

Table of Contents for WolvenKit Tutorial: From Cyberpunk 2077 to Blender

1. Introduction to WolvenKit

- **Tutorial objectives**
- **What is WolvenKit?**
- **Primary uses of WolvenKit for Cyberpunk 2077**
- **Links for more information on WolvenKit and the Cyberpunk 2077 modding community**

2. Installing WolvenKit

- **Downloading WolvenKit** (with link to the official site or GitHub)
- **Step-by-step installation**
- **Initial configuration of WolvenKit**
- **Installation verification**

3. Exploring the WolvenKit Interface

- **Overview of main sections and tools**
- **Explanation of WolvenKit menus and panels**
- **Initial settings for easy export to Blender**
- **Links to introductory videos about the interface**

4. Loading and Exploring Cyberpunk 2077 Files

- **Accessing game files in WolvenKit**
- **Understanding the Cyberpunk 2077 file structure**
- **Finding and previewing 3D models, textures, and other assets**
- **Using filters to locate specific files**

5. Exporting 3D Models from WolvenKit

- **Preparing files for export**
- **Exporting files in compatible formats (FBX, OBJ)**
- **Common export issues and solutions**
- **Links to resources and forums for troubleshooting exports**

6. Importing Models into Blender

- **Opening Blender and initial setup**
- **Importing a 3D model exported from WolvenKit**
- **Basic adjustments for correct model display**
- **Scale and orientation settings**

7. Setting Up Materials and Textures in Blender

- Retrieving textures exported from WolvenKit
- Applying materials and textures in Blender
- Adjusting shaders for optimal rendering
- Tips for enhancing realism in Blender (normal maps, roughness, etc.)

8. Model Optimization and Cleanup in Blender

- Cleaning unnecessary geometry
- Reducing polygons if needed
- Performance improvements
- Recommended tools and plugins for optimization in Blender

9. Final Export and Model Use

- Exporting the optimized model from Blender
- Possible uses in other game engines or 3D software
- Links for deeper integration in personal projects

10. Additional Tips and Resources

- Communities and forums for questions
- Resources for advancing in Cyberpunk 2077 modding
- Articles, videos, and guides on WolvenKit and Blender

Links and Resources

To learn more about Cyberpunk 2077 modding and the tools used, here are some useful resources:

- [BlackRavenVR Discord Community](#): A space to ask questions, share projects, and learn alongside other modders.
- [BlackRavenVR Patreon](#): Access advanced tutorials and exclusive content to enhance your modding skills.
- [WolvenKit GitHub Page](#): Official download, documentation, and latest updates.
- [Download Blender](#): Link to download Blender, essential for working with exported 3D models.
- [Red Modding Wiki](#): An essential source for comprehensive guides on using WolvenKit and modding Cyberpunk 2077.

1. Introduction to WolvenKit for Cyberpunk 2077

This comprehensive tutorial will guide you through **WolvenKit**, a powerful tool for modding **Cyberpunk 2077**. You'll learn how to explore game assets and export them to **Blender** to customize models, textures, and other elements.

Tutorial Objectives

This guide aims to help you:

- **Explore and understand Cyberpunk 2077 files** using WolvenKit.
- **Export 3D models, textures, and other assets** for use in Blender.
- **Import these elements into Blender** for further customization and integration into your personal projects.

This beginner-friendly guide is structured to be welcoming and easy to follow. For additional help, join the [BlackRavenVR Discord community](#), a space for questions, support, and sharing projects. [BlackRavenVR's Patreon](#) also provides exclusive resources for enhancing your skills.

What is WolvenKit?

WolvenKit is an open-source tool allowing users to **view, modify, and export** game files from **Cyberpunk 2077**. Developed by the Cyberpunk modding community, it provides deep access to game files, including 3D models, textures, sounds, animations, and more. WolvenKit is essential for anyone looking to customize or create content within the **Night City** universe.

