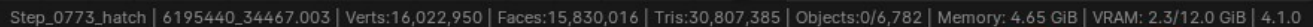


How To Build LEGO Models in Blender

Here is a summary of how I build LEGO kits in blender. I've built several LEGO kits now, the most recent being the huge Star Wars Millennium Falcon (755192). This tutorial will use that Millennium Falcon kit as an example.

In this kit, there are 711 unique parts. You can use the ldraw plug-in in blender to individually load these, but it will take a *long* time and you will need to apply materials to each part. Using a csv database of parts, you can use two python scripts to automatically load the parts and assign materials.

This kit has over 7500 pieces. When I started out, I was worried about running out of memory, so I decided to eliminate the connector pieces. In the end, it didn't use as much memory as I thought, but I'm not going back to put the connectors in. As-built, this model resulted in about 5GB of system RAM and 2.3GB of VRAM.



Step_0773_hatch | 6195440_34467.003 | Verts:16,022,950 | Faces:15,830,016 | Tris:30,807,385 | Objects:0/6,782 | Memory: 4.65 GiB | VRAM: 2.3/12.0 GiB | 4.1.0

Assumptions

- Blender knowledge

I assume you know to use Blender. This will not be a tutorial on how to use Blender. There are a lot of good tutorials online. I'd say this workflow would be aimed for an intermediate-level user.

- Spreadsheet literate

You need to be able to use spreadsheet software. I'm using Linux, so my example will use LibreOffice. You could also use Excel or Google Sheets. We will only be opening and saving csv files here.

- (Very minimal) python experience

You will need to modify several lines in Blender python. This does not require any programming experience.

- Hardware

Obviously, the larger the model, the more system and GPU memory are going to be required. I built the Millennium Falcon on an i7 13th gen, 32GB RAM with an NVIDIA RTX4070 GPU (12GB VRAM), but I have built smaller models with my i5+16GB RAM without a GPU (using Eevee for rendering).



Kubuntu 22.04

<https://www.kubuntu.org>

Software

KDE Plasma Version: 5.24.7

KDE Frameworks Version: 5.92.0

Qt Version: 5.15.3

Kernel Version: 6.5.0-44-generic (64-bit)

Graphics Platform: X11

Hardware

Processors: 24 × 13th Gen Intel® Core™ i7-13700KF

Memory: 31.1 GiB of RAM

Graphics Processor: NVIDIA GeForce RTX 4070/PCIe/SSE2

Workflow

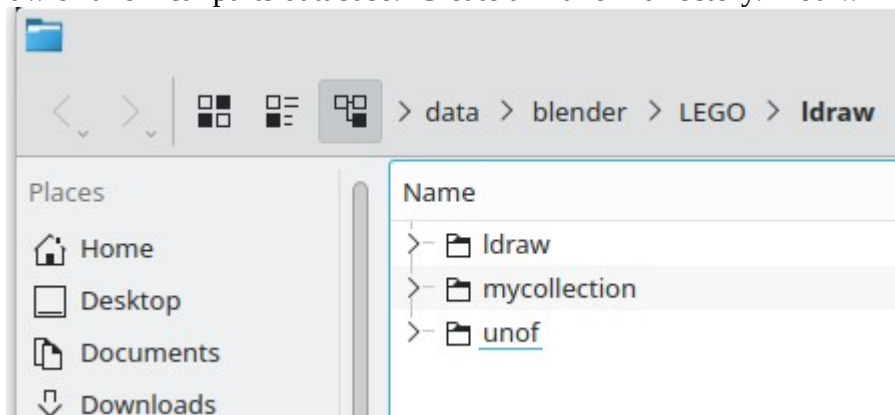
Here is the main workflow. Refer back to this section for the rest of the document.

1. Prepare your system
2. Select LEGO kit and download instructions.
3. Download CSV parts list
4. Verify there is a model for each part. Download missing parts.
5. Import the parts into Blender.
6. Build the kit in Blender.

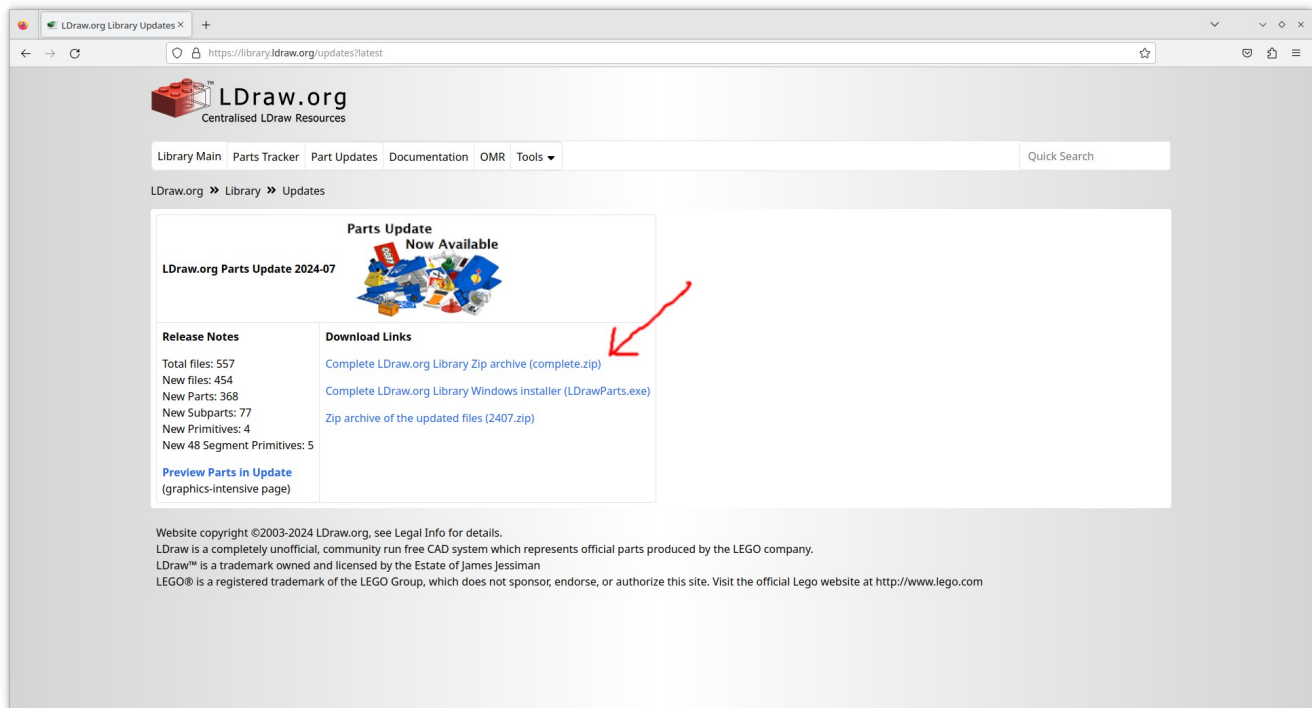
1. Prepare Your System

- Linux, Windows?, or Mac? This workflow should work for all systems. I used Linux.
- Blender
I am using version 4.1
- Spreadsheet Editor
I am using LibreOffice, but you can use Google Sheets or Excel. We will be opening and saving CSV format files.

Create a directory structure that looks like this. There must be three directories with the directory structure as follows. Put the ldraw parts in the “ldraw” directory. Create an *empty* directory called “mycollection”. Here is where you put any additional parts that you have to download that aren’t in the ldraw or unof parts database. Create an “unof” directory. You will fill that below.



