

F 20

NOTES - starting 8/25/77

ending 10/11/77

VERNON
VR
ROYAL

Sandy Ressler

828-2318

RPO 7065

Rutgers College

New Brunswick N.J. 08903



COMPOSITION BOOK

60 SHEETS • 10¼ IN. x 7⅞ IN.

AVAILABLE AS:

No. 1148	(09-9142)	1½" WIDE RULED - NO MARGIN
No. 1148-CM	(09-9144)	¾" COLLEGE RULED - 1¼" MARGIN
No. 1148-¼	(09-9148)	QUAD. RULED 5 SQS. TO INCH
No. 1148-½	(09-9146)	PLAIN
No. 1148-¾	(09-9140)	1½" WIDE RULED - 1¼" MARGIN

Made in U. S. A.

VERNON ROYAL, INC. Elizabeth, N. J. 07208

8/25 Dance

Ref Points

create the sphere of 27 points for referencing the major positions

~~Don't want~~

only one sphere has to be in existence and we have a matrix with radii and positional info which tells where the sphere is to be located.

Symbolic positioning of body



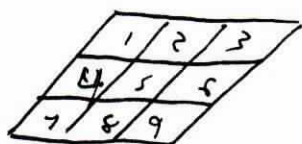
Place cursor inside one of circles to pick bp // what about groups ie legs//



keep on picking and ~~create pick group~~

signal that your finished when put cursor in finish box or finished if don't pick after 5 seconds.

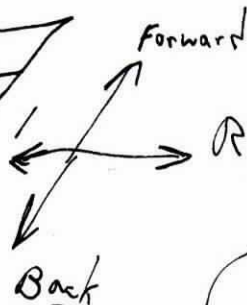
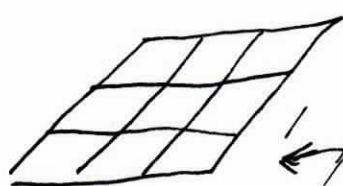
continue box used to keep positioning last byt



Place cursor
in direction box



(The 27 ~~dir~~
move directions)



Depends of kp

Picking a direction pick a ~~place~~ place in ~~a~~ a matrix which corresponds to a point on sphere.

Then bp is rotated $x y z^\circ$ until that point is end point of bp .

The ATTACH routine The rotate next

bp (if group)

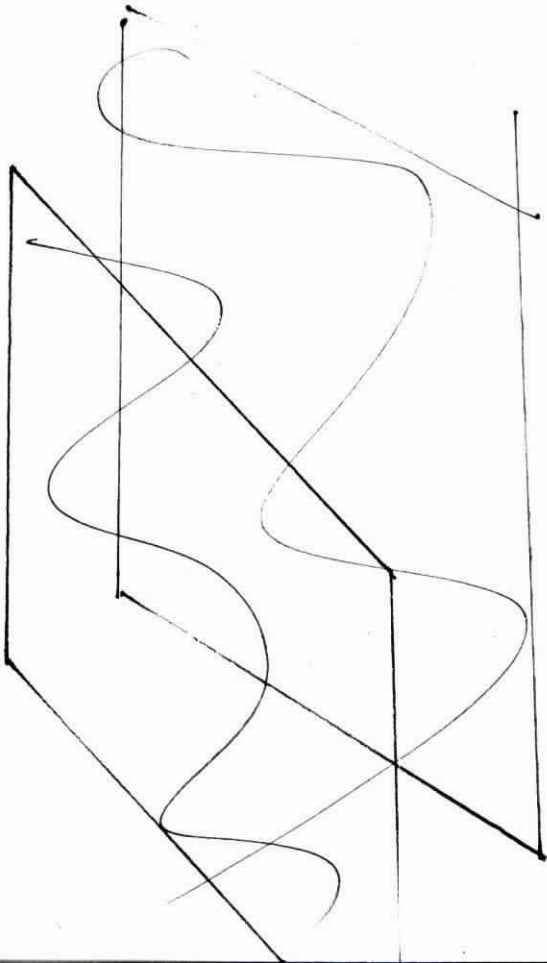
~~The storage of $x y z$ words~~

In early stages draw the 27 grid and let person type

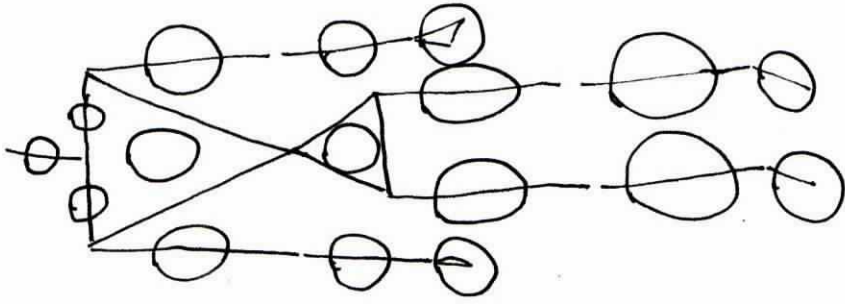
it # manually

Screen

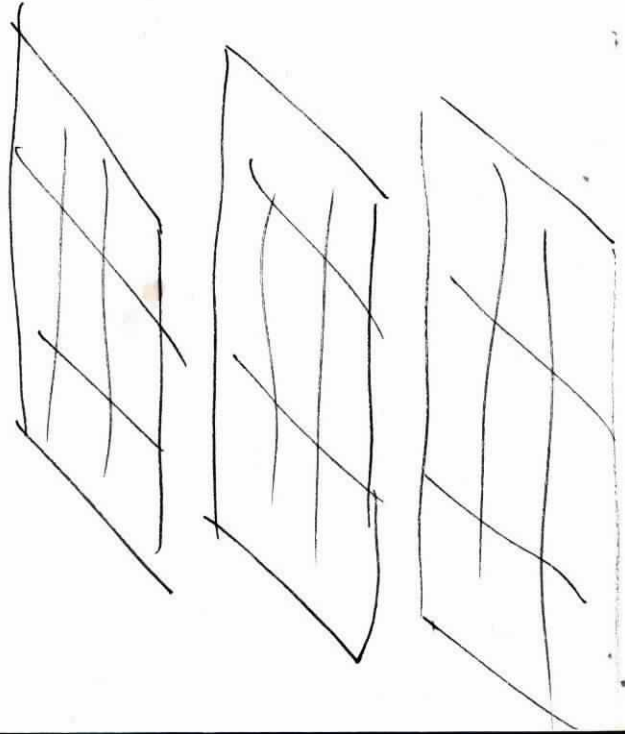
Blinking light
OK



Blinking light
To indicate what to do



Finish



8/21

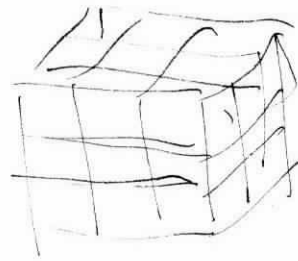
Dance

Ref points

examples

Sphere [1] = ~~High Left~~
High Left Forward

Sphere [2] = High Forward



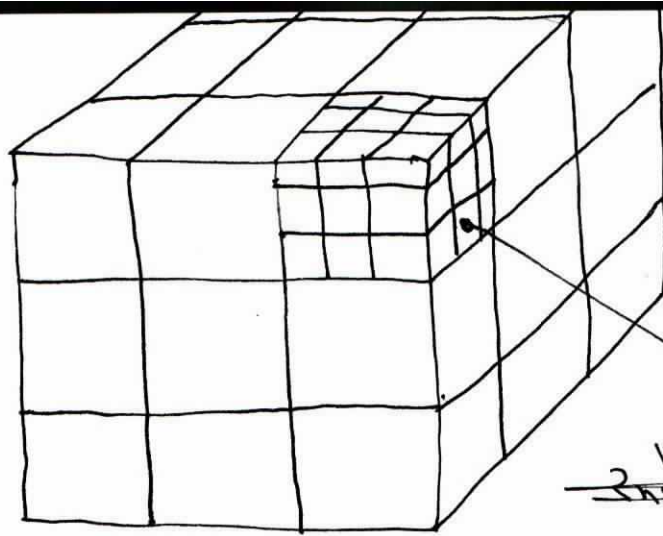
This is the index # which must be obtained

what about more precise positioning
could say higher lower left right

These could be 5° increments,
or maybe could be done by moving
the ^{correct} ref point // maybe by rot // a little
remember the sphere must already be oriented
in the right direction so forward is forward
no matter how the body moves.

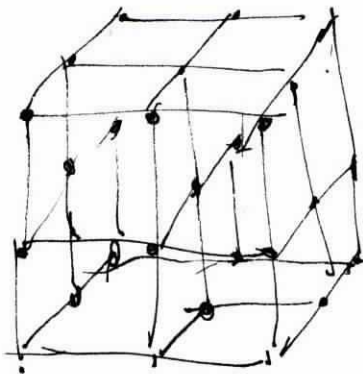
so higher could be 5° rot around rot line & sphere points
+ left
+ ...

8/28

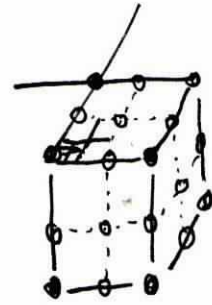


secondary directions

~~2nd degree~~




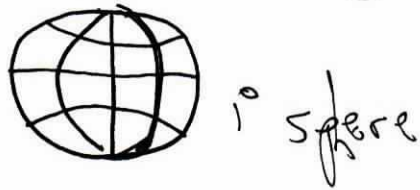
correct cube
 with 27 points
 Not 27 volumes



$\frac{27}{125}$
 $\frac{27}{125}$
 $\frac{27}{125}$

1° directions based on 45°
 2° should be 22.5°
 3° 11.25
 4° 5.5

Don't forget  for a reference point



for 2^o adjustments

Just say up or down
left or right

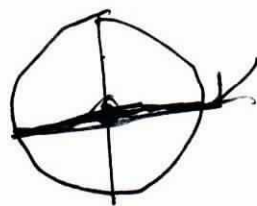
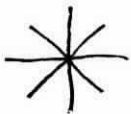
then

"a little"

or

more or "a little more"

Problem is to find the rot line (two of the 26 points)



up or down rot line
for center/forward middle



~~all up or down's are around~~

up or down's on same ring are around
same rot line therefore only 8 up down rot lines
and 8 left right rot lines

need program to determine which rings pt is on
(vert
*
hor)

then reference to the rotline

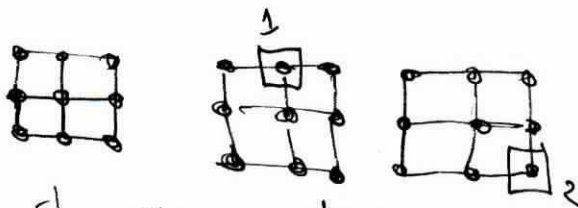
pts on same rings probably go by increments of 3 or 9
program would be something like:

IF pt is = 1 or 4 or 7 or ... Then got to rotline 1
else if pt is 2 5 8 go to rotline 2

rotline 1. if adjustword = "up" then sign is '+' else '-'
if
"down"

8/29 if quantity is "a little" then angle = 5° else
if "more" then angle = 10° else
15° default
or
20°

In Labanotation pins are used to show $\frac{1}{3}$ way toward neighboring
pins are 45° also and signify $\frac{1}{3}$ towards
the major sign pointing towards.
pin is toward
sign is from



Show the present location

Then neighboring locations for the $\frac{1}{3}$ adjustments

1

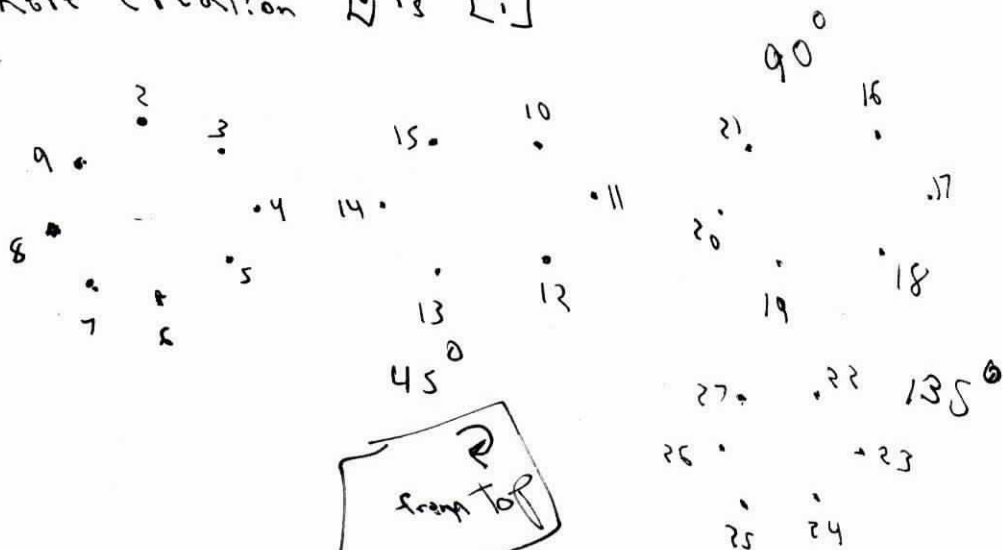


2



Would need algorithm to find adjacent points. And could then indicate $\frac{1}{3}$ or $\frac{2}{3}$ towards any of these points.

