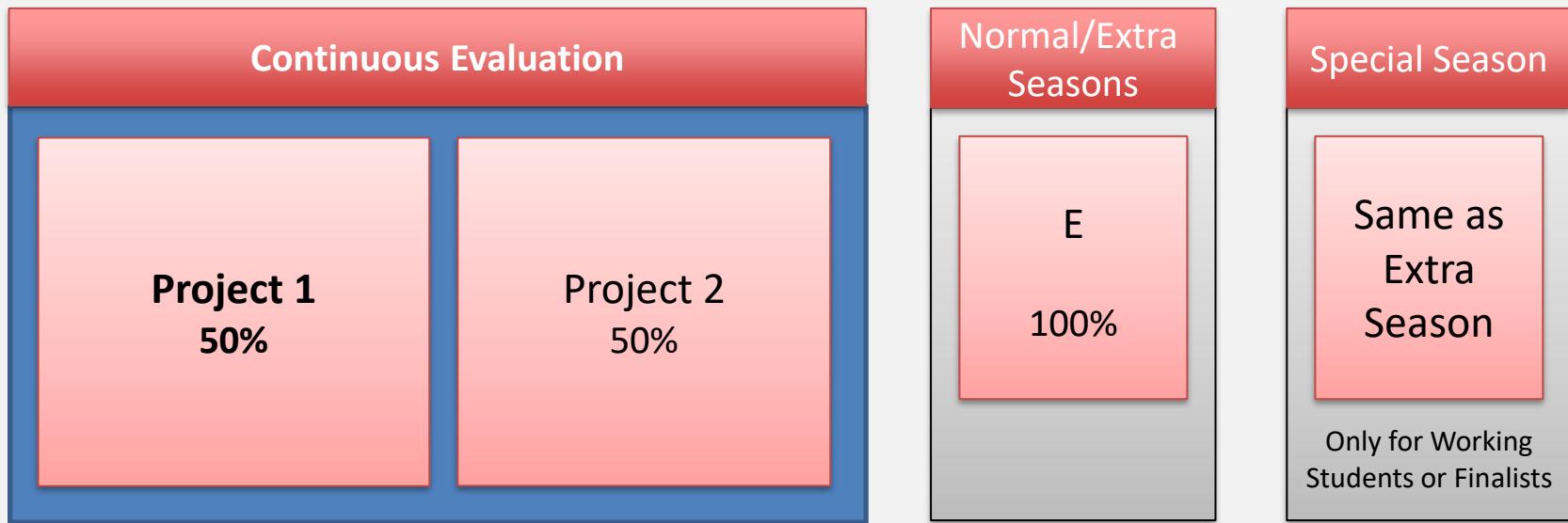
Assessment



CONTINUOUS EVALUATION

- Projects (P1 and P2): minimum grade ≥ 10 values
- Final Grade = $P1 * 0.5 + P2 * 0.5$
- Final Grade ≥ 10

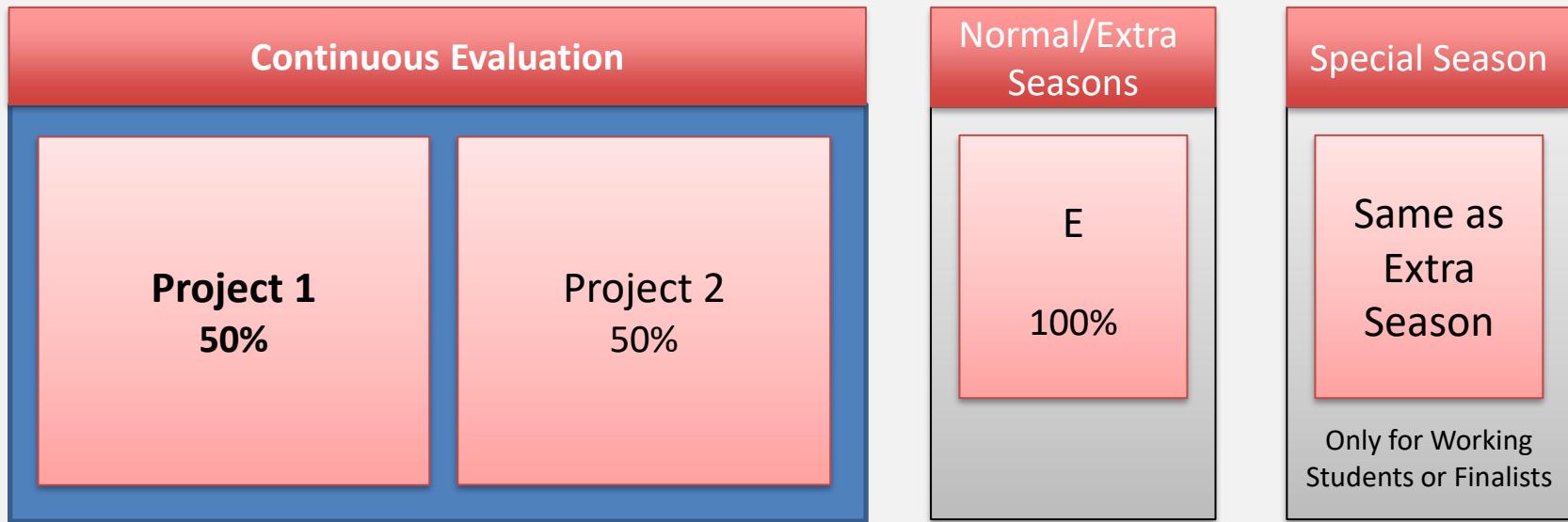
Assessment



CONTINUOUS EVALUATION – Project P1

- 2D web game using the Canvas API and CSS animations/transitions
- Developed in **groups** (except for Working Students or other special cases, previously discussed with the professor)
- Groups are indicated in Moodle (until the end of next week)
- After that, 1 channel per group will be created in Teams for private reunions and discussions between team members and professor

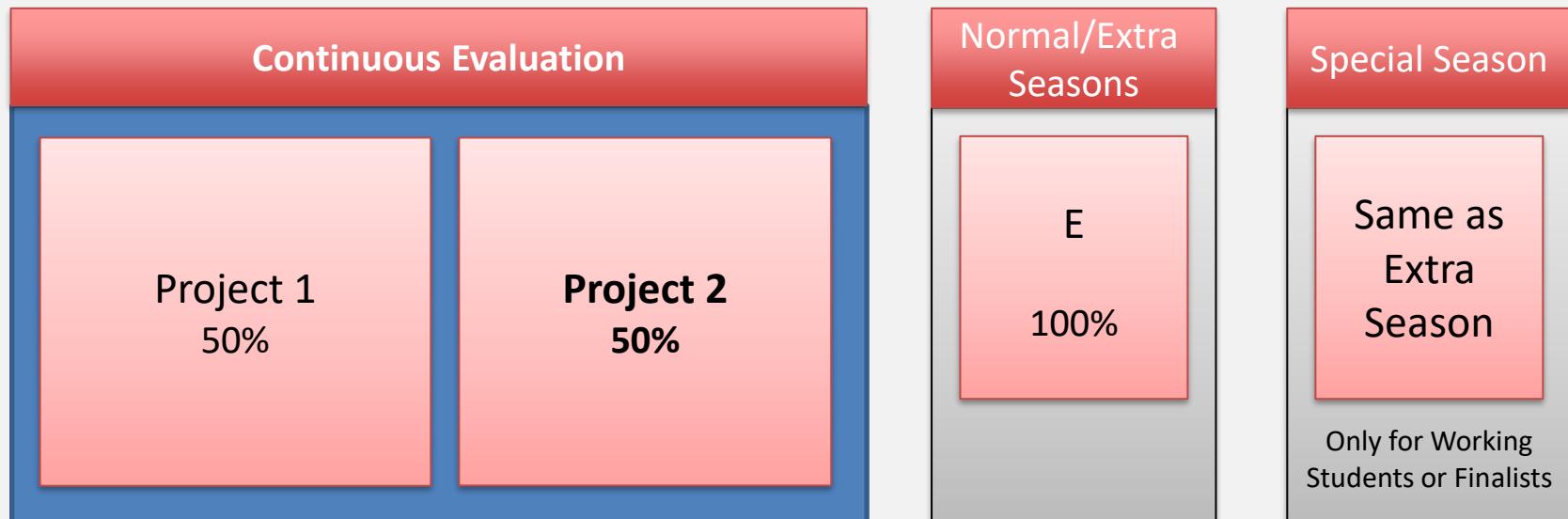
Assessment



CONTINUOUS EVALUATION – Project P1

- In class meetings with the groups will be evaluated for each group member (absence of a student will have a penalty in the project grade)
- There will be mandatory intermediate and final presentations (not evaluated, however if a student fails to be present he will have a penalty in the project grade)

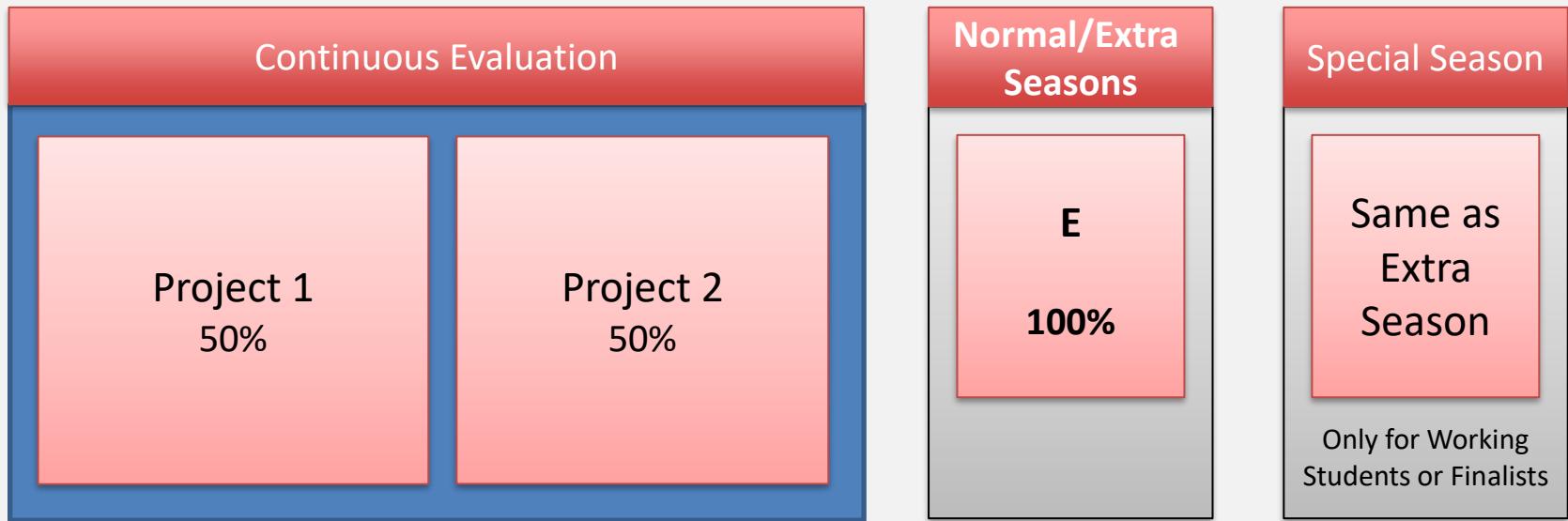
Assessment



CONTINUOUS EVALUATION – Project P2

- 3D web animation using Three.js library
- Developed **individually**
- Students absent in classes dedicated to the project companion will have a penalty in the project grade
- There will be mandatory intermediate and final presentations (not evaluated, however if a student fails to be present he will have a penalty in the project grade)

