



## mein erstes Makro, Teil 1

ein Tutorial von arcitool für ThouVis

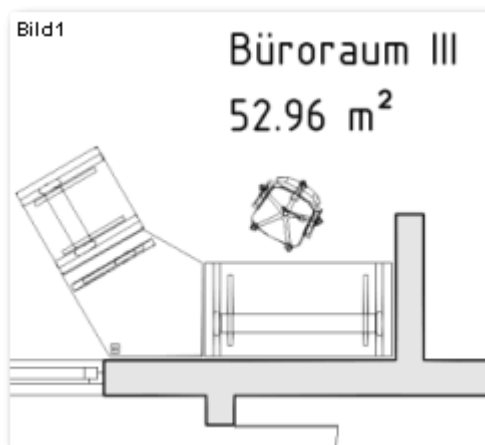
 [Klicken Sie hier, um dieses Tutorial als PDF-Datei zu laden](#) 

### Eine Vorgeschichte oder zurück zu 2D?

vor ein paar Wochen habe ich in einer Baufachzeitschrift gelesen, dass zwischen 70 und 80% der Zeichner, Architekten und Statiker ihr z. T. tausende Euro teures CAD-Programm nicht anders verwenden als zum zeichnen von Linien und Kreisen. Mit anderen Worten: das Zeichnbrett am Bildschirm.

Was ist los mit der seit Jahren versprochenen alleskönnenden Supersoftware? 'Objektorientiertes Planen und Ausschreiben in 3D' (mancher wirbt sogar mit 4D...🤔...). Was auf den Messen beeindruckend präsentiert wird, stellt sich allzuoft als zeitraubend, praxisfern oder unflexibel heraus. Das sogenannte 'Vorführsyndrom' scheint auf den Softwaremessen nicht zu funktionieren.

Bei mancher Software hat man den Eindruck, sie wird von Kaufleuten entwickelt. Sinnfreie Funktionen, kriecht von Leuten, die noch keine Baustelle betreten haben, ersetzen Pflichtaufgaben.



Ein Beispiel: Sie positionieren im Grundriss für den Bauantrag (1:100) einen Bürostuhl. Wenn Sie dicht genug ranzoomen, können Sie bei der Verschraubung der Armlehne die Anzahl der Windungen überprüfen. Bei Doppelklick auf den Stuhl erfahren Sie, wer ihn herstellen kann und was er bei Abnahme von mindestens 100 Stück im Sommerschlussverkauf kostet. Natürlich ist das gute Stück direkt nach der Positionierung mit Ihrer AVA-Software verbunden, bei der Sie aber momentan immer nur die mindestens 100 Stück ausschreiben können - Ein Update für diesen Programmfehler wird Ihnen seit 5 Monaten versprochen... der Softwareentwickler hat sich aber offensichtlich in anderen Details verloren oder entwickelt parallel einen Kinderbettdesigner für den ArchitekturProfi++ von DataSchlecker... OK, etwas überzogen, aber im Prinzip liege ich wohl gar nicht so

daneben!?

Was benötigen wir wirklich? Wir brauchen Pläne! Bauherren, Ämter und Baustellen brauchen schlicht Pläne... Die Entwicklung des objektorientierten, dreidimensionalen CAD kommt nicht wirklich voran. Immer wieder stoßen wir auf neue Hindernisse, die unsere Arbeit und unser vorankommen im Projekt ausbremsen. Lassen sie und die Entwicklung der 'Supersoftware' weiter beobachten und konzentrieren uns zwischenzeitlich auf das zeichnungsorientierte Planen, wie es ohnehin die oben genannten 70 bis 80% unserer Kollegen tun.

Neben den Linien, Kreisen und anderen Funktionen, die ThouVis vorzüglich beherrscht, brauchen wir Variantenkonstruktionen für wiederkehrende Aufgaben, Funktionen, die sich an vorgegebene Schemen und Normen gebunden sind, flexible, intelligent berechnete Mengen etc. Im Grunde weiß jeder Planer bestens, was er eigentlich benötigt, es fehlten ihm nur bisher die Mittel dies umzusetzen. Mit ThouVis 2.0 haben Sie dieses Mittel. Geben Sie sich einen Ruck und steigen Sie ein in die einfache Makroprogrammierung und bauen Sie gemeinsam mit uns ein brauchbares CAD!

