式ライン説明書

はじめに

StoneyDesigner で数式でポリラインを生成するプログラムです。

.NetFramework3.5 以上の環境で動作します。

実行プログラム名

ExprLine.exe	実行プログラム
ExprLn.dll	補助ライブラリ

インストールとアンインストール

インストールの特別な操作はありません。ExprLine.exe と ExprLn.dll を適当なフォルダに置き、ExprLine.exe をダブルクリックなどで起動してご使用ください。レジストリ等の変更はしないので、ファイルを削除すれば、アンインストールになります。

機能の概要

StoneyDesigner 上で数式によりポリラインを生成します。

ポリラインを表すために、t を変数とする式を三つの座標要素に記述します。

操作項目

座標モード

式が表す座標系を選択します。通常は一番上にある直交座標を使用します。

座標定数

一部の座標モードで座標変換時に使用される定数です。通常は 1.0 に設定しておきます。

乱数種

乱数を与える関数 RAND()の元になる乱数系列を決めます。値を 0 にすると、実行するたびに違う乱数系列を使用します。

式

各座標成分に与える数式を入力します。式に基づいて計算した座標成分を座標モードにしたがって変換して頂点座標にします。

式の中では、現在の頂点座標などは特定の変数に代入されています。

t	元の頂点の座標
i	頂点に対応する 0 から始まる序数
a,b,c,d,e,f	式記述を簡略するために使用する変数です。処理に先立って、ダイアログのそれぞれのテキストボックスから値を読み取ってセットします。

変数

式の中で使用できる変数 **a**,**b**,**c**,**d**,**e**,**f** に設定する値です。

開始値

変数 t の開始値

終了値

変数 t の終了値

分割数

区間の分割数。1 以上の整数で指定します。

閉じる

生成する開始値と終了値の間を繋ぎます。

終了値を省略

終了値の省略します。

３Ｄカーソル使用

式ラインは通常、ワールド座標系で動作します。ユーザー座標系が指定されていても、ワールド座標系でオブジェクトを作成してしまうので不便なことがあります。

「3D カーソル使用」チェックボックスをチェックすると、3D カーソルの座標系に従ってオブジェクトを作成します。ユーザー座標系に切り替えたとき、3D カーソルの座標系はユーザー座標系と一致します。従って、結果的にユーザー座標系に従ってオブジェクトを作成することになります。

?

