MARZIPANO

https://www.marzipano.net

Tiene una herramienta para crear el tour, pero no se puede grabar versiones y siempre hay que empezar desde cero:

https://www.marzipano.net/tool/

Instalación

Descargamos el código de Marzipano

```
git clone https://github.com/google/marzipano.git
```

Ejecutamos docker y mapeamos el directorio:

```
docker run --name marzipano -v
/dades/web/htdocs/tour360.lobo99.com/marzipano/:/marzipano -p 8099:8080 -ti
node bash
```

Instalamos Marzipano:

```
npm install
npm run dev
```

Ya lo tenemos funcionando en http://192.168.1.200:8099/demos/app-files/

Parámetros

```
      "initialViewParameters": {
        "yaw": 6.285,
        "pitch": 0,
        "fov": 0
      },
```

yaw: es el giro, de 0 a 6.285 una vuelta entera
pitch: el ángulo del suelo al techo, va de -1 a 1
fov: es el zoom, de 0 a 1.4

```
  var currentScene = null;
```

```
  function switchScene(scene) {
    stopAutorotate();
    if (currentScene !== null) {
      console.log("Cambiando de escena de '" + currentScene.data.id + "' a
'" + scene.data.id + "'.");
      if ( currentScene.data.id == "0-portalon_02" && scene.data.id == "1-
portalon_03" ) {
        scene.data.initialViewParameters.yaw=1.7;
      }
    } else {
      console.log("Cargando la escena inicial: '" + scene.data.id + "'.");
    }
    console.log(scene.data.initialViewParameters.yaw);
    scene.view.setParameters(scene.data.initialViewParameters);
    scene.scene.switchTo();
    startAutorotate();
    updateSceneName(scene);
    updateSceneList(scene);

    currentScene = scene;
  }
```

Modificaciones

Girar escena de un hotspor

Cuando clickamos en un hotspot la escena viene con la orientación por defecto, pero no es lo mismo si venimos por la izquierda que por la derecha o si vamos de adelante hacía atrás.

Añadimos el siguiente parámetro en el hotspot de una nuestra escena. Giro indica lo que giramos yaw en la escena target:

```
      "linkHotspots": [
        {
          "yaw": -2.3604018548682113,
          "pitch": 0.04299930353993808,
          "rotation": 0,
          "target": "1-portalon_03",
          "giro": 1.8
        },
        {
          "yaw": -1.0998945385019852,
          "pitch": -0.04682551712140359,
          "rotation": 0,
          "target": "2-habitaciones_01",
          "giro": 3.8
        }
```

```
      ]
```

En este caso es dentro de la escena Portalon_01 los dos hotspot que tiene. Significa que cuando estemos en la escena “portalon_01” y hagamos click en el hotspot que nos lleva a la escena portalon_03 va a girar la foto 1.8 y si vamos a habitaciones_01 va a girar la foto 3.8

Para que esto funcione, tenemos que modificar el fichero index.js dentro de la función createLinkHotspotElement el bloque “Add click event handler”

ORIGINAL:

```
    // Add click event handler.
    wrapper.addEventListener('click', function() {
      switchScene(findSceneById(hotspot.target));
    });
```

MODIFICADO

```
  // Add click event handler.
  wrapper.addEventListener('click', function() {
    // Aquí es donde modificas la propiedad 'yaw' de la escena destino.
    var scene = findSceneById(hotspot.target);
    if (scene && scene.data.initialViewParameters) {
      // Actualiza el 'yaw' con el valor 'giro' del hotspot.
      scene.data.initialViewParameters.yaw = hotspot.giro;
    }
    // Luego cambia a la escena con el 'yaw' actualizado.
    switchScene(scene);
  });
```

Añadir mapa con la posición y orientación

Copiamos el fichero mapa.png con capas transparentes en el directorio raiz donde está index.js

