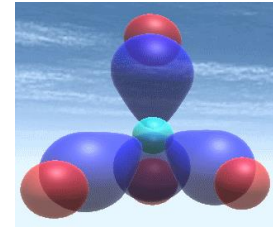


v-Slam documentation for v1.09



v-Slam is a Unity based browser for 3 dimensional pages, specially designed for VR/AR devices. However, the **PC** and **Mac versions** can also conveniently be used and can be downloaded from the main site. V-Slam supports the Web3D Consortium standard '**X3D**' xml (eXtensible 3 Dimensional) as mechanism for the web pages.

General use

The v-Slam browser aims to display informative 3d webpages that can be quickly accessed using features like the search engine, favorites etc. It allows promotion of organizations, products and ideas within the emerging market of virtual and augmented reality, without having to invest in complex programming or technology.

Like with 2d websites, the 3d pages therefore are often clear and simple, presenting the user what he/she is looking for.

Navigation

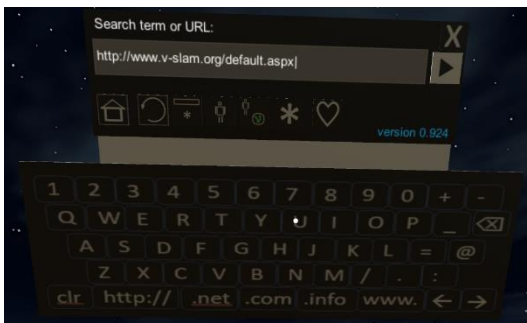
The browser opens with the home page, showing a scene with clickable links to the most recent or popular few sites, visualized as blocks using the sites favorite icons. A small menu always floats above and a little bit in front of the camera, allowing the user to go back to the home page, or open the extended menu at any location.



On PC and Mac, the mouse can be used for navigation. Holding the right mouse button allows the user to look around, the scroll wheel moves the user's camera forth and back in the look direction. When a scene has a specially marked floor, small shoes appear, indicating one can click the floor with the left mouse button to navigate to that location.

v-Slam also supports navigation with the keyboard, using the **WSAD** and **arrow** keys.

When an object has a **clickable link** to another 2- or 3d page, the object grows a bit when hovering over the object. When an URL for a 2d site is supplied, the user is warned that the URL will be opened in another application



In VR and AR devices, the camera will follow the head of the user. For moving longer distances, one can also 'click' on the especially marked floors.



The extended menu holds the **navigation/search** bar. Using the virtual keyboard, a search term or complete URL can be entered. On PC, the regular keyboard can also be used.

Furthermore, there are buttons for special features, like marking a site as **favorite**, **liking** it etc.

Avatars

The main menu allows the user to choose a **nickname**, and to **toggle the avatar function**. When the avatar function is on, others will see the user's avatar when visiting a website, and the user will see the avatars of others. Their movements on the website are also synchronized.

Enabling the **microphone** then allows the visitors of a website to talk with each other.

Single user web scenes

Some sites are meant to be for single users. So, independent of having the avatar function on, these sites will not reveal the presence and position of the avatar to others and vice-versa: only the own avatar will be on that site.

An obvious implementation for this is the login site. Although it is never possible for others to see the writing of the user-name and password, people won't like the idea that someone is at the same scene when specifying this information.

To have a web scene act as a single user page, the meta tag 'singleuser' should be added to the page header with the content set to 'true':

```
<X3D version="3.0" ...>
  <Head>
    <Meta content="true" name="singleuser"/>
  </Head>
```

Virtual presentation: Scene action synchronization

A web designer can mark a site with an 'allow recording' meta tag. On such a site one of the users can start the action recording feature. When this feature is activated all interactive elements in the scene will also be synchronized. This allows the first user to give a **real time virtual presentation**, for example in a virtual meeting, a shop etc.

v-Slam extension on X3D

For v-Slam, the X3D xml is extended using tags in a separate namespace. This allows for example the use of '**prefabs**', a collection of models that are already available in v-Slam, and therefore can be applied in any website with a single xml node.

Convention

Both elements or attributes can be specific to the v-Slam namespace. If an element comes from the v-Slam namespace, the attributes don't need the slm: prefix.

When an attribute is a v-Slam extension of an X3D tag, the slm: prefix should precede the attribute.

Create an X3D page

Make a copy

An easy way to start a new X3D page is to make use of an existing one. One could for example copy the xml-code from one of the examples at the end of this document in any plain text or xml editor (using for example Visual Studio helps because of the **color coding and validation** of the xml, but one could also just use Notepad).

Then remove the part of the xml that will be different in the new page, often everything within the Group or Scene node. The file can then be saved locally to disk during the editing process.

