3D polyhedra modeler for Geogebra with polar polyhedron, Minkowski sum and 3D polar & matrix tools

principle:

The tool returns a list of texts, and each text is a geogebra command.

The list has this caption: " !!! instructions !!! "

A global javascript function captures the name of the last object created. If this object is a list, and if the caption of the list is "!!! instructions !!!", then the list is executed.

This allows sequential instructions activated by a tool, on objects selected by the mouse.

The tool has naturally access to all the objects in the construction. Therefore it is not necessary to pass as parameters all the variables it will use, like conventional tools do. So it is possible to pass as parameters, only those objects that are different for each instance of the construction.

For example, here only the vector to be added is passed to the tool. All the other variables are either taken directly from the construction, or created if they don't exist.

It is also possible to check if a variable exists or not, and to create that variable if it is required.

Some commands require initialising variable names. In that case, 2 successive "execute" are necessary, the first one to replace the names with the variables, the second one to actually execute the command on the variables themselves. The first "execute" creates a variable which is a second list of instructions, where all the variable names have been replaced by the variables, and the second "execute", next in the sequence, executes that list of instructions that was just created.

The script must be edited in word, or in an object's scripting window, and tested line by line in Geogebra. It can then be copied in GGB, when defining a tool, as the object returned by the tool. Comments can be added as "word" comments only. Extra blanks and line feed are removed by GGB, so it becomes almost impossible to understand the script after pasted in Geogebra.

The practical way to do that is to first create the object that you want. You then write the GGB script that creates that object. You then write the GGB script that creates the GGB script that creates that object. It requires a lot of mechanical copy-paste, it’s a little like doing manually the work of a compiler… Maybe someone will come up with such a compiler soon!!!

Author of the script: Laurent Fournier, Kolkata, 2018-2020

* with help from Raymond ("Rami") for the embedded javascript, here:
https://help.geogebra.org/topic/extendmodify-a-list-interactively

* References for the Minkowski Sum:
Marina Konstantatou, Allan McRobie (2017-2020) https://www.repository.cam.ac.uk/handle/1810/303621

* Reference for homogeneous coordinates & matrix algebra:
Jürgen Richter-Gebert, Perspectives on Projective Geometry (2011)
https://www.springer.com/gp/book/9783642172854

Contents

Global Java Script

```
function ggbOnInit() {
  ggbApplet.registerAddListener("exe")
  ggbApplet.registerRemoveListener("exe")
}

function exe() {
ggbApplet.unregisterAddListener("exe")
ggbApplet.unregisterRemoveListener("exe")

lastname=ggbApplet.getObjectName(ggbApplet.getObjectNumber()-1)

if (ggbApplet.getObjectType(lastname) == "list") {
   if (ggbApplet.getCaption(lastname,false) == "!!! instructions !!!" ) {
      ggbApplet.evalCommand("Execute(Object("+'"' +lastname+ '"'+"))")
      ggbApplet.deleteObject(lastname)
       }
```

```
    }

ggbApplet.registerAddListener("exe")
ggbApplet.registerRemoveListener("exe")

}
```

tool createnames (init: number)

```
{"AvInst={}",

"AvInst2={}",

"SetValue(AvInst,If(IsDefined(Object("UnicodeToLetter(34)"tag"UnicodeToLetter(34)")),{},

        {"UnicodeToLetter(34)"tag="init UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"AvA="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvA,0)",

"AvB="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvB,0)",

"Av3D="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(Av3D,0)",

"AvElev="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvElev,0)",

"AvNm="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvNm,0)",

"AvComp=false","SetLineThickness(AvComp,0)",

"AvPoly="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPoly,0)",

"AvPolyElev="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolyElev,0)",

"AvPoly3D="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPoly3D,0)",

"AvPolyline="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolyline,0)",

"AvPt="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPt,0)",

"AvFP="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvFP,0)",

"AvFPsort="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvFPsort,0)",

"AvPF="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPF,0)",

"AvPFshort="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPFshort,0)",

"AvPadj="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPadj,0)",

"Avpt="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(Avpt,0)",

"AvPolPt="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolPt,0)",

"AvPolPtLabels="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolPtLabels,0)",

"AvPolFa="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolFa,0)",

"AvPolFaLabels="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolFaLabels,0)",

"AvC="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvC,0)",
```

```
"AvD="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvD,0)",
"AvPtProj="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPtProj,0)",
"AvFaProj="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvFaProj,0)",
"AvPolPtProj="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolPtProj,0)",
"AvPolPtProjLabels="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolPtProjLabels,0)",
"AvPolFaProj="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPolFaProj,0)",
"AvPadjCir="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvPadjCir,0)",
"AvMinSeg="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvMinSeg,0)",
"AvScale="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvScale,0)",
"AvExternalFace="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvExternalFace,0)",
"AvFadj="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvFadj,0)",
"AvFPcir="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvFPcir,0)",
"AvMinSegFa="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvMinSegFa,0)",
"AvOneFacePt="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvOneFacePt,0)",
"AvOneFaceRec="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvOneFaceRec,0)",
"AvInternPtNb="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvInternPtNb,0)",
"AvInternPt="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvInternPt,0)",
"AvInternRec="UnicodeToLetter(34) UnicodeToLetter(34),"SetLineThickness(AvInternRec,0)"
}
```

