

TRABAJO FIN DE GRADO

Editor de escenas en realidad virtual

Alumno: Miguel Hidalgo Pérez

Tutor: Dr. Jesús M. González Barahona

Curso Académico: 2020-2021

ÍNDICE

- ❑ Objetivos
- ❑ Introducción
- ❑ Tecnologías usadas
- ❑ Proyecto
- ❑ Demo
- ❑ Conclusiones
- ❑ Preguntas

OBJETIVOS

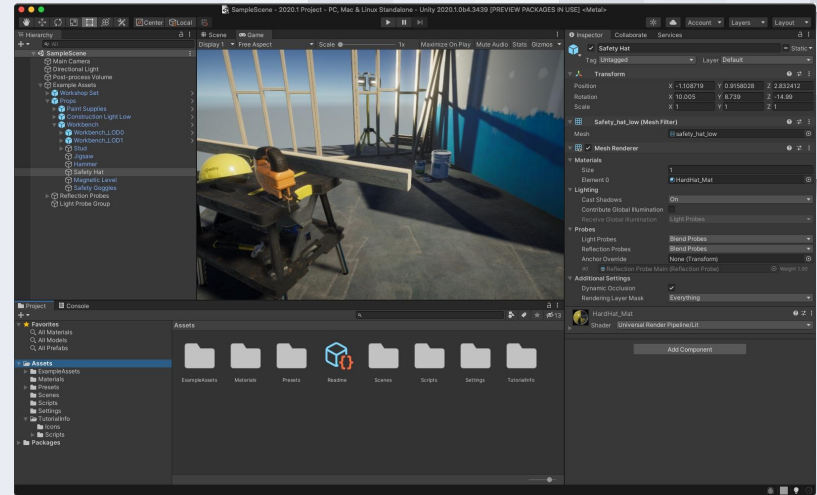
Crear un editor de escenas 3D en realidad virtual que ejecuta en un navegador web.

- Funcionar en múltiples dispositivos.
- La escena será editable dentro de ella misma.
- Creación de escenas mediante menús y eventos.
- Exportar escenas.

EDITORES ESCENAS 3D TRADICIONALES



Blender



Unity

REALIDAD VIRTUAL

□ Sector en crecimiento

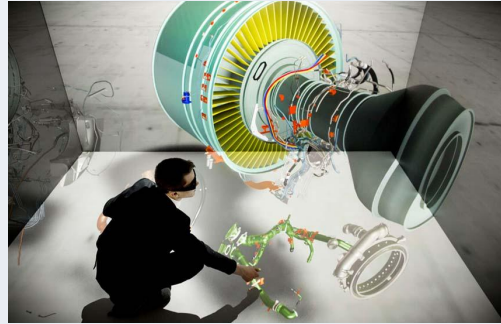


25 Virtual Reality Use Cases And their leading innovators



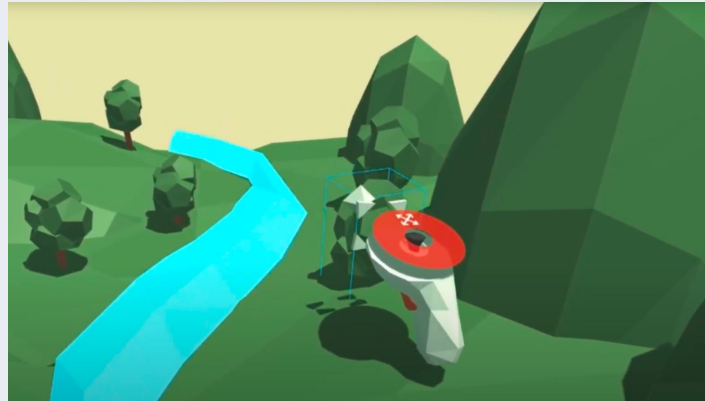
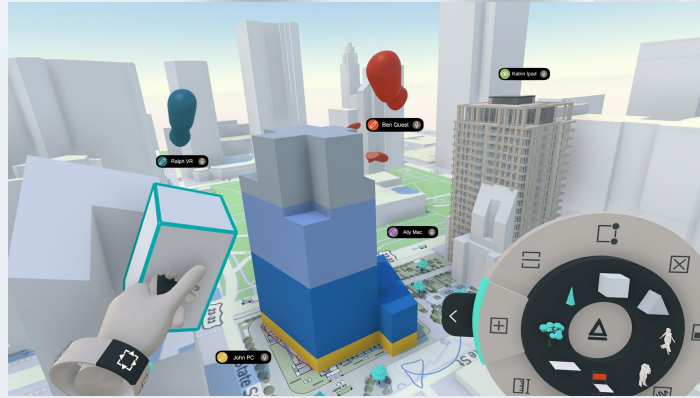
Venture Radar