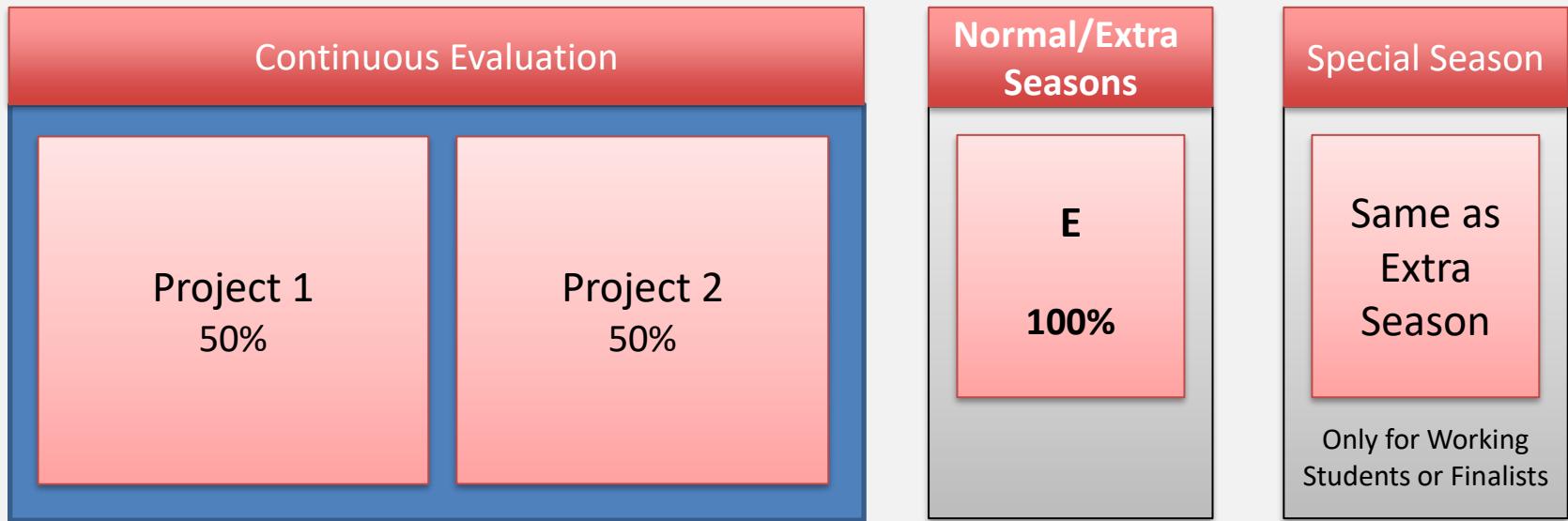
Assessment



Normal Season

- Exam: final grade 100% (minimum grade 10 values)
- Access to Normal Season only for students who do NOT deliver ANY of the projects (P1 and P2)

Assessment



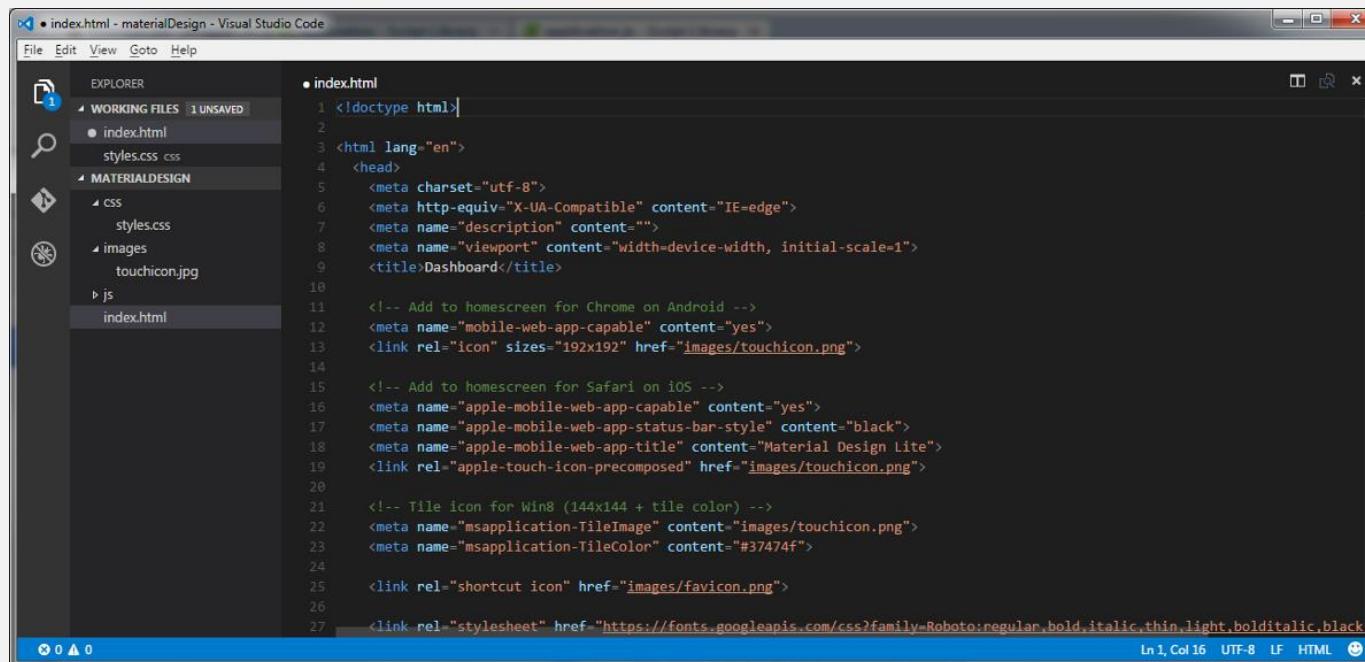
Extra Season

- Exam: final grade 100% (minimum grade 10 values)
- For grade approvement or improvement

Tools

Code development

- Visual Studio Code: <https://code.visualstudio.com/>



Playgrounds

- PlayCode: <https://playcode.io/>

The screenshot shows the PlayCode.io interface. On the left, there's a sidebar with file navigation: HTML (index.html), JS (script.js), CSS (style.css), and LIBS (jquery.js, three.js). The main area has tabs for 'script.js' (selected) and 'preview'. The 'script.js' tab contains the following code:

```
// Try edit message
const data = {
  message: 'Hello world',
}
$( '#msg' ).html( data.message )
console.log( data )
```

The 'preview' tab shows the output: a large green <> icon followed by the text "Hello world". Below the preview, the console output is shown:

```
console ×
{
  message: "Hello world"
}
```

Playgrounds

- Liveweave: <https://liveweave.com/>

The screenshot shows the Liveweave interface. On the left, there's a sidebar with icons for different tools: Web Editor (HTML/CSS editor), CSS Explorer (CSS selector tool), Color Explorer (color palette), and Vector Editor (image editor). The main workspace has a dark header bar with navigation links: Liveweave, Tools, Library, Team Up, Tidy All, Save, and two mode switches (Live Mode and Night Mode) along with a Login button. Below the header is a large list of popular JavaScript libraries categorized by letter (A-Z). To the right of the list is a code editor area with a placeholder "1 // Write JavaScript here" and a yellow "JS" button. The main content area displays the text "Hello World!".

A: Animate CSS (3.7.0)
B: Bootstrap (latest)
C: Create JS (latest)
D: D3 (3.0)
E: Ember JS (latest)
F: Fabric JS (1.3.0)
G: Font Awesome (4.7.0)
J: jQuery (latest) Popular
K: Kinetic JS 4.7.3
L: Modernizr (latest)
M: Moo Tools (latest)
P: Paper JS (0.9.9)
Q: Qooxdoo (3.0.1)
R: Raphael JS (2.1.4)
S: Script.aculo.us (latest)
T: Three JS (latest)
U: Underscore JS (latest)
W: Web Font Loader (1.4.10)
Y: YUI (3.13.0)
Z: Zepto (latest)

```
1 // Write JavaScript here
```

Hello World!

