Join the Party in Social VR

Iffat Mai & Simon Chavez
Speakers

Iffat Mai
Firmwide Development Manager
Perkins and Will
New York

Simon Chavez
Visualization Designer
Perkins and Will
Seattle
Who are you?

- Architects
- Engineers
- Contractors
- Clients
Social VR

- What
- Why
- Who
- Where
- HOW
Learning Objectives

- Learn how to prepare a Revit model for VR export
- Learn how to bake materials and lighting in 3ds Max
- Learn how to setup rendered model in Unity game engine
- Learn how to setup social VR using Photon VR
Social VR
Social VR

Multi-user VR

Collaborative VR

Networked VR

Holodeck

Cave
Virtual Reality (VR) in Design
Basic Workflow
Rendered Workflow

Model → Render → Game Engine

Autodesk Revit, 3ds Max, Unity
Game Engine

Unity

Photon

VR

+VR SDK

VRTK

virtual reality toolkit

Unity

Photon

Unity Networking
Revit Export
Preparing Revit Model for Export

- Create a detached copy of your Revit model
- Delete anything you don’t need to see
- Use Visibility/ Graphics Settings to turn off unnecessary objects
- Use Section Box to isolate what you do want to see
- Export to FBX
- Break the model into sub groups by views if the model is too large.
Update Material in 3DS Max
FBX in 3DS Max
Link FBX into Max

- Link Revit (FBX) into Max for continuous update and changes
- Select Combine entities By Revit Material
Combine Entities By Revit Material

- Revit entities assigned the same material become a single 3ds Max object.
- The object has the name of the material.
- The object is placed on a layer that also has the name of the material.
- Revit metadata (Family and Category) does not appear in the 3ds Max Scene Explorer.
- Multi/Sub-Object material information is lost.
- Reduce Polygon Count
- Group objects as systems
Import FBX into Max

- Use Autodesk Architectural (Revit) Preset
Cleanup Step

- Convert all objects to Editable Poly
- In editable poly, click Quick Weld
- Run Pro-Optimizer (Built-in) – visualizer
- Identify super-high poly objects, run vertex cleaner script (Shiva3d)
- After everything is optimized, then select objects through proximity and materiality
- Run SmartScripts Attach/Detach for grouping (example: mullions)
List of Cleanup Scripts

- Smart attach/detach
- Vertex Cleaner Script
Vertex Cleaner

Submitted by Shiv on Fri, 2009-02-06 05:17

Version: 1
Date Updated: 05/09/2009
Author Name: Yegor Tlyda

Usage:
This script will help you to clean polygonal meshes from the useless vertex, for example after the Boolean operations. Script will find vertices with 2 edges only and, depending on angle between edges (threshold you can control), will clean it. So you can leave a corner vertices and clean vertices on straight lines.

Additional Info:
1) Download script
2) Run it via menu "MacScript - Run Script" or DragAndDrop it on 3dsmax viewpport
3) Choose "Customize - Customize user Interface" and add script to Keyboard shortcut or in Menus, Toolbars or, like on image, in to the quad. You will find it in Category: Shiv Tools

Version Requirement: 3ds Max 5 and higher

Video URL: http://www.shiva3d.net/tutorials/vertex-cleaner/vertex-cleaner-maxscript-screenshot.jpg

SmartScripts Attach/Detach

Submitted by Lande3D on Sat, 2015-12-05 03:17

Version: 1b
Date Updated: 12/05/2015
Author Name: Lande3D

I am releasing these two small macros scripts I wrote to simplify my own workflow in 3dsmax.

They are inspired by the universal attach/detach macros made by Nik and feature much of the same functionality in addition to some extra ones, but code-wise they are complete rewrites.

SmartAttach MacroScript:
With single object selected does this:
-Disable any TurboSmooth/MeshSmooth/OpenSubDiv modifier(s) that are on top of the topmost "Edit modifier"
-Start attach mode
Works for: EditPoly, EditablePoly, EditableSpline, EditSpline, EditsMesh, NURBS/Curves, (Unwrap_UVW stitches verts)

With more than one object selected does this:
-Remove any TurboSmooth/MeshSmooth/OpenSubDiv modifier(s) that are on top of each of the objects topmost "Edit modifier"
-collapses and Attaches all Poly/mesh Objects together to the last selected Poly/mesh object
-collapses and Combines/Attaches all Splines together to the last selected Spline object
-All this works in one loop, so you can select a bunch of both splines and mesh/poly objects and the script will attach them together to 2 objects. If any objects are not compatible (like cameras/ligths etc. they will be left out)
Works for: Any (not tested on NURBS/Patch)
(uses a linear attach method with some optimizations)

For tip: if you want to attach to a specific object then select all the objects you want to attach then unselect and reselect the "attach to" object if you are attaching both splines and mesh/poly at the same time then optionally deselect both the spline and the mesh/poly you want to be attached to and reselect those (it doesn't matter in which order)

