JOSHUA PARKER

SOURCE-CODE

FOR SAMPLE WORKS

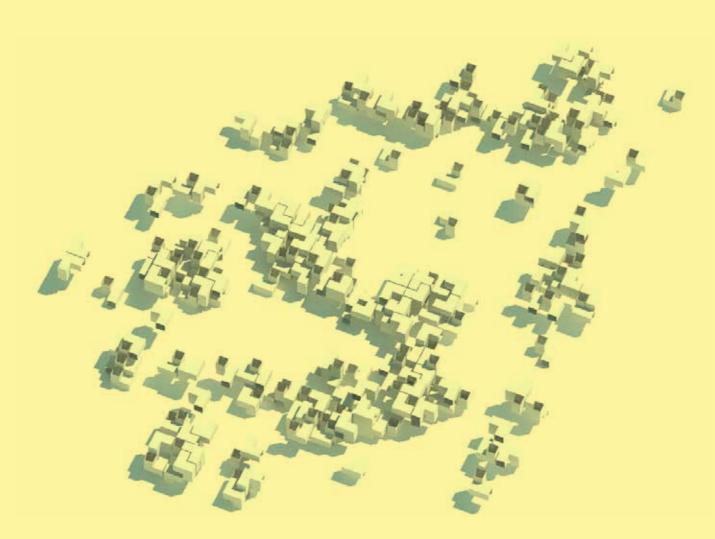
JOSHUA PARKER, M. Arch., Syracuse University School of Architecture (SUSOA), B.S. Electrical Engineering, University of Washington (UW), Certification, Institute of Advanced Architecture of Catalonia (IaaC)

CONTENTS

PROTOCELL, SOURCE CODE	5
UPLIFT HANGER, SOURCE CODE	15
ZHUHAI PAVILION, SOURCE CODE	<u>25</u>
BEING THERE, SOURCE CODE	37
META-SWARM, SOURCECODE	47

PROTOCELL, SOURCE CODE

SOURCE CODE FOR PROTOCELL GENERATIVE HOUSING SYSTEM. THE ALGORITHM, INITIALLY IMPLEMENTED IN RHINOSCRIPT, CAN BE DIVIDED INTO THREE ISOLATED STAGES. THE FIRST DISTRIBUTES CELLULAR MASS ACCORDING A CA LOGIC AND OPTIMIZER THAT ADJUSTS MASSING FOR CIRCULATION PATHS AND OTHER ENCODED CONSTRAINTS LIKE SITE, TOPOGRAPHIC FEATURES, ETC. THE FIRST STAGE REQUIRES AS INPUT AN INITIAL MASSING CONDITION AS CELLULAR AUTOMATA PROCEDES AS INCREMENTAL REFINEMENT.[1] FIRST STAGE IS ALSO RESPONSIBLE FOR CONSTRUCTING A CELL NETWORK, IE. EACH CELL MAINTAINS STATE AND KEEPS TRACK OF NEIGHBOORING CELLS. THIS NETWORK IS PASSED TO STAGE TWO ALONG WITH FURTHER BUILDING CONSTRAINTS LIKE PROGRAM, FLOORSPACE ALLOCATION, ETC. THE SECOND STAGE SUBDIVIDES SINGLE-CELL MASS INTO MULTI-CELL CLUSTERS OR LINKED CELL CHAINS. THIS IS ACCOMPOLISHED BY ESTABLISHING AN INITIAL CONDITION OF RANDOMLY PLACED CONDITIONAL UNITS AND DEFINING A FITNESS FUNCTION THAT EVALUATES THE PLACEMENT AND SITUATION OF THE UNIT AND SIMPLY DISCARDS LOSERS AND IGNORES WINNERS. THIS IS A WEAK GENETIC OPTIMIZATION ALGORITHM THAT IS MORE BRUTE FORCE THAN ANYTHING ELSE, BUT IT DOES FIND SOLUTIONS IN REASONABLE AMOUNT OF TIME. THE FINAL STAGE EMBEDS BUILDING INTELLEGENCE INTO CELL CLUSTERS, PLACES UTILITY CORE, AND ADAPTS UNIT TO BASIC BUILDING CONSTRAINS... ALL STAGES ARE INTENDED TO BE A GUIDED META-DESIGN PROCESSES, IN WHICH DESIGNERS PARTICIPATE IN REAL TIME BY MAKING SUBJECTIVE DECISIONS AND FEEDING THEM BACK INTO THE SYSTEM.



```
Public m arrType2VNs
'Smart cells v1
                                                                                    'neighbor pointers
                                                                                   Public m objNorth
                                                                                   Public m objSouth
                                                                                   Public m_objEast
Option Explicit
                                                                                   Public m_objWest
'Script written by Josh Parker
                                                                                   Public m_objTop
'Script copyrighted by OPEN Architecture
                                                                                   Public m_objBottom
'Script version Tuesday, February 15, 2011 3:25:36 AM
                                                                                    'is cell part of snake
Call Main()
                                                                                   Public m boolIsOpen
Sub Main()
       Dim arrPts, intSteps, intLevels, intSpacing, oMajRule
                                                                                   Private Sub Class Initialize
                                                                                           'Rhino.Print "Initialize cell"
       Dim oSubstr, oSnakeFarm
                                                                                           m boolIsOpen = True
       'get initial data
       arrPts=Rhino.GetObjects("select points to populate",1)
       If Not IsArray(arrPts) Then
                                                                                   Private Sub Class Terminate
               Rhino.Print "no points selected"
                                                                                           'Rhino.Print "Terminate cell"
              Exit Sub
       End If
       intSpacing=Rhino.GetInteger("enter cell spacing",5)
                                                                                   Public Sub Build(strPt, intType)
       intSteps=Rhino.GetInteger("enter number of steps",2)
                                                                                          m_intType = intType
       intLevels=Rhino.GetInteger("enter number of levels",2)
                                                                                           m strPt = strPt
       'create substrate
       Set oSubstr = New Substrate
                                                                                   End Sub
       'substrate "constructor"
       Call oSubstr.Build(intSpacing, intSteps, intLevels)
                                                                                   Public Sub Connect(arrCells, intRadius)
       'add Cells from points
       Call oSubstr.AddCellsByColor(arrPts)
                                                                                           'check arg is valid non-empty array
       'link each cell to its neighbors
                                                                                           If IsEmpty(arrCells) Or IsNull(arrCells) Then
       Call oSubstr.ConnectCells()
                                                                                                  Rhino.Print "Connect arg, arrCells is empty or
                                                                            null"
        'group cells by type
       Call oSubstr.GroupCells()
                                                                                                  Exit Sub
        'color cell pts by cell type
                                                                                           ElseIf Not IsArray(arrCells) Then
       Call oSubstr.ColorByType()
                                                                                                  Rhino.Print "Connect arg, arrCells is not an ar-
                                                                            ray, its a: " + TypeName(arrCells)
       'create snakes
                                                                                                  Exit Sub
                                                                                           End If
       Set oSnakeFarm = New SnakeFarm
       'locate snakes in substrate
       Call oSnakeFarm FindSnakes (oSubstr. 4)
                                                                                          Call FindNeighbors(arrCells, intRadius)
                                                                                           'Call PointAtNeighbors()
       'spine the snakes
       Call oSnakeFarm.SpineSnakes()
                                                                                           Call GroupNeighbors()
       'skin the snakes
                                                                                           Call GroupVonNeumanns()
       Call oSnakeFarm.SkinSnakes((CDbl(intSpacing)/2)*.9)
                                                                                   End Sub
       'create massing
       'Set oMass = New massing
                                                                                    'find neighbors and create ordered array of pointers-----
       'build mass from snake formation
                                                                                   Private Sub PointAtNeighbors()
        'oMass.build(oSnakes)
                                                                                           Dim c, p1, p2
                                                                                           'sort neighbors
                                                                                           For Each c In m_arrNeighbors
                                                                                                   'get coordinate of target and point
'Cell CLASS
                                                                                                  p1=Rhino.PointCoordinates(m strPt)
                                                                                                  p2=Rhino.PointCoordinates(c.m strPt)
                                                                                                  If p2(1) > p1(1) Then
                                                                                                          'y is >, its on north
Option Explicit
                                                                                                          Set m_objNorth = c
'Script written by Josh Parker
                                                                                                  ElseIf p2(1) < p1(1) Then
'Script copyrighted by OPEN Architecture
                                                                                                          'y is <, its on south
'Script version Tuesday, February 15, 2011 3:25:36 AM
                                                                                                          Set m_objSouth = c
                                                                                                  ElseIf p2(0) > p1(0) Then
Class Cell
                                                                                                          'x is >, its on east
                                                                                                         Set m_objEast = c
       Public m_intType
                                                                                                  ElseIf p2(0) < p1(0) Then
       Public m_strPt
                                                                                                          'x is <, its on west
                                                                                                          Set m_objWest = c
       'array of neighbors
                                                                                                  ElseIf p2(2) > p1(2) Then
       Public m arrNeighbors
                                                                                                          'z is >, its on top
                                                                                                  Set m_objTop = c
ElseIf p2(2) < p1(2) Then
       Public m arrVonNeumanns
        'moore subsets
                                                                                                          'z is <, its on bottom
       Public m arrTypeOMs
                                                                                                          Set m_objBottom = c
       Public m arrType1Ms
                                                                                                  Else
       Public m_arrType2Ms
                                                                                                         Rhino.Print "ignore, same cell"
                                                                                                  End If
        'vonneumann subsets
       Public m arrType0VNs
                                                                                           Next.
       Public m_arrType1VNs
```

