



# EUKLID DynaGeo

## Internes

### Inhaltsverzeichnis:

1	Die Syntax von Termen .....	1
2	Die Standard-Funktionen für Terme .....	4
3	Details zu Makros .....	6
3.1	Was ist ein Makro? .....	6
3.2	Makro-Verwaltung: .....	6
3.3	Interne Details zu Start-Objekten.....	6
3.4	Genau planen und dokumentieren! .....	7
4	Grundsätzliches zu Animationen .....	7
4.1	Basispunkte als Steuerobjekte: .....	8
4.2	Zahlobjekte als Steuerobjekte: .....	8
4.3	Die Animations-Leiste.....	9
5	Die Werkzeugleisten.....	9

## 1 Die Syntax von Termen

DynaGeo hat einen integrierten **Term-Compiler**, der eine Vielzahl von Funktionen handhaben kann. Wenn Sie einen Term eingeben, wird er intern auf Korrektheit überprüft. Allerdings kann es selbst bei syntaktisch korrekten Termen zu **Fehlern** beim Berechnen des Term-Wertes kommen. Dies geschieht z.B. dann, wenn eine Funktion mit einem unzulässigen Argument aufgerufen wird. In einem solchen Fall wird der gesamte Term **ungültig**.

Bei der Eingabe von Termen können Sie alle gängigen mathematischen **Operationen** (+, -, \*, /, ^), eine Vielzahl **mathematischer Standard-Funktionen** sowie beliebige Schachtelungen von **Klammern** benutzen. Außerdem stehen Ihnen die **Konstanten** pi und e zur Verfügung.

Darüber hinaus können Sie auch Funktionen verwenden, welche **Abstände** und **Winkelgrößen** aus Ihrer Zeichnung auslesen. Um eine solche Funktion in den Term einzufügen, sollten Sie zuvor in Ihrer Zeichnung für das gewünschte Maß ein Maßobjekt erzeugt haben (siehe *Abstände messen* bzw. *Winkel messen*). Wenn Sie dann bei geöffnetem Term-Eingabefenster auf das Maßobjekt in Ihrer Zeichnung klicken, wird ein entsprechender Funktionsaufruf in Ihren Term eingefügt:

- "d(**Obj1; Obj2**)"

liefert den Abstand der Objekte Obj1 und Obj2.

*Beispiele:*

In einem Dreieck ABC sei c die Strecke [AB], dann stellt "d(A;B)" die Länge von c



dar, während "d(C; c)" den Abstand des Punktes C zur Seite c referenziert (genauer: den Abstand von C zu derjenigen Geraden, auf der die Seite c liegt).

- "**w(P1; P2; P3)**"

liefert das Gradmaß des Winkels, der durch die drei Punkte P1, P2 und P3 festgelegt ist.

*Beispiel:*

Ist *beta* der Winkel mit Scheitel B, dessen erster Schenkel durch C und dessen zweiter Schenkel durch A verläuft, dann ist "w(C; B; A)" das Winkelmaß (in Grad) von *beta*.

Weiter können Sie den **Wert eines Zahlobjekts** in den Term übernehmen, indem Sie den Namen des Zahlobjekts in den Term einfügen. Dies erreichen Sie bei geöffnetem Term-Eingabefenster am einfachsten durch einen Klick auf das entsprechende Zahlobjekt:

- "**ZObj**"

liefert den aktuellen Wert des Zahlobjekts ZObj.

*Beispiel:*

Ist ein Zahlobjekt Z1 in der Zeichnung vorhanden, dann bezeichnet der Term "Z1\*2" das Doppelte des aktuellen Werte von Z1.

Auf dieselbe Art und Weise kann in den Term eines Termobjekts eine Referenz auf den aktuellen **Wert eines anderen Termobjekts** eingefügt werden:

- "**TObj**"

liefert den aktuellen Wert des Termobjekts TObj. Da sich mit dieser Funktion Termobjekte gegenseitig aufeinander beziehen können, besteht hier die Gefahr zirkulärer Abhängigkeiten.

*Beispiel:*

Ist ein Termobjekt T1 in der Zeichnung vorhanden, dann bezeichnet der Term "T1^2" das Quadrat des aktuellen Wertes von T1. Wenn Sie diesen Term in ein Termobjekt T2 eingeben, dann müssen Sie selbst darauf achten, dass der Term von T1 keine Referenz auf T2 oder irgend ein von T2 abgeleitetes Objekt enthält. Andernfalls läuft das Programm in eine nicht abbrechende Rekursion, was unweigerlich zu einem Stacküberlauf führt.

