

Concevoir un clavier virtuel avec A-Frame

Enseignement : Yannick Narbey

Module : BTS SIO /SLAM/bloc 2

Date : 2025/12

Table des matières

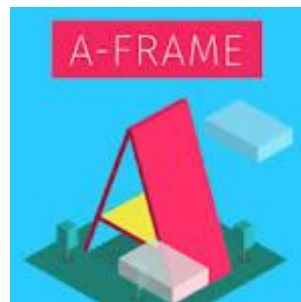
1. A-Frame.....	2
2. Conception d'un clavier avec aframe-xylayout.....	3

1. A-Frame

A-Frame est une bibliothèque web open source basée sur HTML et JavaScript qui permet de créer facilement des expériences de réalité virtuelle directement dans le navigateur.

Elle repose sur Three.js, mais masque sa complexité grâce à des balises simples comme `a-scene`, `a-entity` ou `a-camera`.

A-Frame est particulièrement adaptée à l'enseignement, car elle permet de construire des environnements 3D de manière progressive et visuelle.



Les scènes peuvent être visualisées sur ordinateur, smartphone ou casque VR sans installation complexe.

Le moteur fonctionne avec un système de composants, ce qui favorise la réutilisation du code et une bonne structuration des projets.

Elle permet également l'ajout de lumières, d'animations et d'interactions pour enrichir l'immersion. Grâce à son intégration avec les standards du web, A-Frame facilite le déploiement sur un simple serveur Apache.

2. Conception d'un clavier avec aframe-xylayout

Commencer par créer un projet A-Frame avec un plancher, une couleur de fond et le positionnement de la caméra :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>VR - Clavier Mot de Passe</title>
    <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
  </head>

  <body>
    <a-scene background="color: #222">
      <a-plane rotation="-90 0 0" width="20" height="20" color="#555"></a-plane>
      <a-entity>
        <a-camera position="0 1.6 0"></a-camera>
        <!-- Controleur avec raycaster -->
        <a-entity laser-controls="hand: right"></a-entity>
        <a-entity laser-controls="hand: left"></a-entity>
      </a-entity>
    </a-scene>
  </body>
</html>
```

Ajouter des raycaster aux contrôleurs donnant la capacité à l'utilisateur de cibler et sélectionner des objets :

```
<a-entity laser-controls="hand: right" raycaster="objects: .collidable"></a-entity>
<a-entity laser-controls="hand: left" raycaster="objects: .collidable"></a-entity>
```

La bibliothèque aframe-xylayout est un plugin A-Frame sert à créer des interfaces graphiques 2D ou 3D dans les environnements VR, comme :

- des panneaux,
- des boutons,
- des fenêtres
- ou des claviers.

Charger la bibliothèque xylayout dans la section <head></head>:

```
<head>
  <meta charset="utf-8" />
  <title>VR - Clavier Mot de Passe</title>

  <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
```

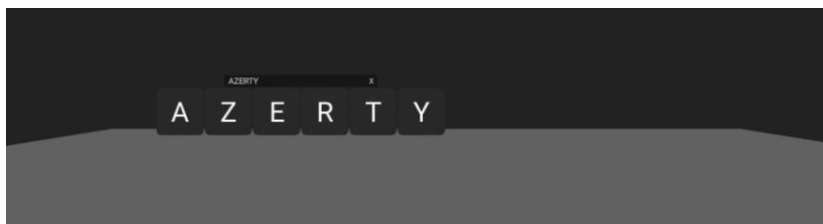
```
<script src="https://binzume.github.io/aframe-xylayout/dist/xylayout-all.min.js"></script>
</head>
```

Ajouter un clavier dans la section principale `<a-entity></a-entity>`, constitué de `<a-xycontainer>` contenant une section `<a-xybutton></a-xybutton>` pour chaque bouton.

