

# Research 2

1. Face tracking – more information's, deeper research
2. Holding an object with both hands
3. Manipulation of small objects
4. Choose software for models optimisation

## 1. Face Tracking

### [Face Tracking for Movement SDK for Unity \(meta.com\)](#)

Face tracking allows developers to map abstracted face expression and translate it into blend shapes representations.

The headset is analysing images of users face in real time and produces set of numbers representing direction of users gaze and facial movement. Images are deleted as soon as they are processed and they never leave headset.

#### Avatar facial expressions

The Face Tracking API convert facial movements detected by sensors into activation of expressions that are based on the FACS – Facial Action Coding System.

One blend shape can contain all 70 expressions or just 2 if design doesn't requires more.

The API returns a weight corresponding to the strength of the expression. This list of weights for each expression activates the blend shapes.

## FACS

[Facial Action Coding System - Wikipedia](#)

[Facial Action Coding System \(FACS\) - A Visual Guidebook - iMotions](#)

The Facial Action Coding System refers to a set of facial muscle movements that correspond to a displayed emotion.

(+) Non-Intrusive Emotion Measurement: natural emotions measurement in real-time

(+) High Temporal Precision – micro-expressions and subtle changes in facial expressions which occur in fraction of a second

(+) Objective and Quantifiable Data: numerical data useful for statistics, suitable for research and commercial applications

Action Units (AU) – mimics in certain area of face, combine together can create emotion. Example: Inner brow raised, cheek puffer, mouth stretch, head turn left, eyes up. Each of this action units is represented by a number. Combination of those numbers can represent emotions.

Emotions:

Happiness/Joy: 6 + 12 = cheek raised + lip corner puller

Sadness : 1 + 4 +15 = Inner Brow Raiser + Brow Lowerer + Lip Corner Depressor

Surprise etc.

## Blend Shapes

Eyes

Brow	Lowerer
Inner brow	Raised
Outer brow	Raiser
Eye	Closed/Look down/Look left/Look right/Look up
Lid	Tightener
Upper Lid	Raiser

Face

Cheek Chin Dimpler(fr. fossette/pl.doleczek)	Puff/Raiser/Suck Raiser_B(Bottom)/Raiser_T(Top)/
--	---

Lips

Jaw	Drop/Sideways/Thrust
Lip	Corner depressor/Corner puller/Funneler(Bottom)/ Funneler(Top)/Pressor/Pucker/Stretcher/Suck/Tightener
Upper lip	Raiser
Lower Lip	Depressor
Mouth	Left/Right

Nose

Nose	Wrinkler
------	----------

Combination

Jaw drop + lips	Toward
-----------------	--------

Tongue

Tip	Interdental/Alveolar/(Front/mid/back) dorsal (...)/out/retreat
-----	--

Blendshapes - [Setting up Blendshapes in Unity - Unity Learn](#) in Unity animation system

Blend shapes are used in 3D animation to interpolate between different sets of geometry for example from smiling into frown.

Blend shapes consist of a base Mesh and target Mesh.

Base Mesh – is a model in a neutral pose

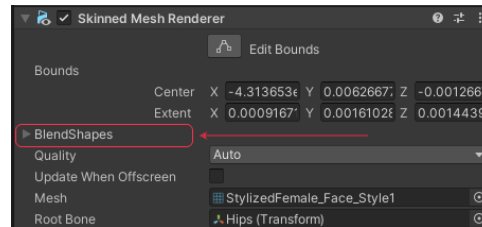
Target Mesh – is a variant of the model in different poses.

Rigged own models – Unity can only use static blend shapes from a FBX exported from DCC.

Where are the Blendshapes in Unity?

Model should contain already Blendshape and component holding it is **Skinned Mesh Renderer**.

[Unity - Manual: Skinned Mesh Renderer component \(unity3d.com\)](#)



The Skinned Mesh Renderer component will include a dropdown which holds floats representing amount of each shape applied to the current mesh state.