Für den Zugriff auf **Punkt-Koordinaten, Kreisradien, Streckenlängen** und **Flächeninhalte** aus Ihrer Zeichnung gibt es derzeit noch kein "graphisches Interface". Daher müssen Sie die folgenden Funktionen jeweils selbst in Ihre Terme einfügen:

- **x(P)** und **y(P)**

liefern die x- bzw. die y-Koordinate des Punktes P.

*Beispiel:*

Ist ein Dreieck ABC gegeben, dann sind liefern die Terme "[x(A)+x(B)+x(C)]/3" und "[y(A)+y(B)+y(C)]/3" stets die Schwerpunkt-Koordinaten dieses Dreiecks.

- **radius(K)**

liefert den Radius des Kreises K.

*Beispiel:*



Ist der Kreis k1 in der Zeichnung vorhanden, dann übergibt "radius(k1)" seinen aktuellen Radius.

- **len(Obj)**

liefert die Länge einer einfachen endlichen Linie. Die folgende Tabelle zeigt die möglichen Typen für das Argument Obj und die Bedeutung des jeweils zurückgelieferten Wertes:

Obj ist:	len(Obj) liefert:
eine Strecke	die Streckenlänge
ein Vektor	den Betrag des Vektors
ein Kreis	den Umfang
ein Kreisbogen	die Bogenlänge
ein Polygon	den Umfang

*Beispiele:*

Ist die Strecke s1 in der Zeichnung vorhanden, dann übergibt "len(s1)" ihre aktuelle Länge. Ist k1 ein Kreis, dann liefert len(k1) den Umfang dieses Kreises.

- **area(N), area(K)**

liefert den Inhalt der vom Polygon N bzw. dem Kreis K umschlossenen Fläche. Im Falle der Polygonfläche wird jedoch vorausgesetzt, dass sich der Rand von N nicht selbst schneidet. Für solche "überschlagenen" Polygone liefert die area-Funktion falsche Werte.

*Beispiel:*

Ist das Dreieck N1 in der Zeichnung vorhanden, dann übergibt "area(N1)" seinen aktuellen Flächeninhalt.

- **tv(A; B; C)**

liefert das Teilverhältnis des Punktes A bezüglich des geordneten Punktpaares (B, C).

B ist dabei der Nullpunkt, C der Einstpunkt der auf der Geraden (BC) errichteten Teilverhältnis-Skala. Wenn A nicht auf (BC) liegt, wird statt A der Fußpunkt A' der Orthogonalen von A auf (BC) genommen, so dass das die tv-Funktion auch für nicht kollineare Punktetripel stets einen sinnvollen Wert liefert. Wenn B und C zusammenfallen, ist das Teilverhältnis nicht definiert; dann liefert tv den (falschen!) Wert 0.

*Beispiel:*

Ist das Dreieck ABC in der Zeichnung vorhanden, dann übergibt "tv(A; B; C)" das Teilverhältnis des Höhenfußpunkts Ha bezüglich der Ecken B und C.

- **if<bed>; <term1>; <term2>**

ist eine "Schalterfunktion": <bed> ist eine Bedingung, also ein boolescher Ausdruck, der entweder wahr oder falsch ist. Ist er wahr, dann wird als Funktionsergebnis der Wert von <term1> zurückgeliefert; ist die Bedingung aber nicht erfüllt, wird der Wert



von <term2> zurückgeliefert.

Die Bedingung kann dabei auch Terme umfassen, die Referenzen auf Objekte der Zeichnung enthalten. Eine typische Anwendung der Schalterfunktion ist der Einsatz zur Implementierung abschnittsweise definierter Funktionen.

**Beispiel:**

Sei S ein an die x-Achse gebundener Punkt und P ein Punkt mit den bestimmten Koordinaten

$$\begin{aligned}x &: x(S) \\y &: \text{if}(x(S)<0; 2-0.25*x^2; 2*\cos(\text{grad}(x)))\end{aligned}$$

Dann beschreibt die Ortslinie von P links von der y-Achse die Hälfte einer nach unten geöffneten Parabel, während sich rechts das Schaubild einer Kosinusfunktion anschließt.

Wenn Sie das Term-Eingabefenster mit *Okay* schließen, wird der Term auf syntaktische Korrektheit geprüft und im Erfolgsfall an das anfordernde Objekt übergeben. Enthält der Term noch Fehler, erhalten Sie eine entsprechende Meldung, und das Term-Eingabefenster bleibt geöffnet.

Bitte beachten Sie, dass es selbst bei einem syntaktisch korrekten Term zu Fehlern bei der späteren Termwert-Berechnung kommen kann. In diesem Fall wird der gesamte Term als ungültig deklariert.

