**A-FRAME**

<https://aframe.io/>

<https://aframe.io/docs/1.0.0/introduction/>

**PAS A PAS**

<https://aframe.io/docs/0.9.0/guides/building-a-basic-scene.html>

**TUTOS**

School : <https://aframe.io/aframe-school/#/>

Vidéos : <https://www.youtube.com/watch?v=dv6_C4UqTfs&list=PLRtjMdoYXLf4inSULAHyCMqpIUj4cmBTr&index=1>

Sources : <https://github.com/SonarSystems/A-Frame-WebVR-Tutorials>

**OUTILS**

Inspecteur : <https://github.com/aframevr/aframe-inspector>

**TIPS**

Remarque : <https://joshondesign.com/2017/07/24/aframe_tips>

Best practices : <https://github.com/aframevr/aframe/blob/master/docs/introduction/best-practices.md>

**COMPONENTS**

Doc : <https://aframe.io/docs/0.9.0/core/component.html#events>

Exemples : <https://github.com/aframevr/aframe-inspector>

**MODELES**

<https://aframe.io/docs/1.0.0/introduction/models.html>

**LOWPOLY**

<https://www.turbosquid.com/3d-models/3d-style-couple-casual-man-model-1387761>

<https://www.turbosquid.com/3d-models/lowpolygon-human-base-mesh-max-free/1049654>

**SCAN2LOWPOLY**

<https://sketchfab.com/blogs/community/tutorial-low-poly-assets-from-3d-scans/>

<https://www.blendernation.com/2018/10/10/tutorial-low-poly-assets-from-3d-scans/>

high to low poly : <https://www.youtube.com/watch?v=Mm4X6_XjSBs>

lowpoly animal : <https://www.youtube.com/watch?v=JjW6r10Mlqs>

blender bake to lowpoly :

<https://www.youtube.com/watch?v=o568T79egTs>

<https://www.youtube.com/watch?v=G_dtR9LgnxE>

**ZORBA**

Film : <https://www.youtube.com/watch?v=ubkGD0Av4yk>

Flash mob :

* Ottawa : <https://www.youtube.com/watch?v=UhDgpXWkFHE>
* Manchester : <https://www.youtube.com/watch?v=iJM5qjbjOdc>
* Austin : <https://www.youtube.com/watch?v=UzEGqQXWPBI>
* Huddersfield : <https://www.youtube.com/watch?v=T6296CBFK9I>

Danse tuto :

* <https://www.youtube.com/watch?v=FPrUZx9AKSc>
* <https://www.youtube.com/watch?v=QXR5JqtKVwg>

**INTRODUCTION :** [**https://aframe.io/docs/1.0.0/introduction/**](https://aframe.io/docs/1.0.0/introduction/)

**0\_ définition** : Frame is a web framework for building virtual reality (VR) experiences. A-Frame is based on top of HTML, making it simple to get started. But A-Frame is not just a 3D scene graph or a markup language; the core is a powerful entity-component framework that provides a declarative, extensible, and composable structure to [three.js](https://threejs.org).

**1\_ exemples :** créer arborescence 1\_demo et tester exemple en ligne

**2\_ fonctionnalités** :

* **Nothing to install :** no build steps.
* **Declarative HTML**: HTML is easy to read, understand, and copy-and-paste
* **Entity-Component Architecture**: A-Frame is a powerful three.js framework, providing a declarative, composable, reusable entity-component structure. HTML is just the tip of the iceberg; developers have unlimited access to JavaScript, DOM APIs, three.js, WebVR, and WebGL.
* **Cross-Platform VR** : Don’t have a headset or controllers? Still works on standard desktop and smartphones.
* **Performance**: optimized from the ground up for WebVR (utilize le DOM, running smoothly at 90fps)
* **Visual Inspector :** Open up any A-Frame scene, hit <ctrl> + <alt> + i, and fly around to peek under the hood!
* **Base** : components such as geometries, materials, lights, animations, models, raycasters, shadows, positional audio, text, and controls for most major headsets.
* **Extension** : community components including environment, state, particle systems, physics, multiuser, oceans, teleportation, super hands, and augmented reality.
* **Proven and Scalable**: A-Frame has been used by companies such as Google, Disney, Samsung, Toyota, Ford, Chevrolet, Amnesty International, CERN, NPR, Al Jazeera, The Washington Post, NASA. Companies such as Google, Microsoft, Oculus, and Samsung have made contributions to A-Frame.