### Using Blendshapes with Code

**Find the Blendshape index by name** Blendshapes are contained in an array and therefore, accessed via their index. Here is a simple script which will return the index value of the Blendshape name you specify.

```
public int GetIndexForShape(string shapeName)
{
    int indexOfShape = -1;
    int blendshapeCount = faceMesh.sharedMesh.blendShapeCount;

    for (int i = 0; i < blendshapeCount; i++)
    {
        if (faceMesh.sharedMesh.GetBlendShapeName(i) == shapeName)
        {
            indexOfShape = i;
        }
    }

    return indexOfShape;
}
```

**Find the Blendshape name by index.** Conversely, you can get the name of a Blendshape at a specified index with the following, which will return a string:

```
public string GetNameForShape(int shapeIndex)
{
    int blendshapeCount = faceMesh.sharedMesh.blendShapeCount;

    if (shapeIndex > blendshapeCount)
    {
        return "";
    }

    return faceMesh.sharedMesh.GetBlendShapeName(shapeIndex);
}
```

**Get the current Blendshape value.** Once you know the index for the Blendshape, you can get its' current value with the following:

```
public float GetCurrentShapeValue(int shapeIndex)
{
    return faceMesh.GetBlendShapeWeight(shapeIndex);
}
```

What is the range of values?

**Set the current Blendshape value.** Additionally, once you know the index for the Blendshape, you can set its' current value with the following:

```
public void SetCurrentShapeValue(int shapeIndex, float valueToSet)
{
    faceMesh.SetBlendShapeWeight(shapeIndex, valueToSet);
}
```

Macro facial Expression Detector - GitHub

[Class MacroFacialExpressionDetector | Unity-Movement Documentation \(oculus-samples.github.io\)](#)

[Class CorrectivesFace | Unity-Movement Documentation \(oculus-samples.github.io\)](#)

MicroFacialExpressionDetector script is where emotions names need to be defined ~

[Obtaining the weight of each of the configured emo... - Meta Community Forums - 1029755](#)

```
public enum MacroExpressionType {  
    Happiness = 0, Sadness, Anger, Surprise  
}
```

Different micro gestures needs to be added to each emotion. Micro gestures are based on FACS system.

Weight of each micro gesture

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using Oculus.Movement.Effects;  
using System.IO;  
using TMPro;  
  
public class ExpressionThreshold : MonoBehaviour  
{  
    [Header("Emotion displayed")]  
    public TextMeshProUGUI _text1;  
    public TextMeshProUGUI _text2;  
  
    public OVRFaceExpressions faceExpression;  
  
    #region expressions and weights  
    //Add as many expressions and weights as you need  
    public OVRFaceExpressions.FaceExpression leftEyeBlink;  
    public OVRFaceExpressions.FaceExpression rightEyeBlink;  
    public float weightL;  
    public float weightR;  
    #endregion  
  
    void Update()  
    {  
        weightL = faceExpression[leftEyeBlink];  
        weightR = faceExpression[rightEyeBlink];  
    }  
  
    //We can display the expression weight using this void  
    public void WeightExpression (float expressionWeight, OVRFaceExpressions.FaceExpression expression)  
    {  
        expressionWeight = faceExpression[expression];  
  
        _text1.text = expressionWeight.ToString();  
        _text2.text = expression.ToString();  
  