---

## Lernen Sie das Programmieren!

Sie trauen sich das nicht zu? Hab' ich auch gesagt, als die Sache auf mich zukam... Kein Grund, das Ding nicht mal anzugehen, oder? Excel war ja auch irgendwann mal Neuland für Sie! Erinnern Sie sich an Ihr Studium... da gab es auch das eine oder andere, wo Sie bewundernd neben den 'Profis' standen, ein paar Wochen später standen Sie als Profi neben dem ahnungslosen Kommilitonen ;)

Mit ThouVis 2.0 wird die Makrosprache eingeführt. Damit haben Sie endlich die Möglichkeit, den Leistungsumfang Ihrer CAD-Software selbst zu bestimmen und fehlende Funktionen gemeinsam mit anderen auf einfache Weise zu entwickeln! Auf arcitool stehen bereits einige gute Beispiele bereit, die auf Ihre Erforschung warten. Wenn Sie einmal nicht weiter kommen, wird Ihnen in unserem Forum sicher gerne weiter geholfen.

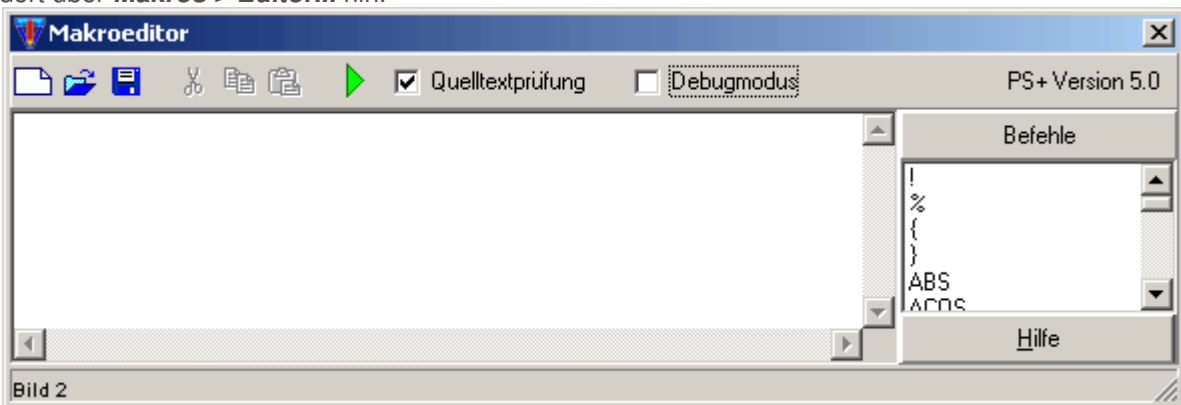
---

## Es geht los!

Sehr schön, Sie sind noch dabei ;) Sie wollen also programmieren. Programmieren heißt im Grunde nichts anderes, als dem Computer eine Reihe von Befehlen zu geben, die er nacheinander abarbeiten soll. Sie haben eine konkrete Vorstellung davon, was ein Programm leisten soll und formulieren es zunächst in Ihrer eigenen Sprache:

*Aufgabe:* zeichne eine 2-Meter lange Linie von A nach B und mache am Ende der Linie einen Kreis mit einem Radius von 50 cm.

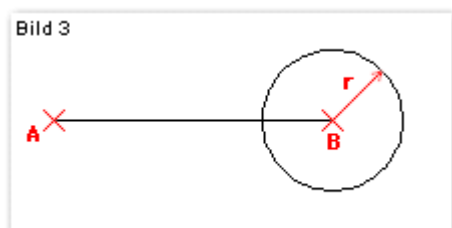
Wenn Sie diesen Text in ThouVis eintippen, wird sicherlich nicht allzu viel passieren, bestenfalls wird er als Text dargestellt, wenn Sie vorher den Textbutton ( **A** ) geklickt haben. Was haben wir falsch gemacht? Zunächst einmal brauchen wir eine Umgebung, die dafür vorgesehen ist, dass hier Programmtext eingegeben wird. In ThouVis heißt diese Umgebung Makroeditor. Sie gelangen dort über **Makros > Editor...** hin.



