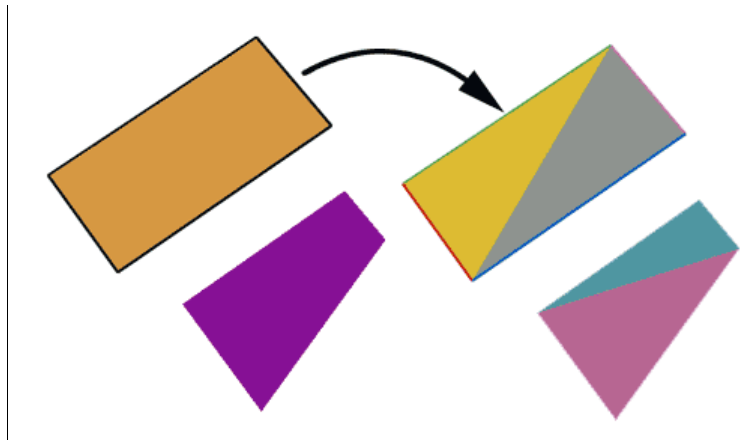


Unrectifier



Rectifier is a useful tool to consolidate triangles into quads and to keep file size down, but doing so make further improvements to the file more difficult. **Unrectifier** does the opposite, it expands Rect primitives into quads + lines, and split quads into triangle pairs. By default **Unrectifier** operates on the whole file, but it can also limit itself to key-colored elements. With some limitations, it is also able to update condlines around modified areas.

It is a simple console application, source code is provided below to anyone willing to integrate it in a more palatable interface. You may also use Michael Heidemann [LETGUI](#) front-end (highly recommended!).

Download

[Unrectifier package](#), including program for Windows, documentation, source files (Visual C++ 6.0), example files.

History

- V1.0: Initial release

Usage

- Prepare the input LDraw file. Nothing special to do if you want to apply **Unrectifier** to the whole file, otherwise color the primitives that you want to inline and the quads that you want to split using key color.
- Launch a command prompt.
- Type the command line: `unrectifier[-k <val>] [-r] [-q][-c][-u][-l <Ldraw path>] LdrawFileIn LdrawFileOut`. **Unrectifier** will create LdrawFileOut, containing the original file with selected Rect primitive inlined and quads split into triangles.

Here is a screen shot of a sample run:

```

C:\WINDOWS\system32\cmd.exe
D:\Unrectifier>Unrectifier.exe -l 1:\ldraw ur1.dat ur2.dat

Unrectifier 1.0 - by Philo
-----

Reading input file ur1.dat
4 lines in file
0 Edge lines
0 Conditional lines
0 Triangles
1 Quads
1 Subfiles

LDraw subfiles search locations:
.\, .\p\, .\s\, 1:\ldraw\p\, 1:\ldraw\parts\, 1:\ldraw\parts\s\, 1:\ldraw\Unoffi
cial\p\, 1:\ldraw\Unofficial\parts\, 1:\ldraw\Unofficial\parts\s\

Expanding Rect primitives

Inlined from subfiles:
4 Edge lines
0 Conditional lines
0 Triangles
1 Quads

Splitting quads

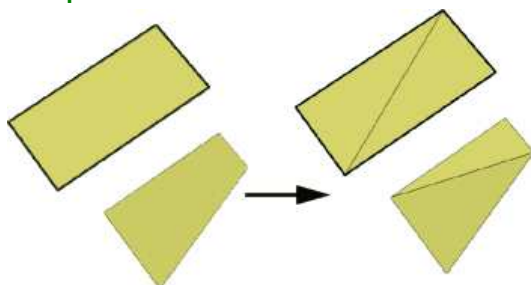
Writing output file ur2.dat
10 lines written in file
4 Edge lines
0 Conditional lines
4 Triangles
0 Quads
0 Subfiles
Press <Enter> to quit

```

How Unrectifier works

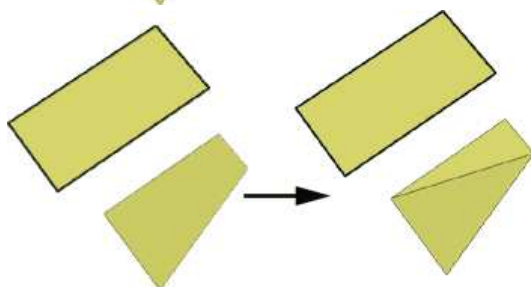
- Input file is read and parsed, stored in an array.
- Array is scanned to find quads or Rect primitives. If no key color is specified, all Rects are inlined and quads get split into 2 triangles.
- The split chosen is the one that gives the smaller angle between the two triangles.
- If key color is specified, only elements that match key color are processed. Note that actually **ALL** key-colored primitives or subfiles are inlined. This allows you to "unboxify" a box primitive, or more generally deep-inline any primitive/subpart.
- If **-c** option is not specified, condlines bordering split quads are deleted, and new ones are generated around each triangles. Only simple generation is done (condline created for any angle $> 0.1^\circ$, no condline created between the triangles and touching primitives/subpart). For more fancy condlines management, use [Edger2!](#).
- Output file is written.