説明表示。現在の設定に対する説明を表示します。Shift キーを押しながらこのボタンを押すと説明の入力ができます。

読込

テキストファイルから設定を読み込みます。

保存

テキストファイルに設定を出力します。

ユーティリティ

式で引用できる点列の設定を行います。

生成

設定に基づいて図形を生成します。

終了

プログラムを終了します。

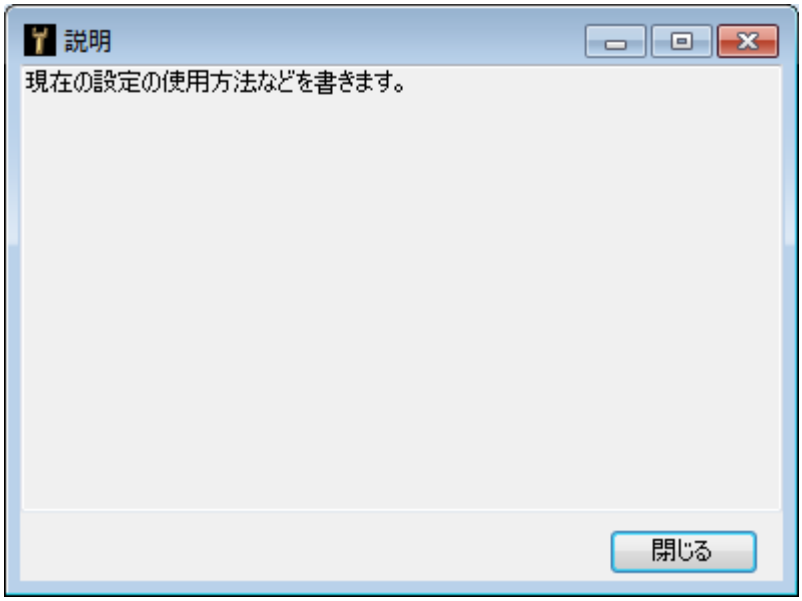
説明

現在の設定に対する説明文の表示や入力を行います。

説明文は現在の設定の一部として読込／保存されます。現在の設定の使用方法などを入力しておくと後日使用する際に便利です。

説明の表示

本ダイアログを通常の方法で呼び出すと説明文を表示します。



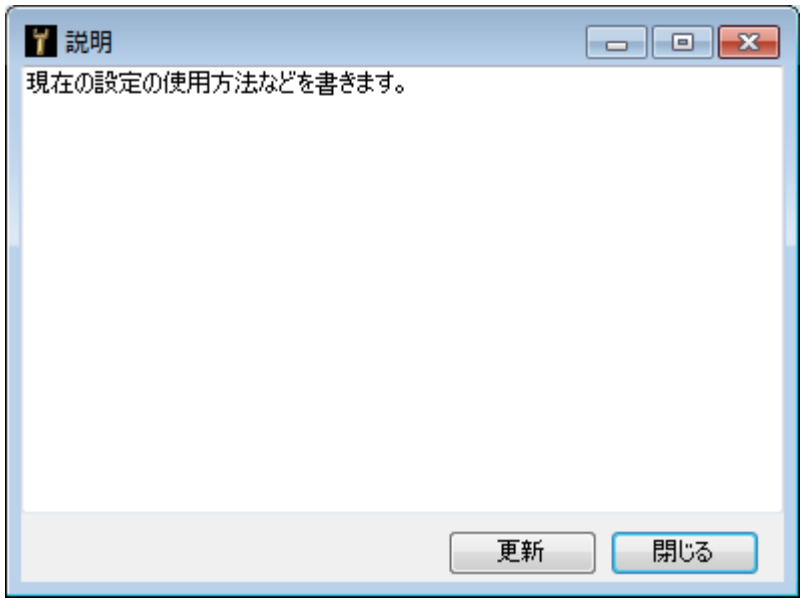
閉じる

ダイアログを閉じます。

説明の入力

Shift キーを押しながら、本ダイアログを呼び出すと、説明文を入力することができます。

説明文を変更した後、**更新ボタン**を押してください。



更新

説明文を記憶して、ダイアログを閉じます。

閉じる

説明文を無視してダイアログを閉じます。

ユーティリティ

このダイアログでは、式の中で引用できる点列を設定します。点列は頂点座標を一次元配列として並べたものです。点列は点列１と点列２の二つがあります。

頂点の設定方法は二種類あります。

1. **ファイルから読み込む**

   読み込みボタンを押してファイルを指定するとテキストファイルから読み込めます。

ファイルでは、一つの頂点を一行で表し、Ｘ座標、Ｙ座標、Ｚ座標の順に「,」で区切って書きます。空行は無視します。

2. **StoneyDesigner のポリラインから読み取る**

   一つのポリラインを選択してから、**取得**ボタンを押すと、ポリラインから点列を取り込めます。ポリラインが複数のセグメントからなる場合には、最初のセグメントからだけ取り込みます。

ダイアログのボタンの機能は次の通りです。

点列 1

頂点数

現在読み込まれている頂点の数を表示します。

読込

ファイルから点列を読み込みます。

取得

StoneyDesigner からポリラインの座標を取り込みます。ポリラインを一つだけ選択してからこ

のボタンを押してください。

点列 2

頂点数

現在読み込まれている頂点の数を表示します。

読込

ファイルから点列を読み込みます。

取得

StoneyDesigner からポリラインの座標を取り込みます。ポリラインを一つだけ選択してからこのボタンを押してください。

閉じる

ダイアログを閉じます。

座標モード

点を表す座標系を選択できます。標準では直交系を使用します。

直交座標	座標要素 x,y,z で表すデフォルトの座標系です。
円筒座標	r:半径　／　θ:角度　／　z:高さ　で表します。 Z 軸周りにある円筒のイメージです。 変換： $x=r\cos\theta$ $y=r\sin\theta$ $z=z$ 逆変換： $r=\sqrt{x^2+y^2}$ $\theta=atan2(y,x)$ $z=z$
球座標	r:半径 ／　θ:天頂角（緯度方向）　／　φ:平面角（経度方向） 地球儀のような座標系です。南極と北極を結ぶ軸はＺ軸に一致します。 天頂角はＺ軸と点ベクトルの間の角度です。 変換： $x=r\sin\theta\cos\phi$ $y=r\sin\theta\sin\phi$ $z=r\cos\theta$ 逆変換： $r=\sqrt{x^2+y^2+z^2}$ $\theta=atan2(\sqrt{x^2+y^2},z)$ $\phi=atan2(y,x)$
回転座標	U:横 ／　V:縦　／　θ:角度 UV 平面に書かれた図形をＶ軸を中心に角度θで回転させるイメージです。 回転軸はＺ軸に一致します。 変換： $x=u\cos\theta$ $y=u\sin\theta$ $z=v$ 逆変換： $u=\sqrt{x^2+y^2}$ $v=z$ $\theta=atan2(y,x)$

