

Synthèse d'images

TD1



Contenu du module

Déroulement

- 4 TDs
- 20h de TPs

Contenu

- Introduction à la synthèse d'images
 - Images en 3D type jeux vidéo, cinéma, etc.
 - Introduction des concepts généraux
 - Introduction aux problématiques propres à la 3D
 - Introduction à la mise en œuvre (implémentation)

Evaluation

Évaluation du module

- 3 Comptes rendus de TPs
 - Le premier portera sur les 4 premières séances (3 semaines, 8h)
 - Le second sur les 6h suivantes (avant le 4^e TD)
 - Le dernier en fin de module
 - Note finale : moyenne des 3 CRs

Évaluation du module

- Un compte rendu consiste en :
 - Le code qui sera remis via la plateforme GitLab <https://git.unistra.fr/>
 - Un rapport rédigé au format Markdown sur Git
 - Ce qui est attendu sera précisé en TP (variable en fonction de l'encadrant de TP)

Évaluation du module

- En ce qui concerne le code les outils seront présentés dans le premier TP
 - Pensez donc à vérifier le packaging de votre travail
 - Le code devrait être opérationnel out-of-the-box
 - cf. Cours sur la qualité : satisfaction du client

Introduction

C'est quoi la synthèse d'images

- Définition

- générer des images à l'aide de l'ordinateur
- images créées à partir de données 3D

C'est quoi la synthèse d'images

- Définition
- Concrètement : les domaines d'utilisation
 - Omniprésent dans la vie quotidienne
 - Secteur économique actif et diversifié

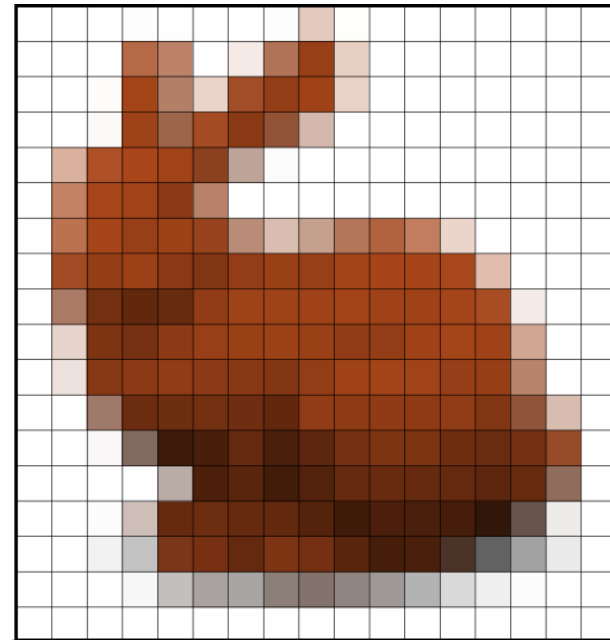
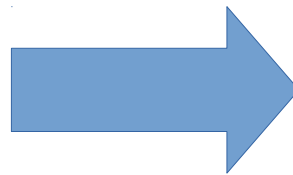
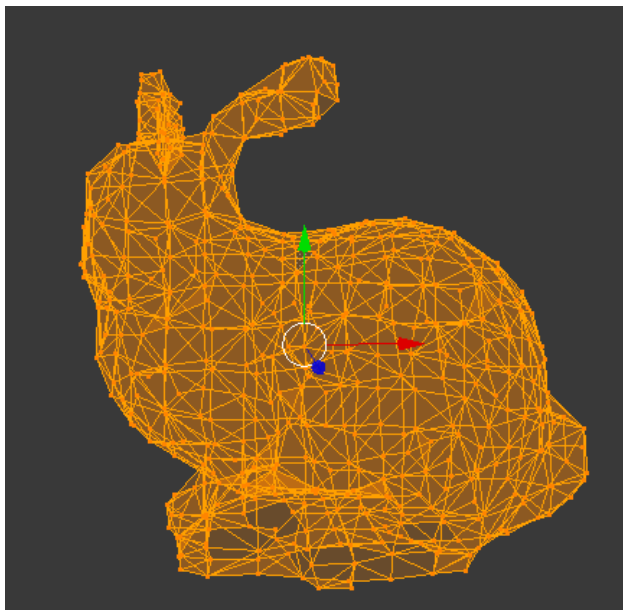
Contenu du module

- Introduction à la programmation 3D RT
 - Présentation des concepts
 - Contraintes temps réel
- Mise en pratique (TPs) WebGL (three.js)
 - Javascript, langage interprété pas de compilation

Rendu

Rendu (Rendering)

- Definition : Processus de génération d'une image 2D à partir d'une scène 3D



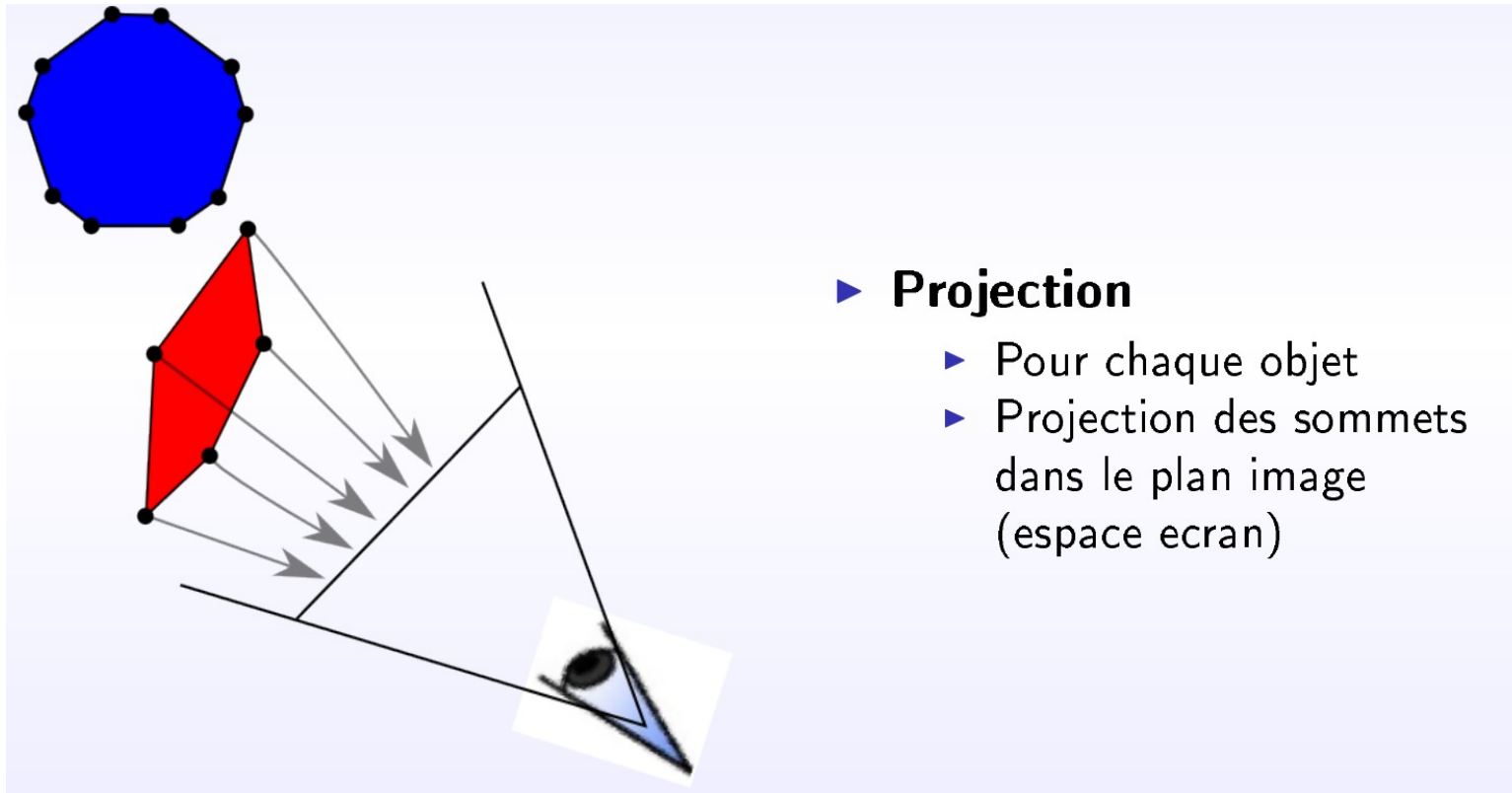
La caméra

- Anglicisme pour définir le point de vue de l'observateur
 - Position de l'observateur
 - Emplacement de l'appareil photo
 - Ou de la caméra vidéo
- Caractéristiques physiques
 - Lentilles, FOV, distorsion, aberrations chromatiques