Back to ~~primary~~ primary directions.
sphere creation [] is []





1		place middle
2		place high
3		left high
4		left middle
5		left deep
6		place deep
7		right deep
8		right middle
9		right high
10		left forward high
11		left forward middle
12		left forward deep
13		right forward deep
14		backward middle
15		right backward high
16		forward high
17		forward middle
18		forward deep
19		backward deep
20		backward middle
21		backward high
22		right forward high
23		right forward middle
24		right forward deep
25		left backward deep
26		left backward middle
27		left backward high

4/30 New Ref system

The rotations involved are all 45° or multiples of
over the x y z axes



shortest path to destination
are there. only ~~2~~ rot lines
one x one y one z

x axis	points	4 → 8
y		2 → 6
z		17 → 20
AXES		

~~assign~~ assign levels of value for each direction



absolute value of
difference (where are)
to
where going)
 $\times 45^\circ$ in that axis.

each point has an x y z level value ~~value~~

sphere point	x level	y value	z
2	3	5	3
3	4	4	2
4	5	3	2
5	4	2	3
8	3	1	2
7	2	2	2
8	1	2	3
9	2	4	2
10		4	
11		2	
12		2	
13		2	
14		2	
15		4	
16	6	4	2
17	6	6	1
18	6	2	2
19	6	2	4
20	6	6	5
21	6	4	4
22		4	
23		6	
24		2	
25		2	
26		6	
27		4	

this is an x y z sort from low to high



my institute Tech

21 2 8 9 5 9 8 7 5

~~scribble~~ X

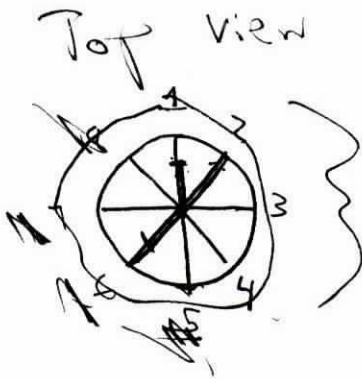
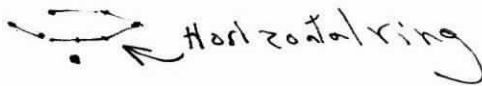


movement within ring involves
one rotation

first check if on ring

if not ~~see~~ get it there } how

↓
by a rot ~~to~~ around 245



The 4 vertical rings

check if in same vert ring
by checking a list

Example

At ring 1

want to move to ring 2

∴ rot 45

~~1st~~ 1st check if
This is closest way

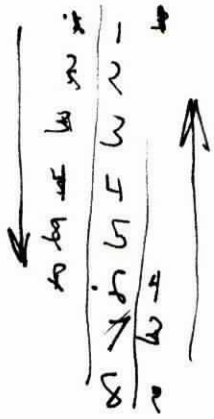
$$\text{ring}_2 - \text{ring}_1 = 1 \times 45^\circ$$



~~by ~~rot 45~~~~

~~↻~~ - want to move to correct side of ring

go from 1 to 8



shorter going backwards

▽ # function #

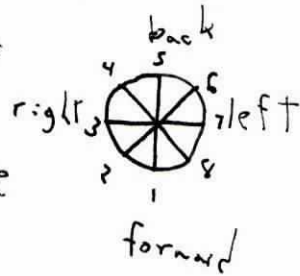
① ~~at~~ identify which of 8 rings pt are on
Table look up.

default

~~sphere, count,]~~

pt	ring
16 17 18 243	1
22 23 24	2
7 8 9	3
13 14 15	4
19 20 21	5
25 26 27	6
3 4 5	7
10 11 12	8

② if on different rings
determine shortest route
to rotate



▽ Fn Pts

8/31

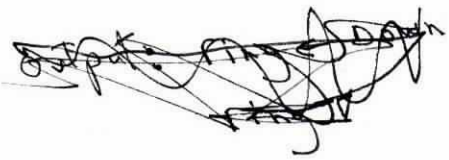
Table to find which ring pt is on

~~8~~ 8 5 16 17 18 21 3 22 23 24 0 0 ...
 ptring ↑ should be 6

Function To determine which rings 2 points are on

start: ~~ptring~~ Down side ← 1
 Ring find pts down; side
 ID ← 1

Found: → (pt = ptring [Down; side]) / output
 → (s ≥ side ← side + 1) / found
 side ← 1
 → (8 ≥ Down ← Down + 1) / found
 'Something Done gone wrong'



Output: → (ID = 2) / RR
 Ring1 ← Down
 ID ← ID + 1
 → Start
 RR: Ring2 ← Down
 Ring1, Ring2

From 1 To 8

3 → 5

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

Forward direction sign is +
 From Route To
 Get ringvec & ringvec ← 1 2 3 4 5 6 7 8
 count ← 1

ringvec ← (From) ringvec

For find: → (To = ringvec[count]) / Forward
 count ← count + 1
 → for find

~~Forward: Fang ← 45 × count~~

Forward: fang ← count
 ringvec ← 1 2 3 4 5 6 7 8
 count ← 1
 ringvec ← To ringvec

Back find: → (From = ringvec[count]) / Backward
 count ← count + 1
 → Back find

Must reverse signs

Backward: Bang ← count

→ (Fang < Bang) / Pos
 Ang ← -45 × Bang
 Pos: Ang ← 45 × Fang

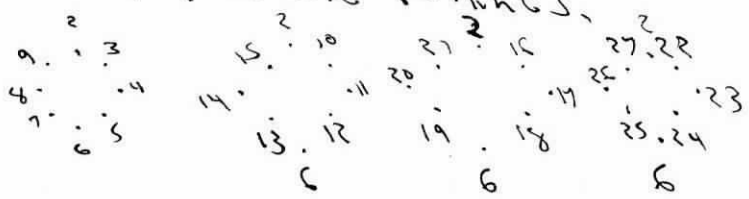
→ (Fang > Bang) / Neg
 Ang ← 45 × Bang
 Neg: Ang ← -45 × Fang

check this

9/11

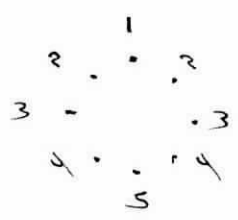
Have now rotated limb to correct side of ~~ring~~ ring
 and must now rotate to correct pt around one of

4 possible rot lines.

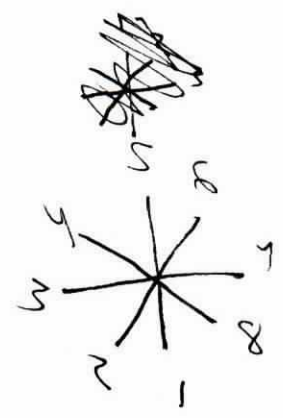
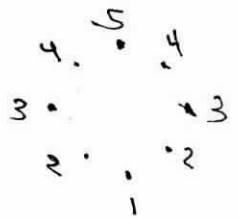


R — L

need to renumber points for this procedure



for all or possibly rings



conversion Table

1	2	2	2	2	2	2	2	2
2	9	3	15	10	21	16	27	22
3	8	4	14	11	20	17	26	23
4	7	5	13	12	19	18	25	24
5	6	6	6	6	6	6	6	6

contab

scan matrix for both points (going from \rightarrow To)

Difference of code values $\times 45^\circ$

sign should work out if 4 code is on correct side.

Must also find rot lines
 only 4

ring# sphere27 code pts.

1	14 11	8 4
2	14 11	
3	17 20	
4	23 26	
5	14 11	8 4
6	14 11	
7	17 20	
8	23 26	

change ptring[i;5]
to
6

8 2 p 8 4 14 11

ring#
rotcode

~~sphere27~~

~~rotcode~~ rotcode[ring1;
or
ring2

Line ← (sphere27 [rotcode[ring#]; 2 3 4])
 YES → { 8 4; 2 3 4 } if so ↗
 ↘ does this index work

if not The 2 3 4 elements of 8 4 rows

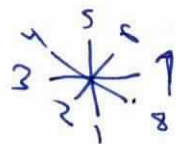
~~Line ← sphere27~~
 count ← 1
 Line ← so
 getline: Line ← Line, sphere27 [rotcode[ring#; count]; 2 3 4]
 → (2 ≤ count ≤ count+1) / getline
 Line

To find correct # of degrees over on ring.

Scan contab for both pts i.e 7 & 9

first identify where at on ring

need IO Table from ring to ~~ring~~
PT to PT



8

from
PT

	ring							
	1	2	3	4	5	6	7	8
2	2	2	2	2	2	2	2	2
3	16	22	9	15	21	27	3	10
4	17	23	8	14	20	26	4	11
5	18	24	7	13	19	25	5	12
6	6	6	6	6	6	6	6	6
7	18	24	7	13	19	25	5	12
8	17	23	8	14	20	26	4	11
9	16	22	9	15	21	27	3	10
10	16	22	9	15	21	27	3	10
11	17	23	8	14	20	26	4	11
12	18	24	7	13	19	25	5	12
13	18	24	7	13	19	25	5	12
14	17	23	8	14	20	26	4	11
15	16	22	9	15	21	27	3	10
16	16	22	9	15	21	27	3	10
17	17	23	8	14	20	26	4	11
18	18	24	7	13	19	25	5	12
19	18	24	7	13	19	25	5	12
20	17	23	8	14	20	26	4	11
21	16	22	9	15	21	27	3	10
22	16	22	9	15	21	27	3	10
23	17	23	8	14	20	26	4	11
24	18	24	7	13	19	25	5	12
25	18	24	7	13	19	25	5	12
26	17	23	8	14	20	26	4	11
27	16	22	9	15	21	27	3	10

inside
are
the
to pts
or
where are
on the
ring

PT to PT

	ring 1	2	3	4	5	6	7	8
Top	16	22	9	15	21	27	3	10
Bot	17	23	8	14	20	26	4	11
	18	24	7	13	19	25	5	12

Am at rings $pts[r]$
 so it is

Am at
 $pt \leftarrow pt_{opt} [pts[r]-1; rings]$

how scan contab to convert to
 rot coordinates

Down \leftarrow side \leftarrow 1

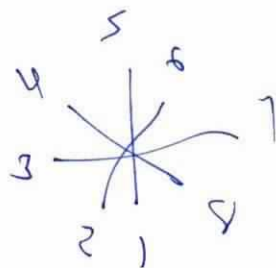
Found: $\rightarrow (pt = [Down; side]) / out$

$\rightarrow (y \geq side \leftarrow side + 1) / Found$
 side \leftarrow 1

$\rightarrow (y \geq Down \leftarrow Down + 1) / Found$
 something wrong

OUT P: ~~contab~~ $[Down; side]$
 rot pt \leftarrow Down

~~rotline~~



~~rotline~~

test Example

RingFind 22 5

Ring1 = 2

Ring2 = 1

Ring1 Route Ring2

Ang ← -135

Am now on correct ring and must determine where on ring I am and where I want to go Then convert into rotcodes Then compute angle of rot and Then find line of rot

Endrot
~~Rot~~; AT; TO; Downside; Examp; ID; Rotat; Rotto

[1] ~~AT~~ ← pttopt[pts[1]-1; ring2]

[2] To ← pts[2]

Must now convert AT and TO
(3) (5)

into rotcoordinates,

scan contab to do this

~~A~~ ID ← 1

Exampt ← AT

Start: Down ← side ← 1

Found: → (Exampt = contab[Down; side]) / Output

→ (8 ≥ side ← side + 1) / Found

side ← 1

→ (5 ≥ Down ← Down + 1) / Found

'something done gone wrong'

~~Output: ROTAT ← Down~~

~~Exampt ← TO~~



Output: → (ID = 2) / R2

ROTAT ← Down

Exampt ← TO

ID ← ID + 1

→ Start

R2: ROTTO ← Down

RotAng ← (ROTAT - ROTTO) × 45

// Signs may be backwards //

have to check on Terminal

RotLine ← sphere27[rotcode[ring2[j]; 2 3 4]

Bodypart is rot Ang, sphere27[2 6; 2 3 4]

Then

rot RotAng, RotLine

need a linear interpolation

of both rotations so they occur simultaneously on filon.

need programs to move spheres to base of
each limb and scale to correct size.

BodyP is the active bp
sphere 27 must be moved ~~to~~ so $\text{sphere } 27[1:] =$
 $\text{BodyP}[1:]$

A Function
in the
Scr Ws?

Sphere should be scaled first

Dis BodyP

Dist, Dist, Dist Abscale ~~BodyP~~ Sphere 27

$\text{Sphere } 27 \leftarrow (\text{BodyP}[1:234] - \text{Sphere } 27[1:234]) \text{ Move Sphere } 27$

9/2 Order of Action

Body is assigned a new value
scale then Move Sphere 27

Determine which sphere pt presently at

▽ Locator
count ← 2

Search: $\rightarrow (\text{body}[2; 2 \ 3 \ 4] = \text{sphere 27}[\text{count}; 2 \ 3 \ 4]) / \text{Found}$
 $\rightarrow (27 > \text{count} \leftarrow \text{count} + 1) / \text{search}$

Found:
Loc ← count

Ask for where ~~to~~ I want to go

In Go program

Angask: Where do you want to move to, (Level, Lock, For, Back, Cent)

~~draw grid~~

create a char. matrix of the 27 direction in
order corresponding to sphere 27 pts,

~~Major~~ Major
 Abbreviated 3 letter

Translation.

Not a good way some can be only 2 letters

1	MPC	Middle Place center	PC
2	HPC	High Place center	PH
3	HLC	High Left center	HL
4	MLC	Middle Left center	ML
5	LLC	Low Left center	LL
6	LPC	Low Place center	PL
7	LRC	Low Right center	LR
8	MRC	Middle Right center	MR
9	HRC	High Right center	HR
10	HRF	High Right Left Forward	
11	MRF	Middle Right Left Forward	
12	LLF	Low Left Forward	
13	LRB	Low Right Backward	
14	MRB	Middle Right Backward	
15	HRB	High Right Backward	
16	AMF	High Middle Forward	HF
17	MMF	Middle Middle Forward	MF
18	LMF	Low Middle Forward	LF
19	MMB	Middle Middle Backward	LB
20	HMB	High Middle Backward	MB
21	HRF	High Right Forward	HB
22	MRF	Middle Right Forward	
23	LRF	Low Right Forward	
24	LLB	Low Left Backward	
25	MLB	Middle Left Backward	
26	HLB	High Left Backward	

Better Ab.

Can use both
 3 & 2 Abbreviations

Angask: where do you want to move?

MoveTo ← \square

count ← 1

MoveTo ← MoveTo, (3 - PMoveTo) P' 1

Search

→ (MoveTo = MapOr(count, j)) / Found

→ (27 ≥ count ← count + 1) / search

'please Reenter a Location'

→ Angask

PTT0 ← count

RingFind

~~PTT0~~
Loc PTT0

Ring? Route Ring?