放物柱座標	ξ ／ η ／ z 変換： $x=\frac{c}{2}(\xi^2-\eta^2)$ $y=c\,\xi\,\eta$ $z=z$
楕円柱座標	ξ ／ η ／ z 変換： $x=c\cosh\xi\cos\eta$ $y=c\sinh\xi\sin\eta$ $z=z$
回転放物面座標	ξ ／ η ／ ψ 変換： $x=c\,\xi\,\eta\cos\psi$ $y=c\,\xi\,\eta\sin\psi$ $z=\frac{c}{2}(\xi^2-\eta^2)$
扁長回転楕円面座標	u ／ v ／ ψ 変換： $x=c\sinh u\,sinv\,v\cos\psi$ $y=c\sinh u\,sinv\,v\sin\psi$ $z=c\cosh u\cos v$
扁平回転楕円面座標	u ／ v ／ ψ 変換： $x=c\cosh u\sin v\cos\psi$ $y=c\cosh u\sin v\sin\psi$ $z=c\sinh u\cos v$
双極座標	u ／ v ／ ψ 変換： $x=c\frac{\sin v}{\cosh u-\cos v}\cos\psi$ $y=c\frac{\sin v}{\cosh u-\cos v}\sin\psi$ $z=c\frac{\sinh u}{\cosh u-\cos v}$

円環座標	u ／ v ／ ψ 変換： $x=c\frac{\sinh u}{\cosh u-\cos v}\cos\psi$ $y=c\frac{\sinh u}{\cosh u-\cos v}\sin\psi$ $z=c\frac{\sin v}{\cosh u-\cos v}$
等角写像０１	u ／v ／z 変換： $Z=z^2$ ※このモード以降に記述する等角写像は複素数を利用して平面上の座標変換を行うものです。(x,y,z)が(u,v,z)に対応します。z は変換されませんから、uv 平面から xy 平面への二次元の変換であることに注意しましょう。
等角写像０２	u ／v ／z 変換： $Z=z^{\frac{1}{2}}$
等角写像０３	u ／v ／z 変換： $Z=z^3$
等角写像０４	u ／v ／z 変換： $Z=z^{\frac{1}{3}}$
等角写像０５	u ／v ／z 変換： $Z=z^n$
等角写像０６	u ／v ／z 変換： $Z=z^{\frac{1}{n}}$
等角写像０７	u ／v ／z 変換： $Z=\frac{1}{z}$
等角写像０８	u ／v ／z 変換： $Z=z+\frac{1}{z}$
等角写像０９	u ／v ／z 変換： $Z=z+\frac{1}{z}+i\,m\log z$
等角写像１０	u ／v ／z 変換： $Z=\log z$

等角写像１１	u ／v ／z 変換： $Z=z+\log z$
等角写像１２	u ／v ／z 変換： $Z=\log(z+a)-\log(z-a)$
等角写像１３	u ／v ／z 変換： $Z=\log(z+a)+\log(z-a)$
等角写像１４	u ／v ／z 変換： $Z=\mathrm{asin}\, z$
等角写像１５	u ／v ／z 変換： $Z=\mathrm{acos}\, z$
等角写像１６	u ／v ／z 変換： $Z=\mathrm{atan}\, z$
等角写像１７	u ／v ／z 変換： $Z=\mathrm{atanh}\, z$