 ϵ

End Sub

```
'Group neighbors by type-----
       'Get points within radius from cell pt-----
                                                                               Public Sub GroupVonNeumanns()
       Private Sub FindNeighbors(arrCells, intRadius)
                                                                                      On Error Resume Next.
                                                                                      'Rhino.Print "Grouping Neighbors"
              Dim c, d, i, j, closeCells(), adjacentCells(), p1, p2
                                                                                      Dim type0(), type1(), type2()
                                                                                      Dim i, a, b, c
                                                                                      a=0
              For Each c In arrCells
                                                                                      b=0
                     'calc distance to cell c
                                                                                      c=0
                     p1=Rhino.PointCoordinates(m_strPt)
                     p2=Rhino.PointCoordinates(c.m strPt)
                                                                                      If IsNull(m arrVonNeumanns) Then Exit Sub
                     d=Rhino.Distance(p1,p2)
                     'if its within radius, its adjacent
                                                                                      For i=0 To Ubound(m_arrVonNeumanns)
                     If d<=(intRadius*1.1) And d>0 Then
                            ReDim Preserve adjacentCells(i)
                                                                                              'group by type
                            Set adjacentCells(i) = c
                                                                                             Select Case m_arrNeighbors(i).m_intType
                                                                                                    Case 0
                            i=i+1
                     End If
                                                                                                           ReDim Preserve type0(a)
                     'if its within 1.5radius, its close
                                                                                                           Set type0(a) = m_
                     If d<=(intRadius*1.5) And d>0 Then
                                                                        arrVonNeumanns(i)
                            ReDim Preserve closeCells(i)
                                                                                                           a=a+1
                            Set closeCells(j) = c
                                                                                                    Case 1
                                                                                                            ReDim Preserve type1(b)
                            j=j+1
                     End If
                                                                                                            Set type1(b) = m_
              Next
                                                                        arrVonNeumanns(i)
                                                                                                           b=b+1
              'set if any
              If i>0 Then
                                                                                                            ReDim Preserve type2(c)
                     m_arrVonNeumanns = adjacentCells
                                                                                                            Set type2(c) = m_{\underline{}}
                                                                        arrVonNeumanns(i)
                     Rhino.Print "no von neumanns"
                     Rhino.ObjectColor m strPt, RGB(0, 0, 255)
                                                                                                           Rhino.Print "GroupVonNeumanss()
                                                                        found bad cell"
               'set if any
                                                                                             End Select
              If j>0 Then
                                                                                      Next
                     m arrNeighbors = closeCells
                                                                                      'copy if smth to copy, else leave empty
              Else
                     Rhino.Print "no moores"
                                                                                      If a > 0 Then m arrType0VNs = type0
                                                                                      If b > 0 Then m arrType1VNs = type1
                     Rhino.ObjectColor m strPt, RGB(0, 255, 0)
                                                                                      If c > 0 Then m_arrType2VNs = type2
              End If
                                                                               End Sub
       End Sub
                                                                                'color by type-----
                                                                               Public Sub UpdateColor()
       'Group neighbors by type-----
       Public Sub GroupNeighbors()
                                                                                      Dim color
              'Rhino.Print "Grouping Neighbors"
                                                                                      'set color var by cell type
              Dim type0(), type1(), type2()
                                                                                      Select Case m_intType
              Dim i, a, b, c
                                                                                             Case 0
              a=0
                                                                                                    color = RGB(255, 0, 0)
                                                                                             Case 1
              c=0
                                                                                                    color = RGB(0, 255, 0)
                                                                                             Case 2
              For i=0 To Ubound(m arrNeighbors)
                                                                                                    color = RGB(0, 0, 255)
                                                                                             Case Else
                                                                                                    color = RGB(255, 255, 255)
                      'group by type
                     Select Case m_arrNeighbors(i).m_intType
                                                                                      End Select
                            Case 0
                                   ReDim Preserve type0(a)
                                                                                      'set object color
                                   Set type0(a) = m_arrNeighbors(i)
                                                                                      Rhino.ObjectColor m_strPt, color
                                   a=a+1
                            Case 1
                                                                                'apply ruleset-----
                                   ReDim Preserve type1(b)
                                   Set type1(b) = m_arrNeighbors(i)
                                                                               Public Sub Apply(oRuleSet)
                                   b=b+1
                            Case 2
                                                                                      Dim newType
                                   ReDim Preserve type2(c)
                                                                                      'calculate new type
                                   Set type2(c) = m_arrNeighbors(i)
                                                                                      'newType = oRuleSet.Apply(m_objNorth, m_objSouth, m_ob-
                                                                        jEast, m_objWest, m_objTop, m_objBottom)
                                   Rhino.Print "GroupNeighbors()
                                                                                      newType = oRuleSet.Apply2(m arrNeighbors)
found bad cell"
                                                                                       'set new type
                                                                                      m intType = newType
                     End Select
              Next
                                                                               End Sub
              'copy if smth to copy, else leave empty
              If a > 0 Then m_arrType0Ms = type0
              If b > 0 Then m_arrType1Ms = type1
                                                                               Public Sub MajorityRuleStep()
              If c > 0 Then m_arrType2Ms = type2
                                                                                      'count number of a,b,c types within neighbors
```

```
'set type to that type
                                                                                                       ReDim Preserve maj(j)
                                                                                                       maj(j) = i
       End Sub
                                                                                                       j=j+1
                                                                                                End If
End Class
                                                                                        Next
                                                                                        'if tie
'MajorityRule CLASS
                                                                                        If(Ubound(maj)>0) Then
                                                                                                'get random max
                                                                                                iRand=RndInt(0,Ubound(maj))
                                                                                                Apply2 = maj(iRand)
Option Explicit
'Script written by <insert name>
                                                                                                Apply2 = maj(0)
'Script copyrighted by <insert company name>
'Script version Thursday, February 17, 2011 11:50:25 PM
                                                                                 End Function
Class MajorityRule
                                                                          End Class
                                                                          Option Explicit
       Public Function Apply(n,s,e,w,t,b)
                                                                          'Script written by Josh Parker
                                                                          'Script copyrighted by OPEN Architecture
                                                                          'Script version Tuesday, February 15, 2011 3:25:36 AM
              Dim cells, counts, c, i
                                                                          ·----
               cells = Array(n, s, e, w, t, b)
                                                                          'SNAKEFARM CLASS
              counts = Array(0,0,0)
               'count number of each type
              For Each c In cells
                      Rhino.Print TypeName(c)
                                                                          Class SnakeFarm
                      If Not IsEmpty(c) Then
                             Select Case c.m_intType
                                                                                  Public m_arrSnakes
                                                                                 Public m intSnakeCnt
                                    Case 0
                                            counts(0) = counts(0) + 1
                                                                                  Public Sub FindSnakes(oSubstr, intSize)
                                    Case 1
                                            counts(1) = counts(1) + 1
                                                                                        Dim snakeCells, newSnakes()
                                    Case 2
                                            counts(2) = counts(2) + 1
                                                                                        Dim i, failed, invalid
                                    Case Else
                                     'do nothing
                                                                                         'check args
                                                                                        If IsNull(oSubstr) Then
                             End Select
                                                                                                Rhino.Print "FindSnakes arg, oSubstr is null"
                              'is null b/c no cell
                                                                                                Exit Sub
                                                                                        ElseIf Not TypeName(oSubstr) = "Substrate" Then
                             Rhino.Print "skip " + TypeName(c)
                                                                                                Rhino.Print "FindSnakes arg, oSubstr not a Sub-
                     End If
              Next
                                                                          strate, its a: " + TypeName(oSubstr)
                                                                                                Exit Sub
               'return most numerous type
                                                                                        End If
              For i=0 To Ubound(counts)
                      If counts(i) = Max(counts) Then
                             Apply = i
                                                                                        failed = 0
                      End If
                                                                                        invalid = 0
       End Function
                                                                                        Do While failed<500
                                                                                                'get next set of cells
       'This one is better
                                                                                                snakeCells = NextSnake(oSubstr, intSize)
       Public Function Apply2(arrCells)
                                                                                                'if not invalid or null, make a snake and add to
                                                                          farm
              Dim cells, counts, c, i, j, maj(), iRand
                                                                                                If IsNull(snakeCells) Then
                                                                                                        'found a short snake, increment failed
               'cells = Array(n,s,e,w,t,b)
                                                                          count
              counts = Array(0,0,0)
                                                                                                       failed=failed+1
                                                                                                       Rhino.Print "found short snake, short
               'count number of each type
                                                                          count: " + CStr(failed)
              For Each c In arrCells
                                                                                               Else
                                                                                                       Rhino.Print "found snake, size: " +
                      If Not IsEmpty(c) Then
                             Select Case c.m_intType
                                                                          CStr(Ubound(snakeCells)+1)
                                    Case 0
                                                                                                       ReDim Preserve newSnakes(i)
                                            counts(0) = counts(0) + 1
                                                                                                        'create new snake
                                    Case 1
                                                                                                       Set newSnakes(i) = New Snake
                                            counts(1) = counts(1) + 1
                                                                                                        'build snake
                                                                                                       Call newSnakes(i).build(snakeCells)
                                            counts(2) = counts(2) + 1
                                                                                                       i=i+1
                                                                                                End If
                                    Case Else
                                     'do nothing
                             End Select
                                                                                        If i>0 Then m arrSnakes = newSnakes
                              'is null b/c no cell
                             Rhino.Print "skip " + TypeName(c)
                     End If
                                                                                  `find snake-----
               Next
                                                                                 Public Function NextSnake(oSubstr, intSize)
               'return most numerous type
                                                                                        Dim oRndCell, arrOpenCells, snakeCells(), snake-
               For i=0 To Ubound(counts)
                                                                          Cells2(), i, n
                                                                                        Dim arrBB, strBB, arrVol, strSpine
                      If counts(i) = Max(counts) Then
```