Der Makroeditor ist also der Bereich, in den wir unsere Befehle hineinschreiben können. Aber auch hier gibt es Probleme, wenn wir unsere Aufgabe wie oben beschrieben formulieren, warum? Nun, die Sprache, die wir verwendet haben ist wohl nicht die, die unser PC nachvollziehen kann... Wir benötigen also eine Sprache, die ThouVis versteht und die haben wir natürlich auch: PS+ ist der Name der Programmiersprache und wurde extra für ThouVis entwickelt. Mit zahlreichen mathematischen und grafischen Befehlen können fast alle denkbaren zeichnerischen Aufgaben gelöst werden.

Zeichnen wir unsere Aufgabe doch zunächst einmal direkt in ThouVis, ohne Programmierung. Das Ergebnis sollte dann etwa so (Bild 3) aussehen. Im Wesentlichen haben wir zwei Schritte getan, um zum gewünschten Ergebnis zu kommen:

1. Zeichnen einer Linie,
2. Zeichnen eines Kreises.



Wenn wir diese Aufgabe in einem Makro verarbeiten wollen, müssen wir die beiden Schritte lediglich in unsere Programmiersprache übersetzen und schon haben wir einen Quelltext für unser Makro. Wenn Sie sich die beiden Schritte aber einmal genauer ansehen, stecken dort schon noch ein paar mehr Informationen drin. Lassen Sie uns diese beiden Schritte einmal in alle Einzelinformationen zerlegen, die darin stecken:

1. lege einen Liniestil fest
2. lege eine Stiftstärke fest
3. definiere die x-Koordinate von Punkt A
4. definiere die y-Koordinate von Punkt A
5. definiere die x-Koordinate von Punkt B, wobei  $B_x = A_x + 2m$
6. definiere die y-Koordinate von Punkt B, wobei  $B_y = A_y$
7. gehe zu Punkt A
8. zeichne eine Linie von A nach B
9. zeichne einen Kreis mit Radius 50cm um B

All diese Informationen haben Sie beim direkten zeichnen auch vorgegeben. Wir haben sie diesmal nur konkret festgehalten. 9 Schritte, sehr fein gegliederte Schritte sind also erforderlich, um unser Gebilde zeichnen zu lassen.

---

## Übersetzen in die Programmiersprache

Wenn wir unsere Anforderung in die Programmiersprache oder in einzelne Befehle übersetzen wollen, müssen wir uns zunächst über ein paar grundlegende Regeln im Klaren sein:

### Parameter in Befehlen

Die meisten Befehle enthalten Parameter. Wenn Sie den Cursor (die aktuelle Mausposition) beispielsweise auf die x-Position 50 und die y-Position 120 bewegen möchten, lautet der Befehl:

```
50 120 moveto
```

Wenn Sie eine Linie vom gerade festgelegten Ausgangspunkt um 200 nach rechts bewegen möchten, lautet der Befehl:

```
200 0 lineto
```


Sie können die beiden Befehle auch in eine Zeile schreiben; das macht keinen Unterschied:

```
50 120 moveto 200 0 lineto
```

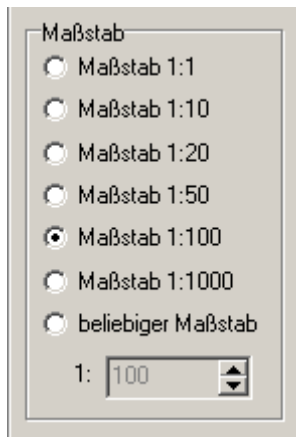
Der Makroeditor geht den Text 'Wort für Wort' durch und ordnet die Parameter automatisch den Befehlen zu. Wichtig ist eigentlich nur, dass Sie die Befehle und deren Parameter immer mit einem Leerzeichen trennen.

Quelltextprüfung

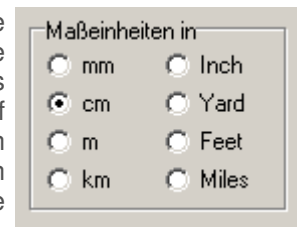
Wenn Sie im Makroeditor die Quelltextprüfung eingeschaltet haben, erkennt das Programm direkt, was ein Befehl und was ein Parameter ist. Befehle werden direkt in Fettschrift dargestellt, so dass Sie direkt für die korrekte Schreibweise eine Kontrollmöglichkeit haben.

Wenn Sie das Häkchen bei der Quelltextprüfung wegschalten, können Sie jederzeit mit der 'Play'-Taste eine Prüfung des Quelltextes durchführen. Bei sehr langen Quelltexten ist dies von Vorteil. Es wird nicht nach jedem Wort die Prüfung durchgeführt (was längere Wartezeiten mit sich führen kann), sondern abschnittsweise, nachdem Sie diese Funktion starten. 