En el fichero index.js, entre

```
  // Initialize viewer.
```

y

```
// Create scenes,
```

Añadimos el minimapa

```
// Crear un contenedor para el mini-mapa
var miniMapContainer = document.createElement('div');
miniMapContainer.style.position = 'absolute';
miniMapContainer.style.right = '10px';
miniMapContainer.style.bottom = '10px';
```

```
miniMapContainer.style.width = '200px'; // Ajusta según necesites, igual que
el mini-mapa
miniMapContainer.style.height = 'auto';
miniMapContainer.style.zIndex = '100';

// Añadir el mini-mapa al contenedor
var miniMap = document.createElement('img');
miniMap.src = 'mapa2.png'; // Asegúrate de usar la ruta correcta
miniMap.style.width = '100%'; // Hace que el mini-mapa ocupe todo el
contenedor
miniMap.style.height = 'auto';
miniMap.style.pointerEvents = 'none'; // Esto permite que los eventos del
ratón pasen a través de la imagen

// Añadir el marcador al contenedor
var marker = document.createElement('div');
marker.style.position = 'absolute';
marker.style.width = '10px'; // Tamaño del marcador
marker.style.height = '10px';
marker.style.backgroundColor = 'red';
marker.style.borderRadius = '50%'; // Hace que el marcador sea un círculo
marker.style.zIndex = '101'; // Se muestra sobre el mini-mapa
// Posicionamiento inicial del marcador dentro del contenedor del mini-mapa
marker.style.transition = 'top 0.5s ease-out, left 0.5s ease-out';
marker.style.left = '50px'; // Ajusta según la ubicación inicial deseada
sobre el mini-mapa
marker.style.top = '50px';

// Añadir el mini-mapa y el marcador al contenedor
miniMapContainer.appendChild(miniMap);
miniMapContainer.appendChild(marker);

// Finalmente, añadir el contenedor del mini-mapa al body o al contenedor
principal
document.body.appendChild(miniMapContainer);
```

Creamos la función actualizarPosicionDelMarcador al final del documento antes de estas líneas:

```
// Display the initial scene.
switchScene(scenes[0]);
```

Función actualizarPosicionDelMarcador

```
function actualizarPosicionDelMarcador(nuevaX, nuevaY) {
    // Suponiendo que 'marker' es el marcador cuya posición quieres cambiar,
    // y 'miniMapContainer' es el contenedor del mapa y del marcador.
    // 'nuevaX' y 'nuevaY' son las nuevas posiciones del marcador en
píxeles,
    // relativas al contenedor del mini-mapa.

    marker.style.left = nuevaX + 'px';
```

```
    marker.style.top = nuevaY + 'px';
}
```

La función la llamamos dentro de la función switchScene:

```
  function switchScene(scene) {
    stopAutorotate();
    scene.view.setParameters(scene.data.initialViewParameters);
    scene.scene.switchTo();
    startAutorotate();
    updateSceneName(scene);
    updateSceneList(scene);
  }
```

La ponemos al final del todo después de updateSceneList

```
  function switchScene(scene) {
    stopAutorotate();
    scene.view.setParameters(scene.data.initialViewParameters);
    scene.scene.switchTo();
    startAutorotate();
    updateSceneName(scene);
    updateSceneList(scene);
actualizarPosicionDelMarcador(scene.data.initialViewParameters.x,scene.data.
initialViewParameters.y);
  }
```

Ahora en cada escena tenemos que poner los valores x,y de la posición que están en el mapa en la sección initialViewParameters. El punto 0,0 es la esquina superior izquierda

```
      "initialViewParameters": {
        "yaw": -1.9068445548687052,
        "pitch": 0.1964205885417094,
        "fov": 1.306532538376576,
        "x" : 125,
        "y" : 190
      },
```

Añadir nuevas fotos sin Marzipano tool

Hay que hacer 3 pasos:

```
1. Convertir foto en directorios y copiar en directorio tiles
2. Modificar data.js
3. Modificar index.html
```

Descargamos la foto en formato rectangular. Por ejemplo si la tenemos subida a kuula y la editamos, abajo a la derecha está la opción download

Para convertir las fotos en formato Marzipano usamos este repositorio nodejs y este repo:

```
https://github.com/jessehhydee/marzipano-pano-tiler.git
```

Por ejemplo tenemos nuestra foto en la ruta:

```
/home/ruth/tmp/tour360/planta2_01.jpg
```

Ejecutamos docker con node

```
docker run -ti -v /home/ruth/tmp/tour360/:/tour360 node bash
```

Bajamos el repositorio:

```
git clone https://github.com/jessehhydee/marzipano-pano-tiler.git
cd marzipano-pano-tiler
npm install
```

Copiamos la foto o fotos en el directorio input

```
mv /tour360/planta2_01.jpg input/
```

Convertimos la foto:

```
node index.js
```

Tarda un poco, unos 2 minutos, pone “..images fetched” y luego acaba:

```
..cubemap created

..planta2_01 - preview created
..planta2_01 - layer 0 created
..planta2_01 - layer 1 created
..planta2_01 - layer 2 created
..planta2_01 - layer 3 created

..images fetched
```