Primary Uses of WolvenKit for Cyberpunk 2077:

- **Explore game assets:** visualize and manipulate 3D models, textures, animations, and other resources.
- **Export objects and characters** to compatible formats like **FBX** for use in 3D software like **Blender**.
- **Edit textures and materials** to create unique elements for the game.
- **Customize visual and technical elements** to create mods tailored to individual preferences.

For a complete description of WolvenKit's capabilities and additional modding resources, refer to the [Red Modding Wiki](#).

Why Use WolvenKit with Blender?

Blender is a free, open-source 3D modeling software offering a wide range of tools for editing, texturing, and animating 3D models. Combining WolvenKit's ability to export Cyberpunk 2077 assets with Blender's power allows you to:

- Edit models in detail to integrate them into your own projects.
- Customize textures and materials for a unique visual style.
- Create realistic or stylized renders using Blender's advanced options.

To download **Blender**, visit the official [Blender website](#).

2. Installing WolvenKit

To start using **WolvenKit** for **Cyberpunk 2077** modding, a proper installation and setup are essential. This step-by-step guide will ensure that everything is configured correctly for smooth asset exploration and export.

Downloading WolvenKit

1. **Access the official site:** You can download WolvenKit directly from its [official GitHub page](#), which contains the latest versions and documentation.
2. **Choose the correct version:** On the **releases** page, locate the latest stable version of WolvenKit. Click on the **.exe** or **.zip** file for the latest release to start the download. A stable version is recommended to avoid bugs.
3. **Cyberpunk 2077 modding community:** For additional advice on versions and installation, also refer to the [Red Modding Wiki](#), which offers guides and recommendations.

Installing WolvenKit

1. **Run the installer:**
 - If you downloaded an **.exe** file, double-click to start the installation.
 - If you downloaded a **.zip** file, extract it to a folder of your choice. Then, open the extracted folder and run **WolvenKit.exe**.
2. **Follow installation steps:**
 - During installation, choose a location where WolvenKit can access Cyberpunk 2077 game files. Ideally, this folder should be on the same drive as the game for faster file access.
 - WolvenKit may require **administrator permissions**. Click **Yes** if Windows prompts for permission to install.
3. **Set up project folders:**

- Upon opening WolvenKit, you'll be asked to configure folders containing game files. Select the main Cyberpunk 2077 folder on your PC, usually located in the **Steam** or **GOG Games** directory if downloaded from those platforms.

Initial Configuration of WolvenKit

1. Select the game's resource folder:

- In **WolvenKit**, go to **Settings** and ensure that the path to **Cyberpunk 2077** game files is configured.
- Under **Game Path**, specify the folder where Cyberpunk 2077 is installed, e.g., C:\Program Files (x86)\Steam\steamapps\common\Cyberpunk 2077.

2. Enable plugins and extensions (optional):

- WolvenKit offers **plugins** and **extensions** to expand its functionality. Refer to the [Red Modding Wiki](#) to learn more about compatible extensions for advanced modding.

3. Verify the installation:

- Once setup is complete, test the installation by trying to open basic game files in WolvenKit. This ensures the software has access to **Cyberpunk 2077** files.
- If you encounter any errors, check that paths are correct and that WolvenKit is allowed to access the files.

3.Exploring the WolvenKit Interface

With **WolvenKit** installed, it's time to explore its interface and familiarize yourself with the tools for working with **Cyberpunk 2077** files. WolvenKit provides a rich, intuitive interface specifically designed for modding. This section introduces each component to help you navigate and understand the main tools.

Overview of WolvenKit's Main Sections

1. Main Menu:

- **File:** Allows you to open, save, and manage your projects.
- **Edit:** Includes options to undo or redo changes, helpful when exploring or modifying game files.
- **View:** Customize the interface layout by hiding or showing specific sections to suit your workflow.
- **Tools:** Provides advanced options for asset manipulation and project management.
- **Help:** Links to documentation and support resources.