        Debug.Log(expressionWeight);  
    }  
}
```

## 2. Holding an object with both hands

Component TwoHandGrabInteractable – Add it to an object

[Introduction to VR in Unity - PART 10 : TWO HAND GRAB \(youtube.com\)](#)

## 3. Manipulation of small objects

## 4. Softwares for optimisation in VR

### Files accepted by Unity

FBX – Autodesk Filmbox, developed by Autodesk, contain mesh, material, texture, geometry, skinning, lightning and skeletal animation data (from: Maya, MotionBuilder, Adobe After Effect,

DAE – based on COLLADA format, is used to exchange digital assets like images, textures and 3D models (from: Adobe Photoshop, AutoCAD, Maya, Blender)

DXF – drawing exchange format developed by Autodesk for storing CAD models

OBJ – Object type file developed by Wavefront Technologies. Contain: meshes, polygonal geometry, color, texture.

[Unity - Manual: Support for proprietary model file formats \(unity3d.com\)](#)

Softwares from which Unity can import files and convert them into .fbx: Autodesk Maya, Blender, Modo, Cheetah3D.

## Optimisation for VR

Stages and types of assets for optimization:

- Textures in Adobe Photoshop
- 3D models meshes in two stages: SolidWorks/Creo as much as its possible, then inside Blender or Maya
- Code/lights/occlusion culling /anisotropic filtering(not from distance) / mip maps (from distance) / particle's optimization / anti-aliasing / batching (for static objects) optimisation in Unity

Substance Painter and Adobe Photoshop are invaluable for texture optimization. They allow for the creation and optimization of texture maps, including normal, diffuse, and specular maps, ensuring high quality with lower file sizes.

[RizomUV](#) can help create efficient [UV maps](#) that minimize texture stretching and overlapping.

Blender vs. Maya

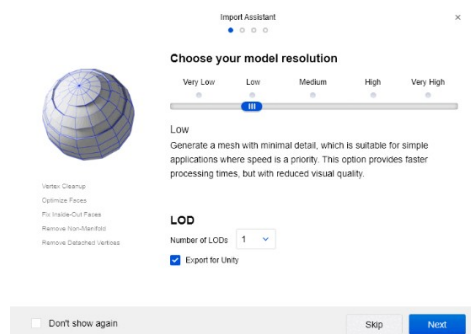
	Blender	Maya
Subscription	Free	1 875\$ +800\$ for extra packages
Modeling	Less steps for action	More steps for action
Active community	A lot of users, a lot of tutorials	Confusing
Rendering	3 rendering engines, real-time render abilities	Single rendering engine – Arnold working in preload
Animations	Phyton	MEL & Phyton
Drawing	Possible	
Visual effects	Blizzards, dust	Lava, concrete, frost, cloth
Usability	A lot of windows of blender at once	One maya window
Updates	Constantly changing	Stable, updates are released ones per year
Test with headset	Possible from Blender	<a href="#">Maya VR Modeling - Tips and Tricks (youtube.com)</a>
Interactions	Interactive elements, buttons, triggers	
Polycount	Decimate – reduction of polycounts	Smoothing polycounts exponentially (keeping shape) and linearly (control amount). Functions: Reduce, remesh, retopologize, smooth
Normal maps	Low-poly versions of detailed surfaces	
Texture	Texture atlases, Physically Based Rendering, UV Mapping, Baking Textures, Substance Tools – smart textures	UV Mapping,
Objects management		Instances of objects

**MeshMatic** – Conversion and optimization of CAD files for real-time rendering.

[Meshmatic CAD Conversion and Optimization Features \(meshmatic3d.com\)](#)

Features:

- CAD Conversion  
Supports: CATIA, SolidWorks, Creo, SketchUp with files like: STEP, IGES, STL, OBJ, FBX  
Translates CAD data to mesh and lets adjust model resolution and quality



- Mesh Simplification  
Reduction of Number Of Polygons through decimation tools, algorithms, faces cleaning
- Removal of duplicated meshes  
Automatic instances linking for modifications and updates
- LOD generator for all 3D projects, LOD – Level of detail

**Edit Profile\*** ×

Profile Name \*

3D Printing #3

Profile Options

Export Settings

**LOD**

Advanced

Number Of LODs 3

LOD 1 80%

LOD 2 33%

LOD 3 25%

☐ Save LODs in single file (for Unity)

Engineer software:

Inventor – OBJ no

SolidWorks – OBJ yes + OBJ Exporter [Unlocking New Possibilities: Exploring the OBJ Exporter for SolidWorks - ProtoTech Solutions](#)

**Exporting selected body part**

[SOLIDWORKS Tutorial - Selective Export \(youtube.com\)](#)

**Prises:** [OBJ Exporter for SOLIDWORKS With Updated Features \(prototechsolutions.com\)](#)

Creo