			End If	
	'check for valid args		E E-	ch c In arrCells
	If IsNull(oSubstr) Or IsEmpty(oSubstr) Then Rhino.Print "NextSnake arg, oSubstr is null or		FOI Ea	c.m boolIsOpen = True
pty"			Next	
	NextSnake = Null			
	Exit Function ElseIf Not IsArray(oSubstr.m arrType0) Then	End Sı	ab	
not an arra	Rhino.Print "NextSnake arg, oSubstr.m_arrType0 y, its a: " + TypeName(oSubstr.m_arrType0)		_	on NextCell(currCell)
	NextSnake = Null Exit Function End If		Dim ar	rOpenCells
	End II		'check	args
	<pre>'get open cells arrOpenCells = OpenCells(oSubstr.m_arrType0)</pre>		If IsN	ull (currCell) Or IsEmpty(currCell) Then Rhino.Print "currCell is null or empty" NextCell = Null
	If IsNull(arrOpenCells) Then			Exit Function
	'found a short snake, break & return null		ElseIf	Not TypeName(currCell) = "Cell" Then
	Rhino.Print "no more snakes" NextSnake = Null	TypeName(curr	Cell)	Rhino.Print "currCellis not a Cell, its a: " +
	Exit Function	1) portamo (oull	0011)	NextCell = Null
	End If		End If	Exit Function
	'get random cell from open cells		1 - 1 1	
	Set oRndCell = RndCell(arrOpenCells)		'check	<pre>args ull(currCell.m arrType0VNs) Or IsEmpty(currCell.m</pre>
	'create snake ReDim Preserve snakeCells(0)	arrType0VNs)		Rhino.Print "currCell.m arrTypeOVNs is null or
	Set snakeCells(0) = oRndCell	empty"		drrijpovino 10 narr 01
	<pre>snakeCells(0).m_boolIsOpen = False</pre>			NextCell = Null
	i=1		Flootf	Exit Function
	'while snake size is not met Do While i < intSize		FISEII	Not IsArray(currCell.m_arrType0VNs) Then Rhino.Print "currCell.m_arrType0VNs is not an
	<pre>'if no where for snake to grow, break If IsNull(NextCell(snakeCells(i-1))) Then 'found a short snake</pre>	array, its a:	" + Туре	eName(currCell.m_arrType0VNs) NextCell = Null Exit Function
]]() roturno	Rhino.Print "NextSnake call to Nextdonull, its just means its a short snake"		End If	
II() Tecurne	'release cells and return null and exit ReleaseCells(snakeCells) NextSnake = Null			pen von neumanns nCells = OpenCells(currCell.m_arrType0VNs)
	Exit Function		If Not	IsNull(arrOpenCells) Then
	End If			<pre>'pick one at random Set NextCell = RndCell(arrOpenCells)</pre>
	'get next cell		Else	
	<pre>Set n = NextCell(snakeCells(i-1)) 'add to tmp array</pre>	manns"		Rhino.Print "NextCell found no open von neu-
	ReDim Preserve snakeCells(i)	manns		NextCell = Null
	Set snakeCells(i) = n		End If	
	<pre>'close cell snakeCells(i).m_boolIsOpen = False</pre>	End Fi	unction	
	i=i+1 Loop	\		
	поор			neSnakes()
	'if snake is 2d, reject it, release cells, and return		_	
11	If CellsCoplanar(snakeCells) Then		Dim sn	ake, strNewSpine
ct"	Rhino.Print "NextSnake produced 2d snake, re-		For Ea	ch snake In m_arrSnakes
	'release cells and try again ReleaseCells(snakeCells)			strNewSpine = StrokeCells(snake.m_arrCells)
	<pre>'try again recursively NextSnake = NextSnake(oSubstr,intSize)</pre>			<pre>If Not IsNull(strNewSpine) Then</pre>
	Exit Function			Else
	End If			`smth is wrong Rhino.Print ``Null Spine, this should
	'return valid snake	never happen"		initio, itali opino, onto onouta
	<pre>NextSnake = snakeCells</pre>			Exit Sub
End Fun	action			End If
1-1-0	rs	End Su	Next	
_	Sub ReleaseCells(arrCells)	Ena Si	10	
1 422110	Dim c			pts coplanar?? on CellsCoplanar(arrCells)
	'check arg is valid non-empty array	111441		arr3dPts(), strPt
	If IsEmpty(arrCells) Or IsNull(arrCells) Then		m +/	
	Rhino.Print "ReleaseCells arg, arrCells is empty		For i=	0 To Ubound(arrCells)
null"	Exit Sub			'get pts ReDim Preserve arr3dPts(i)
	ElseIf Not IsArray(arrCells) Then			strPt = arrCells(i).m strPt
	Rhino.Print "ReleaseCells arg, arrCells is not			arr3dPts(i)=Rhino.PointCoordinates(strPt)
array, its	a: " + TypeName(arrCells) Exit Sub		Next	