2. Navigation Panel:

- Located on the left, this panel allows access to game files. Here, you'll find various folders containing **Cyberpunk 2077 assets**: 3D models, textures, sounds, animations, etc.
- Navigation is simplified with filters and sorting options to help you quickly find the files you need. For more details on file structure, check out the [Red Modding Wiki](#).

3. Preview Panel:

- When a file is selected in the navigation panel, its content appears in the central preview panel.
- **3D Models**: WolvenKit allows you to preview 3D models directly, making it easy to identify assets.
- **Textures and Materials**: You can also preview textures, letting you examine details before exporting.
- **Animations and Other Assets**: Some files, like animations, can be partially previewed to verify their content.

4. Properties Panel:

- Located on the right, this panel displays detailed information and properties for the selected files.
- It allows you to adjust certain asset parameters, such as material properties or animation details. This is especially useful for preparing files before exporting them to Blender.

Using Menus and Tools in WolvenKit

1. Filtering and Searching for Files:

- Cyberpunk 2077's database is extensive, so finding specific files can be challenging. Use the search bar and category filters to narrow down results.
- Categories like **Meshes** (for 3D models) and **Textures** help you target your search. This feature is essential for avoiding lengthy folder browsing.

2. Exploring Assets:

- WolvenKit allows you to quickly view asset details. For example, selecting a 3D model lets you inspect its structure and geometry.
- For textures, you can check resolution and layer types. This preview is helpful for understanding if the asset fits your needs before exporting.

3. Preparing Assets for Export:

- Before exporting, make sure to review each asset's details in the **Properties Panel**. This step allows you to make adjustments if necessary, like modifying materials or optimizing models for use in Blender.

- Once the asset is ready, you can export it using the **Export** option (this will be covered in detail in the next section).

4. Loading and Exploring Cyberpunk 2077 Files

Now that you're familiar with the **WolvenKit** interface, it's time to dive into **Cyberpunk 2077**'s game files and learn how to navigate its data structure. This step is crucial for identifying the 3D models, textures, and other assets you want to modify or export to Blender.

Accessing Cyberpunk 2077 Files in WolvenKit

1. Opening Game Files:

- To access **Cyberpunk 2077** files, ensure the game path is correctly configured in **WolvenKit**. If this isn't set up yet, refer to the instructions in **Part 2: Installing WolvenKit**.
- In the **navigation bar** on the left, you'll see Cyberpunk's folder structure, containing all game files organized by category: models, textures, sounds, animations, and more.

2. Exploring the File Structure:

- Cyberpunk 2077 uses a complex file structure to organize its assets. You'll find folders such as **base**, **characters**, **vehicles**, **environment**, and so on.
- Familiarizing yourself with these folders will help you locate the types of files you're looking for. For example, main character models are often located in **base\characters\main_npc**, while vehicles are found in **base\vehicles**.

For more details on Cyberpunk 2077's file structure, check out the **Red Modding Wiki page on folder structure**, which explains the folder hierarchy and naming conventions.

Finding and Previewing 3D Models, Textures, and Other Assets

1. Searching for a Specific 3D Model:

- Use the search bar at the top of the navigation panel to type keywords related to the asset you're looking for (e.g., "V's Apartment" to find assets related to the main character's apartment).
- **3D models** are often stored in **.mesh** or **.app** format. When you select a **.mesh** file, WolvenKit will attempt to display a preview in the central panel, allowing you to verify the asset.

2. Previewing Textures and Materials:

- Textures are typically in **.dds** (DirectDraw Surface) format. When you select a texture file, it will display in the **preview panel**.

- You can check the resolution and quality of each texture before deciding to export. For materials linked to models, WolvenKit also allows you to view details like layers and shader settings.

3. Exploring Animations and Sounds:

- Animations, though sometimes limited in preview, are available in **.anim** or **.animgraph** formats, letting you view motion sequences.
- Game sounds are often in **.wem** or **.bnk** format. WolvenKit allows you to play certain sounds, helpful for confirming the correct audio file.

