Discussion 2

Project Update
Material Loading and Touch Controller Support
Line-Sphere Intersection
Others: Oculus Avatar + Run code without Rift

Project Update

Let’s go through the project page:

http://ivl.calit2.net/wiki/index.php/Project1S17

Material Loading

The tutorial previously posted only grabs the texture information from the material.

We need to grab the **ambient/diffuse/specular/shininess** values.

Material Loading

Example:

```
material->Get(AI_MATKEY_COLOR_DIFFUSE, color);
```

This fills out a aiColor3D with the corresponding attribute.

Material Loading

Use the info to fill out your own material class or struct that you send to a uniform in your shader.

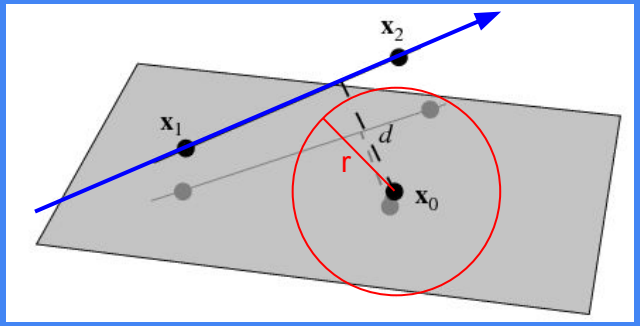
Oculus Touch Controller Support

- Oculus PC SDK Developer Guide contains all the detailed information about the controller: Hand Tracking, Button State, Button Touch State and Haptic Feedback. https://developer3.oculus.com/documentation/pcsdk/latest/concepts/dg-input-touch/
- A few simple examples on how to get the position and trigger press states and vibration feedback to work with the Oculus Controller: https://rdmilligan.wordpress.com/2016/12/10/oculus-touch-controllers-with-c/
- Simply putting these state check in your render loop.

Code sample from the simple tutorial

```
ovrInputState inputState;
bool leftHandTriggerPressed = false;
if (OVR_SUCCESS(ovr_GetInputState(session, ovrControllerType_Touch,
&inputState))) {
    if (inputState.HandTrigger[ovrHand_Left] > 0.5f) {
        leftHandTriggerPressed = true;
    }
}
// render
if (leftHandTriggerPressed) {
   Meshes[i]->Render(view, proj);
}
```

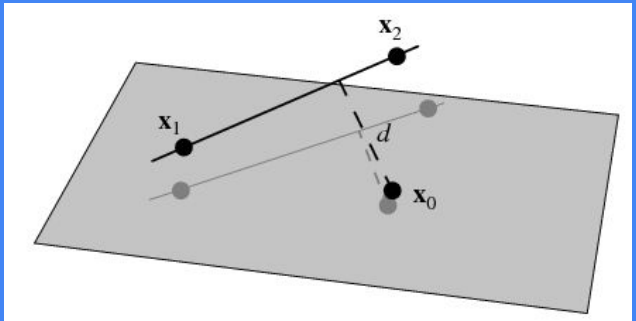
Line-Sphere Intersection



Imagine the line is a laser from our controller, **x1, x2** are two points on the line. **x0** is the center of the rendered co2 model, by comparing the distance **d** with the sphere radius **r,** we can decide if the line intersects the sphere or not.

- Point-line distance reference: http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html

Line-Sphere Intersection



Point-Line Distance Review:

- **x1**=(x1, y1, z1), **x2** = (x1, y1, z1)
- A vector along the line is given by:

$$\mathbf{v} = \begin{bmatrix} x_1 + (x_2 - x_1)\,t \\ y_1 + (y_2 - y_1)\,t \\ z_1 + (z_2 - z_1)\,t \end{bmatrix}.$$