		End Sub
	'return true if pts are coplanar	
	CellsCoplanar = Rhino.PointsAreCoplanar(arr3dPts, 1)	End Class
End Fun	ation	\
End run	(CC10II	'SNAKE CLASS
'return	s polyline from ordered cell array	1
	Function StrokeCells(arrCells)	1
		·
	Dim i, arr3dPts(), strPt, strShortSpine, strLongSpine	Class Snake
	The 1 0 me tile (1/2) (0.11a)	P. hilling at a Oaling
	For i=0 To Ubound(arrCells) 'get pts for next snake	Public m_strSpine Public m strSkin
	ReDim Preserve arr3dPts(i)	Public m arrCells
	strPt = arrCells(i).m strPt	Public m intSegs
	arr3dPts(i)=Rhino.PointCoordinates(strPt)	
	'label joints	Public Sub Build(arrCells)
	<pre>Call Rhino.AddText(CStr(i), arr3dPts(i))</pre>	
		Dim i
	Next	m ammCalla — ammCalla
	Narrata polulina	<pre>m_arrCells = arrCells m intSegs = Ubound(arrCells)+1</pre>
	'create polyline If Not IsEmpty(arr3dPts) Then	m_incsegs - obound(allcells)+1
	'create spine	'For i=0 To Ubound(arrCells)
	strShortSpine = Rhino.AddPolyline(arr3dPts)	' arrCells(i).m boolIsOpen = False
	'extend curve to accound for snub ends	'Next
	<pre>strLongSpine = Rhino.ExtendCurveLength(strShortS</pre>	End Sub
pine, 0, 2, 2.5	5)	
	'assign to snake	End Class
	<pre>'m_arrSnakes(i).m_strSpine = strLongSpine</pre>	'
	Else	
	Rhino.Print "arr3dPts is empty" StrokeCells = Null	'[SNAKE]SKIN CLASS
	End If	i
		\
	'return polyline	Class Skin
	StrokeCells = strLongSpine	
_		Private arrScales
End Fun	ction	Total Class
Public	Sub SkinSnakes(dblRadius)	End Class
TUDITO	Sub Skinonakes (ublikatius)	\
	Dim i, strP0, strP1, arrP0, arrP1	'[SNAKE]SPINE CLASS
	Dim arrNorm, arrPlane, arrPlane2, strCirc, arrBB	1
	Dim strCrossSect, strSpine, arrSweeps, strSkin	1
	For i=0 To Ubound(m_arrSnakes)	Class Spine
	'get orgin	lorder array of nto
	<pre>'strP0 = m_arrSnakes(i).m_arrCells(0).m_strPt</pre>	`order array of pts Private arrCells
	'strP1 = m arrSnakes(i).m arrCells(1).m strPt	TITVUCC UTTOCTTO
	'arrP0 = Rhino.PointCoordinates(strP0)	'render snake method
	<pre>'arrP1 = Rhino.PointCoordinates(strP1)</pre>	
	'get normal vector	End Class
	'arrNorm = Rhino.VectorCreate (arrP0, arrP1)	
	'create unit plane from normal vector	\[ONAVE]OOATE OTAGO
rNorm)	'arrPlane = Rhino.PlaneFromNormal (arrP0, ar-	'[SNAKE]SCALE CLASS
INOIMI		1
	'get spine	·
	<pre>strSpine = m_arrSnakes(i).m_strSpine</pre>	Class Scale
	'curve frame	
0.45	<pre>arrPlane2 = Rhino.CurvePerpFrame(strSpine,</pre>	Private strType
-2.45)		Private oSrf
	If Not IsArray(arrPlane2) Then	<pre>'maybe pointers to neighboors (ids) 'or shit maybe just reference the obj</pre>
	Rhino.Print TypeName(arrPlane2)	or only maybe just reference the obj
	Rhino.ObjectColor m strSpine, RGB(0,	End Class
255, 0)	-	
	End If	
	'add cirlcle	
	strCirc = Rhino.AddCircle (arrPlane2, dblRadius)	Option Explicit
	<pre>'get bounding box of circle arrBB = Rhino.BoundingBox(strCirc, arrPlane2)</pre>	'Script written by Josh Parker 'Script copyrighted by OPEN Architecture
	'create crosssection curv	'Script version Tuesday, February 15, 2011 3:25:36 AM
	strCrossSect = Rhino.AddPolyline(arrBB)	
	'sweep spine	Class Substrate
	arrSweeps = Rhino.AddSweep1(strSpine, strCross-	
Sect)		Public m_arrCells
	'get skin guid	Public m_intSpacing
Mana a Marina d	'Rhino.Print "arrSweeps type: " +	Public m_intSteps
TypeName(arrSwe		Public m_intLevels
	<pre>'strSkin = arrSweeps(0) 'assine skin to snake</pre>	`subsets
	'm arrSnakes(i).m strSkin = strSkin	Public m arrType0
	Next	Public m_arrType1

