Shelly

COLLABORATORS

	TITLE : Shelly	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 15, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Shelly

1.1 Contents

6.6.1994

Welcome to:

SHELLY V1.5

1.2 introduction

INTRODUCTION:

```
Shelly is a tool that generates 3D-Objects of various
seashells (Ammonites, Slug-houses etc.)
for: POV-V2.0, Real3DV2,X3D and T3Dlib (the last means Imagine,
DXF, Rayshade, Vort, Post-Script etc. support! ).
(take a look at "examples.jpg")

It uses an algorithm found in:
 Computer&Graphics Vol. 17,No. 1,pp. 79-84, 1993
 ("DIGITAL SEASHELLS" by M.B. Cortie.)

It was written in (portable) C using GCC2.3.3 with TDS, on
an AMIGA.

The POV output of Shelly consists of triangles.
The Real3D output is a RPL-Macro executable via
"Execute named" in the "Macros"-menu and produces
a big B-Spline-mesh (except for new calculation mode).
The T3D-output can be converted via TDDD2xxx to many different
formats (of course you'll have to get the converters first)
(available on Aminet, look in 'pub/aminet/gfx/3d' ...)

Have fun !
```

1.3 changes

```
Changes:
--------

This will be the last version of Shelly unless a big amount of
letters, postcards or E-Mail convince me to keep on!


from V1.4 to V1.5:
------------------
- changed NODULE-mode (consult the Usage section!)
  (some keywords are gone!)

- added new calculation-mode (NEWNOD)^^^!

- added new output (X3D, a VERY fast model-viewer for X11)

- included a small Tcl/Tk-GUI (start with 'wish -file sh.tcl'
  it should be self explanatory, so no documention ... yet?)

- fixed some problems with not initialized data

- allows to set cameraposition for pov-output now (overrides internal
  calculation of the position)


from V1.3 to V1.4:
------------------
- fixed a bug in the NODULE calculation-mode (Shelly could end up
```

```
  in an infinite loop for some parameters)

- fixed a bug that caused "Float-Exceptions" (when calculating
  shells without nodules) on Alpha-Architecture (and others?)

- added "RENDER"-feature (Shelly calls POV itself to render a
  preview)

- added "Scale"-feature (the shell may be scaled now)

- Shelly now supports 3 different nodule-types in one shell
  (see threenod.shy)

- Shelly now produces outputfiles with the right versionnumber ;)


from V1.2 to V1.3:
------------------
- fixed a bug in the nodule creation (nodules were not
  symmetrical)

- added new calculation mode (I call it "dynamic_stepsize"),
  please consult the Usage section!

- added new outputtype ("RAW")

- shelly uses a temporary file for calculation of the camera-
  position (rather than calculating all points twice) now!


from V1.0 to V1.2:
------------------
- added "autofocus" for POV-output (automatic placement of the
  camera in the right distance)

- the file 'shelly.pov' is never consultet now!

- the RPL-output now generates much (100 times) smaller objects

- added T3Dlib-support (new Keyword is 'T3D'!)

- the silly countdown is gone

- this time Shelly comes with only one guide ! (hope so :))
```

1.4 contents

```
CONTENTS:
---------

Shelly15.lha contains:
```

```
- shelly (the executable for C= Amiga,
          uses MC68020 and up & MC68881 and up )

- shelly.c, shelly.h (source & header, ready to compile on various
                      machines)

- examples.jpg (a picture of all examples)
               (except 'Lyria' all Shells rendered on my Amiga
                (A4000/030, 2C/2F) with Real3D2(Demo:)))

- Planorbis.shy, Nautilus.shy, Lyria.shy, Ammonite.shy
  Oxystele.shy, Natalina.shy...         (some example data-files)

- Shelly.guide (this document)

- Blank.shy   (a blank datafile)

- nodule.gif  (example-picture of the results of the new
               calculation-mode)

- sh.tcl   (a small Tcl/Tk GUI for shelly)
```

1.5 requirements

```
REQUIREMENTS:
-------------


Shelly requires atleast:

 (to use the executable provided)

  - an Amiga (harddisk & fast processor recommended)

  - ixemul.library (NOT! in this package)

  - POV-V2.0 or Real3DV2 to look at the results

  - or the TDDDlib-converters if you want Shelly to
    create objects for Imagine etc.

  Shelly has been tested on the following configurations:

  -A4000/030
  -A2000D+A2630
  -SUN4/10
  -IBM/RS6000
```

1.6 installation

```
INSTALLATION:
-------------

The installation of Shelly is very easy ...
Just copy the Drawer "Shelly" to a place
where you like to install it.

and give it a '(g)cc shelly.c -o shelly -lm -O2' (not needed on Amiga)
```