Playgrounds

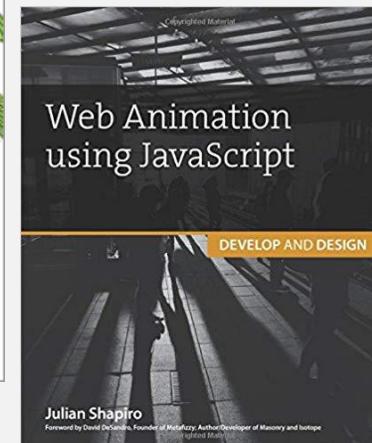
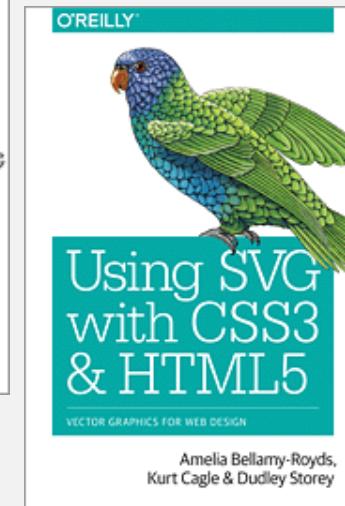
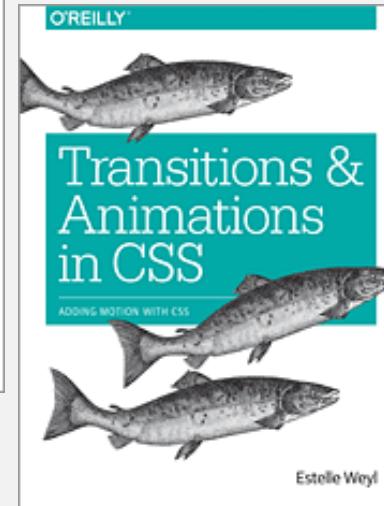
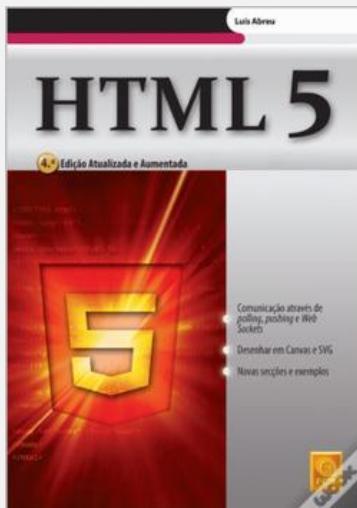
- CodePen: <https://codepen.io/>

The screenshot shows the CodePen interface with a dark theme. At the top, there's a navigation bar with icons for heart, save, fork, settings, change view, and a dropdown. Below the bar, the title "Knight Rider (Canvas Animation)" is displayed along with the author "A PEN BY Teresa Terroso". The interface is divided into three main sections: HTML, CSS, and JS. The HTML section contains the code for creating a canvas element. The CSS section contains the code for setting the body width and height to 100%. The JS section contains the code for initializing a canvas, getting its context, setting line width, and defining an animate function. On the right side, the preview area shows a horizontal bar with several red segments of varying widths, representing the Knight Rider effect.

```
2 <canvas id="myCanvas" width="600" height="50" style="border:solid 3px black"></canvas>
1 html, body {
2   width: 100%;
3   height: 100%;
4
5 //INICIALIZA
6 //ctx.strokeStyle =
7 ctx.fillStyle = "red";
8 var x = 0, inc = 50;
9
10 function animate() {
11   //
```

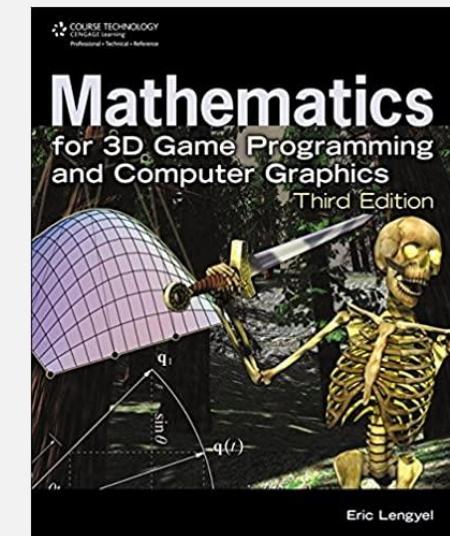
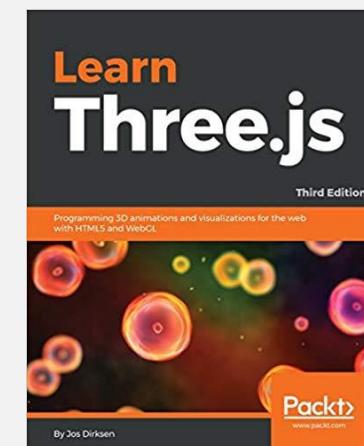
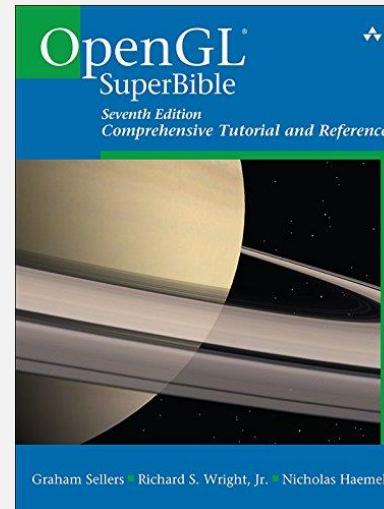
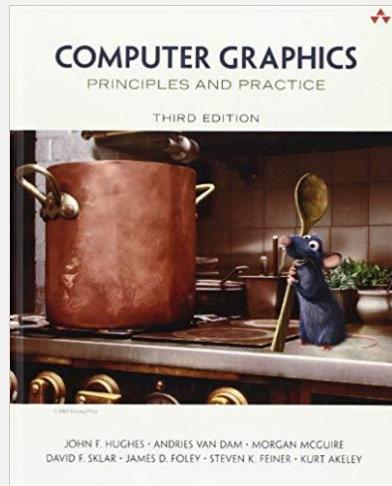
Studying tools (2D graphics)

- Books



Studying tools (3D graphics)

- Books



Online support (2D)

- W3schools: <https://www.w3schools.com/graphics/default.asp>

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP MORE ▾ REFERENCES ▾ EXER...

HTML Graphics

Graphics HOME

Google Maps

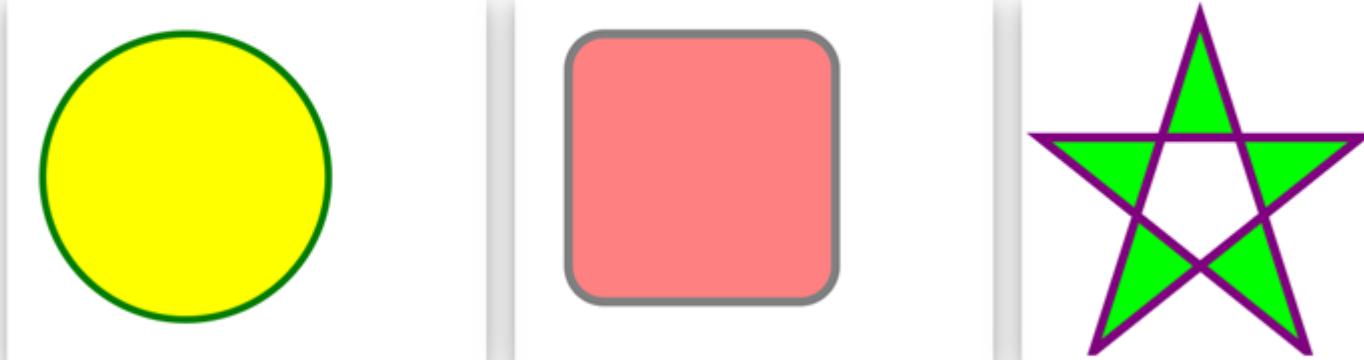
Maps Intro
Maps Basic
Maps Overlays
Maps Events
Maps Controls
Maps Types
Maps Reference

SVG Tutorial

SVG Intro
SVG in HTML5
SVG Rectangle

SVG

The HTML <svg> element allows vector based graphics in HTML:



HTML Canvas

The HTML <canvas> element can be used to draw graphics on a web page:

Online support (2D & 3D)

- MDN web docs: <https://developer.mozilla.org/en-US/docs/Web/Guide/Graphics>

The screenshot shows the MDN web docs homepage. At the top left is the logo 'MDN web docs' with 'moz://a'. To its right are three dropdown menus: 'Technologies', 'References & Guides', and 'Feedback'. On the far right are 'Sign in' and a magnifying glass icon for the search bar.

Graphics on the Web

Languages [Edit](#) [⚙️](#)

Websites and applications often need to present graphics, such as images. Static images can easily be displayed using the `` element, or by setting the background of HTML elements using the `background-image` property. You can also construct graphics on-the-fly, or manipulate images after the fact. These articles provide insight into how you can accomplish this.

2D Graphics [🔗](#)

Canvas

The `<canvas>` element provides APIs to draw 2D graphics using JavaScript.

SVG

Scalable Vector Graphics (SVG) lets you use lines, curves, and other geometric

3D Graphics [🔗](#)

WebGL

A guide to getting started with WebGL, the 3D graphics API for the Web. This technology lets you use standard OpenGL ES in web content.

Video [🔗](#)

Online support (2D)

- Tutorials Point:

<https://www.tutorialspoint.com/svg/index.htm>

The screenshot shows the homepage of the Tutorials Point SVG tutorial. It features a large 'SVG' logo with a stylized cloud-like shape above it. Below the logo, the text 'LEARN SVG' is displayed in a large orange button, followed by the subtitle 'scalable vector graphics' in a smaller orange font. A navigation menu below the subtitle includes links for 'SVG Tutorial', 'SVG - Home', 'SVG - Overview', 'SVG - Shapes', and 'SVG - Tutorials'. At the bottom, there are links for 'PDF Version', 'Quick Guide', 'Resources', 'Job Search', and 'Discussion'.

The screenshot shows the homepage of the Tutorials Point SVG tutorial. It features a large 'SVG' logo with a stylized cloud-like shape above it. To the right of the logo, a large orange button contains the text 'LEARN SVG' and the tagline 'simply easy learning'. Below this, a section titled 'SVG Tutorial' is shown with five buttons: 'PDF Version', 'Quick Guide', 'Resources', 'Job Search', and 'Discussion'. A descriptive paragraph at the bottom states: 'Scalable Vector Graphics commonly known as SVG is a XML based format to draw vector images. It is used to draw two dimensional vector images.'

Online support (3D)

- WebGL Fundamentals: <https://webglfundamentals.org/>

The screenshot shows the homepage of the WebGL Fundamentals website. At the top, there is a navigation bar with a dropdown menu set to "English". Below the header, the title "WebGL Fundamentals" is displayed in a large, bold, dark font. Underneath the title, there is another language selection dropdown also set to "English". A sub-headline reads "WebGL from the ground up. No magic." followed by a descriptive paragraph about the nature of the articles. At the bottom, there is a call to action encouraging users to visit the website for WebGL2 fundamentals.

Fix, Fork, Contribute

WebGL Fundamentals

English

WebGL from the ground up. No magic.