座標定数

一部の座標モードで座標変換時に使用される定数です。通常は 1.0 に設定しておきます。

乱数種

乱数を与える関数 RAND()の元になる乱数系列を決めます。値を 0 にすると、実行するたびに違う乱数系列を使用します。

数式

式の要素

式の要素は、いずれも数値で、内部では実数として取り扱っています。

定数

整数

数字の並びです。式記述上は整数として書きますが、内部では実数に変換します。

実数

二つの書き方があります。

整数．小数

1.5 とか、3.146 といった書き方です。

整数．小数 E 指数

1.5 E 3.2 といった形式です。現在は、1E15 といった小数点を省略した書き方を出来ません。

変数

変数名は大文字と小文字を区別しません。割り当てられた変数名はアプリケーションによって異なります。

式ラインでは次の変数を使用できます。

t 範囲指定で自動的に変化する変数です。

i t に従って生成する頂点の整数序数です。0 から始まります。

a,b,c,d,e,f 式記述を簡略するために使用する変数です。処理に先立って、ダイアログのそれぞれのテキストボックスから値を読み取ってセットします。

演算子

式の中では通常の数式と同様に、次の演算子が使用できます。優先順位も普通の数式の規則に従います。

括弧	(式)	
負符号	-	単項演算子

乗算	*	
除算	/	
剰余	%	
加算	+	
減算	-	
小さい	<	小さければ 1.0。そうでなければ 0.0。
等しいか小さい	<=	等しいか小さければ 1.0。そうでなければ 0.0。
大きい	>	大きければ 1.0。そうでなければ 0.0。
等しいか大きい	>=	等しいか大きければ 1.0。そうでなければ 0.0。
等しい	==	等しければ 1.0。そうでなければ 0.0。
等しくない	!=	等しくなければ 1.0。そうでなければ 0.0。
論理積	&&	両辺が真（0.0 以外）ならば 1.0。そうでなければ 0.0。
論理和	||	両辺のいずれかが真（0.0 以外）ならば 1.0。そうでなければ 0.0。
式羅列	,	式を,で繋ぐいだ最後の式の値
論理否定	!	単項演算子。論理値を反転します。 !の後ろの項が真(0.0 以外）ならば 0.0。そうでなければ 1.0。

関数

次の数学的関数を使用できます。関数名は大文字と小文字を区別しません。

エラー処理	ER(x)	x が不正な値であれば 0.0。そうでなければ x。
エラー処理	ERS(x,a,b,c)	x が負の無限大ならば a、不定値なら b、正の無限大ならば c、そうでなければ x。
エラー判定	ERP(x)	x が不正な値であれば 1.0、そうでなければ 0.0。
正の無限大判定	ERP_P(x)	x が正の無限大であれば 1.0、そうでなければ 0.0。
不定値判定	ERP_N(x)	x が不定値であれば 1.0、そうでなければ 0.0。

負の無限大判定	ERP_M(x)	x が負の無限大であれば 1.0、そうでなければ 0.0。
円周率	PI	円周率 π
自然対数	_E	自然対数 e
符号	SGN(x)	x の符号。-1.0:負。0.0:零。1.0:正
絶対値	ABS(x)	$\lvert x \rvert$
最大の整数	FLOOR(x)	x 以下の最大の整数
最小の整数	CEIL(x)	x 以上の最小の整数
整数化	TRUNC(x)	x の整数部
周期化	PERIOD(x [,p])	x の小数部を利用した周期化。周期は p で p を省略すると 1.0。x-floor(x)の値。
最大	MAX(a[,b...])	引数の中で最大の値。引数の数は一つ以上。
最小	MIN(a[,b...])	引数の中で最大の値。引数の数は一つ以上。
合計	Sum([a[,b...]])	引数の合計の値。引数の数は任意。
平方和	Sum2([a[,b...]])	引数の平方和。引数の数は任意。
平均	Mean([a[,b...]])	引数の平均。引数の数は任意。
階段化	SNAP(x,a,b)	x を、a を中心に幅を b とした格子に位置を合わせる。
クリッピング	CLIP(x,a,b)	x を区間(a,b)内に丸める。 $a \le b$ であること。
単位クリッピング	CLIP1(x)	x を区間(0.0,1.0)内に丸める。
比率	TVAL(x,a,b)	x を区間(a,b)の相対位置で表す。 戻り値を t とすれば、 $x=(1.0-t)a+tb$ （BLEND(t,a,b)）
混合	BLEND(x,a,b)	x の比率で a,b を線形に混合する。x=0.0 で a に、x=1.0 で b になる。
二次の混合	BLEND2(x,a,b,c)	x の比率で a,b,c を二次混合する。x=0.0 で a に、x=1.0 で c になる。
三次の混合	BLEND3(x,a,b,c,d)	x の比率で a,b,c,d を三次混合する。x=0.0 で a に、x=1.0 で d になる。