9/3 need to restructure the go program so it makes more sense.

1 Ask for command then → appropriate spot
view Reset Help Name + Stop are no problem.
Position is the best one.

ask for Name
~~and obtain it~~ and create a stack
for group bps

Getbpname

~~Getname~~: 'what is the Body Part'

Name ←

count ← 1 ~~count~~

Bnamelen ← 1 + BPNames

Name ← Name, (4 - pName) p' /

scan: → (4 - (t/Name = BPNames[count:j])) / ~~scan~~ ^{count hold}

→ (Bnamelen ≥ count ← count + 1) / scan

'please reenter a body part'

→ Getbpname

count hold: chold ← count

→ Getpos

▽ BPInput; count; Bnamelen

Next go to the pos: input

Getpos: → Pos: input

• where do you want to move (Left, Right, Forw. Back.)

Now know name of bp (not if a group)
and where want to move to

18



~~set find~~

now find if a group + all assoc values.

~~if count~~

~~if child > 18 bp is a group else~~

go on

↓
find values
and and put in a stack

9/4

stack ← 0 4/0

count ← 1

Lim ← +/pointers [child;] ≠ 0

GetStack

stack ← p, Bpnames [pointers [child; count ↑ Lim]

Joistack ← p, Joinames [pointers [child; count ↑ Lim]

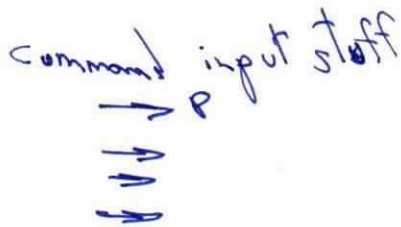
stack ← stack cat stack
Joistack ← Joistack cat Joistack

(check if order of
stacking is correct)

⇒ (Lim ≥ count - count + 1) / GetStack

▷ StackPr▷

9/2 what is BP
 LF what is Pos
 HLF



[] BP Input

[] Pos Input

[] Stacker A creates the BP value stack and Joint stack A

[] Activator need to make Body the active bp.

[] Rotsh: ^{Locator} Dis Body

Sphere 27 ← (DI, DE, DI) ABSCALE SPHERE 27

Sphere 27 ← (Body [1; 2 3 4] - Sphere 27 [1; 2 3 4]) Move Sphere 27

Sgpts ← Loc, PITTO

Ring find Sgpts

Ring 1 Route Ring 2

Body ← Body Rot Ang, Spheres [2; 2 3 4], Spheres [6; 2 3 4]

Ex ↓ Rot

Body ← Body Rot RotAng RotLine

ATTACH // check if more body can be activated else → AG.
 → Activator Line.

Activator

~~Lim~~ Lim ← 1 ↑ P Stack

GetBody: Body ← 2 4 ↑ P Stack

~~Lim~~ counter ← 2

Joint ← 1 6 ↑ counter 0

Stack ← -(Lim-2) 4 ↑ Stack

Activator

Lim ← 1 ↑ P Stack

→ (Lim=2) / End

Getbody: Body ← 2 4 ↑ Stack

Joint ← 1 6 ↑ Joistack

~~Stack~~
Stack ← -(Lim-2) 4 ↑ Stack

Joistack ← -((Lim-2):2) 6 ↑ Joistack
→ 0

End: → ~~Display~~ Display

Disp: Ground
Draw Body

Problem - not all bp's are
2 deep so stack will be screwed
up

bp's should be 2 deep and
extra lines for hand + Torso
should be mapped on

Com:
 → Act:
 ∇ Com Input
 [1] AG: 'COMMAND:'
 [2] ~~INP~~ INP ←
 → (1P 'P')/PCOM
 → (V)/VCOM
 → (R)/RCOM
 → (S)/O
 → (H)/HCOM
 → (N)/NCOM

∇ RC
 [1] COM: COM Input
 [2] VCOM: SEE
 → COM
 [4] RCOM: Reset
 → COM
 [5] HCOM: HELP
 [7] → COM
 [8] NCOM: 'What... ..'
 [9] [11]

[9] ~~AG~~ 'Please reenter a command or Type Help'

[10] → AG ∇

VCOM: SEE
 → COM

RCOM: Reset
 → COM

HCOM: ● ∇ HELP
 [1] 'Do you Type B'
 [3] HINP ←
 [4] → (Hinp = 'B')/BHBL
 [5] 'THE COMMANDS'
 [6] COMHELP
 [7] → COM
 [8] BHBL: 'The Body Parts'
 [9] BPHELP ∇

~~RCOM~~ → ~~COM~~

→ COM

NCOM: 'What is the name for this present position?'

Posname ←
 ≠ Posname ← 'Body'
 → COM

ActiveMap

Need a Mapping program maybe 2 which will
map stuff onto the skeleton
like the extra lines that make the
hand & torso a triangle.
(get an index from stacker)

~~Mapper~~

Take expression
pointers[hold; -1] ... whatever

Take will yield a 1 through 18 index
is in the ~~x~~ level of a 30 matrix
and display it.

Reset would reset the map values also.

MAPMAT $\leftarrow 18 \times 5 \times 4$

Mapper

Index \leftarrow pointer [.....]

ActiveMap \leftarrow MAPMAT [Index \leftarrow]

Then whenever ~~body part~~ body part is rotated say

ActiveMap \leftarrow ActiveMap rot Ang ^{rot line} ~~rot line~~

Must also attach Active Map to correct place
may require an AttachMap program
Just like the original Attach program

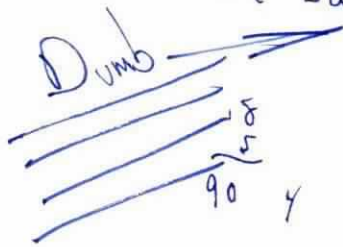
9/7 Mapping

MBBname

M B B Name for the saved version for resetting

at present only 3 maps LH RH ~~SP~~

for the final positions the mapped portions should also be saved so



cat Body ← Body cat ~~Map~~

Map is Mapmat reshaped to be

90 4 To much waste

Instead of 1 big 3D matrix have 18 Taylor made 2D matrices.

Problem is in referencing them

Map001

Map002

~~0-39~~

Index is a no. from 1 to 18

ActiveMat ← Map, \neq Index

ActiveMat ← \neq ActiveMat

→ (Ring 2 = 1 or ~~2~~) / L1 8 4

▽ Linefind

→ (Ring 2 = 2 or 3) / L2 11 14

3 7 / L3 17 20

4 8 / L4 ~~23 26~~
25 23

L1 RotLine ←, sphere → [8 4; 2 3 4]

→ 0

11

~~→ (Ring 2 = 1), (Ring 2 = 2), / L1, L1, L2~~

L2:

The Joint Stack is useless
not to mention Linefind (almost)
for now

→ ComInput

Vcom

Hcom

Scom

.....

Pcom: BPIInput

Posinput

//PTTO determined//

Stacker

RolsL: ~~ACT~~

Activator

→ (ActLim=2) / Display

~~ACT~~

Dis Body

Move & Scale Sphere27

Locator // Find present pt pos

spts ← Loc, PTTO

RingFind spts (Finds present and going to rings)
through string

Ring1 Route Ring2

9/8 Painting

Picasso's ~~art~~ simultaneity of views



Take a generalized face and put it on computer



split up various portions
eyes nose lips at random and put together

Maybe result could crudely simulate some simultaneous figures like Picasso. Then Paint it.

9/9

Status: At this point the dance shit seems to work only for single bp's not groups like LA or LL.

So something isn't rotating or looping or checking somewhere.

Pain in the
Balls

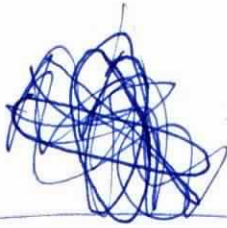
Must be consistent as to what is left and what is right

will be using the figures left and right and it will face front (for default)

Must also start to convert to the better perspective system.

9/10

Maybe Try to rotate the bp's in one shot
by knowing where are & where going
get ~~the~~ angle between those two lines
The finding line of rot (maybe by a table look up)



check out all of The modules, thoroughly
and if they work then problem is in how put
together ~~→~~ ~~→~~ ~~→~~

Bringup

child is the # of use gotten

so for LH

child should be 9

oBod should be 27

Pos: input

PTIO is # word from this

PA — PTIO should be 2

PL — PTIO should be 5

~~PTIO~~

MBB — 14

Stacks — create a big stack of values

in a stack of LL

stack should put LT on top

Then LC

" LF

Activator

should assign Body the e 49 stack

and pop the stack

D: 5

Find length of body

Locator

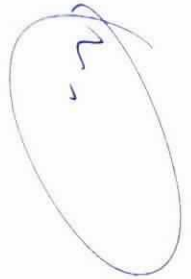
Find location of body [2j] in sphere 27 pts

For LF

Loc ← 17

For LS Loc ← 8

LA ← 6



Ring find PT1, PT2

Determines the rings 2 points are on

PTs 2, 4, 3 ⇒ Ring 1 ← 1

Ring 2 ← 7

PTs 17, 25 ⇒ Ring 1 ← 1

Ring 2 ← 8

PTs 10, 22 ⇒ Ring 1 ← 8

Ring 2 ← 2

$BPName[Pointers[chold; PLHold[i]]]$

was always LL

when

LL was input

$chold$ obtained in $BPinput$

$PLHold$ must be incremented

PL was 3

$ACi: Lin \leftarrow + (Pointers[chold;] \neq 0)$

\downarrow $BPNames$

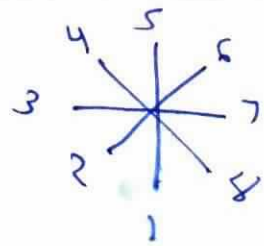
~~$\rightarrow (PLHold = Lin) / ACi$~~

$\rightarrow (Lin = PLHold \leftarrow PLHold + 1) /$

MB position is screwed up

Route

finds angle bet rings



(8 Route 2) 90

(4 Route 3) 135

(3 Route 8) -135

(2 Route 6) 180

(1 Route 4) 135

~~EndRot~~
EndRot

AT ← LOC

To ← PTS [2]

should be To ← SUPPL

AT and To should be on same ring

2 4 6 are on all rings

(7 - 8) 45°

(9 - 7) -90°

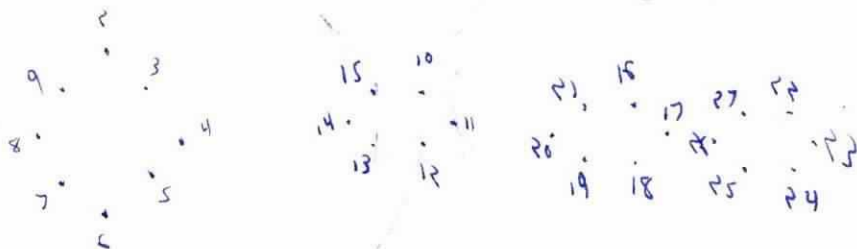
(5 - 8) 90°

(2 - 6) 180

(2 - 12) 135

Make table which will convert pts ~~to~~ 2×6
 into correct #'s so when you say
 (AT - T0) x 4s it'll work

9/11

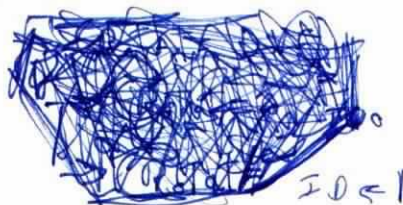


convertes

ring#	2	6
1	15	19
2	21	25
3	10	6
4	16	12
5	22	18
6	28	24
7	2	6
8	9	13

Put in Endrot

R2: RotT0 ← down



ATT: Test ← Rotat
 → check

ToT: Test ← RotT0

check! → (Test = 2) / Two

→ (Test = 6) / Six

~~Normal~~

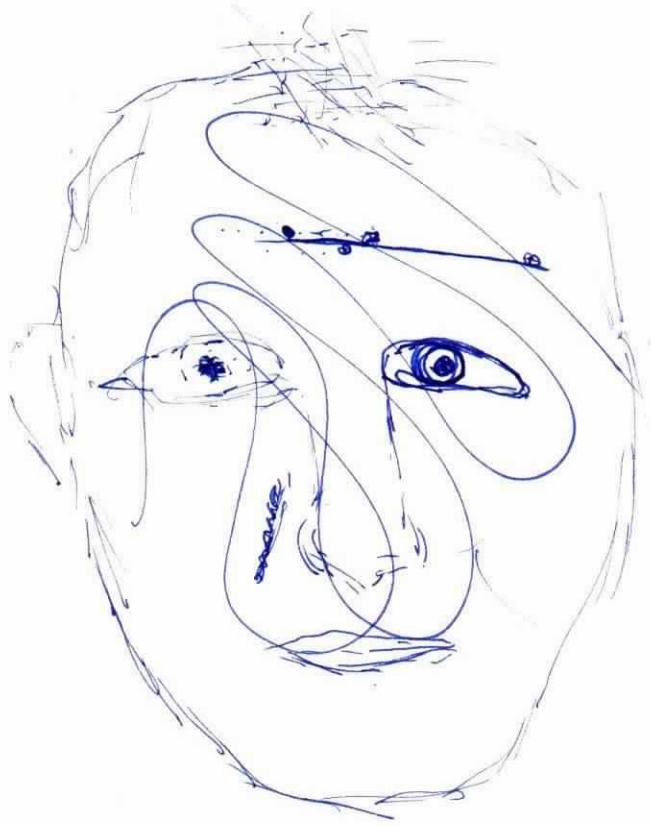
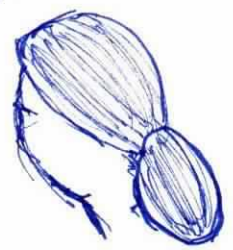
~~Dot~~

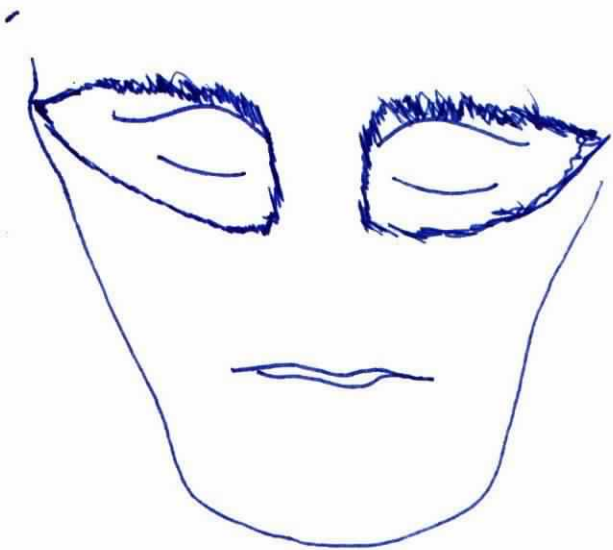
→ (~~Rot~~ ID ← ID + 1) / Tot

ring#

ring#; ID

~~Two Rot~~





9/14

R2: Roto ← Down

ID ← 1

index ← 1

Att: Test ← Rota

→ check

Tot: Test ← Roto

check: → (Test = 2) / Two

chs: x → (Test = 6) / Six

→ (R ≥ ID ← ID + 1) / Tot

Two: Test ← convertes [ring 1; index]

→ (ID = 1) / chs: x To ← Test

~~chs~~

R ← AT checkes To

Six:

~~(Test = 2), (Test = 6)~~

⇒

convertes

At:

Index ← 1

At: Test ← Rotto

→ check

check → (Test = 2) / Two

→ (Test = 6) / Six

→ (Index = 2) / Tot

Two: ~~Test~~ ← convertes [ring 1; index]
index ← ~~Index = 1~~
→ Tot

Six: ~~Test~~ ← convertes [ring 1; index + 1]

check → ~~Rotto = 2~~ / Two A
→ ~~Rotto = 6~~ / Six A
→ cTo.