Public	m_arrType2		Rhino.Print "G	rouping	Cells"
Privat	e Sub Class_Initialize 'Rhino.Print "Initialize substrate"		Dim type0(), ty		type2()
End Su			a=0 b=0		
Privat	e Sub Class_Terminate		c=0		
End Su	'Rhino.Print "Terminate substrate" b			_	ells Type: " + TypeName(m_arrCells ells Size: " + CStr(Ubound(m_ar-
'seed	substrate	rCells))	MILIO.IIIIC I	arrcc	iii bize. Goti (obouna (m_ar
Public	<pre>Sub Build(intSpacing, intSteps, intLevels)</pre>		For i=0 To Ubou	_	
	m intensing - intensing		'group		e "Select Type: " + CStr(m
	<pre>m_intSpacing = intSpacing m intSteps = intSteps</pre>	arrCells(i).m		FIIIIC	Serect Type. + CSCI (III_
	m_intLevels = intLevels	_	•	Case m	_arrCells(i).m_intType
End Su	b			Case 0	ReDim Preserve typeO(a) Set typeO(a) = m arrCells(i)
'creat	e random-type cell array from points				a=a+1
	Sub AddCells(arrPts)			Case 1	
	Rhino.Print "Adding Cells"				ReDim Preserve type1(b) Set type1(b) = m_arrCells(i) b=b+1
	Dim newCells(), i			Case 2	
	If IsNull(arrPts) Then				ReDim Preserve type2(c)
	Rhino.Print "AddCells arg, arrPts, is null"				Set type2(c) = m_arrCells(i)
	Exit Sub End If			Case E	c=c+1
	III II			case i	Rhino.Print "GroupCells() found
	For i=0 To Ubound(arrPts)	bad cell"			
	<pre>'incr cell array by 1 ReDim Preserve newCells(i)</pre>		End Sel Next	ect	
	<pre>'create new cell i Set newCells(i) = New Cell</pre>		'copy if smth t	to copy	, else leave null
	'cell "constructor"		If $a > 0$ Then r		
	$\label{eq:newCells} \mbox{newCells(i).Build arrPts(i), RndInt(0,2)} \\ \mbox{Next}$		If $b > 0$ Then r If $c > 0$ Then r		
	\"copy" array		ADhino Drint Wr	m arrmii	mon. " + CS+r(libound(m arr#unon))
	m_arrCells = newCells		'Rhino.Print "r	m_arrTy	<pre>rpe0: " + CStr(Ubound(m_arrType0)) rpe1: " + CStr(Ubound(m_arrType1)) rpe2: " + CStr(Ubound(m arrType2))</pre>
End Su	b				
\creat	e random-type cell array from points	End Sul	0		
	Sub AddCellsByColor(arrPts)	`color	substrate by ty	pe	
	Rhino.Print "Adding Cells By color"	Public	Sub ColorByType		
	Dim newCells(), i, intType		Rhino.Print "Co	oloring	Cells"
	Dim newcers (), i, incrype		Dim i, color		
	If IsNull(arrPts) Then		For i=0 To Ubou	_	
	Rhino.Print "AddCells arg, arrPts, is null" Exit Sub		'color		type ls(i).UpdateColor()
	End If		Next	_arrcer	is(i).opdatecolor()
		End Sul			
	For i=0 To Ubound(arrPts) 'incr cell array by 1	\			
	ReDim Preserve newCells(i)		Sub Apply(oRule		
	'create new cell i		Rhino.Print "Ap	pplying	ruleset"
	Set newCells(i) = New Cell 'cell "constructor"		Dim i, j, iRano	d	
	Call newCells(i).Build(arrPts(i),		For j=0 To int		
TypeFromColor					be replaced by random cell picker
	Next		For 1=U		ound(m_arrCells) oint(string id) at random
	'"copy" array				RndInt(0, Ubound(m_arrCells))
	<pre>m_arrCells = newCells</pre>				pt by type
End Su	h	Apply(oRuleSet)	Call m	_arrCells(iRand).
End Su	D	whhil (ownieser	Next		
	ct to neighbors				
Public	Sub ConnectCells() Rhino.Print "Connecting Cells"		Next		
	Millio. Filine Connecting CCTS		'done so regrou	up cell	s by type
	Dim i		<pre>GroupCells()</pre>		
	For i=0 To Ubound(m_arrCells) Rhino.Print "Connecting Cells " + CStr(i) + " of	End Suk			
" + CStr(Ubour	nd(m_arrCells)) 'connect cell to its neighbors				
	m arrCells(i).Connect m arrCells, 5		SCRA		
	Next	`			
End Su	h	\ra+;;;;	n reference to r	andom o	cell
Elia Su	~		n reference to r Function RndCel		
	cells by type				
Public	Sub GroupCells()		RndCellold = m	_arrCel	<pre>ls(RndInt(0, Ubound(m_arrCells)))</pre>

```
Randomize
                                                                                  'RndInt=Int((highest - lowest) * Rnd + lowest)
       End Function
                                                                                 RndInt=Int((highest - lowest + 1) * Rnd + lowest)
       'return reference to random cell
       Public Function RndTypeOold()
                                                                          End Function
              RndTypeOold = m_arrTypeO(RndInt(0, Ubound(m_arrTypeO)))
                                                                          'return reference to random open type0 cell
       End Function
                                                                          'this function will eventually have to time-
       'return reference to random open type0 cell
                                                                          Function OpenCells(arrCells)
       'this function will eventually have to timeout
       Public Function RndOpenTypeOold()
                                                                                 Dim i, j, arrOpenCells()
                                                                                  'check arg is valid non-empty array
               Set c = m arrType0(RndInt(0, Ubound(m arrType0)))
                                                                                 If IsNull(arrCells) Then
              If c.m_boolIsOpen Then
                                                                                         Rhino.Print "OpenCells arg is null"
                                                                                         OpenCells = Null
                      return c
                      RndOpenTypeOold = c
                                                                                         Exit Function
              Else
                                                                                 ElseIf Not IsArray(arrCells) Then
                                                                                        Rhino.Print "OpenCells arg is not an array, its a: " +
                      'not found try again
                      RndOpenTypeOold = RndOpenTypeO()
                                                                          TypeName(arrCells)
              End If
                                                                                         OpenCells = Null
                                                                                         Exit Function
                                                                                 End If
       End Function
End Class
                                                                                 For i=0 To Ubound(arrCells)
Option Explicit
'Script written by Josh Parker
                                                                                         'if open, add to array
'Script copyrighted by OPEN Architecture
                                                                                         If arrCells(i).m_boolIsOpen Then
'Script version Tuesday, February 15, 2011 3:25:36 AM
                                                                                                ReDim Preserve arrOpenCells(j)
                                                                                                Set arrOpenCells(j) = arrCells(i)
Call Main()
                                                                                                j=j+1
Sub Main()
                                                                                         End If
       Dim arrPts, intSteps, intLevels, intSpacing, oMajRule
       Dim oSubstr, oSnakeFarm
                                                                                  'return subsets if any open, else null
                                                                                 If j>0 Then
       'get initial data
                                                                                         'Rhino.Print "OpenCells found some open cells in array"
       arrPts=Rhino.GetObjects("select points to populate",1)
                                                                                         OpenCells = arrOpenCells
       If Not IsArray(arrPts) Then
                                                                                 Else
              Rhino.Print "no points selected"
                                                                                         Rhino.Print "OpenCells found no open cells in array"
                                                                                         OpenCells = Null
              Exit Sub
       End If
                                                                                         Exit Function
       intSpacing=Rhino.GetInteger("enter cell spacing",5)
                                                                                 End If
       intSteps=Rhino.GetInteger("enter number of steps",2)
       intLevels=Rhino.GetInteger("enter number of levels",2)
                                                                          End Function
       'create substrate
                                                                          'return reference to random open type0 cell-----
       Set oSubstr = New Substrate
                                                                          Function RndCell(arrCells)
       'substrate "constructor"
       Call oSubstr.Build(intSpacing, intSteps, intLevels)
                                                                                  'check arg is valid non-empty array
       'add Cells from points
                                                                                  If IsNull(arrCells) Then
       Call oSubstr.AddCells(arrPts)
                                                                                         Rhino.Print "RndCell arg is null"
       'link each cell to its neighbors
                                                                                         RndCell = Null
       Call oSubstr.ConnectCells()
                                                                                         Exit Function
                                                                                 ElseIf Not IsArray(arrCells) Then
                                                                                        Rhino.Print "RndCell arg is not an array, its a: " +
       'create cellular automata obj
                                                                          TypeName(arrCells)
       Set oMajRule = New MajorityRule
                                                                                         RndCell = Null
       'apply ruleset to each cell x times
                                                                                         Exit Function
       Call oSubstr.Apply(oMajRule, 1)
                                                                                 End If
       'color cell pts by cell type
       Call oSubstr.ColorByType()
                                                                                  'arg is non-zero sized array to return random value from it
                                                                                 Set RndCell = arrCells(RndInt(0, Ubound(arrCells)))
       'create snakes
       Set oSnakeFarm = New SnakeFarm
                                                                          End Function
       `locate snakes in substrate
       Call oSnakeFarm.FindSnakes(oSubstr, 4)
                                                                          'return reference to random open type0 cell-----
       'render snakes?
                                                                          Function TypeFromColor(strPt)
       Call oSnakeFarm.RenderSnakes()
       'create massing
       'Set oMass = New massing
                                                                                 c = Rhino.ObjectColor(strPt)
       'build mass from snake formation
       'oMass.build(oSnakes)
                                                                                  'set color var by cell type
                                                                                  Select Case c
                                                                                        Case RGB(255, 0, 0)
                                                                                        typ = 0 Case RGB(0, 255, 0)
End Sub
Option Explicit
'Script written by <insert name>
                                                                                        typ = 1 Case RGB(0, 0, 255)
'Script copyrighted by <insert company name>
'Script version Wednesday, February 16, 2011 8:20:37 PM
                                                                                               typ = 2
                                                                                         Case Else
Function RndInt(lowest, highest)
                                                                                                Rhino.Print "pt is not valid color: " + c
                                                                                                Exit Function
```

```
End Select
       'return type
       TypeFromColor = typ
End Function
\_____
'----SCRAP--
'_____
'If you need to repeat a sequence of random numbers,
'you should call the Rnd function with a negative
 'number as an argument immediately prior To using
'Randomize With any numeric argument.
Sub RepeatNumbers()
       Dim arr(9, 3)
       Dim loopCtr, intCtr
       Dim strMsg
       For loopCtr = 0 To 3
              Rnd -1
              Randomize(100)
              For intCtr = 0 To 9
                    strMsg = strMsg & Rnd() & " "
              Next
              strMsg = strMsg & vbCrLf
       Next
       MsgBox strMsg
End Sub
'return reference to random open type0 cell
'this function will eventually have to timeout
Function RndOpenCell(arrCells)
       Set c = arrCells(RndInt(0, Ubound(arrCells)))
       If c.m_boolIsOpen Then
               return c
              RndOpenCell = c
       Else
              'not found try again
RndOpenCell = RndOpenCell(arrCells)
       End If
End Function
Option Explicit
'Script written by Josh Parker
'Script version Tuesday, February 15, 2011 3:25:36 AM
Class Village
       Private arrStructure
       Private arrTrees
       Private arrGrass
       Private arrPave
End Class
Class Snake
      Private oSpine
Private oSkin
End Class
Class Skin
       Private arrScales
End Class
Class Spine
      'order array of pts
Private arrCells
       'render snake method
End Class
Class Scale
       Private strType
```