Cinema JAUNT	Meditation GUIDED MEDITATION	Sports Training STRIVR	Recruitment Wade & Wendy	Pain Relief DEEPSTREAM VR	Mental Health Clevr	Travel Marriott relevent FRAMESTORE
Architecture iris	Education UNIVERSIV	“People Will Spend the Majority of Waking Time in Virtual Reality by 2020” - Zack Kanter			Surgery Training conquer mobile	Automotive Design WORLDVIZ
Sports Spectating liveike	Pilgrimage MECCA3D	Social Networking AltspaceVR	Workspaces Breakroom	Industrial Training eon reality	Marketing 10XARMY	Courtroom Universität Zürich
Journalism EMBLEMATIC GROUP	Flying FLYBI	Data Visualisation Lumacode MASTERS OF PIE	Cognitive Training CEREVRUM	Manufacturing VIRTALIS	Gaming CYBERITH	Shopping TRILLENIUM



REALIDAD VIRTUAL

- ¿Editores en Realidad virtual?



TECNOLOGÍAS USADAS



A-Frame

Framework web para construir experiencias de realidad virtual en el navegador.

three.js

Biblioteca 3D basada en WebGL. API para crear escenas, sombras, materiales, texturas, etc.



webpack

Transpilador y empaquetador de módulos.



JavaScript

Lenguaje utilizado para dotar de funcionalidad a páginas web, inicialmente pensado para animaciones e interacciones cliente.



Lenguaje de marcado para construir páginas web. Incluye nuevas funcionalidades respecto HTML4. (WebXR)



git

Herramienta de control de versiones para gestionar el proyecto.



