This tutorial shows how to create an interface with buttons. The interface will help you to control several interactive functions inside the virtual world, for example to toggle between cameras, to activate another button and how to switch textures on a 3d model. Although buttons are commonly used for web applications, they can also be used to test installations with external input devices. After testing the interface with buttons, you can override each button with an external input device of your choice.

I suggest that you use the message system which is the most reliable way to control multiple actions in the world. The message system, independent from variables and parameters, is the most stable way to code playback inside a web browser.

The following tutorials cover

1- Creating a button with a 2D Sprites and the ‘Mouse waiter” BB. Good for simple static buttons. We use this setup to toggle textures on a 3D object.
2- Creating a button with a 2D Frames and the “Push Button” BB. Better for dynamic buttons changing appearance according to the mouse controls for example rollovers.

Importing an image in Virtools

Virtools uses several image formats

- jpg
- bmp
  Both the normal and RLE modes of this format is supported (note that RLE cannot be saved).
- pcx
  This format can only be read, it cannot be saved.
- tga
  This format is now fully supported, including images in 32 bits with an alpha channel. Both RLE and standard modes are supported.
- png
- tif
  Not all varieties of this format can be read. If you wish to save in this format, you are limited to RLE and RAW modes.
The format .gif is **not supported** for licensing reasons. "Animated Gifs" can easily be simulated in Virtools by using multiple texture slots for a 2D Sprite.

To import an image in a composition, put the file in your Data resource and drag the image from the Data Resource to the 3D Layout or the Level Manager with one of the following keystrokes.

- Dragging an image will create a **Texture**
- Shift+drag creates a **3D Sprite** (with associated material and texture)
- Ctrl+drag creates a **2D Sprite**
- Ctrl+Shift+drag creates **2D Frame** (with associated material and texture)

**1- Creating a button with a 2D Sprite and the “Mouse waiter’ BB. This tutorial is recommended for creating simple static buttons.**

Let's create a **2D sprite**. Drag and hold the CTRL key.
In the **2D Sprite Setup**, 
Select “Relative to Viewport” in order to keep the button inside the frame of the camera.

Select “AlphaTest” and “Blend” in order to create a button with transparency. The next steps show you how to build the behavior from scratch.

Please note that you can also drag and drop the ready-made behavior from Virtual Spaces Resource > Behavior Graphs > Mouse Button
Let’s add a script to the button, 2D Sprite. This interface management, script can be created for the Level. Go to Level Manager > Level, select the word “Level”, right-click and select “create script” in the pull down menu. This creates an empty script slot. Click on the triangle in front of the word “Level Scripts”, double click on the word “Level Scripts” > this opens the script’s slot in schematics.

Go to Building Blocks > Controllers > Mouse > “Mouse Waiter”, drag and drop the “Mouse Waiter” BB.
Edit the “Mouse Waiter” BB, outputs are added to the BB, as you check the mouse functions.

Go to BB > Interface > Screen > select the “2D Picking” BB that detects 2D or 3D objects located under the mouse pointer.

Go to BB > Logics > Streaming > “Switch on Parameter”, drag and drop the BB. The parameters inputs are located on the top part pf the BB. The behaviors outputs are located on the right side of the BB.
Please note the Parameter Operations BB that can be found under BB > Logics > Calculator > Op. Drag and drop inside the script. Right click on the BB and choose Edit Settings in the pull down menu. The BB constrain the 2D picking to the surface of the rendering window, the 3D Layout window.

Edit the “Switch on Parameter” BB, right click on the BB and select Construct > Add Parameter Input. Create one parameter input per button or 2D Sprite. Select Construct > Add Behavior Output with one output per action.
Right click on the BB and select **Edit Parameters** in the pull down menu, edit pin names and numbers with buttons, 2D sprites names.

We use this setup to toggle textures on a 3D object. Press on the 2D sprites on the left in order to change textures on the surface of the billboard.

Go to **BB > Logics Streaming > Parameter selector**.

Let's add a **Parameter Selector BB** in order to control the **Texture Load BB**. Each mouse click loads a different texture stored in **Level Manager > Textures**.
Compare the above script for displaying textures stored inside the composition with the script below for loading external textures.

In the case you want to upload the texture from a texture outside your composition, you can add the **Create String** BB. The string will add a URL in front of the file's name.

This variation on Texture Load shows how to upload online textures from a server.
2- Creating a button with a 2D Frames and the “Push Button” BB. This method is better for creating dynamic buttons that change appearance according to mouse controls for example rollovers.

Go to Photoshop, start creating a transparent button, save as a .png file.

Drag and drop a texture and hold **CTRL and SHIFT keys**. In Level Manager, a 2D frame is created with Materials and Texture. We will use the materials in order to create several versions of the dynamic button.
In Level Manager, right Click on the 2DFrame’s name, select “2D Frame Setup” in the pull down menu. Step 1, click on the Materials button > Step 2, in Materials Setup, select the “AlphaTest” and “Blend” setting.

Please note that you can also drag and drop the ready-made behavior from Virtual Spaces Resource > Behavior Graphs > Mouse Push Button

Let’s add a script to the button, 2D Frame. This interface management, script can be created for the 2D Frame. Go to Level Manager > 2D Frame, button, select the word “button”, right-click and select “create script” in the pull down menu. This creates an empty script slot. Click on the triangle in front of the word “button”, double click on the word “button Script” > this opens the script’s slot in schematics.

Go to Building Blocks > Interface > controls > “Push Button”, drag and drop the “Push Button” BB. Right click on the BB > Edit Settings, more outputs are added to the right of the BB as you check mouse functions in the Pushbutton Settings window.
Right click on the BB, choose Edit Parameters, select materials from the same texture or from other textures. This will create a button that can change appearance according to the type of mouse click and rollover.

Let's test the button.
Step 1, let’s use a **Text Display BB** in order to display a text when clicking the button.

Step 2, Go to **BB > Interface > Text > Text Display** for each output of the PushButton BB. Right click on the Text Display BB, choose Edit Parameters and add a Text.

Step 3, go to **BB > Logics > Message > Send Message**, add several “**Send Message**” BB inside the button’s script.

Step 4, let’s use a **Sequencer BB** in order to switch outputs, when clicking several times on the same button. Go to **BB > Logics > Streaming > Sequencer**.
This section shows how to use the **Push Button BB** in order to activate and to disactivate a second button. The first push button sends message to the second button.
Let’s create behaviors to activate or disactivate for the second push button. Step 1, pressing the first button –top- activates the second button –bottom.
Step 2, pressing again the top button with disactivate the bottom button.