1.7 quickstart

```
QuickStart:
-----------

To get started quickly :

 - install Shelly (described in Installation)

 - open a shell (CLI), cd to the directory "Shelly"

 - type 'shelly Planorbis.shy xxx.pov'
   (Planorbis is one of the examples, xxx is the name of the
    POV-Scene Shelly will create)

 - now go and render the file 'xxx.pov'
   (e.g. 'pov -ixxx.pov -f +d' (assuming you have pov in your path))

 perhaps you have to edit the file 'xxx.pov' (camera position etc.)
 and try it again to get the best result...

For detailed information look into the Usage section.
```

1.8 usage

```
    Usage:
    ------

 General

 Datafiles

 Calculation modi

 Problems

 Hints
```

1.9 general

```
GENERAL
-------

just type

'Shelly infile outfile'

to run Shelly from a shell (CLI)

- infile is the (path+)name of a datafile
- outfile is the (path+)name of the POV/RPL/T3D/RAW/X3D outputfile

- outfile will be overwritten (if it exists)!

- Shelly will also open a file "(path+)outfilename.tmp"!
  and overwrite it (this is a temporary file, it will be
  deleted after calculation)
```

1.10 datafiles

```
DATAFILES:
----------

Shelly uses own datafiles in a simple format to get
the arguments into the program.


   Fileformat:
   -----------

The files are of a very simple (and easy to process :)) format:

- every line of the file is scanned for keywords.

- if a line contains no keyword it is treated like a comment

- if a line contains a keyword it is interpreted
  (the number behind the keyword is copied into an internal structure
   ("alpha:30" sets the internal alpha value to 30)
  or a flag is set
   (the 'RPL' keyword sets the flag "we_have_to_produce_an_RPL-file"))

- as you can see we have keywords that need parameters behind them
  and keywords that just have to be there to set something

- the only "Flag-keywords" the program knows are: 'POV','RPL','T3D',
  'RAW','X3D','NORMAL','NODULE','NEWNOD','RENDER'
  all other keywords need to be combined with a number
  (as the ':' states)

- everything is casesensitive! ('RPL' != 'rPl')

- the file is not checked for anything else
```

```
- double use of the same keyword causes an overwriting of the
  last set value

- lines like "alpha:Blafasel" will cause NO errormessage
  (there are no messages at all ;))


note: the keyword does not have to stand alone!
      if you write a line like:

      "/*RPL*/" or "BlafaseRPLl"

      the RPL-flag will be set! But you could also write:

      "render this in RPL pleazze :)"



There is a special file ("Blank.shy") prepared for you that is blank
(contains only some necessary keywords).

consider:

 - some parameters have to be given in degrees, some not
   (look into the algorithm section)

 - if you want Shelly to create a RPL file as output add a line
   like "RPL" or "pleazze do it in RPL" to the file
   ("T3D" will switch to T3D-output)
   (POV output is default)

 - be careful with the parameters, don't try to fool Shelly
   ("what does it do if I enter an infinite value :)?")
   it will end up in a mess or coredump or our beloved friend
   because the values are not checked!

   You should just change the given examples slightly until you
   know what you are doing...

 - smin,smax,sd,omin,omax,od are very critical parameters
   because they determine the size of the output and the memory
   consumption while calculating the shell

 - o must be positive! (omin >= 0, omax > omin, od > 0)

 - always remember:
   This program has still the status "experimental"!

 - there is a new calculation mode in V1.3 (Modi).
```

1.11 calcmod

```
CALCULATION MODI:
-----------------
```

There are several ways of calculating a shell (means dividing the surface of the shell into subfaces, triangles)

currently there are 3 methods implemented in Shelly:

The NORMAL-MODE:

Nothing special, just lay a grid (specified by smin,smax,sd, omin,omax,od) over the surface and calculate the knotpoints. Then connect the knotpoints to triangles, or use them to build a B-Spline-mesh.

The NODULE-MODE:

This is a new (changed in 1.5!!!) calculation mode that can save you object-data, up to 40%! (but only in shells with nodules!)

It is called "dynamic_stepsize" and can be invoked by using the NODULE keyword. (default mode is NORMAL)

How it works:

- imagine a shell with nodules (look at "nodule.gif"), if there is enough space between the nodules it might be possible to increase the stepsize (in o-direction) between the nodules without loss of information.

- this means high resolution is only used when needed (in the nodules) the rest of shell is calculated in a lower resolution

- Attention! In this new mode it is possible to choose a stepsize (od) that is bigger than the size of the nodules (W2) without fear of missing nodules ("jumping" over a small nodule with a big step) ALL nodules are calculated correctly.