tool addpoint in plan view (A, B: point)

```
{"AvInstTemp={}",
"SetValue(AvInstTemp,If(IsDefined(Object("UnicodeToLetter(34)"AvInst"UnicodeToLetter(34)")),{},
        {"UnicodeToLetter(34)"Execute(createnames(1))"UnicodeToLetter(34)"}))",
"Execute(AvInstTemp)",
"Delete(AvInstTemp)",
"SetValue(Av3D,"UnicodeToLetter(34) Name(A) UnicodeToLetter(34)")",
"SetValue(AvA, Av3D"UnicodeToLetter(34)"_{xy}"UnicodeToLetter(34)")",
"SetValue(AvB, Av3D"UnicodeToLetter(34)"_z"UnicodeToLetter(34)")",
```

Comment [A1]:
creates list AvInst={} only once

"SetValue(AvElev, Av3D"UnicodeToLetter(34)"_e"UnicodeToLetter(34)")",

"Rename("Name(B)",Av3D"UnicodeToLetter(34)"_z"UnicodeToLetter(34)")",

"Rename("Name(A)",Av3D"UnicodeToLetter(34)"_{xy}"UnicodeToLetter(34)")",

"SetValue(AvInst,{

AvElev"UnicodeToLetter(34)"=(x("UnicodeToLetter(34)"AvA
"UnicodeToLetter(34)"),y("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)"))"UnicodeToLetter(34)",

Av3D"UnicodeToLetter(34)"=(x("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)"
),y("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)"
),y("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)"))"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",3)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",2)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",3)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",blue)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"Av3D"UnicodeToLetter(34)",red)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",1)"UnicodeToLetter(34)"

})",

Comment [A2]: Defines names and renames A and B

Comment [A3]: Set the values of A_e and A_{3D}

Comment [A4]: set the 2D entities invisible in 3D graphics

Comment [A5]: sets nice appearance

"Execute(AvInst)"}

tool addpoint in 3D view (A: point)

```
{"AvInstTemp={}",
"SetValue(AvInstTemp,If(IsDefined(Object("UnicodeToLetter(34)"AvInst"UnicodeToLetter(34)")),{},
        {"UnicodeToLetter(34)"Execute(createnames(1))"UnicodeToLetter(34)"}))",
"Execute(AvInstTemp)",
"Delete(AvInstTemp)",
"SetValue(Av3D,"UnicodeToLetter(34) Name(A) UnicodeToLetter(34)")",
"SetValue(AvA,Av3D"UnicodeToLetter(34)"_{xy}"UnicodeToLetter(34)")",
"SetValue(AvElev,Av3D"UnicodeToLetter(34)"_e"UnicodeToLetter(34)")",
"SetValue(AvInst,{
AvElev"UnicodeToLetter(34)"=(x("UnicodeToLetter(34)"Av3D
"UnicodeToLetter(34)"),z("UnicodeToLetter(34)"Av3D"UnicodeToLetter(34)"))"UnicodeToLetter(34)",
AvA"UnicodeToLetter(34)"=(x("UnicodeToLetter(34)"Av3D"UnicodeToLetter(34)"),y("UnicodeToLetter(34)"Av3D"Uni
codeToLetter(34)"))"UnicodeToLetter(34)"
})",
"Execute(AvInst)",
"SetValue(AvInst,{
"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)",
"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)"
})",
"Execute(AvInst)",
"SetValue(AvInst,{
"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",
"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",
"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"Av3D"UnicodeToLetter(34)",blue)"UnicodeToLetter(34)",
"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",
```

Comment [A6]: creates list AvInst={} only once

Comment [A7]: Set the values of A_e and A_{xy}

Comment [A8]: set the 2D entities invisible in 3D graphics

Comment [A9]: sets nice appearance

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",1)"UnicodeToLetter(34)"

})",

"Execute(AvInst)"}

tool add point to face (A: point)

{"SetValue(AvInst,

{"UnicodeToLetter(34)"SetValue(AvNm,"UnicodeToLetter(34)Name(UnicodeToLetter(34)"Face"
UnicodeToLetter(34))UnicodeToLetter(34)"tag"UnicodeToLetter(34) UnicodeToLetter(34)
")"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,If(IsDefined(Object(AvNm)),{},{AvNm"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvA,"UnicodeToLetter(34) Name(A) UnicodeToLetter(34)")",

"SetValue(AvInst, {

"UnicodeToLetter(34)"

If(Last(AvA,5)==
"UnicodeToLetter(34)"Name("UnicodeToLetter(34)"_{xy}"UnicodeToLetter(34)")"UnicodeToLetter(34)",
SetValue(AvA,First(AvA,Length(AvA)-5)))

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",black)"UnicodeToLetter(34)"

})",

Comment [A10]:
Assign AvNm = "Face"tag

Comment [A11]:
creates list Facetag={} if doesn't exist

Comment [A12]: creates AvA=Name(A)

Comment [A13]: if AvA ends in "_{xy}" then remove "_{xy}"