- Technique actuelle utilisées dans les GPUs
 - Projection des segments et triangles sur le plan de l'image

Rastérisation

- Technique actuelle utilisées dans les GPUs
 - Projection des segments et triangles sur le plan de l'image

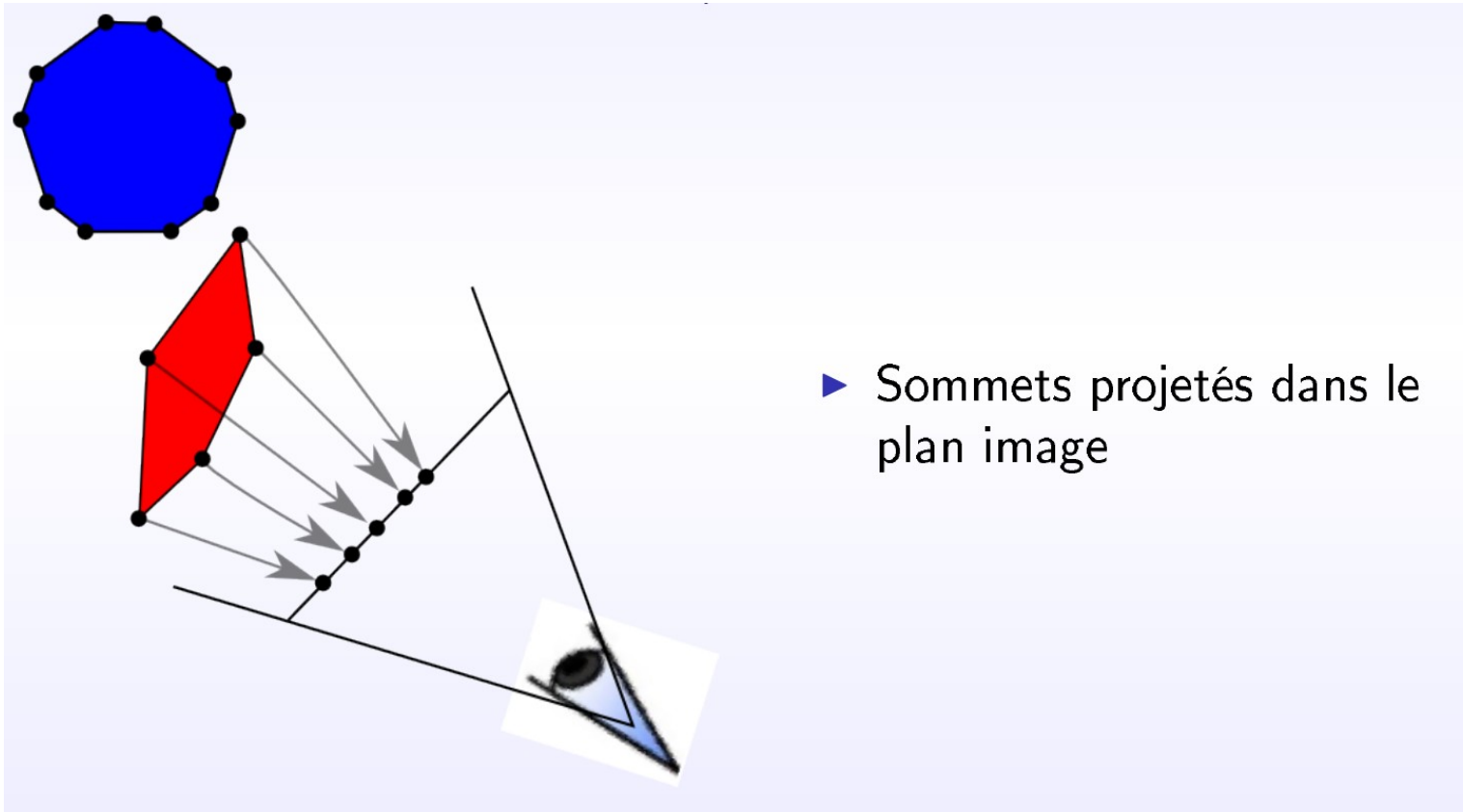


► Projection

- Pour chaque objet
- Projection des sommets dans le plan image (espace écran)

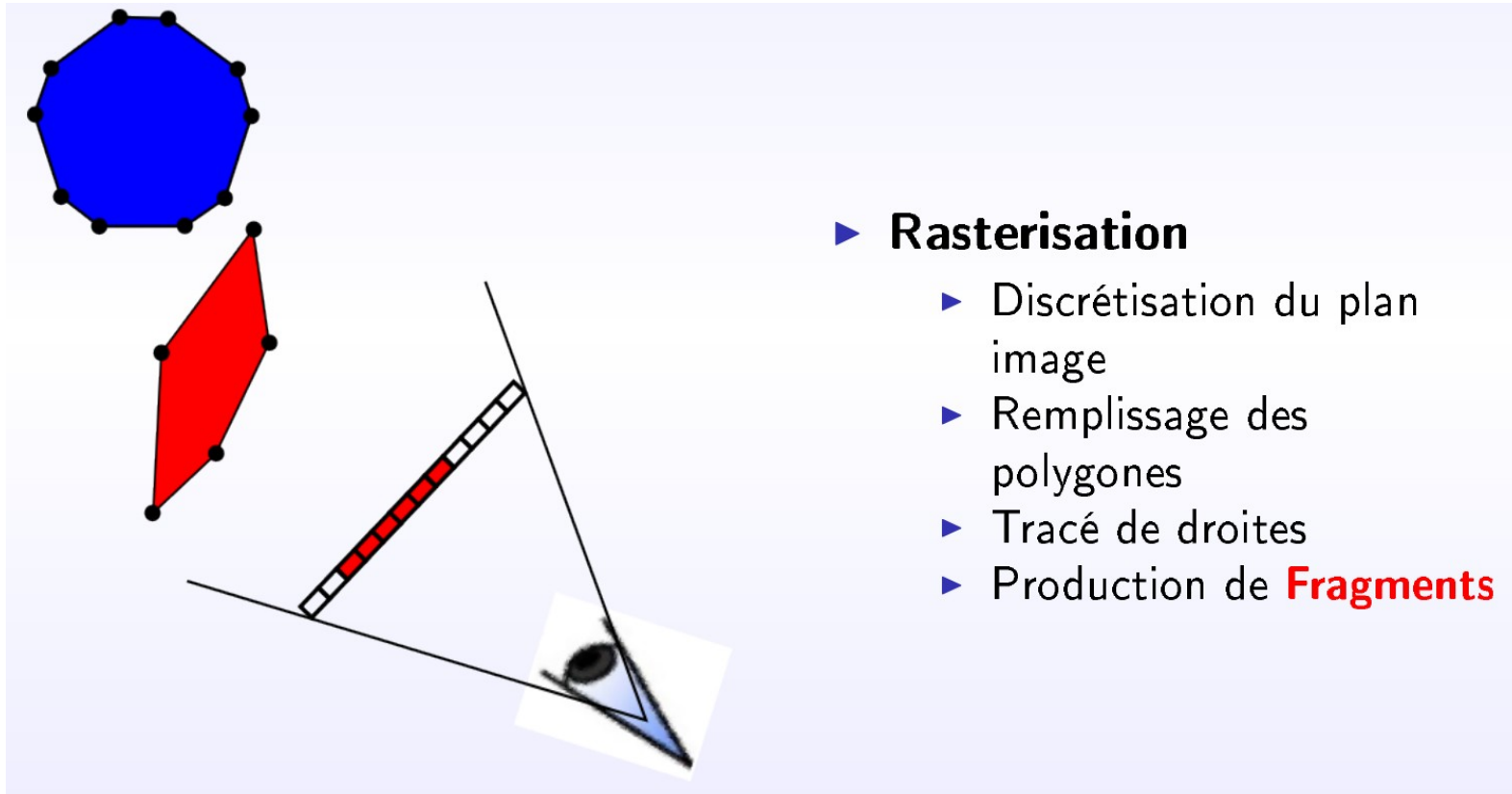
Rastérisation

- Technique actuelle utilisées dans les GPUs
 - Projection des segments et triangles sur le plan de l'image