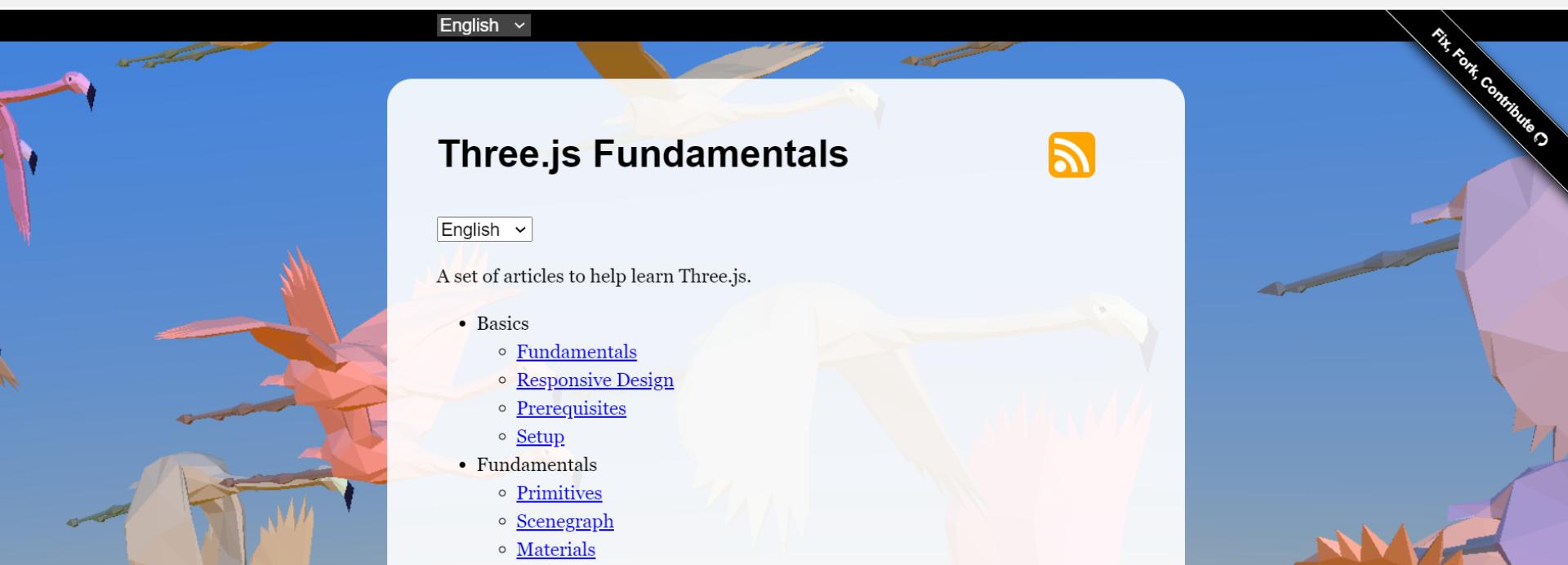
These are a set of articles that teach WebGL from basic principles. They are NOT old rehashed out of date OpenGL articles like many others on the net. They are entirely new, discarding the old out of date ideas and bringing you to a full understanding of what WebGL really is and how it really works.

If you are interested in WebGL2 please see webgl2fundamentals.org

- Fundamentals
 - [Fundamentals](#)

Online support (3D)

- Three.js Fundamentals: <https://threejsfundamentals.org/>



Online support (3D)

- Three.js Examples: <https://stemkoski.github.io/Three.js/>

Three.js Examples

The goal of this collection is to provide a set of basic and instructive examples that introduce the various features in **Three.js**. The source code for each page contains detailed comments.

Hosted at [GitHub](#). Total examples: 83. Last updated: 23 August 2013. Compatible with Three.js version 60.

Basic Examples

- [Hello World](#) • [Template](#) • [Info Box](#) • [Materials - Solid](#) • [Color Explorer](#) • [Wireframe](#) • [Lines and Dashed Lines](#) • [Helpers](#) • [Outline Effect](#) • [Face/Vertex Colors](#) • [Shapes](#) • [Extrusion](#) • [Text3D](#)

Texture Examples

- [Textures](#) • [Texture - Repeat](#) • [Texture - Text3D](#) • [Translucence](#) • [Shadow](#) • [Subdivision Modifier - Cube](#) • [SkyBox](#) • [Reflection](#) • [Refraction](#) • [Bubble](#) • [Texture from Canvas](#) • [Texture Animation](#)

Sprite Examples

- [Sprites](#) • [Sprite Text Labels](#) • [Labeled Geometry](#)

Mouse and Keyboard Examples

- [Mouse Sprite](#) • [Mouse Click](#) • [Mouse Hovering](#) • [Mouse Tooltip](#) • [Keyboard Events](#)

Camera/Renderer Examples

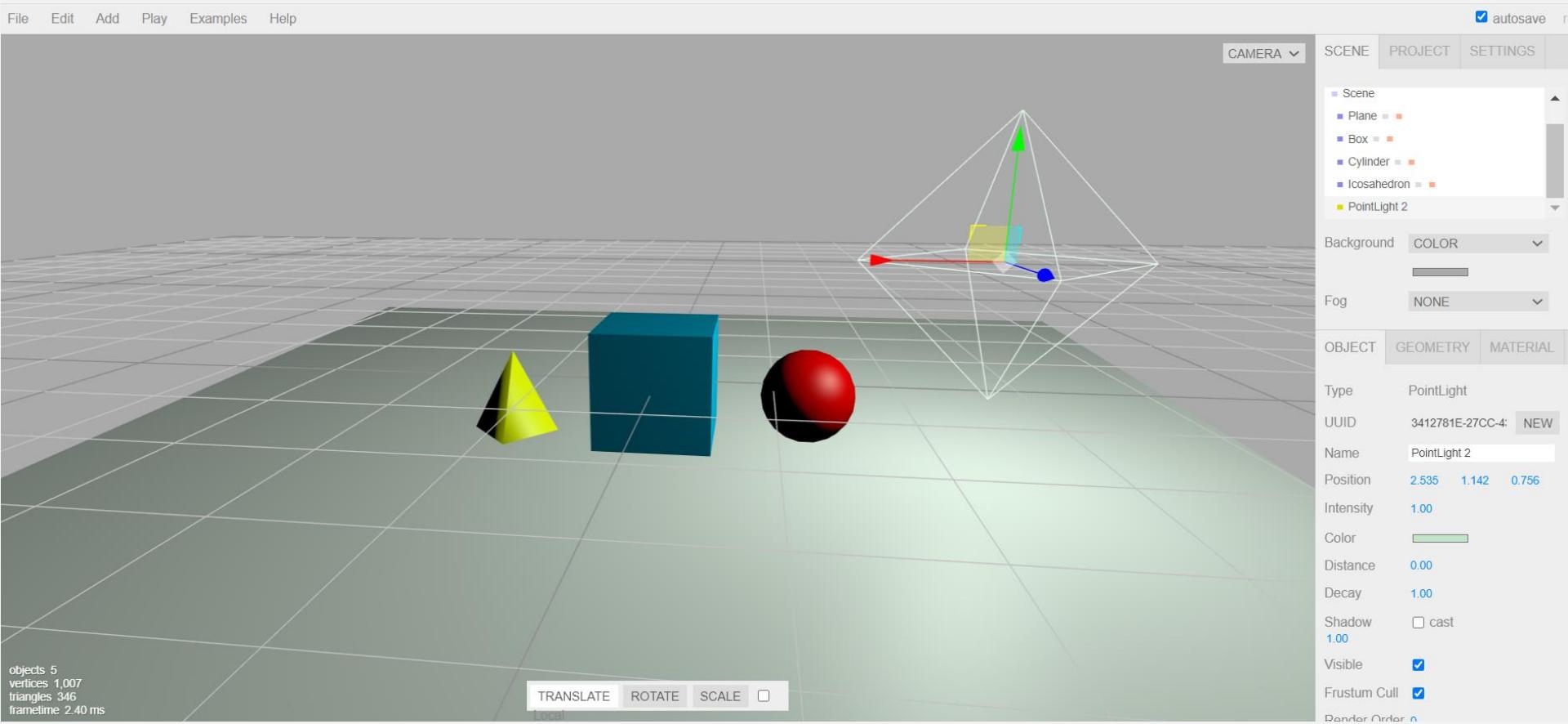
- [Mesh Movement](#) • [Chase Camera](#) • [Multiple Cameras](#) • [Camera to Texture](#) • [Viewports - Dual](#) • [Viewports - Quad](#) • [Embedded HTML](#) • [Red/Blue Anaglyph](#)

Shader Examples

- [Shader - Simple](#) • [Shader - Explorer](#) • [Shader - Sphere Unwrapping](#) • [Shader - Attributes](#) • [Shader - Animated Materials](#) • [Shader - Animated Fireball](#) • [Shader - Glow Effect](#) • [Simple Glow \(non-shader\)](#)

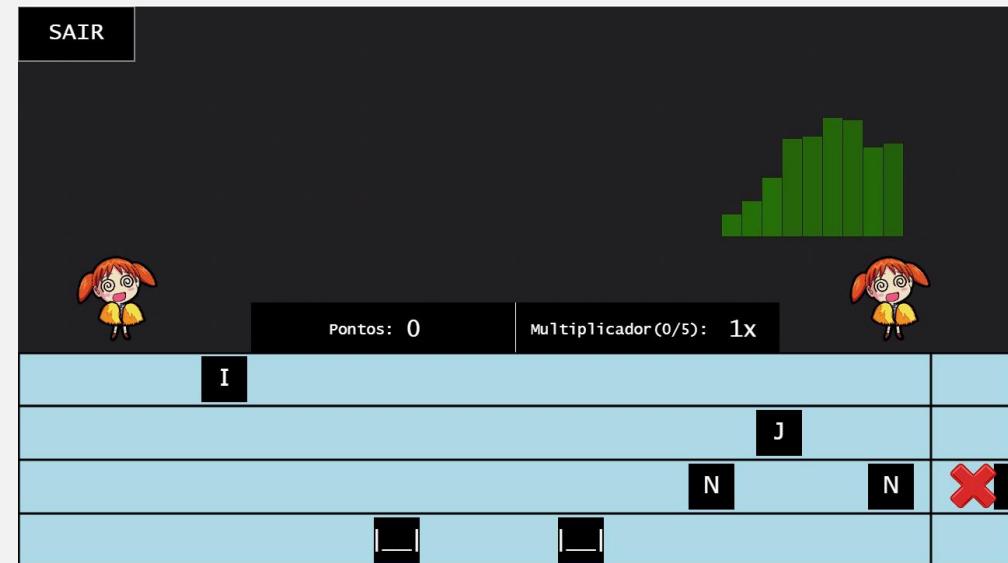
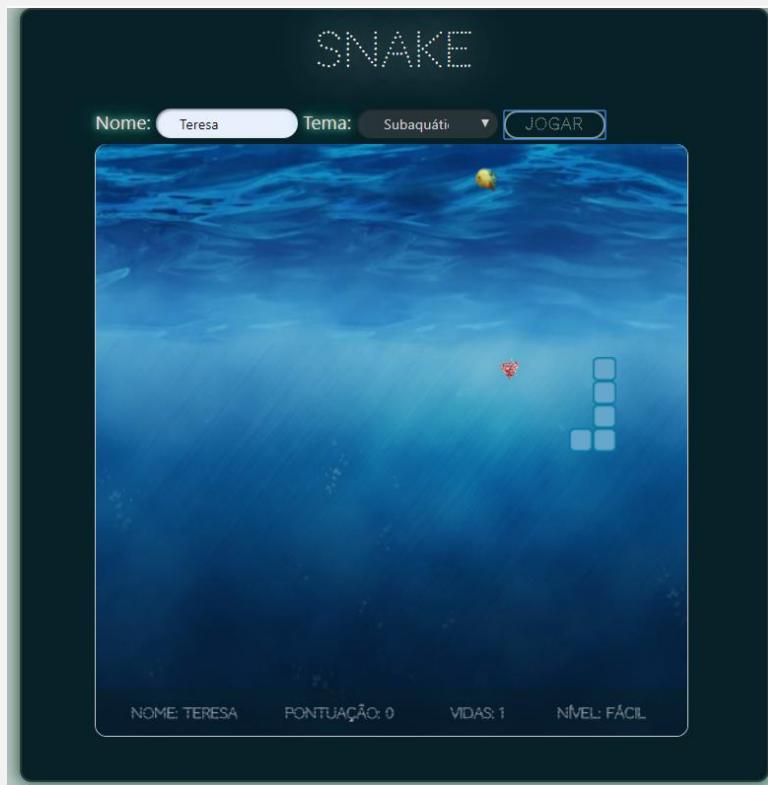
Online support (3D)