Comment [A14]: reduces the visibility of Point A

"Execute(AvInst)",

Comment [A15]: in case the last point is the same as the first point then the face is completed

"SetValue(AvInst, {

"UnicodeToLetter(34)"

If(AvA=="UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)"(1),SetValue(AvComp,true),SetValue(AvComp,false))

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

Comment [A16]: updates list Facetag by adding the name

"SetValue(AvInst,{"UnicodeToLetter(34)"

SetValue("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",

KeepIf(IsDefined(Object(a)),a,

Append("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",AvA)))"UnicodeToLetter(34)"})",

"Execute(AvInst)",

Comment [A17]: updates names
AvPoly = "F"tag
AvPolyElev="Fe"tag
AvPoly3D="f"tag

"SetValue(AvPoly,"UnicodeToLetter(34)"F"UnicodeToLetter(34)"tag)",

"SetValue(AvPolyElev,"UnicodeToLetter(34)"Fe"UnicodeToLetter(34)"tag)",

"SetValue(AvPoly3D,"UnicodeToLetter(34)"f"UnicodeToLetter(34)"tag)",

Comment [A18]: creates polygon AvPoly3D

"SetValue(AvInst, If(IsDefined(Object(AvPoly3D)),{},

{AvPoly3D"UnicodeToLetter(34)"=

Polygon(Zip(Object(a),a,"UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)"))

"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

Comment [A19]: creates polygon AvPoly

"SetValue(AvInst,If(IsDefined(Object(AvPoly)),{},

{AvPoly"UnicodeToLetter(34)"=

Polygon(Zip(Object(a

"UnicodeToLetter(34)"UnicodeToLetter(34)"UnicodeToLetter(34)"_{xy}"UnicodeToLetter(34)"
UnicodeToLetter(34)"UnicodeToLetter(34)"),a,"UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)"))

"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst, If(IsDefined(Object(AvPolyElev)),{},

{AvPolyElev"UnicodeToLetter(34)"=

Polygon(Zip(Object(a

"UnicodeToLetter(34)"UnicodeToLetter(34)"UnicodeToLetter(34)"_e"UnicodeToLetter(34)"
UnicodeToLetter(34)"UnicodeToLetter(34)"),a,"UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)"))

"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvPoly"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvPolyElev"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",black)"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst, {"UnicodeToLetter(34)"

If(AvComp, SetValue("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",
First("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",Length("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)"
)-1)))

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"If(AvComp,SetValue(tag,tag+1))"UnicodeToLetter(34)"})",

"Execute(AvInst)"}

Comment [A20]: creates polygon AvPolyElev

Comment [A21]: set the 2D entities invisible in 3D graphics

tool add face (f1: polygon)

{"SetValue(AvInst,If(IsDefined(Object("UnicodeToLetter(34)"polytag"UnicodeToLetter(34)")),{},

{"UnicodeToLetter(34)"polytag=1"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"SetValue(AvNm,"UnicodeToLetter(34)Name(UnicodeToLetter(34)"FaNm"
UnicodeToLetter(34))"polytag"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,If(IsDefined(Object(AvNm)),{},{AvNm"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"

SetValue(AvA,

"UnicodeToLetter(34)Name(UnicodeToLetter(34)"Face"UnicodeToLetter(34))UnicodeToLetter(34)"
+Take(Name("Name(f1)"),2))

" UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"

SetValue("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",

KeepIf(IsDefined(Object(a)),a,

If(AvA ∈"UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",

Remove("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",{AvA}),

Append("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",AvA))))"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvB,"UnicodeToLetter(34)"Polyhedron"UnicodeToLetter(34)"polytag)",

Comment [A22]:
creates number polytag=1 if doesn't exist

Comment [A23]:
Updates the value of "FaNm"polytag each time a new list of polyhedra is created

Comment [A24]:
creates list "FaNm"polytag={} if doesn't exist

Comment [A25]:
converts the name of the parameter polygon "f1" to the name of the list of points "Face1"
so any polygon with 1 letter before the number can be used

Comment [A26]:
updates list namelist

Comment [A27]:
Sets text AvB = "View"tag

Comment [A28]:
creates the list of polygons to highlight

"SetValue(AvInst,If(IsDefined(Object(AvB)),{},

{AvB"UnicodeToLetter(34)"=

Zip(Polygon(a),a,Zip(Zip(Object(a),a,Object(b)),b,

"UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)"))

"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(Av3D,"UnicodeToLetter(34)"NbFa"UnicodeToLetter(34)"polytag)",

"SetValue(AvInst,If(IsDefined(Object(Av3D)),{},
{Av3D"UnicodeToLetter(34)"=Length("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)")"UnicodeToLetter(34)"}))",

"Execute(AvInst)"}

Comment [A29]:
creates number Length(FaNm1) if doesn't exist

tool polar point from face (f1: polygon)

Comment [A30]:
converts the name of the parameter polygon "f1" to the name of the list of points "Face1"
so any polygon with 1 letter before the number can be used

{"SetValue(AvInst,{"UnicodeToLetter(34)"

SetValue(AvA,

"UnicodeToLetter(34)Name(UnicodeToLetter(34)"f_"UnicodeToLetter(34))UnicodeToLetter(34)"
+Take(Name("Name(f1)"),2))

" UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"

SetValue(AvB,

"UnicodeToLetter(34)Name(UnicodeToLetter(34)"F_"UnicodeToLetter(34))UnicodeToLetter(34)"
+Take(Name("Name(f1)"),2))

" UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"

SetValue(Av3D, "UnicodeToLetter(34)Name(UnicodeToLetter(34)"f"UnicodeToLetter(34))UnicodeToLetter(34)"
+Take(Name("Name(f1)"),2))

" UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,If(IsDefined(Object(AvB)),{},

{AvB"UnicodeToLetter(34)"=

matpoint(z1to1(polar3D(planemat(Plane("UnicodeToLetter(34)"Av3D"UnicodeToLetter(34)")))))

"UnicodeToLetter(34)",

AvA"UnicodeToLetter(34)"=

projection("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)")

"UnicodeToLetter(34)"

}))",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",-1,false)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",1,false)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",2,true)
"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",2)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",black)"UnicodeToLetter(34)"

})",

"Execute(AvInst)"}

Comment [A31]:
creates the polar point and its projection

tool polar polyhedron (FaNm1: list)

{"AvInstTemp={}",

Comment [A32]:
creates variables names

```
"SetValue(AvInstTemp,If(IsDefined(Object("UnicodeToLetter(34)"AvInst"UnicodeToLetter(34)")),{},
        {"UnicodeToLetter(34)"Execute(createnames(1))"UnicodeToLetter(34)"}))",
"Execute(AvInstTemp)",
"Delete(AvInstTemp)",
"SetValue(AvInst,{"UnicodeToLetter(34)"
SetValue(AvA,Take(Name("Name(FaNm1)"),5))
" UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvB,"UnicodeToLetter(34)"Fa"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvB)),{},
        {AvB"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",
        Zip(Object(b), b, FaNm"UnicodeToLetter(34)"AvA"UnicodeToLetter(34)"))"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvElev,"UnicodeToLetter(34)"PtNm"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvElev)),{},
        { AvElev"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",
        Unique(Flatten("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)")))"UnicodeToLetter(34)"})",
"Execute(AvInst)",
```

Comment [A33]:
Gets the tag of the polyhedra

Comment [A34]:
Creates the name "Fa"tag

Comment [A35]:
creates list Fa1={} if doesn't exist

Comment [A36]:
Sets the value of list Fa1

Comment [A37]:
Creates the name "PtNm"tag

Comment [A38]:
creates list PtNm1={} if doesn't exist

Comment [A39]:
Sets the value of list PtNm1

Comment [A40]:
Creates the name "FPsort"tag

```
"SetValue(AvFPsort,"UnicodeToLetter(34)"FPsort"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvFPsort)),{},
        {AvFPsort"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)",
        Zip(Zip(IndexOf(a,"UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)"),a,b),b,
"UnicodeToLetter(34)"AvB"UnicodeToLetter(34)"))"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvPt,"UnicodeToLetter(34)"Pt"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvPt)),{},
        { AvPt"UnicodeToLetter(34)"=
        Zip(Object(a),a,"UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvPoly,"UnicodeToLetter(34)"Polyhedron"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvPoly)),{},
        {AvPoly"UnicodeToLetter(34)"=
                Zip(Polygon(Zip(Element("UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)",a),a,b)),b,
                        "UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvC,"UnicodeToLetter(34)"NbPt"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvC)),{},
```

Comment [A41]: creates list FPsort1={} if doesn't exist

Comment [A42]: Sets the value of list FPsort1

Comment [A43]: Creates the name "Pt"tag

Comment [A44]: creates list Pt1 if doesn't exist

Comment [A45]: Creates the name "Polyhedron"tag

Comment [A46]: creates list Polyhedron1 if doesn't exist

Comment [A47]: Creates the name "NbPt"tag

Comment [A48]: creates number NbPt1 if doesn't exist

```
        { AvC"UnicodeToLetter(34)"=
                Length("UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A49]: Creates the name "NbFa"tag

```
"SetValue(AvD,"UnicodeToLetter(34)"NbFa"UnicodeToLetter(34)"AvA)",
```

Comment [A50]: creates number NbFa1 if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvD)),{},
        { AvD"UnicodeToLetter(34)"=
                Length("UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A51]: Creates the name "FP"tag

```
"SetValue(AvFP,"UnicodeToLetter(34)"FP"UnicodeToLetter(34)"AvA)",
```

Comment [A52]: creates list FP1={} if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvFP)),{},
        {AvFP"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A53]: Sets the value of list FP1

```
"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"AvFP"UnicodeToLetter(34)",
        Zip(Sequence(If(a ∈ l,a),a,1,"UnicodeToLetter(34)"AvC"UnicodeToLetter(34)"),
                l,"UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)"))
        "UnicodeToLetter(34)"})",
"Execute(AvInst)",
```

Comment [A54]: Creates the name "PF"tag