SmartDetach MacroScript:
With more than one object selected:
-opens the clone dialog
-in subobject mode does this:
-VertexSplineKnot/Unwrap_UVW: break vertex/knot/uv selection
-Edge/Border: Start create shape from selection dialog and select if it if it was created
-Face/Element/Segment/Spine: Start detach dialog and if detached to a object select it

Installation instructions:
Exit 3dsmax, unpack the zip file and copy the folders contained inside to the root of the 3dsmax program folder (usually "C:\Program Files\Autodesk\3ds Max 2016" etc.) Restart 3dsmax
The macros will appear under the category "SmartScripts". Assign them to hotkeys and/or toolbar buttons

Written for and tested in 3dsmax 2016 sp1 and sp2 but should work for previous versions too (down to 2012?)

Version Requirement: 3dsmax 2012
Cleanup
Reduce Poly Count
Use Scene Converter

Select conversion script “Autodesk Material to Standard Material”

Change the Standard material name to different unique material names

Textures - Make sure your textures are sourced already from your Unity project or copied into a folder called \textures in your project

For objects with bump maps that are not exported.

- Converted bump map into Normal map
- Re-assign to objects
Light and Material Baking in Max
Render to Texture (hotkey: 0)
Export to FBX from 3DS Max to Unity

- Cameras and Lights are not currently imported into Unity
- For baking Vray lighting onto material in Max, in Unity, you want switch from standard shader to the unlit texture shader for efficiency and better frame rate.
Unity3D and template
Unity Template for Social VR

- Import the 3D FBX model
- Light Baking & Skybox
- Material adjustment
- NavMesh Baking
- Virtual Reality Toolkit (VRTK) from StoneFox
- Create the Player Prefab
- Photon Unity Networking (PUN)
Import FBX
Lighting
Sun and Lights

- Leave Sun (Directional Light) in real time
- HDR Light (change default sky)
- Add Area light (bake option only) to simulate ceiling mounted lights
  - (example 2x2 or 2x4 ratio, adjust for ft-in)
- Change color as needed
- Use point light for downlights
Skybox

- Lighting tab, select Skybox
Adjusting Glass Transparency

- Select the Glass Panel object
- Under Glass Shader
- Select the Albedo color
- Changing the Alpha channel value to 0 for full transparency
- Max out smoothness
Lightmap Static

- If light baking in Unity3D, using Enlighten
- Select all none moving objects, and check Lightmap Static option
- Select all lights (other than the Sun )
- Set the lights to bake
Generate Lighting (Baking)
NavMesh

- Nav Mesh
- Nav Mesh Agent
- Off-Mesh Link
- NavMesh Obstacle
NavMesh Baking

- Select all building objects
- Check Navigation Static
- **Adjust** agent size
  - *Agent Radius*
  - *Agent Height*
  - *Max Slope*
  - *Step Height*
- Open Navigation tab
- Click on Bake button
Game Engine
VR Download

- Download Virtual Reality Toolkit (VRTK) - https://vrtoolkit.readme.io/
- Download SteamVR (Vive) OR
- Download Oculus (Rift)
Unity Asset Store

VRTK - Virtual Reality Toolkit - [VR Toolkit]

SteamVR Plugin

Oculus Integration
VR Setup Overview

- Import Assets/VRTK folder into your Unity project
- Create an empty gameObject VRTK_SDK
- Add the VRTK_SDK Manager script to the empty VRTK_SDK Game Object
- Add the SteamVR_SDK as a child to the VRTK_SDK
VR Setup shortcut

- Open Example Scene (Basic Teleport)
- Create an empty gameobject at 0,0,0
- Rename it “Player”
- Then drag into the player
  - [CameraRig]
  - [VRTK]
- Then make a prefab out of the player
Scene Setup

- Open your new scene
- Drag “Player” prefab into your scene
- Delete the existing “Main Camera”
Player Avatar

- Create a box, add a custom logo
- Or use any 3D model, double check orientation
  - Y as UP, -Z as Forward (FBX export)
- Use the headset model from SteamVR folder
Photon vs. Unity Networking

- Host Model
- Connectivity
- Performance
- Features & Maintenance
- Master Server
Photon Unity Networking Free

Scripting/Network
Exit Games

FREE

Add to Downloads

⭐⭐⭐⭐⭐ (1637)

FREE Edition

#1 Platform for Multiplayer

✅ Unity Network Compatible
✅ FREE 20 CCU Subscription
✅ 5 Demos + Code

More than 200,000 Developers:
Get Started!

Your Free Plan is just an e-mail away:

youname@yourcompany.com

Register

I'm not a robot
Your Photon Cloud Applications

Show in Status Sort by Order
All Apps Active Peak CCU Descending

Create a new App

App ID: c3b7187f...
This app is on the free plan. We recommend you to upgrade before using it in production.

