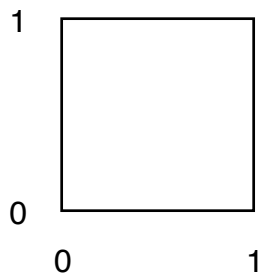


Explanation of the Polygon Constants SPLINE, ARC, ARC_AUTO, NEW_LOOP, NEW_POLY, NO_CORE, and END_POLY

Important: These tag constants start at 256 and go up. Both X *and* Y coordinates of polygon points must be less than these tag values (i.e. less than 256). Some glyph-processing code *assumes* this. Glyph coordinate values don't go above about 60, so this is not a problem.)

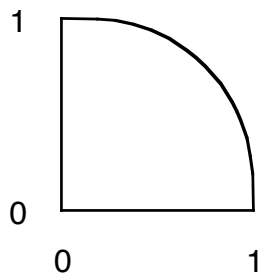
Note: Not all of these constants are supported in any given function that uses them. A function will support only a subset of them, as described in that function's comments.

Also see the webpage <http://alienryderflex.com/polyspline>



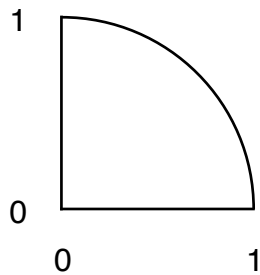
```
0,0, 0,1, 1,1, 1,0, END_POLY
```

Each polygonal font glyph consists of a list of items, terminated by the END_POLY tag. These items are usually coordinate pairs (also called “hard points”) that specify the points of a polygon.



```
0,0, 0,1, SPLINE,1,1, 1,0, END_POLY
```

The spline point creates a simple spline curve from the preceding point to the following point.



```
0,0, 0,1, ARC,0,0, 1,0, END_POLY
```

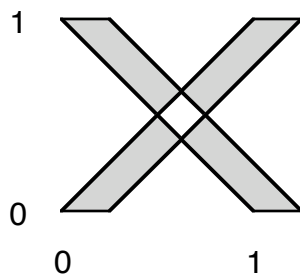
or

```
0,0, 0,1, .001,1, ARC_AUTO, 1,.001, 1,0, END_POLY
```

ARC and ARC_AUTO can be used to make a semi-circular curve instead of a spline curve. It actually creates two spline curves that approximate a semi-circular curve. This curve should not be pushed significantly beyond 90° or it will start to look less than circular. It may fail completely at 180°. See the file “ARC Points” for more information.

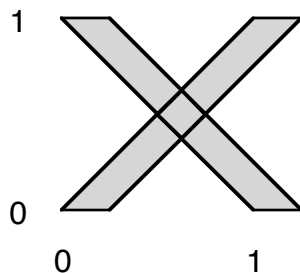
When using ARC, the ARC point specifies the *center* of the circle. Note that even if the preceding point and subsequent point are not exactly the same distance from the ARC point, the curve will still connect to them perfectly — it will just be a tiny bit non-circular. The ARC point must not be the first item in its polygon (or polygon loop; see NEW_LOOP below), nor should it immediately follow a NEW_LOOP tag, a NEW_POLY tag, or another ARC point. It is OK for the ARC point to be the last point in a polygon loop (keeping in mind that the polygon continues at its first point).

The ARC_AUTO tag does not come with coordinate points. It specifies that the previous point and the subsequent point should be connected with a semi-circular curve that is colinear with what the polygon was doing at those points. The ARC_AUTO tag must come after two consecutive hard corners (simple coordinate pairs) and before two consecutive hard corners (simple coordinate pairs). The ARC_AUTO tag must not be the first or second item in its polygon (or polygon loop); i.e. it must actually be preceded by two hard corners. But it may be the last or second-to-last item in its polygon or polygon loop (provided the required two hard corners occur as the loop wraps back to its start).



```
0,0, 1,1, 1.25,1, .25,0, NEW_LOOP,
0,1, 1,0, 1.25,0, .25,1, END_POLY
```

The NEW_LOOP tag allows a polygon to include multiple polygon loops. These loops may overlap, as in this example, which will cause the overlap area to cancel out.



```
0,0, 1,1, 1.25,1, .25,0, NEW_POLY,
0,1, 1,0, 1.25,0, .25,1, END_POLY
```

The NEW_POLY tag works exactly like NEW_LOOP, but the overlap area does not cancel out.

NO_CORE

The NO_CORE tag is used only in the Title font. That font's glyphs consist of yellow polygons with slightly-inset black cores. The black cores are not included in the font's data, but are generated automatically by the rendering code.

If used in a glyph, NO_CORE must come at the start of the glyph, or immediately after a NEW_LOOP tag. It indicates that all subsequent polygon loops in this glyph should *not* have a black core inside them. NO_CORE does not need to be used more than once per glyph.