- The Shell is scanned in (O-direction) for nodules (with a stepsize given via new keyword 'Scano:'). In a nodule Shelly will compute the height difference to the next line. If the height difference exceeds a threshold (keyword 'Hdo:'), the stepsize will be made smaller just that it exactly matches the height difference. The smallest possible step is limited by 'Scano:' too!)

- Default value for Scano: is 0.05
  Hdo: is 0.1
  this should be best for the most cases, but feel free to experiment, as smaller the Hdo: as finer are the nodules calculated.

```
   The NEWNOD-MODE:
   ----------------

This mode is very similar to the NODULE mode, except that it does
the same thing (refinement dependant on height difference) for
the S-direction too.(use 'NEWNOD' to switch to it)
This causes an irregular structure of the resulting grid, so that
output as B-Spline-mesh is impossible (the RPL-output will create
a bunch of triangles instead)!

The keywords for the NODULE mode ('Scano:','Hdo:') are used aswell
as two new ones ('Scans:','Hds:') in the same way.

- Default value for  Scans: is 0.05
                     Hds:   is 0.1

  this should be best for the most cases, but feel free to experiment,
  as smaller the Hds: as finer are the nodules calculated.
```

1.12 problems

```
PROBLEMS
--------

  - it is nothing to be seen in POV:
     probably the camera/light positions are wrong
     take a look at the data in your pov-file and correct this


  - POV tells me something from "degenerated triangles"
     well this problem did not occur yet (in shelly) but I know
     it could happen (former projects)
     nothing serious, just some triangles with 2 points the same

  - Imagine does not accept the Shelly-output
     convert the output via 'readwrite' from the TDDD-Package
     to the binary TDDD-format

  - Real3Ds annoying "Stack full" message comes up everytime
    a macro is executed:

     - change the RPL-stacksize (menu: Settings/RPL)
        (increase the "Parameter Stack")
     - open a new RPL-window
     - type: '"(path+)macroname" LOAD'


  - strange numbers (NaN's) occur in the output:

     Well this problem is known to me but no solution (sorry).

     Since the algorithm is somewhat complex I really don't
     want to have to find out which combination of which
```

```
      parameters cause this.

      It is also a problem of the sideeffects and (numerical)
      stability of the "mathematic" functions I call.

      note: I suppose zeros are the source of all this
            -> try to avoid them
```

1.13 hints

```
Hints
-----

for Real3D-users:

- remember that in a B-Spline-mesh the first and last line
  (and in each line the first and last point)
  of the mesh will be invisible (unless you switch objecttype
  to Polygon or Phong)
  this means for a shell with smin:10, smax:210, sd:20 that you
  will see a shell created from smin:30 to smax:190!
  (all examples will suffer from this if you just add the RPL
   keyword)
  solution: increase the ranges of s and o.


- if you want nodules in RPL-objects:

  You should choose proper values of od and sd to see the nodules
  at all
   (if you have nodules that are 10\textdegree{} wide (in o-direction) and ↩
       you
    choose an od of 40\textdegree{} you will see probably no nodules!)
   (this is also important for the POV-output)

  You should double the nodule height (L) for B-Spline objects
  to get the same height of the nodules as a POV-output!


- if you want to create a shell without nodules you may
  double the sd and od values for B-Spline objects without loss
  of quality in many cases
```

1.14 algorithm

```
The Algorithm:
--------------

In this section you will find more detailed information on the
algorithm used by Shelly and on the parameters it uses.
```

- The basic idea of the algorithm is to simulate a shell shape
  by rotating & moving (&copying) an ellipse (or a part of an
  ellipse, or any other curve (a cardiod)) around an axis.
  This will end up in some sort of spiral-shape.

- The shape produced will depend on many things like:
      -starting size/place/orientation of the ellipse
      -exact form of the ellipse (nodules)
      -how fast is the ellipse growing while rotating etc.

- you can find the exact formulas in the original article or
  in the sourcecode (too lazy to write them here again, they are
  very complex)

- here is a list of all parameters that shelly needs to generate
  a shell:

 -angular parameters (given in degrees):
  alpha   :equiangular angle of spiral
  beta   :angle between z-axis and line from aperture local
            origin to xyz-origin
  phi   :tilt of ellipse major axis from horizontal plane
  omega   :amount of azimuthal rotation of aperture
  my   :amount of "leaning over" of aperture

  smin   :angle at which aperture generating curve begins
  smax   :angle at which aperture generating curve ends
  sd   :stepsize in s-direction
  omin   :angle at which spiral begins
  omax   :angle at which spiral ends
  od   :stepsize in o-direction

  P   :position of nodule, in terms of angle s
  W1   :width of nodule in s-direction
  W2   :width of nodule in o-direction