Examples



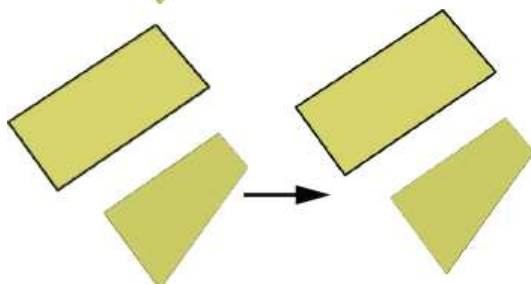
The input file contains a Rect primitive (top left) and a warped quad. Using default options, the Rect is inlined, then the quads are split into triangles.

Command line: `Unrectifier -l 1:\ldraw ur1.dat ur2.dat`



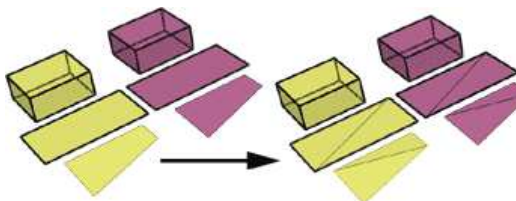
-r option prevents inlining of Rect primitives. Note that if you don't specify the path of your LDraw file, **Unrectifier** is unable to inline primitives, and the result is the same!

Command line: `Unrectifier -r -l 1:\ldraw ur1.dat ur3.dat`
 Command line: `Unrectifier ur1.dat ur3.dat`



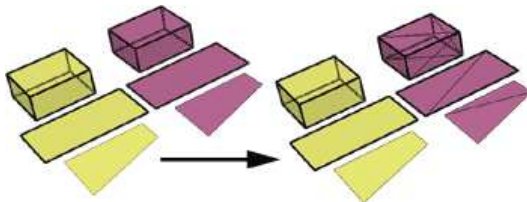
-q option prevents splitting of quads into triangles. Notice the very uninspired thumbnail nearby ;)

Command line: `Unrectifier -q -l 1:\ldraw ur1.dat ur4.dat`



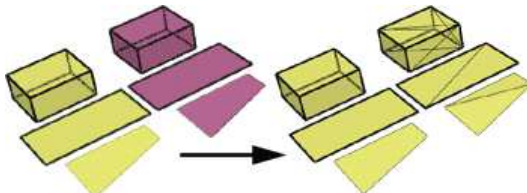
If no key color is specified, **Unrectifier** does its job on all quads and Rect primitives. Note that Box primitives are unaffected.

Command line: `Unrectifier -l 1:\ldraw ur5.dat ur6.dat`



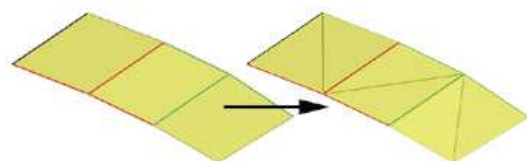
With **-k <color>** option, only elements that have the proper color are processed. This time, ALL key-colored primitives are inlined. This allows you to "unboxify" a box primitive, or more generally deep-inline any primitive/subpart.

Command line: `Unrectifier -k 5 -l 1:\ldraw ur5.dat ur7.dat`



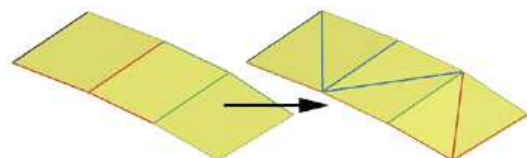
-u option returns the whole file to uncolored state.

Command line: `Unrectifier -k 5 -u -l 1:\ldraw ur5.dat ur8.dat`



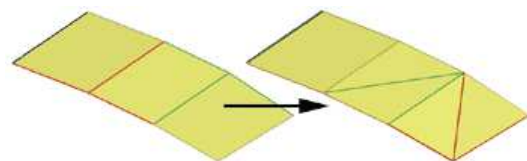
-c option prevents conditional lines management: the condlines existing in original file are left as-is. But as you can see on nearby thumbnail, they probably won't match properly the triangles, and no conditional line is added when a warped quad is split. If you use this option you should at some time regenerate condlines with [Edger2](#).

Command line: `Unrectifier -c -l 1:\ldraw ur9.dat ur10.dat`



Default operation (without **-c** option) deletes existing condlines around split quads, and regenerates them. Note that this process, applied one triangle at a time, is very inefficient and may take some time on big files.

Command line: `Unrectifier -l 1:\ldraw ur9.dat ur11.dat`



Regeneration of condlines is basic, a condline is created for any angle $> 0.1^\circ$, and no condline created between the triangles and touching primitives/subpart. Here for example no condline was created along remaining quad (**-r** option). For more fancy condlines management, use [Edger2](#)!

Command line: `Unrectifier -r -l 1:\ldraw ur9.dat ur12.dat`



[Version Française ici.](#)