## 2 Die Standard-Funktionen für Terme

Die folgende Liste enthält die mathematischen **Standard-Funktionen**, die in Termen verwendet werden können. Bei den verwendeten Funktionsbezeichnern kommt es nicht auf Groß- oder Kleinschreibung an. Alle trigonometrischen Funktionen arbeiten im Gradmaß.

Einige der Funktionen haben einen **beschränkten Definitionsbereich**, sei es **aus mathematischen Gründen** (wie z.B. bei der ARCSIN- und ARCCOS-Funktion, denen nur Werte aus dem Intervall [-1; 1] übergeben werden dürfen), sei es **aus maschinen-internen Gründen** (wie z.B. bei den Rundungsfunktionen RND, INT, FLOOR und CEIL, deren Argumente stets im Bereich der darstellbaren INTEGER-Zahlen liegen müssen). Wird eine Funktion mit einem ungültigen Argument aufgerufen, dann wird der ganze Term ungültig.

- **ABS():**

liefert den Absolutbetrag des übergebenen Arguments.

*Beispiele:* ABS(-2.5) liefert 2.5, ABS(0.5) liefert 0.5, ABS(0) liefert 0.

- **SQRT() und SQR():**

berechnet die Quadratwurzel bzw. das Quadrat des übergebenen Arguments.

*Beispiele:*

SQRT(9) liefert 3, SQRT(pi) liefert die Wurzel aus Pi, SQR(d(P,Q)) liefert das Quadrat des Abstandes der Punkte P und Q.

- **EXP(), LN() und LOG():**

liefert die Werte der natürlichen Exponentialfunktion, der natürlichen Logarithmusfunktion und den Logarithmus zur Basis 10.



*Beispiele:*

EXP(1) liefert e (=2,718281...), LN(e) liefert 1, LOG(1000) liefert 3.

• **SIN(), COS() und TAN():**

berechnet die Werte der Winkelfunktionen; das Argument muss dabei im Gradmaß angegeben werden.

*Beispiele:*

SIN(30) liefert 0.5, COS(180) liefert -1, TAN(w(A;B;C)) liefert den Tangens des Winkels, der durch die drei Punkte A, B und C festgelegt ist, wobei B der Scheitel ist.

• **ARCTAN(), ARCSIN() und ARCCOS():**

berechnet die Hauptwerte der Umkehrfunktionen zu den vorgenannten trigonometrischen Funktionen; der gelieferte Funktionswert ist stets im Gradmaß zu interpretieren.

*Beispiele:*

ARCTAN(1) liefert 45 (Grad!), ARCSIN(0.5) liefert 30, ARCCOS(0.5) liefert 60.

• **BOGEN() und GRAD():**

ermöglichen die Konversion von Winkelmaßen vom Grad- ins Bogenmaß und zurück.

*Beispiele:*

BOGEN(90) liefert Pi/2, GRAD(PI/3) liefert 60.

• **SGN():**

liefert 1, 0 oder -1, je nach dem Vorzeichen des übergebenen Arguments.

*Beispiele:*

SGN(3) liefert 1, SGN(-PI) liefert -1, SGN(0) liefert 0;

ist der Betrag des übergebenen Arguments kleiner als die interne

Rechengenauigkeit (voreingestellter Wert: 1E-8), dann wird das Argument als Null interpretiert.

• **RND():**

rundet das übergebene Argument zur nächstgelegenen ganzen Zahl; liegt das Argument genau zwischen zwei ganzen Zahlen, so wird die betragsgrößere als Ergebnis zurückgegeben.

*Beispiele:*

RND(3.6) liefert 4, RND(-3.6) liefert -4, RND(-3.2) liefert -3;

RND(2.5) liefert 3, RND(-2.5) liefert -3.

• **INT() und FRAC():**

diese Funktionen dienen zum Zerlegen einer Fließkommazahl in einen ganzzahligen und einen echt gebrochenen Anteil; dabei schneidet INT den Nachkomma-Anteil des Arguments immer zur Null hin ab, während FRAC stets den Wert  $\{x - \text{INT}(x)\}$  hat.

*Beispiele:*

INT(3.6) liefert 3, FRAC(3.6) liefert 0.6; INT(-3.6) liefert -3, FRAC(-3.6) liefert -0.6.

• **FLOOR() und CEIL():**

dies sind Funktionen, die eine dem übergebenen Argument benachbarte ganze Zahl zurückgeben; FLOOR implementiert die "Gauß-Klammer"-Funktion ("Größte-Ganze"-Funktion), welche das Argument immer nach unten abrundet; CEIL



implementiert das entsprechende Gegenstück ("Kleinste-Ganze"-Funktion) und rundet stets nach oben auf.