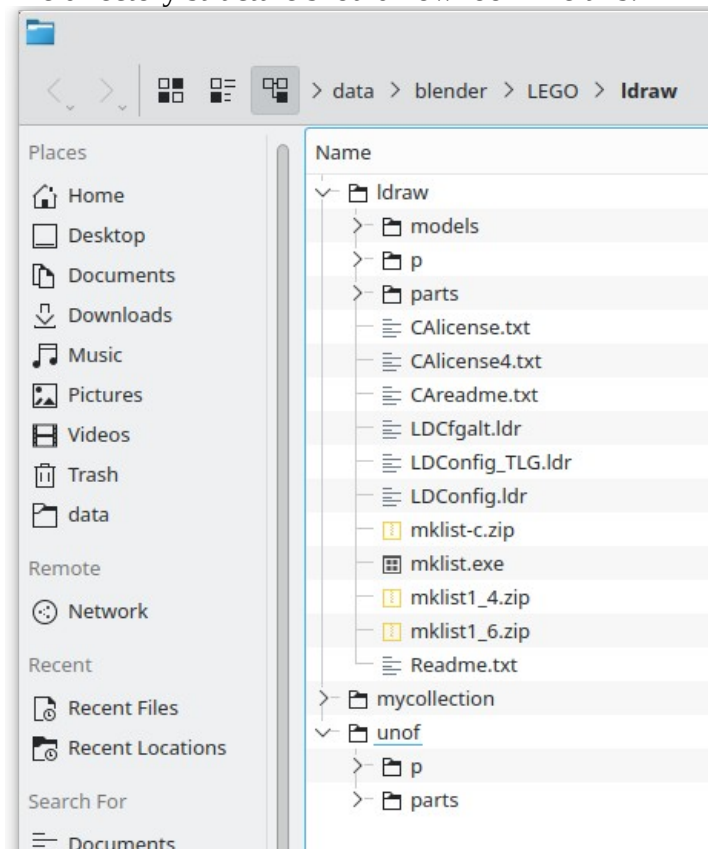
- Download LDraw database and **unzip** this into the “ldraw” directory.
<https://library.ldraw.org/updates?latest>



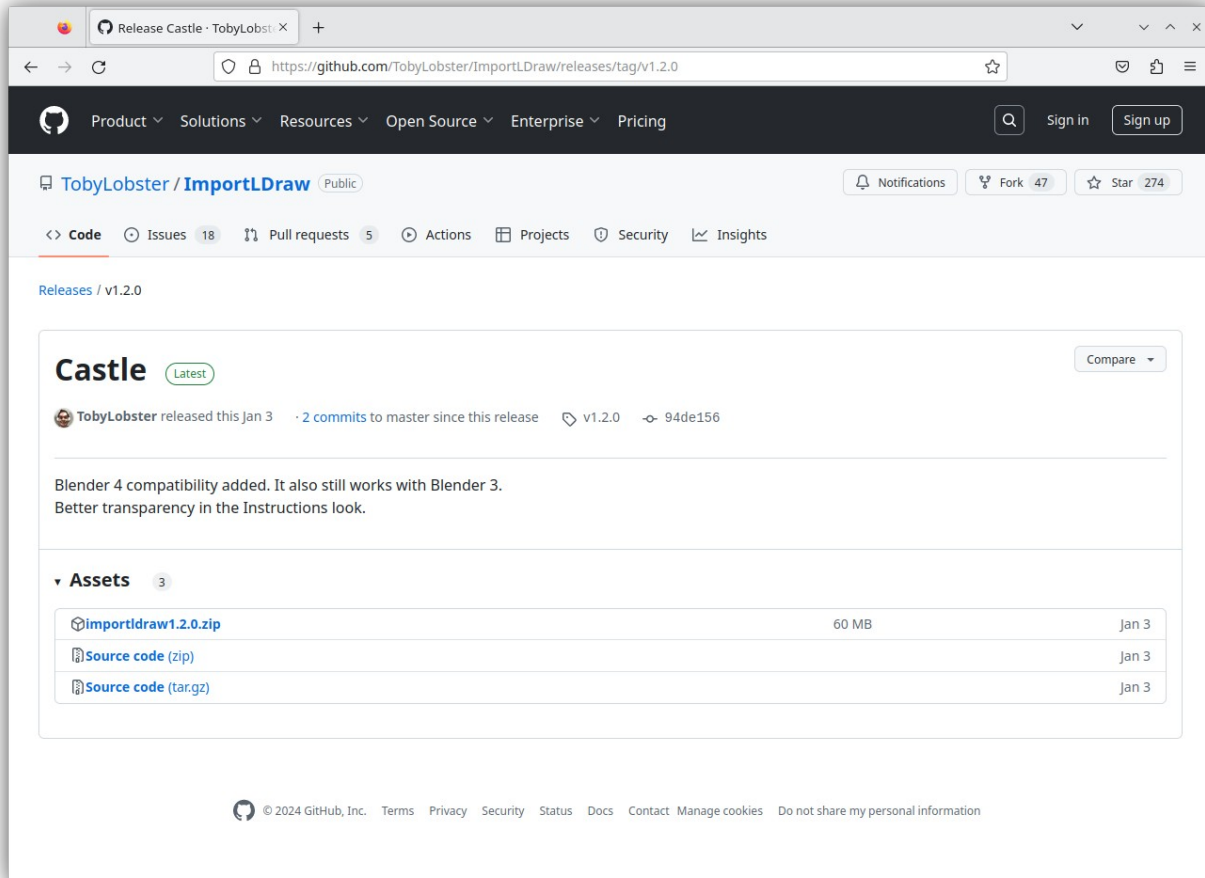
Download the “Complete Lddraw.org Library Zip archive”

- Download unofficial parts database and **unzip** the parts into the “unof” directory
<https://library.ldraw.org/library/unofficial/ldrawunf.zip>

The directory structure should now look like this:

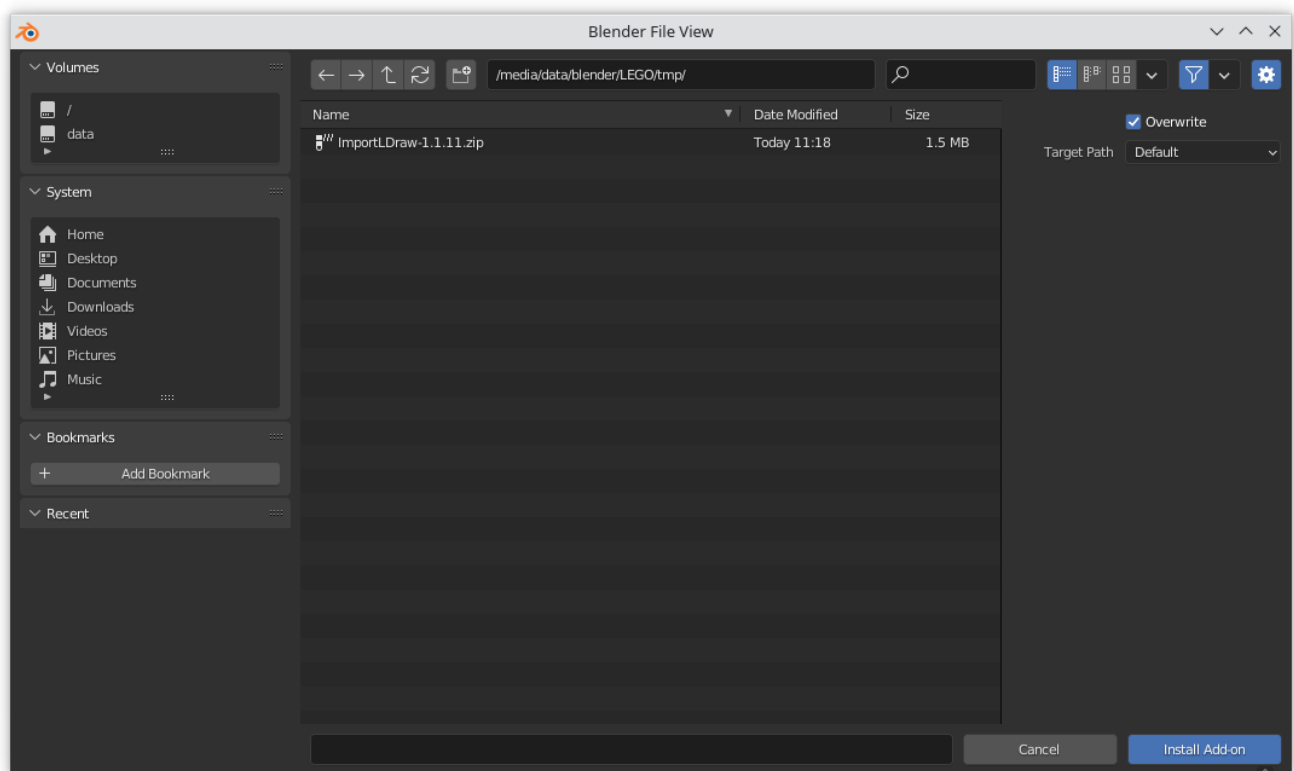


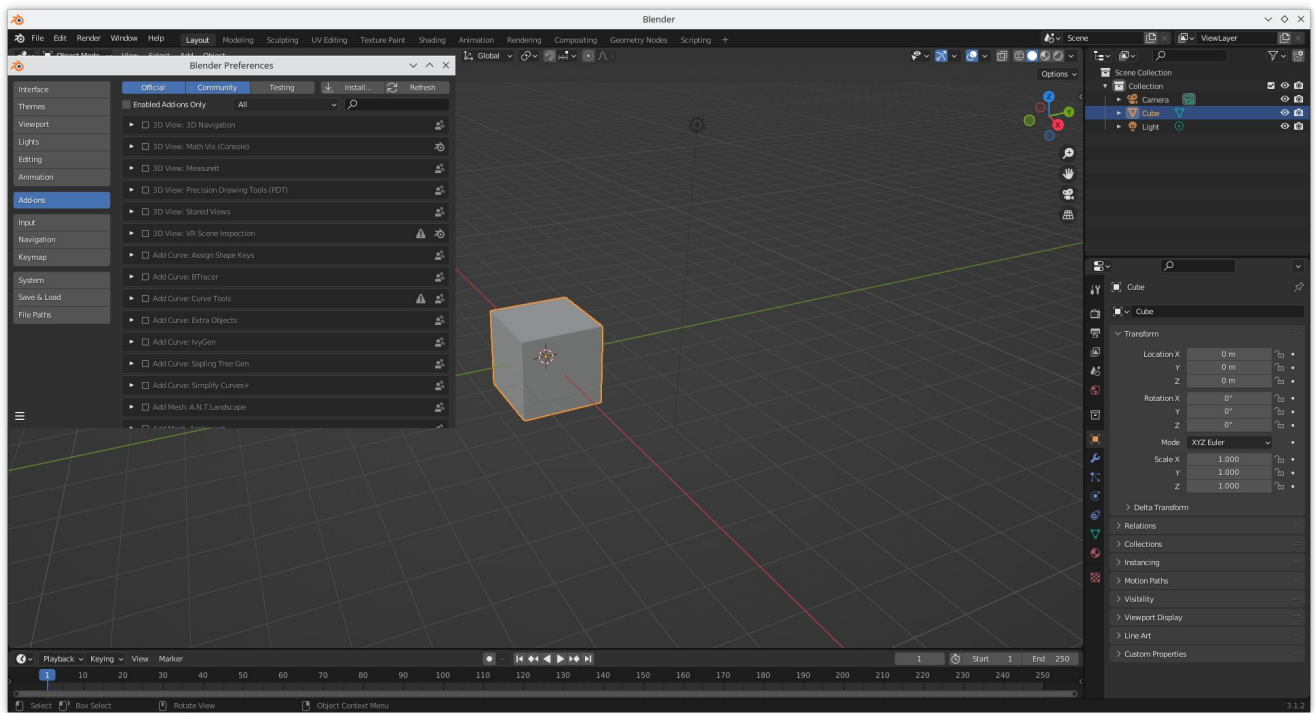
- Download/install ImportLDraw add-in into Blender
<https://github.com/TobyLobster/ImportLDraw/releases>
For Blender 4.1, I am using the “Castle” release.

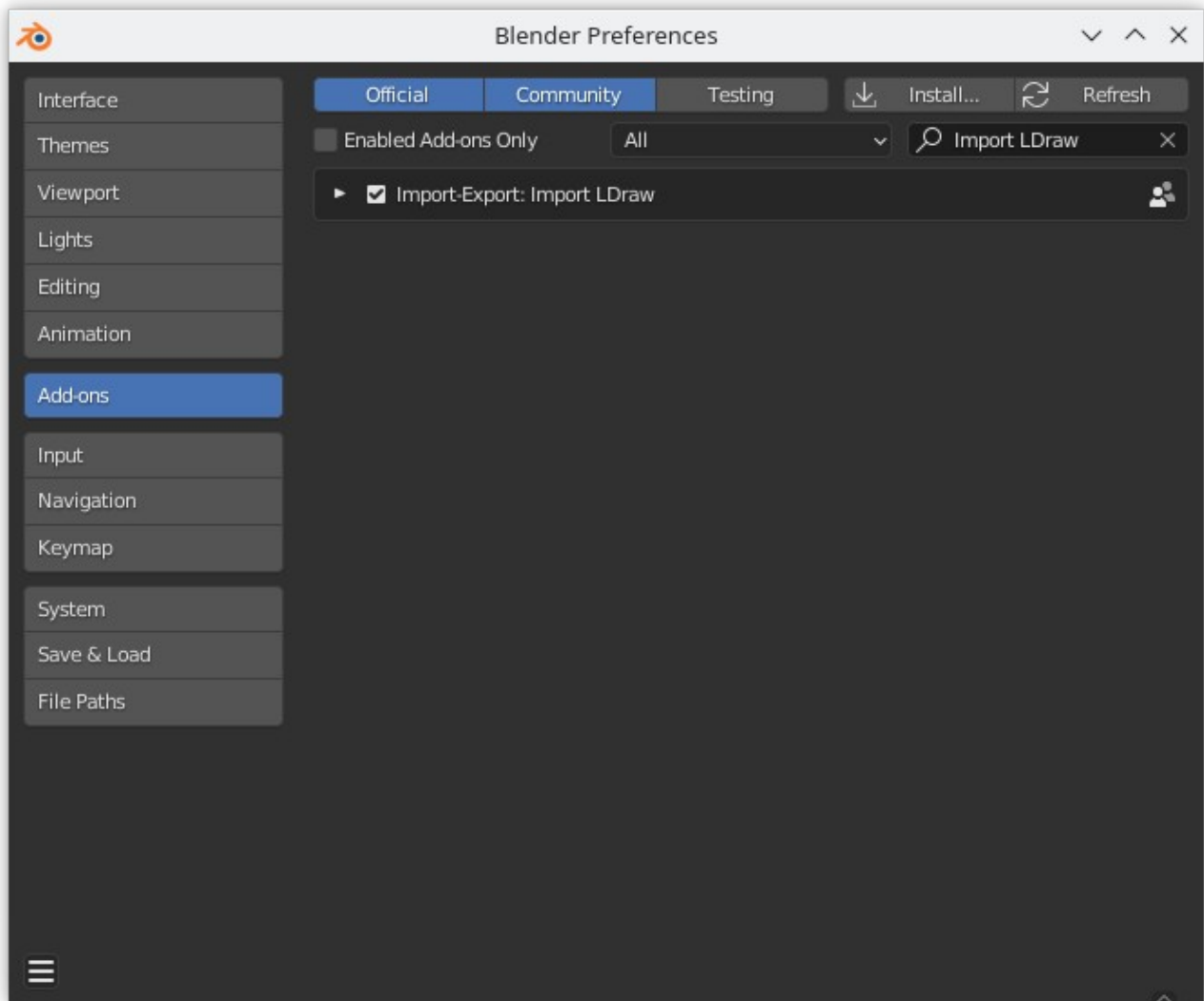


At

this point, I assume you know to add add-ins to Blender. Open Blender and install the Importldraw add-in directly from the zip file. Enable it.