 -linear dimensions
  A   :distance from main origin of aperture at o=0
  a   :major radius (long axis) of ellipse at o=0
  b   :minor radius (short axis) of ellipse at o=0
  L   :height of nodule at o=0

 -other
  N   :number of nodules per whorl

- the parameters smin,smax,sd,omin,omax,od determine
  how many triangles (controlpoints) are generated
  (how smooth is the shell and how many whorls are generated)
  -> be careful with these: memory usage and filesize depend
     directly on this parameters

- the parameters alpha,beta,phi,omega,my determine the orientation
  of the ellipse before (and while) rotating

- the parameters A,a,b determine starting place and size of the

```
  ellipse

- the parameters P,N,L,W1,W2 determine number,size and place
  of nodules
```

1.15 credits

```
Credits:
--------


- M.B. Cortie for his article "Digital Seashells"
- Martin Huttenloher for the icon of the guide
  (Thanks for MagicWB!)

Thanks to the people who ported GCC & CSH to the Amiga
and to Soulman (IRC) who helped me to realize the difference
between 2 and 2.0 ;).
```

1.16 distribution

```
DISTRIBUTION:
-------------

Shelly may be distributed FREELY via any media as long as:

1) The archive shelly.lha and its content remains unchanged.

2) No money (except a small copying fee) changes hand.



(Although Shelly is Freeware I won't reject gifts like
 money, chocolate, your latest piece of (gfx related) code etc..
 My adress can be found under "contacting the author".)
```

1.17 disclaimer

```
DISCLAIMER:
-----------

This program comes with no warranty, either expressed or implied.
The author is in no way responsible for any damage or loss that
may occur due to direct or indirect usage of this software.
Use this software entirely at your own risk.
```

1.18 adress

```
send
chocolate, money, your programs, bug reports (NOOOOOOO!:) etc.
to:

Randolf Schultz
Unter den Linden 51
19079 Mirow
GERMANY


INTERNET: rschultz@informatik.uni-rostock.de

          or (not regularly read)

          tfb512@hp1.rz.uni-rostock.de
```

1.19 keywords

```
The following keywords are supported:


'alpha:'
'beta:'
'phi:'
'omega:'
'my:'
'smin:'
'smax'
'sd:'
'omin:'
'omax:'
'od:'
'P:'
'L:'
'A:'
'a:'
'b:'
'W1:'
'W2:'
'N:'
'RPL' (switches to RPL-output)
'POV' (guess)
'T3D' (hmm)
'RAW' (simply the coordinates of the created triangles)
'X3D' (output for fast viewing in x3d on X11)

'NORMAL' (default calculation mode)

'NODULE' (switches to new calculation mode (dynamic stepsize)
          only useful when rendering shells with nodules!)
```

```
     'NEWNOD' (switches to new calculation mode (dynamic stepsize)
               only useful when rendering shells with nodules!)

     'Scale:' defines a scale factor for the shell (default is 1.0)

     'RENDER' switches preview on (Shelly will automatically call POV
              after calculation)
              ONLY available when output-type == POV! and
              'pov' must be in the search-path!

     'POVARGS:' defines arguments of the pov-call
                (default is "-f +d +w200 +h160")
                use of this keyword overwrites all default-arguments
                passed to pov!
                (be sure to specify a complete argument-string for pov)
                "-ixxx" is added automatically (don't use this!)

     'P2:'    P of second nodule
     'W12:'   W1 of second nodule
     'W22:'   W2 of second nodule
     'L2:'    L of second nodule
     'N2:'    N of second nodule
     'Off2:'  offset (in W2 (O) direction) between
              nodule1 and nodule2 (in degrees)

     'P3:'    P of third nodule
     'W13:'   W1 of third nodule
     'W23:'   W2 of third nodule
     'L3:'    L of third nodule
     'N3:'    N of third nodule
     'Off3:'  offset (in W2 (O) direction) between
              nodule1 and nodule3 (in degrees)

NEW in V1.5:

      'Scano:' stepsize for scanning the shell for nodules (in O-dir)
               & minimal possible stepsize! (default is 0.05)
      'Hdo:'   defines maximal height difference between two lines
               (default is 0.1)

      'Scans:' stepsize for scanning the shell for nodules (in S-dir)
               & minimal possible stepsize! (default is 0.05)
      'Hds:'   defines maximal height difference between two knotpoints in
               S-dir
               (default is 0.1)


      'camx:'
      'camy:'  x,y,z position of the camera for POV-output
      'camz:'


     (note that the ':' belongs to the keyword!  you can use
      for instance the word 'alpha' with no risk in comment lines)

     (The meaning of a special parameter (keyword) can be found
```

in the section about the algorithm.)