

Inhalt

Import von SVG Curves.....	1
SVG-Datei erstellen.....	1
SVG Konvertierung zu OBJ in Blender.....	2
Import als C-Array.....	8

Import von SVG Curves

Curves, welche in einer SVG-Datei gespeichert sind, sollen in ein Vertex- und ein Line-Array konvertiert werden.

Ablauf:

- SVG-Datei erstellen, z.B. mit Affinity Designer.
- SVG-Datei in Blender importieren.
- Export aus Blender in Wavefront OBJ-Datei.
- Manuelle Nachbearbeitung und Konvertierung zu Array.

SVG-Datei erstellen

In Affinity Designer eine Bitmap-Vorlage laden und mittels Pen Tool die Curves tracen:

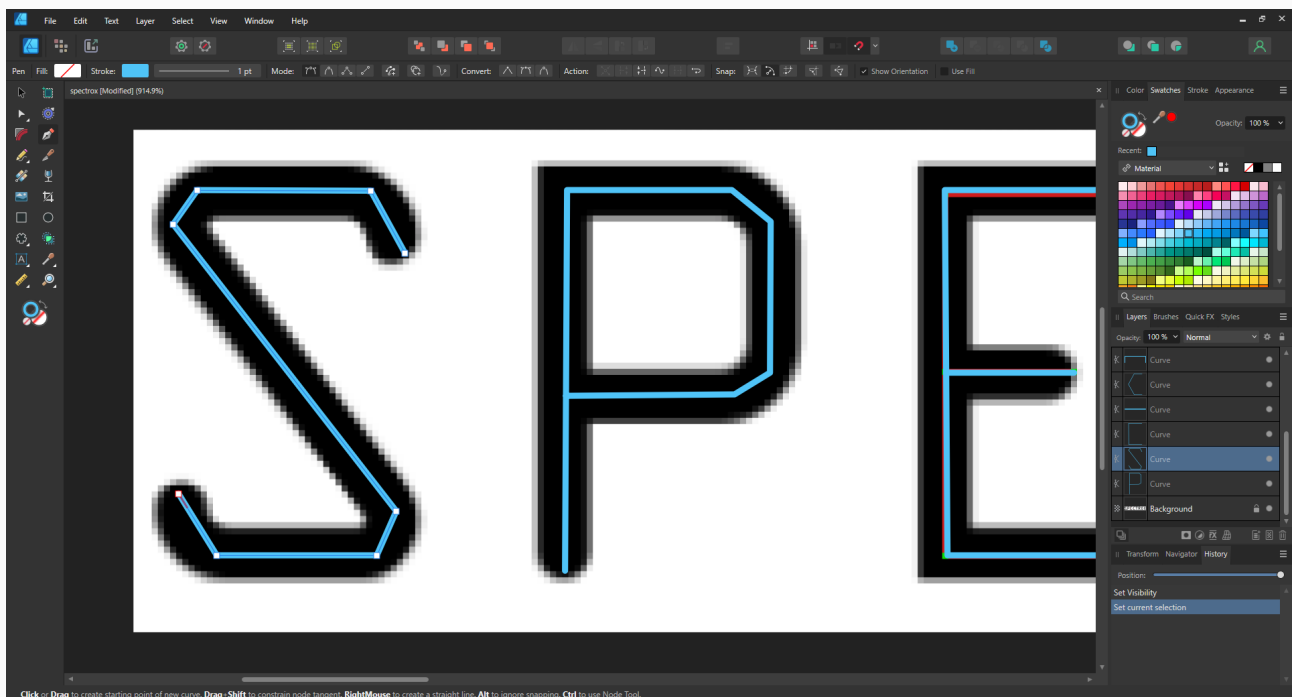


Figure 1: Line Tracing in Affinity Designer.

Danach den Bitmap-Layer löschen oder deaktivieren und mittels **File** → **Export...** als SVG-Datei exportieren:

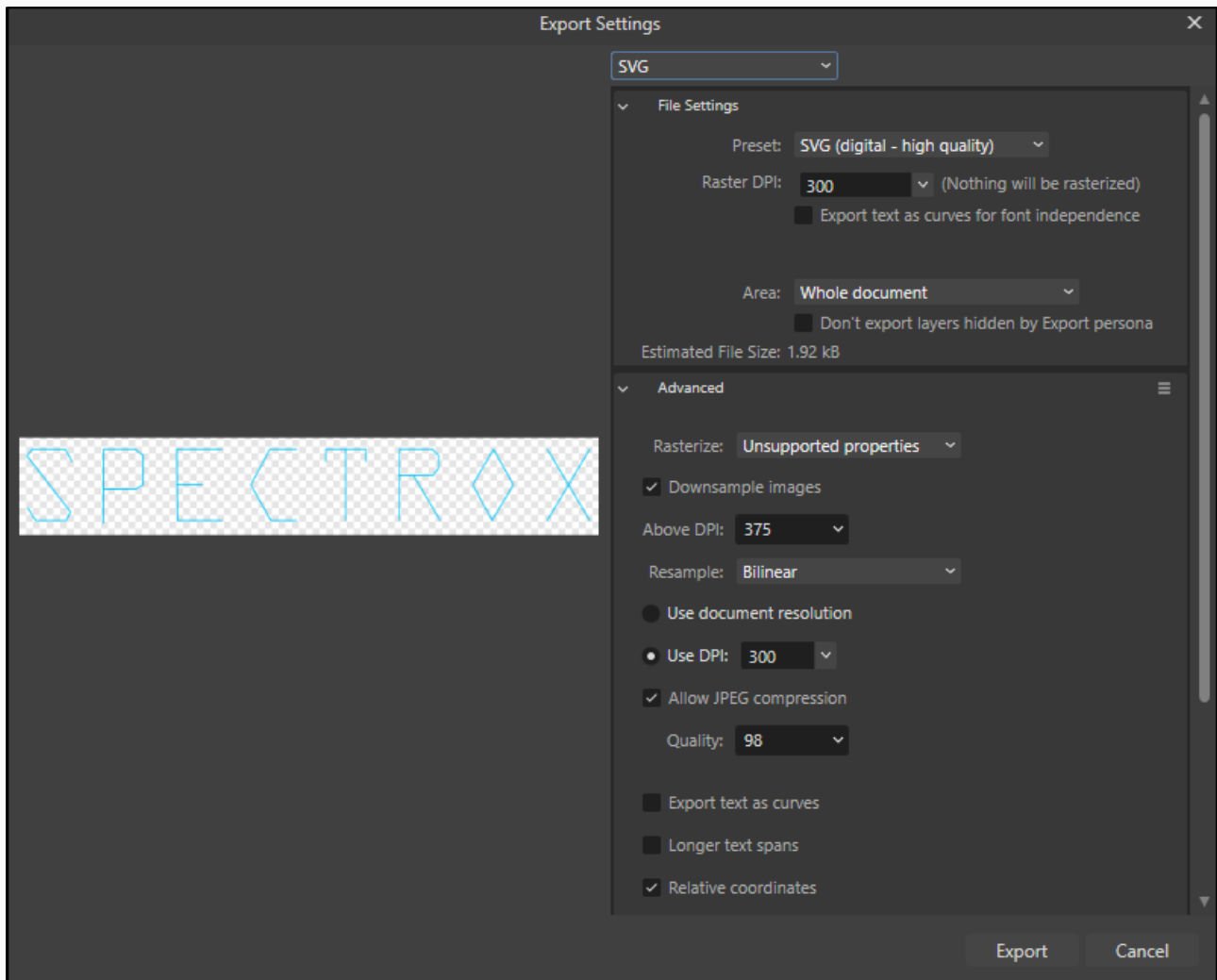


Figure 2: SVG-Export.

SVG Konvertierung zu OBJ in Blender

Blender öffnen und ein neues General Projekt erstellen. Den Default-Würfel, Kamera und Lichtquelle löschen.

File → **Import...** → **Scalable Vector Graphics (.svg)** ausführen und die SVG-Datei importieren.

Zum Modeling-Modus wechseln und Ansicht auf **Top Orthographic** stellen (Z-Icon anklicken):