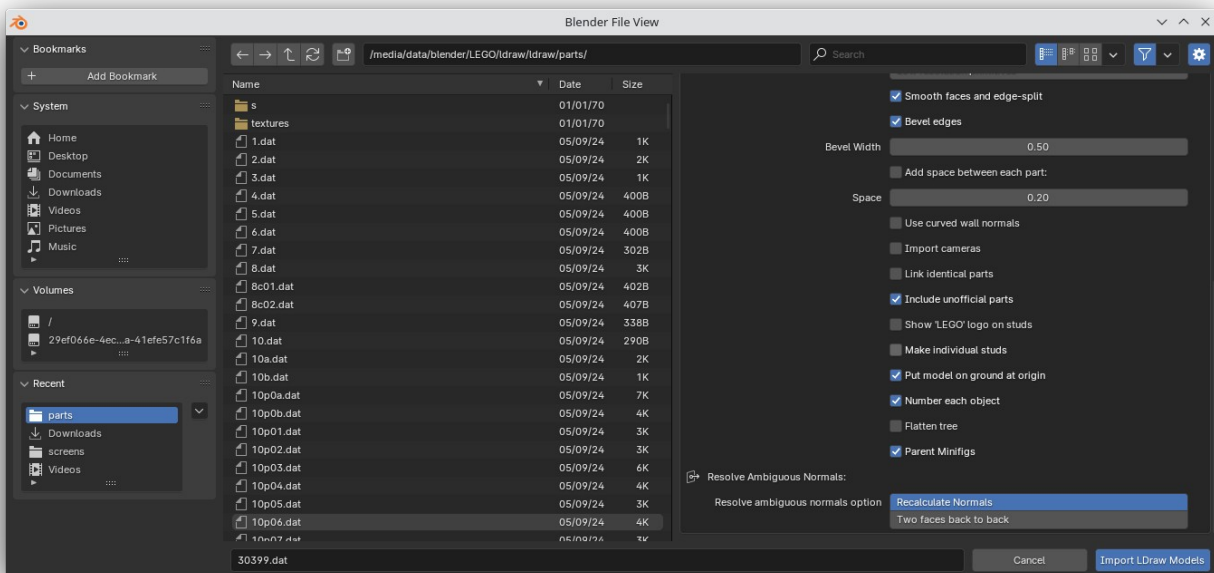
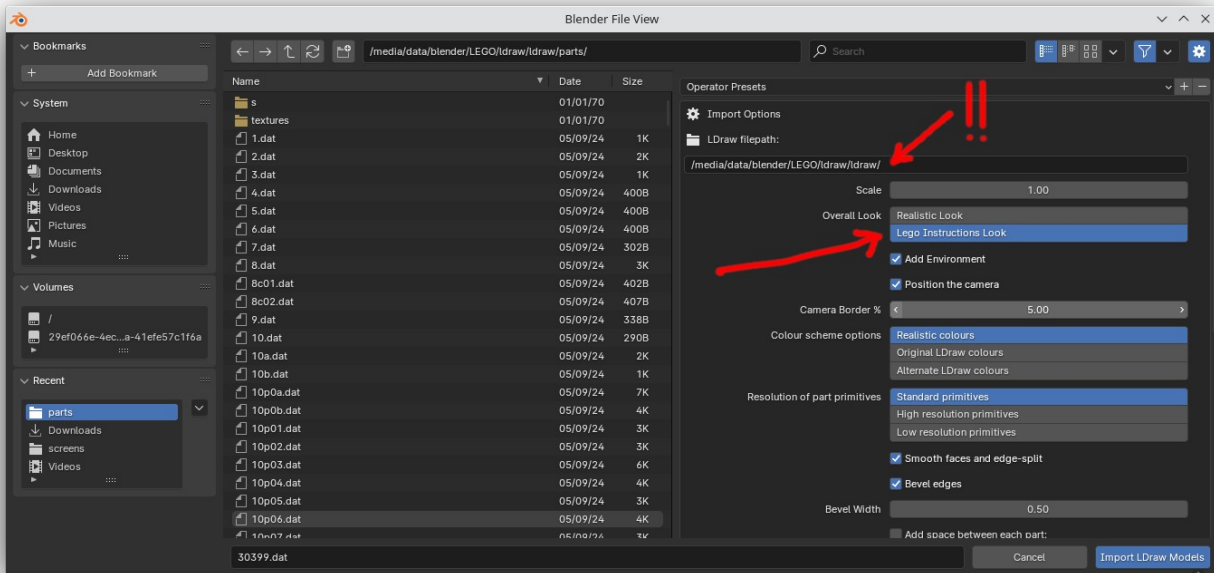






Test the Importldraw add-in. Open a new blender file.

- Delete the default cube.
- Click Import->LDraw.
- In the pop-up menu, navigate to your ldraw parts directory and select a random part.
- Check/uncheck the options as per the images below.
- Add the full pathname of your LDraw database to the form.
- Be sure to set the look to “Lego Instructions Look”



Lastly, download the following files from <https://github.com/Burl26/buildLEGOInBlender>

read_kit_v2.blend

xref_csv.py

Burl26/buildLEGOInBlender

https://github.com/Burl26/buildLEGOInBlender

Burl26 / buildLEGOInBlender

Type to search

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

buildLEGOInBlenderPublic

PinUnwatch1Fork0Star0

main1 Branch0 Tags

Go to fileAdd fileCode

Burl26Add files via upload7aa7a39 · 1 minute ago3 Commits

README.md	Update README.md	5 minutes ago
read_kit_v2.blend	Add files via upload	1 minute ago
xref_csv.py	Add files via upload	1 minute ago

README

buildLEGOInBlender

How To Build LEGO Models In Blender

You will need the PDF instruction guide, the python file (.py), and the blender (.blend) file.

