Hello and welcome to the **Qt Design Studio: Realistic Visuals**! My name is Piotr, and I’ll be your instructor for this course.

In this video, I will explain the basic concepts that will help you achieve realistic visuals in the Qt Design Studio. This is the second part of a 3D-focused learning path designed to help you better understand how to use Qt Design Studio as a 3D tool.

This video will be divided into two parts:

First, I’ll introduce you to various technics that you can find in Qt Design Studio, and that will help you create more visually attractive 3D scenes. Then I’ll we will use our ADAS scene, that we’ve made in previous video, and use the knowledge we have gain to make it even better!

**HDR maps**

Let’s start by understanding **HDR maps**.

**HDR maps**, or **High Dynamic Range** maps, are specialized images that capture a wide range of lighting information, far beyond what standard image formats can handle. Unlike regular images that clip extreme highlights or shadows, HDR maps preserve details in both bright and dark areas. This makes them perfect for simulating realistic lighting and reflections in 3D scenes.”

In 3D rendering, HDR maps are often used as **environment maps**. They wrap around a scene, providing not only a visual background but also lighting and reflection data. This enhances realism, as objects in the scene interact naturally with the environment’s light and reflections.

With their ability to adjust lighting without losing detail, HDR maps are essential for creating immersive and lifelike 3D environments.

To add a light map we need to **import it to the project first**. We can do that by either the **Assets Window** or **Material Browser**. Let’s do that with **Assets Window**. First click **Plus** button, select the map and click **OK**. Now we have the map added to the Assets but it’s not visible in the texture section in the Material Browser. That’s because we need to select it first, click **Right Mouse Button** and select **Add Light Probe**.

Now, if we’ll go to the **sceneEnviroment** properties and scroll down to image-based lighting property we can change the map to the added light probe.

Notice the immediate change in lighting and how elemets of HDR map are reflected in the glossy materials.

From here we can also change the **Exposure**, which is basically the amount of light emitted by the light probe. The higher the exposure, the lighter the scene becomes.

**Horizon property**, when defined with increasing values, adds black color to the bottom half of the environment, thus forcing the lighting to come predominantly from the top of the image and removing specific reflections from the lower half. This property is useful for accounting for a ground plane that would have the effect of obscuring the reflection of the light probe from the ground. This is necessary because **light probe** contributions come directly from the image without consideration for the content of the scene. You can set the value of the **Horizon property** between **0.0** and **1.0.** Using the default value of the property applies the entire light probe without adjustment.

**Orientation** allows us to manipulate the map. Changing **Y axis** is especially useful, because it allows us to rotate the map horizontally.

When using HDR maps it is also important to use proper **Tone Mapping** that will suit your needs.

**Tone mapping** determines how colors are rendered in the scene. Available options include:

* **Linear** which produces gamma-corrected sRGB colors.
* **ACES** which utilize the Academy Color Encoding System for cinematic visuals.
* **HejlDawson** that applies a specific tone curve for unique results.
* and **Filmic** which provides a cinematic and dynamic look.

In general, if you want to have realistic lighting it is good to go with **Linear**, which gamma corrects colors and returns them in sRGB color space

**Reflections**

Next important topic in the realistic visual concepts are reflections.

In real life every object receives and cast reflections. To simulate this behavior in 3D engines various technics are used. One of these technics are **Reflection Probes**. And this is exactly what **Qt Design Studio** is using.

**Reflection probes** capture localized reflections for objects within a defined **Box Size**. That means that objects that are in the box will receive reflections and those outside it not. It does not mean that the objects outside of the box will not be reflected since Reflection probe acts as a sort of 360 camera that is catching everything it sees into a map. It can be **done once** or **every frame** if you want to have real time reflections.

If you want some model to be excluded from the reflection you need uncheck Cast Reflections option in the objects’ properties window.

Reflections are turned off by default. If you want the object to receive reflections, you must turn it in the object’s property by toggling **Recieves Reflections** in its properties. As you can see, by toggling this property on, spheres are being added to the reflections.