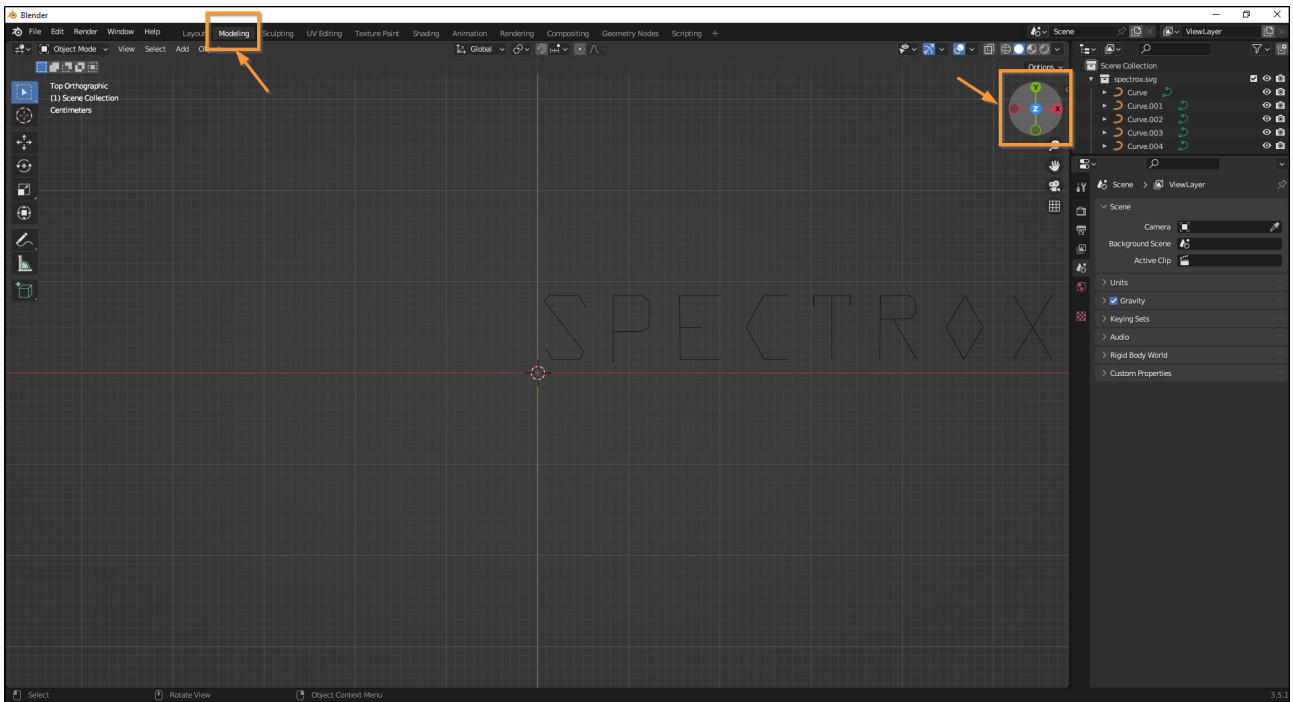


Figure 3: Modeling-Modus und ortographische Top-Ansicht.

Den ersten Buchstaben selektieren (helles Orange), danach alle anderen Buchstaben (dunkles Orange). Danach **Object** → **Join** ausführen.

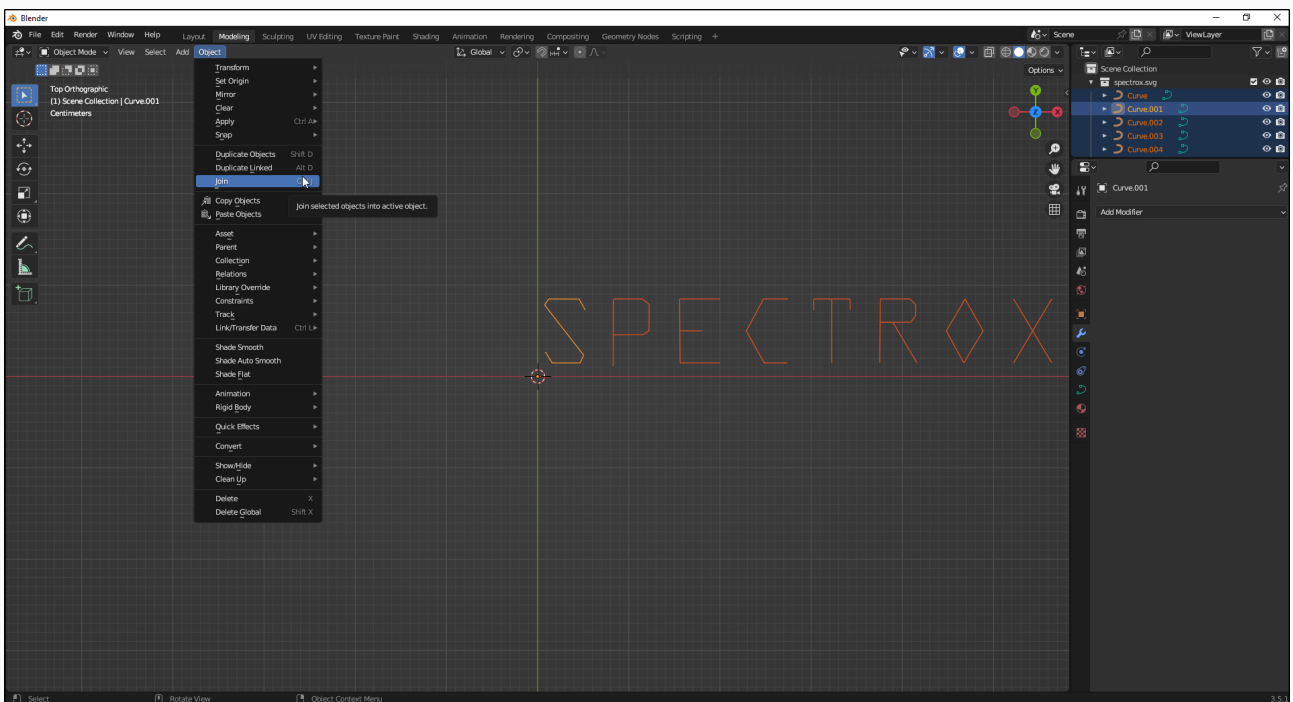


Figure 4: Linien joinen.

In der Scene-Collection am rechten, oberen Rand sieht man nun bloss noch ein Kurven-Objekt.

Als nächstes soll das Objekt auf den World-Origin zentriert werden.

Dazu sind folgende Schritte nötig:

- Objekt-Origin in das Zentrum der Bounding-Box setzen.
- Vorher kontrollieren, dass die Bounding-Box wirklich bloss die gewünschten Vertices enthält.
- Verschieben des Objekts zum World-Origin.

