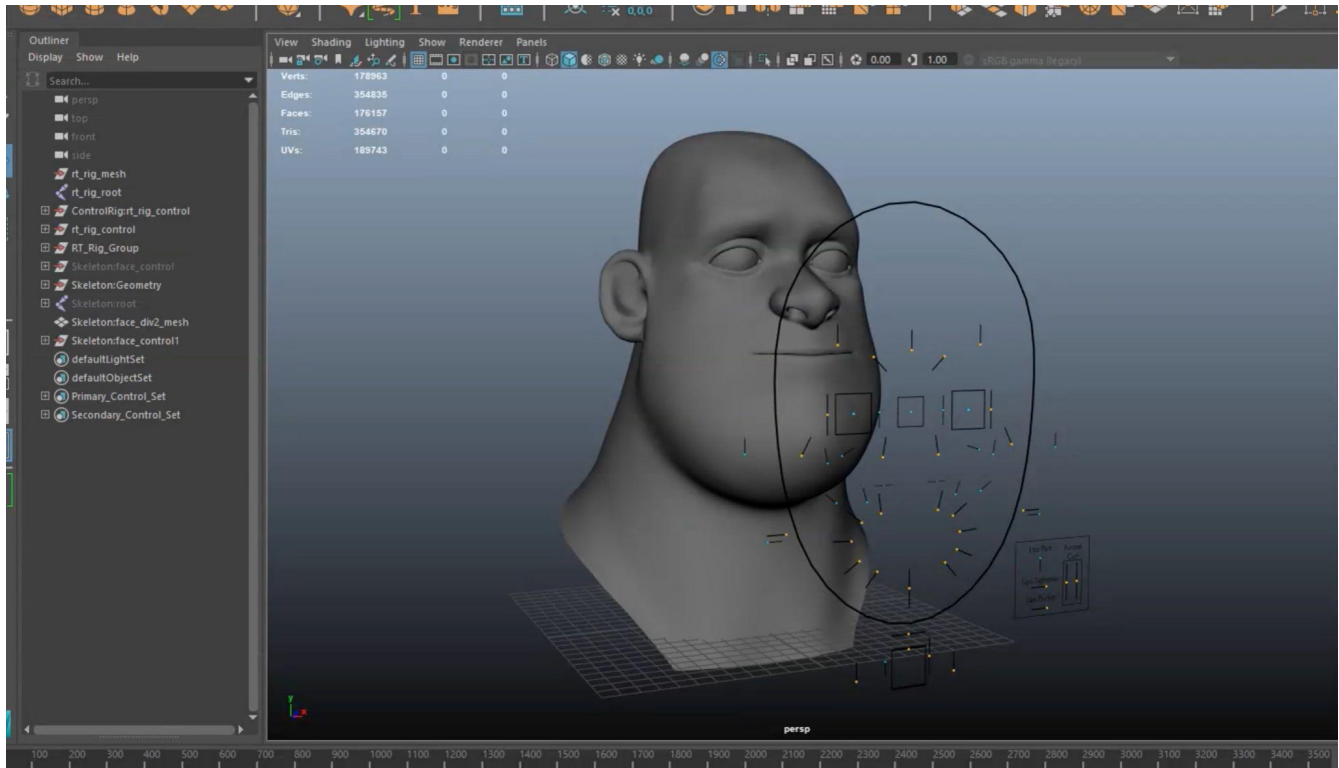




ZFT Face Instructions

Everything you need to get started with your ZFT Face asset



Authoring motion on a Ziva Face Trainer (ZFT) Face is simple:

1. Use performance capture (e.g. ARKit) to provide a baseline animation;
2. Polish the motion using the puppet controls in Maya (pictured above);
3. Export animation curves from Maya to an FBX file; [CLICK TO WATCH TUTORIAL](#)
4. Import the FBX of animation curves into your game (or wherever the ZivaRT asset for your ZFT Face has been deployed). [CLICK TO WATCH TUTORIAL](#)

Your ZFT Face also comes with a few animation clips, which you can apply directly to the puppet in Maya to see it in motion immediately (see the [AnimClips](#) directory).