Using Filters to Find Specific Files

1. File Categories:

- WolvenKit offers filters to narrow down displayed files. For example, selecting **Textures** or **Meshes** in filters will show only those files in the navigation panel.
- This can be particularly useful to save time when searching for a specific file type (such as a model or texture).

2. Using Tags and Naming Conventions:

- Cyberpunk 2077 uses specific naming conventions (e.g., “base_veh” for base vehicles). Understanding these conventions enables more targeted searches.
- The **Red Modding Wiki page on folder structure** provides more information on naming conventions and specific game files.

Tips for Organizing Your Exploration- Create a Project in WolvenKit to organize assets you intend to export. This makes it easier to keep track of modifications for each file.

- **Use notes or a journal** to record file paths and asset names you want to export.

To create a **custom search filter in TweakDB using WolvenKit**, follow these steps, utilizing various search options to precisely target game assets and data.

Steps to Set Up a Search Filter in TweakDB with WolvenKit

1. Open TweakDB and Access the Search Bar:

- Launch WolvenKit and open the **TweakDB Browser**. The search bar at the top of the interface lets you filter specific data, such as weapons, vehicles, or NPC assets.

2. Use Search Operators for Targeted Results:

- WolvenKit supports several operators for fine-tuning searches:
 - **File Type:** To search for specific file types, add the file extension in your search, e.g., **.mesh** for 3D models or **.xbm** for textures.

- **Full or Partial Path:** Specify a full or partial file path to limit results to a specific folder, which is helpful for targeting assets within Cyberpunk 2077's base directory.
- **Excluding Terms:** Add ! before a term to exclude it, such as !morphtarget to avoid files containing that term.
- **Regular Expressions:** For advanced searches, regular expressions (regex:) allow you to specify complex search patterns, ideal for pinpointing elements with certain characters or long names.

3. Save and Reuse the Filter:

- After configuring search criteria, save the filter for future use by selecting **Save Filter** in TweakDB Browser.

5. Exporting 3D Models from WolvenKit

Now that you've explored **Cyberpunk 2077** files in **WolvenKit**, the next step is to **export 3D models** (such as characters, vehicles, or objects) for use in **Blender**. This section explains how to prepare a model for export and address common issues.

Preparing Files for Export

1. Select the Model to Export:

- In the **navigation panel** of WolvenKit, locate the 3D model you want to export, typically found in **.mesh** or **.app** format.
- Click on the file to display a preview in the central panel, allowing you to verify it's the correct asset.

2. Check Model Settings:

- Use the **Properties panel** (on the right) to adjust the model's details if necessary. For instance, you can disable certain materials or configurations that may cause issues in Blender.

3. Choose the Export Format:

- For compatibility with Blender, it's recommended to export models as **FBX** or **OBJ**, preserving geometry, basic textures, and material information.
- In some cases, exporting as **GLTF** may be preferred if the model contains specific data better handled by Blender with that format.

Export Procedure

1. Initiate the Export:

- Select the model and right-click to access export options, then choose **Export > Export as FBX** (or another compatible format).
- A window will appear to select the export destination folder. Choose an accessible location, like a work folder on your desktop.

2. Check Export Options:

- Before confirming, adjust the export settings. For example, configure texture resolution or model scaling. This step is essential for proper model display in Blender.
- Click **OK** to start the export. Depending on the model's complexity, this may take a few minutes.

3. Resolving Export Errors:

- If an error occurs, check the model settings in WolvenKit, like materials or attached animations, and retry with specific options disabled.
- The [WolvenKit GitHub page](#) and [Red Modding Wiki](#) offer solutions to common export issues.

Common Export Issues and Solutions

1. Missing or Pink Textures:

- Sometimes textures appear pink in Blender, indicating they weren't found during import. To fix this, ensure texture files are included in the export folder.
- If necessary, export each texture individually from WolvenKit, then manually apply them in Blender.