File extension .x3dx

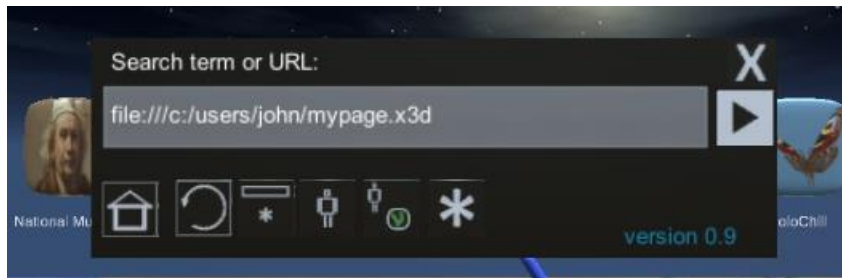
For v-Slam, the **extension** of the xml file is **not important**. For convenience, often .x3d(x) is chosen. For some webservers however, special configuration is necessary to allow .x3d(x) files to be served. In that case, perhaps .xml is a better choice.


The PC version associates the file extension .x3dx with the v-Slam browser during setup, allowing opening a scene by double-clicking it. Therefore .x3dx might be the preferred extension.

Future releases will also associate this extension on other platforms.

Set URL in v-Slam browser

Selecting the menu allows the user to enter a new URL. This URL could just be a file location, in which case it should be something like 'file:/// C:/users/john/mypage.x3dx'. Then press the 'Go to' button to load the empty scene in the v-Slam browser.



After loading the own x3d page, one can reopen the menu, and select the  button to make this temporarily the home page. This way it is easy to reload the page after each change, by pressing the large home button, to see the effect of the change.

Parts for the new page can best be added by copying similar parts from other websites, and then change the properties. Most often, this will be the complete xml starting and ending with a transform tag.

Publish your page

To allow others to be able to see your page, it must be accessible by a **complete URL**, just like with html pages. Therefore, the x3d page must be placed on a **web server** (you might want to use the .xml extension, see earlier).

Favicon

v-Slam uses a **favicon.png** image that should be in the same folder as the page being called for easy recognition of the page. The preferred size for the favicon.png is 128 x 128 or 256 x 256 pixels.

Keywords

Like 2d pages, a meta tag can be added in a header containing keywords. These keywords can help in the search algorithm.

```
<?xml version='1.0' encoding='utf-8'?>
```

```
<X3D profile='Immersive' version='Immersive' xmlns:xsd='http://www.w3.org/2001/XMLSchema-
instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.3.xsd'
xmlns:slm='http://www.v-slam.org'>
```


```
  <Head>
    <Meta name='keywords' content='X3D, v-slam, information, browser'></Meta>
  </Head>
  <Scene>
```

...

Register page

When the page, keywords and the favicon image have been prepared, and you people access de x3d page with an URL (like <http://www.mydomain.org/mypage.x3d>), you can use the v-Slam browser to register your page.

First use the main menu to enter your URL, and browse to the page. Then reopen the main menu, and

use the  button to request registration. After a small check from the v-Slam administrator, the page will then be accessible from the v-Slam search menu.

Content and other pages

Image textures, Inlines, audio etc. can be referred to using relative URL's, just like one does when composing html webpages. This is also the case for other x3d pages, that are called from the one page that is registered.

Preparing Movies

V-Slam supports displaying movies on textures. Currently the movie should be converted to the ogg format. There are several conversion possibilities, but not all with equally good results. The use of the script as shown below, give nice results. It uses ffmpeg (<https://trac.ffmpeg.org/wiki/TheoraVorbisEncodingGuide>):

```
ffmpeg -i myMovie.mp4 -c:v libtheora -c:a libvorbis -q:v 6 -q:a 5 myMovie.ogg
```

Uri Content Addition Mechanism

v-Slam supports **UCAM**, a powerful feature to have users load additional content in the scene. For users that are familiar with HTML, it is a bit like the use of frames, except that any item in the scene can be used to replace its content.

To implement this, one just should add the name of the item to the URL that targets the new content, preceded by a '#'. Furthermore, the target of the URL should be specified as “_self”.

```
<Cube name='item1' slm:href='vase.x3d#presentationPosition'
      slm:target='_self' slm:toolTip='Show model' />
```

Enabling the presentation feature

Any page can be marked with a **meta tag** to allow the **presentation feature**.

```
<X3D xmlns:slm='http://www.v-slam.org'>
  <Head>
    <Meta name='allowrecording' content='true' />
  </Head>
```

<Scene>

One of the visitors of the page – normally the initiator of the presentation – can then use the main menu to start the synchronization feature. All others who visit this page at the same time will see all the actions of the presenter in happening in their own environment. So, the presenter can choose to show the relevant slide, model etc. while he is holding his presentation.