- Three.js online editor: <https://threejs.org/editor/>



Project works examples (2D)

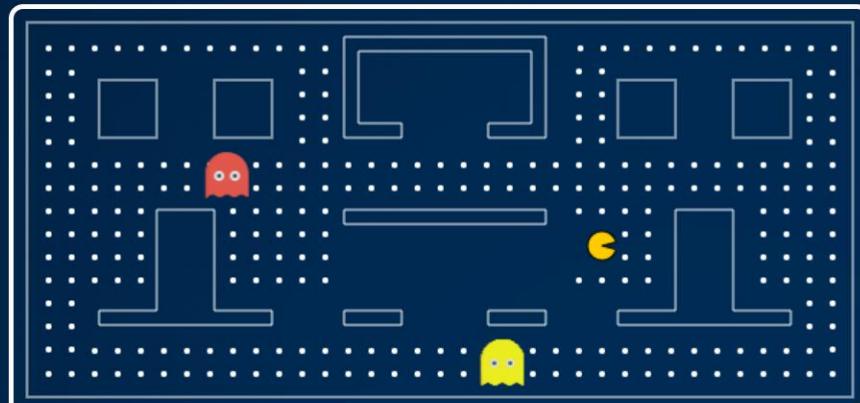
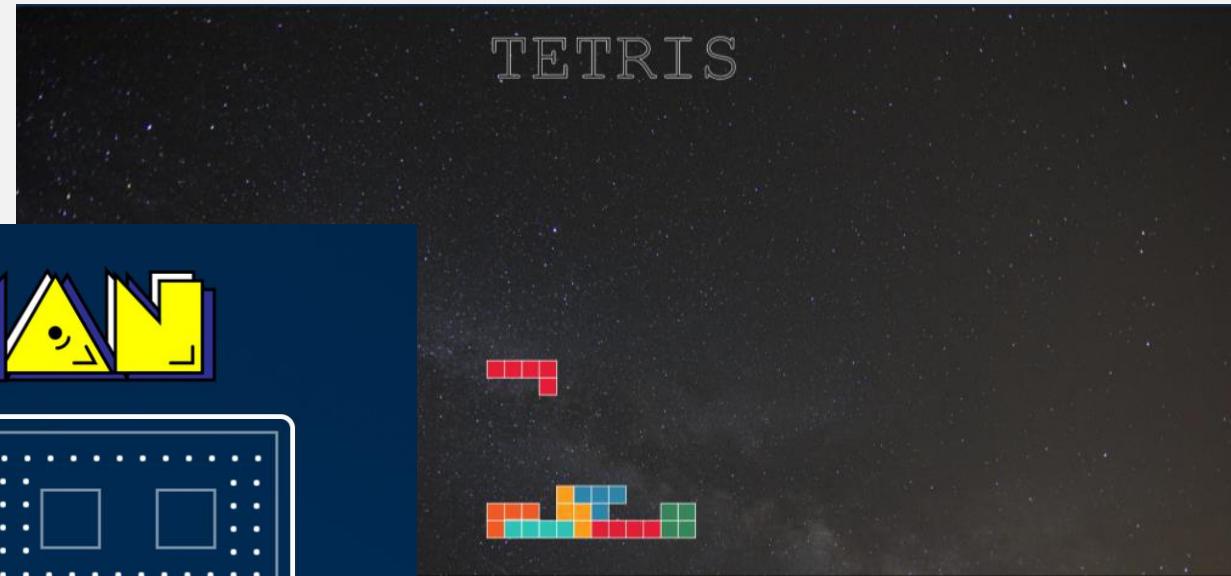
Finger Tap



Snake

Project works examples (2D)

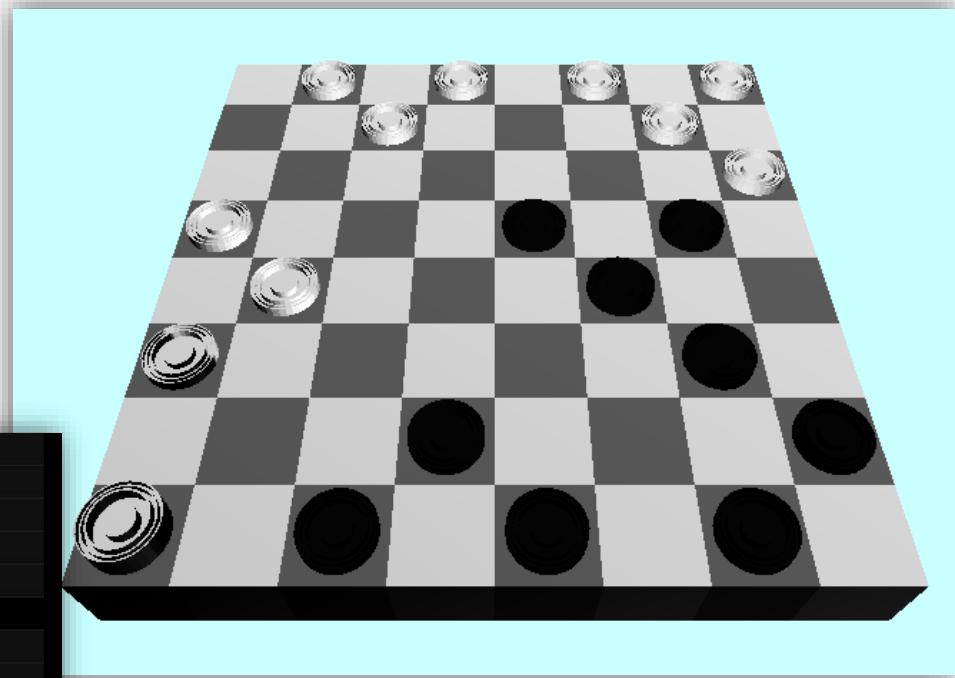
Tetris



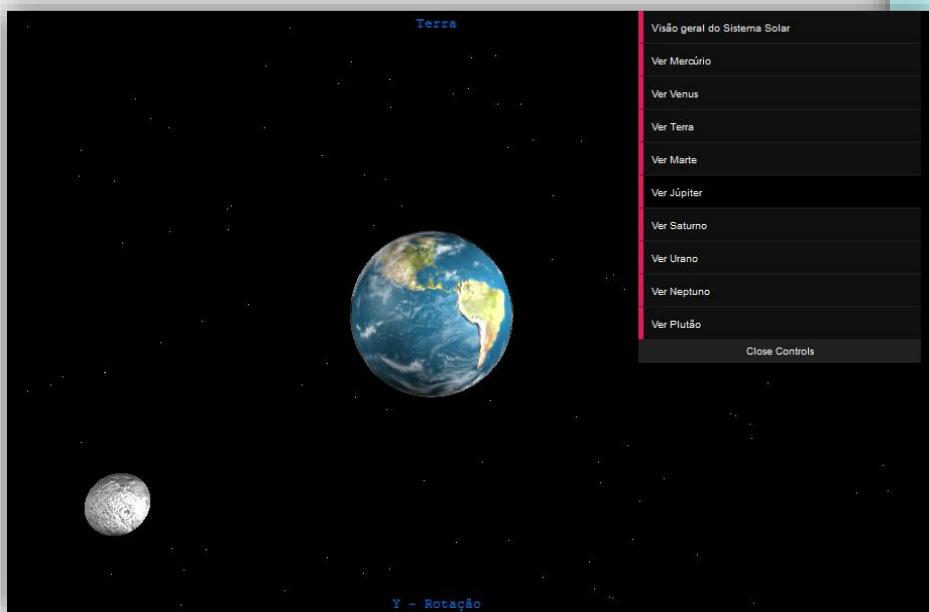
PacMan

Project works examples (3D)

Checkers

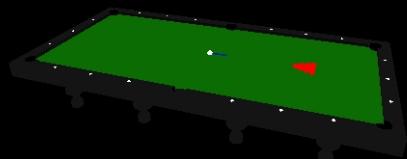


Solar System



Project works examples (3D)