About

How To Build LEGO Models In Blender

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

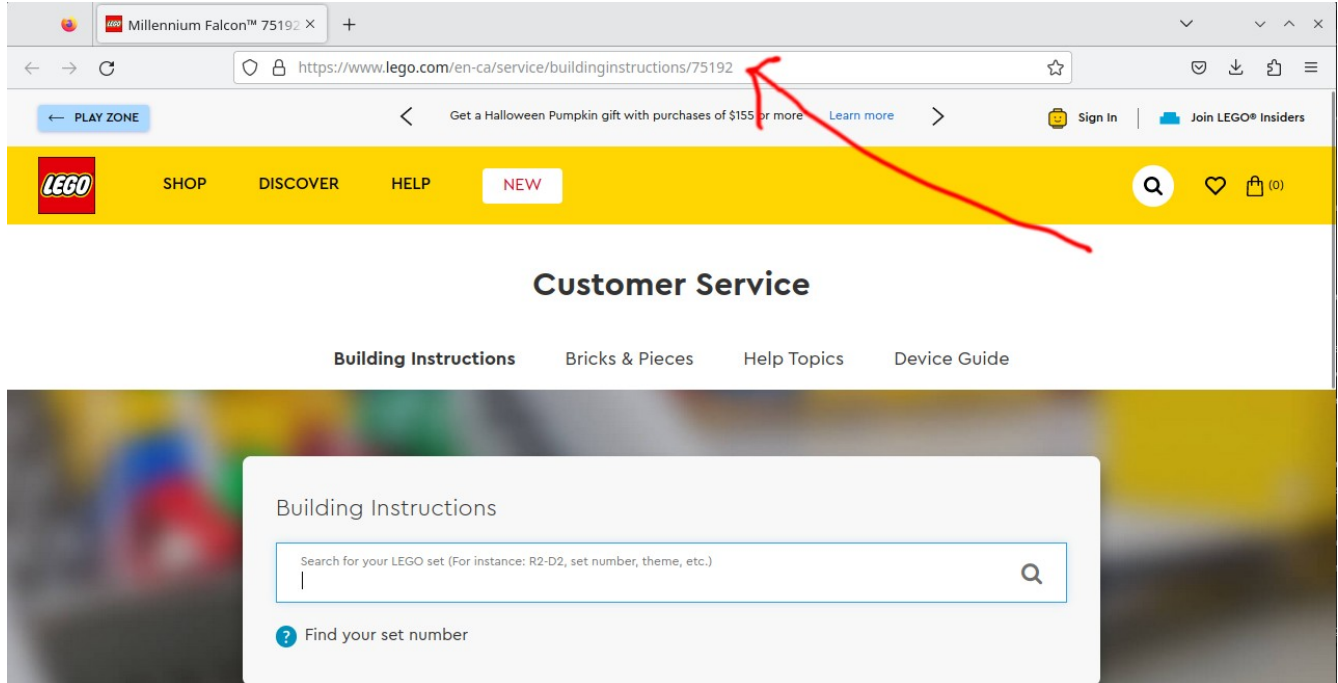
No packages published

Publish your first package

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

2. Select Your Kit

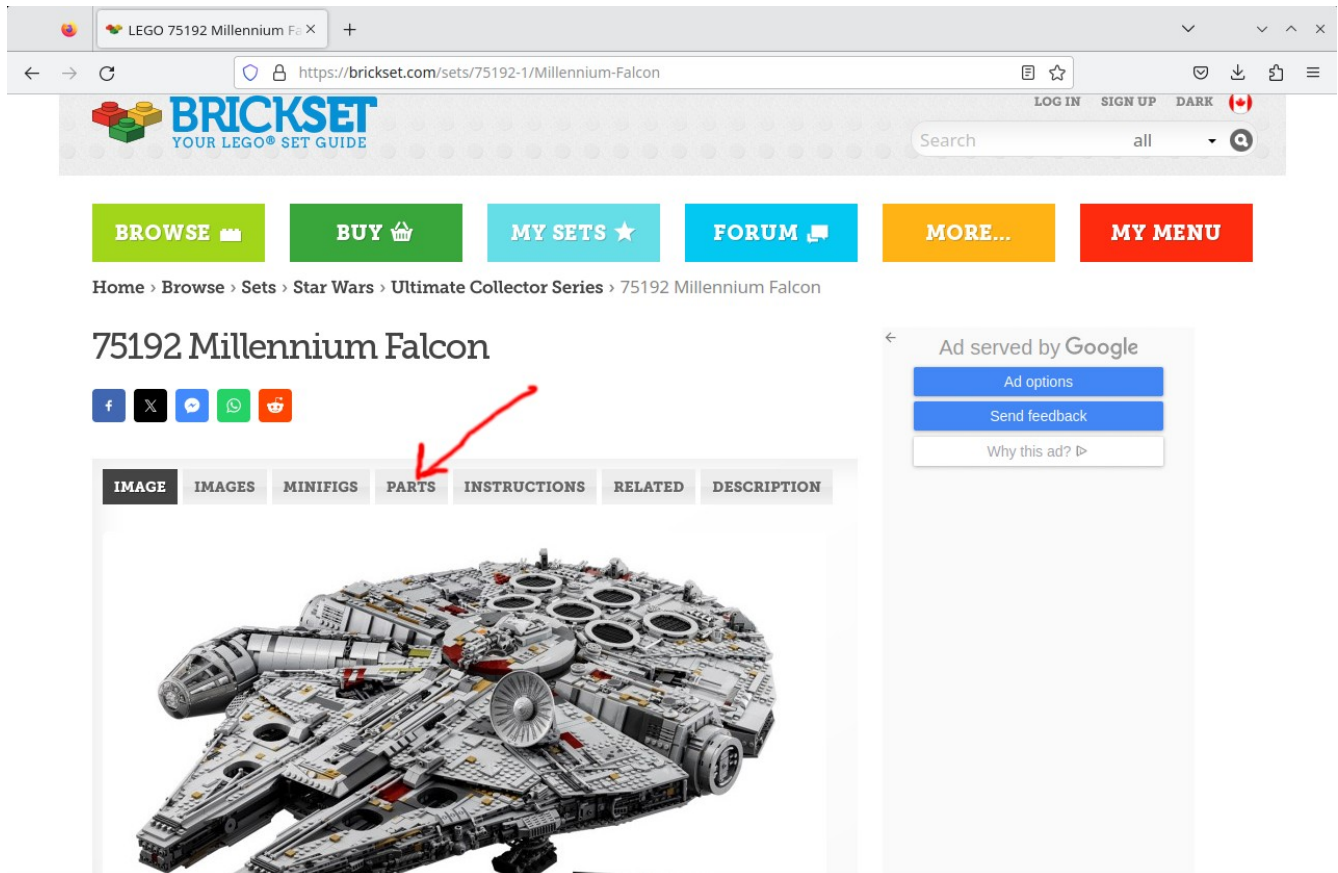
Go to the LEGO website. Find a kit and download the instruction manual.



3. Download the Parts List in CSV Format

Download the CSV parts list from brickset.com. Search for the kit number. Click the the “PARTS” button.

<https://brickset.com>



Click “View inventory as a table”.

75192 Millennium Falcon



[IMAGE](#) [IMAGES](#) [MINIFIGS](#) **[PARTS](#)** [INSTRUCTIONS](#) [RELATED](#) [DESCRIPTION](#)

This parts list has been obtained from [LEGO Customer Services](#). The set has **7541** parts and the inventory contains **7553**, so the inventory is **100%** complete.

[View inventory as a table](#) | [in the parts database](#)

Inventories at other sites are likely to be more accurate: [BrickLink](#) | [Rebrickable](#)

Then, “Download the inventory as a CSV file”


LEGO Inventory for 75192-1 X LEGO Set 75192-1 Millennium Falcon X

← → ↻

https://brickset.com/inventories/75192-1

☆

📄 ⬇️ 📄 ☰

 **BRICKSET**
YOUR LEGO® SET GUIDE

LOG IN SIGN UP DARK 🇨🇦

Search all 🔍

[BROWSE](#) [BUY](#) [MY SETS](#) [FORUM](#) [MORE...](#) [MY MENU](#)

Home > Browse > Sets > 75192-1 > Inventory for 75192-1: Millennium Falcon



Copy the csv to your working directory. Print this page as a PDF. This is used as a cross-reference between the LEGO part number and Ldraw part number. You can also verify the colour of the parts.

4. Verify Parts

In this step, a python program, xref_csv.py is used to verify each of the ldraw parts exist in the database. The program will search through the ldraw directory, the unofficial parts directory, and the mycollection directory. (This is why these directories need to be in the correct configuration, as directed in step 1).

Download and run xref_csv.py into your working directory. Depending on your system, the python executable may be different. Open a terminal (Windows users: command prompt) and type:

```
python xref_csv.py <LDIR root directory> <CSV file>
```

or*

```
python3 xref_csv.py <LDIR root directory> <CSV file>
```

or*

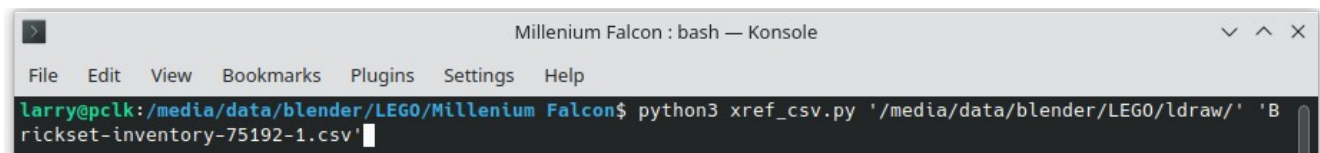
```
py xref_csv.py <LDIR root directory> <CSV file>
```

*depends on your system

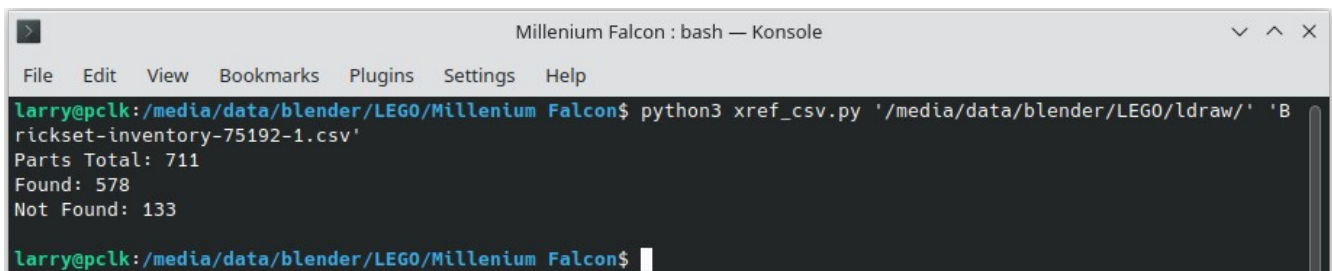
NOTE: Windows users may need to download and install python.

<https://www.python.org/downloads/windows/>

Here is an example of how to run xref_csv.py:

A terminal window titled "Millenium Falcon : bash — Konsole" with a menu bar (File, Edit, View, Bookmarks, Plugins, Settings, Help). The command entered is: `larry@pclk:/media/data/blender/LEGO/Millenium Falcon$ python3 xref_csv.py '/media/data/blender/LEGO/ldraw/' 'Brickset-inventory-75192-1.csv'`

The statistics will be printed. In this case, 133 parts were not found in the Ldraw database.