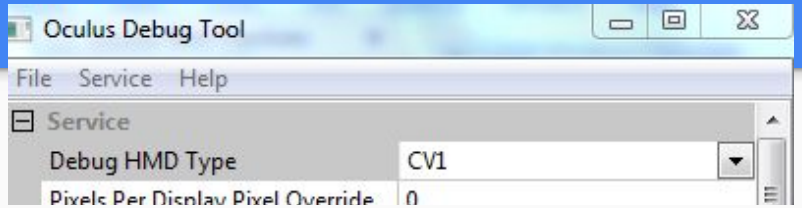
$$d = \frac{|(\mathbf{x}_2 - \mathbf{x}_1)\times(\mathbf{x}_1 - \mathbf{x}_0)|}{|\mathbf{x}_2 - \mathbf{x}_1|} = \frac{|(\mathbf{x}_0 - \mathbf{x}_1)\times(\mathbf{x}_0 - \mathbf{x}_2)|}{|\mathbf{x}_2 - \mathbf{x}_1|}$$

- We then can calculate distance squared distance **D** between a point on **v** and the target point **x0.**
- By taking the derivative of **D** and set it to 0, we can calculate t based on the three points.
- Using t we can calculate the shortest distance, **d=sqrt(D),** between **x0** and **v**.

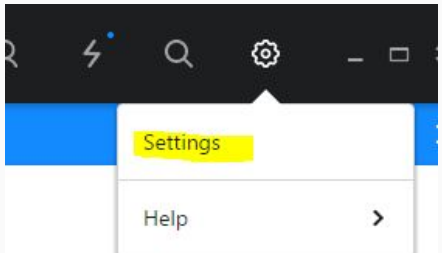
Oculus C++ Avatar Support

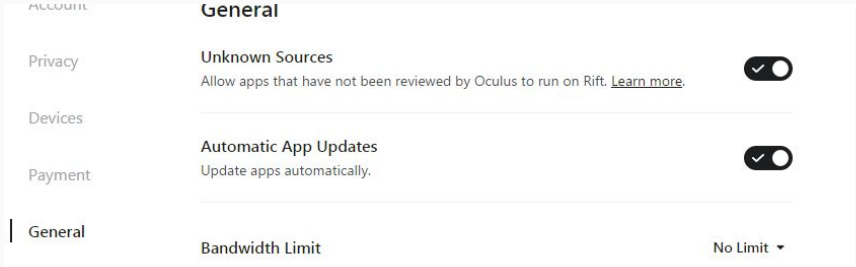
- Avatar SDK: https://developer.oculus.com/downloads/package/oculus-avatar-sdk/
- Official documentation on how to use Avatar SDK: https://developer3.oculus.com/documentation/avatarsdk/latest/concepts/avatars-gsg-native-intro/
- Simple online samples on how to use the Avatar SDK: https://developer3.oculus.com/documentation/avatarsdk/latest/concepts/avatars-sdk-native-intro/
- This is a bonus feature (not required)

Run code without the headset



- Tutorial: https://forums.oculus.com/developer/discussion/35639/developing-for-rift-without-a-headset
  - OculusSDK\Tools\OculusDebugTools.exe app
  - Allow unknown resources to use Rift
  - Mirro texture: PC developer's guide:
    "If no Rift is plugged in during detection,ovrHmd_Create(0)will return a null handle. In this case, you can useovrHmd_CreateDebugto create a virtual HMD of the specified type. "





References

- Official Assimp Material Documentation: http://www.assimp.org/lib_html/structai_material.html
- Official Touch Controller Documentation: https://developer3.oculus.com/documentation/pcsdk/latest/concepts/dg-input-touch/
- Unofficial Touch Controller Tutorial: https://rdmilligan.wordpress.com/2016/12/10/oculus-touch-controllers-with-c/
- Official Oculus Avatar SDK: https://developer.oculus.com/downloads/package/oculus-avatar-sdk/

References

- Official Oculus Avatar SDK Documentation: https://developer3.oculus.com/documentation/avatarsdk/latest/concepts/avatars-gsg-native-intro/
- Unofficial Oculus Avatar Tutorial: https://rdmilligan.wordpress.com/2016/12/10/oculus-touch-controllers-with-c/
- Unofficial Description for Running Oculus SDK without the Rift: https://forums.oculus.com/developer/discussion/35639/developing-for-rift-without-a-headset

References

- Line-Point Distance: http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html