#### **Beispiele:**

FLOOR(3.7) liefert 3, FLOOR(-3.2) liefert -4,  
CEIL (3.1) liefert 4, CEIL (-3.9) liefert -3.

## 3 Details zu Makros

### 3.1 Was ist ein Makro?

Ein Makro ist eine **Zusammenfassung mehrerer Konstruktionsbefehle** zu einem neuen "Meta-Befehl". Um ein Makro zu erstellen, müssen Sie die Konstruktion, die das Makro ausführen soll, zunächst einmal vollständig in Ihrer aktuellen Zeichnung durchführen. Erst dann können Sie das Makro "aufzeichnen", indem Sie zunächst die vollständige Reihe aller "Startobjekte" angeben, also alle Objekte, von denen die zu erzeugenden abhängen, und sodann die "Zielobjekte", also eben diese zu erzeugenden Objekte.

### 3.2 Makro-Verwaltung:

Makros gehören nach ihrer Erzeugung zu der Zeichnung, in der sie definiert wurden, und werden mit dieser Zeichnung abgespeichert. Wenn Sie die Zeichnung später erneut laden, stehen die Makros ebenfalls wieder zur Verfügung. Damit Sie aber ein solches Makro auch in einer **anderen** Zeichnung benutzen können, müssen Sie es aus der aktuellen Zeichnung exportieren und in die andere Zeichnung importieren. Beim Export wird das Makro in eine Datei mit der Kennung ".mak" geschrieben. Es ist zwar nicht (mehr) unbedingt nötig, jedes Makro in eine eigene ".mak"-Datei zu schreiben, weil es ja mit der Zeichnung abgespeichert wird. Gleichwohl ist es empfehlenswert, dies trotzdem zu tun, um nämlich die erstellten Makros für **alle** Zeichnungen verfügbar zu machen.

### 3.3 Interne Details zu Start-Objekten

Auf Wunsch einiger EUKLID DynaGeo - Anwender wurde eine Automatik in die Makroaufnahme eingebaut, die dazu führt, dass mit einer **Strecke** als Startobjekt auch ihre **Endpunkte** im Makro verfügbar werden. Es werden also die Streckenendpunkte als **implizite Startobjekte** mitregistriert. Diese auf den ersten Blick komfortabel erscheinende Erweiterung erweist sich jedoch manchmal als unpraktisch, weil der Anwender dann zur Laufzeit des Makros die **Orientierung** dieser Startstrecke nicht mehr kontrollieren kann. In allen Fällen, in denen es Ihnen also auf die Orientierung einer Startstrecke ankommt, sollten Sie auch weiterhin als Startobjekte statt einer Strecke ihre Endpunkte verwenden. Weitere implizite Startobjekte sind für Kreise die jeweiligen Mittelpunkte und für Polygone alle Eckpunkte. Darüberhinaus kennt jedes Makro als generelle implizite Startobjekte die Achsen und den Ursprung des Koordinatensystems.

Alle geometrischen Objekte, die Sie als "Eltern" in Konstruktionen verwenden, können auch in Makros verwendet werden. Damit sind **Ortslinien** prinzipiell von der Verwendung in Makros ausgeschlossen. Bei der Verwendung von **Polygonen** müssen Sie darauf achten, dass die **Anzahl der Ecken** zur Laufzeit des Makros mit der zur Designzeit vorliegenden Eckenzahl übereinstimmt. Andernfalls erhalten Sie eine Fehlermeldung, und die Makroausführung wird abgebrochen.

Beachten Sie, dass **jedes Zielobjekt** sich vollständig aus den jeweils **angegebenen Startobjekten** ableiten lassen muss. Dies ist speziell beim Einsatz von Kreisen mit bestimmtem Radius oder Geraden in bestimmtem Winkel zu bedenken, denn die **Terme**



für Radien oder Winkel können durchaus Bezüge auf andere Objekte beinhalten, an die Sie zunächst nicht gedacht haben.

Die Typen der Startobjekte, die Sie zur Entwurfszeit eines Makros wählen, legen die zulässigen Typen der Startobjekte zur Laufzeit fest, und zwar gemäß der folgenden Tabelle:

Startobjekt zur Entwurfszeit	Mögliche Startobjekt-Typen zur Laufzeit
ein Punkt	alle Punkt-Typen
eine Strecke	alle Strecken-Typen
eine Gerade	alle Geraden- und Strecken-Typen
ein Kreis	alle Kreis-Typen

Bitte beachten Sie dabei besonders die Behandlung von Strecken und Geraden:  
Wenn Sie wollen, dass der Benutzer Ihres Makros für ein bestimmtes Startobjekt **eine Strecke oder eine Gerade** eingaben kann, dann müssen Sie zur Entwurfszeit für dieses Startobjekt **eine Gerade** benutzen; nehmen Sie zur Entwurfszeit eine Strecke, dann kann der Benutzer zur Laufzeit auch nur eine Strecke verwenden!  
(Mit anderen Worte: in der Typklasse der Geraden ist die Typklasse der Strecken enthalten.)