Private oSrf 'maybe pointers to neighboors (ids)
'or shit maybe just reference the obj End Class

UPLIFT HANGER, SOURCE CODE

Code written for Interactive paneling tool developed for use in design of structurally integrated prefabricated concrete panel system for design competition of airship hanger in anhui, china for competition submission completed in collaboration with OPEN Architecture and the Chinese Academy of Building Research (CABR)

Uses the following processing libraries:

controlP5 - gui toolkit

- http://www.sojamo.de/libraries/controlP5/

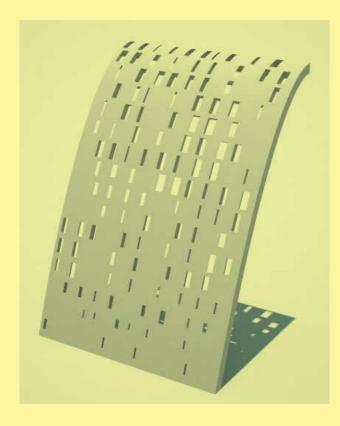
Proscene - 3d scene library

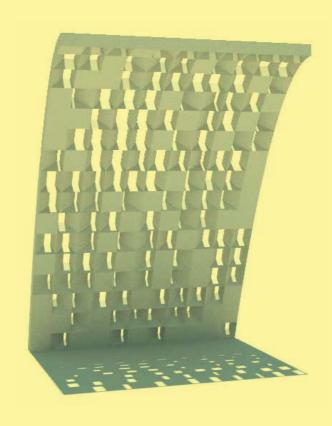
http://code.google.com/p/proscene/

Objloader - 3d object loader

http://code.google.com/p/saitoobjloader/

test sections of concrete massing generated by overlaying algorithmic patterns on top of solar data gathered from eco-tect.





```
//CONSTRUCTOR
                                                                            //MODEL: Parametric Data for panel system
Aper[na]tures
                                                                                                                                         public Controller (PApplet parent, Hanger Myhanger,
Author: Joshua Parker
                                                                            float snap = (float)((D-(T*2))/SUBDIVS);
                                                                                                                                Gui Mygui, View Myview) {
Date: 2011
                                                                           myhanger = new Hanger(this, U, V, D, T, snap,
                                                                                                                                                 p = parent;
                                                                PATH);
                                                                                                                                             mvhanger = Mvhanger;
Interactive paneling tool developed for use in
                                                                                                                                             mvaui = Mvaui;
design of structurally integrated prefabricated
                                                                            //VIEW: draws a representation of model to
                                                                                                                                             mvview = Mvview:
concrete panel system for design competition
of airship hanger in anhui, china for competition
                                                                            myview = new View(this, myhanger);
submission completed in collaboration with OPEN
                                                                                                                                          Architecture and the Chinese Academy of Building
                                                                           //GUI: needs to access view scene for drawing 2d
                                                                                                                               gui on top of 3d scene
Research (CABR)
                                                                           mygui = new Gui(this, myview);
                                                                                                                                          //gui noise increment
Implementated with Simple Model-View-Controller
                                                                                                                                          public void setNoiseInc(float v) {
                                                                           //CONTROLLER: updates model & views in response
                                                                to ui events routed to it
Gui - is a wrapper for controlP5, which supplies
                                                                           myController = new Controller(this, myhanger,
 2d user interface toolkit
MyControlListener - recieves ui events, ids and
                                                                                                                                          public void setNoiseSeed(int v) {
  routes them to controller
                                                                          //LISTENER/ROUTER: recieves ui events, ids and
                                                                                                                                                  myhanger.setSeed(v);
Controller - contains methods for updating data
                                                                routes them to controller
                                                                           myListener = new Listener(this, myController);
 model & controlling view(s)*
View - contains the scene (empty space and camera),
                                                                                                                                          //set coefficient with index i
  imported 3d base model, and data model defining
                                                                                                                                          public void setPanelXcf(int i, float C) {
 panel system. View calls draws to screen via
                                                                                                                                                  myhanger.setXcf(i, C);
  viewState interface, though only one viewState
                                                                         public void draw(){
 is implemented: View3d, a basic 3d view.
                                                                           background(50);
Hanger - data model defining panel system, composed
 of subclass skin, which is composed of panels.
                                                                           //Continually redraw view and gui via controller
                                                                                                                                         public void moveCameraByTheta(float v) {
 Panels are basically two sets of parameters: one
                                                                           myController.update();
                                                                                                                                       PVector v0;
  which defines the panel's size and position in
                                                                                                                                       float r = 500;
  space, and another that defines the size and
                                                                                                                                       float phi = 0;
  relative position of a single rectangular void
                                                                         //required to run as application
                                                                                                                                       float theta = 0;
  in the panel. Panel also contains data from ecotect
                                                                         public static void main(String args[]) {
                                                                                                                                        //r = myview.scene.camera().sceneRadius();
  about the amount of direct and diffuse radiation
                                                                          PApplet.main(new String[] { "--present", "Apernat-
                                                                                                                                       v0 = myview.scene.camera().position();
Uses the following processing libraries:
                                                                                                                                       theta = PApplet.radians(v); //azimuth 0-2PI
                                                                         //send ui events to listener, so it route them to
                                                                                                                                        setCameraPosition(phi, theta, r);
controlP5 - gui toolkit
                                                                controller
- http://www.sojamo.de/libraries/controlP5/
                                                                        public void controlEvent(ControlEvent theEvent)
Proscene - 3d scene library
                                                                {myListener.controlEvent(theEvent);}
                                                                        public void mousePressed(){myListener
                                                                                                                                         public void moveCameraByPhi(float v){
- http://code.google.com/p/proscene/
Objloader - 3d object loader
                                                                mousePressed(mouseX, mouseY);}
                                                                                                                                       PVector v0;
- http://code.google.com/p/saitoobjloader/
                                                                         public void keyPressed() {myListener.
                                                                                                                                       float r = 500;
                                                                kevPressed(kev);}
                                                                                                                                       float phi = 0;
Inheritted License: GPL, V3
                                                                                                                                       float theta = 0;
                                                                                                                                       //r = myview.scene.camera().sceneRadius();
Fourth Natures Lab | www.fnl.com
                                                                                                                                       //r = 500:
                                                                                                                                       v0 = myview.scene.camera().position();
theta = (float) Math.acos(v0.z/r);
import processing.core.*;
                                                                                                                                       phi = PApplet.radians(v); //azimuth 0-2PI
import controlP5.*;
                                                                                                                                        setCameraPosition(phi, theta, r);
import processing.opengl.*;
                                                                /****************
import controller.Listener
import controller.Controller;
import model.Hanger;
                                                                                                                                          public void setCameraPosition(float phi, float
                                                                MVC Controller contains methods for updating data
import view.*;
                                                                                                                                theta, float r) {
                                                                                                                                             float x = r*(PApplet.sin(theta)*PApplet.
import gui.*;
                                                                model & controlling view(s)*
                                                                                                                                cos(phi));
public class Apernatures extends PApplet{
                                                                Part of:
                                                                                                                                             float y = r*(PApplet.sin(theta)*PApplet.
                                                                                                                                sin(phi));
         private static final long serialVersionUID = 1L;
                                                                                                                                             float z = r*PApplet.cos(theta);
                                                                Aper[na]tures
                                                                Author: Joshua Parker
                                                                Date: 2011
                                                                                                                                             PVector v = new PVector(x,v,z);
         int U = 37; //number of modules in U direction
                                                                                                                                             PVector v2 = new PVector(0,0,0);
         int V = 42; //number of modules in V direction
                                                                Inheritted License: GPL, V3
                                                                                                                                             PVector v3 = new PVector(0,0,-1);
         float D = 10f; //panel lengh/width dimension, D
                                                                myview.scene.camera().setPosition(v);
         float T = .5f; //thickness, T of frame (meters)
                                                                package controller;
                                                                                                                                             myview.scene.camera().setOrientation(theta,
         float SUBDIVS = 20f; //subdivide panel for snapping
                                                                 import processing.core.PApplet;
                                                                                                                                              myview.scene.camera().setUpVector(v3,true);
         //path to tabular data from ecotect analyses
                                                                import processing.core.PVector;
                                                                                                                                             myview.scene.camera().lookAt(v2);
         String PATH = "data/ecotectData.txt";
                                                                import model.Hanger;
                                                                import view. View;
                                                                import gui.Gui;
          Listener myListener;
                                                                                                                                          public void focusGui(){
                                                                public class Controller{
                                                                                                                                                   myview.scene.disableMouseHandling();
          Hanger myhanger;
                                                                                                                                                   myview.scene.disableKeyboardHandling()
                                                                          //The parent PApplet
         View myview;
         Gui mygui;
                                                                         PApplet p;
         public void setup(){
                                                                         //PROPS
                                                                                                                                         public void focusScene(){
           size (1280, 1020, PGraphicsOpenGL.OPENGL);
                                                                         private Hanger myhanger;
                                                                                                                                                   myview.scene.enableMouseHandling();
           background(0);
                                                                         private View myview;
                                                                                                                                                   mvview.scene.enableKevboardHandling();
```