A terminal window titled "Millenium Falcon : bash — Konsole" with a menu bar (File, Edit, View, Bookmarks, Plugins, Settings, Help). The command entered is: `larry@pclk:/media/data/blender/LEGO/Millenium Falcon$ python3 xref_csv.py '/media/data/blender/LEGO/ldraw/' 'Brickset-inventory-75192-1.csv'`. The output displayed is: `Parts Total: 711
Found: 578
Not Found: 133`

The processed csv file will be labelled with an XREF in the name. For instance, the above command will produce a file named 'Brickset-inventory-75192-1XREF.csv'.

Because 133 parts were not found, you will need to provide alternate ldraw part numbers. You will need to edit the csv file and provide alternate ldraw part numbers, then rerun the xref_csv.py program.

Open the XREF file in a spreadsheet editor (LibreOffice, Excel, Google Sheets, etc).

Brickset-inventory-75192-1XREF.csv - LibreOffice Calc										
SetNumber	ElementID	Qty	Colour	Category	DesignID	ElementName	ImageURL	ElementSetCount	Use Part	Path
75192-1	6347700	16		Bricks	42929	BRICK 2X2 W. SNAP AND CROSS	/cdn/product-assets/element.img lod5photo.192x192/6347700.jpg	19		
75192-1	6351293	1		Plates	40687	WHEEL SUSPENSION 2X4 W/ SNAP	/cdn/product-assets/element.img lod5photo.192x192/6351293.jpg	34		40687/media/data/blender/LEGO/ldraw/ldraw/parts/40687.dat
75192-1	6356180	8		Bricks	65514	BEARING ELEMENT 2X2 W.D. SNAP	/cdn/product-assets/element.img lod5photo.192x192/6356180.jpg	17		
75192-1	6405646	19 (Not Specified)		Bricks	65514	BEARING ELEMENT 2X2 W.D. SNAP	/cdn/product-assets/element.img lod5photo.192x192/6405646.jpg	25		
75192-1	9339	1	Black	Minifigure Parts	73200	MINI BODY LOWER PART BLACK	/cdn/cs/set/assets/bb074ee8d31a6c5d59339.jpg	1014		9339/media/data/blender/LEGO/ldraw/ldraw/parts/9339.dat
75192-1	241226	102	Black	Plates	2412	RADIATOR GRILLE 1X2	/cdn/cs/set/assets/bb15d25d36e28392b2/241226.jpg	919		2412/media/data/blender/LEGO/ldraw/ldraw/parts/2412.dat
75192-1	242026	2	Black	Plates	2420	CORNER PLATE 1X2X2	/cdn/cs/set/assets/bb1c4825c65a356114w/242026.jpg	734		2420/media/data/blender/LEGO/ldraw/ldraw/parts/2420.dat
75192-1	244526	6	Black	Plates	2445	PLATE 2X12	/cdn/cs/set/assets/bb045a7d1e2763adea/244526.jpg	400		2445/media/data/blender/LEGO/ldraw/ldraw/parts/2445.dat
75192-1	30001	9	Black	Bricks	30001	BRICK 2X4	/cdn/cs/set/assets/bb15d25d36e28392b2/30001.jpg	734		30001/media/data/blender/LEGO/ldraw/ldraw/parts/30001.dat

Column B: LEGO part number.

Column F: Ldraw part number

Column J: **This is where you will enter the alternate part numbers to search.**

Column K: The path of the part number that was found.

Go to www.brickowl.com. Column K contains empty cells for all the parts that were not found. For each blank entry, use brickowl to search for ldraw part number from column F. The search results will indicate alternate part numbers. Enter this part number in column J.

LEGO Inventory for 75192-1 x LEGO Set 75192-1 Millennium Falcon x Search for "65514" | Brick Owl

https://www.brickowl.com/search/catalog?query=65514

Brick Owl Catalog 65514 Login / Register

Browse Wishlist Stores Cart

You are here: Home > Catalog > Search for "65514"

Search for "65514"

2 Items

Default

LEGO Brick 2 x 2 with Pins and Axlehole (30000 / 65514)

From: C\$0.01






LEGO Bonus/Value Pack Set 65514

2 Items

Default

Brickset-inventory-75192-1XREF.csv - LibreOffice Calc

File Edit View Insert Format Styles Sheet Data Tools Window Help

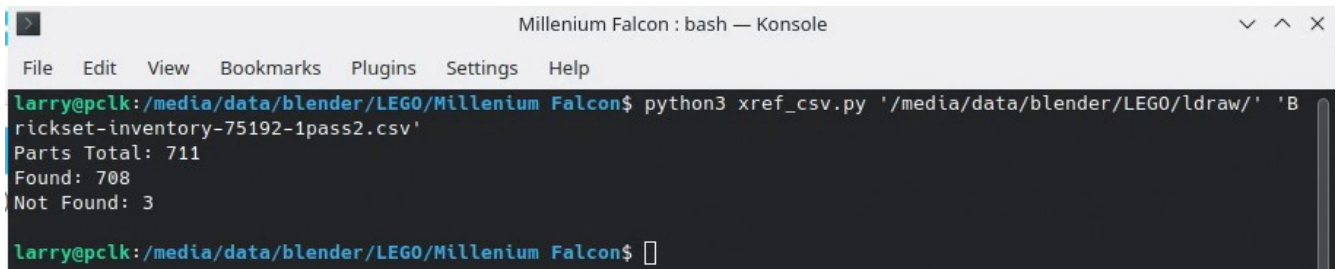
Liberation Sans 10 pt **B** *I* U     

F58 f_x Σ ▾ = 18747

	B	E	F	G	J	K
1	ElementID	Category	DesignID	ElementName	Use Part	Path
2	6347700	Bricks	42929	BRICK 2X2 W. SNAP AND CROSS	6232	
4	6356180	Bricks	65514	BEARING ELEMENT 2X2 W.D. SNAP	30000	
5	6405646	Bricks	65514	BEARING ELEMENT 2X2 W.D. SNAP	30000	

Once you have all the cross references in column J, then re-run xref-csv.py.

In my case, there were still 3 parts that could not be found. Note in the example below, that I renamed my XREF file 'Brickset-inventory-75192-1XREF.csv' to 'Brickset-inventory-75192-1pass2.csv'.




```
Millenium Falcon : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
larry@pclk:/media/data/blender/LEGO/Millenium Falcon$ python3 xref_csv.py '/media/data/blender/LEGO/ldraw/' 'Brickset-inventory-75192-1pass2.csv'
Parts Total: 711
Found: 708
Not Found: 3
larry@pclk:/media/data/blender/LEGO/Millenium Falcon$
```

Open the resulting ...XREF.csv file again. For each of the missing parts, try downloading the remaining parts (.dat file format) from digital-bricks.de. Put these .dat files into the “mycollection” directory and rerun xref-csv.py.

www.digital-bricks.de/en/file.php?part=<your missing part>

digital-bricks lxf to ldr konv X

www.digital-bricks.de/en/file.php?part=20105






deutsch
englisch

Digital Bricks

Home | Converter | Links | LDraw | Impressum & Privacy Clause

LDraw -> Not in LDraw

LDD Brickset: 2670
LDraw Update: 2020-02
PT Server time: 2020-12-15 14:50:42

Id	Name	Status	Image
6023	PORTABLE AIR UNIT	official	
30121	ALIEN, BREAST PLATE	official	
30323	POLAR RUCKSACK	official	
32008	FLIPPER	official	
42013	SKATE FOR BELVILLE FIGURE	official	
59275	FROGMAN'S FEET	unofficial	 3D .dat
62408	BREAST PLATE, 2	not available	 3D .dat
99415	MINI ARMOUR W/ARMS	not available	

With all the part paths defined in column K, you can proceed to import the parts into Blender.

```

Millenium Falcon : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
larry@pclk:/media/data/blender/LEGO/Millenium Falcon$ python3 xref_csv.py '/media/data/blender/LEGO/ldraw/' 'B
rickset-inventory-75192-1pass2.csv'
Parts Total: 711
Found: 708
Not Found: 3

larry@pclk:/media/data/blender/LEGO/Millenium Falcon$ python3 xref_csv.py '/media/data/blender/LEGO/ldraw/' 'B
rickset-inventory-75192-1pass2.csv'
Parts Total: 711
Found: 711
Not Found: 0
larry@pclk:/media/data/blender/LEGO/Millenium Falcon$


```

5. Import Parts into Blender

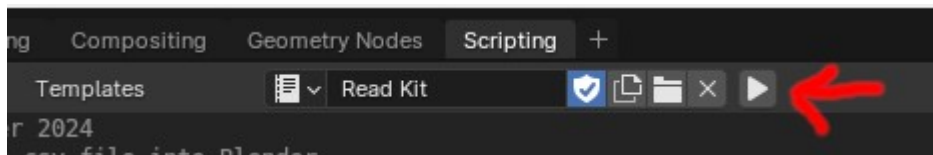
Download read_kit_v1.blend from my github. This is an empty file with some python scripts. The blender file will open directly to the scripting tab and you will see the python code.