Two A Index A

Rotto ← convertes [ring 1; index]

→ cTo

Six A: Rotto ← convertes [ring 1; index + 1]

cTo: → (Rotto = 2) / Two B

→ (Rotto = 6) / Six B

→ 0

Two B: Rotto ← convertes [ring 1; index]
→ 0

Six B: Rotto ← convertes [ring 1; index + 1]

9/13

Need To Test if all direction work

Pos

Acts

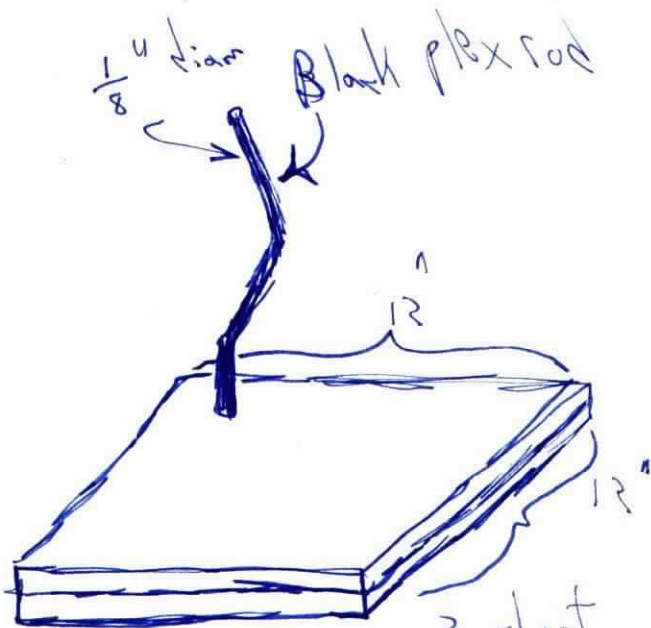
DCT = 10

Pattern of bugs
1st time it doesn't work
second time in row it does
3rd is ok

DCT = 5 For

PL	ok
PH	can't use blow's up
PH	OK
HL	OK
ML	OK
LL	OK
LR	OK
MR	No Good Like PL 2nd ok
HR	No Good Like HL 2nd stick at LR
HLF	OK No PH Like PH
MLF	No Good Like LLF
LLF	OK
LRB	OK No 1st 2nd ok
MAB	from PH → then ok
HRB	Like LLF different on subsequent
MMF HF	Like MF consistent but to →
MMF MF	OK
LMF LF	OK
LMB LB	OK
MAB MB	at PL changes to MB From PL → MF
HB	No Good changes to HB
HRF	OK
MRF	No 2nd time ok
RF	consistent MRF
LLB	No 2nd ok
MLB	OK OK
HLB	No MLB but not quite exact consistent

9/15 Sculpture



2 sheets clear plex
drawing goes between two

Top sheet thin $\frac{1}{16}$ " or less

~~Bottom sheet~~

Bottom thick $\frac{1}{2}$ " $\frac{1}{4}$ " if possible

Too expensive?

~~range~~

9/17 Dance

Need to trace various variable and

→ find where things are going screwy

If the locator & ringfind are working

then route is giving wrong ~~←~~
endrot

They seem to be working ok separately

so put together messed up

9/18

Make the AP2PLOT routine (in Fortran) interactive
with the following options:

of plots

size

color

space between successive plots

should eventually check into lettering and
using all those internal Calcomp symbols.

9/19 Got the 5.0.B To work

by saying

Body [2;] \rightarrow sphere 27 [0T0;]

not dealing with the rotations for the moments
ie CHEATING.

need To be able to move sp

pointers [27;] is UBOB


change order To move 2 (sp)
To back

1 3 4 5 7 9 6 8 10 2 0

will this work?

~~that~~ tink so

sphere 27 is now an ~~an~~ absolute coordinate
system
ie forward is always towards me.





From Rubens

Two Satyrs

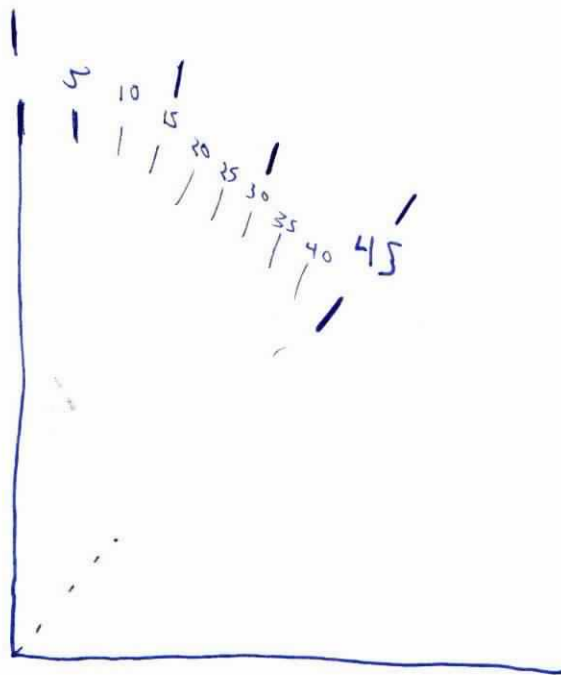
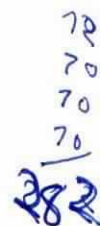
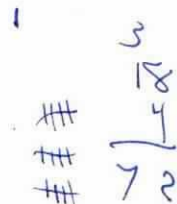


From Rubens
Adoration of
The Magi

9/20

Need a way to refine the positioning of figure

create a sphere with 5° increments that means
282 pts



make a diagram like above plus

ring1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0

wrong
there would have
to be 70 rings.

pick out data with cursors

using C in follow

(Maybe for principle directions too)
ip/all

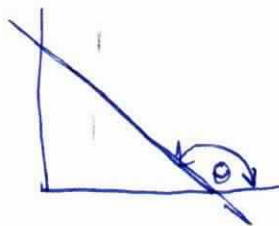
9/21

$$m = \tan \theta$$

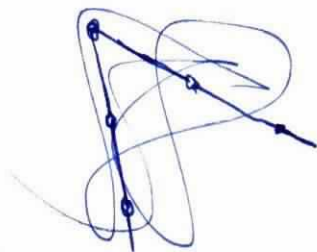
$$\text{slope} = \tan \theta$$

$$\text{if slope} = \frac{\Delta y}{\Delta x}$$

what about **Z**



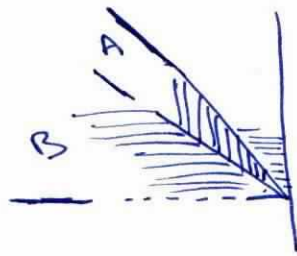
given ~~two~~ arm positions



with a basic bp there maybe a more necessary before the rotation.

The move and the rotation are simultaneous.

one the rotation is determined it moves with the bp.



First find angle A (the big \angle)

Then find angle B (the amount off the z plane)
surface

will then rotate rotline B° and attach to vertex (joint)

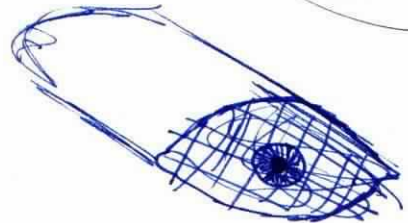
slope for B angle will be $\frac{\Delta z}{\Delta x}$

