Batch Render

Release 1.00

Simon Sanchez

Nov 27, 2020

CONTENTS:

BlenderMarket | Gumroad | Artstation | Youtube

Batch Render is a *command line tool* that makes use of **Blender** to quickly render multiple objects from multiple angles.

Features:

- **Recursive rendering**, renders all objects from current folder and it's subfolders.
- Support for **.obj** and **.fbx**
- Automatic material creation and assignment
  - **Albedo** support
  - **Opacity** support
  - **Emissive** support
  - **Metallic** support
  - **Roughness** support
  - **Normal Map** support
  - **Height** support
- Automatically **fits the camera to the object**
- Easy **selection and creation of rendering presets**
- Easy **turntable generation** with custom amount of frames
- Render engine defined by render preset (supports **Eevee**, **Cycles** and **Workbench**)

Example of turntable rendered with Batch Render

CHAPTER

ONE

INSTALLATION

This section details instructions to install **Batch Render** in your computer, the tutorial is focused on *Windows* devices, but the steps are pretty much the same for *MacOs/Linux*.

Having **Blender** installed is a pre-requesite.

1. **Download** Python
    1. When installing **Python**, make sure to check the **'Add Python X.x to PATH'**
2. After downloading **Batch Render**, you can place the folder anywhere you want
3. On *Windows*, search for **'Edit environment variables for your account'**
4. Click on **'Environment Variables…'**
5. Double click on the variable **'Path'** either on *User variables* or *System variables*
6. Click **'New'** and paste the *address* of **Batch Render's** folder
7. Click **'New'** again and paste the *address* of **Blender's** installation folder
8. **Confirm** all changes

To add a variable to **PATH** in *Linux and MacOs* run the following command from the terminal

```
export PATH=$PATH:/direction/to/folder
```

You might need to restart your computer for the installation to take effect.

1.1 Why do we need to do this?

Adding **Blender** and **Batch Call** to **PATH** means that you can call them from the terminal without including the full path to the file, it's much more convenient.

To check if the installation was successful open cmd/terminal:

- **Run** `Blender`, if **Blender** launches, it's correct
- **Run** `batchCall.py`, if **preset selection** shows up, it's correct

CHAPTER

TWO

USAGE

This section explains how to actually use **Batch Render**.

Batch Render is a *command line tool*, meaning that it's completely operated via **cmd/terminal**.

2.1 Basic Rendering

To start rendering you must launch your terminal and navigate to the folder where the **.obj** and **.fbx** files that you want to render are located.

On *Windows*, you can open the folder from the explorer and in the address of the folder you can type '**cmd**' and launch the terminal directly in that folder.

Once you're in that folder from within the terminal you can run the following command:

```
batchCall.py
```

This will **prompt you** to select a **preset** for rendering. Each preset has a *number associated*, you must enter that number and press enter to use that preset for rendering.

Batch Render is recursive, meaning that *it will render all files in the current folder and in all the subfolders*.

2.1.1 What are presets?

Presets are simply .blend files that you can customize to your liking, the renderer will use the *selected preset* to render with. These presets can include HDR maps, lighting setups, etc…

The render engine and the render properties that Batch Render will use depend on the preset and must be configured from withing **Blender** itself.

Presets are located in the **RenderPresets** folder, *this folder must not be deleted*.

2.1.2 Settings

Once you select a preset, you will be able to configure the settings you want for the current batch.

- **Render turntable**: If you say yes to this setting, the renderer will render a turntable for each object
  - **Turntable frames**: This settings indicates how many frames will make up the turntable
- **Render 42 angles**: This setting won't show up when rendering a turntable, the setting indicates if 42 angles per object must be rendered, if not, 12 angles will be rendered
- **Render width and height**: Setting determines the dimensions of the render in pixels

- **Camera push in X**: Offsep of the camera in the X axis, it's suggested to use 0 unless you're rendering a turntable
- **Camera push in Z**: Offset of the camera in the Z axis, it's not suggested for it to be 0, as part of the rendered object might end up outside of the camera's vision
- **Displacement intensity**: Defines the strength of the displacement map (height map), this setting is only relevant if the preset selected uses Cycles as it's render engine, and if a height map is present
- **Emission intensity**: Defines the strength of the emissive map, only relevant if an emissive map is present
- **Texture tile**: Defines how many times all maps will be tiled in the X, Y and Z axis, options is mostly relevant when rendering UV Spheres with materials to avoid stretching

2.1.3 How to cancel?

If you need to interrupt the program, you can press `Ctrl+C` or `Command+.` inside the terminal. You might need to do it multiple times until the execution stops.