Modify the three lines containing the path and csv filename.

```
16
17 # these links worked up to October 2024
18 # https://github.com/TobyLobster/ImportLDraw/releases (tool to import into Blender)
19 # https://library.ldraw.org/updates?latest (ldraw database)
20 # https://library.ldraw.org/library/unofficial/ldrawunf.zip (unofficial parts)
21 # https://www.brickowl.com/catalog (cross-referencing)
22
23 #=====
24 # EDIT THESE LINES!!
25 #
26 CURRENT_PATH = '/media/data/blender/LEGO/Star Wars/Millennium Falcon/'
27 LDRAW_PATH = '/media/data/blender/LEGO/ldraw/ldraw/'
28 mycsvfile = CURRENT_PATH + 'Brickset-inventory-75192-1XREF.csv'
29 #
30 #=====
31
32 colour_palette = [
33 ["AQUA", 0xD3F2EA, 1, 0.1, 0.6, 0.05], \
34 ["BLACK", 0x000000, 1, 0.1, 0.6, 0.05], \
35 ["BRIGHT GREEN", 0x58A443, 1, 0.1, 0.6, 0.05], \
```



Run the python program.



Once all the parts are imported:

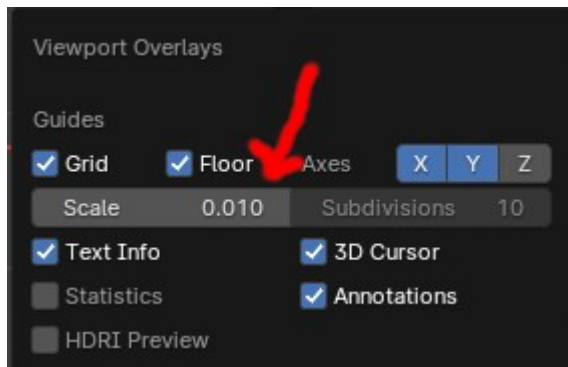
- Clean Up->Recursive Unused Data Blocks
- Create a single collection called “bricks” and put everything in there. You can hide these from both viewport and render views.

Important:

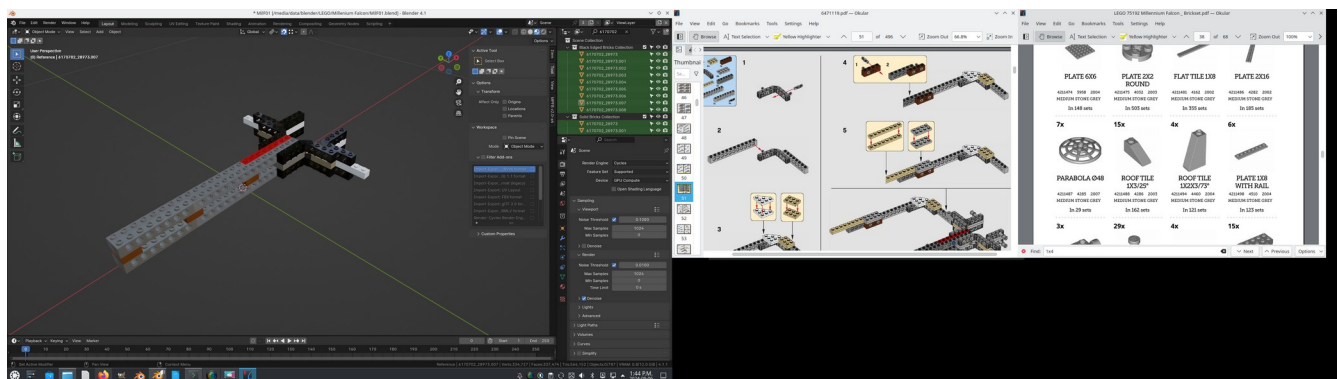
To maximize performance, the bricks were imported **without** bevel edges.

6. Start Building

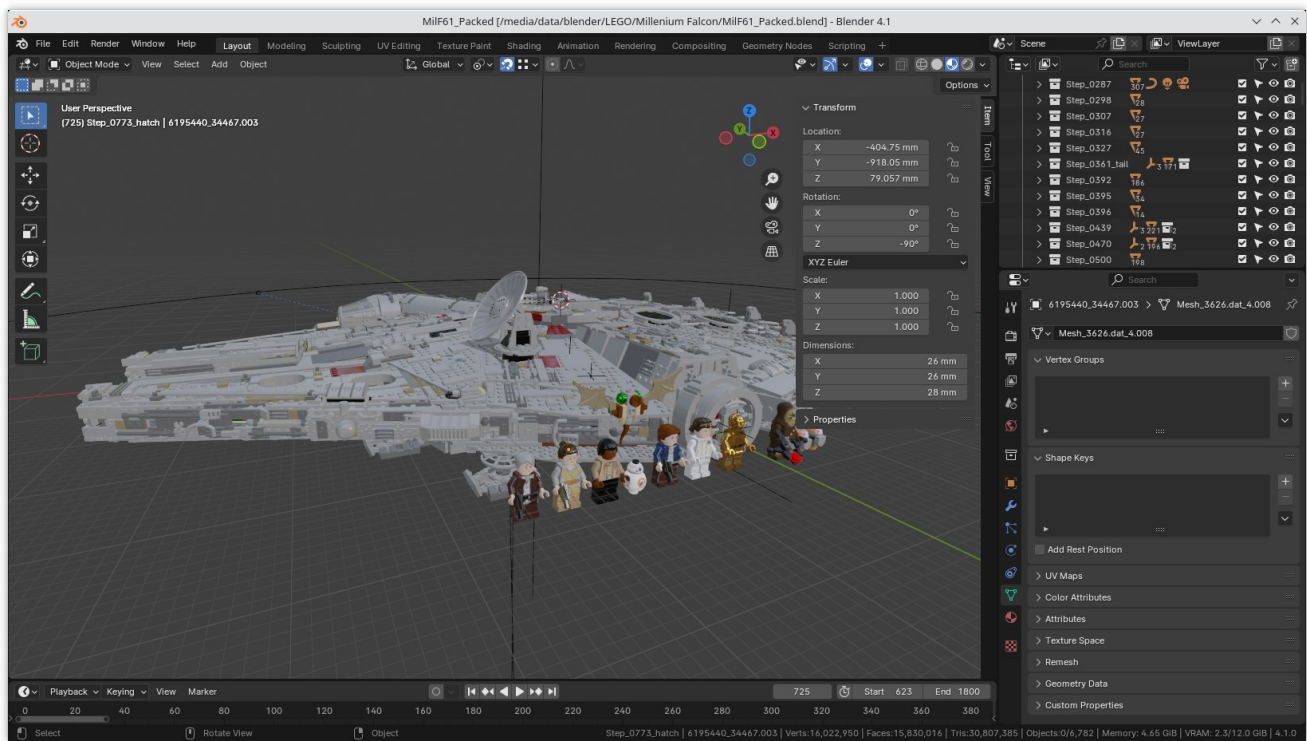
- I set my scale to 0.01 and used increment snapping



- I used two screens. My left screen was Blender. On the right screen, I had both the instruction manual and the PDF of the parts inventory from brickset.com list.