Rastérisation

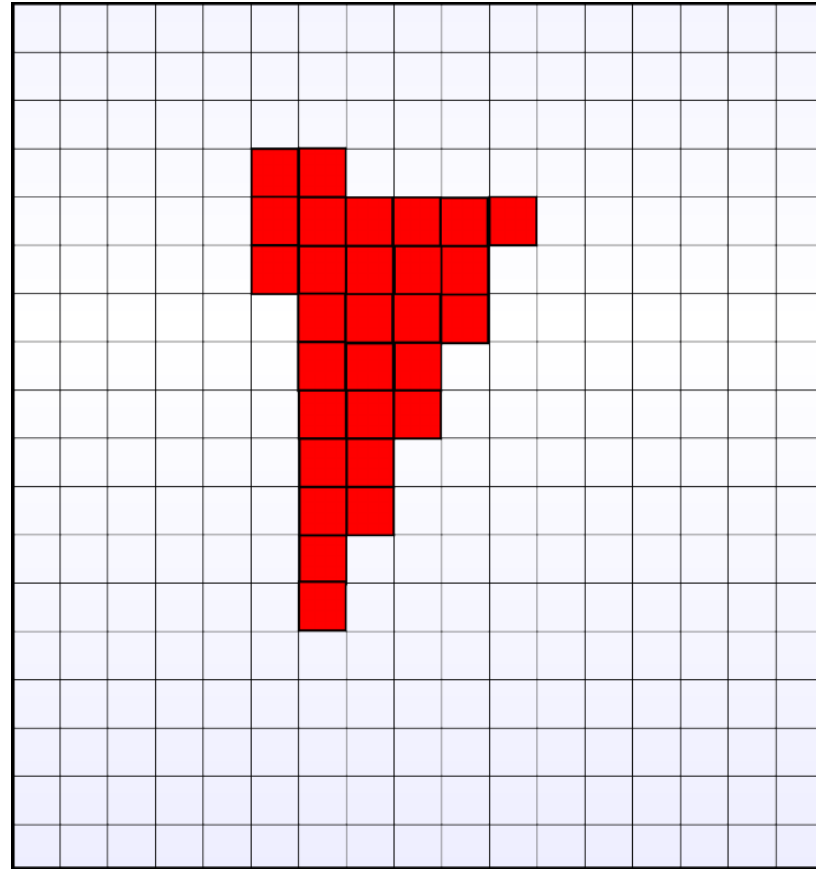
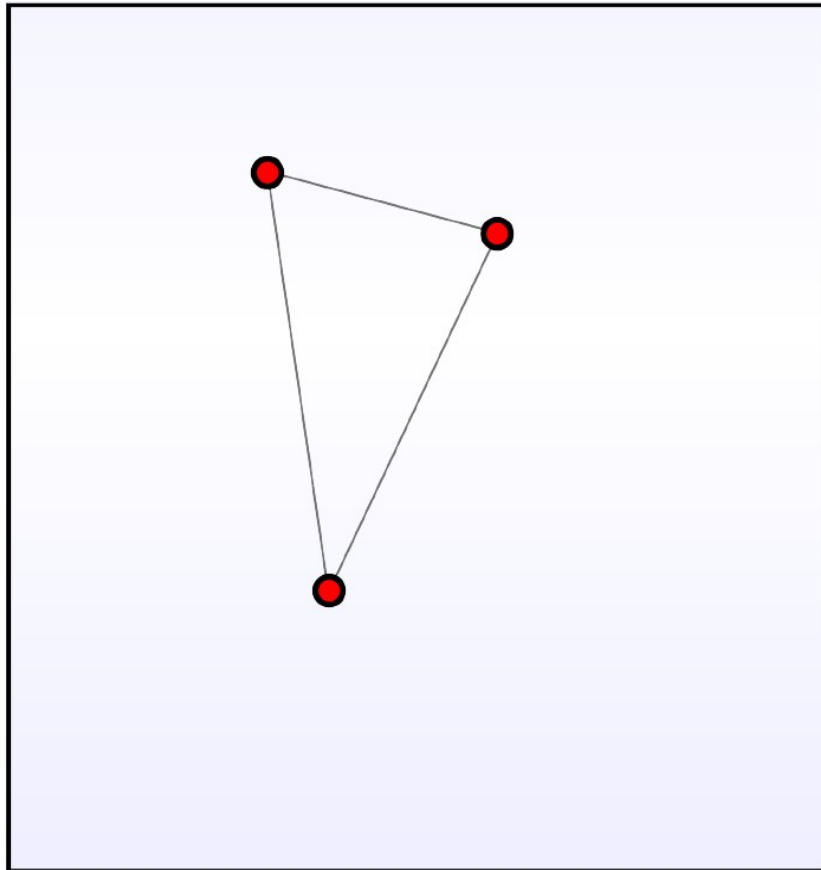
- Technique actuelle utilisées dans les GPUs
 - Projection des segments et triangles sur le plan de l'image



Rastérisation

Illustration pour un triangle

– Discrétisation par pixels



Avantages et inconvénients

- Simple à comprendre, donc à implémenter, en software ET en hardware
- Plus facilement parallélisable :
 - l'architecture des GPUs actuels en est la conséquence
- Algorithme moins gourmand en puissance de calcul que le lancer de rayons (*raytracing*, cf. annexe), mais plus limité
 - Utilisation « d'astuces » pour émuler certains phénomènes
 - Réflexions
 - Illumination indirecte
 - ...

Scène

Notion de scène

- Description « virtuelle » du monde à représenter
 - À la manière d'une mise en scène cinématographique

Contenu d'une scène

- Objets
 - Décors
 - Personnages
- Sources de lumière
- Observateur (caméra)

Notion de scène

- Description « virtuelle » du monde à représenter
 - À la manière d'une mise en scène cinématographique
 - Il faut pouvoir placer les différents éléments dans la scène
- ▶ Repérage et coordonnées

Système de coordonnées

- Monde réel = $\mathbb{R}^3 + t$
- Repère en coordonnées cartésiennes
- C'est un espace vectoriel
 - On peut utiliser les outils mathématiques de calcul vectoriel (cf. M12, M4x)
 - C'est un repère orienté

Objets 3D

- **Forme (Shape/Mesh) :**
 - Éléments de géométrie
 - Équation implicite
 - Maillages
 - Données relevées
- **Aspect (Shading) :**
 - Couleurs
 - Textures
 - Fonctions de réflectance

Application pratique : Présentation

- APIs :
 - OpenGL, Direct3D (MS), Metal (Apple), Vulkan (Khronos)
- Khronos Group
 - Consortium de spécification d'API
 - Open*(GL , GL ES, CL, ...), WebGL

Moteurs 3D

- Apportent les outils de productivité au-dessus des APIs bas-niveau
- Concentration du marché :
 - Niveau d'expertise élevé requis
 - Middleware (Unreal Engine, Unity 3D, ...)
- Three.js
 - Middleware javascript

Three.js

- Simple d'accès
 - Pas besoin d'apprendre à utiliser un IDE
- Pas de compilation
 - Javascript interprété
- Fonctionne dans tous les navigateurs modernes



Bon compromis pour une introduction

Application pratique : Environnement de travail

Outils

- Programmation Javascript
 - Éditeur de texte
- Navigateur supportant Javascript et WebGL
 - Chrome, Firefox
 - Utilisez un débogueur

Éditeur de texte

- Utilisez un éditeur de texte avancé :
 - Coloration syntaxique de plusieurs langages
 - (html, css, JS)
 - Augmentez votre productivité en apprenant les raccourcis clavier
 - Si c'est un IDE c'est encore mieux !

