

CSE 104 - Web Programming

General Organization

Personal presentation

Damien Rohmer - damien.rohmer@polytechnique.edu

Prof. of CS/DIX, Researcher at LIX

Geometric & Visual Computing - <https://www.lix.polytechnique.fr/geovic/>

Team in Computer Graphics & Computer Vision

2/13



Research area

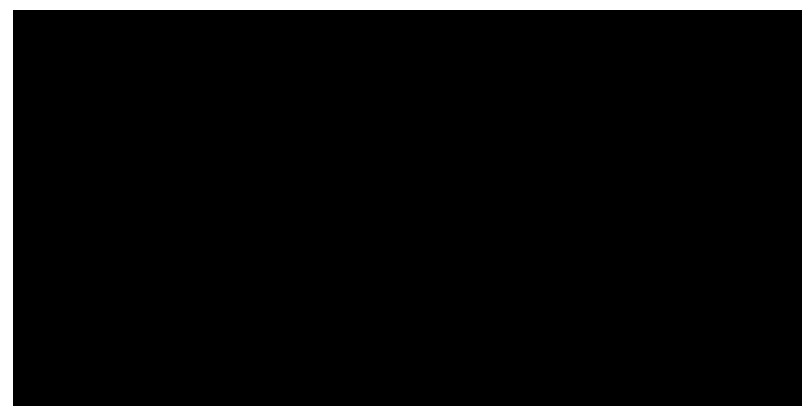
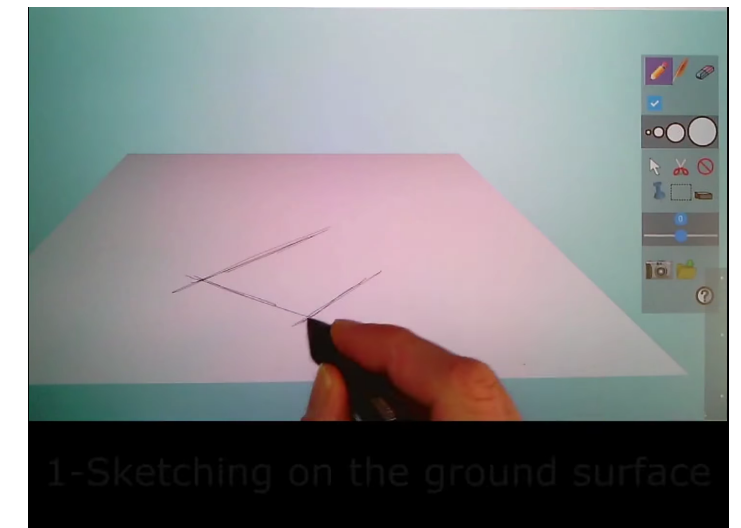
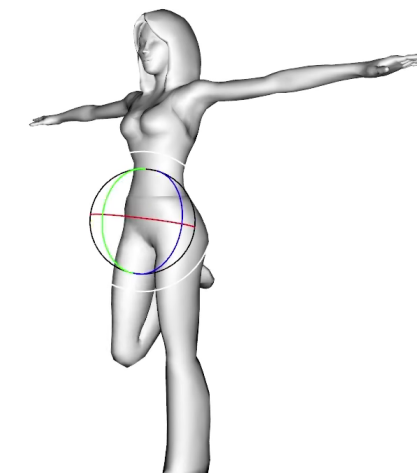
Develop new algorithms & method to ease and control 3D content

Interactive creation of 3D shapes

Designing 3D animated shapes & phenomenom.

Main applications:

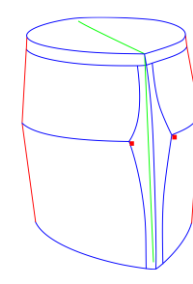
- Entertainment: Animation Cinema, Video Games, AR/VR
- Fabrication: Design & Prototyping
- Natural Sciences: Medical application