private Gui mvgui;

15

noStroke();

					private	e void g2ControlEvent(ControlEvent theEven
	//		import control	P5.ControlEvent;	{	
	public	<pre>void toggleTexture() {</pre>	import control	P5.ControlListener;		
		w.bTexture) {		ing.core.PApplet;		float v;
			Import process	ing.core.rappiec,		
		ture();				switch(theEvent.getController().getId(
	} else {		public class L	istener implements ControlListener{	{	
	hideTex	ture();				case(1):
	}		//Th	e parent PApplet		<pre>v = theEvent.getController()</pre>
	}		PApp	let p;	<pre>getValue();</pre>	
				• *	3	<pre>myController.setNoiseInc(v);</pre>
	//		//pp	O.D.O.		=
			//PR			break;
	public	<pre>void toggleMaterial() {</pre>	Cont	roller myController;		case(2):
		<pre>if(!myview.bMaterial) {</pre>				<pre>v = theEvent.getController()</pre>
		<pre>showMaterial();</pre>	//co	NSTRUCTOR	<pre>getValue();</pre>	
		} else {	lduq	ic Listener(PApplet parent, Controller MyCon-		<pre>myController.setNoiseSeed((int)(v));</pre>
		hideMaterial();	troller) {	11 11 11 11 11 11 11 11 11 11 11 11 11		break;
		indematerial(),	CIOITEI)(bleak,
		}		<pre>p = parent;</pre>		}
	}			<pre>myController = MyController;</pre>	}	
			}			
	//				//pars	e group 3 (effect sliders) event!
	public	void toggleFill(){	// P	UBLIC METHODS ////////////////////////////////////	******	
	F		///////////////////////////////////////		nrivat	e void q3ControlEvent(ControlEvent theEven
		<pre>if(!myview.getFill()) {</pre>	///////////////////////////////////////	/////	private	e void gacontroisvent(controisvent thesven
		showFill();			{	
		} else {	//ro	ute slider events to controller via private		
		hideFill();	handlers			float v;
		}	ldug	ic void controlEvent(ControlEvent theEvent) {		switch(theEvent.getController().getId(
	1	•			ſ	, ,
	1				î.	(5)
				<pre>switch(theEvent.getController().getPar-</pre>		case(5):
	//		ent().getId())	{		<pre>v = theEvent.getController()</pre>
	public	<pre>void showTexture() {</pre>		case(1):	<pre>getValue();</pre>	
		<pre>myview.model.enableTexture();</pre>		//route gl (shortcut info)		<pre>myController.setPanelXcf(0,v);</pre>
		myview.bTexture = true;	event!	5- (31040 10)		break;
		myview.biexture - true;	event:			
	}			//empty so break		case(6):
				break;		<pre>v = theEvent.getController()</pre>
	//			case(2):	getValue();	
	public	void hideTexture(){		//route g2 (noise sliders)		<pre>myController.setPanelXcf(1,v);</pre>
		<pre>myview.model.disableTexture();</pre>	event!			break;
				-20		case (7):
		<pre>myview.bTexture = false;</pre>		g2ControlEvent(theEvent);		
	}			break;		<pre>v = theEvent.getController()</pre>
				case(3):	<pre>getValue();</pre>	
	//			//route g3 (effect sliders)		myController.setPanelXcf(2,v);
	public	<pre>void showMaterial(){</pre>	event!			break;
		<pre>myview.model.enableMaterial();</pre>		<pre>g3ControlEvent(theEvent);</pre>		case(8):
		<pre>myview.bMaterial = true;</pre>		break;		<pre>v = theEvent.getController()</pre>
	}			case(4):	getValue();	
				//route g4 (camera dials)		<pre>myController.setPanelXcf(3,v);</pre>
	//		event!			break;
	public	<pre>void hideMaterial(){</pre>		q4ControlEvent(theEvent);		case(9):
	F			break;		
		<pre>myview.model.disableMaterial();</pre>		Dreak;		<pre>v = theEvent.getController()</pre>
		<pre>myview.bMaterial = false;</pre>		}	getValue();	
	}		}			<pre>myController.setPanelXcf(4,v);</pre>
						break;
	//		//ro	ute mouse events to controller		case(10):
		void showFill(){		is should be replaced with focus on hover		v = theEvent.getController()
	public			-		v = theEvent.getController()
		<pre>myview.enableFill();</pre>	publ	ic void mousePressed(int mx, int my) {	<pre>getValue();</pre>	
	}			if (mx < 200) {		<pre>myController.setPanelXcf(5,v);</pre>
				<pre>myController.focusGui();</pre>		break;
	//			} else {		case(11):
		void hideFill(){		<pre>myController.focusScene();</pre>		
	Puntic			myconcrorrer.rocusacene();		<pre>v = theEvent.getController()</pre>
		<pre>myview.disableFill();</pre>		}	<pre>getValue();</pre>	
	}		}			<pre>myController.setPanelXcf(6,v);</pre>
						break;
	//		//ro	ute key press events to controller		case(12):
		<pre>void update(){</pre>		ic void keyPressed(char key) {		v = theEvent.getController()
	Partic		Pubi		gg+17-1/\	v chebvent.getController()
		myhanger.update(); //update the model			<pre>getValue();</pre>	
				if(key == 't') {		<pre>myController.setPanelXcf(7,v);</pre>
		myview.draw(); // draw the 3d scene		and the second s		break;
		<pre>myview.draw(); // draw the 3d scene mygui.draw(); // then draw the gui on</pre>		//toggle the texture listed in		
top			.mtl file	//toggle the texture listed in		case(13):
top	}		.mtl file	<pre>//toggle the texture listed in myController.toggleTexture();</pre>		<pre>case(13): v = theEvent.getController()</pre>
top	}		.mtl file		gotValvo():	
}	}	<pre>mygui.draw(); // then draw the gui on</pre>	.mtl file	<pre>myController.toggleTexture(); }</pre>	<pre>getValue();</pre>	<pre>v = theEvent.getController()</pre>
}	} *****		.mtl file	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v);</pre>
} /*****		<pre>mygui.draw(); // then draw the gui on</pre>		<pre>myController.toggleTexture(); }</pre>	getValue();	<pre>v = theEvent.getController()</pre>
}		<pre>mygui.draw(); // then draw the gui on</pre>	.mtl file	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v);</pre>
} /*****		<pre>mygui.draw(); // then draw the gui on</pre>		<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break;</pre>
} /******* Listener	r Class	mygui.draw(); // then draw the gui on		<pre>myController.toggleTexture(); } else if(key == 'm') { //toggle the material listed</pre>		<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
} /***** Listener	r Class r recieves	<pre>mygui.draw(); // then draw the gui on ***********************************</pre>		<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	<pre>getValue(); getValue();</pre>	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14): v = theEvent.getController()</pre>