of A

Angles will be $180 - \frac{\text{slope}}{\tan}$

Dumb
wrong

$$\theta = \frac{\text{slope}}{\tan}$$

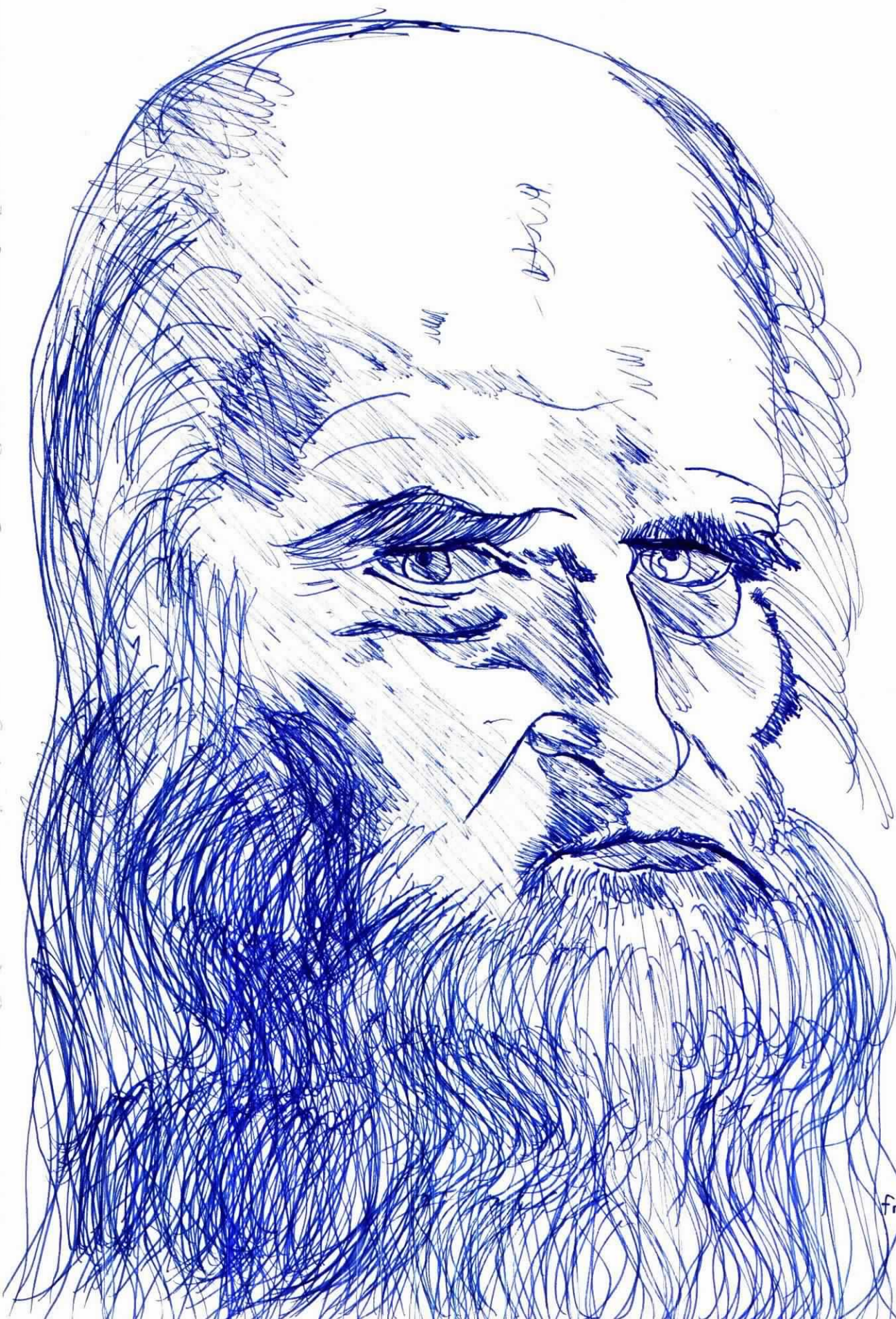




From Rubens
self portrait

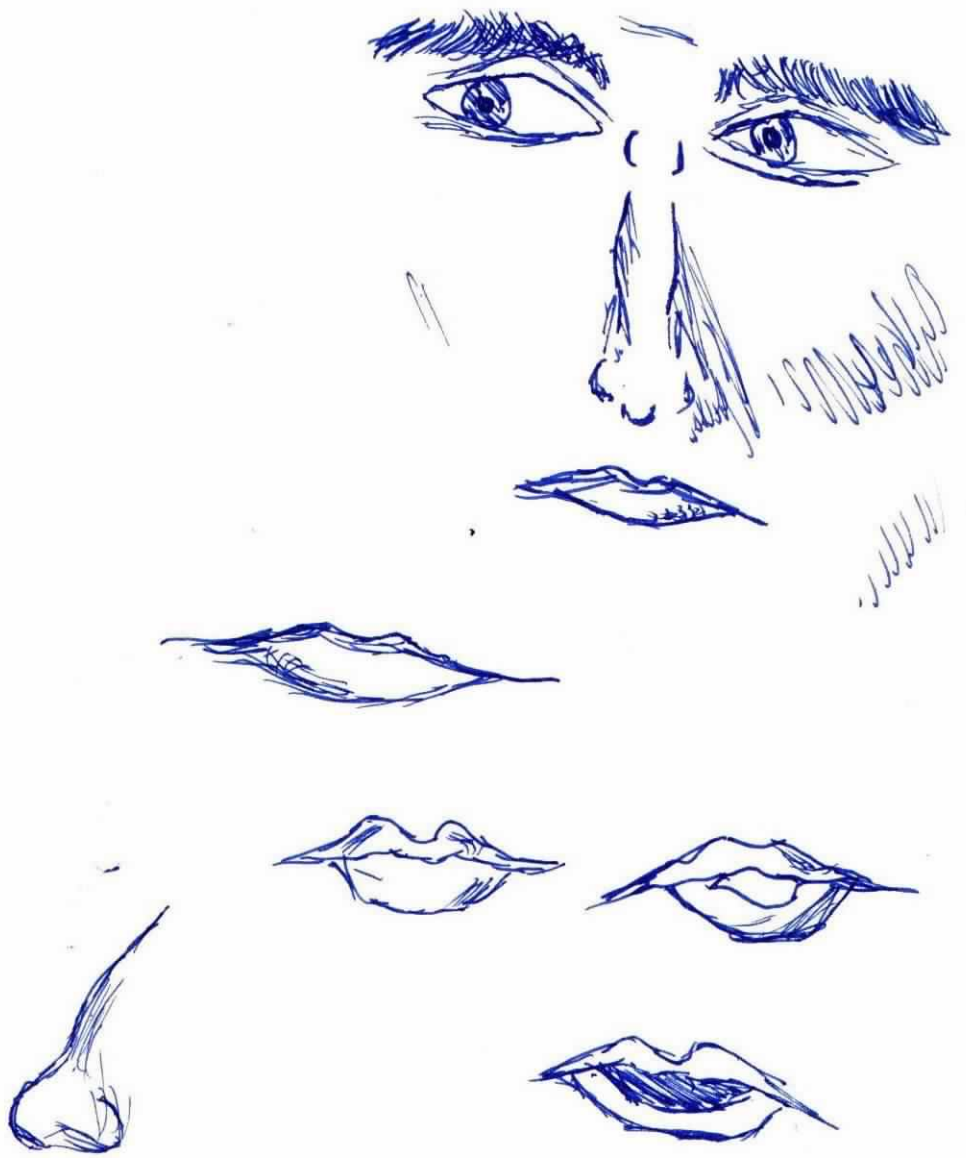
1624-5

9/22

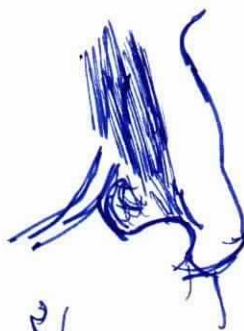
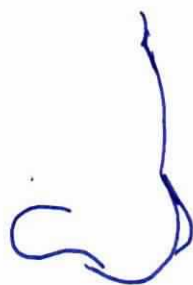
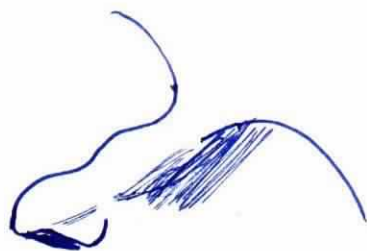
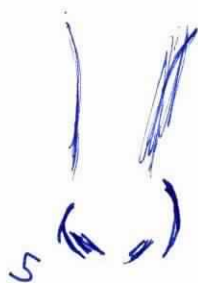


from Leonardo
Da Vinci
Self Portrait

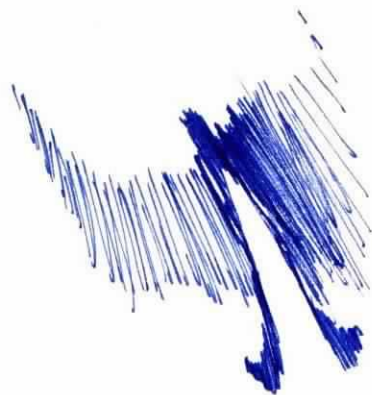




13.3
3 $\sqrt{40}$
3
10
9
10



from ~~Leonardo~~





from Leonardo
Head of a lady

51

9/24

From Rubens
1627 Por. of
a woman
#44



9/25 Need a rotational based interpolation routine for the ~~body~~ body

Given a matrix of $x \times 4$
step down the mat and examined each line
of both matrices

~~Mat~~

no. ~~Mat~~ Interp 'Mat1/Mat2'

GetLines: $L1 \leftarrow \text{Mat1}[\text{count count}+1]$
 $L2 \leftarrow \text{Mat2}[\text{count count}+1]$

$\rightarrow (L1=L2) / \text{GetLines}$

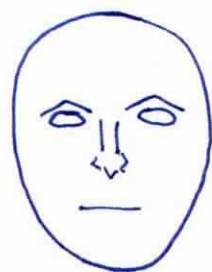
$L1 \leftarrow$

To be continued

Draw a generalized face

Determined ranges of
changes which could be

made ie fat face big nose....



Then randomly pick out parts to be changed
You would get computer generated faces.
should be possible to get real looking faces

Given a 3D body made of rectangular boxes and triangles
should be able to build it.

streted plex to look like motion

or

plex made like strobe effect,

or

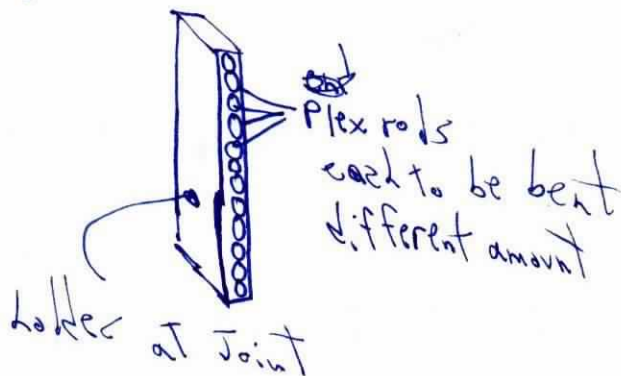
Make body out of plex rods

may be problem with connecting

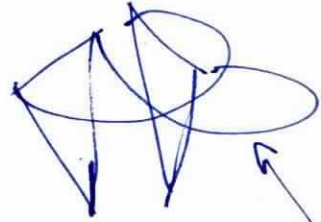
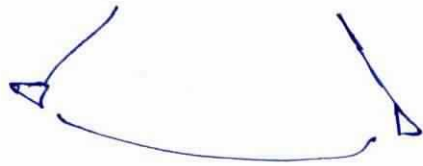
maybe use plex cylinders with hollow middle
and bend these connectors

for mapping routine need to know the angles from
the interpolation routine in order to move and
rotate the boxes.

to make a body with strobe like effect
use a piece like:

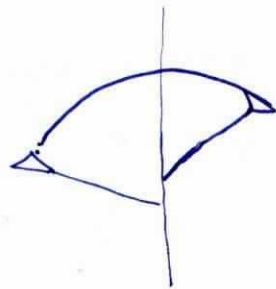
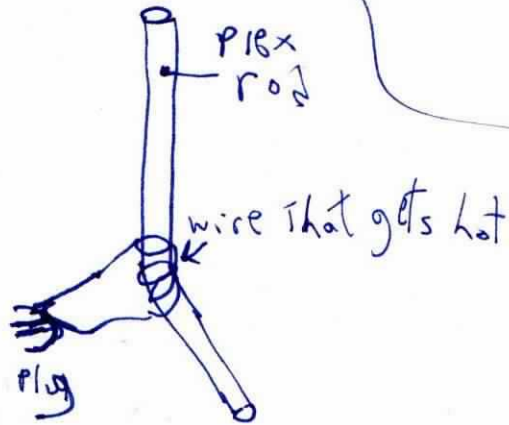


Another way to show motion would be to bend ~~curves~~ paths out of plex rods attached to a leading and trailing figure



plex paths

To bend rod →



For Video Film

List of objects actions & reactions

choose from these lists at random

Cause and effect

very close up shots of eye & mouth
and the nose

no sound except when
nose shows -

better yet

3 monitors
1 eye

EYE

MOUTH

NOSE

AFTER a while

eye changes to Mouth

MOUTH to NOSE

NOSE to eye

then ~~another~~ switch

Sequence is

EYE → MOUTH → NOSE



9/26 Random Position Generator

constraints matrix
of sphere 27 points

Body
Rbod ← ~~Body~~

~~count~~ ← 1
Lim ← ? 27

// # of times to pick a bp //

Getbp: pointer ← -1 + 2 × ? 18

Body ← Body [(pointer, pointer + 1);]

Dis Body

Di ← Di × 2

sphere 27 ←

Sphere 27 ←

RNO ← 1 + ? 22

Body [?;] ← sphere 27 [constraints [count; RNO];]

Rbod ← Rbod cat Body

→ (Lim ≥ count ← count + 1) / Getbp

pointer must be odd up to 35

Body [~~count~~ pointer, ~~count~~ pointer + 1]

pick num from 1 → 18

$$(((? 18) \times 2) - 1)$$

$$pointer \leftarrow -1 + 2 \times ? 18$$

~~Rbod [(pointer, pointer + 1);]~~

Rbod ← Body

// Body should be a reset body \subseteq Body //

Rbod [(pointer, pointer + 1);] ← sphere 27 [constraints [index; RNO];]

→ (Lim ≥ count ← count + 1) / Getbp

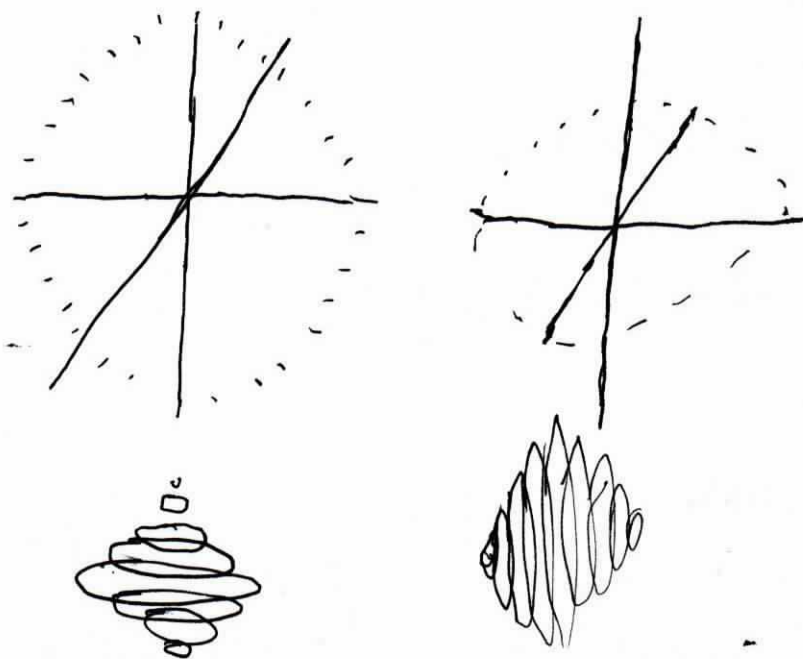
It Don't work make

Maybe use all the endrot
route
ringfind
locator } skit to determine angle after
the fact