**LANGAGE :** [**https://aframe.io/docs/1.0.0/introduction/html-and-primitives.html**](https://aframe.io/docs/1.0.0/introduction/html-and-primitives.html)

* **Syntaxe** : langage de balise # HTML : <a-entity>
* **Primitives** : entités préfabriquées : exp : <a-box> <a-sky>
* **Exemples** :
* <a-box color="red" width="3"></a-box>
* <a-entity geometry="primitive: box; width: 3" material="color: red"></a-entity>
* **Physique** : librairie de lois de la physique

**INTEGRATION**

* Modele 3D :

Réf : <https://aframe.io/docs/1.0.0/introduction/models.html>

Loaders : <https://github.com/donmccurdy-up-for-adoption/aframe-extras/tree/master/src/loaders>

Exemple : <https://github.com/KhronosGroup/glTF-Sample-Models/tree/master/2.0>

Gltf : <https://aframe.io/docs/1.0.0/components/gltf-model.html>

|  |
| --- |
| In comparison to the older OBJ format, which supports only vertices, normals, texture coordinates, and basic materials, glTF provides a more powerful set of features. In addition to all of the above, glTF offers:   * Hierarchical objects * Scene information (light sources, cameras) * Skeletal structure and animation * More robust materials and shaders   For simple models with no animation, OBJ is nevertheless a common and reliable choice. |

* COMPONENT

Réf  : <https://aframe.io/docs/1.0.0/core/component.html>

<https://www.npmjs.com/search?q=aframe-component&page=1&ranking=optimal>

Exemple de composants : <https://aframe.io/aframe-registry/>

* ANIMATION

Réf : <https://github.com/donmccurdy-up-for-adoption/aframe-extras/tree/master/src/loaders>

Exemple : animation control : <https://github.com/rexraptor08/animation-controls>

Animation mixer : <https://github.com/n5ro/aframe-extras/blob/master/src/loaders/README.md>

Démo : <https://rexraptor08.github.io/animation-controls/>

Tuto blender create actions :

* + <https://www.youtube.com/watch?v=Gb152Qncn2s>
  + <http://blender.freemovies.co.uk/cubeecraft/>

Tuto blender export :

* + <https://unboring.net/workflows/animation.html>
  + <https://github.com/arturitu/threejs-animation-workflow/>

Animation-mixer :

* + Parameters : <https://github.com/n5ro/aframe-extras/blob/master/src/loaders/README.md>

Tuto multiple actions

* + <https://www.youtube.com/watch?v=Uomt2ABN5XM>
  + Question : <https://github.com/n5ro/aframe-extras/issues/137>
* ACTIONS

Dans Blender, les animations sont regroupés en actions, elles sont visibles sur la fenêtre des canaux du Dope Sheet et du Graph Editor avec leur images clefs éditables. Ce sont elles qui contiennent les images clefs. On peut modifier ces actions dans le mode Action Editor du dope Sheet.

Le NLA Editor (Animation Non linéaire) va nous permettre de manipuler ces blocs d'actions qui pourront être mixés, accelerés, multipliés, regroupés.

<http://www.formation-blender.org/page/interface/nlaEditor/anlaEditor/nlaEditor.html>

Le NLA Editor va travailler comme le ferai un editeur de montage et un logiciel de retouche photo en même temps. C'est une Timeline quelque peu améliorée, à ceci prés qu'elle ne gère pas des images clefs mais des actions.

<https://www.youtube.com/watch?v=o_gmm9mi1FE>

TREE.JS

Modèles : <https://threejs.org/examples/>

* CAMERA

Réf : <https://aframe.io/docs/1.0.0/components/camera.html>

Défaut (intégrée par défaut) :

<a-entity camera="active: true"

look-controls

wasd-controls

position="0 1.6 0"

data-aframe-default-camera>

</a-entity>

* SON

Réf : <https://aframe.io/docs/1.0.0/primitives/a-sound.html>

Entité : <https://aframe.io/docs/1.0.0/components/sound.html>

TUTOT THREE.JS

2016 : <https://www.youtube.com/watch?v=axGQAMqsxdw> 13 videos

2015 : <https://www.youtube.com/watch?v=9fg93FzfamU&list=PLOGomoq5sDLutXOHLlESKG2j9CCnCwVqg> (23videos)

2017 : <https://www.youtube.com/watch?v=YKzyhcyAijo&list=PLRtjMdoYXLf6mvjCmrltvsD0j12ZQDMfE> (10 vidéos)

2018 : <https://www.youtube.com/watch?v=uzkedMF-l4Q&list=PLbu98QxRH81KkLTN00OXhD8Y-pRVgTCnM> (14 vidéos)

<https://github.com/donmccurdy-up-for-adoption/aframe-physics-system>

<https://www.youtube.com/watch?v=Az1NWhKs308> placement aléatoire objets dans l’espace

cours général \_ aframe

<https://aframe-course.glitch.me/0700-360vr.html>