### 3.4 Genau planen und dokumentieren!

Wenn Sie ein Makro programmieren, sollten Sie mit **viel Umsicht** zu Werke gehen. Selbst dann kann es passieren, dass beim Ablauf des Makros nicht genau die Objekte erzeugt werden, die Sie sich vorgestellt hatten. Prüfen Sie in diesem Fall genau, ob die **Randbedingungen** zur Ablaufzeitzeit hinreichend genau mit denen zur Aufnahmezeit übereinstimmen.

Planen Sie beim Entwurf eines Makros die relativen **Orientierungen** von Start- und Zielobjekten bewusst mit ein. Falls mehrere Startobjekte gleichen Typs direkt nacheinander einzugeben sind, sollten Sie dem Benutzer im **Hilfetext** des Makros auch erklären, welchen Einfluss die Reihenfolge der Startobjekte auf die Lage der Zielobjekte hat.

## 4 Grundsätzliches zu Animationen

Eine **Amination** ist eine **kontrollierte Veränderung** Ihrer Zeichnung **in der Zeit**. Sie wird durch den Benutzer gestartet, läuft dann aber selbstständig ab. Sie kann damit als eine **Automatisierung des Zugmodus** angesehen werden. Die Rolle des von der Maus gezogenen Objekts übernimmt nun ein sogenanntes **Steuerobjekt**: in beiden Fällen werden die vom bewegten Objekt abhängigen Objekte ihrerseits entsprechend mitbewegt.

Wenn Sie also eine Zeichnung animieren wollen, müssen Sie sie so konstruieren, dass die zu bewegenden Objekte von einem **Steuerobjekt** abstammen. Allerdings kommen als Steuerobjekte nur solche Objekte in Frage, die vom Programm intern kontrolliert verändert



werden können. Es gibt derzeit in DynaGeo zwei Arten von Steuerobjekten für Animationen:

1. **Basispunkte**, die an Strecken, Kreise oder Polygone gebunden sind
2. **Zahlobjekte**

Beachten Sie, dass Basispunkte, die an Geraden oder Ortslinien gebunden sind, nicht als Steuerobjekte geeignet sind. Der Ort des gebundenen Basispunktes auf seiner Trägerlinie wird nämlich intern durch einen **Positions-Parameter** beschrieben, der für eine Animation auf ein **abgeschlossenes Intervall** eingeschränkt sein muss. Dies ist bei Linien endlicher Länge wie Strecken und Kreisen kein Problem, während Geraden als Trägerlinien im Hinblick auf Animationen quasi "zu groß" sind. Und im Falle von Ortslinien mahnt die schiere Vorsicht zur Zurückhaltung: ob eine Ortslinie für alle künftigen Konfigurationen Ihrer Zeichnung stets von beschränkter Länge sein wird, ist numerisch kaum sicher zu entscheiden.

#### 4.1 Basispunkte als Steuerobjekte:

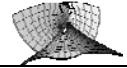
Wenn Sie nun eine Animation mit einem Basispunkt S als Steuerobjekt ablaufen lassen wollen, müssen Sie zunächst diesen Punkt S zum Steuerobjekt ernennen: im Dialogfenster *Animations-Parameter* wird Ihnen eine **Liste möglicher Steuerobjekte** angeboten. Ist der Punkt S an eine geeignete Linie gebunden, wird er in dieser Liste angezeigt, und Sie können ihn **auswählen**. Nach dem Schließen des Dialogs erscheinen auf der Werkzeugeiste "Animation" die Schaltflächen für die **Navigationsbefehle**: mit deren Hilfe können Sie die Animation starten und wieder anhalten.

Wenn Sie z.B. auf den grünen Pfeil nach rechts klicken, startet die Animation. Solange sie läuft, wird "einfach" der obengenannte Positions-Parameter des Punktes S intern um eine bestimmte **Schrittweite** verändert, die neue Position von S errechnet und S dorthin verschoben; und schließlich werden alle von S abhängigen Objekte ebenfalls an ihre entsprechenden neuen Positionen gebracht. Danach wird der Parameter wieder um die Schrittweite verändert usw...

Normalerweise stoppt die Animation automatisch, wenn der Parameterwert den Rand des vorgegebenen Intervalls erreicht. Wenn das Steuerobjekt jedoch ein Basispunkt ist, der an eine **geschlossene Linie** gebunden ist, dann wird das Parameter-Intervall intern ebenfalls "geschlossen" gedacht: wenn das eine Ende erreicht wird, erfolgt automatisch ein Rück-Sprung zum anderen Ende. Dies führt zu einer **Endlos-Animation**, die nur noch durch einen Klick auf die "Stop-Taste" angehalten werden kann.