```
"SetValue(AvPF,"UnicodeToLetter(34)"PF"UnicodeToLetter(34)"AvA)",
```

Comment [A55]: creates list PF1={} if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPF)),{},
        {AvPF"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"AvPF"UnicodeToLetter(34)",

Sequence(Sequence(If(a ∈ "UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)"(n),n),n,1,
"UnicodeToLetter(34)"AvD"UnicodeToLetter(34)"),a,1,"UnicodeToLetter(34)"AvC"UnicodeToLetter(34)"))

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvPFshort,"UnicodeToLetter(34)"PFshort"UnicodeToLetter(34)"AvA)",

"SetValue(AvInst,If(IsDefined(Object(AvPFshort)),{},

{AvPFshort"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"AvPFshort"UnicodeToLetter(34)",

Zip(RemoveUndefined(a),a,"UnicodeToLetter(34)"AvPF"UnicodeToLetter(34)"))

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(Avpt,"UnicodeToLetter(34)"pt"UnicodeToLetter(34)"AvA)",

"SetValue(AvInst,If(IsDefined(Object(Avpt)),{},

{Avpt"UnicodeToLetter(34)"=1"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvPadj,"UnicodeToLetter(34)"Padj"UnicodeToLetter(34)"AvA)",

"SetValue(AvInst,If(IsDefined(Object(AvPadj)),{},

{AvPadj"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)",

Comment [A56]:
Sets the value of list PF1

Comment [A57]:
Creates the name "PFshort"tag

Comment [A58]:
creates list PFshort1={} if doesn't exist

Comment [A59]:
Sets the value of list PFshort1

Comment [A60]:
Creates the name "pt"tag

Comment [A61]:
creates number pt=1 if doesn't exist

Comment [A62]:
Creates the name "Padj"tag

Comment [A63]:
creates list Padj={} if doesn't exist

Comment [A64]:
sets the value of the number pt to 1

```
                    1

            )"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)

            "SetValue("UnicodeToLetter(34)"AvPadj"UnicodeToLetter(34)",

                    "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)",

            Last(
            IterationList(
                    Element(RemoveUndefined(Zip(
                                    If(IsDefined(PT),
                                            If(PT ≠ Element(prev,1), {PT,F})),
                                            PT,
                                            Element(Zip(
                                                    Zip(If(a≠b,
                                                            Element(Remove(Intersection(
                                                    "UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)"(a),

                                                    "UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)"(b)),

                                                    {"UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)"}),

                                                            1)),
                                                            b, "UnicodeToLetter(34)"AvPF"UnicodeToLetter(34)"(
                                                    "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)")),
                                                     a,"UnicodeToLetter(34)"AvPF"UnicodeToLetter(34)"(
                                            "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)")),
                                            Element(prev,2)),
                                    F, Sequence("UnicodeToLetter(34)"AvD"UnicodeToLetter(34)"))),
                            1),
                    prev,{{1,Element("UnicodeToLetter(34)"AvPFshort"UnicodeToLetter(34)",
            "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)",1)}},
                    Length("UnicodeToLetter(34)"AvPFshort"UnicodeToLetter(34)"(
            "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)"))),
                    Length("UnicodeToLetter(34)"AvPFshort"UnicodeToLetter(34)"(
            "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)")))
                    )"
                    UnicodeToLetter(34)"," UnicodeToLetter(34)
"SetValue("UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)","UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)"+1)"
UnicodeToLetter(34)"})",

"SetValue(AvInst2,
{"UnicodeToLetter(34)"
Repeat("UnicodeToLetter(34)"AvC"UnicodeToLetter(34)",Execute(AvInst))
"UnicodeToLetter(34)"})",
"Execute(AvInst2)",


"SetValue(AvPolPt,"UnicodeToLetter(34)"PolPt"UnicodeToLetter(34)"AvA)",
```

Comment [A65]:
Creates the name "PolPt"tag

Comment [A66]:
creates list PolPt if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPolPt)),{},
        {AvPolPt"UnicodeToLetter(34)"=
        Zip(matpoint(z1to1(polar3D(planemat(Plane(a))))),a,"UnicodeToLetter(34)"AvPoly"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A67]: Creates the name "PolFa"tag

```
"SetValue(AvPolFa,"UnicodeToLetter(34)"PolFa"UnicodeToLetter(34)"AvA)",
```

Comment [A68]: creates list PolFa if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPolFa)),{},
        {AvPolFa"UnicodeToLetter(34)"=
        Zip(Polygon(Zip("UnicodeToLetter(34)"AvPolPt"UnicodeToLetter(34)"(Element(a,2)),a,l)),
        l,"UnicodeToLetter(34)"AvPadj"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A69]: Creates the name "PolPtLabels"tag

```
"SetValue(AvPolPtLabels,"UnicodeToLetter(34)"PolPtLabel"UnicodeToLetter(34)"AvA)",
```

Comment [A70]: creates list PolPtLabels if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPolPtLabels)),{},
        {AvPolPtLabels"UnicodeToLetter(34)"=
        Zip(Text(Take(f,5),p),
                f, "Name(FaNm1)",
                p,"UnicodeToLetter(34)"AvPolPt"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A71]: Creates the name "PolFaLabels"tag