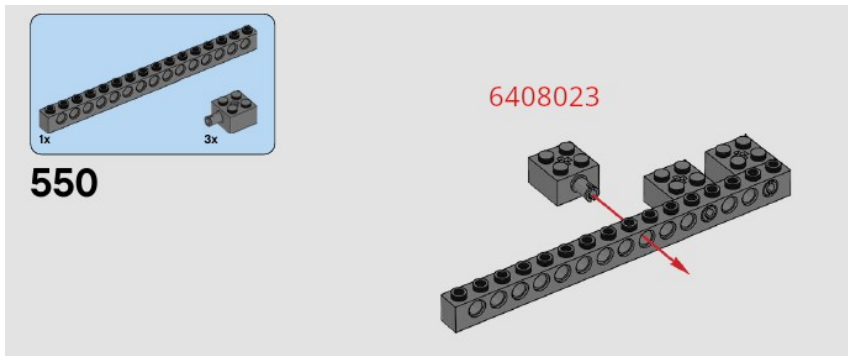
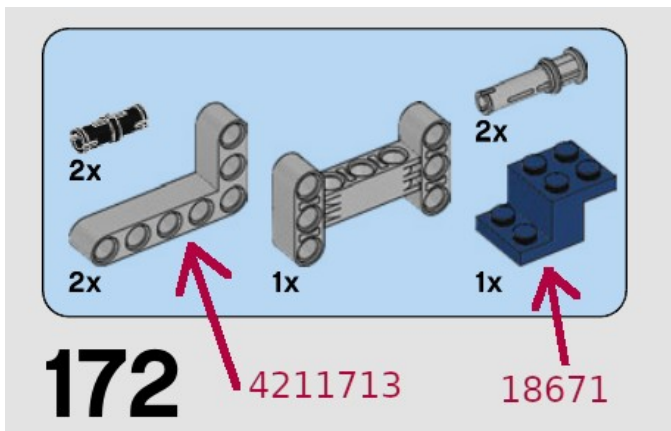


- Every part in the model is a copy of the master set.
- I grouped the parts into collections corresponding the step number from the manual. This made it easier to isolate sections for fitting assemblies together.

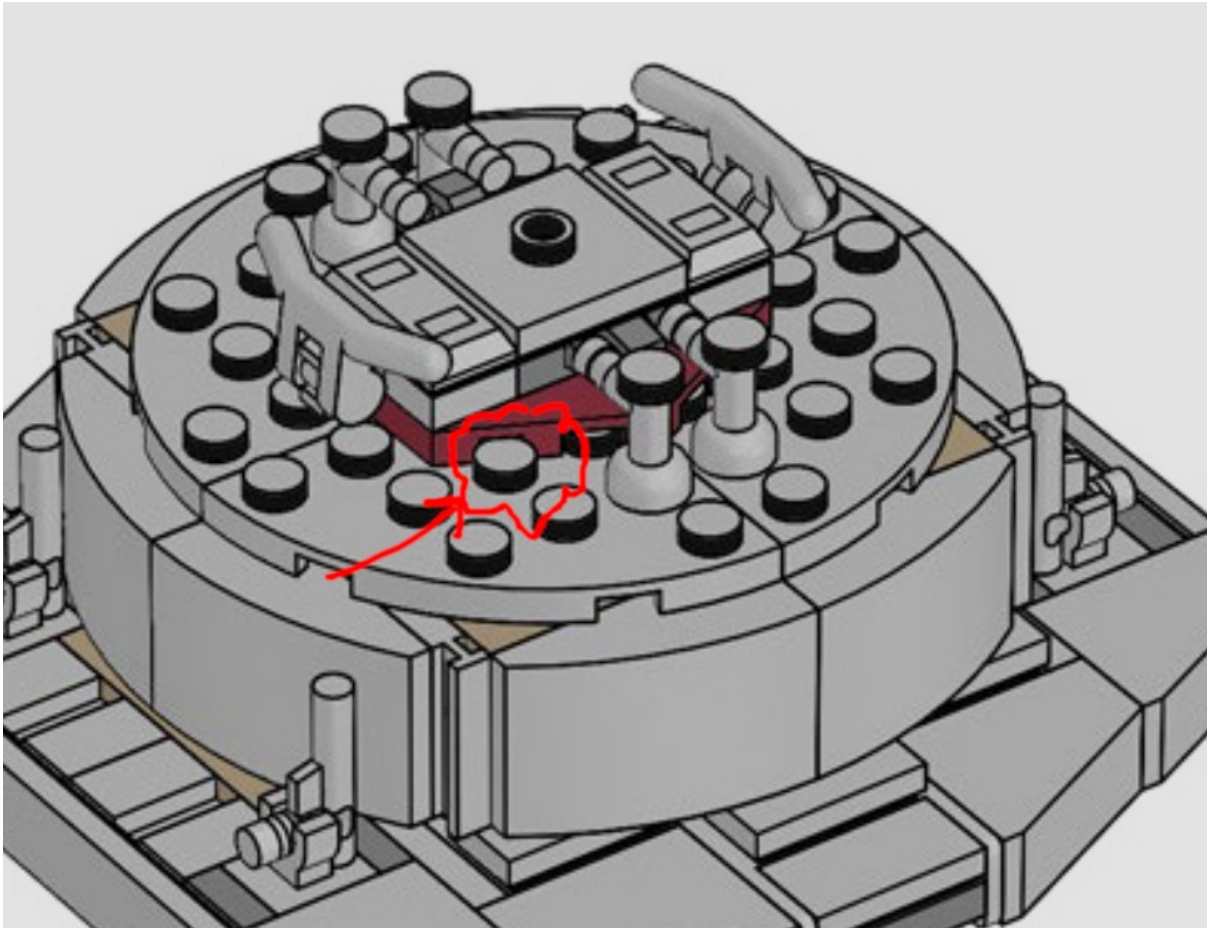


Notes:

- The csv data set may not be accurate. Some parts may be missing or may have the incorrect material assigned. You can fix this up as you build. For missing parts, you can always use the import->Ldraw to load single objects.
- These objects do have textures. You will need to create these textures yourself from online images or the instruction manual.
- There may be errors in the instruction manual. For this kit, I found parts that were not in the parts list at the end of manual:



- The Blender models are “rigid”. I’ve noticed that the some geometry in LEGO does not work out trigonometrically. You will have to make some positioning adjustments. Additionally, there will be some interferences. In real LEGO, there is enough compliance in the assembly that the assembly will bend into a workable position. In Blender, you will have to overlook some interferences. Here is a view from the instruction manual that clearly show an interference, so the designers also had to overlook this.



Additional Utilities

In read_kitV2.blend, there is another script file called Utils, containing two useful python scripts.

reduceMaterials()

If you append items, the materials will duplicate. This utility reassigns the materials back to the original.

bevelModifier()

After you are finished, you may wish to apply the bevel modifier to make the brick edges look more realistic. This utility will save you from adding the modifier to every part. It takes a while to add the modifier to a project of this size, so give it a lot of time to complete.

To use them, switch over to the Utils script file. Uncomment (delete the #) the utility you want to run and execute the script.

```
View Text Edit Select Format Templates Utils
1 import bpy
2
3
4 def reduceMaterials():
5     # select all
6     objects = bpy.context.scene.objects
7     #
8     for o in objects:
9         #if bpy.app.version[0] == 4 and bpy.app.version[1] == 1 :
10        print(f'{o.name}')
11        #bpy.ops.object.select_all(action='DESELECT')
12        #o.select_set(True)
13        matslots = o.material_slots
14        if not matslots == None:
15            for m in matslots:
16                try:
17                    abb = float(m.material.name[len(m.material.name)-4:])
18                except:
19                    abb = 999
20                if abb < 1:
21                    print(f'fixing {o.name}')
22                    basemat = m.material.name[:len(m.material.name)-4]
23                    m.material = bpy.data.materials[basemat]
24
25 def bevelModifier():
26     #bpy.context.space_data.context = 'MODIFIER'
27     for o in bpy.data.collections['Model'].all_objects:
28         if len(o.modifiers) == 0:
29             print(f'bevel {o.name}')
30             bpy.ops.object.select_all(action='DESELECT')
31             o.select_set(True)
32             bpy.context.view_layer.objects.active = o
33             argflag = True
34             try:
35                 bpy.ops.object.modifier_add(type='BEVEL')
36             except:
37                 print(f'no bevel modifier added for {o.name}')
38                 argflag = False
39             if argflag:
40                 bpy.context.object.modifiers["Bevel"].width = 0.0005
41                 bpy.context.object.modifiers["Bevel"].segments = 5
42                 bpy.context.object.modifiers["Bevel"].limit_method = 'WEIGHT'
43     print('done')
44
45 #un-comment out the utility you want to run
46 #reduceMaterials()
47 #bevelModifier()
48 |
```