Bounding-Box einschalten in Object Properties → Bounds → Box:

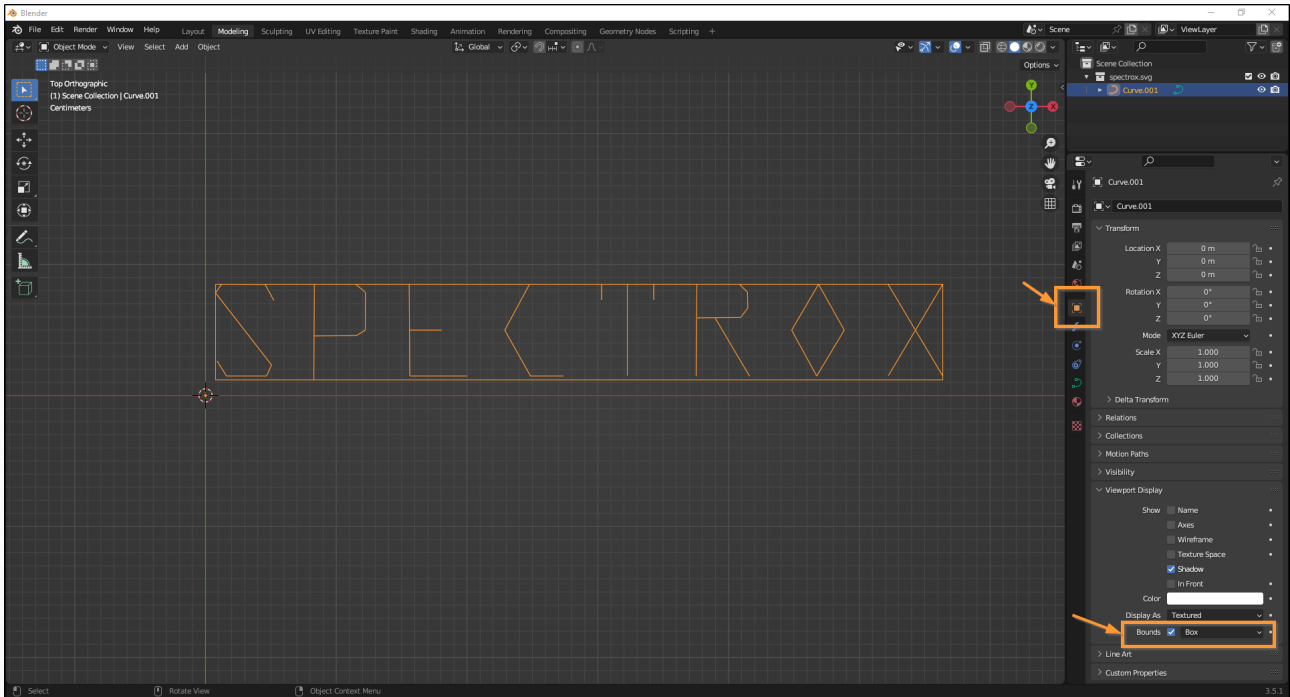


Figure 5: Bounding-Box anzeigen.

Obwohl die Bounding-Box in Ordnung aussieht, müssen die einzelnen Vertices kontrolliert werden.

Dazu mit TAB in den Edit-Modus wechseln und mit A alle Vertices selektieren:

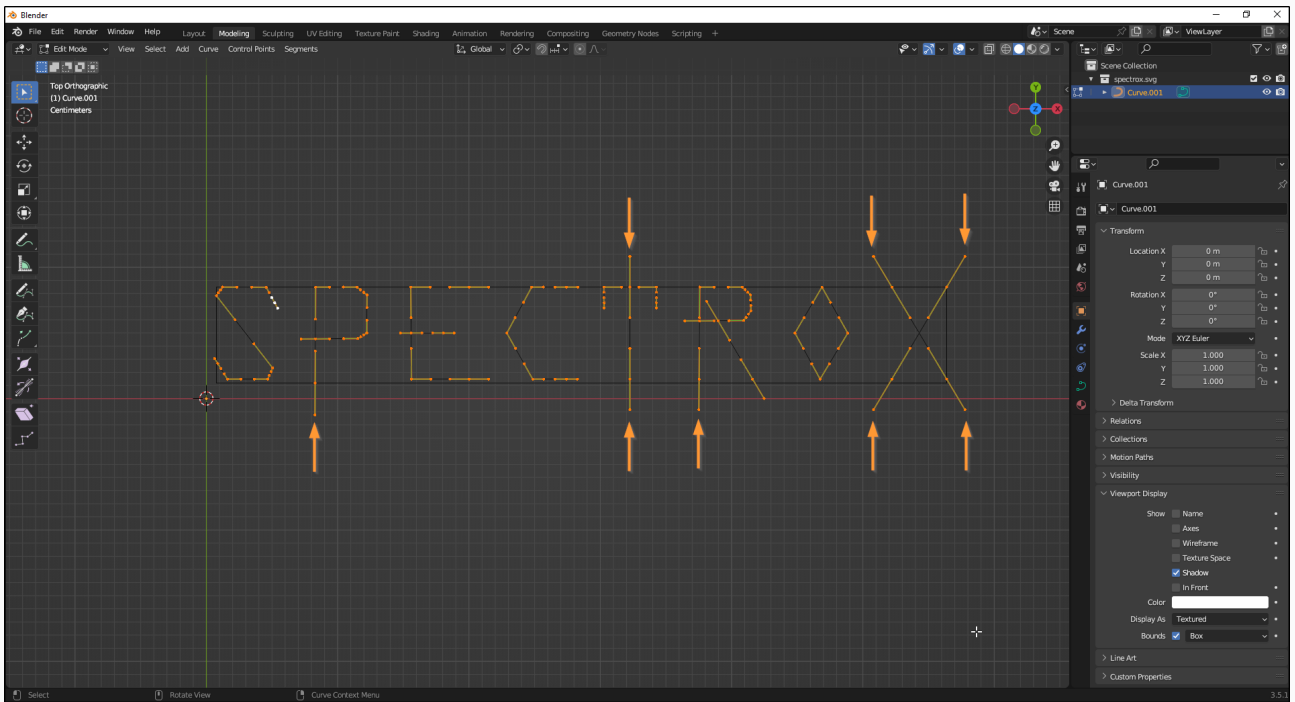


Figure 6: Kontrolle der Vertices und Vertex-Handles.

Man sieht dass gewisse Vertex-Handles über die Bounding-Box schiessen. Da diese bei der Berechnung des Bounding-Box-Zentrums auf einberechnet werden, müssen sie in die Box verschoben werden.

Mit der Maus die Handles selektieren, danach mit G in den Move-Modus wechseln, die Handles in die Bounding-Box verschieben und mit linker Maustaste bestätigen:

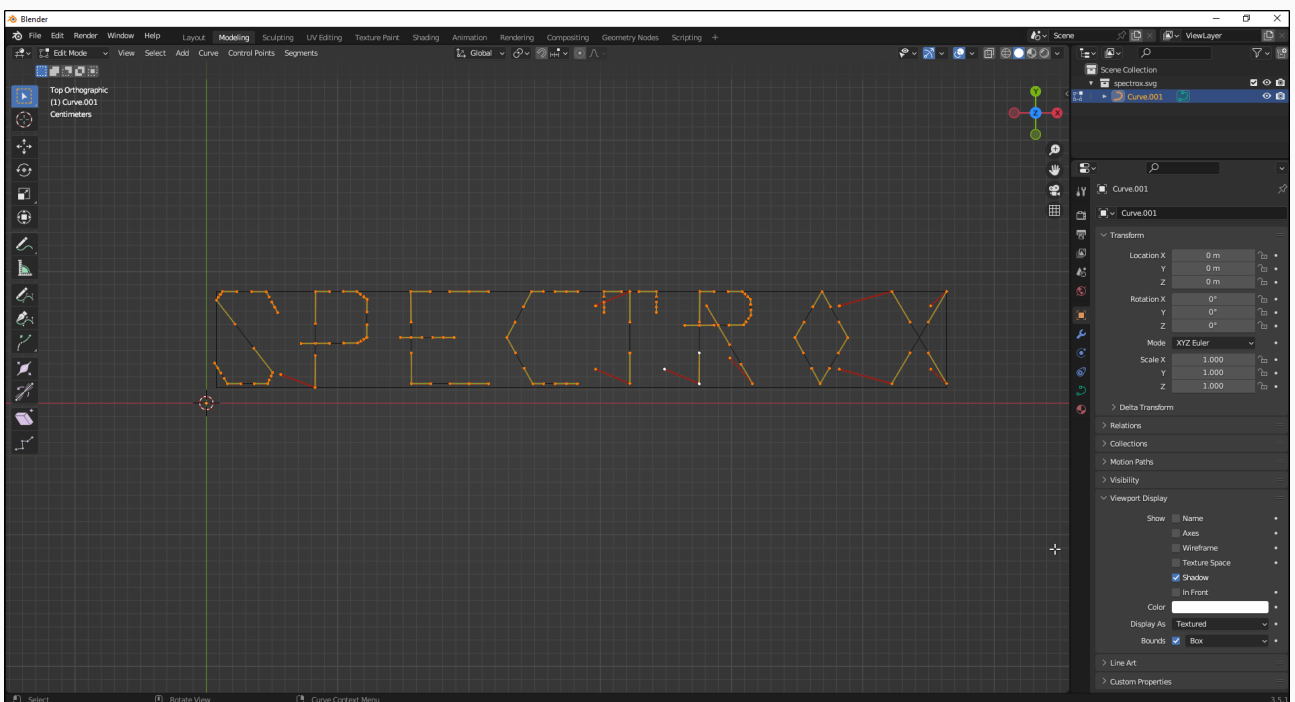


Figure 7: Alle Handles sind nun innerhalb der Bounding-Box.

Danch kann der Edit-Modus mit TAB wieder verlassen werden.