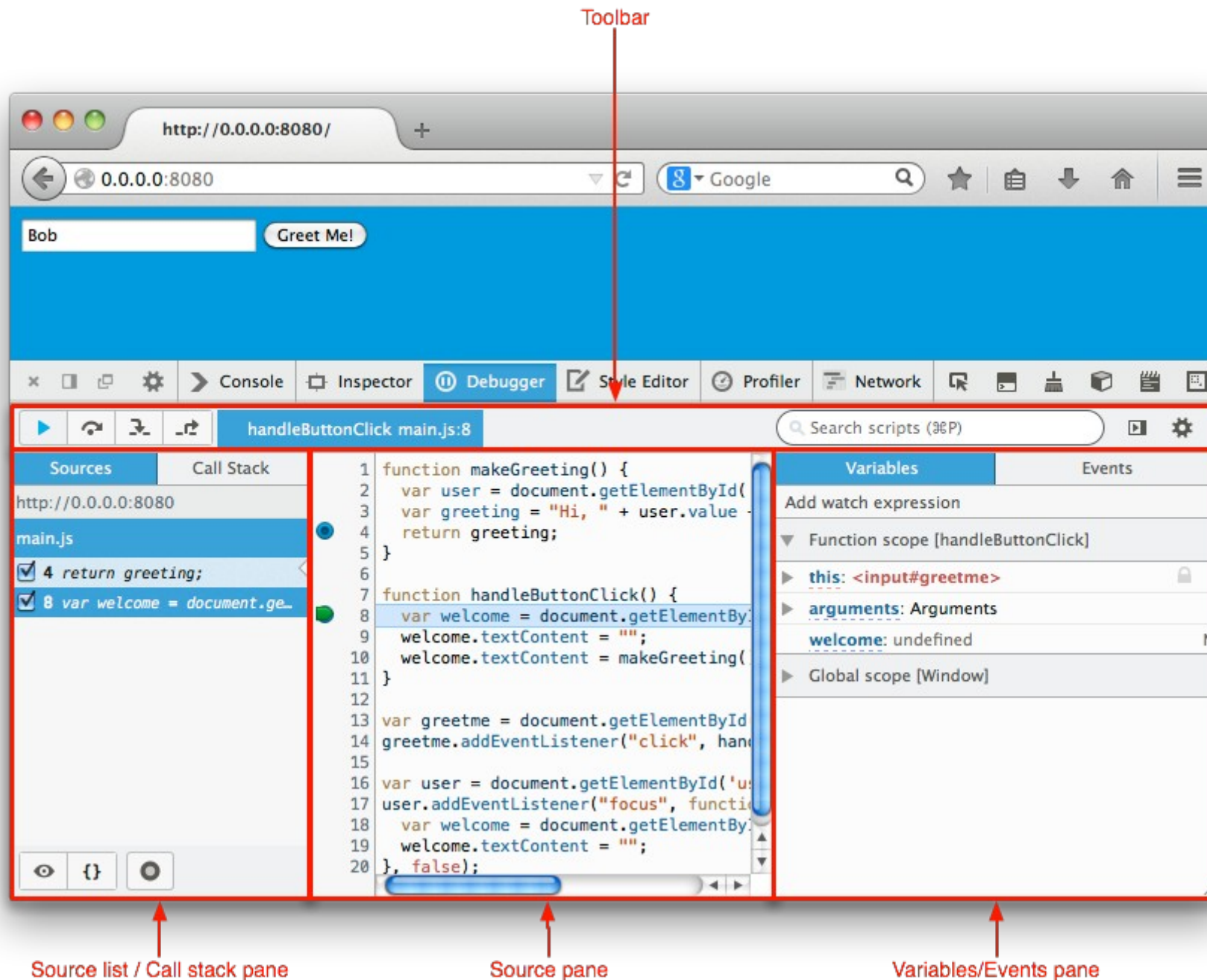
Débogueur

- Outil indispensable pour tout développeur qui se respecte :
 - Fini le « printf debugging »
 - Il existe des outils pour voir le contenu des variables en mémoire
 - On arrête le massacre, et on travaille professionnellement et efficacement

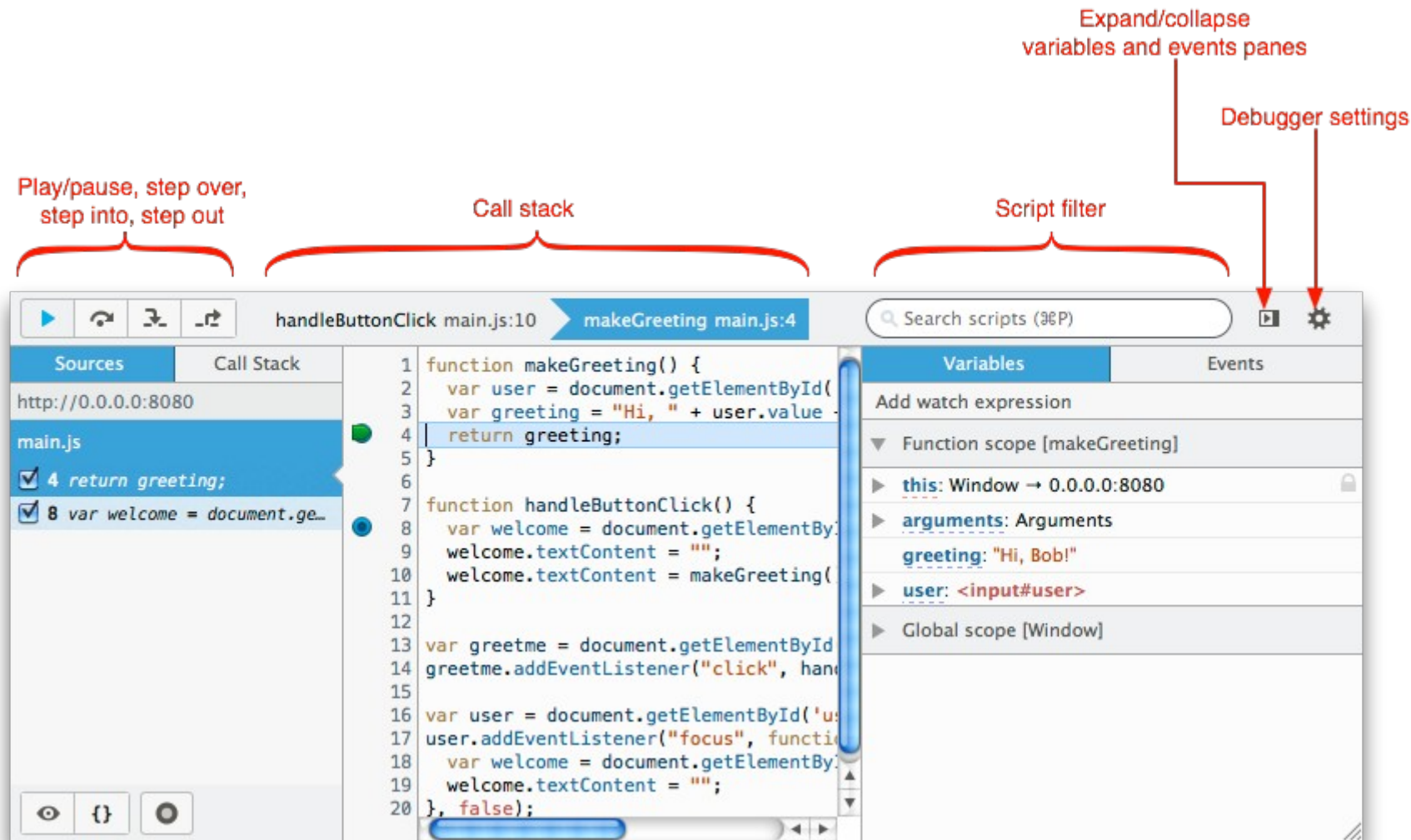
Débogueur : Firefox

- Tools - Web Developer – Debugger (Ctrl+Shift+S)
- Des extensions existent également
- Documentation :
 - Prenez le temps de la lire :
<https://developer.mozilla.org/fr/docs/Outils/D%C3%A9bogueur>