En el directorio output nos ha creado lo siguiente con un montón de fotos en cada directorio:

```
.
`-- planta2_01
    |-- 1
    |   |-- b
    |   |   `-- 0
    |   |-- d
    |   |   `-- 0
    |   |-- f
    |   |   `-- 0
    |   |-- l
```

```
|   |   `-- 0
|   |-- r
|   |   `-- 0
|   `-- u
|       `-- 0
|-- 2
|   |-- b
|   |   |-- 0
|   |   `-- 1
|   |-- d
|   |   |-- 0
|   |   `-- 1
|   |-- f
|   |   |-- 0
|   |   `-- 1
|   |-- l
|   |   |-- 0
|   |   `-- 1
|   |-- r
|   |   |-- 0
|   |   `-- 1
|   `-- u
|       |-- 0
|       `-- 1
|-- 3
|   |-- b
|   |   |-- 0
|   |   |-- 1
|   |   |-- 2
|   |   `-- 3
|   |-- d
|   |   |-- 0
|   |   |-- 1
|   |   |-- 2
|   |   `-- 3
|   |-- f
|   |   |-- 0
|   |   |-- 1
|   |   |-- 2
|   |   `-- 3
|   |-- l
|   |   |-- 0
|   |   |-- 1
|   |   |-- 2
|   |   `-- 3
|   |-- r
|   |   |-- 0
|   |   |-- 1
|   |   |-- 2
|   |   `-- 3
|   `-- u
```

```
    |       |-- 0
    |       |-- 1
    |       |-- 2
    |       `-- 3
    |-- 4
    |   |-- b
    |   |   |-- 0
    |   |   |-- 1
    |   |   |-- 2
    |   |   |-- 3
    |   |   |-- 4
    |   |   |-- 5
    |   |   |-- 6
    |   |   `-- 7
    |   |-- d
    |   |   |-- 0
    |   |   |-- 1
    |   |   |-- 2
    |   |   |-- 3
    |   |   |-- 4
    |   |   |-- 5
    |   |   |-- 6
    |   |   `-- 7
    |   |-- f
    |   |   |-- 0
    |   |   |-- 1
    |   |   |-- 2
    |   |   |-- 3
    |   |   |-- 4
    |   |   |-- 5
    |   |   |-- 6
    |   |   `-- 7
    |   |-- l
    |   |   |-- 0
    |   |   |-- 1
    |   |   |-- 2
    |   |   |-- 3
    |   |   |-- 4
    |   |   |-- 5
    |   |   |-- 6
    |   |   `-- 7
    |   |-- r
    |   |   |-- 0
    |   |   |-- 1
    |   |   |-- 2
    |   |   |-- 3
    |   |   |-- 4
    |   |   |-- 5
    |   |   |-- 6
    |   |   `-- 7
    |   `-- u
```

```
    |       |-- 0
    |       |-- 1
    |       |-- 2
    |       |-- 3
    |       |-- 4
    |       |-- 5
    |       |-- 6
    |       `-- 7
    `-- cube-map
```

Copiamos el directorio dentro de la instalación de marzipano en el directorio tiles

Por ejemplo si hemos añadido planta2_01 se suele poner el ID: 3-planta2_01 para tenerlos ordenados.

El directorio tiles nos quedaría mas o menos así

```
└── tiles
    ├── 0-portalon_01
    ├── 1-portalon_02
    ├── 2-patio_01
    └── 3-planta2_01
```

Modificamos el fichero data.js añadiendo otra entrada al vector json de Scenes. Podemos copiar otro tal cual y después ir cambiando los parámetros, solo tenemos que cambiar ID para que coincida con el que tenemos:

```
    {
      "id": "3-planta2_01",
      "name": "planta2_01",
      "levels": [
        {
          "tileSize": 256,
          "size": 256,
          "fallbackOnly": true
        },
        {
          "tileSize": 512,
          "size": 512
        },
        {
          "tileSize": 512,
          "size": 1024
        },
        {
          "tileSize": 512,
          "size": 2048
        }
      ],
      "faceSize": 1368,
      "initialViewParameters": {
        "pitch": 0,
```

```
        "yaw": 0,
        "fov": 1.5707963267948966
      },
      "linkHotspots": [],
      "infoHotspots": []
    }
```

Y por último tenemos que añadir el DIV en el fichero index.html:

```
<div id="sceneList">
  <ul class="scenes">

      <a href="javascript:void(0)" class="scene" data-id="0-portalon_01">
        <li class="text">portalon_01</li>
      </a>

      <a href="javascript:void(0)" class="scene" data-id="1-portalon_02">
        <li class="text">portalon_02</li>
      </a>
      <a href="javascript:void(0)" class="scene" data-id="2-patio_01">
        <li class="text">patio_01</li>
      </a>
      <a href="javascript:void(0)" class="scene" data-id="3-planta2_01">
        <li class="text">planta2_01</li>
      </a>
  </ul>
</div>
```