Universidad
Rey Juan Carlos

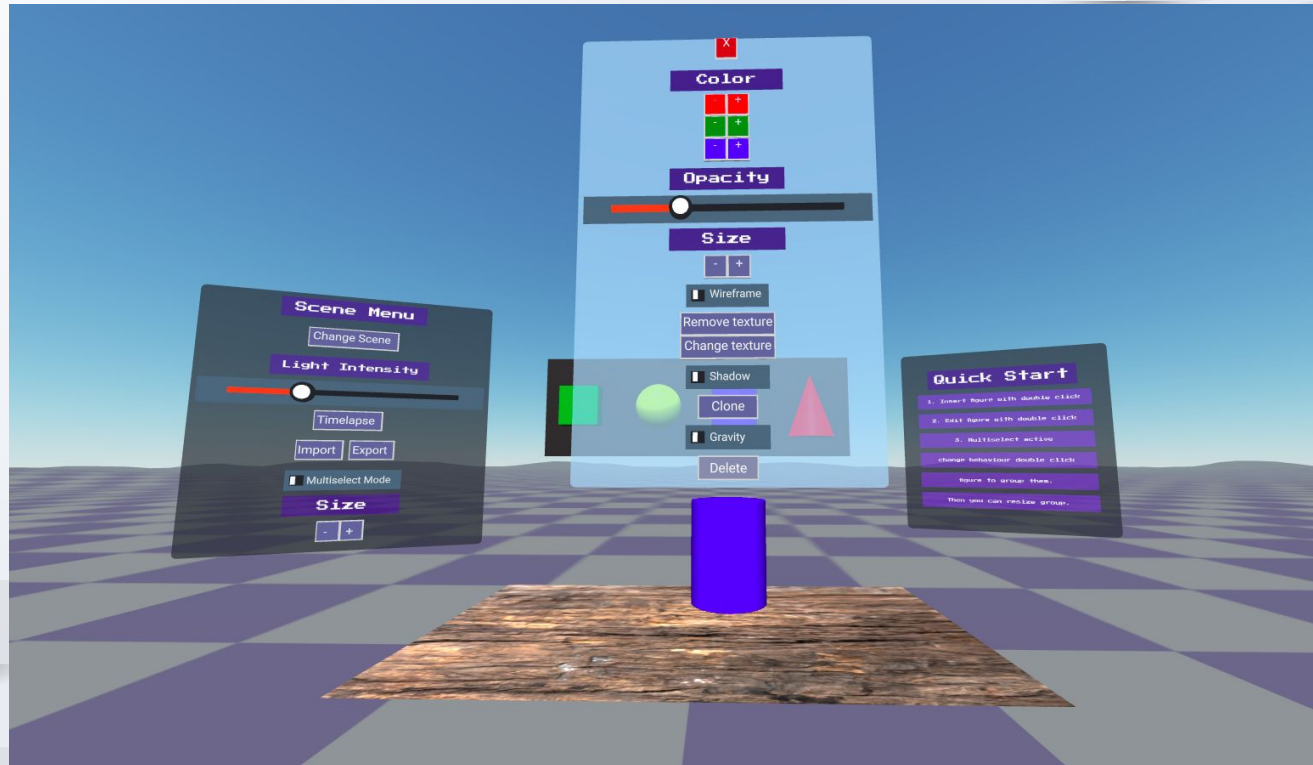
A-Frame Editor Scene

Funcionalidad

- ❖ Componentes basados en A-Frame.
- ❖ API para controlar figuras, menús y escena.
- ❖ Materiales, físicas, luces y sombras.
- ❖ Multiselección.
- ❖ Exportar e importar.

A-Frame Editor Scene

Resultado



A-Frame Editor Scene

Escena HTML

```
<a-scene physics="driver: ammo;">
  <!-- Camera. -->
  <a-entity id="rig" position="0 1.6 5.2">
    <!-- Mouse camera-->
    <a-entity id="camera-mouse" camera look-controls wasd-controls
      cursor="rayOrigin: mouse"
      raycaster="object: .selectable-superhands [gui-interactable]"
      super-hands="colliderEvent: raycaster-intersection;
                    colliderEventProperty: els;
                    colliderEndEvent: raycaster-intersection-cleared;
                    colliderEndEventProperty: clearedEls;"
    ></a-entity>
  </a-entity>
  <a-entity id="background-scene" environment="preset: default; groundColor: #445; grid: cross"></a-entity>
</a-scene>
```

A-Frame Editor Scene

Código Componentes Escena

```
const lightScene = new LightScene( props: {
  type: 'directional',
  castShadow: true,
  intensity: 0.9,
  shadowCameraVisible: false,
  position: "0 3 4"
});

const floor = new Plane( plane: {
  id: clonePodiumId,
  height: 3,
  width: 5,
  rotation: '-90 0 0',
  material: {
    src: textures.WOODEN,
    roughness: 1
  },
  physics: {
    body: 'static',
    shape: 'box'
  }
});
```

```
const initialFigures: Array<Figure> = [
  new Cone( cone: {
    'radius-bottom': 0.3,
    height: 0.8,
    color: 'red'
  }),
  new Cylinder( cyl: {
    radius: 0.3,
    height: 0.8,
    color: 'blue'
  }),
  new Sphere( sphere: {
    radius: 0.3,
    color: 'yellow'
  }),
  new Box( box: {
    height: 0.5,
    width: 0.5,
    depth: 0.5,
    color: 'green'
  })
];
```

A-Frame Editor Scene

Componentes Figuras

Modelo y Herencia

Comportamiento

```
class Figure {
  id?: string;
  htmlRef?: HTMLElement;
  primitive?: string;
  color?: string = 'white';
  material?: any;
  shadow?: boolean = false;
  opacity?: number = 1;
  wireFrame?: boolean = false;
  physics?: Physics = null;

  setColor?(color: string) {
    this.color = color;
    this.htmlRef.setAttribute(qualifiedName: 'color', color);
  }

  setMaterial?(material: any) {
    this.material = material;
    if (material) {
      const materialAttr: string = propsInline(material);
      this.color !== 'white' && this.setColor('white');
      this.htmlRef.setAttribute(qualifiedName: 'material', materialAttr);
    } else {
      this.htmlRef.removeAttribute(qualifiedName: 'material');
    }
  }

  setOpacity?(percent: number) {
    this.opacity = percent
    this.htmlRef.setAttribute(qualifiedName: 'opacity', percent.toString());
  }
}
```

```
class Cylinder extends Figure {
  radius: number;
  height: number;

  constructor(cyl: Cylinder) {
    const {height, radius} = cyl;
    cyl.primitive = 'a-cylinder';
    super(cyl);
    this.height = height;
    this.radius = radius;
  }
}
```

```
export function registerSelectableFigureScene() {
  const selectableFigureSceneComponent = {
    dependencies: ['raycaster'],
    init: function () {
      const figSelected = this.el;
      let cachedProps: any;

      // Double click
      this.el.addEventListener('click', function (evt) {
        showFigureMenu(figSelected);
      });

      const opacityReduction = 0.2;

      // Hover
      this.el.addEventListener('mouseover', function (evt) {
        cachedProps = cloneProperties(figSelected);
        const figOpacity = Number(cachedProps?.opacity);
        const opacityHover = figOpacity - opacityReduction;
        figSelected.setAttribute('opacity', opacityHover.toString());
      });
    }
  };
}
```