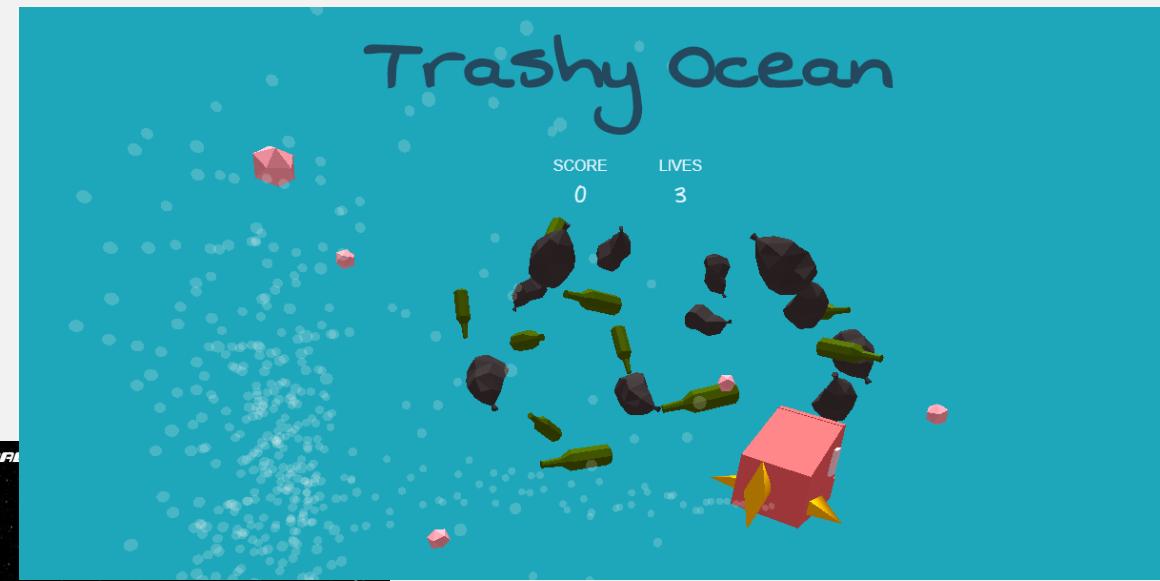
Robot Paw



Snooker

Project works examples (3D)

Trashy Ocean



Space Shooter

Some great experiments (2D)

8 Bit Color Cycle

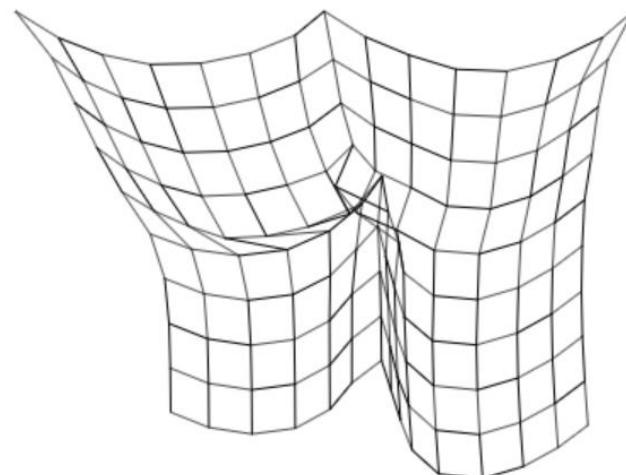
<http://www.effectgames.com/demos/canvascycle/>



Some great experiments (2D)

Cloth experiment

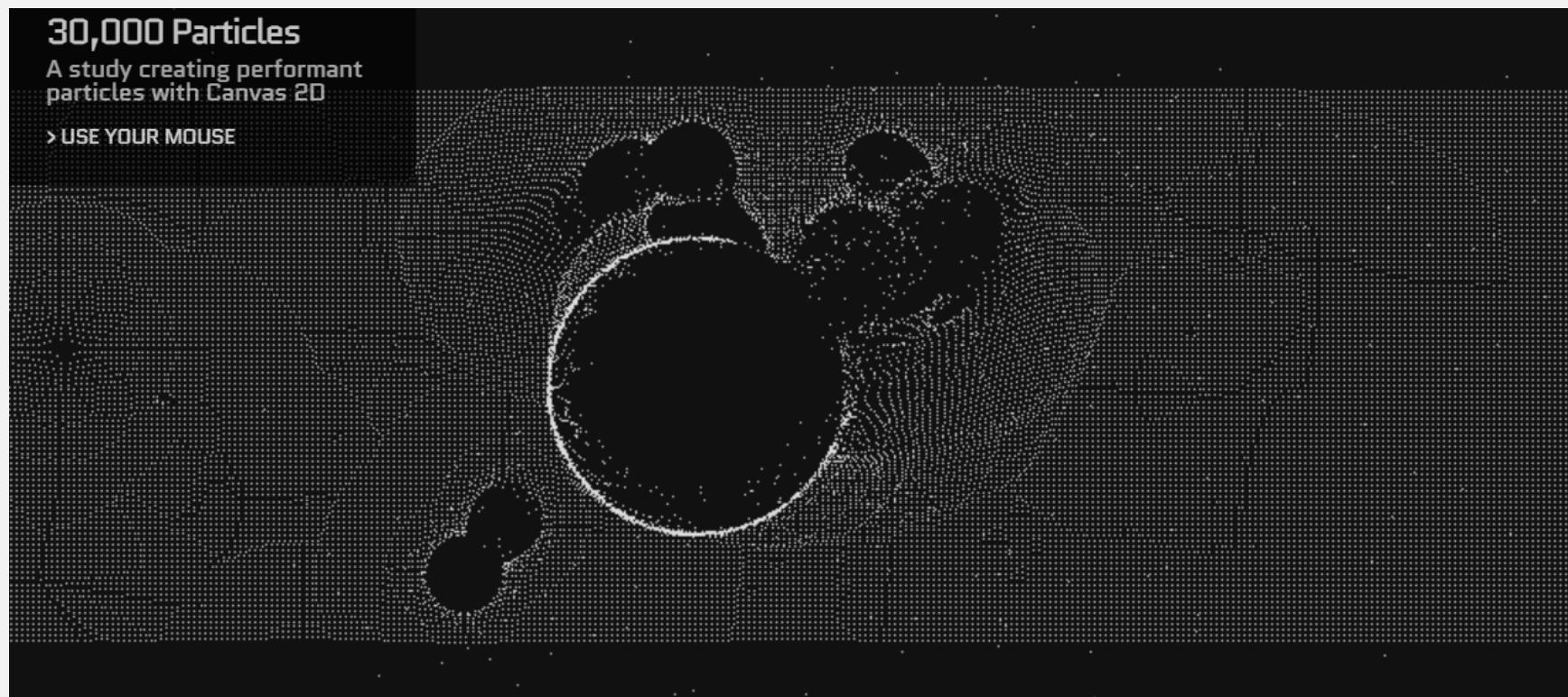
<http://andrew.wang-hoyer.com/experiments/cloth/>



Some great experiments (2D)

30000 Particles

<https://codepen.io/soulwire/pen/Ffvlo>



Some great experiments (2D)

Tumbler

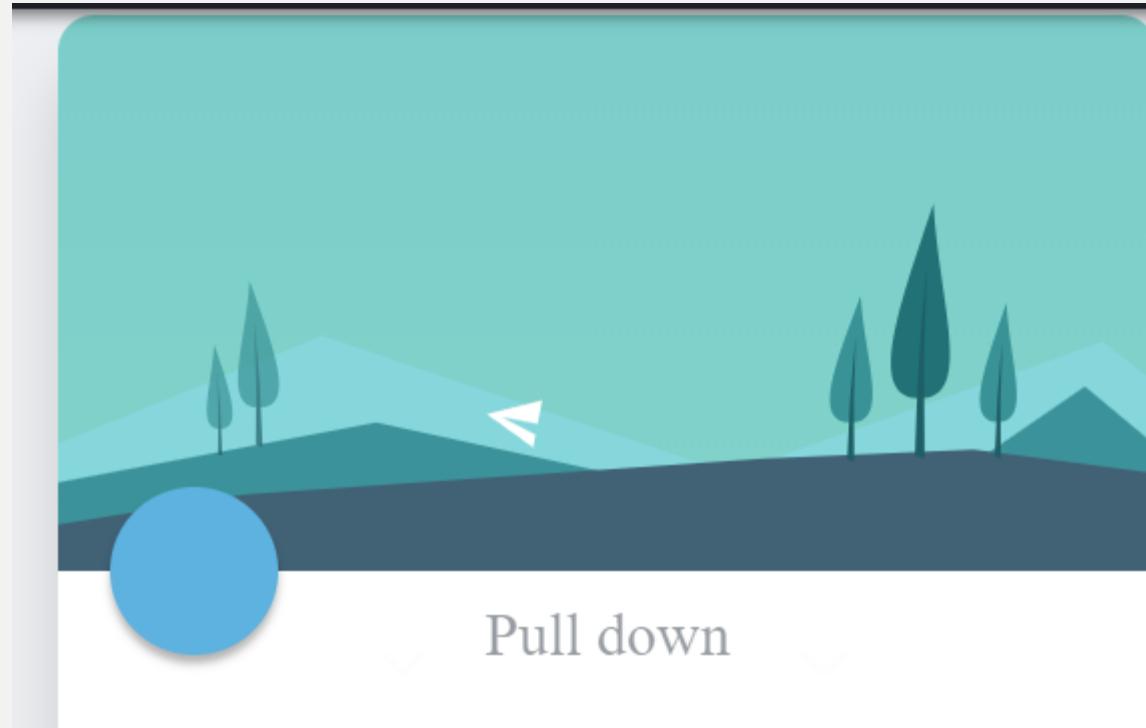
<http://andrew.wang-hoyer.com/experiments/tumbler/>



Some great experiments (2D)

Pull Down to Refresh

<https://codepen.io/suez/pen/oXLroX>



Some great experiments (2D)

Flat Workspace

<https://codepen.io/thiennhat/pen/BNByzJ>



Some great experiments (2D)

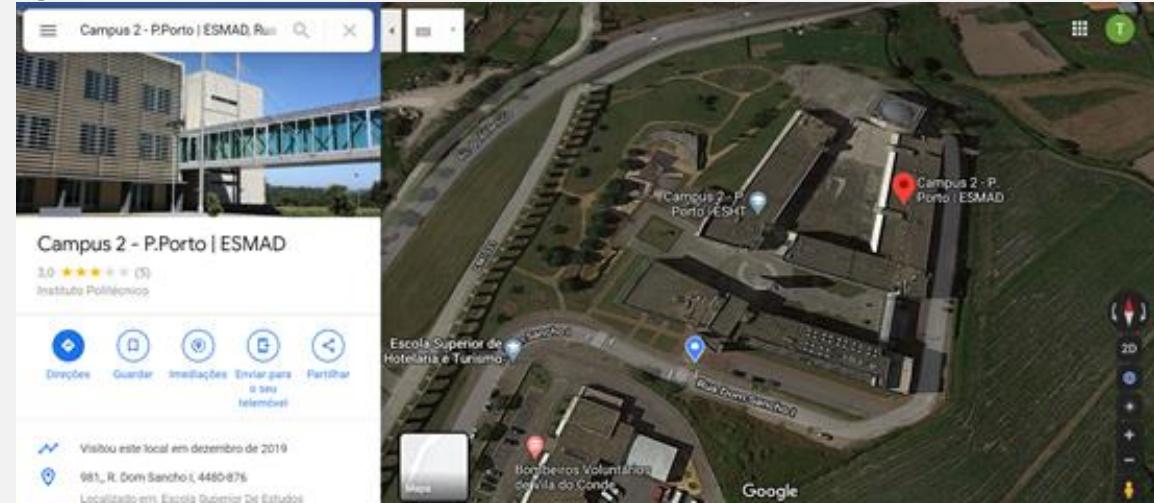
Pure CSS Biker

<https://codepen.io/miocene/pen/jLzmJq>



Web 3D examples

Google Maps
(terrain view)