For this to work correctly, all clickable shapes that can be activated in the scene, should have **unique names** (see the Cube having been given the unique name 'item1' in the code snippet).

```
<Transform position='0 0 0' scale='0.5 0.5 0.02'>
  <Shape>
    <Cube name='item1' slm:href='vase.x3d#presentationPosition'
          slm:target='_self' slm:toolTip='Show model' />
  </Shape>
</Transform>
```

Users that enter the presentation late will see their scene **fast-forward** to the present situation of the scene of the presenter.

Use of asset bundles

For **larger models** v-Slam supports the use of loading asset bundles. The code below shows an asset bundle with the file name 'rs.bundle', and an approximate size of 2672000 bytes. Using this code, the user will be prompted whether he wants to download it. If he confirms, asset bundle is downloaded to the user's device. When ready, the current page is reloaded. The models in the asset bundle can be presented in the scene using the slm:prefab tag. The item attribute should hold the name of the model, and the bundle attribute the name of the assetbundle as specified in the slm:assetbundle tag.

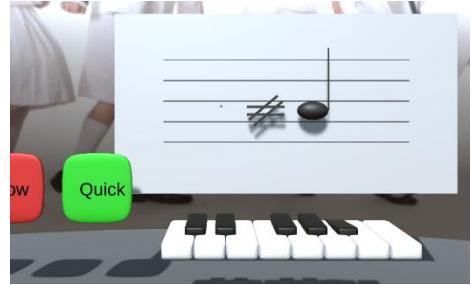
Multiple asset bundles can be specified. They remain available between web pages, so the models in the assetbundle can be reused, and the bundles don't have to be reloaded again until the browser has been closed.

Assetbundles which sizes are **below 10 Mb** do not need confirmation from the user.

```
<Scene>
  <slm:assetbundle name="abundle1" url="rs.bundle" sizeInBytes="2672000" />
  <Transform name='space' translation='0 -2.7 -12.5' >
    <Shape>
      <slm:prefab item='space' bundle='abundle1' />
    </Shape>
  </Transform>
</Scene>
```

Timed scene actions

A web scene can use scene actions that performs select events at specified times. This mechanism acts as if the user is selecting these objects. Using the content addition described earlier (UCAM), more automatic scene actions can be loaded. In this case, the programmer can decide to reset the scene time first to a certain value, using the special 'slam_reset_actions' keyword.



The action name must correspond to a name property given to the shape specifying tag (ie. Cube, Sphere, slm:prefab etc.). Of course the select only works if there is also an action assigned to the object, for example by specifying a link (slm:href) on it. Below a part used in the 'Piano' website:

```
<Scene>
  <slm:sceneaction actionname='slam_reset_actions' scenetime='2' />
  <slm:sceneaction actionname='key_c' scenetime='3' />
  <slm:sceneaction actionname='key_d' scenetime='3.6' />
```

v-Slam specific tags and attributes

The X3D xml is extended with xml tags and attributes that allows displaying special objects, features etc. Here a list of these extensions is displayed, accompanied with a small xml example.

slm:TargetPlatform	Allows parts of the xml to be applied only or not on the platform specified	
Type	Element	
Parent tags	Shape	
Attributes		
platform	Windows_uwp, hololens	The platform indicator concerned. Multiple platform indicators can be specified
mode	Include, exclude	Whether to include or exclude the inner xml tags when running on the specified platform.

```
<Transform slm:eulerrotation='0 32.255 0' scale='2.34 0.08 4'
  position='-2.1 0 -3.59'>
  <Shape>
    <slm:TargetPlatform platform='windows_uwp|hololens' mode='include'>
      <Cylinder slm:href='ms-windows-store://pdp/?ProductId=9pd82297jgd4'
        slm:target='_2D' slm:toolTip='Visit the store' />
    </slm:TargetPlatform>
    <slm:TargetPlatform platform='windows_uwp|hololens' mode='exclude'>
      <Cylinder
        slm:href='https://www.microsoft.com/en-us/store/p/holochill/9pd82297jgd4'
        slm:target='_2D' />
    </slm:TargetPlatform>
  <Appearance>
    <Material DEF='MaterialWhite' diffuseColor='1 1 1' />
```

```

    <ImageTexture url='"store.png"' />
  </Appearance>
</Shape>
</Transform>

```

slm:Prefab	Instantiates one of the prefab models from the v-Slam library	
Type	Element	
Parent tags	Shape	
Attributes		
item	The name of the prefab	
group	The group (folder) the prefab belongs to	
Current group, name combination examples	Plants	Montestera
	Avatars	Avatar
	Devices	HoloLens
	Primitives	Arrow1
		SmoothCube
		SmoothCube2
	Furniture	OfficeChair
		OfficeTable

```

<Transform name='plant' position='2.78 -0.59 9.24' scale='1 1 1'
  slm:eulerrotation='0 40 0'>
  <Shape>
    <slm:Prefab item='montestera' group='plants' />
  </Shape>
</Transform>