```
<a-xywindow title="AZERTY" width="6" height="3" xycontainer="direction: column; spacing:0.05">
  <a-xycontainer direction="row">
    <a-xybutton label="A"></a-xybutton>
    <a-xybutton label="Z"></a-xybutton>
  </a-xycontainer>
</a-xywindow>
```

```
<a-entity>
...
  <!-- Clavier AZERTY -->
  <a-xywindow position="-0.8 1.3 -2.0" scale="0.15 0.15 0.15" width="6.5" height="3.2"
title="AZERTY" xycontainer="direction: column; spacing: 0.05">
    <a-xycontainer direction="row" spacing="0.05">
      <a-xybutton label="A" width="2" height="2" keyboard-key="value:A"></a-xybutton>
      <a-xybutton label="Z" width="2" height="2" keyboard-key="value:Z"></a-xybutton>
      <a-xybutton label="E" width="2" height="2" keyboard-key="value:E"></a-xybutton>
      <a-xybutton label="R" width="2" height="2" keyboard-key="value:R"></a-xybutton>
      <a-xybutton label="T" width="2" height="2" keyboard-key="value:T"></a-xybutton>
      <a-xybutton label="Y" width="2" height="2" keyboard-key="value:Y"></a-xybutton>
    </a-xycontainer>
  </a-xywindow>
</a-entity>
```

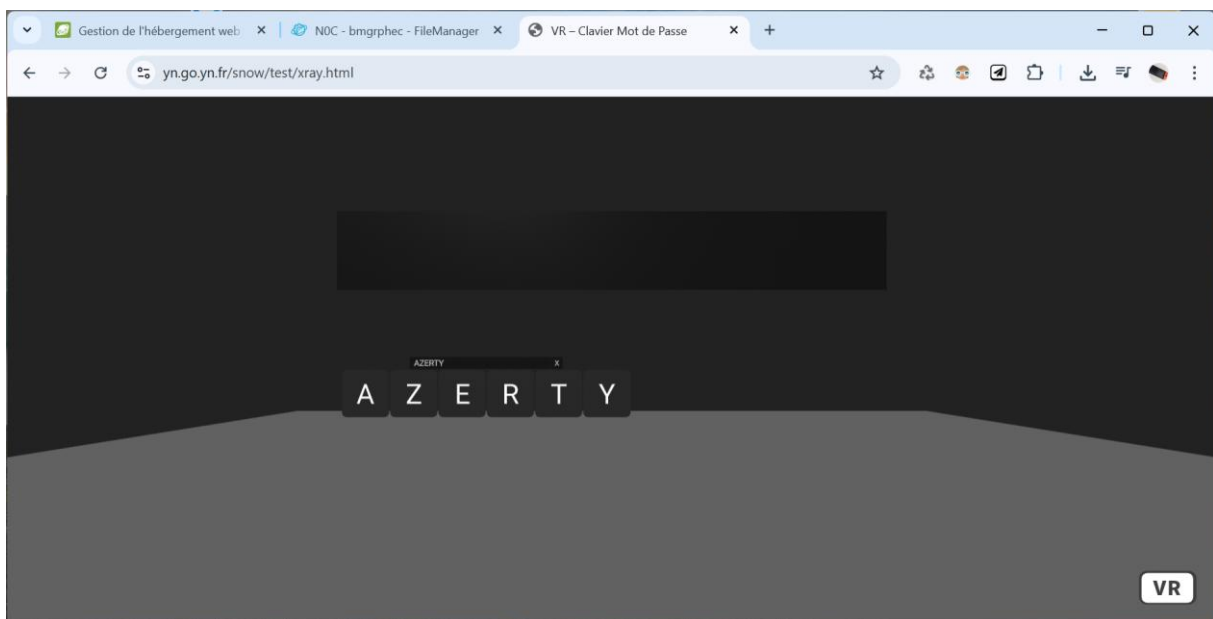
Tester :



Ajouter ensuite un champ de saisie « id=inputText » au dessus du clavier dans lequel les lettres sélectionnées devront apparaître :