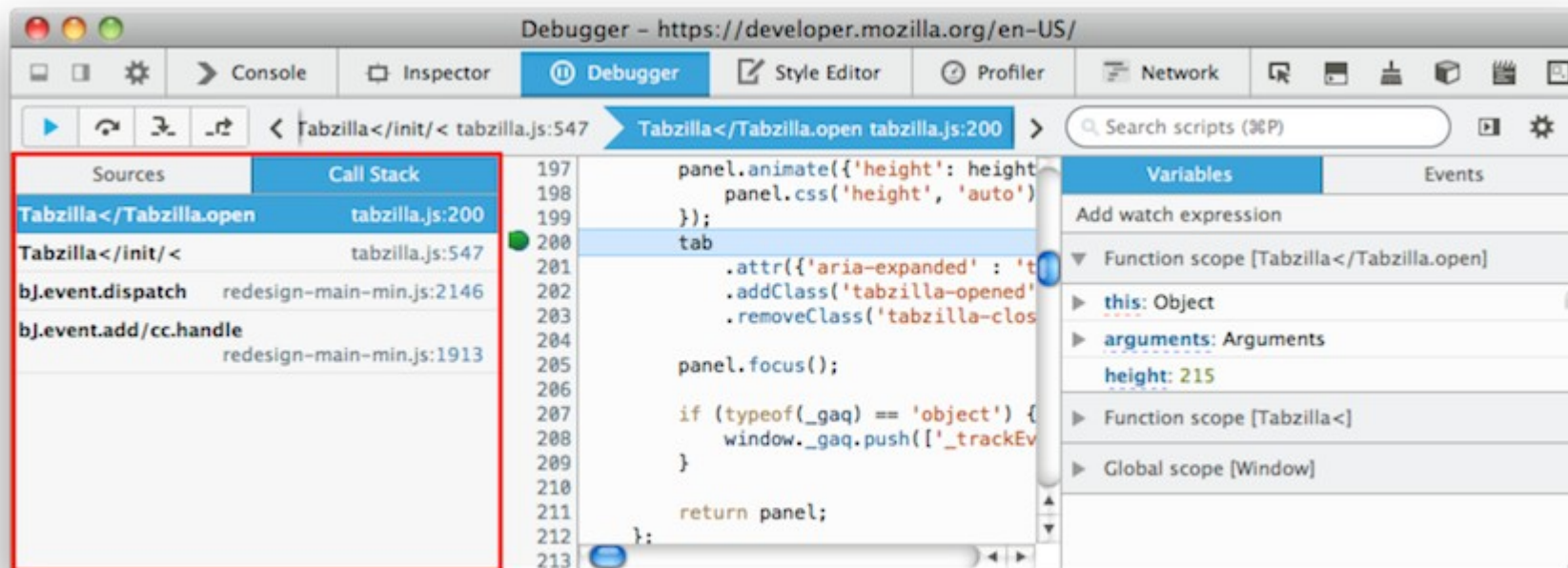
Débogueur : Firefox



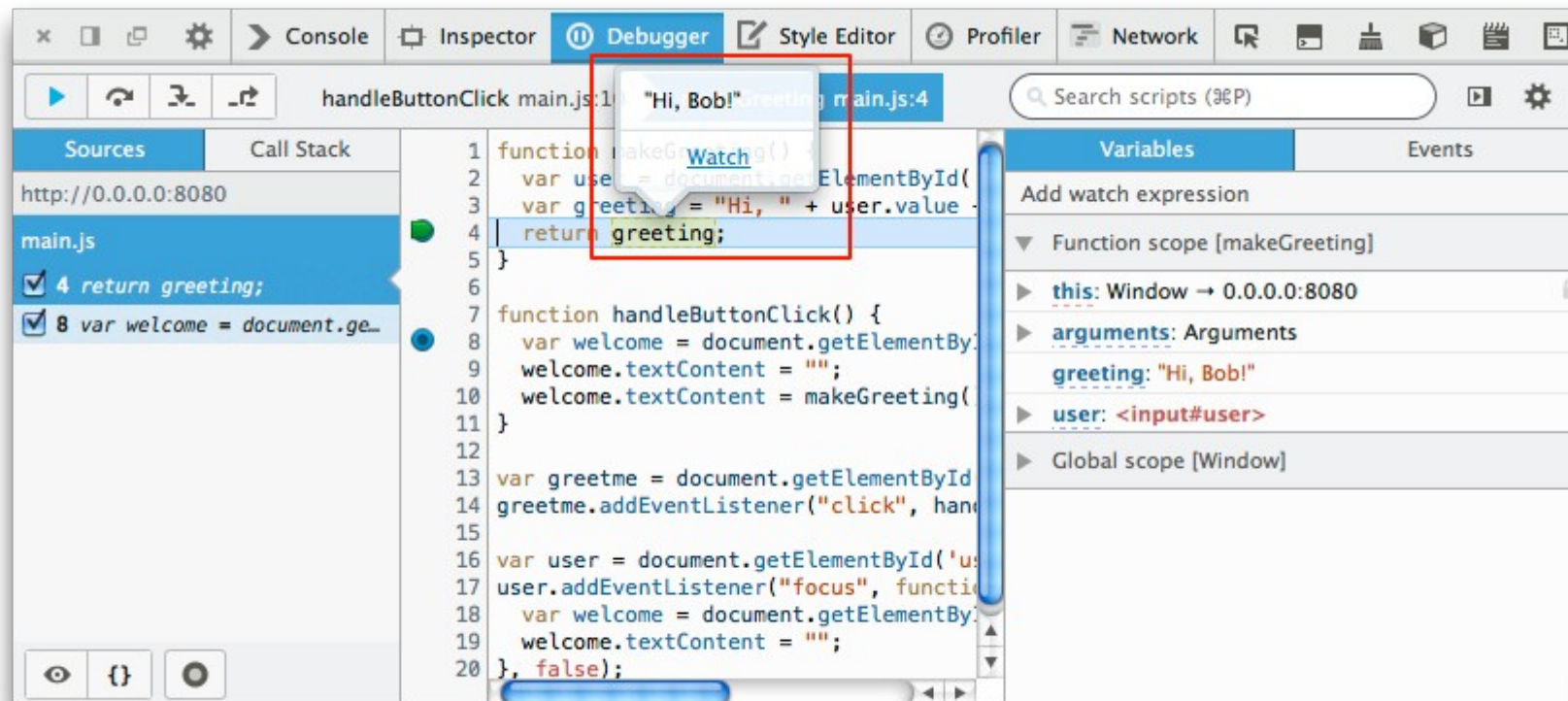
Débogueur : Firefox



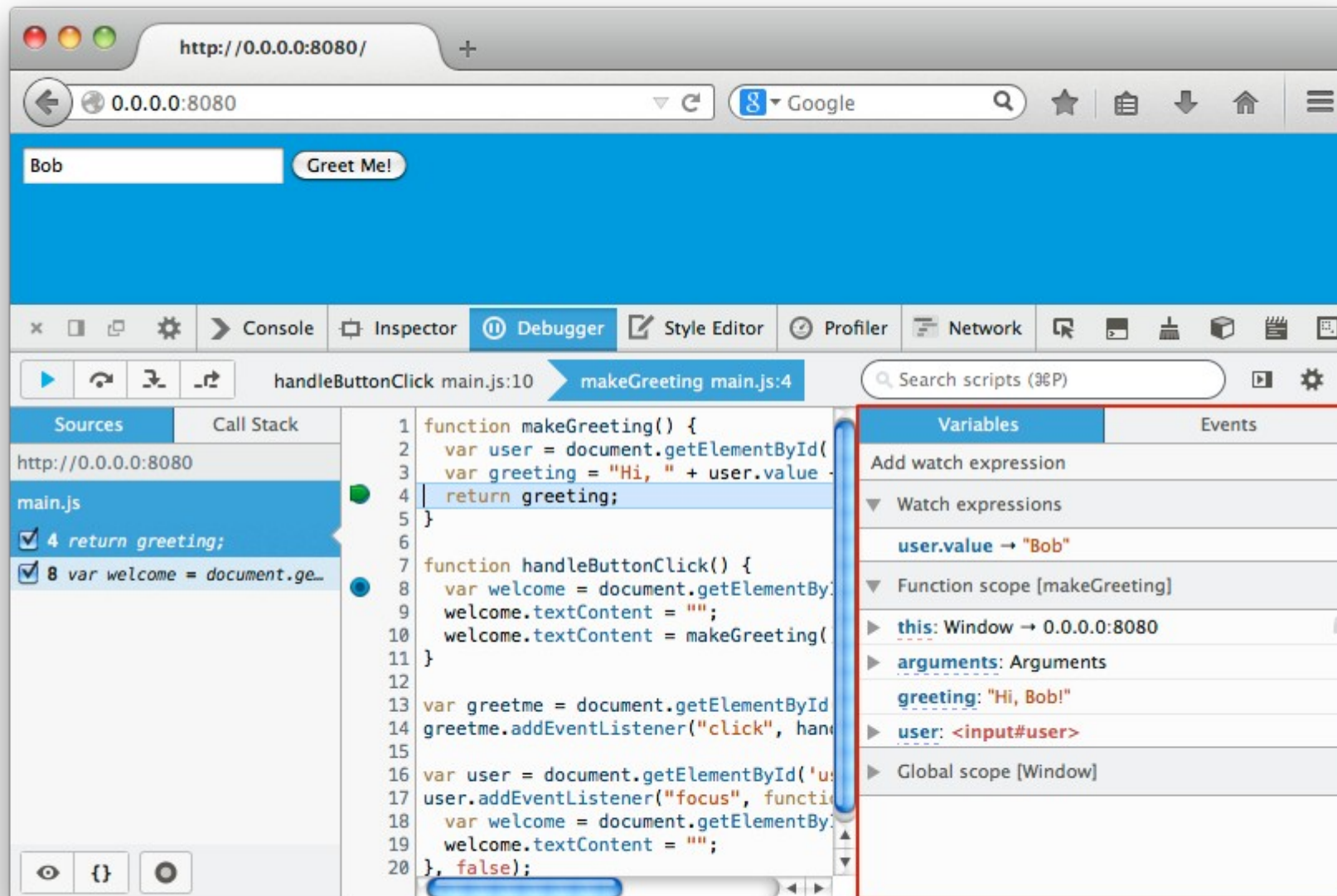
Débogueur : Firefox



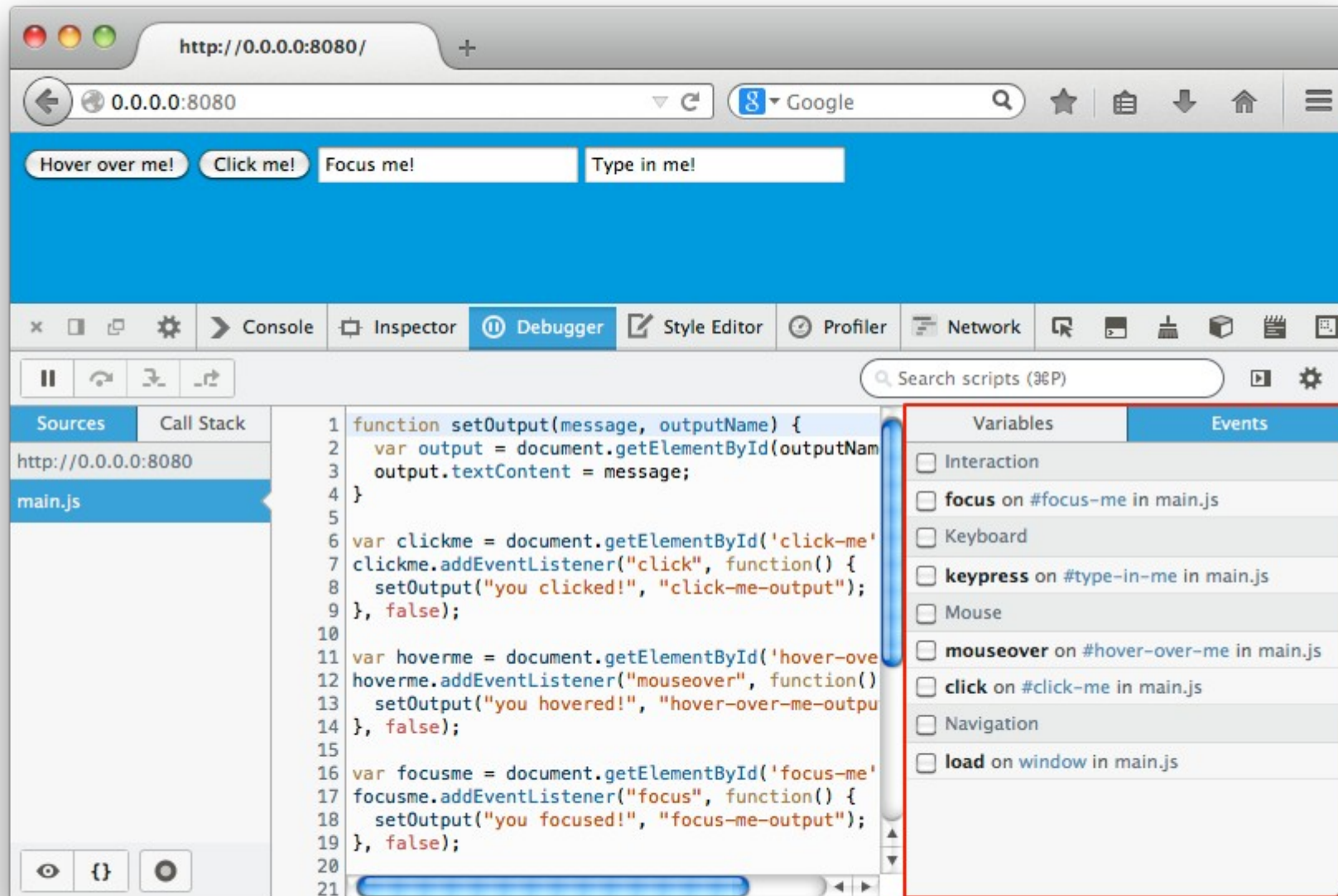
Débogueur : Firefox



Débogueur : Firefox



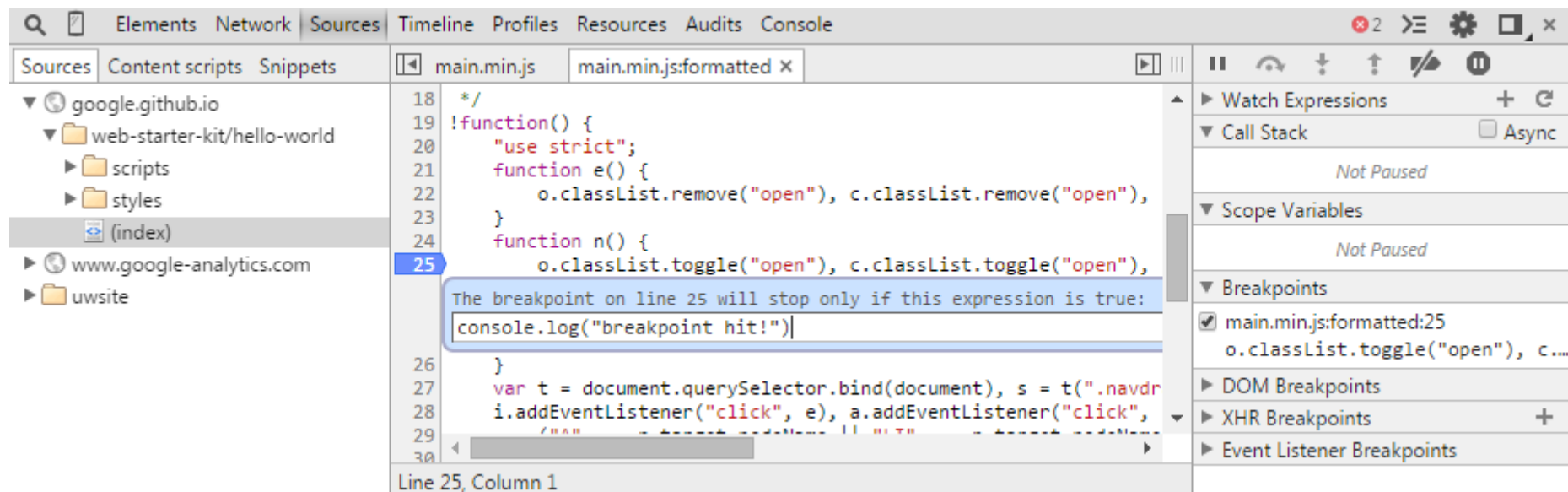
Débogueur : Firefox



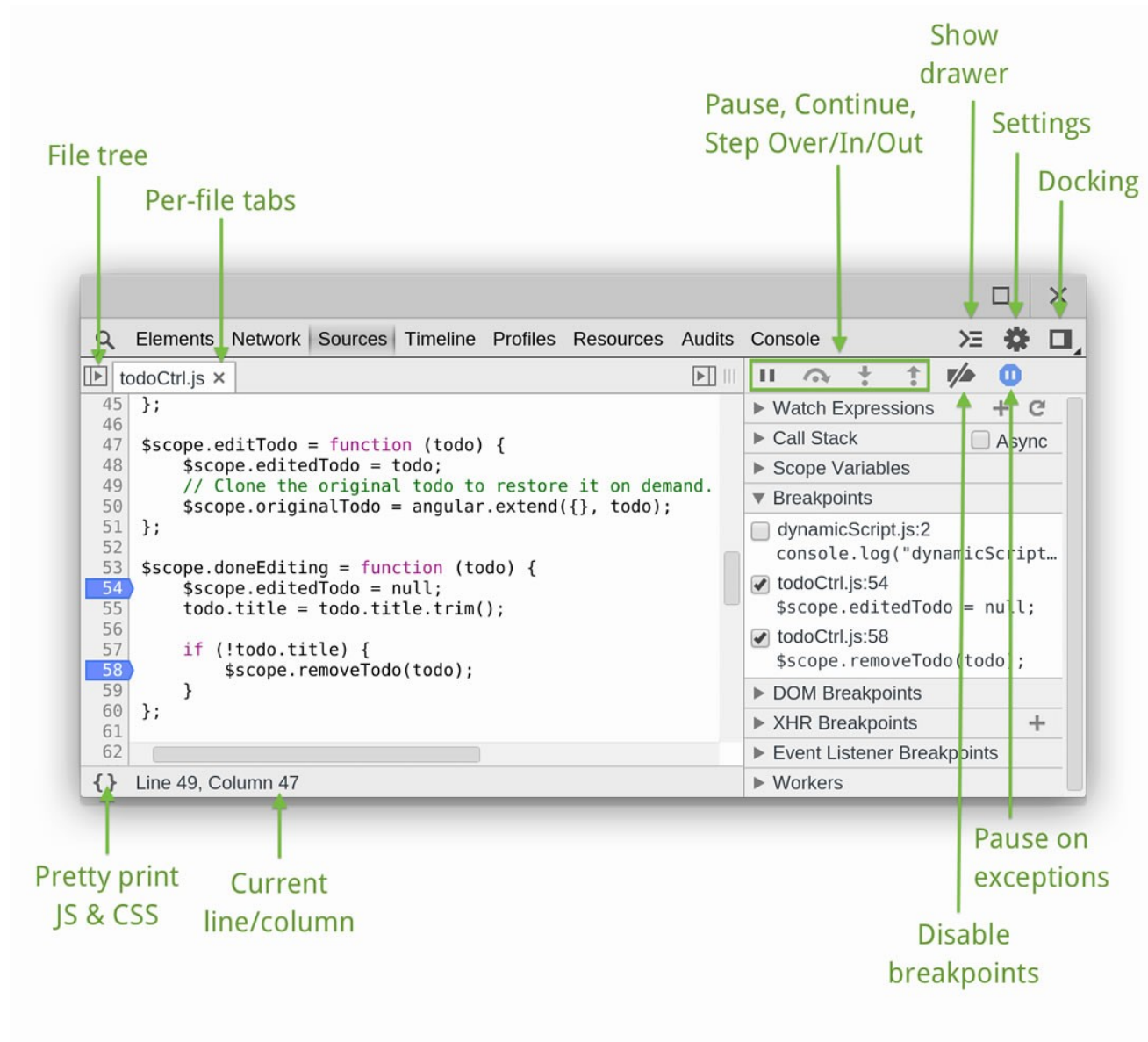
Débogueur : Chrome

- Chrome menu - Tools > Developer Tools
- Ou Plus d'outils – Outils de développement
- Ctrl+Shift+I
- Documentation :
 - <https://developer.chrome.com/devtools>
 - <https://developer.chrome.com/devtools/docs/javascript-debugging>

Débogueur : Chrome



Débogueur : Chrome



Débogueur : les grands principes

- Vérifier le contenu des variables, de la mémoire
- Suivre l'exécution pas à pas et la pile d'appel
- Comprendre les problèmes et les résoudre
 - On arrête de corriger les bugs au « pif » !

Application pratique

Méthodologie

Méthodologie

- Le faux problème : la compétence technique