2. Incorrect Model Scaling:

- Some models may appear too large or too small in Blender. Adjust the scale in WolvenKit's export settings or within Blender after import.

3. Misaligned Materials:

- If materials don't display correctly, check that shader settings and textures are properly configured in Blender. Blender may require adjustments to match WolvenKit settings.

6. Importing Models into Blender

With your 3D models exported from **WolvenKit**, the next step is to **import them into Blender** for further editing. This section walks you through the import process and helps you adjust settings to display models correctly in Blender.

Opening Blender and Initial Setup

1. Open Blender:

- Launch Blender on your computer. Make sure to use the latest version for better compatibility with formats like **FBX** and **GLTF**.
- To check your version, go to **Help > About Blender**.

2. Prepare a New Project:

- Start a new project by going to **File > New > General** for a clean scene.
- Delete the default cube (select it and press **X**, then **Delete**).

Importing a 3D Model Exported from WolvenKit

1. Import the File:

- Go to **File > Import** and select the file format you used for export in WolvenKit (e.g., **FBX** or **OBJ**).
- Navigate to your destination folder and select the exported model.

2. Adjust Import Settings:

- In the left panel of the import window, check the following options to ensure proper display:
 - **Scale:** Adjust to 1.0 if needed to avoid the model being too large or too small.
 - **Import Cameras** and **Import Lights:** Uncheck these if you only need the model.
- Click **Import FBX** to finalize the import into Blender.

3. Verify the Import:

- After importing, check that the model appears in Blender's 3D viewport. If it's not properly positioned, use the translate (G), rotate (R), and scale (S) tools to adjust it within the scene.

Basic Adjustments for Correct Model Display

1. Fix Missing Textures:

- If the model appears pink, Blender couldn't locate the textures. In the **Shader Editor** (Shading tab), manually connect the exported textures using **Image Texture** nodes.
- You can import each texture file individually and link them to the model's materials to correct this.

2. Verify Scale and Orientation:

- Blender may display a different scale and orientation than WolvenKit. Use the transformation tools to adjust the model's size, rotation, and position.
- For models that are upside down or misaligned, apply a 180° rotation on the **Z-axis**.

3. Apply Transforms:

- Once the model is properly adjusted, select it and press **Ctrl + A** then choose **All Transforms**. This sets the current position as the base, preventing issues during future manipulations.

7. Setting Up Materials and Textures in Blender

After importing your 3D models into **Blender**, configuring materials and textures correctly is essential to achieve the desired look, matching **Cyberpunk 2077**'s aesthetics. This step covers applying textures exported from **WolvenKit** and adjusting shaders in Blender.

Retrieving Textures Exported from WolvenKit

1. Verify Texture Files:

- During the WolvenKit export, ensure all textures associated with the model were exported, such as **diffuse (albedo)**, **normal maps**, **roughness**, and **metallic**.
- Place all textures in the same folder as your Blender file to streamline the import process.

2. Import Textures into Blender:

- Go to Blender's **Shading** tab to access the **Shader Editor**. Select your model in the 3D view.
- In the Shader Editor, you'll see a default **Principled BSDF** node. This shader is ideal for recreating physically based materials.

3. Add Texture Nodes:

- For each texture (diffuse, normal, roughness, metallic), add an **Image Texture node** in the Shader Editor.

- Click **Open** on each Image Texture node, and select the corresponding texture file from your texture folder.

Applying Materials and Textures in Blender

1. Set Up the Diffuse Texture:

- Connect the **diffuse (albedo)** texture to the **Base Color** input of the Principled BSDF node. This applies the base color to the model.

2. Add the Normal Map:

- Place a **Normal Map node** between the Image Texture node of the normal map and the **Normal** input of the Principled BSDF node.
- This adds surface detail, simulating depth and small irregularities on the model.