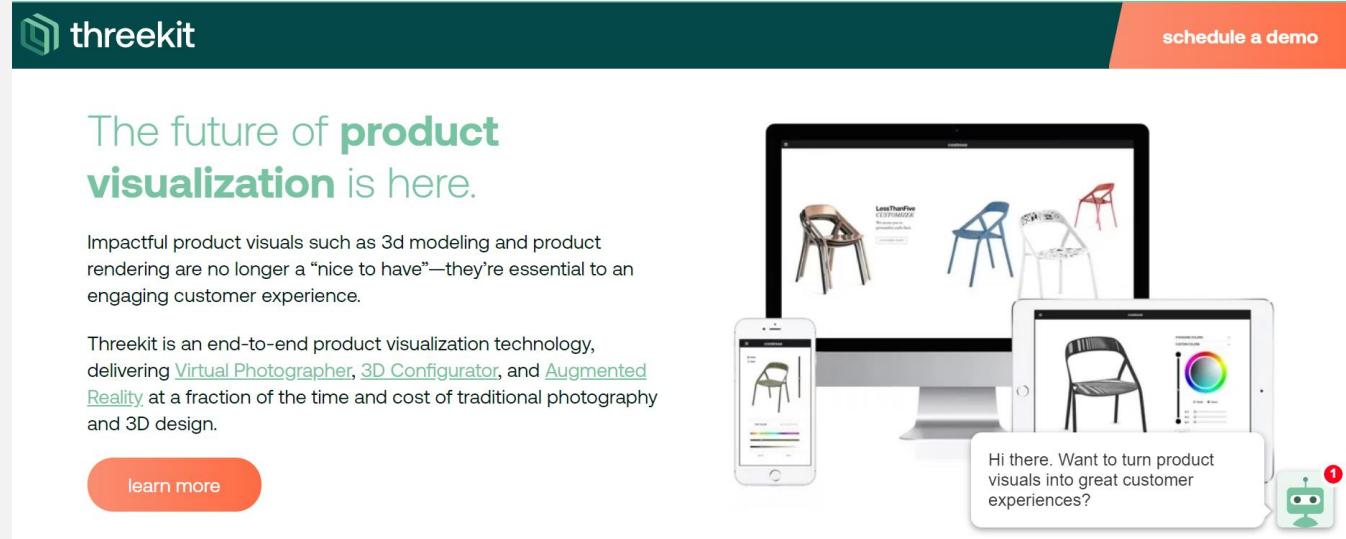


A screenshot of a 3D anatomical model from BioDigital. The model shows a human head and neck in profile, with a translucent skin layer revealing internal structures like the brain, spinal cord, and major blood vessels. The interface includes a sidebar with navigation buttons and a "REGIONAL" section titled "Head and Neck". The text describes the head as the most superior region of the body and the neck as the region between the head and trunk, serving as a passageway for vital vessels and structures.

Biomedical
(e.g. [BioDigital](#))

Web 3D

Online shopping
(e.g. [ThreeKit](#))



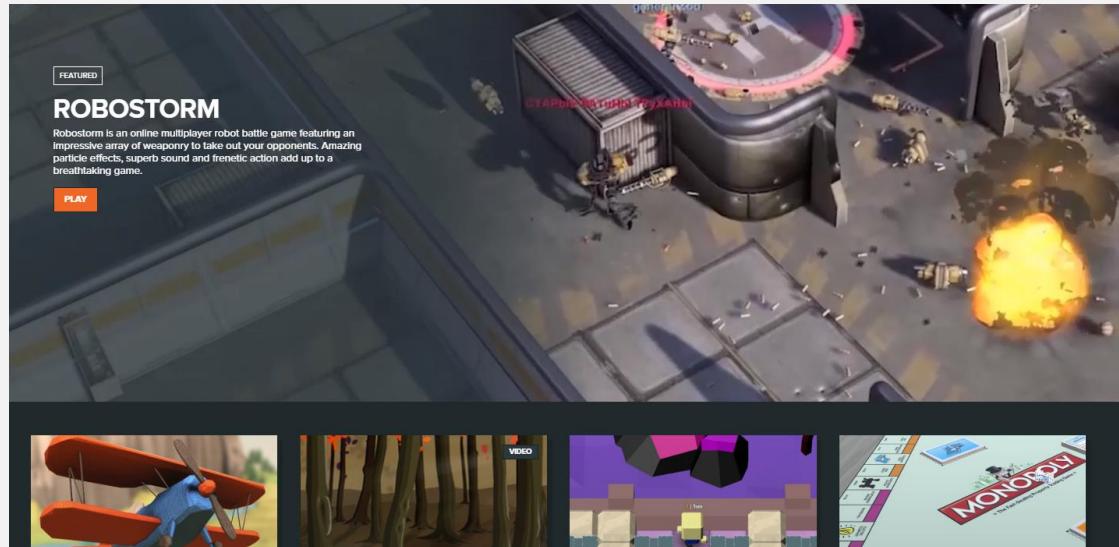
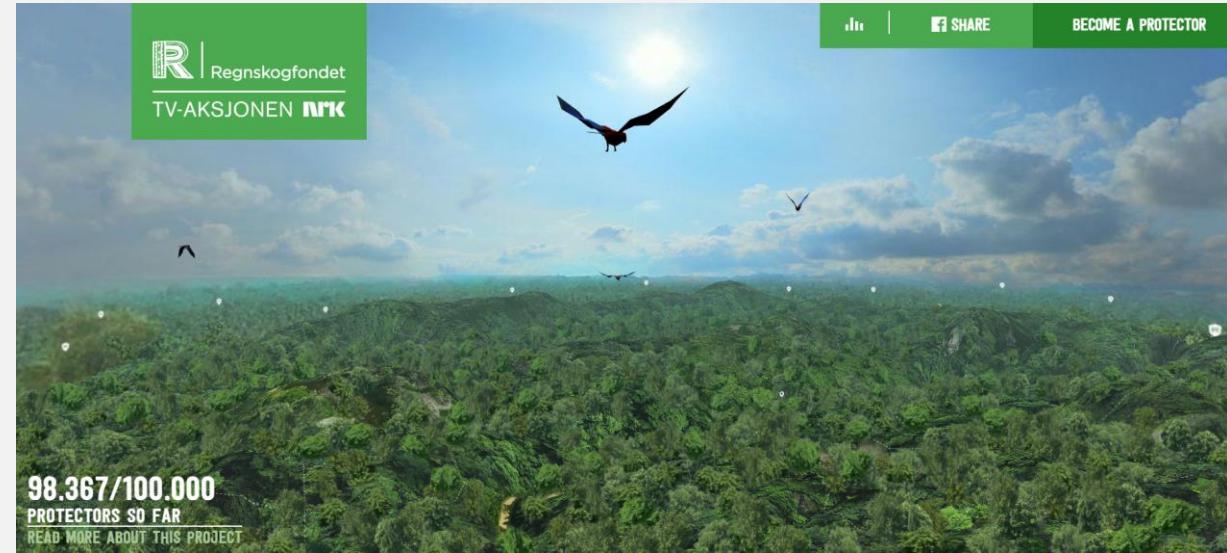
The screenshot shows the ThreeKit website. At the top, there's a dark header with the ThreeKit logo and a "schedule a demo" button. Below the header, a teal banner features the text "The future of **product visualization** is here." In the center, there's a paragraph about the impact of product visualization and a description of ThreeKit's technology. A "learn more" button is located below this text. To the right, there's a section showing a tablet, a smartphone, and a desktop monitor all displaying different 3D chair models. A speech bubble from a robot icon asks, "Hi there. Want to turn product visuals into great customer experiences?"



Architecture /
Interior design
(e.g. [Archilogic](#))

Web 3D

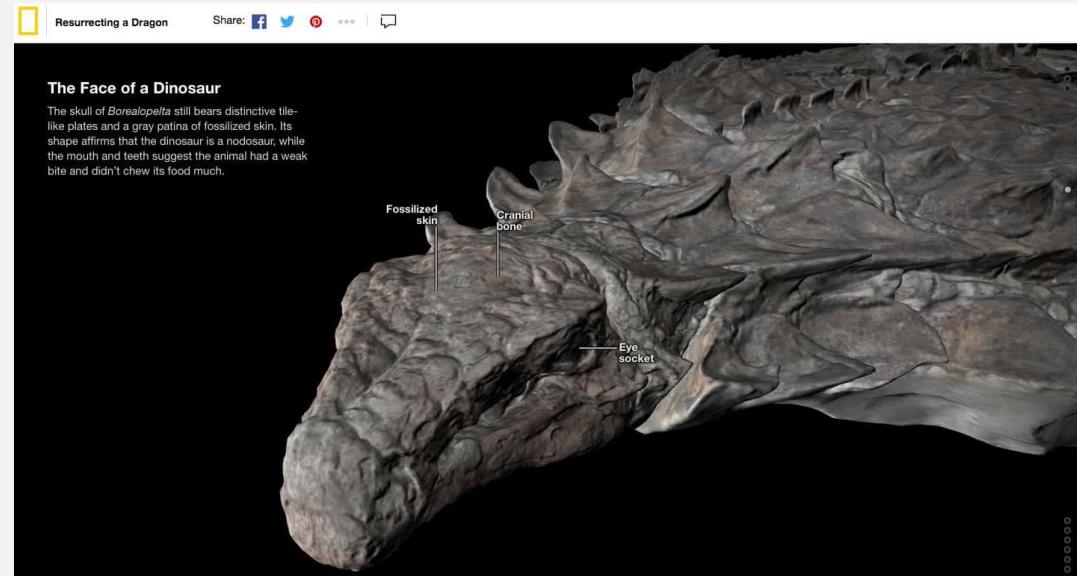
WebVR
(e.g. [Environment](#))



Game engines
(e.g. [PlayCanvas](#))

Web 3D

News
(e.g. [National Geographic](#))



FEATURED
ROBOSTORM
Robostorm is an online multiplayer robot battle game featuring an impressive array of weaponry to take out your opponents. Amazing particle effects, superb sound and frenetic action add up to a breathtaking game.
[PLAY](#)

The image shows a top-down view of a futuristic robot combat arena. A large, cylindrical industrial-looking structure dominates the center. In the foreground, a robot is engulfed in a massive fireball, indicating a recent explosion or hit. The ground is made of grey tiles, and there are various debris and smaller robots scattered around.