```

slm:eulerrotation	Define rotation of an object using Euler x-y-z angels, in degrees (the normal 'rotation' attribute in X3D expects the 4 values of a quaternion)
Type	Attribute
Parent tags	Transform

```

<Transform slm:eulerrotation='0 32.255 0' scale='2.34 0.08 4'

```

slm:href	Defines a clickable URL on an object. This URL can point to both 2d as 3d content. The slm:target property species how the content is rendered	
Type	Attribute	
Parent tags	Cube, Cylinder, Plane, Sphere, Capsule	
Special values	Some special URL's can be used for specific purposes	
	#slam_menu	Opens the navigation menu

slm:target	Is used in combination with slm:href. The slm:target property species how the content is rendered	
Type	Attribute	
Parent tags	Cube, Cylinder, Plane, Sphere, Capsule	
Allowed values	_blank	The target is rendered as a new 3d webpage
	_2D	A confirmation popup is displayed, after which the 3D experience is left, and the target content is displayed in the default 2d browser

	<code>_self</code>	The 3d target content is added to the current scene
Special options	When using “ <code>_self</code> ” as target, an anchor can be added to the url with the name of a target object. The target object then will be the parent of the extra 3d content	

slm:tooltip	Displays a message when an object is pointed at by the user. (Can also be used without specifying a <code>slm:href</code>)	
Type	Attribute	
Parent tags	Cube, Cylinder, Plane, Sphere, Capsule	

```
<Transform name='presentationPosition' position='1 0.32 0' scale='1 1 1'>
</Transform>
  <Transform position='0 0 0' scale='0.5 0.5 0.02'>
    <Shape>
      <Cube slm:href='vase2.x3d#presentationPosition'
        slm:target='_self' slm:toolTip='Show model' />
    </Shape>
  </Transform>
</Transform>
```

slm:faceCamera	Keeps the object faced to or from the camera	
Type	Attribute	
Parent tags	Cube, Cylinder, Plane, Sphere, Capsule	
Values	Face	Keeps object facing the camera
	Back	Keeps objects back facing the camera
	Lock-y	Keeps the object locked in the y-direction.
Special	The ‘lock-y’ can be used in combination with ‘face’ or ‘back’	

```
<Transform name='avatar' position='1.578 -0.542 0.968' scale='1 1 1'
rotation='0 0.7 0 -0.9'>
  <Shape>
    <slm:Avatar slm:faceCamera='face/lock-y' />
  </Shape>
</Transform>
```

Prefabs

Prefabs are low or high poly models that can be accessed and used in a 3D scene by specifying the Prefab xml tag (see above). The list shows the model that are currently available in v-Slam. The ‘primitive’ is the most relevant, because these primitives allows the display of complex models, just by composing them from the simple models from this group. Using the Transform tag