2D Input



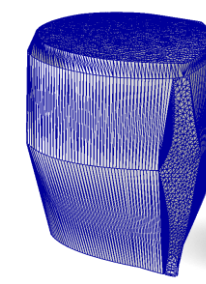
Inspiring image



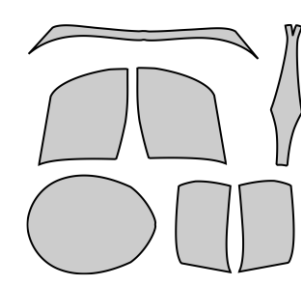
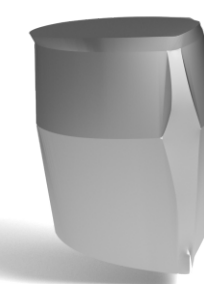
Sketch



Output



3D Developable Mesh



Patterns



Contacts

3/13

Damien Rohmer (teacher in charge)
damien.rohmer@polytechnique.edu

Fabio Roda - *Researcher Optimization/Industrial contracts X*
fabio.roda@polytechnique.edu

Jiayi Wei - *PhD student - Computer Graphics*
wei@lix.polytechnique.fr

Pablo Donato - *PhD Student - Computer assisted proofs*
donato@lix.polytechnique.fr

General objective of CSE104

4/13

- Be able to **create webpages from scratch** which are
 - Valid, standard compliant, responsive using **HTML and CSS**
 - Coding dynamic behavior of webpages using **JavaScript**
- Basic notions on server developement (PHP, nodeJS)
- Improve general computer development skills
 - General CS skills: Files handling
 - Good practices: Validity checking
 - Autonomy: Documentation research, Debug

What this class is not

5/13

- Not a network class

Focus on the use of language, not communication protocols

- Not a theoretical/conceptual class

Focus on practical development (few classes, lots of practice)

- Not a framework class (no bootstrap, jQuery, Django, Symfony, etc)

Focus on current language standard

- Not an advanced server side class (no login, security/hack, DB, etc)

Most time spent on client-side development, server development will be only an introductory level

Methodology

6/13

- Mostly **practical** computer lab class
 - Set of online tutorials and exercises
 - You code, develop, try, at your speed
 - You develop a small-project

⇒ Get your hands in the code !

- Some sparse general presentation before each new concepts
 - HTML/CSS
 - JavaScript
 - Server programming

Provisional schedule

7/13

- HTML/CSS , JavaScript , Server start with ~1h lecture.
- Schedule may be adapted

11/02 HTML/CSS 1

Lecture + Lab

18/02 HTML/CSS 2

25/02 (No Class)

04/03 HTML/CSS 3

11/03 JavaScript 1

*Lecture + **Test 1/HTML-CSS***

18/03 JavaScript 2

25/03 JavaScript 3

01/04 JavaScript 4

08/04 JavaScript 5

12/04 JavaScript 5

15/04 Project

Test 2/JS + Project

22/04 Project

29/04 (No Class)

06/05 Project

13/05 Server

Lecture + Lab

20/05 Server

27/05 Server

*Lab + **Test 3/Server***

Evaluation

8/13

- 3 graded tests in class (50%)
 - HTML/CSS (25%) [webpage to reproduce and/or quiz]
 - JavaScript (25%) [dynamic webpage to reproduce and/or quiz]
 - Sever (bonus/malus) [quiz]
- 1 project (50%)
 - Develop a dynamic website with HTML/CSS, JavaScript (client-side).
 - Work in pairs (or alone if you prefer)
- No final exam

Note: Tutorials and exercises are not aimed to be graded

- *Don't hesitate to ask questions*
- *Work on them seriously*
- *Make sure you understand it before going to the next one*

Beware of plagiarism in your project

9/13

- Web code is easy to share and reuse
- It is ok normal to share ideas and get inspiration on internet!
- It is **not ok** to use other's ideas and code **without citing it!**

Rule: If you inspire from

- Some internet website
- a book
- your colleagues, previous students
- ... (non exhaustive list)

⇒ **Cite them !** ⇐

Better to cite "too much" than not enough

Web languages

10/13

You can find *everything* on web development on the web !

- *everything* can be good - comprehensive, advanced notion.
- *everything* can be bad - outdated, non standard, overkill, dangerous practices.

⇒ Use internet resources, but wisely !

- Do not follow blindly every website that seems to have *a solution*
- Keep analyzing the code
- Double check comments and suggestions

Common issues: Out-date practices, Over-use of heavy library/frameworks

- HTML has evolved a lot through years
- Many *non developer* users (as long as it works)
- Old clever tricks may be replaced by simpler/cleaner standard code
CSS animation, flexbox, grid-layout, new JS standard, etc.