To add a reflection Proble to the scene, simple find it in the Components window by typing Reflection Proble in the Search field. Next, drag it to the scene node in the Navigation Window.

The **Reflection Probe** in Qt Design Studio comes with a variety of properties that allow you to customize reflections within your 3D scene.

The **Box Size** property defines the dimensions of the probe’s bounding box, which determines the area where objects can “see” the captured reflections. For example, our box size is **1000, 1000, 1000** which creates a large cubic area for reflections. If you need to adjust its position relative to its default placement, the **Box Offset** allows you to shift the box without moving the probe itself.

To **improve accuracy**, enabling **Parallax Correction** ensures reflections are adjusted to account for the viewer’s perspective, especially when objects are close to the edges of the bounding box.

For troubleshooting or fine-tuning, you can turn on **Debug View**, which visualizes the probe’s area in the **render view**, helping you confirm its placement and effectiveness.

**The Clear Color** property lets you define a fallback background color when generating the reflection map. By default, it’s transparent, but you can choose a solid color as needed.

When it comes to the quality of reflections, Reflection Map Quality gives you options ranging from **VeryLow** to **VeryHigh**, balancing visual detail with performance demands. For updating the reflection map, the **Refresh Mode** property provides two choices:

* **FirstFrame**, which generates the reflection map once at the start for static scenes.
* **EveryFrame**, which continuously updates the reflections, ideal for dynamic scenes but at a higher performance cost.

To optimize reflection updates, **Time Slicing** splits the update process across frames. You can choose None to update everything in a single frame, **AllFacesAtOnce** to process the cube map all at once, or **IndividualFaces** to update one face per frame for smoother performance.

By fine-tuning these properties, you can achieve high-quality, efficient reflections tailored to the needs of your scene—whether it’s static or dynamic. This is especially helpful If we want to optimize our scene for a specific render budget.

Lastly, the **Override Texture** property lets you apply a pre-baked reflection map instead of generating one dynamically. This is useful when you want to use custom environment maps to control the appearance of reflections more precisely.

**Material Editor and Browser**

Next important concept in Qt Design Studio are materials that basically lets you define how 3D models are supposed to look.

In **Qt Design Studio**, **Material Editor** and **Material Browser** allow you to create, manage, and customize materials efficiently.

There are couple of material types in Qt Design Studio that you can choose from.

For simple objects, the **Default Material** offers basic color and texture options.

If realism is your goal, the **Principled Material** is the go-to choice, providing physically accurate rendering with control over properties like **Metalness** and **Roughness**, as well as support for texture maps.

**Specular-Glossy Material** allows simple control over reflectivity and glossiness for less complex workflows. For ultimate flexibility, the **Custom Material** lets you use GLSL shaders to create unique, highly stylized effects.

Each material type serves a specific purpose, from realism to performance or creative freedom, ensuring you can craft exactly the look you need for your 3D scene.

To add a material go to the Material Browser and click Add a Material button. This will create a default, new material. To add that material to the scene drag and drop it in the **3D** or **Navigator** view. Alternatively, right-click the material and select “Apply to selected” to instantly update your model’s appearance.

Now let’s check the Material Editor and check our new material!

As you can see the material type is Principled Material and that’s perfect since we want to focus on the **Realistic Visuals.**

**Principled Material**

The **Principled Material**, based on the **Physically Based Rendering** metal/roughness workflow, provides a versatile and intuitive approach to crafting realistic materials.

Let’s dive into the properties and explore how they can transform your model.

**Understanding the Basics**

At its core, the Principled Material revolves around three fundamental properties: **metalness**, **roughness**, and **base color**. These attributes define whether a surface appears metallic, how smooth or rough it is, and the base color’s influence on light interaction.

1. **Base Color** combines diffuse and specular information, depending on the metalness value. For dielectrics, the base color defines the diffuse light scattering, while for metals, it largely determines the specular reflection. Use a texture to define the base color of the material. Enable **Use Single Channel** if you need to extract data from a single channel (e.g., red or green).