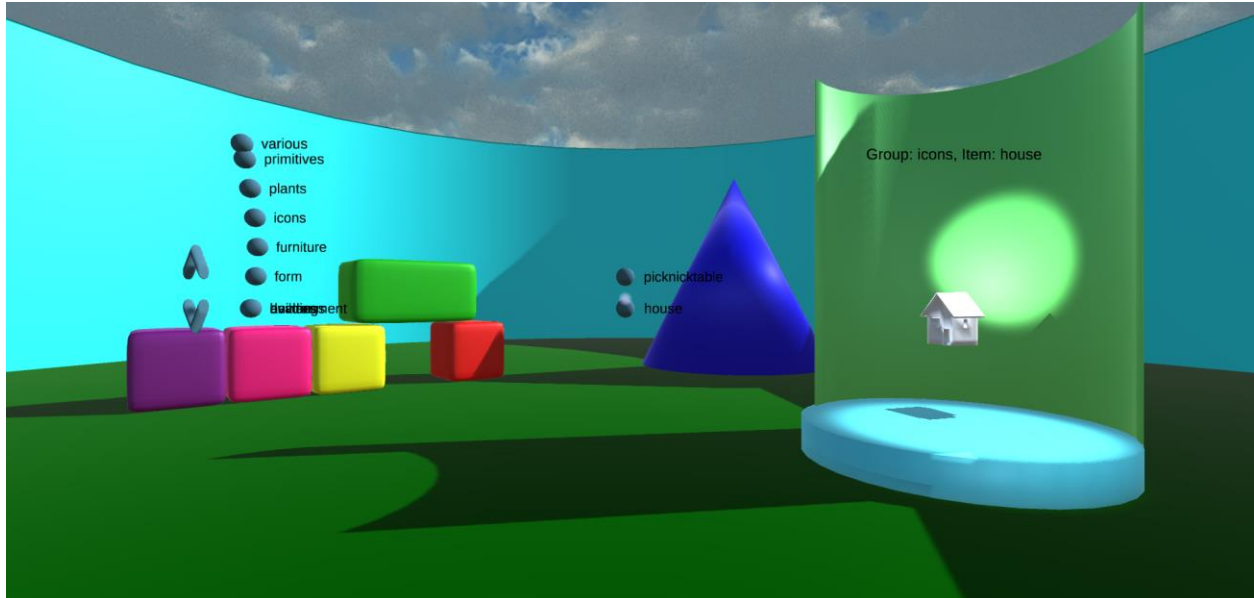








Figure 1: A scene made of primitives

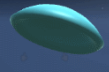
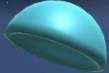
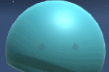

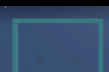
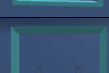




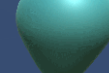



from the x3d xml the location, scale and rotation of these primitives can be specified, as well as the material.

With newer versions of v-Slam, this list of prefabs will be extended. Suggestions for new prefabs are welcome.

A more complete list of prefabs can be found on the v-Slam prefabs 3D site.



Model	Group	Item
	Primitives	Arrow1
	Primitives	cone
	Primitives	thorushalf
	Primitives	thorushalf1
	Primitives	thorus1
	Primitives	thorus

	Primitives	sphereonequarter
	Primitives	spherehalf
	Primitives	spherethreequarter
	Primitives	smoothcube
	Primitives	frame
	Primitives	frame2
	Primitives	frame3
	Primitives	triangle
	Primitives	arc
	Primitives	balloon
	Primitives	bulb
	Primitives	curvedcylinder
	Primitives	cylinderhalf
	Primitives	cylinderquarter
	Primitives	

Contents

v-Slam documentation for v1.02	1
General use	1
Navigation	1
Avatars	2
Virtual presentation: Scene action synchronization	2
v-Slam extension on X3D	2
Convention	2
Create an X3D page.....	2
Make a copy	2
File extension .x3dx.....	3
Set URL in v-Slam browser	3
Publish your page.....	3
Favicon	3
Keywords.....	3
Register page.....	4
Content and other pages	4
Preparing Movies	4
Uri Content Addition Mechanism	4
Enabling the presentation feature	4
Use of asset bundles	5
v-Slam specific tags and attributes	6
Prefabs	8
Index.....	13
Supplement 1: Example code full pages:	14
Start page	14
GiftShop	17

Index

slm:eulerrotation, 4
slm:faceCamera, 5
slm:href, 4
slm:Prefab, 3

slm:target, 4
slm:TargetPlatform, 3
slm:tooltip, 5

Supplement 1: Example code full pages:

Start page

```
<?xml version='1.0' encoding='utf-8'?>
<X3D profile='Immersive' version='Immersive' xmlns:xsd='http://www.w3.org/2001/XMLSchema-
instance' xsd:noNamespaceSchemaLocation='http://www.web3d.org/specifications/x3d-3.3.xsd'
xmlns:slm='http://www.v-slam.org'>
  <Scene>
    <ViewPoint DEF='ViewUpClose' position='0 1 3'></ViewPoint>
    <Group>
      <Transform rotation='0 1 0 3' position='0 1.1 0' scale='0.3 0.3 0.3'>
        <Shape>
          <Sphere slm:href='#slam_menu' slm:toolTip='Select to open main menu'></Sphere>
          <Appearance>
            <Material DEF='MaterialLightBlue' diffuseColor='0.1 0.5 1' emissiveColor='0 0
0.2'></Material>
            <ImageTexture url='images/earth-topo.png'></ImageTexture>
            <slm:Movement rotate='0 1 0'></slm:Movement>
          </Appearance>
        </Shape>
      </Transform>
      <Transform rotation='0 1 0 3' position='-0.5 1.1 0' scale='0.1 0.1 0.1'>
        <Shape>
          <Sphere></Sphere>
          <Appearance>
            <Material DEF='MaterialMoon' diffuseColor='0.5 0.5 0.5'></Material>
            <ImageTexture url='images/moon-4k.png'></ImageTexture>
            <slm:Movement rotate='0.03 0.2 0' center='0.001 0 0'></slm:Movement>
          </Appearance>
        </Shape>
      </Transform>
      <Transform position='0 -0.5 0'>
        <Shape>
          <Plane slm:walkfloor='0'></Plane>
          <Appearance>
            <Material DEF='MaterialFloor' diffuseColor='0.5 0.5 0.5'
transparency='0.2'></Material>
            <ImageTexture url='images/wally.jpg'></ImageTexture>
          </Appearance>
        </Shape>
      </Transform>
      <Transform translation='0 0.9 0' scale='0.03 0.03 0.03'>
        <Shape>
          <Text string='"v-Slam" "browse the world!"' slm:facecamera='back/lock-
y'></Text>
          <Appearance>
            <Material USE='MaterialLightBlue'></Material>
            <ImageTexture></ImageTexture>
          </Appearance>
        </Shape>
      </Transform>
      <Transform position='1.67417318012075 1 0.295201902033782' scale='0.2 0.2 0.03'>
        <Shape>
```

```

        <slm:Prefab name='smoothcube2' group='primitives' slm:href='museum/museum.x3d'
slm:target='_Blank'></slm:Prefab>
        <Appearance>
            <Material DEF='MaterialNational Museum' diffuseColor='1 1 1'></Material>
            <ImageTexture url='images/rembrandt.jpg'></ImageTexture>
            <slm:Movement rotate='0.03 1.2 0' applyToParent='true'></slm:Movement>
        </Appearance>
    </Shape>
</Transform>
<Transform position='1.67417318012075 0.7 0.295201902033782' scale='0.03 0.03
0.03'>
    <Shape>
        <Text slm:href='museum/museum.x3d' slm:target='_Blank' string='"National
Museum"'></Text>
        <Appearance>
            <Material USE='MaterialNational Museum' diffuseColor='0.1 0.35
0.46'></Material>
            <ImageTexture></ImageTexture>
        </Appearance>
    </Shape>
</Transform>
<Transform position='1.47224318643355 1 -0.85' scale='0.2 0.2 0.03'>
    <Shape>
        <slm:Prefab name='smoothcube2' group='primitives' slm:href='epona/epona.x3d'
slm:target='_Blank'></slm:Prefab>
        <Appearance>
            <Material DEF='MaterialEpona' diffuseColor='1 1 1'></Material>
            <ImageTexture url='images/Epona-logo.png'></ImageTexture>
            <slm:Movement rotate='0.03 1.2 0' applyToParent='true'></slm:Movement>
        </Appearance>
    </Shape>
</Transform>
<Transform position='1.47224318643355 0.7 -0.85' scale='0.03 0.03 0.03'>
    <Shape>
        <Text slm:href='epona/epona.x3d' slm:target='_Blank' string='"Epona"'></Text>
        <Appearance>
            <Material USE='MaterialEpona' diffuseColor='0.1 0.35 0.46'></Material>
            <ImageTexture></ImageTexture>
        </Appearance>
    </Shape>
</Transform>
<Transform position='0.581434243653637 1 -1.59747745533604' scale='0.2 0.2 0.03'>
    <Shape>
        <slm:Prefab name='smoothcube2' group='primitives'
slm:href='holochatspace/holochatspace.x3d' slm:target='_Blank'></slm:Prefab>
        <Appearance>
            <Material DEF='MaterialHoloChatSpace' diffuseColor='1 1 1'></Material>
            <ImageTexture url='images/HoloChatSpace.png'></ImageTexture>
            <slm:Movement rotate='0.03 1.2 0' applyToParent='true'></slm:Movement>
        </Appearance>
    </Shape>
</Transform>
<Transform position='0.581434243653637 0.7 -1.59747745533604' scale='0.03 0.03
0.03'>
    <Shape>
        <Text slm:href='holochatspace/holochatspace.x3d' slm:target='_Blank'
string='"HoloChatSpace"'></Text>
        <Appearance>