#### 4.2 Zahlobjekte als Steuerobjekte:

Die zweite Sorte von Steuerobjekten sind die **Zahlobjekte**. Bei diesen übernimmt der Zahlwert selbst die Rolle des obengenannten Positionsparamters. Wenn Sie das Zahlobjekt Z1 als Steuerobjekt für eine Animation benutzen wollen, müssen Sie die zu bewegenden Objekte in Abhängigkeit von Z1 konstruieren. Dies ist zum Beispiel möglich, indem Sie **Objekte mit numerischen Parametern** verwenden (Punkte mit bestimmten Koordinaten, Kreise mit bestimmtem Radius und Geraden in bestimmtem Winkel). Im Gegensatz zum Fall der Basispunkt-Steuerobjekte können Sie bei den Zahlobjekten die **Grenzen des Parameter-Intervalls** frei wählen: im Dialog "Animations-Parameter" sind die entsprechenden Felder nun editierbar. Ansonsten funktioniert eine durch ein Zahlobjekt gesteuerte Animation genau wie oben beschrieben.



Die **Geschwindigkeit der Animation** ist direkt von der **Schrittweite** abhängig, mit der der Positionsparameter verändert wird: große Schrittweiten führen zu einem schnellen Ablauf, kleine zu einem langsameren. Auch die Schrittweite können Sie im Dialog "Animations-Parameter" einstellen. Beachten Sie aber, dass die Animation nicht "echtzeit-gesteuert" läuft: dieselbe Schrittweite kann in zwei verschiedenen Zeichnungen zu sehr unterschiedlichen Ablaufgeschwindigkeiten führen, wenn diese sich in Größe (d.h. Anzahl der Objekte) oder Komplexität (d.h. Art der Objekte) deutlich voneinander unterscheiden. Ebenso kann die Ablaufgeschwindigkeit von Computer zu Computer variieren, da der konkrete Zeitbedarf für einen Animationsschritt eben auch von der Leistungsfähigkeit des verwendeten Prozessors abhängt. Gegebenenfalls können Sie den Wert für die Schrittweite zur Laufzeit der Animation mit Hilfe der "Plus"- und "Minus"-Tasten im numerischen Ziffernblock Ihrer Tastatur anpassen.

### 4.3 Die Animations-Leiste

Diese Leiste ist nur dann sichtbar, wenn Ihre Zeichnung geeignete Animations-Steuerobjekte enthält. Andernfalls wird die Leiste komplett verborgen. Aber auch wenn sie angezeigt wird, kann sie eventuell zunächst nur den ersten der folgenden Befehle zeigen:



Animations-Parameter einstellen

Ein Klick auf diesen Knopf führt Sie zu einem Fenster, in dem Sie das Steuer-Objekt für die Animation wählen und den Startwert der Ablauf-Geschwindigkeit beeinflussen können.

Nachdem Sie ein Steuer-Objekt für Ihre Animation gewählt haben, werden zusätzlich die folgenden **Navigations-Befehle** verfügbar:



Diese Befehle arbeiten analog zu den Tasten eines Cassetten- oder Video-Recorders: mit den beiden grünen Pfeilen starten Sie die Animation (rückwärts oder vorwärts), ein Klick auf den mittleren (roten) Knopf hält sie wieder an, und mit den beiden äußeren Knöpfen springen Sie ganz an den Anfang bzw. an das Ende der Animation.

Zur Laufzeit der Animation können Sie den Ablauf durch (kurzes!) Drücken der "Plus"- und "Minus"-Tasten im numerischen Ziffernblock Ihrer Tastatur **beschleunigen** oder **verlangsamen**; die "Mal"-Taste kehrt die Bewegungsrichtung der Animation um.