Jetzt kann das Objekt-Origin ins Zentrum der Bounding-Box gesetzt werden: den Transform Pivot Point setzen auf Bounding Box Center:

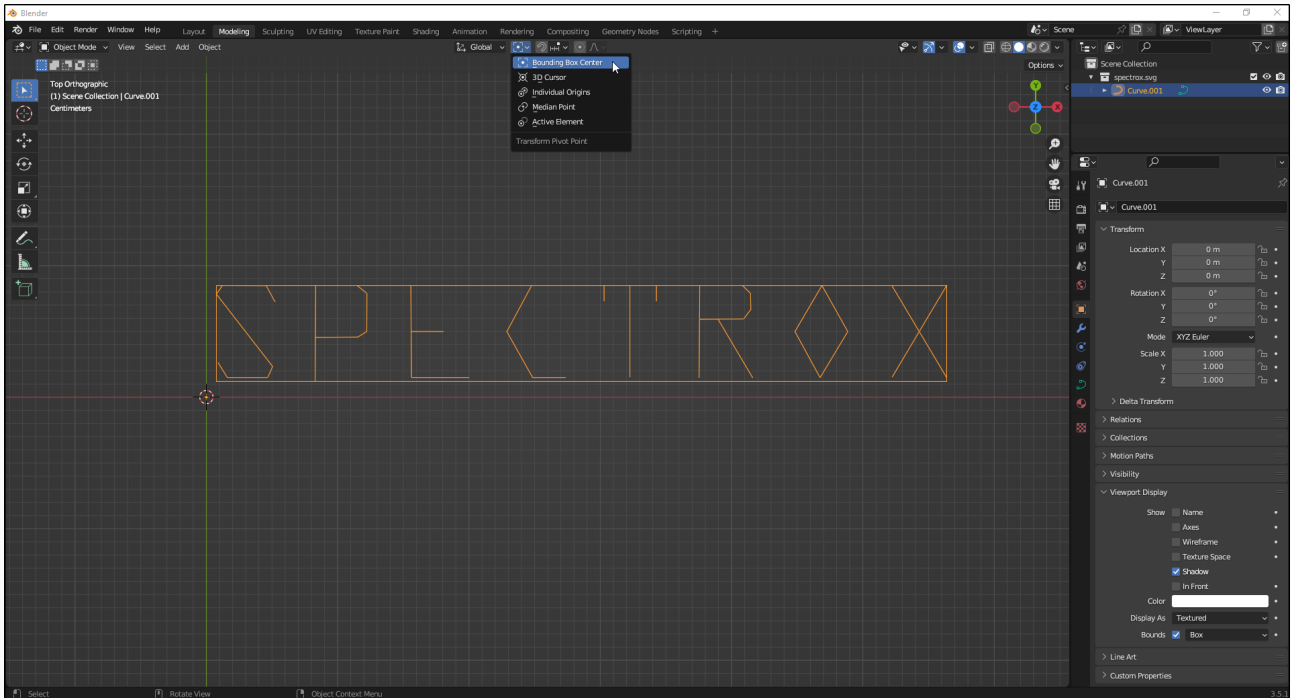


Figure 8: Transform Pivot Point.

Das Objekt selektieren und Object → Set Origin → Origin to Geometry ausführen:

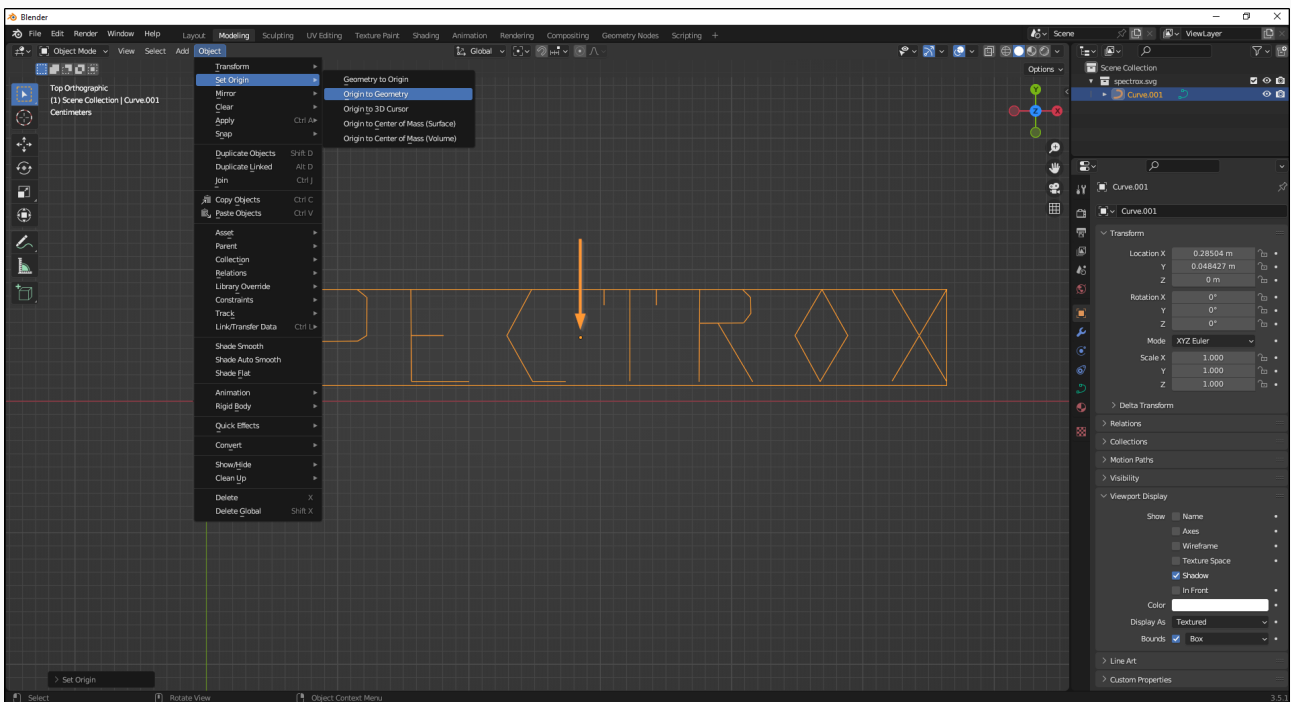


Figure 9: Objekt-Origin setzen. Es ist nun im Zentrum der Bounding-Box.

Jetzt kann das Objekt ins World-Origin verschoben werden. Das Objekt selektieren, mit **SHIFT + S** das Menü öffnen und **Selection to Cursor** wählen. Dies verschiebt das selektierte Objekt zum aktuellen 3D-Cursor, welcher sich standardmässig im World-Origin befindet:

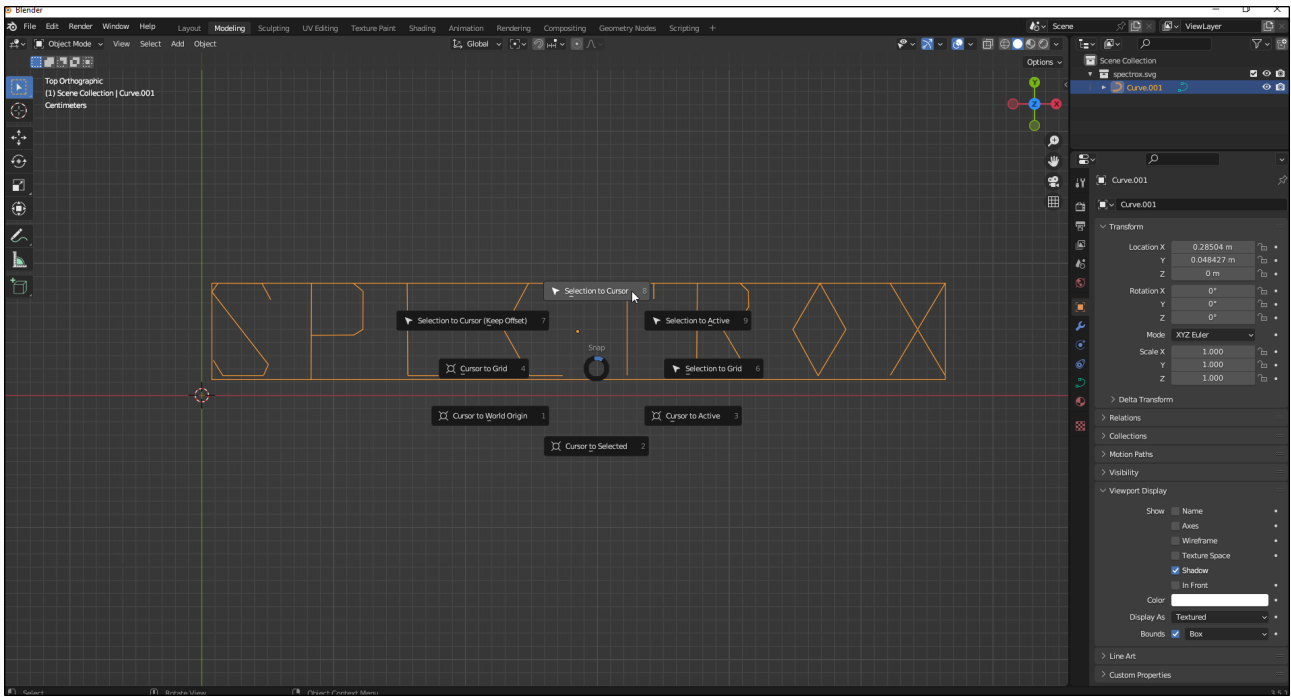


Figure 10: Verschieben zum World-Origin.

Wenn nötig, Fein-Justierungen vornehmen, indem das Objekt selektiert und mittels **G** in den Move-Modus gewechselt wird. Nun können die Cursor-Tasten verwendet werden, um das Objekt zu verschieben. Mit **Return** kann die Anpassung bestätigt und abgeschlossen werden.