滑らか混合	BLENDW(x,a,b)	x の比率で a,b を滑らかに混合する。x=0.0 で a に、x=1.0 で b になる。
大小比較	CPR(a,b)	$a>b$ ならば、1。 $a=b$ ならば 0。 $a<b$ ならば、-1。
二値判定	IF(x,a,b)	x が 0.0 でなければ、a。0.0 ならば b。 x の値に関わらず、a、b の計算自体は行われる。
三値判定	TRIS(x,a,b,c)	$x<0.0$ ならば a。 $x=0.0$ ならば b。 $x>0.0$ ならば c。 x の値に関わらず、a、b、c の計算自体は行われる。
区間判定	SEG(x,a,b)	$a \leq x < b$ ならば 1。そうでなければ 0。
区間判定のバリエーション	SEG_CC(x,a,b)	$a \leq x \leq b$ ならば 1。そうでなければ 0。
区間判定のバリエーション	SEG_CO(x,a,b)	$a \leq x < b$ ならば 1。そうでなければ 0。
区間判定のバリエーション	SEG_OC(x,a,b)	$a < x \leq b$ ならば 1。そうでなければ 0。
区間判定のバリエーション	SEG_OO(x,a,b)	$a < x < b$ ならば 1。そうでなければ 0。
表参照	IX(x,a[,b..])	最初の引数 x の番号で、以降の引数から一つを選択し、値とする。番号は 0 から始まる整数。引数の数は二つ以上。
表補間参照	IXV(x,a,b[,c..])	二番目以降の引数の並びが区間を表すものとして、最初の引数で区間から補間して読み出す。区間のインデックスは、左端が 0.0 で右端が 1.0 とする。引数の数は三以上。
表補間参照	IXV_T(x,a,b[,c..])	IXV()の補間無し版。
表補間参照	IXV_L(x,a,b[,c..])	IXV()の直線補間版。
表補間処理	IXV_S(x,a,b,c,d[,e...])	二番目以降の引数並びが区間を表すものとして、３次補間して読み出す。インデックスは左端が 0.0 で右端が 1.0 とする。引数の数は五以上。

表補間処理	IXV_SS(x,a,b,c,d[,e...])	二番目以降の引数並びが区間を表すものとして、３次補間して読み出す。区間の最初と最後の値は傾きを決定するだけに使用される。したがって全区間の最初の値は先頭から二番目の値で、区間の最後の値は末尾から二番目の値。 インデックスは左端が 0.0 で右端が 1.0 とする。 引数の数は五以上。
常に 0	ZERO([a[,b...]])	常に 0 を返す。引数は評価するが値は用いられない。引数の数は任意。
常に 1	ONE(x)	常に 1 を返す。引数は評価するが値は用いられない。引数の数は任意。
最初の引数	FRONT(a[,b...])	最初の引数を返す。残りの引数は評価するが値は用いられない。引数の数は一つ以上。
最後の引数	REAR(a[,b...])	最後の引数を返す。残りの引数は評価するが値は用いられない。引数の数は一つ以上。
小さい	LT(a,b)	$a<b$ ならば 1.0、そうでなければ 0.0
等しいか小さい	LE(a,b)	$a\leq b$ ならば 1.0、そうでなければ 0.0
大きい	GT(a,b)	$a<b$ ならば 1.0、そうでなければ 0.0
等しいか大きい	GE(a,b)	$a<b$ ならば 1.0、そうでなければ 0.0
等しい	EQ(a,b)	$a=b$ ならば 1.0、そうでなければ 0.0
等しくない	NE(a,b)	$a\neq b$ ならば 1.0、そうでなければ 0.0
否定	NOT(x)	$x=0$ なら 1.0、そうでなければ 0.0。ZEROP(x)と同じ関数。
論理単位化	UNI(x)	$x=0$ なら 0,0、そうでなければ 1.0
正か	PLUSP(x)	$x>0$ なら 1.0、そうでなければ 0.0
負か	MINUSP(x)	$x<0$ なら 1.0、そうでなければ 0.0
ゼロか	ZEROP(x)	$x=0$ なら 1.0、そうでなければ 0.0。NOT(x)と同じ関数。
偶数か	EVENP(x)	整数 x が偶数なら 1.0、奇数なら 0.0。
奇数か	ODDP(x)	整数 x が次数なら 1.0、偶数なら 0.0。