## Einheiten, Maßstäbe etc.



Wenn Sie innerhalb Ihres Quelltextes keine besonderen Aussagen machen, werden die aktuellen Programmeinstellungen für das auszuführende Makro übernommen. Die auf den Bildern dargestellten Einstellungen werden auch für Ihre Parameter übernommen. In unserem Beispiel (50 120 **moveto**) würde quasi die Stiftposition um 50cm nach rechts und 120cm nach oben bewegt.



Für den Anfang soll uns diese Kenntnis ausreichen. Es sei aber noch erwähnt, dass Sie innerhalb Ihrer Makros auch auf diese Einstellungen einfluss nehmen können. Dazu später mehr...

### Kommentare im Quelltext

Damit Sie nach einiger Zeit noch nachvollziehen können, was Sie programmiert haben, ist es hilfreich, den Quelltext kommentieren zu können. ThouVis kennt hierfür den Befehl % :

```
50 120 moveto % positioniert den Cursor
```

oder:

```
% -----Positionierung des Cursors -----
50 120 moveto
% -----
```

Wie Sie sehen, werden diese Kommentare automatisch hellgrau dargestellt, um sie vom eigentlichen Programmiercode unterscheiden zu können. Wenn Sie nach einem Kommentartyp wieder mit neuen Parametern und Befehlen fortfahren möchten, müssen Sie eine neue Zeile beginnen.

Nachdem die Grundlagen nun gelegt sind, wollen wir wieder zu unserem kleinen Programm kommen. direkt in die Programmiersprache von ThouVis übersetzt würde es etwa so aussehen:

<i><b>Muttersprache</b></i>	<i><b>Makrosprache</b></i>
1. lege einen Linienstil fest	1 <b>setdash</b>
2. lege eine Stiftstärke fest	0.35 <b>findpen setpen</b>
3. definiere die x-Koordinate von Punkt A	/Ax 0 <b>def</b>
4. definiere die y-Koordinate von Punkt A	/Ay 0 <b>def</b>
5. definiere die x-Koordinate von Punkt B, wobei $B_x = A_x + 2m$	/Bx Ax 200 <b>add def</b>
6. definiere die y-Koordinate von Punkt B, wobei $B_y = A_y$	/By Ay <b>def</b>
7. gehe zu Punkt A	Ax Ay <b>moveto</b>
8. zeichne eine Linie von A nach B	Bx By <b>lineto</b>
9. zeichne einen Kreis mit Radius 50cm um B	50 <b>circle</b>

Das wars! Unser erstes Makro ist im Quelltext fertig und wir können es starten. Das Makro zeichnet unser gewünschtes Gebilde momentan immer an die Position, wo der Koordinaten-Nullpunkt sich gerade befindet. Ein weiteres Problem ist, dass die Zeichnungseinheit auf 'cm' eingestellt sein muss. Diese und weitere Punkte werden wir im zweiten Teil dieses Tutorials verbessern. Natürlich

werden wir auch etwas Flexibilität einbauen, das wird unserem kleinen Makro dann auch einen tieferen Sinn geben ;)

---

Ich hoffe, dass Sie ein wenig Interesse an der Makroprogrammierung gefunden haben und würde mich freuen, wenn Sie beim zweiten Teil wieder dabei sind!

Beste Grüße,

Ihr Kundenservice von 'arcitool' und **ThouVis**

---

### **Gemeinsam zum CAD für die Praxis**

Mit ThouVis haben Sie als Anwender die Möglichkeit, über Makros und einfache Makroprogrammierung Ihr eigenes CAD zusammen zu stellen. Arcitool bietet für diese zusätzlichen Funktionen die Plattform für den Austausch. Wenn auch Sie interessante Makros programmieren, beteiligen Sie sich auf 'arcitool' an der community. Hier können Sie Ihre Makros verkaufen oder kostenlos anderen Anwendern zur Verfügung stellen. Wir helfen Ihnen auch gerne bei der Gestaltung und freuen uns auf jedes Feedback!

---

[Startseite](#) | [Forum](#) | [AGB](#) | [Impressum](#) | [Hilfe/FAQ](#)

Copyright © 2006, arcitool, Inh. Dagmar Dobner. Alle Rechte vorbehalten.