```

```

    <Material USE='MaterialHoloChatSpace' diffuseColor='0.1 0.35
0.46'></Material>
    <ImageTexture></ImageTexture>
    </Appearance>
  </Shape>
</Transform>
<Transform position='-0.581434243653636 1 -1.59747745533604' scale='0.2 0.2 0.03'>
  <Shape>
    <slm:Prefab name='smoothcube2' group='primitives'
slm:href='holochill/holochill.x3d' slm:target='_Blank'></slm:Prefab>
    <Appearance>
      <Material DEF='MaterialHoloChill' diffuseColor='1 1 1'></Material>
      <ImageTexture url='images/holochill.png'></ImageTexture>
      <slm:Movement rotate='0.03 1.2 0' applyToParent='true'></slm:Movement>
    </Appearance>
  </Shape>
</Transform>
<Transform position='-0.581434243653636 0.7 -1.59747745533604' scale='0.03 0.03
0.03'>
  <Shape>
    <Text slm:href='holochill/holochill.x3d' slm:target='_Blank'
string="HoloChill"></Text>
    <Appearance>
      <Material USE='MaterialHoloChill' diffuseColor='0.1 0.35 0.46'></Material>
      <ImageTexture></ImageTexture>
    </Appearance>
  </Shape>
</Transform>
<Transform position='-1.47224318643355 1 -0.850000000000001' scale='0.2 0.2 0.03'>
  <Shape>
    <slm:Prefab name='smoothcube2' group='primitives' slm:href='vslam/vslam.x3d'
slm:target='_Blank'></slm:Prefab>
    <Appearance>
      <Material DEF='Materialv-Slam Browser' diffuseColor='1 1 1'></Material>
      <ImageTexture url='images/vslam.png'></ImageTexture>
      <slm:Movement rotate='0.03 1.2 0' applyToParent='true'></slm:Movement>
    </Appearance>
  </Shape>
</Transform>
<Transform position='-1.47224318643355 0.7 -0.850000000000001' scale='0.03 0.03
0.03'>
  <Shape>
    <Text slm:href='vslam/vslam.x3d' slm:target='_Blank' string="v-Slam
Browser"></Text>
    <Appearance>
      <Material USE='Materialv-Slam Browser' diffuseColor='0.1 0.35
0.46'></Material>
      <ImageTexture></ImageTexture>
    </Appearance>
  </Shape>
</Transform>
<Transform position='-1.67417318012075 1 0.295201902033781' scale='0.2 0.2 0.03'>
  <Shape>
    <slm:Prefab name='smoothcube2' group='primitives'
slm:href='giftshop/giftshop.x3d' slm:target='_Blank'></slm:Prefab>
    <Appearance>
      <Material DEF='MaterialGiftShop Example' diffuseColor='1 1 1'></Material>
      <ImageTexture url='images/giftshop.png'></ImageTexture>

```



```

        <slm:Movement rotate='0.03 1.2 0' applyToParent='true'></slm:Movement>
    </Appearance>
</Shape>
</Transform>
<Transform position='-1.67417318012075 0.7 0.295201902033781' scale='0.03 0.03
0.03'>
    <Shape>
        <Text slm:href='giftshop/giftshop.x3d' slm:target='_Blank' string='"GiftShop
Example"'></Text>
        <Appearance>
            <Material USE='MaterialGiftShop Example' diffuseColor='0.1 0.35
0.46'></Material>
            <ImageTexture></ImageTexture>
        </Appearance>
    </Shape>
</Transform>
<Transform slm:eulerrotation='0 180 0' position='-0.246 0.731 0' scale='0.5 0.5
0.5'>
    <Shape>
        <slm:Prefab name='arrow1' group='primitives' slm:href='default.aspx?p=1'
slm:toolTip='Next page'></slm:Prefab>
        <Appearance>
            <Material DEF='MaterialPager2' diffuseColor='0.12 0.24 0.8'></Material>
            <ImageTexture></ImageTexture>
        </Appearance>
    </Shape>
</Transform>
<DirectionalLight direction='66.478 -226.251 -215.012' intensity='1.0'
shadowIntensity='1.0'></DirectionalLight>
</Group>
<Sound>
    <AudioClip url='"sound/soft.wav"' loop='true' enabled='true'
volume='0.2'></AudioClip>
</Sound>
<Background Name='skyBox' skyBox='sky5X5'></Background>
</Scene>
</X3D>

```

GiftShop

```

<?xml version='1.0' encoding='UTF-8'?>
<X3D xmlns:slm='http://www.v-slam.org'>
    <Scene>
        <Transform name='presentationPosition' position='1 0.32 0' scale='1 1 1'>
            </Transform>
        <Transform name='presentationfloor' position='0.401 0.32 0.668' scale='1 0.05 1'>
            <Shape>
                <Cube />
                <Appearance>
                    <Material DEF='MaterialItem' diffuseColor='1 1 1'></Material>
                    <ImageTexture url='"oldwood.jpg"' />
                </Appearance>
            </Shape>