probably much slower

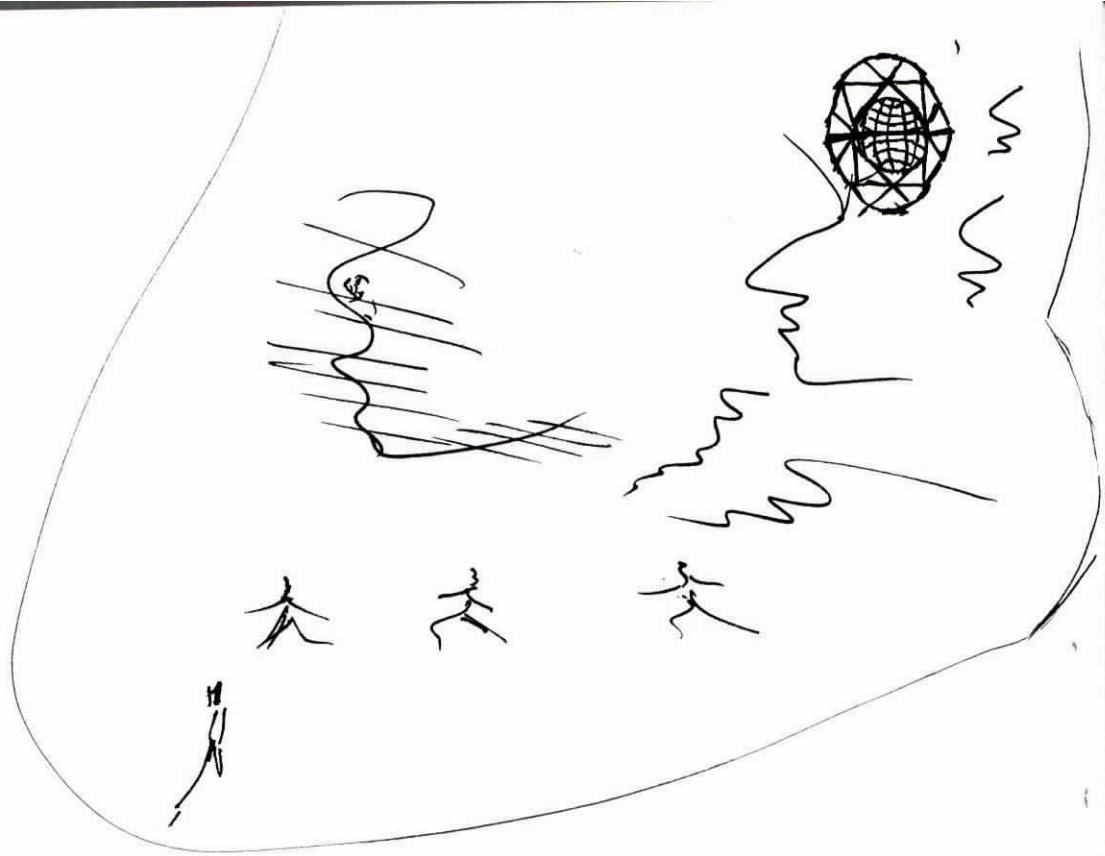
EYE on video with computer images
in it

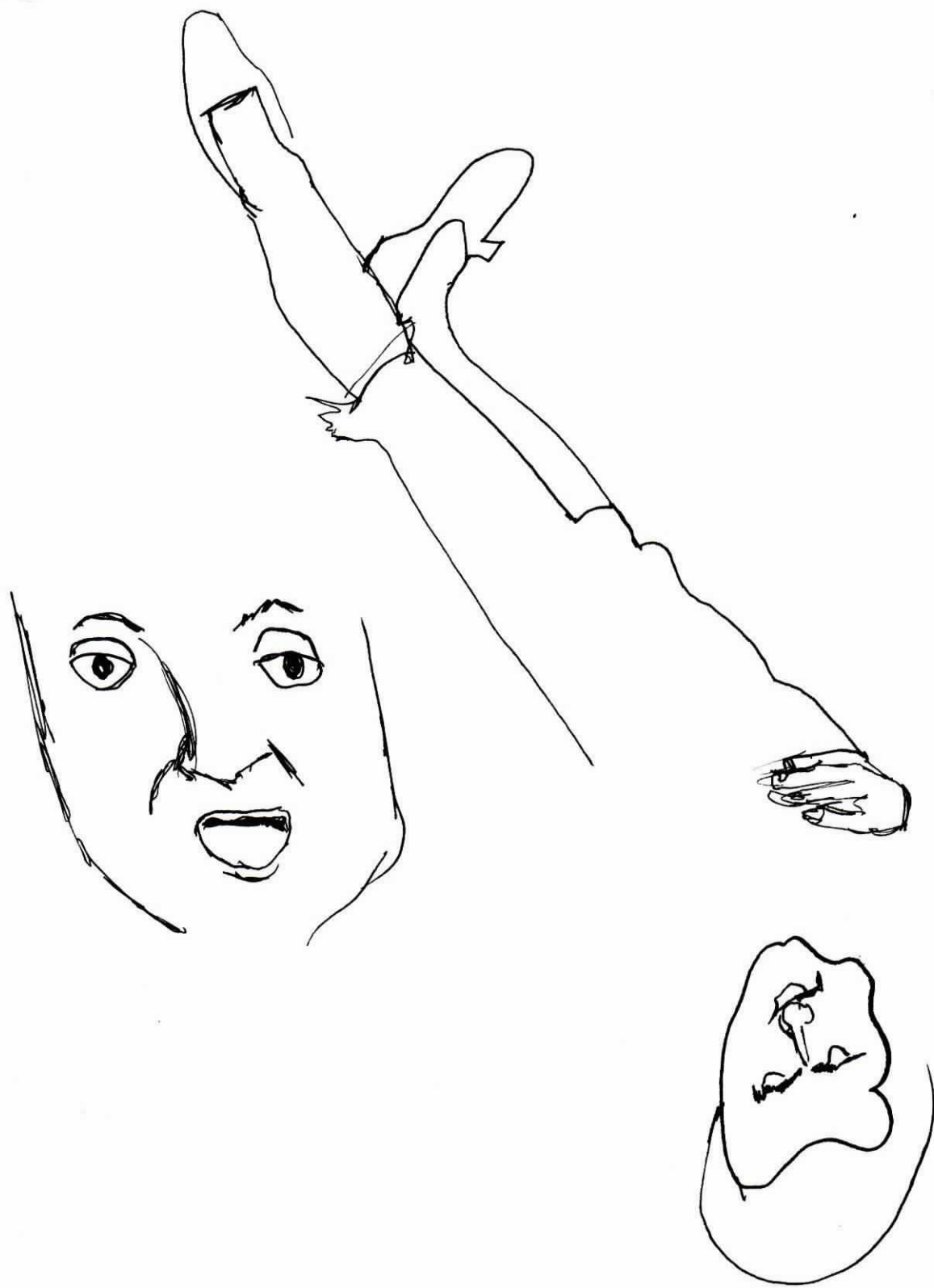
Diagram for refined movement

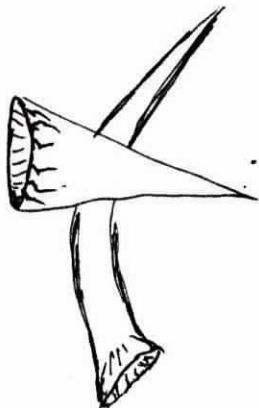
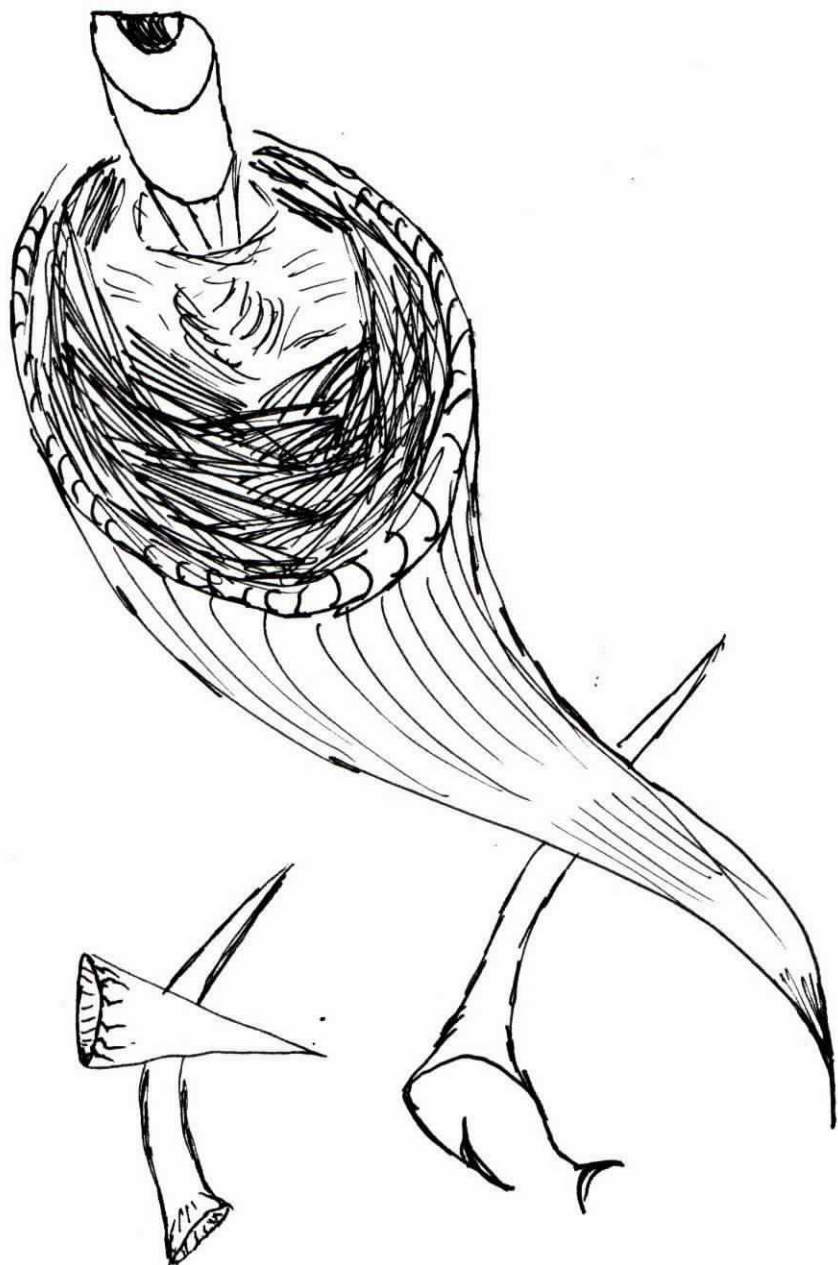


Maybe better to build a sphere physically with ref #'s

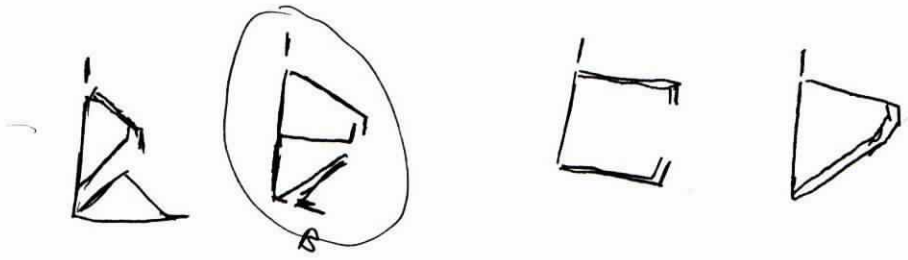
Move cursor to location on diagram
and point on sphere blinks a few times



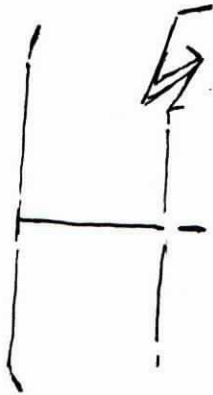




Q



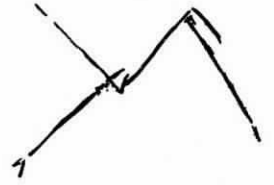
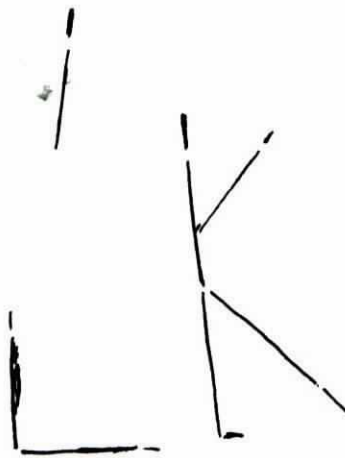
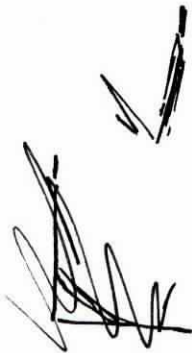
G

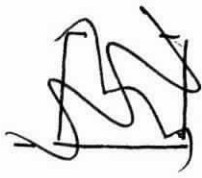


J H

K

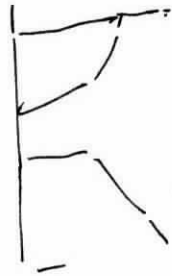
M



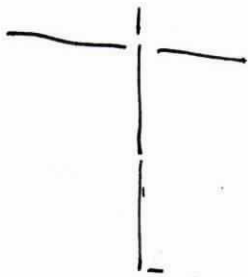


R

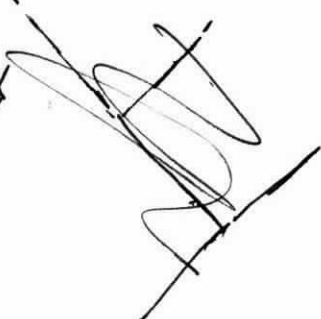
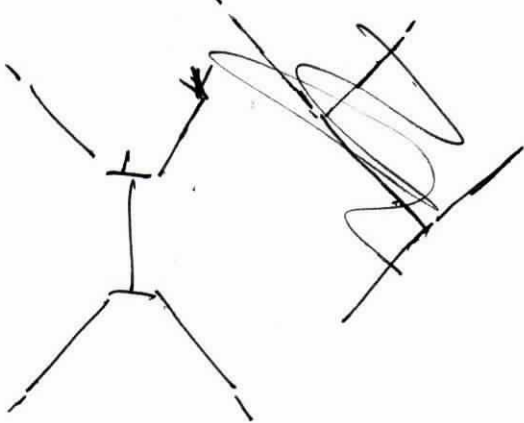
9



5



Z

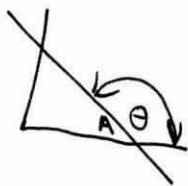


Z

9/27

Rotation Based Interpolation

$$\text{slope} = \tan \theta$$



Angle want is θ

~~Angle~~

~~Angle~~

$\theta \leftarrow 180 - \text{Arctan of slope}$
check: if θ has to be moved
first need to move (if at all) by's

scan down both matrices
checking where a move must occur
and create a vector of i's and o's
to locate where the move should occur

next move where necessary

need a formula for line perpendicular to plane and
intersecting a specific pt.