 - Comment réaliser le TP
 - Avec un éditeur de texte
 - Du Javascript
 - Des outils
- Solutions :
 - Documentation
 - Recherche internet
 - Questions aux camarades ou au prof

Méthodologie

- Le vrai problème : la méthode
 - Comment arriver au bout du TP
 - Identifier les sous-étapes
 - Les accomplir une par une
 - Revérifier le bien fondé des étapes définies après chaque réalisation
- Solutions :
 - Analyse
 - Réflexion
 - Prototypage
 - Essais

Méthodologie

- Créativité et technicité
 - la créativité est dans la conception de la solution
 - On doit imaginer comment arriver au but, et toutes les étapes pour s'y rendre
 - Les limites techniques limitent la créativité
 - Implémenter consiste juste à matérialiser cette réflexion préalable grâce à la technique

- Cheminement de la pensée
 - Identifier les étapes
 - Subdiviser les étapes jusqu'à atteindre des étapes dont on voit clairement le début et la fin
 - Une partie des étapes peut ne pas être complètement définie sans empêcher de commencer à travailler
 - Cela permet déjà d'aller dans la bonne direction
 - Étape = micro-commit !

Application pratique Gestion de versions avec Git

Gestion de versions

- Conserver un historique du code et des modifications apportées
- Permettre de revenir en arrière sur des versions antérieures
- Permettre de partager le code entre plusieurs développeurs

Gestion de versions

- Repository
- Version de travail (Update/Checkout)
- Commit
- Partage (Push/Pull)

Git

- TortoiseGit (Windows)
- Gitk + git gui (Linux)
- SourceTree (win/mac)

- <https://git.unistra.fr/>
- Serveur pour partager vos travaux via git

Annexe : Réalité Virtuelle

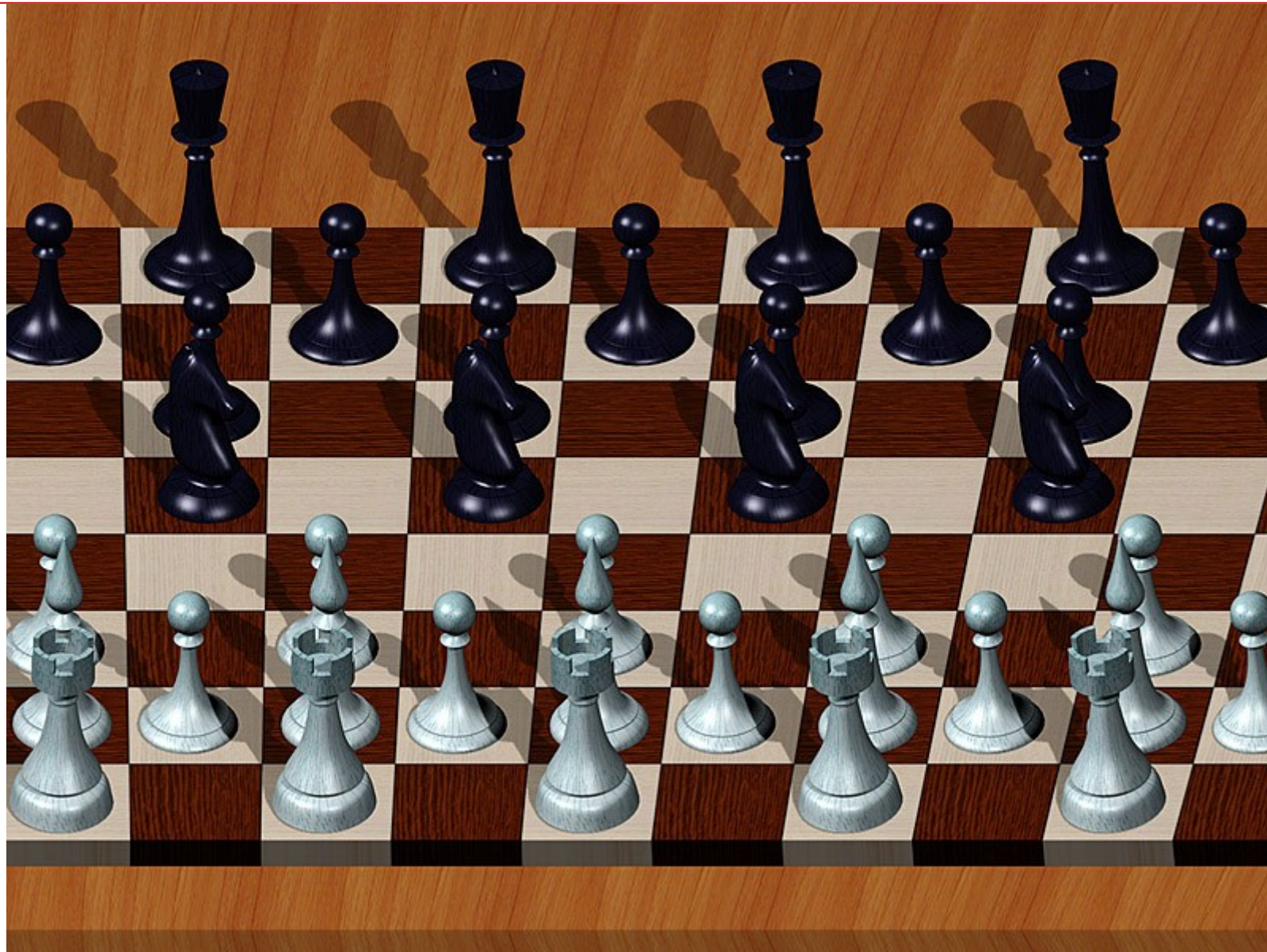
La perception du volume

- 2 façons principales d'appréhender le monde en relief :
 - 2 images légèrement différentes (2 yeux)
 - Perception physiologique du relief
 - Jeu d'ombre et de lumière : connaissance acquise par l'expérience, le vécu
 - Illusion d'optique possible