2.1.4 Material Matching

Batch Render will automatically try to *assign the material maps to each material*. However, some **requirements** must be met.

Each **.obj** or **.fbx** should have a folder named '*textures*', inside this folder all the different maps for each material in the object should be located.

The **material matching depends on the name of each material**, for example, let's suppose we have an object with two materials:

1. **Red**
2. **Green**

Inside the '*textures*' folder we have the following maps:

- object_red_BaseColor.png
- object_red_Metallic.png
- object_red_Roughness.png
- object_red_Normal.png

- object_green_BaseColor.png
- object_green_Metallic.png
- object_green_Roughness.png
- object_green_Normal.png

In this example, all the maps that contains the word '**red**' will be assigned to the **Red** material, same goes for '**green**'. *Casing in this case doesn't matter*, if the material is named **Red.001** it will also be properly matched.

Each type of map will be assigned to the material depending on their names, here are the terms for each map:

- **Base Color**: *diffuse, diff, albedo, base, col, color, basecolor*
- **Metallic**: *metallic, metalness, metal, mtl*

- **Roughness**: *roughness, rough, rgh*
- **Normal Map**: *normal, nor, nrm, nrml, norm*
- **Height**: *height, displacement, displace, disp, dsp, heightmap*
- **Emissive**: *emissive, emission, emiss*

If any of these terms are **part of** the name of a map, it will be used as such, *the terms are case insensitive*.

A tool called batchRename is included to quickly suffix and posfix text to files names for proper matching, this tool is discussed in it's own section.

Turntable Generation

The **turntable option** allows to quickly create a turntable for the objects, the **interpolation is linear**, meaning that *it will loop seamlessly*.

You can **define how many frames you want for the turntable**. Each frame will be individually rendered as a **.png** image inside the '**turntable**' folder which will be inside the folder with the name of the preset chosen to render.

Pushing the camera in the X and Z axis is recommended, as the camera's default position is in the center of the object/scene.

These frames **can be converted into a video** with many video editing software, as well as within **Blender**.

Preset Creation

The creation of rendering presets is **extremely simple**, you just need to open **Blender**, create your own render setup (with lights, hdr, emissive planes, etc…) and save the **.blend** file inside the **RenderPresets** folder.

The newly create preset will appear in the preset selection list, ready to be used.

Batch Render will use the settings from the **.blend** file to render, except some that are overriden (*such as the width and height of the render.*) These settings include things such as the render samples, the transparent background, the render engine to be used, and so on.

CHAPTER

THREE

BATCH RENAME

Batch Rename is a tool included with **Batch Render** that allows you to *suffix and posfix* to the name of all the files in the current folder **from the terminal**.

This can be useful in case you want to add the name of a material to all the maps in the current folder so the material is matched properly by **Batch Render**. And of course, it can also be used for other stuff outside *Batch Render*.

To use this tool, from the terminal, you must navigate to the folder where the files you want to modify are located, beware that **this will modify the name of all the files in that folder**, you might want to isolate them in temporary subfolders.

3.1 Arguments

- **-s**: *Adds* a **suffix** to all the files
- **-rs**: *Removes* a **suffix** to all the files
- **-p**: *Adds* a **posfix** to all the files
- **-rp**: *Removes* a **posfix** to all the files

For this example, let's suppose we have 3 files in the current folder:

- baseColor.png
- roughness.png
- height.png

Running the command `batchRename.py -s red_` *(insert suffix red_)* will leave us with:

- red_baseColor.png
- red_roughness.png
- red_height.png

Running the command `batchRename.py -p _myObject` *(insert posfix _myObject)* will leave us with:

- red_baseColor_myObject.png
- red_roughness_myObject.png
- red_height_myObject.png

Running the command `batchRename.py -rs red_` *(removes suffix red_)* will leave us with:

- baseColor_myObject.png
- roughness_myObject.png

- height_myObject.png

Running the command `batchRename.py -rp _myObject` *(removes posfix _myObject)* will leave us with:

- baseColor.png
- roughness.png
- height.png

CHAPTER FOUR

CONTACT

For **questions**, **bug reports** or **feature suggestions** please contact me through mail:

simonsanchez.art@gmail.com