Mat1 * Mat2

(count-1) ÷ 2

count ← 1
movevec ← 360

// scanning down mats
~~count~~
flagging where a move
is necessary //

R ← Mat1[count;] = Mat2[count;]

~~movevec[(count-1) ÷ 2]~~

movevec[count] ← R

~~checker~~ movevec[count] ← Mat1[count;] = Mat2[count;]

→ (36 ≥ count ← count + 2) / checker

~~Movevec~~ ← -1 + 2 × 18 ↑ Movevec

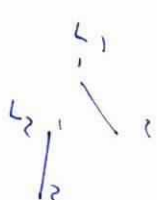
// Move the shit //

~~remainder of~~
~~236 ÷ 2 ↑ 36~~
-1 + 2 × 18 ↑ Movevec

count2 ← ~~1~~ Angvpc ← 1800

Guts: → (1 = Movevec[count2]) / Mover

→ (18 ≥ count2 ← count2 + 1) / Guts
"End of Scan"



Mover: Index ← (count × 2) + 1 - 1 + count × 2

L1 ← Mat1[Index Index+1;]

L2 ← Mat2[Index Index+1;]

L1 ← (L1[i;] · L2[i;]) Move L1

~~$R1 \leftarrow Mat1[Index_i]$~~
 ~~$R2 \leftarrow Mat2[Index_i]$~~

~~$L1 \leftarrow R1 - R2$~~ Move ~~$L1$~~
~~Print~~

~~$L1 \leftarrow Mat1[Index_i] - Mat2[Index_i]$~~ ~~Angle = 180~~

$L1 \leftarrow (L1[1:] - L2[1:])$ move $L1$

$\rightarrow (180 \geq \text{count} \text{ count} + 1) / \text{bits}$

10/11

Find angle between $L1$ & $L2$ and put into vector

~~Angle [count] < 7~~

Angle [count] \leftarrow $L1$ Angle $L2$

$q = (+ / \text{nat}_i = \text{Mat}_i)$



19
from Rubens
Hippopotamus Hunt
1614-18

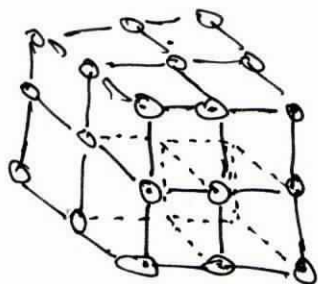
9/28

Move by x°

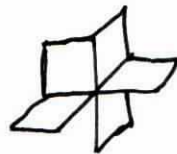
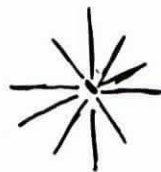
1-4s Toward a certain direction

This means a rotation towards a point
around a line

↑
hopefully 2 of the sphere 2 pts

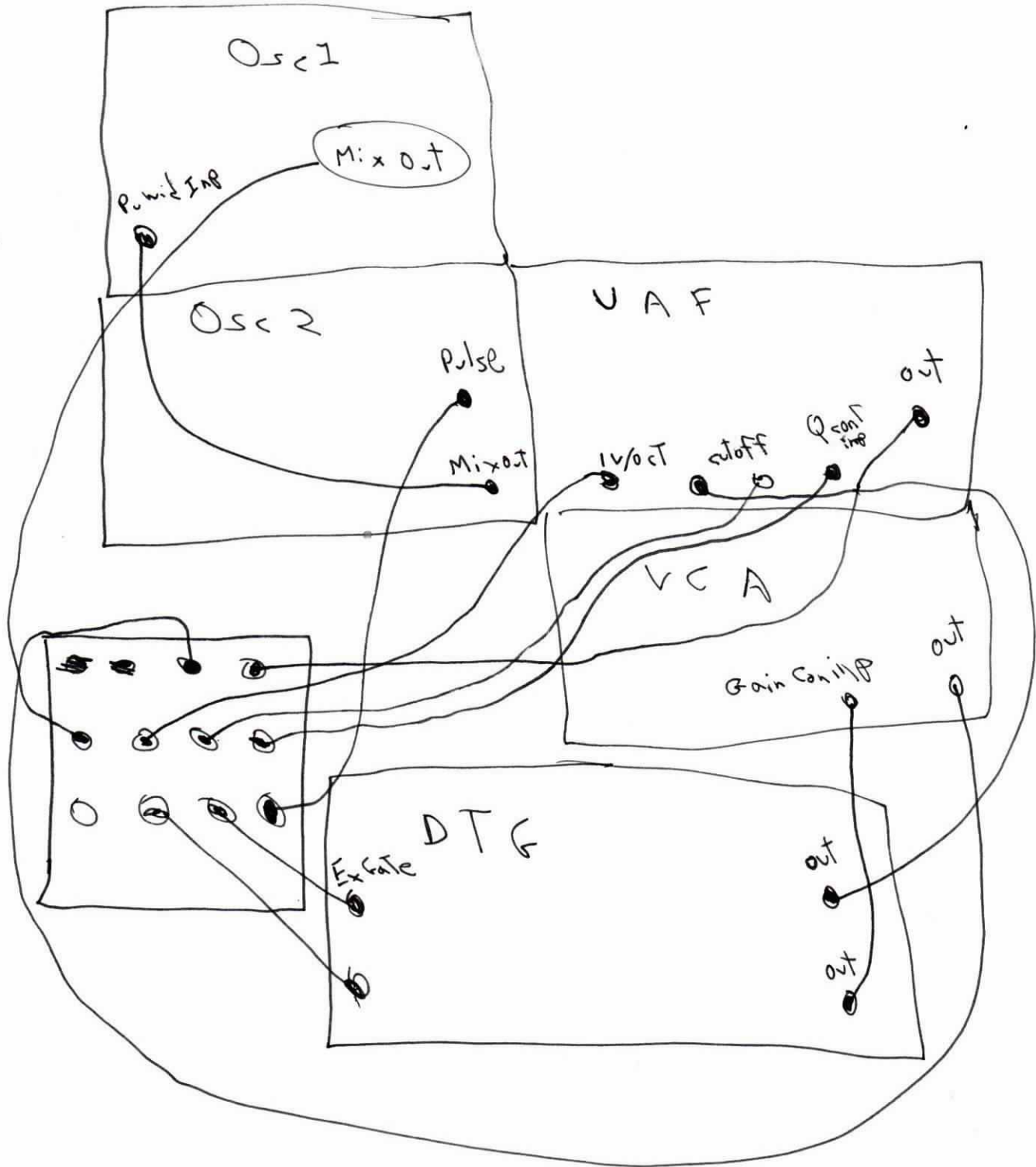


if I make



9/29

Synthesizer patches

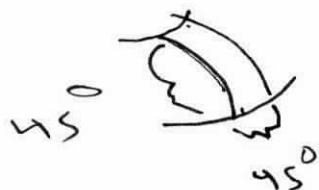


9/30

Given a cube top or on arbitrary side create



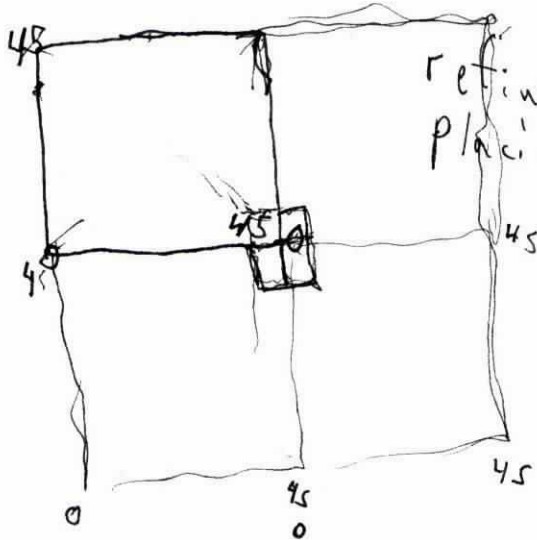
a coordinate system within that area



Do it visually
Draw a square and
with cursor pick out a point



~~9/31~~
10/1



refine the position by
placing cursor in square

4 squares,
and currently at center
This way you pick which
square dealing with and
pos. in it at same time.

For the plotting of AutoCAD's

□ RL ← 1

can do refined pos by saying Higher lower
LEFT right
Forwards backwards

To controll refining inc
↓
Δ Inc 5°

After you say it
it draws it immediately no
screen clearing

Command: Higher

need to remember which bp was used

In Bpinput
put for the last line
~~remember~~
remember ← name

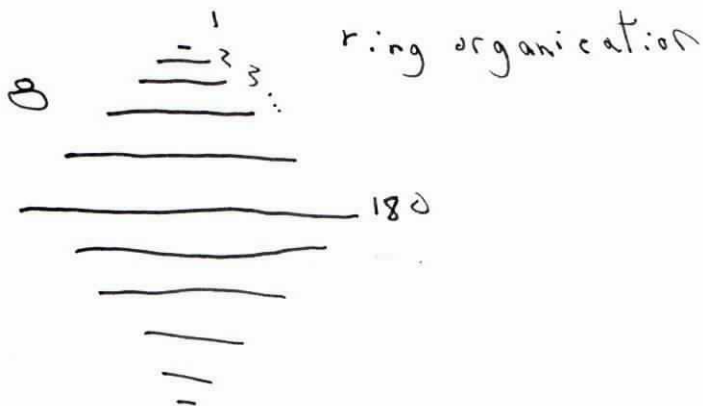
In Cominput say
→ (3 ↑ Inc = 'lef') / refine
→ (3 ↑ Inc = 'rig') / refine
→ (3 ↑ Inc = 'Hig') /
'low'
'for'
'back'

pas
vial
Help
Name
Eye

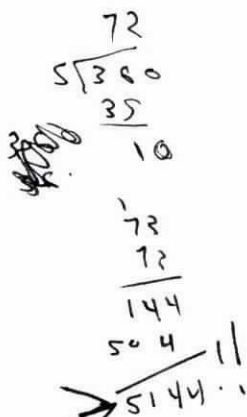
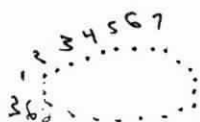
~~need to cut to find~~

need a way of moving \uparrow in desired direction
along sphere

refining



create the sphere as



so higher means

same pt on lower # ring

if $\Delta \text{inc} = 5$

then go to ring 5 #'s lower

lower is reverse

Too many pts for a s^2 sphere

must system size as you go along

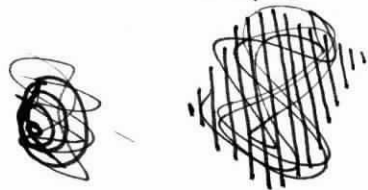
so you do have to store such a huge sphere

Organize the rings with surfaces
parallel to screen (easier to create)

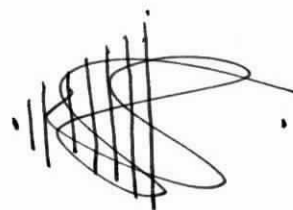
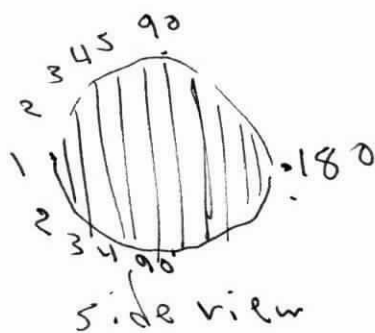
So higher means stay on same ring
and depending on side go up or down #'s



side view



front view of different rings
with surfaces



10/2

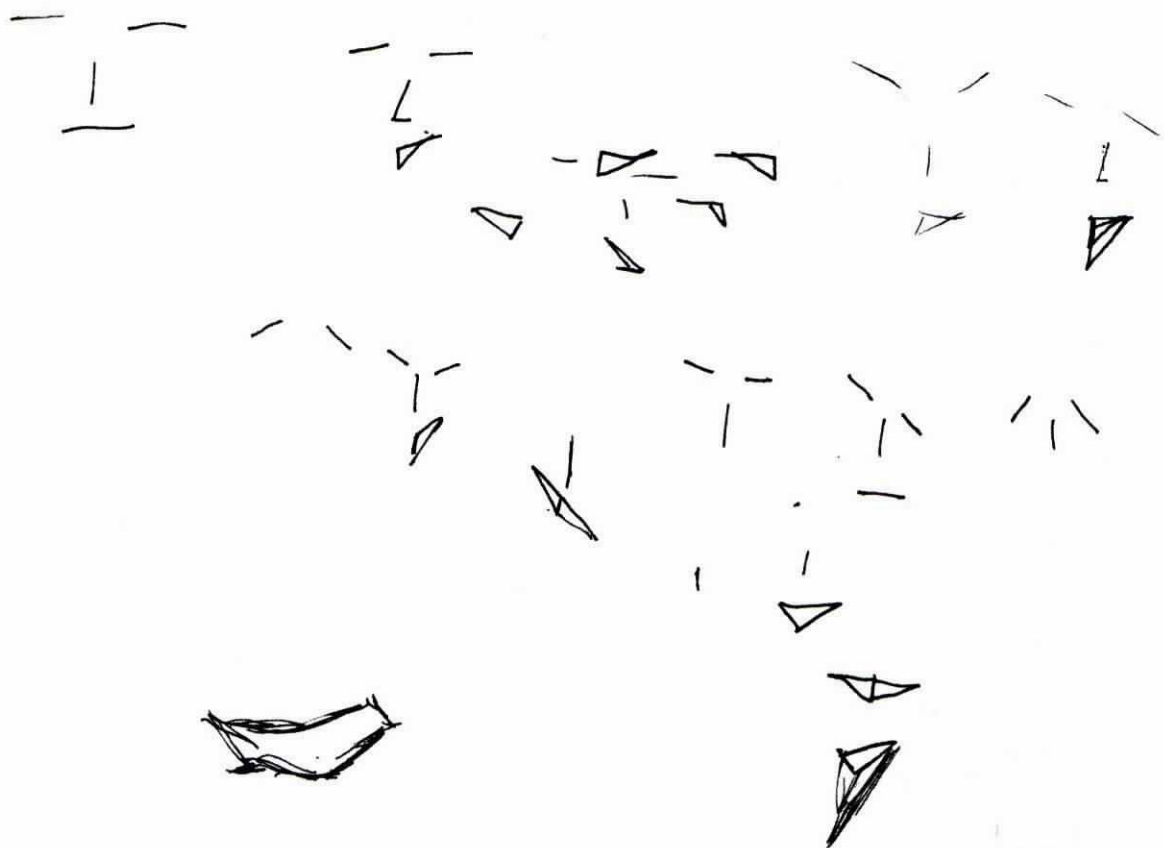


Suck!