Anaglyphe



Stéréogramme



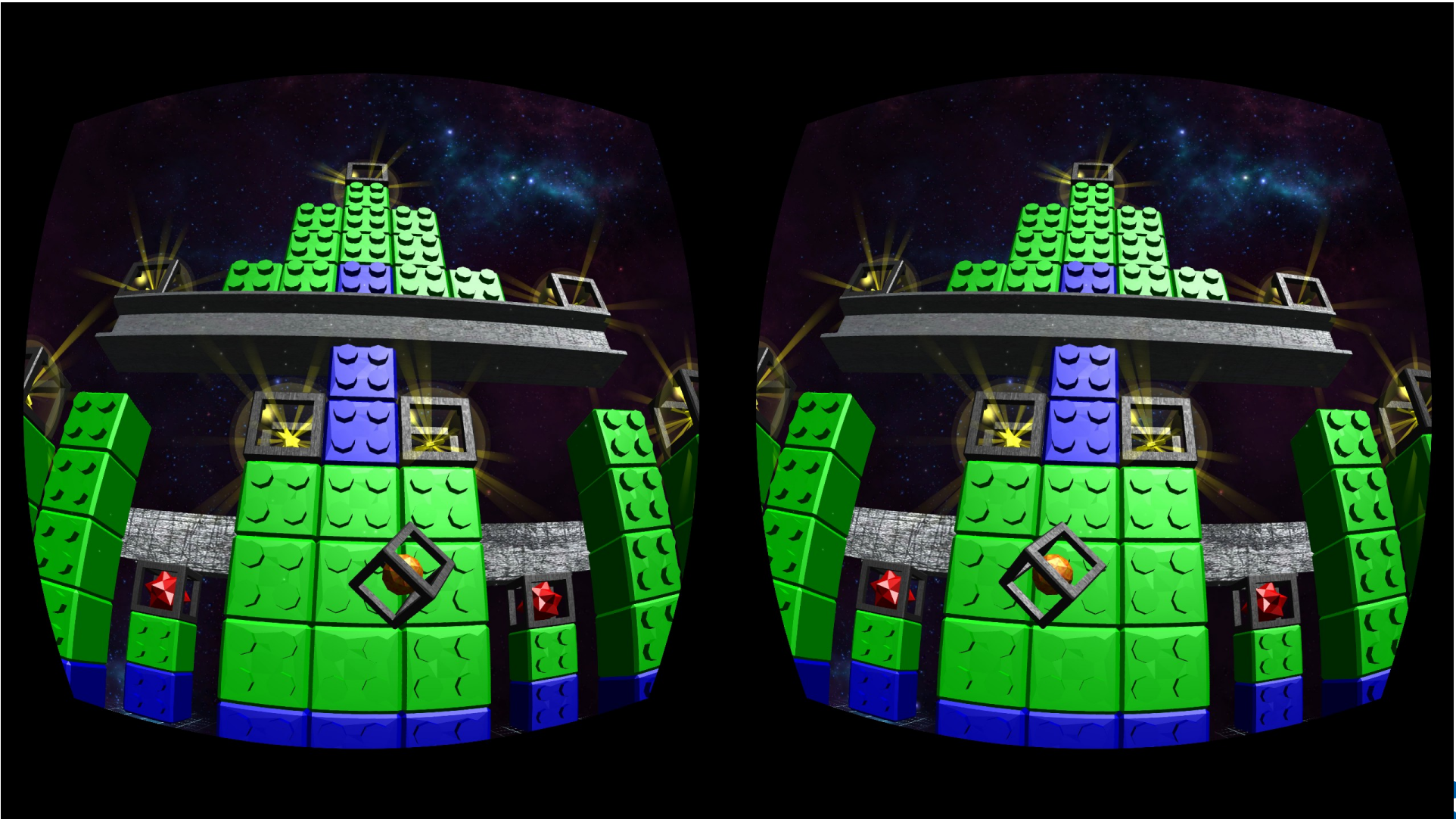
(source : wikipedia)

Rendu stéréo

- Moniteur 3D (passif ou actif), Cinéma
 - Générer une image légèrement différente pour chaque œil
- Casque VR
 - Idem + déformation pour compenser la lentille et le champ de vision



VR

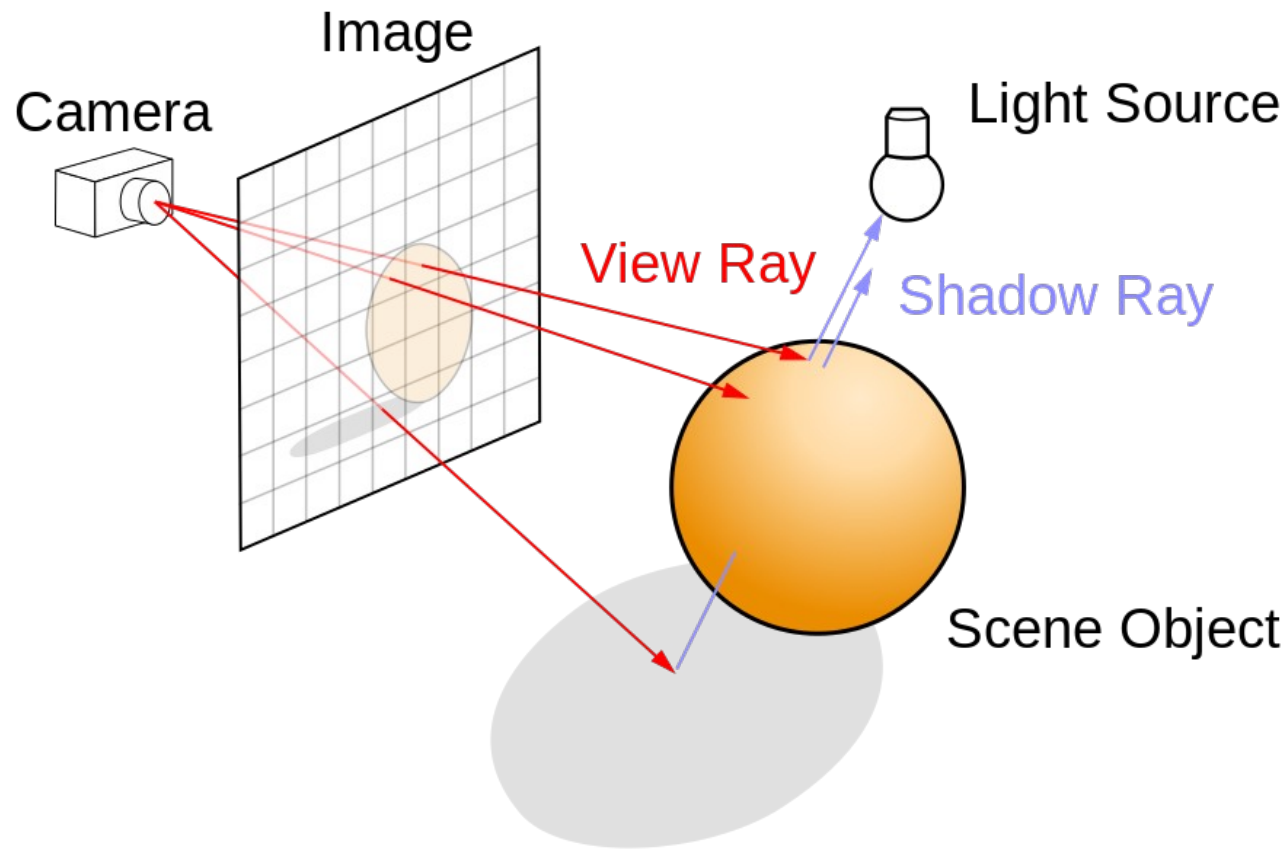


(source : reddit - samsung Gear VR demo screenshot)

Annexe : Lancer de rayons

Lancer de rayons (raytracing)

- Essentiellement utilisé pour le rendu offline



Rastérisation vs Lancer de rayons

Lancer de rayons

- Proche du modèle physique (transport des photons)
- Algorithme générique, c'est le modèle de simulation physique qui change
- Concept simple à appréhender

Rastérisation

- Nécessite moins de puissance de calcul
- Plus facilement parallélisable
- Architecture des GPUs adaptée