## 5 Die Werkzeugeleisten

Um den Zugriff auf häufig gebrauchte Befehle zu erleichtern, verfügt EUKLID DynaGeo über eine Werkzeugeleiste: unter dem Hauptmenü erscheint eine Zeile mit kleinen Symbolbildern. Klickt man ein solches **Symbol** an, dann wird der zugehörige **Befehl** ausgeführt, gerade so, als wäre er aus dem entsprechenden Menü gewählt worden. Es gibt 6 verschiedene **Werkzeugpaletten**, zwischen denen Sie durch Klicken auf die Reiter am unteren Rand der Werkzeugeleiste umschalten können:

- **Die Hauptleiste** enthält die wichtigsten Befehle aus den Menüs *Datei* und *Bearbeiten*, sowie den Befehl zum Erzeugen von Ortslinien.
- **die Konstruieren-Leiste** enthält Befehle aus dem Menü Konstruieren;
- **die Abbilden-Leiste** enthält Befehle aus dem Menü Abbilden;



- **die Form-&-Farbe-Leiste** enthält Hilfsmittel zum Editieren der Zeichnung. Für ihre einzelnen Symbole gibt es keine direkt entsprechenden Menübefehle;
- **die Messen-&-Rechnen-Leiste** enthält diejenigen Befehle, die sich auf Koordinaten, Maße und Terme beziehen.
- **die Animations-Leiste** enthält Befehle, mit denen Sie Animationen Ihrer Zeichnung steuern können. Diese Leiste ist allerdings nur sichtbar, wenn Ihre Zeichnung für Animationen geeignet ist.

---

▪ **Freier Punkt:**  
Ein Klick mit der linken Maustaste auf eine beliebige Stelle des Zeichenblattes erzeugt einen frei beweglichen Punkt.

▪ **Punkt an Linie binden:**  
Klicke mit der Maus diejenige Linie an, auf der ein Punkt erzeugt werden soll. Mögliche Trägerlinien für den Punkt sind:

- Gerade Linien (also: Geraden, Halbgeraden und Strecken)
- Kreise (auch: Kreisbögen)
- Polygone
- Ortslinien

▪ **Schnittpunkt(e) zweier Linien:**  
Klicke mit der linken Maustaste die zwei Linien (Strecken, Geraden oder Kreise) an, die du miteinander schneiden willst.

▪ **Mittelpunkt:**  
Klicke die beiden Punkte an, zwischen denen du die Mitte suchst.  
Oder: Klicke auf die Strecke, deren Mittelpunkt du suchst.

▪ **Strecke zwischen zwei Punkten:**  
Klicke nacheinander auf den Anfangs- und den Endpunkt der gewünschten Strecke.

▪ **Vektor:**  
Klicke nacheinander auf den Anfangs- und den Endpunkt des gewünschten Vektors.

▪ **Gerade durch zwei Punkte:**  
Klicke nacheinander auf zwei Punkte, durch die die Gerade verlaufen soll.

▪ **Gerade in bestimmten Winkel:**  
Dieser Menüpunkt dient zum Zeichnen einer Geraden, die mit einer gegebenen Geraden oder Strecke (im Folgenden als Bezugslinie bezeichnet) einen von Ihnen festgelegten Winkel einschließt. Dies geschieht in den folgenden Schritten:

1. Zunächst öffnet sich ein kleines Fenster, in dem die gewünschte Winkelweite (in Grad) eingegeben werden muss.
2. Sodann wird ein Punkt auf der Bezugslinie angeklickt, durch den die zu erzeugende Gerade nicht verlaufen soll.
3. Als nächstes wird ein Punkt der Bezugslinie angeklickt, durch den die zu zeichnende Gerade verlaufen soll.
4. Falls eine konstante Winkelweite ohne Vorzeichen eingegeben wurde, erwartet EUKLID DynaGeo noch die Eingabe eines weiteren Punktes, der bestimmt, auf welcher Seite der Bezugslinie der Winkel angetragen werden soll.

▪ **Lot bzw. Senkrechte:**  
Klicke zunächst auf den Punkt, durch den die zu erzeugende Senkrechte gehen soll, und dann auf die Gerade (oder Strecke), zu der die neue Gerade senkrecht sein soll.

▪ **Parallele:**  
Klicke zunächst auf den Punkt, durch den die zu erzeugende Parallelle gehen soll, und dann auf die Gerade (oder Strecke), zu der die neue Gerade parallel sein soll.

▪ **Mittelsenkrechte:**  
Klicke eine Strecke an zu der die Mittelsenkrechte erzeugt werden soll.  
Oder: Klicke zwei Punkte an, zu denen die Symmetriechse erzeugt werden soll.

▪ **Winkelhalbierende:**  
Klicke drei Punkte an, die den Winkel festlegen, dessen Halbierende du konstruieren möchtest.  
Halte dich dabei an folgende Reihenfolge:



- 
1. Punkt auf dem ersten Schenkel
  2. Scheitelpunkt
  3. Punkt auf dem zweiten Schenkel
- 

**Kreis um Mittelpunkt durch Kreispunkt:**

Klicke zuerst auf den Mittelpunkt und dann auf einen Punkt, durch den die Kreislinie verlaufen soll.

---

**Kreis mit bestimmtem Radius:**

In das sich öffnende Fenster den gewünschten Radius des Kreises eingeben und anschließend den Mittelpunkt des Kreises anklicken.