* **Metalness** tells the renderer whether the material is metallic or non-metallic (dielectric). Metallic surfaces, like gold or steel, are highly reflective and rely heavily on their base color for specular reflections. Setting the **metalness** to 0 makes the surface dielectric, ideal for materials like plastic or wood. Intermediate values create blended effects, such as corroded or partially reflective metals. use map if you want to have more control over which part are supposed to be reflective or not.
* **Roughness** describes the texture of the surface. A lower roughness value produces a smooth, mirror-like finish, while higher values result in a rough, matte appearance. This property is key to defining how polished or weathered a material looks. Similar to metlaness, you can use map to have more control over that.

**Controlling Transparency**

The Principled Material offers flexible transparency options through its **alpha modes**. For fully opaque objects, use the default setting. If you want parts of your model to disappear based on a texture, switch to **Mask** mode and adjust the **alphaCutoff** value to set the transparency threshold. For smooth transparency transitions, like glass or foggy materials, the **Blend** mode is your go-to.

* **SourceOver** ensures opaque objects occlude those behind them.
* **Screen** lightens the resulting color, ideal for glowing or translucent effects.
* **Multiply** darkens colors, great for overlays or shading effects.

**Adding Advanced Surface Features**

For materials that require an extra layer of realism, the Principled Material provides powerful tools:

1. **Clearcoat Layer** adds a glossy finish that sits on top of the base material. Just like in a shiny car hood or a polished wooden table. You can adjust **Amount** to control its intensity and **Roughness Amount** for its smoothness. If you want to vary the effect across the surface, you can use **Map** to add texture. **Fresnel Effects** adjust how light reflects off surfaces at different angles. Use **fresnelScale**, **fresnelBias**, and **fresnelPower** to fine-tune the balance between head-on and glancing reflections.
2. **Normal and Height Maps** create the illusion of depth and fine details without adding complexity to the geometry. A **normal map** simulates tiny bumps and grooves by altering how light interacts with the surface, while a **height map** uses parallax occlusion mapping to create a more pronounced sense of depth. Fine-tune these effects with **Strength** and **Amount** for precise results.
3. To control **Specularity** go to **Overrides** section where you can use **specular Map** for varying specularity and **specular ReflectionMap** to add environment-based reflections, such as a shiny floor reflecting a window. This way you can override global Enviroemnt Maps. Use the **Amount** property to define the strength of highlights, making surfaces appear glossy or matte.

**Simulating Light Interaction**

The **Principled Material** offers several features to help **simulate optical effects. You can find them in Refraction property**:

1. **Transmission and Thickness**: Transparent materials, like glass or thin plastic, benefit from the **Transmission Factor**, which determines how much light passes through. Pair this with the **thickness Factor** to simulate the depth of the material for effects like frosted glass or tinted windows.
2. **Index of Refraction (IOR)** controls bending of light through transparent materials, like water or glass. It is controlled by the **Index of Refraction**. Typical values range from **1.0** which is the equivalent of air to around **1.5** which is how glass bends light. This property is essential for creating convincing transparent or translucent materials.
3. **Emissive Color** letsyou to create materials that glow, like neon lights or LED panels, use the **Factor** to define the light color and strength. Combine it with an **Map** for textured lighting effects.
4. **Opacity and opacity Maps**: let’s you control transparency of the model (not to confuse with transmission).

**Refining Shading with Fresnel and Occlusion**

For better control how material receive light you can use **Occlusion** property. Simulate shadows in crevices or under overhangs with by using a **Map**, which defines areas that receive less light. Then use the **Amount** slider to control the strength of this effect.

For efficiency, enable **vertexColors** to use mesh data for color or mask properties. You can even use **vertexColorsMaskEnabled** to dynamically control roughness, metalness, or other features based on vertex data.

**Content Library**

If you want to quickly use some nice materials remember that you can uses the **Content Library** which offers a rich collection of materials, textures, effects, and environment bundles. This feature, exclusive to the Enterprise license, simplifies your design workflow by giving you access to pre-made, high-quality assets that can be seamlessly integrated into your projects. Let’s explore how you can use it to bring your designs to life. (check if that’s relevant)

To use it simply go to Content Library window and browse the selection.