乱数	RAND()	$0.0 \leq val < 1.0$ の範囲にある乱数
記憶	WR(i,v)	v を返す。同時に、i 番目のレジスタに v を記憶する。i は整数で 0-127。 レジスタはプログラム起動中は保持される。
記憶取り出し	RD(i)	i 番目のレジスタの値。i は整数で 0-127。 WT(i)で書き込んだ値を読み込む。
平方	SQ(x)	２乗。 $x*x$
立方	CUB(x)	３乗。 $x*x*x$
平方根	SQRT(x)	$\sqrt{x}$ 平方根。x が負の場合は、オーバーフローを避けるために $-\sqrt{-x}$ を計算する。
べき乗	POW(x,y)	x^y x の y 乗
指数関数	EXP(x)	e^x
自然対数	LOG(x)	$\log x$ または $\ln x$
常用対数	LOG10(x)	$\log_{10} x$
平面長さ	DET2(x,y)	$\sqrt{x^2+y^2}$
立体長さ	DET3(x,y,z)	$\sqrt{x^2+y^2+z^2}$
平面距離	DET2P(x0,y0,x1,y1)	$\sqrt{(x1-x0)^2+(y1-y0)^2}$ 。 点(x0,y0)と点(x1,y1)の間の距離。
立体距離	DET3P(x0,y0,z0,x1,y1,z1)	$\sqrt{(x1-x0)^2+(y1-y0)^2+(z1-z0)^2}$ 。 点(x0,y0,z0)と点(x1,y1,z1)の間の距離。
度に変換	DEG(x)	ラジアンを度に変換
ラジアンに変換	RAD(x)	度をラジアンに変換
サイン	SIN(x)	$\sin x$ 。x の単位はラジアン。以下同じ
コサイン	COS(x)	$\cos x$
タンジェント	TAN(x)	$\tan x$
セカント	SEC(x)	$\sec x$ 。 $\frac{1}{\cos x}$ に等しい。
コセカント	COSEC(x)	$\mathrm{cosec}\, x$ 。 $\frac{1}{\sin x}$ に等しい。
コタンジェント	COTAN(x)	$\cot x$ 。 $\frac{1}{\tan x}$ に等しい。

逆サイン	ASIN(x)	$\arcsin x$ または、 $\sin^{-1} x$ 。戻り値はラジアン
逆コサイン	ACOS(x)	$\arccos x$ または、 $\cos^{-1} x$ 。戻り値はラジアン
逆タンジェント	ATAN(x)	$\arctan x$ または、 $\tan^{-1} x$ 。戻り値はラジアン
逆タンジェント2	ATAN2(y,x)	y,x 座標指定から逆タンジェントを計算。
双曲線サイン	SINH(x)	$\sinh x$
双曲線コサイン	COSH(x)	$\cosh x$
双曲線タンジェント	TANH(x)	$\tanh x$
逆双曲線サイン	ASINH(x)	$\operatorname{arcsinh} x$ または $\sinh^{-1} x$
逆双曲線コサイン	ACOSH(x)	$\operatorname{arccosh} x$ または $\cosh^{-1} x$ 。 $x>1$ で有効
逆双曲線タンジェント	ATANH(x)	$\operatorname{arctanh} x$ または $\tanh^{-1} x$ 。 $\lvert x\rvert<1$ で有効
一次元ノイズ	NOISE1D(x)	単位格子を用いた一次元のノイズ関数。-1 から 1 の間で返却する。二次の補間が行われる。 単位格子のサイズは 1.0 で、以降の NOISE 系列、TURB 系列の関数で共通。
階段状一次元ノイズ	NOISE1D_T(x)	NOISE1D()のバリエーション。補間を行わない。
直線状一次元ノイズ	NOISE1D_L(x)	NOISE1D()のバリエーション。直線補間を行う。
曲線状一次元ノイズ	NOISE1D_S(x)	NOISE1D()のバリエーション。三次の補間を行う。
二次元ノイズ	NOISE2D(x,y)	単位格子を用いた二次元のノイズ関数。-1 から 1 の間で返却する。二次の補間が行われる。
階段状二次元ノイズ	NOISE2D_T(x,y)	NOISE2D()のバリエーション。補間を行わない。
直線状二次元ノイズ	NOISE2D_L(x,y)	NOISE2D()のバリエーション。直線補間を行う。
曲線状二次元ノイズ	NOISE2D_S(x,y)	NOISE2D()のバリエーション。三次の補間を行う。