---

**N-Eck:**

Klicke nacheinander die Eckpunkte des gewünschten N-Ecks an. Zum schließen des N-Ecks muss zum Schluss noch einmal der erste Eckpunkt angeklickt werden.

---

**Objekt an Achse spiegeln:**

Klicke das abzubildende Objekt und die Spiegelachse an.

---

**Objekt an Punkt spiegeln:**

Klicke das abzubildende Objekt und das Symmetriezentrum an.

---

**Objekt verschieben:**

Klicke das abzubildende Objekt und den Verschiebungsvektor an.

---

**Objekt drehen:**

Klicke das abzubildende Objekt, das Drehzentrum und einen Kreisbogen (für den Drehwinkel) an.

---

**Objekt zentrisch strecken:**

Klicke das abzubildende Objekt, das Streckzentrum und ein Zahl- bzw. Term-Objekt an, das den Streckfaktor enthält.

---

**Punkt an einem Kreis spiegeln:**

Klicke den abzubildenden Punkt und den Kreis an, an dem gespiegelt werden soll.

---

**Ortslinie eines Punktes aufzeichnen:**

Klicke zunächst mit der Maus denjenigen Punkt an, dessen Ortslinie erzeugt werden soll ("Generatorpunkt"). Danach kannst du an einem Basispunkt (dem "Zugpunkt" der Ortslinie) ziehen, von dem der Generatorpunkt abhängt. Wenn der Generatorpunkt dabei bewegt wird, hinterlässt er eine Serie von Punkten, die seine alten Positionen markieren. Sobald du nach dem Verziehen des Zugpunktes die linke Maustaste wieder loslässt, wird die Ortslinie als (möglichst) glatte Kurve durch die aufgezeichneten Punkte dargestellt.

---

**Objekt löschen:**

Klicke das zu löscheinde Objekt an.

---

**Abstand messen:**

Klicke die beiden Objekte an, deren Abstand du messen willst. Hierbei sind alle Punkte, Geraden und Kreise in beliebiger Kombination zugelassen.

---

**Winkel messen:**

Klicke einen Winkelbogen an. Falls der zu messenden Winkel noch nicht markiert ist, müssen die drei definierenden Punkte in folgender Reihenfolge angeklickt werden:

1. Punkt auf dem ersten Schenkel
  2. Scheitelpunkt des Winkels
  3. Punkt auf dem zweiten Schenkel
- 

**Speichern unter:**

Es wird ein Dialogfenster geöffnet, in dem ein gültiger Dateiname eingeben werden muss. Außerdem muss der Ordner festgelegt werden, in den die EUKLID DynaGeo-Datei gespeichert werden soll.

---

**Animation starten:**

Hiermit wird eine Animation gestartet.

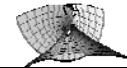
Zur Laufzeit der Animation kann der Ablauf durch (kurzes!) Drücken der "Plus"- und "Minus"-Tasten im numerischen Ziffernblock der Tastatur beschleunigt oder abgebremst werden. Die "Mal"-Taste kehrt die Bewegungsrichtung der Animation um.

---

**Textbox einfügen:**

Es wird ein Fenster geöffnet, in das ein beliebiger Text eingeben werden kann.

---



### Zahlobjekt erzeugen:

Ein Zahl-Objekt ist ein Schieberegler, mit dessen Hilfe eine variable Zahl aus einem vorgegebenen Intervall zur Verfügung steht, die in der Zeichnung weiterverwenden werden kann.  
Wenn dieser Befehl ausgeführt wird, wird ein Zahlobjekt mit Standard-Daten erzeugt und automatisch an einer freien Stelle der Zeichnung angezeigt. In der oberen Hälfte des Objekts werden die untere Intervallgrenze, der aktuelle Zahlwert und die obere Intervallgrenze angezeigt. Darunter erscheint der eigentliche Schieberegler zur Veränderung des aktuellen Wertes. Die Werte können über das Kontext-Menü (rechter Mausklick) eingestellt werden.



### Termobjekt erzeugen:

Ein Term-Objekt ist ein Objekt, das in einem Rahmen zwei Textzeilen auf dem Bildschirm darstellen kann: in der ersten Zeile wird ein von Ihnen eingegebener Term angezeigt, in der zweiten Zeile sein jeweils aktueller Wert.  
Wenn der Befehl ausgeführt wird, erscheint ein Term-Eingabefenster. Nachdem der Term vollständig eingegeben und das Fenster geschlossen wurde, wird das Termobjekt erzeugt und automatisch an einer freien Stelle der Zeichnung angezeigt



### Makro ausführen:

Ein (das letzte) Makro wird ausgeführt.