Plan Current Month Peak Previous Month Rejected Peers
20 CCU 0 CCU 0

Analyze Manage Change CCU Add Coupon / PUN

App ID: 911a6e64...
This app is on the free plan. We recommend you to upgrade before using it in production.
Lobby, Create Room and Join Room
PUN – Photon Unity Networking

- Installing the Photon Unity Networking package
- Setup Photon account
- Implement server creation and joining an existing host.
- Spawning as a player and how objects can be created on the network.
- Network communication using State Synchronization and Remote Procedure Calls.
- Interpolating and predicting values between data packages.
PUN Wizard

This window should help you find important settings for PUN, as well as documentation.

- Locate PhotonServerSettings
- Cloud Dashboard Login
- Setup Project

Converter
- Start
- Reference PDF
- DevNet / Manual

Documentation
- I am not quite sure how ‘my own host’ compares to ‘cloud’.
- Cloud versus OnPremise
- Open Forum

Client Settings
- Auto-Join Lobby
- Enable Lobby Stats

Hosting
- Photon Cloud
- Region: Us
- AppId
- Protocol: Udp

Rpc List
- Refresh RPCs
- Clear RPCs
- Log HashCode

Photon Voice Settings
- Voice AppId
Create Network Manager

- Create NetworkManager under Player
- Create Net Manager Script
- Create Photon Voice Settings script
Connect

- Create Net_Manager script
  - Add connection string to Start()
  - Add OnConnectedToMaster() ➔ PhotonNetwork.JoinRandomRoom()
  - Add OnPhotonRandomJoinFailed() ➔ PhotonNetwork.CreateRoom()
  - Add OnFailedToConnecttoToPhoton() ➔ Disconnect if failed to join
  - Add OnJoinedRoom ➔ Successfully joined a room
    ➔ Instantiate game objects

PhotonNetwork.Instantiate(PrefabName, position, rotation, group, data)
public class NetManager : MonoBehaviour {

    public GameObject headPrefab;
    public GameObject leftHandPrefab;
    public GameObject rightHandPrefab;

    public void OnJoinedRoom()
    {
        Debug.Log("OnJoinedRoom() called by PUN. Now this client is in a room. From here on, your game would be running. For reference, all callbacks are listed in PhotonNetwork.Instantiate(headPrefab.name, ViveManager.Instance.head.transform.position, ViveManager.Instance.head.transform.rotation, 0);
        PhotonNetwork.Instantiate(leftHandPrefab.name, ViveManager.Instance.leftHand.transform.position, ViveManager.Instance.leftHand.transform.rotation, 0);
        PhotonNetwork.Instantiate(rightHandPrefab.name, ViveManager.Instance.rightHand.transform.position, ViveManager.Instance.rightHand.transform.rotation, 0);
    }
}
Resources

Assets ▶ Prefabs ▶ Resources
- Materials
- Avatar
- AvatarUnwrap
- LHand
- Player
- PWLogoAvatar
- PWLogoAvatar
- RHand

Net Manager (Script)
- Script: NetManager
- Head Prefab: Avatar
- Left Hand Prefab: LHand
- Right Hand Prefab: RHand
- Auto Connect: On
- Version: 1
Net Manager

```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class NetManager : MonoBehaviour {

    public GameObject headPrefab;
    public GameObject leftHandPrefab;
    public GameObject rightHandPrefab;

    /// <summary>Connect automatically? If false you can set this to true later.</summary>
    public bool AutoConnect = true;

    public byte Version = 1;

    /// <summary>if we don't want to connect in Start(), we have to "remember" it</summary>
    private bool ConnectInUpdate = true;

    public virtual void Start() {
        // below, we implement some callbacks of PUN ...
        public virtual void OnConnectedToMaster() {
        }

        public virtual void OnJoinedLobby() {
        }

        public virtual void OnPhotonRandomJoinFailed() {
            // the following methods are implemented to give you some context. re-implen
        }

        public virtual void OnFailedToConnectToPhoton(DisconnectCause cause) {
        }

        public void OnJoinedRoom() {
        }
    }
}
RenderTexture & Texture2D

- A very quick way to make a live selfie camera in your game:
  - Create a new Render Texture asset using Assets > Create > Render Texture.
  - Create a new Camera using GameObject > Camera.
  - Assign the Render Texture to the Target Texture of the new Camera.
  - Create a wide, tall and thin box to represent the selfie camera screen.
  - Drag the Render Texture onto it to create a Material that uses the render texture.
  - Enter Play Mode, and observe that the box’s texture is updated in real-time based on the new Camera’s output.
Selfie Group Shot!
Thank you

Please fill out your survey

Iffat.Mai@perkinswill.com
Simon.Chavez@perkinswill.com