How to Open the ZFT Face in Maya

ZivaRT Maya Player Setup

To open and interact with your ZFT Face, you will need to properly install the ZivaRT Player and activate the associated ZivaRT Player License. The installation file and license file are both available in your original ZFT Demo Face order on your Ziva Store account (store.zivadynamics.com)

Once you've downloaded your license file and installation file from the store please follow these instructions to set up your ZivaRT Maya Player:

1. Copy the `zrtMayaPlayer.mll` file to the plugins directory for your version of Maya. On Windows for Maya version 2019, this is located at:
“C:\Users\User\Documents\maya\2019\plug-ins”
For more Maya installation instructions see the [Maya Plugin Installation Instructions](#).
2. Now, open your version of Maya (in this example, 2019).
Navigate to Windows → Settings/Preferences → Plug-in Manager.
Navigate or search for “zrtMayaPlayer” and check the “loaded” checkbox.
Now the plug-in is installed.
3. To activate your license, place the ZivaRT Maya Player license file wherever you like, and add an environment variable ‘zivadyn_LICENSE’ that contains the path to that directory. The zivadyn_LICENSE environment variable can be added to your system variables, e.g.:
`zivadyn_LICENSE=<path/to/license/file/directory>`

ZFT Face Maya Setup

How to open your ZRT Face for the first time:

1. Before launching Maya, make sure to set `MAYA_MODULE_PATH` to point to the extracted puppet directory. It contains `plugins.mod` which will take care of sourcing plugins, setting script path, etc.
2. Launch Maya (we presently support 2019, 2020 & 2022)
3. Make sure to set the Maya project to the extracted puppet path (File → Set Project, then select the directory containing the `workspace.mel` file)
4. Now, open (or reference) the `puppet.ma` file (from the `scenes` directory) to begin using your ZFT Face in Maya.

How to Adjust the Teeth

What to do if you find undesirable penetration between the teeth and face tissue

Watch a video of this process here: [ZFT Face Prep Part 5 | Using a ZFT Face in Maya](#)

First, set all controls to zero. This process is safer when all the joints at the bind pose, because we'll run a script that creates part of the rig; so, it makes sense to make it at the bind pose and not when the rig is posed.

Next, use the script below to delete the jaw setup and select the relevant joints.

```
import maya.cmds as cmds

# delete jaw setup
cmds.delete("Jaw_Upper_Joint")
cmds.select("Skeleton:Jaw_Lower_Joint",
"Skeleton:Upper_Teeth_Joint")
```

Then, move the joints by hand into the desired position, so as to eliminate the undesired intersection. Make sure the Move Manipulator Tool is in world space (otherwise, each joint will be moved in local space in a funny way).

Finally, run this script to recreate the jaw setup:

```
import maya.cmds as cmds

# create jaw setup
newJoints = cmds.duplicate("Skeleton:Jaw_Upper_Joint")
cmds.parent("Jaw_Upper_Joint", "Head_ParentSpace")
cmds.delete("Tongue_01_Joint")
cmds.delete("Lower_Teeth_Joint")
cmds.ikHandle( n='ik_jaw', sj='Jaw_Upper_Joint', ee='Jaw_End_Joint'
)
cmds.pointConstraint("locator1", "locator2", "locator3", "locator4",
"ik_jaw", maintainOffset=True)
cmds.orientConstraint( 'Jaw_Upper_Joint',
'Skeleton:Jaw_Upper_Joint', maintainOffset=1 )
cmds.orientConstraint( 'Jaw_Lower_Joint',
'Skeleton:Jaw_Lower_Joint', maintainOffset=1 )
```

How to Use ARKit with a ZFT Face

Applying facial performance capture by your iPhone for base animation

To use ARKit to drive your Face Trainer puppet, follow these steps:

- Buy an iPhone
- Install the "Live Link Face" app ([App Store link](#))
- In the *Live Link Face* app, enable calibration in the settings
- Point the camera at your face, make a neutral expression (mouth closed, eyes relaxed), and hit the calibrate button.
- Hit the record button and record a performance.
- Save the performance (as a zip file).

Get the CSV file on your computer:

- Open the zip file in the Files App and find the CSV file ending in "_cal". This is the "calibrated" result. The full filename will be something like "MySlate_8_sarahs_iPhone_cal.csv". Email this file to yourself (or use some other means to get it from your phone to your computer)

Maya workflow:

- With puppet.ma open (or referenced), in the Maya script editor and open a Python tab, run the following if you ****opened**** the puppet.ma:

```
...  
from load_live_link_csv import load_csv_animation  
load_csv_animation("C:/path/to/MySlate_foo_cal.csv")  
...
```

- If you **referenced** the puppet.ma with a namespace, then instead run the following:

```
...  
from load_live_link_csv import load_csv_animation  
# Where "<my_namespace>" refers to the namespace you used when  
referencing puppet.ma  
# Please note that the namespace string **must** end with a ":"  
load_csv_animation("C:/path/to/MySlate_foo_cal.csv",  
target_ns="<my_namespace>:")  
...
```

- That should have produced some additional connections in the Maya scene and added new animation keys to the attributes on the `arkit_control` node. At this point, the animation from ARKit should play back on the puppet.