```
<a-entity>
...
  <!-- Champ de saisie inputText -->
  <a-plane position="0 2.3 -2.0" width="3.5" height="0.5" color="#111">
    <a-text id="inputText" value="" position="-1.7 0 0.01" width="3.4"></a-text>
  </a-plane>
  <!-- Clavier AZERTY -->
  <a-xywindow position="-0.8 1.3 -2.0" scale="0.15 0.15 0.15" width="6.5"
height="3.2" title="AZERTY" xycontainer="direction: column; spacing: 0.05">
...
    </a-xycontainer>
  </a-xywindow>
</a-entity>
```

Le champ de saisie apparait mais rien ne s'y affiche lorsque l'on sélectionne des lettres :



Ajoutons un script javascript dans la section <head></head> dont l'objectif sera d'ajouter les sélections faites avec les contrôleurs dans le champ de saisie id=inputText:

```
<script src="https://binzume.github.io/aframe-xylayout/dist/xylayout-all.min.js"></script>

<script>
  AFRAME.registerComponent("keyboard-key", {
    schema: { value: { type: "string", default: "" } },
    init: function () {
      this.el.classList.add("collidable");
      this.el.addEventListener("click", () => {
```

```

const input = document.querySelector("#inputText");
let current = input.getAttribute("value");
if (this.data.value === "space") {
    input.setAttribute("value", current + " ");
    return;
}
if (this.data.value) {
    input.setAttribute("value", current + this.data.value);
}
});
},
});
</script>
</head>

```

Objectif : remplir
inputText

Les lettres sélectionnées avec les contrôleurs se reportent maintenant dans le champ de saisie, ajoutons un petit effet sonore à chaque ajout :

```

<script>
const audioCtx = new (window.AudioContext || window.webkitAudioContext)();

function playBeep() {
    const osc = audioCtx.createOscillator();
    const gain = audioCtx.createGain();
    osc.type = "square";
    osc.frequency.value = 800;
    gain.gain.value = 0.05;
    osc.connect(gain);
    gain.connect(audioCtx.destination);
    osc.start();
    osc.stop(audioCtx.currentTime + 0.05);
}

AFRAME.registerComponent("keyboard-key", {
    schema: { value: { type: "string", default: "" } },
    init: function () {
        this.el.classList.add("collidable");
        this.el.addEventListener("click", () => {
            playBeep();
            const input = document.querySelector("#inputText");
            ...

```

Ajoutez maintenant plusieurs lignes au clavier virtuel :

```

<!-- Clavier -->
<a-xywindow position="-0.8 1.3 -2.0" scale="0.15 0.15 0.15" width="6.5"
height="3.2" title="AZERTY" xycontainer="direction: column; spacing: 0.05">

```

```

<!-- Rangée 1 -->
<a-xycontainer direction="row" spacing="0.05">
  <a-xybutton label="A" width="2" height="2" keyboard-key="value:A"></a-xybutton>
  <a-xybutton label="Z" width="2" height="2" keyboard-key="value:Z"></a-xybutton>
  <a-xybutton label="E" width="2" height="2" keyboard-key="value:E"></a-xybutton>
  <a-xybutton label="R" width="2" height="2" keyboard-key="value:R"></a-xybutton>
  <a-xybutton label="T" width="2" height="2" keyboard-key="value:T"></a-xybutton>
  <a-xybutton label="Y" width="2" height="2" keyboard-key="value:Y"></a-xybutton>
  <a-xybutton label="U" width="2" height="2" keyboard-key="value:U"></a-xybutton>
  <a-xybutton label="I" width="2" height="2" keyboard-key="value:I"></a-xybutton>
  <a-xybutton label="O" width="2" height="2" keyboard-key="value:O"></a-xybutton>
  <a-xybutton label="P" width="2" height="2" keyboard-key="value:P"></a-xybutton>
</a-xycontainer>

<!-- Rangée 2 -->
<a-xycontainer direction="row" spacing="0.05">
  <a-xybutton label="Q" width="2" height="2" keyboard-key="value:Q"></a-xybutton>
  <a-xybutton label="S" width="2" height="2" keyboard-key="value:S"></a-xybutton>
  <a-xybutton label="D" width="2" height="2" keyboard-key="value:D"></a-xybutton>

```