**Materials** define the look and feel of your 3D models, and the Content Library makes it incredibly easy to apply them:

1. **Finding and Downloading Materials**

Start by browsing the library to find the perfect material for your model. Once you’ve identified one, download it to your project.

2. **Adding Materials to Your Project**

When the material is ready, you can add it by clicking the “+” icon or by right-clicking it and selecting “Add an instance to the project.” This step ensures the material is available in your project for use.

3. **Applying Materials to Models**

To give your model a new look, drag and drop the material onto it in the **3D** or **Navigator** view. Alternatively, right-click the material and select “Apply to selected” to instantly update your model’s appearance.

**Textures** and **Environments** can add depth and realism to your scenes:

1. **Incorporating Textures:** Right-click on an image in the library and choose how to use it:
   1. **Add as an image:** Makes it accessible in the Assets view.
   2. **Add as a texture:** Adds it to the Material Browser under Textures.
   3. **Add as a light probe:** Uses the image for realistic scene lighting and as a skybox.
2. **Customizing Environments:** Use light probes or HDR images to create immersive environments that enhance your scene’s lighting and reflections.

**Effects** can dramatically enhance the impact of your designs. You can add an effect by dragging it into your **3D** or **Navigator** view, or by right-clicking and selecting “Add an instance.”

**Managing and Reusing User Assets**

One of the most powerful features of the Content Library is the **User Assets** section. Here, you can store custom assets that you want to reuse across multiple projects:

**To save an assets just** Right-click it in **Assets**, **Navigator**, **Material Browser or 3D view** and choose “Add to Content Library” to save it for later.

Now you can simply add them to the scene in this or any other project you’re working on.

For **materials**, right-click and select “Add an instance to project” or drag them directly onto your 3D model.

For **textures**, drag them onto a model to apply them instantly or use options like “Add as image” or “Add as light probe.”

Last but not least, for **3D components**, simply drag them into your scene or right-click and choose “Add an instance to project.”

**Adjusting our ADAS screen**

Now let’s use all that knowledge and adjust our ADAS scene.

First let’s take care of that road. I think it would be nice if the road would fade out to the horizon. We can do that by adding a gradient map to opacity.

Go to Content Library and then select Textures. In here click to 4K Gradient map to download it, then right click on it and select add texture. Something is wrong. It’s the orientation. To change that we need to go to Material Browser and double click on the 4K Gradient. In here let’s change rotation to 90 and check Flip U. That is better.

I think it would be nice to have some texture on it too. Go to Content Library and download 4k Arrow Pattern then add it as a texture.

Now go to Normal property and select the map. Looks weird, we need to change the scaling. Double click on the texture in Material Browser and change scale to 6 and 60. That’s better.

Now let’s adjust that pain job on the car.

Select **carModel** in the **Navigator**, right click on it and select **Edit Component**.

Let’s check the Material Browser and search for **carPaint to start adjusting it’s properties**.

First, we want to change the amount of **Metalness** to **1,0** and **Roughness** to **0,4**. Next go to Clearcoat and change it to **0,6.** Don’t worry that the model looks a little off, this is because we edit it outside the main scene where we have defined the environment and lights.

I think it would be also nice to have glowing tail lights. To do that we need to adjust two materials. First let’s adjust the GlassRedLights – this material is basicly the tail light shade so we need it to be a little bit transparent and slick. Change Roughness to 0,0 and Opacity to 0,9.

Now let’s select GlassMatRedLights – this material is the actual LED light that we have in the shade. We want it to emit light so let’s change the emmisive colors’ factor to 16 in red channel. Great! Now let’s save it and go back to the main scene by selecting it in the main bar at top.

As you can see the the lights are glowing and the carpain looks much more realistic that before.

Since we have glowing lights, it would be also nice to add a little bit of glow effect to the render view.

Let’s select sceneEnviroment and check Enabled in the Glow section and adjust it strength to 1,5 and Blur Level to 2.

Perfect!

Now let us save the project and click Run Project button so we can see our ADAS screen in its full glory!

Thank you for watching!