```

```

</Transform>
<Transform name='bulletList' position='-1.61 2.7 -6' scale='1 1 1'>
  <Transform name='bullet'>
    <Transform position='0 0 0' scale='0.5 0.5 0.02'>
      <Shape>
        <Cube slm:href='vase.x3d#presentationPosition' slm:target='_self'
slm:toolTip='Show model' />
        <Appearance>
          <Material DEF='MaterialItem' diffuseColor='1 1 1'></Material>
          <ImageTexture url='vase.png' />
        </Appearance>
      </Shape>
    </Transform>
    <Transform name='keepitsimpleTitle' translation='-0.37 0.19 0' scale='0.1 0.1
0.1'>
      <Shape>
        <Text DEF='TextMessage' string='"Granit vase"' slm:text-anchor='left'>
          <FontStyle justify='LEFT' />
        </Text>
        <Appearance>
          <Material DEF='keepitsimpleTitle' diffuseColor='1 1 1' />
        </Appearance>
      </Shape>
    </Transform>
  </Transform>
  <Transform name='bullet' position='0 -1 0'>
    <Transform position='0 0 0' scale='0.5 0.5 0.02'>
      <Shape>
        <Cube slm:href='vase2.x3d#presentationPosition' slm:target='_self'
slm:toolTip='Show model' />
        <Appearance>
          <Material DEF='MaterialItem2' diffuseColor='1 1 1'></Material>
          <ImageTexture url='vase2.png' />
        </Appearance>
      </Shape>
    </Transform>
    <Transform name='keepitsimpleTitle' translation='-0.37 0.19 0' scale='0.1 0.1
0.1'>
      <Shape>
        <Text DEF='TextMessage' string='"Vase clay, white"' slm:text-anchor='left'>
          <FontStyle justify='LEFT' />
        </Text>
        <Appearance>
          <Material DEF='keepitsimpleTitle' diffuseColor='1 1 1' />
        </Appearance>
      </Shape>
    </Transform>
  </Transform>
  <Transform name='bullet' position='0 -2 0'>
    <Transform position='0 0 0' scale='0.5 0.5 0.02'>
      <Shape>
        <Cube slm:href='vase3.x3d#presentationPosition' slm:target='_self'
slm:toolTip='Show model' />
        <Appearance>
          <Material DEF='MaterialItem3' diffuseColor='1 1 1'></Material>
          <ImageTexture url='vase3.png' />
        </Appearance>
      </Shape>
    </Transform>
  </Transform>

```

```

    </Shape>
  </Transform>
  <Transform name='smallvaseTitle' translation='-0.37 0.19 0' scale='0.1 0.1 0.1'>
    <Shape>
      <Text DEF='TextMessage' string='"Small vase, glass"' slm:text-anchor='left'>
        <FontStyle justify='"LEFT"'/>
      </Text>
      <Appearance>
        <Material DEF='keepitsimpleTitle' diffuseColor='1 1 1'/>
      </Appearance>
    </Shape>
  </Transform>
</Transform>
<Transform name='bullet' position='0 -3 0'>
  <Transform position='0 0 0' scale='0.5 0.5 0.02'>
    <Shape>
      <Cube slm:href='chair.x3d#presentationPosition' slm:target='_self' />
      <Appearance>
        <Material DEF='MaterialItem4' diffuseColor='1 1 1'></Material>
        <ImageTexture url='"chair.png"' />
      </Appearance>
    </Shape>
  </Transform>
</Transform>
<Transform name='smallvaseTitle' translation='-0.37 0.19 0' scale='0.1 0.1 0.1'>
  <Shape>
    <Text DEF='TextMessage' string='"Garden chair"' slm:text-anchor='left'>
      <FontStyle justify='"LEFT"'/>
    </Text>
    <Appearance>
      <Material DEF='keepitsimpleTitle' diffuseColor='1 1 1'/>
    </Appearance>
  </Shape>
</Transform>
</Transform>
<Transform DEF='New_Text' translation='1.4 1.4 0.2' slm:eulerrotation='0 -13 0'
scale='0.07 0.07 0.07' >

  <Transform name='Introduction' >
    <Shape>
      <Text DEF='TextMessage' string='"GiftShop" "Example"'>
        <FontStyle justify='"LEFT"'/>
      </Text>
      <Appearance>
        <Material DEF='AppTitle' diffuseColor='1 1 1'/>
      </Appearance>
    </Shape>
  </Transform>
  <Transform DEF='New_Text' translation='0 -5.2 0' scale='0.5 0.5 0.5' >
    <Shape>
      <Text DEF='TextMessage' slm:rowLength='40' string='"The example shows pictures
of the items to sell. This makes the scene quick to load in the browser. When an item is
selected, a 3D model is loaded in the scene, at the predefined presentation location"'>
        <FontStyle justify='"LEFT"'/>
      </Text>
      <Appearance>
        <Material USE='AppTitle' diffuseColor='1 1 1'/>
      </Appearance>
    </Shape>
  </Transform>

```

```
    </Shape>
  </Transform>
</Transform>
<directionalLight direction='67.94 136.028 29.943' intensity='1.0'
shadowIntensity='1.0'> </directionalLight>
  <Transform name='storetransform' position='-2.18 0 -3.56' slm:eulerrotation='91.39899
-41.10101 109.886' scale='0.25 0.25 0.25'>
    <inline nameSpaceName='store' url='store.x3d' ></inline>
  </Transform>
</Scene>
</X3D>
```