三次元ノイズ	NOISE3D(x,y,z)	単位格子を用いた三次元のノイズ関数。-1 から 1 の間で返却する。二次の補間が行われる。
階段状三次元ノイズ	NOISE3D_T(x,y,z)	NOISE3D()のバリエーション。補間を行わない。
直線状三次元ノイズ	NOISE3D_L(x,y,z)	NOISE3D()のバリエーション。直線補間を行う。
曲線状三次元ノイズ	NOISE3D_S(x,y,z)	NOISE3D()のバリエーション。三次の補間を行う。
一次元再帰ノイズ	TURB1D(x,oct)	単位格子を用いた一次元の再帰ノイズ関数。oct は再帰数で正の整数を指定する。-1 から 1 の間で返却する。二次の補間が行われる。
階段状一次元再帰ノイズ	TURB1D_T(x,oct)	TURB1D()のバリエーション。補間を行わない。
直線状一次元再帰ノイズ	TURB1D_L(x,oct)	TURB1D()のバリエーション。直線補間を行う。
曲線状一次元再帰ノイズ	TURB1D_S(x,oct)	TURB1D()のバリエーション。三次の補間を行う。
二次元再帰ノイズ	TURB2D(x,y,oct)	単位格子を用いた二次元の再帰ノイズ関数。oct は再帰数で正の整数を指定する。-1 から 1 の間で返却する。二次の補間が行われる。
階段状の二次元再帰ノイズ	TURB2D_T(x,y,oct)	TURB2D()のバリエーション。補間を行わない。
直線状の二次元再帰ノイズ	TURB2D_L(x,y,oct)	TURB2D()のバリエーション。直線補間を行う。
曲線状の二次元再帰ノイズ	TURB2D_S(x,y,oct)	TURB2D()のバリエーション。三次の補間を行う。
三次元再帰ノイズ	TURB3D(x,y,z,oct)	単位格子を用いた三次元の再帰ノイズ関数。oct は再帰数で正の整数を指定する。-1 から 1 の間で返却する。二次の補間が行われる。
階段状三次元再帰ノイズ	TURB3D_T(x,y,z,oct)	TURB3D()のバリエーション。補間を行わない。
直線状三次元再帰ノイズ	TURB3D_L(x,y,z,oct)	TURB3D()のバリエーション。直線補間を行う。
曲線状三次元再帰ノイズ	TURB3D_S(x,y,z,oct)	TURB3D()のバリエーション。三次の補間を行う。
鋸波	WAVE_SAW(x)	上昇型の鋸波。周期は 1.0 で、値の範囲は、 $0.0 \leq val < 1.0$
三角波	WAVE_TRI(x)	下降＋上昇を繰り返す三角波。周期は 1.0 で、値の範囲は、 $0.0 \leq val \leq 1.0$ 。
矩形波	WAVE_RECT(x)	矩形波。周期は 1.0。前半で 0.0、後半で 1.0 の値をとる。

ＰＷＭ波	WAVE_PWM(x,p)	変調波。周期は 1.0。p は $0<p<1$ の範囲で周期内の分割点を表す。分割点の前が 0.0、後が 1.0 の値をとる。
階段関数 u 成分	STAIR_U(x)	平面上で階段を生成する際の横軸成分を発生する。段の周期は 1.0 で、その半分で進行方向が切り替わる。
階段関数 v 成分	STAIR_V(x)	平面上で階段を生成する際の縦軸成分を発生する。段の周期は 1.0 で、その半分で進行方向が切り替わる。
点列頂点数	PL_C([n])	組み込み点列 n の頂点数。n を省略すると 0 番目の点列。
点列頂点の x 成分	PL_X([n,] i)	組み込み点列 n の i 番目の頂点の x 成分。i は 0 から増加する整数。頂点数を超えるとまた最初から数えなおす。n を省略すると 0 番目の点列。
点列頂点の y 成分	PL_Y([n,] i)	組み込み点列 n の i 番目の頂点の y 成分。n を省略すると 0 番目の点列。
点列頂点の z 成分	PL_Z([n,] i)	組み込み点列 n の i 番目の頂点の z 成分。n を省略すると 0 番目の点列。
点列の x 成分	PLT_X([n,] u)	点列 n の x 成分の階段読み取り。u は 0.0 から 1.0 で 0.0 が最初の点、1.0 が最後の点を示す。n を省略すると 0 番目の点列。
点列の y 成分	PLT_Y([n,] u)	点列 n の y 成分の階段読み取り。u は 0.0 から 1.0 で 0.0 が最初の点、1.0 が最後の点を示す。n を省略すると 0 番目の点列。
点列の z 成分	PLT_Z([n,] u)	点列 n の z 成分の階段読み取り。u は 0.0 から 1.0 で 0.0 が最初の点、1.0 が最後の点を示す。n を省略すると 0 番目の点列。
点列の x 成分	PLL_X([n,] u)	点列 n の x 成分の直線補間読み取り。u は 0.0 から 1.0 で 0.0 が最初の点、1.0 が最後の点を示す。n を省略すると 0 番目の点列。