```
"SetValue(AvPolFaLabels,"UnicodeToLetter(34)"PolFaLabel"UnicodeToLetter(34)"AvA)",
```

Comment [A72]: creates list PolFaLabels if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPolFaLabels)),{},
        {AvPolFaLabels"UnicodeToLetter(34)"=
        Zip(Text(a,Centroid(f)),
```

```
                a,"UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",

                f,"UnicodeToLetter(34)"AvPolFa"UnicodeToLetter(34)")

        "UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPoly"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvPoly"UnicodeToLetter(34)",1)"UnicodeToLetter(34)
",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPolPt"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvPolPt"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPolPtLabels"UnicodeToLetter(34)",black)"UnicodeToLetter(
34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPolFa"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvPolFa"UnicodeToLetter(34)",1)"UnicodeToLetter(3
4)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPolFaLabels"UnicodeToLetter(34)",red)"UnicodeToLetter(3
4)"

})",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvPolPt"UnicodeToLetter(34)",-1,true)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvPolPtLabels"UnicodeToLetter(34)",-1,true)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvPolFa"UnicodeToLetter(34)",-1,true)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvPolFaLabels"UnicodeToLetter(34)",-1,true)
"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",
```

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvC"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvD"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvElev"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvFP"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvPF"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvPFshort"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvPadj"UnicodeToLetter(34)",0)"UnicodeToLetter(34)"

})",

"Execute(AvInst)"}

++

tool Minkowski Sum (FaNm1: list)

{"AvInstTemp={}",

"SetValue(AvInstTemp,If(IsDefined(Object("UnicodeToLetter(34)"AvInst"UnicodeToLetter(34)")),{},

{"UnicodeToLetter(34)"Execute(createnames(1))"UnicodeToLetter(34)"}))",

"Execute(AvInstTemp)",

"Delete(AvInstTemp)",

"SetValue(AvInst,{"UnicodeToLetter(34)"

SetValue(AvA,Take(Name("Name(FaNm1)"),5))

" UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvB,"UnicodeToLetter(34)"Fa"UnicodeToLetter(34)"AvA)",

Comment [A73]:
creates variables names

Comment [A74]:
Gets the tag of the polyhedra

Comment [A75]:
Creates the name "Fa"tag

"SetValue(AvElev,"UnicodeToLetter(34)"PtNm"UnicodeToLetter(34)"AvA)",

Comment [A76]:
Creates the name "PtNm"tag

"SetValue(AvFPsort,"UnicodeToLetter(34)"FPsort"UnicodeToLetter(34)"AvA)",

Comment [A77]:
Creates the name "FPsort"tag

"SetValue(AvPt,"UnicodeToLetter(34)"Pt"UnicodeToLetter(34)"AvA)",

Comment [A78]:
Creates the name "Pt"tag

"SetValue(AvPoly,"UnicodeToLetter(34)"Polyhedron"UnicodeToLetter(34)"AvA)",

Comment [A79]:
Creates the name "Polyhedron"tag

"SetValue(AvC,"UnicodeToLetter(34)"NbPt"UnicodeToLetter(34)"AvA)",

Comment [A80]:
Creates the name "NbPt"tag

"SetValue(AvD,"UnicodeToLetter(34)"NbFa"UnicodeToLetter(34)"AvA)",

Comment [A81]:
Creates the name "NbFa"tag

"SetValue(AvFP,"UnicodeToLetter(34)"FP"UnicodeToLetter(34)"AvA)",

Comment [A82]:
Creates the name "FP"tag

"SetValue(AvPF,"UnicodeToLetter(34)"PF"UnicodeToLetter(34)"AvA)",

Comment [A83]:
Creates the name "PF"tag

"SetValue(AvPFshort,"UnicodeToLetter(34)"PFshort"UnicodeToLetter(34)"AvA)",

Comment [A84]:
Creates the name "PFshort"tag

"SetValue(Avpt,"UnicodeToLetter(34)"pt"UnicodeToLetter(34)"AvA)",

Comment [A85]:
Creates the name "pt"tag

"SetValue(AvPadj,"UnicodeToLetter(34)"Padj"UnicodeToLetter(34)"AvA)",

Comment [A86]:
Creates the name "Padj"tag

"SetValue(AvPolFa,"UnicodeToLetter(34)"PolFa"UnicodeToLetter(34)"AvA)",

Comment [A87]:
Creates the name "PolFa"tag

"SetActiveView(2)",

"SetValue(AvScale,"UnicodeToLetter(34)"Scale"UnicodeToLetter(34)"AvA)",

Comment [A88]:
Creates the name

"SetValue(AvInst,If(IsDefined(Object(AvScale)),{},

Comment [A89]:
creates scale if doesn't exist

```
        { AvScale"UnicodeToLetter(34)"=Slider(0,1,0.05)"UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A90]: Creates the name

```
"SetValue(AvExternalFace,"UnicodeToLetter(34)"ExternalFace"UnicodeToLetter(34)"AvA)",
```

Comment [A91]: creates slider ExternalFace if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvExternalFace)),{},
        {AvExternalFace"UnicodeToLetter(34)"=
                Slider(1,"UnicodeToLetter(34)"AvD"UnicodeToLetter(34)",1)
                "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A92]: Creates the name