A-Frame Editor Scene

Componentes Interfaces

Componentes Menú

```
export class EditMenuFigure {  
  
  private entityRef: HTMLElement;  
  private readonly figure: Figure;  
  
  constructor(fig: Figure) {  
    this.figure = fig;  
  
    this.createMenuContainer();  
  
    // Add controls properties  
    addControlCloseMenu(this.entityRef, fig);  
    addControlEditColor(this.entityRef, fig);  
    addControlEditOpacity(this.entityRef, fig);  
    addControlEditSize(this.entityRef, fig);  
    addControlEditWireframe(this.entityRef, fig);  
    addControlEditMaterial(this.entityRef, fig);  
    addControlEditShadow(this.entityRef, fig);  
    addControlCloneFigure(this.entityRef, fig);  
    addControlPhysicsFigure(this.entityRef, fig);  
    addControlDeleteFigure(this.entityRef, fig);  
  }  
}
```

Lógica Componente

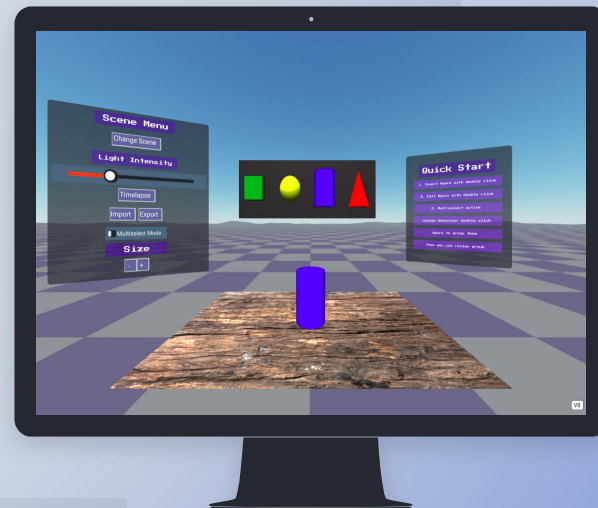
```
export function addControlEditOpacity(parentMenu: HTMLElement, figure: Figure) {  
  // Label  
  const label = createLabel({ text: 'Opacity', props: {  
    'margin': '-0.05 0 0.15 0'  
  }});  
  parentMenu.appendChild(label);  
  
  // Create Slider  
  const opacityControl = createSlider({ props: {  
    percent: '0.3'  
  }});  
  
  // Interaction  
  const customAction = 'slideOpacity' + new Date().getTime();  
  opacityControl.setAttribute('qualifiedName: 'onClick', customAction);  
  
  window[customAction] = function (event, percent) {  
    event.stopPropagation();  
    if (event instanceof CustomEvent) {  
      figure.setOpacity(percent);  
    }  
  }  
  
  parentMenu.appendChild(opacityControl);  
}
```

Componente Visual

```
export function createSlider(props?): HTMLElement {  
  const editControl = document.createElement('a-gui-slider');  
  
  // Style properties  
  const defaultProps = {  
    width: '2.5',  
    height: '0.25',  
    percent: '0.99',  
    margin: '0 0 0.05 0',  
    opacity: '0.8',  
    // 'slider-bar-height': '0.01',  
    'handle-outer-radius': '0.1',  
    'handle-inner-radius': '0.07',  
    'background-color': '#50687d'  
  };  
  
  setHtmlTags(editControl, defaultProps);  
  setHtmlTags(editControl, props);  
  
  return editControl;  
}
```

Múltiples dispositivos

Puede ser visualizado en cualquier navegador, ya sea escritorio, móvil o dispositivos de realidad virtual.



FASES DEL PROYECTO

Toma de requisitos

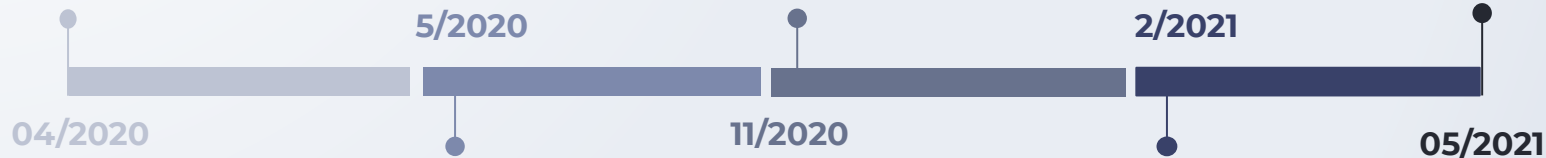
Estudio de la librería A-Frame y desarrollo de una demo con interacciones.