点列のｙ成分	PLL_Y([n,] u)	点列ｎのｙ成分の直線補間読み取り。ｕは0.0から1.0で0.0が最初の点、1.0が最後の点を示す。ｎを省略すると0番目の点列。
点列のｚ成分	PLL_Z([n,] u)	点列ｎのｚ成分の直線補間読み取り。ｕは0.0から1.0で0.0が最初の点、1.0が最後の点を示す。ｎを省略すると0番目の点列。

注

1. 関数における論理値の扱い

   現在は、論理型を持たないので論理値は数値で表しています。0.0が偽（False）を表し、1.0が真（True）を表します。論理値を返す関数では、0.0と1.0を返します。論理値を引数とする関数では、0.0以外の数値は真（True）として扱います。

   将来、論理型を導入するかもしれません。それに備えて、数値としての値に依存しないように記述することをお勧めします。

2. 三値判定

   関数CPR()は二つの数値を比較し、1.0、0.0、-1.0の値を返します。この値は、関数TRIS()の最初の引数と用いると便利なように決められたものです。

3. レジスタ

   レジスタは数値の記憶場所で、128個用意されています。その値はプログラムの起動中は保持されます。番号0~127で指定して、関数WR()で書き込み、関数RD()で取り出せます。処理の中で繰り返し使われる式の値などを記憶すると便利です。WR()は書き込んだ値をそのまま返すので、式中の保存したい部分をWR()で囲めば式の計算をしながら保存が出来ます。

4. 評価するが用いられない引数

   ZERO()、ONE()、FRONT()、REAR()では、評価するが用いられない引数があります。これは、WR()でセットした値を式中で用いない場合に便利です。

5. 点列の扱い

   点列は、頂点を並べたもので、現在は二組あります。アプリケーションのユーティリティ機能を使用してStoneyDesignerのポリラインの頂点並びやテキストファイルから値を設定できます。式中では、点列1は番号0、点列2は番号1で指定します。

変数ｔの範囲の指定

変数ｔは、ダイアログで指定した初期値から終了値まで変化する。分割数の直線セグメントを生成するように、ｔには範囲を均等に分割した値が順次割り当てられる。

通常は生成される直線は範囲全域を覆う開いた折れ線だが、次のオプションで動作を変更できる。

閉じる（チェックボックス)

終了点から開始点へ向かう線分を追加してポリラインを閉じる。

終了値を省略(チェックボックス）

終了値へ向かう最後の線分を省略する。円のように初期値と終了値が同じ点を示す場合、円周の全域を指定して、「閉じる」と「終了値を省略」の双方をチェックすると良い。

3D カーソル使用

ExprPlane は通常、ワールド座標系で動作します。ユーザー座標系が指定されていても、ワールド座標系でオブジェクトを作成してしまうので不便なことがあります。

「3D カーソル使用」チェックボックスをチェックすると、3D カーソルの座標系に従ってオブジェクトを作成します。ユーザー座標系に切り替えたとき、3D カーソルの座標系はユーザー座標系と一致します。従って、結果的にユーザー座標系に従ってオブジェクトを作成することになります。

保存と読込

　ダイアログの内容をテキストファイルとして保存、読込できます。「保存」ボタンで保存、「読込」ボタンで読み込みます。

バージョン

Ver.1.0.0.0 イニシャルリリース

Ver.1.0.0.1 座標モード機能を追加

保存／読込機能付加

Ver.1.0.0.2 座標モードを拡張した。

「放物柱座標」、「楕円柱座標」、「回転放物面座標」、「扁長回転楕円面座標」、「扁平回転楕円面座標」、「双極座標」、「円環座標」

Ver.1.0.0.3 座標モードを拡張した。

等角写像０１から等角写像１７までを追加した。

関数 SQ/CUB/RAND/BLENDW を追加した。

ダイアログに「乱数種」項目を追加した。

3D カーソル座標上に描画できるようになった。

Ver.1.0.0.4 ExprLn.dll の仕様変更に対応した。

Ver.1.0.0.5 変数 a-f の扱いの誤りを修正した。

Ver.1.0.0.6 ユーティリティ機能を追加した。

本ソフトウェアの取り扱い

大げさですが、一応、アルゴ算法堂が著作権を保持します。

ご自由にご使用ください。また、再配布は、配布されたものをそのまま、配布してください。

アルゴ算法堂