```
"SetValue(AvPtProj,"UnicodeToLetter(34)"PtProj"UnicodeToLetter(34)"AvA)",
```

Comment [A93]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPtProj)),{},
        {AvPtProj"UnicodeToLetter(34)"=
        Zip(    ((1-"UnicodeToLetter(34)"AvScale"UnicodeToLetter(34)")x(A),
                (1-"UnicodeToLetter(34)"AvScale"UnicodeToLetter(34)")y(A)),
        A,"UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)")
                "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A94]: Creates the name

```
"SetValue(AvFaProj,"UnicodeToLetter(34)"FaProj"UnicodeToLetter(34)"AvA)",
```

Comment [A95]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvFaProj)),{},
        {AvFaProj"UnicodeToLetter(34)"=
        Zip(Polygon(Zip("UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(a),a,l)),
                l, "UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)")
                "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A96]: Creates the name

```
"SetValue(AvPolPtProj,"UnicodeToLetter(34)"PolPtProj"UnicodeToLetter(34)"AvA)",
```

Comment [A97]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPolPtProj)),{},
        {AvPolPtProj"UnicodeToLetter(34)"=
        Zip(Intersect(z = 0,
                PerpendicularLine((0, 0, -"UnicodeToLetter(34)"AvScale"UnicodeToLetter(34)"), Plane(a))),
                a, "UnicodeToLetter(34)"AvPoly"UnicodeToLetter(34)")
                "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A98]: Creates the name

```
"SetValue(AvPolPtProjLabels,"UnicodeToLetter(34)"PolPtProjLabels"UnicodeToLetter(34)"AvA)",
```

Comment [A99]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPolPtProjLabels)),{},
        {AvPolPtProjLabels"UnicodeToLetter(34)"=
        Zip(Text(Take(f,5),p),
                f, "Name(FaNm1)",
                p,"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A100]: Creates the name

```
"SetValue(AvPolFaProj,"UnicodeToLetter(34)"PolFaProj"UnicodeToLetter(34)"AvA)",
```

Comment [A101]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvPolFaProj)),{},
        {AvPolFaProj"UnicodeToLetter(34)"=
        Zip(Polygon(Zip("UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(a, 2)),
                        a, l)),
        l, "UnicodeToLetter(34)"AvPadj"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A102]: Creates the name

```
"SetValue(AvPadjCir,"UnicodeToLetter(34)"PadjCir"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvPadjCir)),{},
        {AvPadjCir"UnicodeToLetter(34)"=
        Zip(Append(a,Element(a,1)),a,"UnicodeToLetter(34)"AvPadj"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvMinSeg,"UnicodeToLetter(34)"MinSeg"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvMinSeg)),{},
        {AvMinSeg"UnicodeToLetter(34)"=
        Zip(Sequence({{pt1,Element(PF,n+1,1)},{Element(PF,n,2),Element(PF,n+1,2)}},n,1,Length(PF)-1),
        PF,"UnicodeToLetter(34)"AvPadjCir"UnicodeToLetter(34)",
        pt1,Sequence("UnicodeToLetter(34)"AvC"UnicodeToLetter(34)"))
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvFadj,"UnicodeToLetter(34)"Fadj"UnicodeToLetter(34)"AvA)",
"SetValue(AvInst,If(IsDefined(Object(AvFadj)),{},
        {AvFadj"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",
"Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)",
                1
        )"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)
        "SetValue("UnicodeToLetter(34)"AvFadj"UnicodeToLetter(34)",
                "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)",
        Last(
```

Comment [A103]: creates list if doesn't exist

Comment [A104]: Creates the name

Comment [A105]: creates list if doesn't exist

Comment [A106]: Creates the name

Comment [A107]: creates list if doesn't exist

Comment [A108]: sets the value of the number pt to 1

```
IterationList(
        Element(RemoveUndefined(Zip(
                If(IsDefined(PT),
                        If(PT ≠ Element(prev,1), {PT,F})),
                        PT,
                        Element(Zip(
                                Zip(If(a≠b,
                                        Element(Remove(Intersection(
                                Element("UnicodeToLetter(34)"AvPFshort"UnicodeToLetter(34)",a),

                                Element("UnicodeToLetter(34)"AvPFshort"UnicodeToLetter(34)",b)),

                                {"UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)"}),

                                        1)),
                                        b,Element("UnicodeToLetter(34)"AvFP"UnicodeToLetter(34)
                                "
                                        ,"UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)")),
                                 a,Element("UnicodeToLetter(34)"AvFP"UnicodeToLetter(34)"
                                ,"UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)")),
                        Element(prev,2)),
                F, Sequence("UnicodeToLetter(34)"AvC"UnicodeToLetter(34)"))),
                1),
        prev,{{0,Element("UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)",
                "UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)",1)}},
        Length(Element("UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)"
                ,"UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)"))),
        Length(Element("UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)"
                ,"UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)")))
        )"
        UnicodeToLetter(34)"," UnicodeToLetter(34)
"SetValue("UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)","UnicodeToLetter(34)"Avpt"UnicodeToLetter(34)"+1)"
UnicodeToLetter(34)"})",