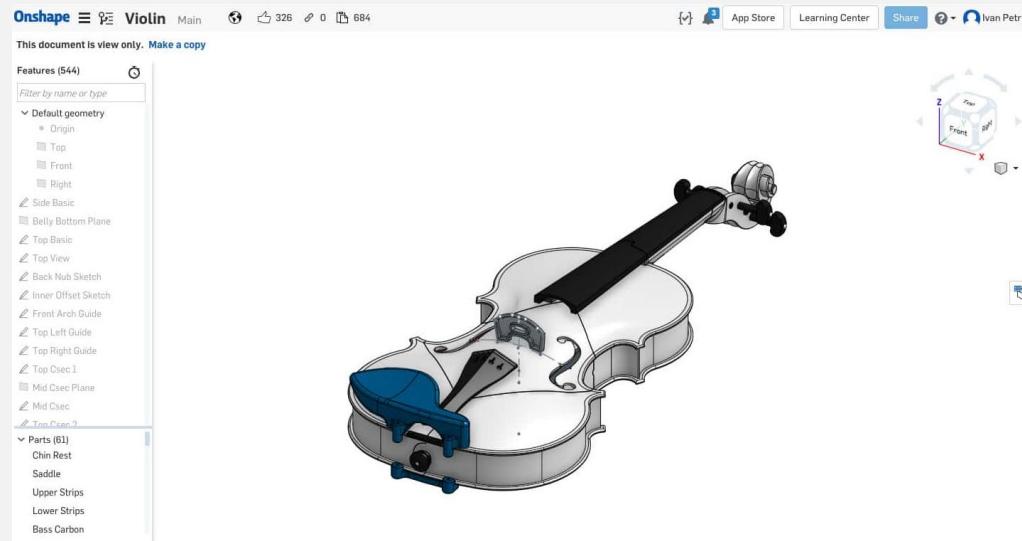
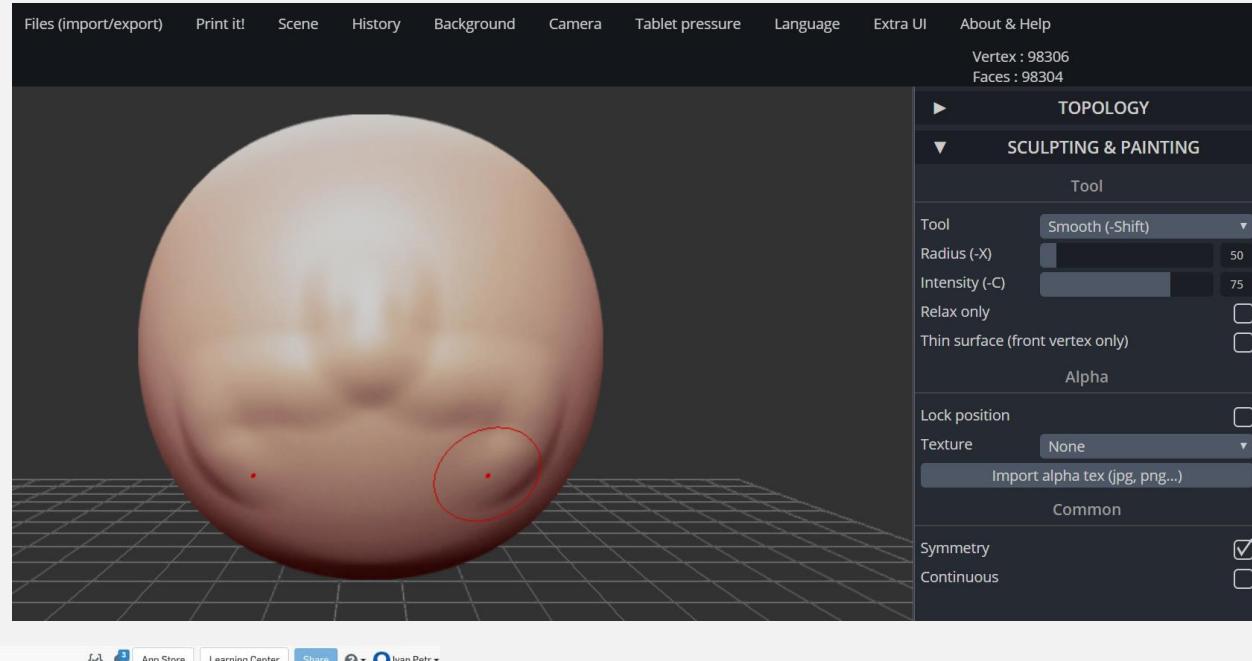
VIDEO

Below the main image, there are three smaller thumbnail images: a blue biplane, a forest scene, and a Monopoly board.

News
(e.g. [New York Times](#))

Web 3D

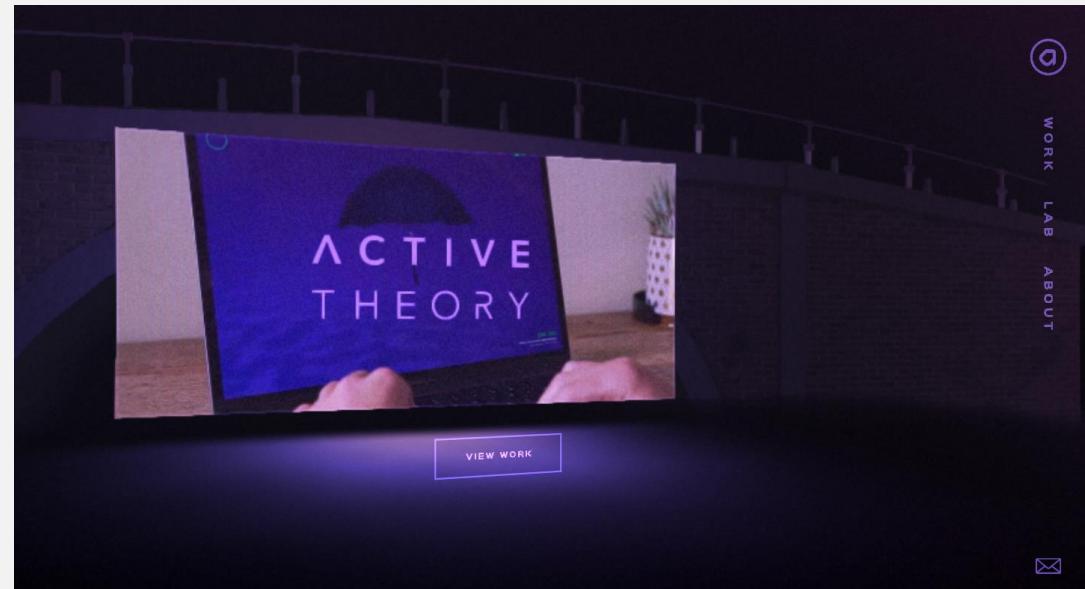
3D Modelling
(e.g. [SculptGL](#))



3D Sketching
(e.g. [Onshape](#))

Web 3D

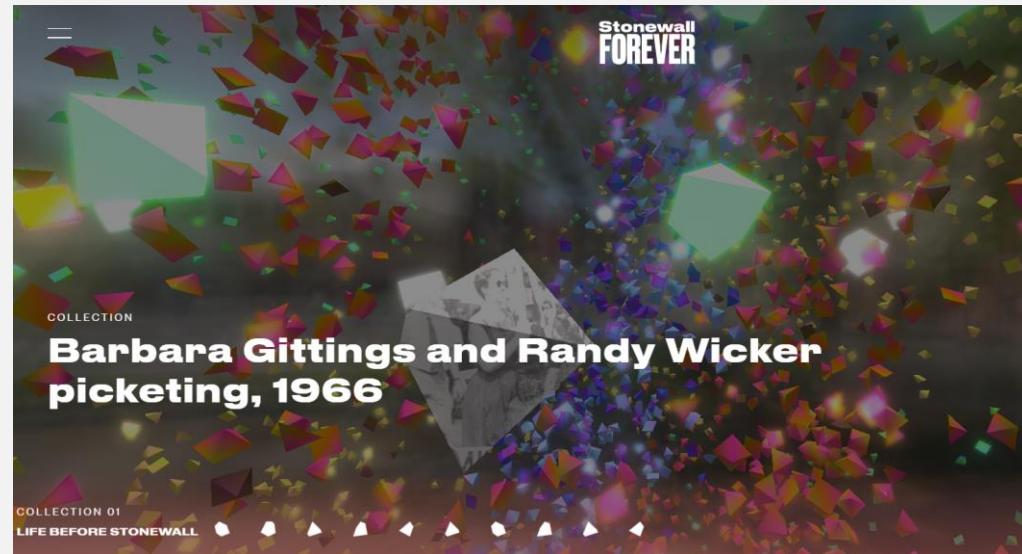
Personal websites
(e.g. [Portefolio](#))



Storytelling
(e.g. [Cross with Care](#))

Web 3D

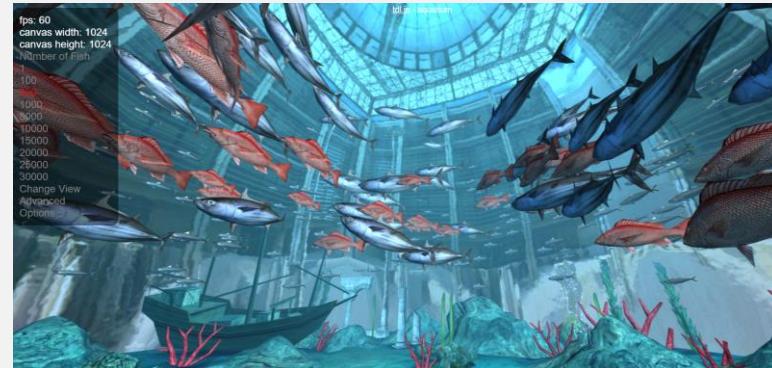
Product configuration
(e.g. [Autoneum](#))



Unconventional site navigation
(e.g. [Stonewall](#))

Other WebGL experiments

[Aquarium](#)



[HelloRacer](#)

[Curated list](#) (examples from 2010-2020)