3. Apply Roughness and Metallic Textures:

- For metallic materials, connect the **roughness texture** to the **Roughness** input and the **metallic texture** to the **Metallic** input of the Principled BSDF node. This controls the model's reflection and shine, matching the original in-game materials.

Adjusting Shaders for Optimal Rendering

1. Use an HDRI Environment (optional):

- For a realistic render, use an **HDRI** in Blender's **World Settings**. This adds natural environment lighting and reflections.
- Sites like **Poly Haven** offer free HDRIs, perfect for 3D rendering. Poly Haven HDRI.

2. Adjust Texture Intensity:

- Imported textures may appear too strong or weak. Use **ColorRamp** or **Math** nodes to adjust intensity (especially for roughness and metallic textures) as needed.

3. Optimizing Materials for Cycles and Eevee:

- Blender has two rendering engines: **Cycles** for realistic rendering and **Eevee** for faster rendering. For realism, enable **Cycles** in the Render Settings, or for better performance, **Eevee** can be adjusted to approximate material appearance.

Tips for Better Integration

- **Group Textures:** For complex models, group texture nodes in the Shader Editor to organize materials easily.
- **Use Shader Libraries:** Blender allows you to save shaders for reuse on similar models.

8. Model Optimization and Cleanup in Blender

After applying materials and textures, **optimizing and cleaning up the model** in **Blender** is recommended. This step improves performance and makes future modifications easier, especially if you plan to integrate the model into other projects. This section will guide you through reducing geometry, removing unnecessary elements, and adjusting model settings.

Cleaning Up Unnecessary Geometry

1. **Switch to Edit Mode:**

- Select the model and press **Tab** to enter **Edit Mode**. This allows you to view and modify the model's geometry structure.

2. **Remove Duplicates:**

- Use the **Merge by Distance** tool to remove duplicate vertices, reducing model complexity without changing its appearance.
- Access this tool by selecting all vertices (**A**) then **M > Merge by Distance**. Adjust the distance if needed for sensitivity.

3. **Simplify Topology:**

- Imported models may have excessive detail. Use tools like **Dissolve Edges** or **Limited Dissolve** to reduce face count without compromising visual quality.
- Access **Dissolve Edges** by selecting unnecessary edges, pressing **X**, then choosing **Dissolve Edges**.

Reducing Polygons if Necessary

1. **Use the Decimate Modifier:**

- The **Decimate Modifier** is helpful for reducing a model's polygon count while retaining its shape. In the **Modifiers panel**, add a **Decimate Modifier**.
- Adjust the **Ratio** to control polygon reduction. A ratio of 0.5 cuts the model's faces in half. Preview changes to avoid losing too much detail.

2. **Optimize Based on Final Use:**

- If the model will be used in a real-time game engine like Unity or Unreal, a significant reduction in polygons is often necessary. For static renders, minimal reduction may be needed to retain details.

Adjustments to Improve Performance

1. Use Optimized Textures:

- Ensure that textures are optimized. Resize them to suit your scene's needs (e.g., 1024x1024 for secondary objects, 2048x2048 for main elements).
- Sites like [TinyPNG](#) can help compress files without quality loss.

2. Clean Up Materials:

- Remove unused materials in the **Material Properties** panel to streamline the model. Blender sometimes retains unused materials, which can bloat the file.

3. Apply Modifiers:

- Once optimized, apply modifiers like **Decimate** or **Subdivision Surface** to finalize adjustments. This ensures that changes are permanent and simplifies the file structure.

Recommended Tools and Plugins for Optimization in Blender

1. Mesh Cleanup Add-on:

- Blender has a **Mesh Cleanup** add-on to automate some cleanup tasks. Enable it in **Edit > Preferences > Add-ons**, then search for "Mesh Cleanup".

2. UV Packmaster Add-on (optional):

- If working with complex UV textures, the **UV Packmaster** add-on can help optimize UV layouts for better texture distribution. [UV Packmaster Link](#).