From Leo da Vinci

#74



10/3 Refined Positioning

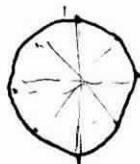


Given present position of bp

~~rate~~

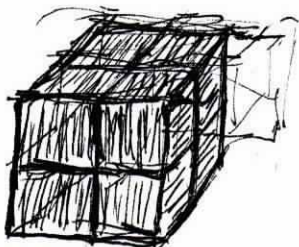
for higher lower use

17 + 20



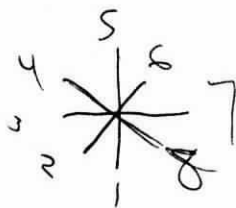
left, right

2 + 6

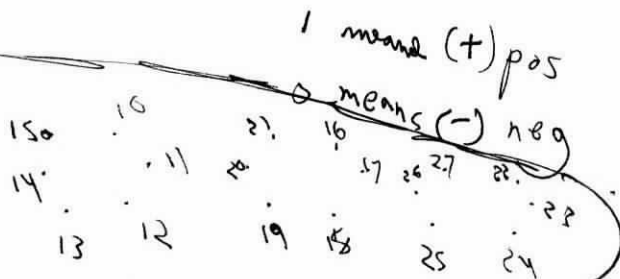
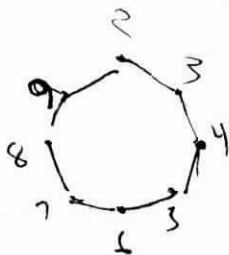


for, back

2 + 6



h low left rig for back



If at one pos of sphere27

▽ Refiner
2-1 ← 3-1-8
→ ((Inp = 'High'), (Inp = 'Low'), (Inp = 'Left'), (Inp = 'Right')) (E...
)) / L1, L2, L3, L4, L5, L6



↳ Draw Bsp rot's

First determine preset location

Determine whether rotation is neg or pos
~~and then rotate~~

Use STACKER

Using stacker rotate pieces appropriate amounts while moving ^{scaling} sphere27 to bp[i]

In Pos:put

MoveTo ← □

yes
Count maybe index to use

* Am at sphere27 [count;]

need index for the pos neg matrix

AT L1 L2 L3

assign the index no for posneg

→ (posneg[count; index] = 1) / pos

→ Neg

Pos: Ang ← ~~Increment~~

→ Mem

Neg: Ang ← -Increment

→ Mem

L1: ~~Mem~~

Mem ← 0LC
Index ← 1

L2: Mem ← 0LC
Index ← 2

DoIT: Draw TStack

Pop Stack by running stacker

→ (~~stack~~
if P TStack ≥ 0) / DoIT

L1: Mem ← 0LC

Mem ← 0LC + 2

→ Pos Neg filter

Index

0C

Pop pos

-59-

!

	High	Low	Left	Right	For	Back
1						
2	0	1			1	0
3	0	1				
4	0	1			0 *	1 *
5	0	1				
6	0	1				
7	1	0				
8	1	0			1 *	0 *
9	1	0				
10						
11						
12						
13						
14						
15						
16					1	0
17						
18					0	1
19					0	1
20					1	0
21					1	0
22						
23						
24						
25						
26						
27						

↑
 at presently at



L 1: mem ← OLC

→ (Pos neg [~~PTO~~ index] = 1) / pos

→ neg

Does
stacker
Pop the stack?
No - Activator Does

? → Pop: ~~Stacker~~
Activator

Draw ~~is~~ rot ~~Ang~~ #17020
Body

→ (↑ P Stack ≠ 0) / Pop

→ 0

Pos: Ang ← Increment

→ Mem + 3

Neg: Ang ← -Increment

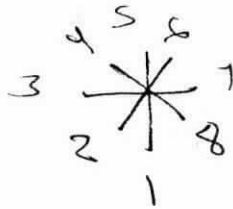
→ Mem + 3

LT |

LT Rot 20, ΔIP

/

+ 20° ΔHS ... (.)



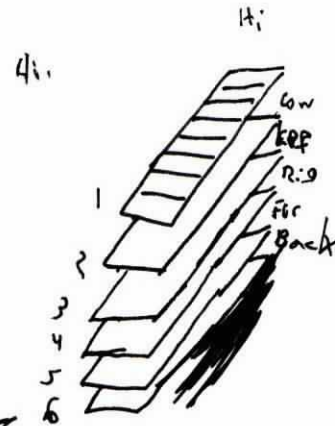
3D matrix

$$8 \times 3 \times 6$$

Ringfind PTO

ring	High	Low	Left
1	8 4	8 4	
2	14 11	=	
3	17 20	=	
4	23 26	=	
5	8 4	=	
6	14 11	=	
7	17 20	=	
8	23 26	=	

but
opposit
sign Angl



RefRotLinks

no good need dynamic
sphererz shift!

PTO rot code matrix
back in 9/1

Saying High Low Left...

~~means~~ means movement towards a pt

one of 6 pts For Back

	High	Low	Left	Right	For	Back
1						
2	26					
3	20 17					
4	20 17					
5	20 17					
6	8 4					
7	20 17					
8	20 17					
9	20 17					
10	23 26					
11	23 26					
12	23 26					
13	23 26					
14	25 28					
15	23 26					
16	8 4					
17	8 4					
18	8 4					
19	8 4					
20	8 4					
21	8 4					
22	14 11					
23	14 11					
24	14 11					
25	14 11					
26	14 11					
27	14 11					

where at

where going

only 6 possibilities

find angle

and then the rotation

Need program which:

Given pts ABC find θ and line \perp to plane and intersecting B



10/4 Synthesizer

#2 Tape 0-170

170-315

315 -

10/6 New Tape - 140 → 250 space
250 → 350
350 → 590?

Osc 1 Mix → VAC cutoff Freq cont.

Osc 2 Pulse → Multig



VAC - Band P → VCA - Sig Inp

Distance d_0 between the point $(x_0, y_0, z_0) = (r_0)$ and the straight

equation of the normal through the point (x_0, y_0, z_0)

to the plane $Ax + By + Cz + D = 0$ is

$$\frac{x - x_0}{A} = \frac{y - y_0}{B} = \frac{z - z_0}{C}$$

A B & C are
direction numbers
~~see~~ (over)

equations of Plane

General form

$$Ax + By + Cz + D = 0$$

$D = 0$ is when plane through
the origin

Three point form

$$P_1 \equiv (x_1, y_1, z_1) \equiv (r_1) \quad P_2 \equiv (x_2, y_2, z_2) \equiv (r_2) \quad P_3 \equiv (x_3, y_3, z_3) \equiv (r_3)$$

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = 0 \quad \text{or} \quad [(r - r_1)(r - r_2)(r - r_3)] = 0$$

Intercept form intercepting x axis at $x = a$

y axis at $y = b$

z " " $z = c$

$$\frac{x}{a} + \frac{y}{b} + \frac{z}{c} - 1 = 0$$

Any set of 3 numbers a_x, a_y, a_z proportional (with the same proportionality constant) to the direction cosines $\cos \alpha_x, \cos \alpha_y, \cos \alpha_z$ are called direction numbers of the line in question and

$$\cos \alpha_x = \frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}}$$

$$\cos \alpha_y = \frac{a_y}{\sqrt{a_x^2 + a_y^2 + a_z^2}}$$

$$\cos \alpha_z = \frac{a_z}{\sqrt{a_x^2 + a_y^2 + a_z^2}}$$

When the eqn of a plane is given in the general form (1), the quantities $a, b, c, p, \cos \alpha_x, \cos \alpha_y, \cos \alpha_z$ defined above are related in following manner

$$a = -\frac{D}{A} \quad b = -\frac{D}{B} \quad c = -\frac{D}{C}$$

$$\cos \alpha_x = \frac{A}{(A^2 + B^2 + C^2)^{\frac{1}{2}}} \quad \cos \alpha_y = \frac{B}{\dots}$$

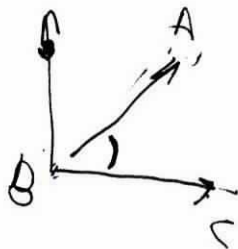
$$\cos \alpha_z = \frac{C}{\dots} \quad p = \frac{D}{(A^2 + B^2 + C^2)^{\frac{1}{2}}}$$

Let $p > 0$ be the length of the directed perpendicular between origin and a plane

From Mathematical Handbook for Scientists & Engineers... Korn
QA37.k74

Terry Butler 3rd fl.

10/10



$$\vec{BA} = A - B$$

$$\vec{BC} = C - B$$

$$\begin{vmatrix} i & j & k \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix} \begin{matrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{matrix}$$

~~$(y_1 z_2 - y_2 z_1)$~~

$\vec{R} = (y_1 z_2 - y_2 z_1, z_1 x_2 - z_2 x_1, x_1 y_2 - x_2 y_1)$

$$((y_1 z_2) - (y_2 z_1)) \vec{k}$$

$B + \rightarrow$

Dot prod $\vec{BA} \times \vec{BC}$

+ . x

\vec{B}

+ . x

$\vec{B} + \vec{R}$

QA303 T42

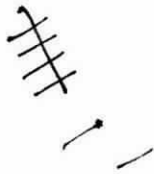
$$\arccos \left(\frac{\vec{BA} \cdot \vec{BC}}{|\vec{BA}| \times |\vec{BC}|} \right)$$

$$-20 (\vec{BA} \cdot \vec{BC}) \div ((|\vec{BA}| * 2) * .5) \times (|\vec{BA}| * 2) * .5$$

100, 100, 100

100, 100, 0

200, 100, 0



21 ~~+~~

Ref: nex

6 pt of movement towards

sphere27 [2] - Higher

[5] - Lower

[4] - Left

[8] - Right

[17] - Forward

[20] - Backward

B ← Body [1]

D ← Body [2]

~~A ← sphere27 [index]~~
~~sphere [27]~~

~~Ref: nex~~

Index ←

Intrefine

Pop: Activator

Draw Body ←

Φ BNames (Pointers [Child; Inc];) ' ← Body '

Refiner

→ L1 L2

L1 Mem ← OLC

Index ←

→ (Posneg [PTO; Index] = ~~1~~ / Pos

→ Neg

Stacker

Movsphere

Intrefine

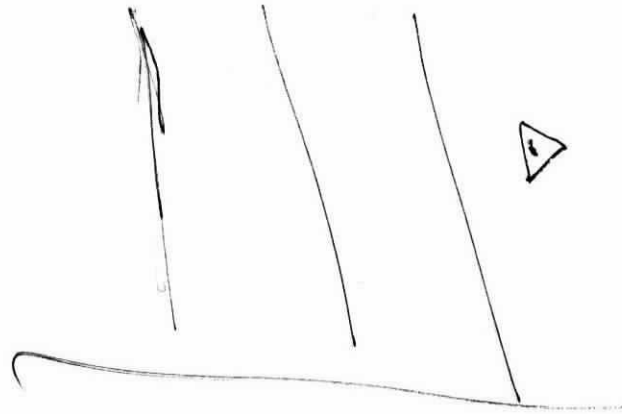
Refiner ←

→ Gening

L2

LL

LF
actr like for
index 2



[3] → 14
[14] 14 9
(15) 2
(15) 5
(59) → 59
(60) 5
9 18

$$\rightarrow (+/Inp = 'Hig') = 3$$

$$(3 = +/Inp = 'Hig'), (3)$$

10/11

Synthesizer Tap 450 - 483

483 - 712

Buzzes \rightarrow 712 - end

5:18

0 - 208

~~208~~ - 315
208

Refiner - Almost works
moves in correct plane so problem
is with sign of Ang

Posnes is 27 x 6

By is moving from one of 27 meter pts
toward one of 6 directions

The refiner is like a mine interpolator
based on rotation.

1st to a rotation
~~if~~

No Table Lookup
(posneg)

find dist of $A \rightarrow D$

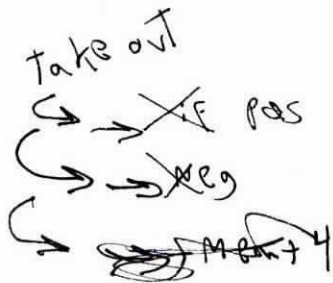
rotate

~~if dist~~ find dist $A \rightarrow D$ again

if $dist_2 < dist_1$ keep that sign

else change sign to neg and

do the rotations



A Bodyp[2]
D sphere[2][x]

in Intrefine

~~Dist~~ Dist $A \rightarrow D$
~~Bodyp~~ Bodyp rot Ang R Line
Dist ~~is~~ Testp[2; 2 3 4] $\Rightarrow D$

$\rightarrow (Dist_2 < Dist_1) / \text{cont}$
Angle - Ang

cont: original form of intrefine

want to avoid check many times for sign
redundantly as in a group. So set some flag
AngleFlag or signFlag

For The Big Interpolator

see 9/27

need to move all things around first
to determine angles

but for display the only move
to take place is if hips are not equal

Need Angfinder

$\angle 1 [i; j]$
 $\angle 2 [i; j]$
must be
equal

$$\Delta \text{ Ang} \leftarrow \angle 1 \text{ Angfind } \angle 2$$

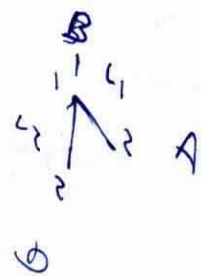
$$BA^{vec} \leftarrow \angle 1 [r; r; 3; 4] - \angle 1 [i; r; 3; 4]$$

$$BC^{vec} \leftarrow \angle 2 [r; r; 3; 4] - \angle 2 [i; r; 3; 4]$$

$$\text{Dis } BA \leftarrow \left((\angle 1 [r; r; 3; 4] + \angle 1 [i; r; 3; 4]) * 2 \right) * .5$$

$$\text{Dis } BC \leftarrow \left((\angle 2 [r; r; 3; 4] + \angle 2 [i; r; 3; 4]) * 2 \right) * .5$$

$$\text{Ang} \leftarrow 20 \left(BA^{vec} + .x BC^{vec} \right) \div (\text{Dis } BA \times \text{Dis } BC)$$



Now have a vector of angles
and need $\Delta y \geq \text{Dis}$ from $\angle \text{Hip} [i; r; 3; 4]$ of $\text{Pos} 1$
to $\angle \text{Hip} [i; r; 3; 4]$ of $\text{Pos} 2$

$$\text{Move int} \leftarrow \text{Pos} 1 [x; r; 3; 4] - \text{Pos} 2 [x; r; 3; 4]$$

syntax will be

~~Post/Pass~~
~~# of int~~ Fill

Post Fill Post

(13)

Intfac → □

// not in header because
whole program will
be used within main
program



dis is dis A+C

$$\tan \alpha = \frac{m_2 - m_1}{1 + m_1 m_2}$$

$$\cos \theta = \sqrt{1} + \sqrt{2} - \sqrt{2}$$

THE Bastard works!!!

Can ^{now} type Hig Low Left Rig For Bac
And it does it.

Angfinder also works!

ARS 323

~~Dance: Dohy~~

~~T. SPR~~

1234 Newvoice