Edición de Figuras

Estudio de propiedades de A-Frame y librerías de interfaces para los menús.

Fin desarrollo del proyecto

Inicio de la memoria del proyecto y una web con demos.



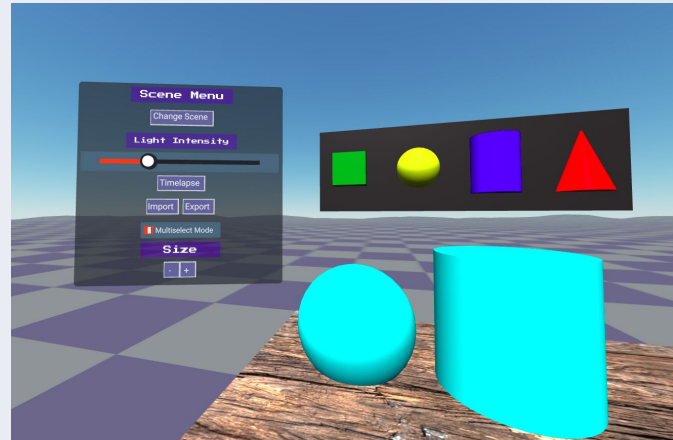
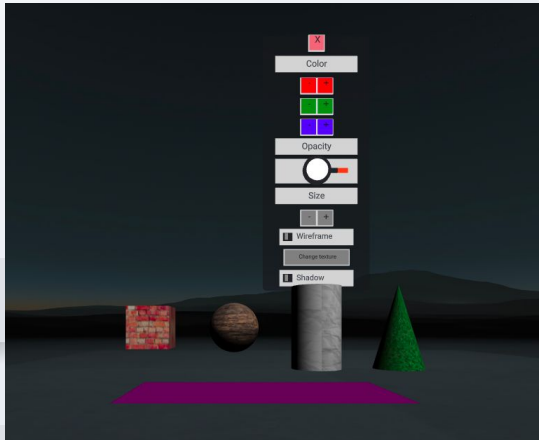
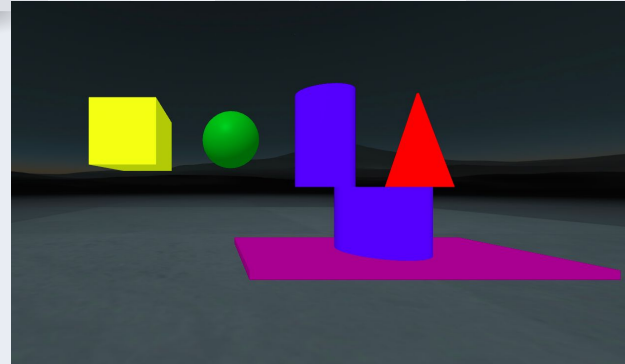
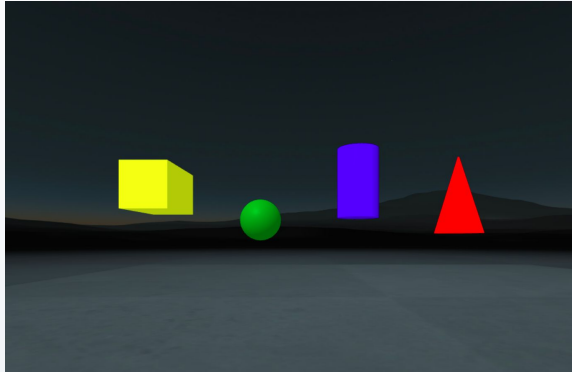
Desplazado y Clonado de figuras

Replicado de los elementos HTML en otra parte de la escena que pueden ser arrastrados.

Multiselección y extras

Multiselección de figuras y otras operaciones de escena como exportar o editar la luz.

FASES DEL PROYECTO



CONCLUSIONES

Se ha alcanzado el objetivo de crear una aplicación que permite construir escenas en realidad virtual para navegadores en múltiples dispositivos:

- ❑ Funcionalidad básica editores 3D.
- ❑ Componentes y Api para creación de figuras e interfaces usuario.
- ❑ Escena editable en tiempo real.
- ❑ Líneas futuras. Inspirarse en más funcionalidades de editores, como detectar colisiones, moldear figuras, inventario de modelos 3D...

DEMO



<https://hpmiguel.github.io/aframe-editor-scene/>

¡Gracias!

¿Preguntas?

Podéis encontrarme en:

- ❑ E-mail: hpmiguel@hotmail.com
- ❑ GitHub: <https://github.com/hpmiguel>