Schliesslich kann das Objekt mittels **File → Export... → Wavefront (.obj)** in eine OBJ-Datei exportiert werden. Da im OBJ File-Format die Y- und Z-Achsen vertauscht sind, muss bei den Export-Einstellungen die **Forward Axis** auf **Y** gestellt werden. Die **Up Axis** wird dadurch automatisch auf **Z** gesetzt:

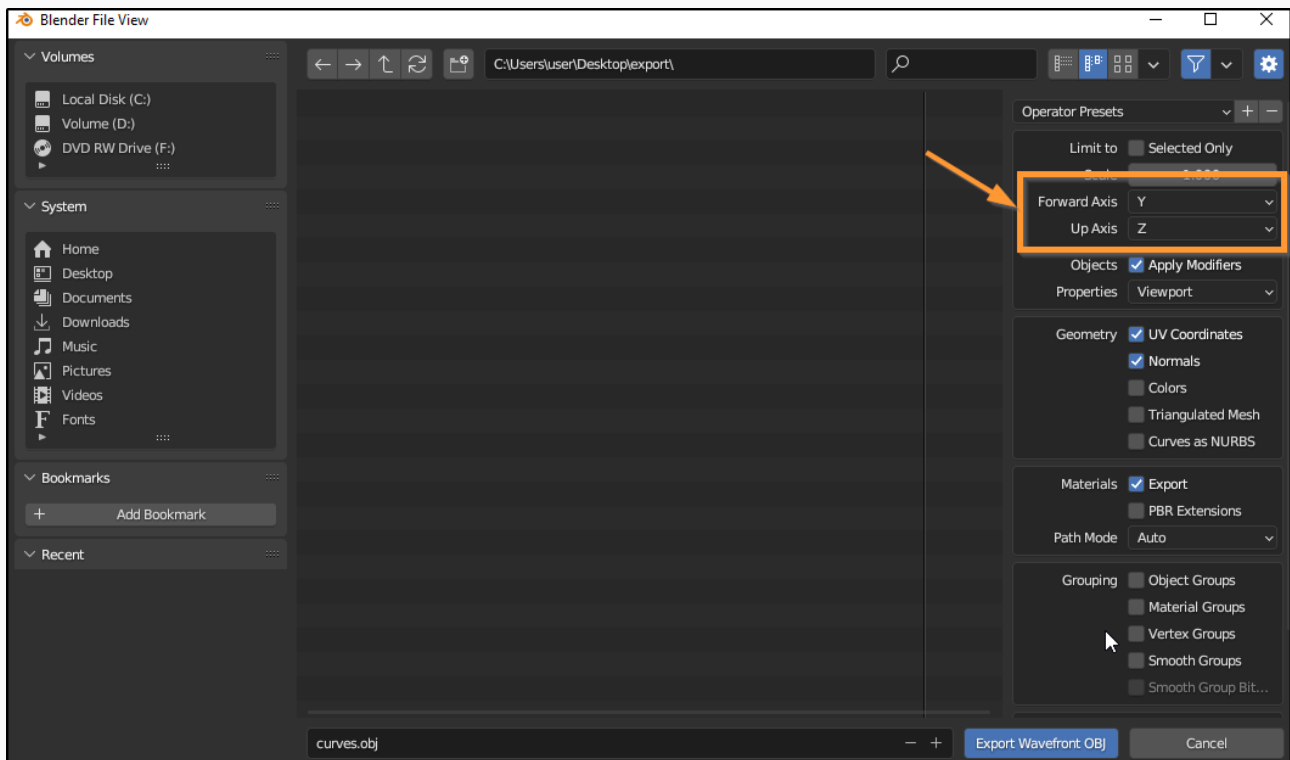


Figure 11: OBJ Export-Einstellungen.

Import als C-Array

Das exportierte OBJ-File besteht aus Vertex- und Line-Definitionen:

```
o Curve.001
v -0.233413 0.022883 0.000000
v -0.239969 0.034834 0.000000
v -0.273258 0.034962 0.000000
...
v 0.236503 -0.035073 0.000000
l 1 2
l 2 3
...
l 48 49
```

In Notepad++ sind die Vertex-Zeilen, diese starten mit v, mittels Search/Replace/Regex in die Form {x, y, z} zu bringen. Diese können als Vertex-Array des Typs `fw_vec3f` in den C-Code übernommen werden. Falls der Z-Wert 0 beträgt, kann dieser auch weggelassen werden:

```
fw_vec3f vertices[] = {
    {-0.233413,0.022883},
    {-0.239969,0.034834},
```



```
{-0.273258,0.034962},  
// ...  
{0.236503,-0.035073},  
};
```

Die Line-Zeilen starten mit 1 und werden auf diesselbe Weise in ein Array konvertiert.

Es ist zu beachten, dass im OBJ-Format die Vertices, auf welche sich die Lines beziehen, 1-based sind und nicht, wie im C-Array 0-based. Aus diesem Grund müssen alle Line-Indexes um 1 verringert werden.

Dies kann z.B. mit vim gemacht werden: Die Datei, welche die Line-Indexe enthält in vim öffnen und mit Doppelpunkt : in den Command-Line Modus wechseln und mittels der Regex alle Zahlen im File mit -1 verrechnen¹:

```
%s/\d\+/\=submatch(0)-1/g
```

Danach mittels :x die Änderungen speichern und vim verlassen.

Die so erhaltenen Zeilen können jetzt als fw_vec2i-Array in den C-Code übernommen werden:

```
fw_vec2i lines[] = {  
    {0,1},  
    {1,2},  
    {2,3},  
    // ...  
    {47,48},  
};
```

Alternativ kann auch ChatGPT alle Zahlen mit -1 verrechnen.

¹ <https://stackoverflow.com/a/10420969/891846>