"SetValue(AvInst2,
{"UnicodeToLetter(34)"
Repeat("UnicodeToLetter(34)"AvD"UnicodeToLetter(34)",Execute(AvInst))
"UnicodeToLetter(34)"})",
"Execute(AvInst2)",

"SetValue(AvFPcir,"UnicodeToLetter(34)"FPcir"UnicodeToLetter(34)"AvA)",

"SetValue(AvInst,If(IsDefined(Object(AvFPcir)),{},

        { AvFPcir"UnicodeToLetter(34)"=

        Zip(Append(a,Element(a,1)),a,"UnicodeToLetter(34)"AvFadj"UnicodeToLetter(34)")

        "UnicodeToLetter(34)"}))",

"Execute(AvInst)",
```

Comment [A109]: Creates the name

Comment [A110]: creates list if doesn't exist

Comment [A111]: Creates the name

```
"SetValue(AvMinSegFa,"UnicodeToLetter(34)"MinSegFa"UnicodeToLetter(34)"AvA)",
```

Comment [A112]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvMinSegFa)),{},
        {AvMinSegFa"UnicodeToLetter(34)"=
        Zip(Sequence({{pt1,Element(PF,n+1,1)},{Element(PF,n,2),Element(PF,n+1,2)}},
        n,1,Length(PF)-1),
        PF,"UnicodeToLetter(34)"AvFPcir"UnicodeToLetter(34)",
        pt1,Sequence("UnicodeToLetter(34)"AvD"UnicodeToLetter(34)"))
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A113]: Creates the name

```
"SetValue(AvOneFacePt,"UnicodeToLetter(34)"OneFacePt"UnicodeToLetter(34)"AvA)",
```

Comment [A114]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvOneFacePt)),{},
        {AvOneFacePt"UnicodeToLetter(34)"=
                Zip({   "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,2,1))
                        +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,1,1)),
                        "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,2,1))
                        +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,1,2)),
                        "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,2,2))
                        +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,1,2)),
                        "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,2,2))
                        +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,1,1))}
                , Rec,"UnicodeToLetter(34)"AvMinSegFa"UnicodeToLetter(34)"(
        "UnicodeToLetter(34)"AvExternalFace"UnicodeToLetter(34)"))
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A115]: Creates the name

```
"SetValue(AvOneFaceRec,"UnicodeToLetter(34)"OneFaceRec"UnicodeToLetter(34)"AvA)",
```

Comment [A116]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvOneFaceRec)),{},
```

```
        {AvOneFaceRec"UnicodeToLetter(34)"=
        Zip(Polygon(l),l,"UnicodeToLetter(34)"AvOneFacePt"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A117]: Creates the name

```
"SetValue(AvInternPtNb,"UnicodeToLetter(34)"InternPtNb"UnicodeToLetter(34)"AvA)",
```

Comment [A118]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvInternPtNb)),{},
        {AvInternPtNb"UnicodeToLetter(34)"=
        Remove(Sequence("UnicodeToLetter(34)"AvC"UnicodeToLetter(34)"),
"UnicodeToLetter(34)"AvFPsort"UnicodeToLetter(34)"(
"UnicodeToLetter(34)"AvExternalFace"UnicodeToLetter(34)"))
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

Comment [A119]: Creates the name

```
"SetValue(AvInternPt,"UnicodeToLetter(34)"InternPt"UnicodeToLetter(34)"AvA)",
```

Comment [A120]: creates list if doesn't exist

```
"SetValue(AvInst,If(IsDefined(Object(AvInternPt)),{},
        {AvInternPt"UnicodeToLetter(34)"=
                Zip(Zip({       "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,1,1))
                                +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,2,1)),
                                "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,1,1))
                                +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,2,2)),
                                "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,1,2))
                                +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,2,2)),
                                "UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)"(Element(Rec,1,2))
                                +"UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)"(Element(Rec,2,1))}
                , Rec,"UnicodeToLetter(34)"AvMinSeg"UnicodeToLetter(34)"(pt)),
                pt,"UnicodeToLetter(34)"AvInternPtNb"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"}))",
"Execute(AvInst)",
```

"SetValue(AvInternRec,"UnicodeToLetter(34)"InternRec"UnicodeToLetter(34)"AvA)",

"SetValue(AvInst,If(IsDefined(Object(AvInternRec)),{},

{AvInternRec"UnicodeToLetter(34)"=

Zip(Zip(Polygon(a),a,b),b,"UnicodeToLetter(34)"AvInternPt"UnicodeToLetter(34)")

"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvOneFacePt"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvOneFacePt"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvOneFaceRec"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvOneFaceRec"UnicodeToLetter(34)",blue)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvInternPt"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvInternPt"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvInternRec"UnicodeToLetter(34)",red)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvInternRec"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPtProj"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvFaProj"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)",0)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPolPtProj"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPolPtProjLabels"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

Comment [A121]:
Creates the name

Comment [A122]:
creates list if doesn't exist

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvPolFaProj"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineStyle("UnicodeToLetter(34)"AvPolFaProj"UnicodeToLetter(34)",5)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvPolFaProj"UnicodeToLetter(34)",black)"UnicodeToLetter(34)"

})",

"Execute(AvInst)"}