Programming rules (even in non web development)^{11/13}

1- Aim at **most simple readable** solution

- No tricky code (even if it works): un-readable => un-maintainable => un-reliable
- KISS principle (Keep It Simple, Stupid)
(Writing simple and readable code may not be simple!)

2- Prefer simple **standard** code rather than using existing libraries

Standard will be maintained through time, libraries may not

Libraries add extra cost (bandwidth, latency, maintainance, bugs, etc.)

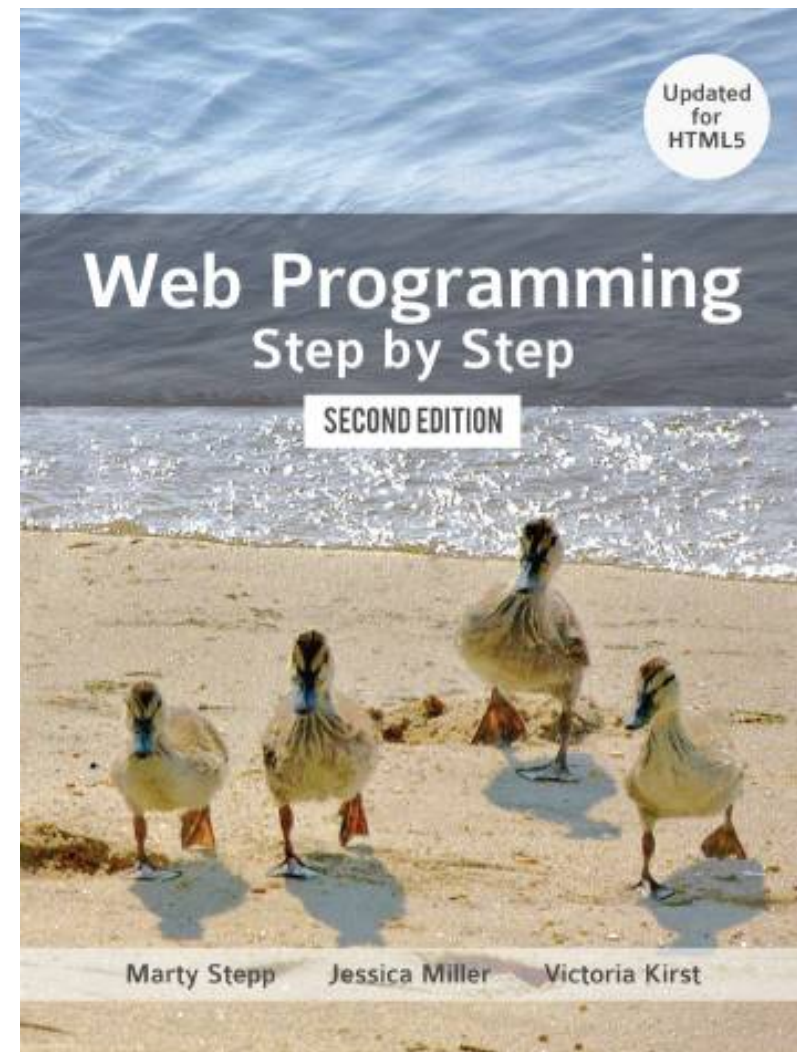
3- But prefer existing libraries to complex home made tricks

Do not re-invent the wheel for real projects (only to learn)

Main references

12/13

Victoria Kirst, CS193X, Web programming Fundamentals, Stanford.



Marty Stepp, Jessica Miller, Victoria Kirst

- In general: working on your own laptop should be fine on all system
 - You need a text editor (ex. [Visual Studio Code](#))
 - A modern Browser:
 - *Prefer Firefox or Chrome (better: try on both).*
 - *Some issues with Safari (late/incorrect standard implementation)*
- No need of specific softwares.
 - You will need Python, PHP, and Node.js for Server class (in May)

Warning : Don't trust too much your browser

- Web browsers try to hide your errors as much as they can
 - May looks ok on your browser/OS, but not with others*
- Always check (in particular your project) on different browsers and different screen resolutions.
- Check, if possible, with Chrome and Firefox
 - It is ok if IE/Safari doesn't work (didn't implement latest standard)*