9. Final Export and Use of the 3D Model

With your model optimized and ready, it's time for the **final export from Blender**. This is a crucial step if you plan to integrate your model into a game engine like Unity or Unreal Engine or use it in other 3D software. This guide covers how to set up the export for compatibility and organize files for efficient use.

Exporting the Optimized Model from Blender

1. Select the Objects:

- Before exporting, make sure to select all objects you want to include, generally the main model and all associated elements (textures, materials, etc.).

2. Access Export Options:

- Go to **File > Export** and choose the appropriate format for your use (typically **FBX** or **GLTF** for game engines).

- **FBX** is recommended for Unity and Unreal Engine due to its support for animations and materials.
- **GLTF** is also versatile, particularly for web-based renders or compatible game engines.

3. **Configure Export Settings:**

- **Scale:** Ensure the scale is set correctly, usually at 1.0, to avoid size issues in the target software.
- **Include:** Check **Selected Objects** to export only selected items, and **Apply Transform** to retain transformations.
- **Materials:** Ensure materials are included. For formats like FBX, Blender will incorporate basic materials.

4. **Export Textures Separately** (if necessary):

- If exporting to software requiring separate textures, make sure to copy texture files to the same folder as your exported file.
- Organize by type (diffuse, normal, roughness) to ease re-application in a different environment.

Using the 3D Model in Other Software

1. **Importing into a Game Engine** (Unity/Unreal Engine):

- In **Unity**, go to **Assets > Import New Asset** to load the FBX or GLTF model, and verify material and scale settings in the Inspector.
- In **Unreal Engine**, use the **Content Browser** to import the model, and adjust texture and shader settings as needed.

2. **Check for Compatibility:**

- Ensure the model is properly oriented and textures display as expected. Adjust shaders and material settings if any compatibility issues arise.
- In game engines, set up collisions, physics, and other relevant properties as required.

3. **Using for Renders or Animations:**

- If using the model for renders or animations, set up lighting and camera in Blender for optimal results.
- Use **Cycles** for realistic renders or **Eevee** for quick renders with high visual quality.

Organizing Files for Efficient Integration

- **Create a Folder Structure:**
 - Organize your project by creating subfolders for **Models**, **Textures**, **Shaders**, and **Renders**. This makes integration and model transfer between projects or users more manageable.
- **Document Settings:**
 - For complex projects, create simple documentation listing essential material and texture settings. This is useful if you or others need to re-integrate the model later.

10. Additional Tips and Resources

This final section gathers practical tips to deepen your use of **WolvenKit** and **Blender** for **Cyberpunk 2077** modding. It also summarizes the key steps in the modding process and offers resources for further exploration.

Practical Tips for Going Further

1. **Experiment with WolvenKit:**
 - Once you're comfortable with the basics, explore WolvenKit's advanced features, such as modifying animations, adjusting scripts, or creating new assets. The [Red Modding Wiki](#) provides useful guides and examples.
2. **Enhance Textures and Materials in Blender:**
 - For realistic renders, experiment with **advanced nodes** in Blender, like **Bump** and **Displacement** nodes. These techniques add depth and realism to surfaces.
 - For advanced shader and material effects, consider platforms like [BlackRavenVR's Patreon](#) or YouTube tutorials.
3. **Integration into Game Engines:**
 - If you're working with Unity or Unreal Engine, explore each engine's specifics to make the most of imported models. Unity offers numerous plugins to enhance 3D model rendering, while Unreal Engine is particularly suited for advanced shaders.

Essential Resources

- [Red Modding Wiki](#): A comprehensive wiki with tutorials and documentation on Cyberpunk 2077 modding.
- [Blender Guru](#): A popular site for learning advanced modeling, rendering, and texturing techniques in Blender.
- [Poly Haven](#): A resource for high-quality HDRI and PBR textures, ideal for realistic rendering in Blender.