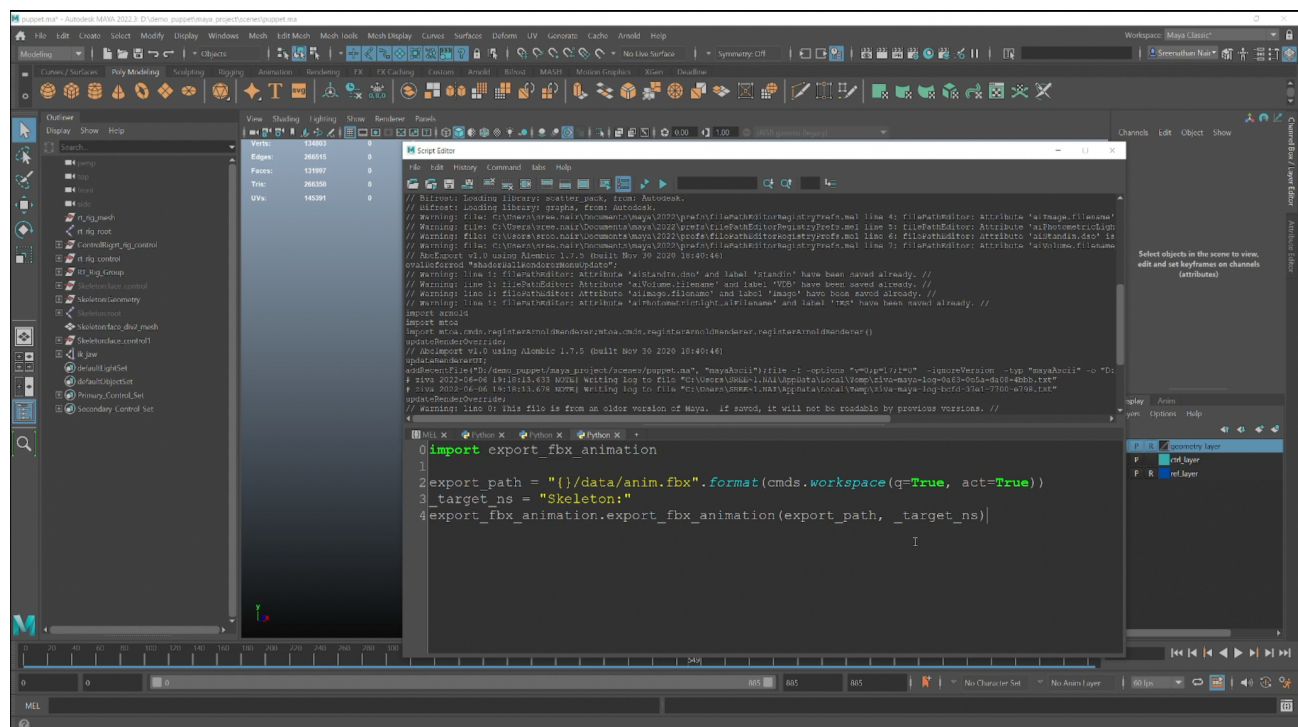
Sources:

- In the "scripts" directory of the project, there should be a "load_live_link_csv.py" file. This python file contains the maya commands that will read in a csv file and hook up the animation to the puppet.

How to Export Animation into a Game Engine

Export animations curves created on your ZFT Face from Maya to FBX

[CLICK TO WATCH TUTORIAL VIDEO](#)



Open the script editor and use the basic Maya script below (also available in the "scripts" directory):

```
import export_fbx_animation
# Default's to current project's "data" directory
export_path = "{}data/anim.fbx".format(cmds.workspace(q=True,
act=True))

_target_ns = "Skeleton:"
export_fbx_animation.export_fbx_animation(export_path, _target_ns)
```

The "export_path" variable in the script defaults to exporting to the "data" directory of the current maya project. The "_target_ns" variable contains the namespace to target in the scene. If you OPENED the

puppet then the namespace will be "Skeleton:". If you REFERENCED the puppet, then you simply have to change "_target_ns" to whatever the reference namespace is.

When the script is done exporting, it will print out the full path on disk to where it was written. The FBX can be used to hookup ZivaART in your chosen DCC or game engine.

How to Deploy a ZFT Face in Unreal Engine

[CLICK TO WATCH TUTORIAL](#)