} /***** Listener	r Class r recieves	mygui.draw(); // then draw the gui on		<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>		<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14): v = theEvent.getController() myController.setPanelXcf(9,v);</pre>
} /***** Listener Listener	r Class r recieves controlle	<pre>mygui.draw(); // then draw the gui on ***********************************</pre>	in .mtl file	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>		<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14): v = theEvent.getController() myController.setPanelXcf(9,v); break;</pre>
} /***** Listener	r Class r recieves controlle	<pre>mygui.draw(); // then draw the gui on ***********************************</pre>		<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>		<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14): v = theEvent.getController() myController.setPanelXcf(9,v);</pre>
} /***** Listener Listener	r Class r recieves controlle	<pre>mygui.draw(); // then draw the gui on ***********************************</pre>	in .mtl file	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>		<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
} /***** Listener Listener	r Class r recieves controlle	<pre>mygui.draw(); // then draw the gui on ***********************************</pre>	in .mtl file	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>		<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
} /***** Listener them to Part of: Aper[na]	r Class r recieves controlle :	mygui.draw(); // then draw the gui on ***********************************	in .mtl file	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
} /***** Listener them to Part of: Aper[na] Author:	r Class r recieves controlle :]tures Joshua Pa	mygui.draw(); // then draw the gui on ***********************************	in .mtl file	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
} /***** Listener them to Part of: Aper[na]	r Class r recieves controlle :]tures Joshua Pa	mygui.draw(); // then draw the gui on ***********************************	<pre>in .mtl file shapes</pre>	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
listener Listener them to Part of: Aper[na] Author: Date: 20	r Class r recieves controlle :]tures Joshua Pa	mygui.draw(); // then draw the gui on ***********************************	<pre>in .mtl file shapes } // P</pre>	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
listener Listener them to Part of: Aper[na] Author: Date: 20	r Class r recieves controlle :]tures Joshua Pa	mygui.draw(); // then draw the gui on ***********************************	<pre>in .mtl file shapes</pre>	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
listener Listener them to Part of: Aper[na] Author: Date: 20	r Class r recieves controlle :]tures Joshua Pa	mygui.draw(); // then draw the gui on ***********************************	<pre>in .mtl file shapes } // P</pre>	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	getValue();	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>
listener Listener them to Part of: Aper[na] Author: Date: 20	r Class r recieves controlle :]tures Joshua Pa 011 ted Licens	mygui.draw(); // then draw the gui on ***********************************	in .mtl file shapes } // P	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	<pre>getValue(); getValue();</pre>	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14): v = theEvent.getController() myController.setPanelXcf(9,v); break; case(15): v = theEvent.getController() myController.setPanelXcf(10,v); break; case(16): v = theEvent.getController()</pre>
listener Listener them to Part of: Aper[na] Author: Date: 20 Inheritt	r Class r recieves controlle :]tures Joshua Pa 011 ted Licens	mygui.draw(); // then draw the gui on ***********************************	in .mtl file shapes // P	<pre>myController.toggleTexture(); } else if(key == 'm') {</pre>	<pre>getValue(); getValue();</pre>	<pre>v = theEvent.getController() myController.setPanelXcf(8,v); break; case(14):</pre>

getValue(
getValue(<pre>v = theEvent.getController().</pre>	//parent	PApplet		.setColorBackground	d(BarClr)	
-	();		PApplet	p;				
		<pre>myController.setPanelXcf(12,v);</pre>				.setColorForeground	d(HoverClr)	
		break;	//cp5 ob					
		case(18):		ControlP5 cp5;		.setColorLabel(Labe	,	
getValue(() •	<pre>v = theEvent.getController().</pre>	private	Accordion accordion;			;	
getvalue((),	<pre>myController.setPanelXcf(13,v);</pre>	//compos	it view			// group 2 (noise sli	dore)
		break;		View myview;			controlP5.Group g2 =	
		case (19):	pilvaco	view myview,		addGroup("myGroup2		opo.
		<pre>v = theEvent.getController().</pre>	//CONSTF	UCTOR				setId(2)
getValue(();		public 0	Gui(PApplet parent, V:	iew Myview) {			setLabel("Noise")
		<pre>myController.setPanelXcf(14,v);</pre>					. 1	setBackground-
		break;		//initialize		Height (40)		
		case(20):		<pre>p = parent;</pre>				setBarHeight(BarH)
		<pre>v = theEvent.getController().</pre>		cp5 = new ControlP	5(p);			
getValue(();			cp5.setAutoDraw(fa	lse);	.setBackgroundColo	r(BgClr) //bg of block	
		<pre>myController.setPanelXcf(15,v);</pre>		myview = Myview;				
		break; case(21):		//		.setColorBackground	d(BarClr) //bar color	
		<pre>case(21): v = theEvent.getController().</pre>		<pre>//setup gui element setupP5Gui();</pre>	T.S.	sotColorForogroup	d(HoverClr) //bar color	on hower
getValue(() •	v - thervent.getcontroller().	ì	secuprodui();		.seccolorroreground	u(Novelcii) //bai coloi	OII HOVEL
getvarue((),	<pre>myController.setPanelXcf(16,v);</pre>	1			.setColorLabel(Labe	elClr) //label text col	or
		break;	// PUBI.1	C METHODS ////////	///////////////////////////////////////		;	
		case (22):	///////////////////////////////////////				,	
		v = theEvent.getController().					// group 3 (effect sl	iders)
getValue(();		public v	oid draw() {			controlP5.Group g3 =	cp5.
		<pre>myController.setPanelXcf(17,v);</pre>				addGroup("myGroup3	")	
		break;		//save scene (so ye	ou can draw 2d gui on			setId(3)
			top)					setLabel("x = f()")
		}		saveState(myview.so	cene);		• 1	setBackground-
	}					Height (300)		
				//draw gui			•	setBarHeight(BarH)
	//parse (group 4 (camera dials) event! *********		cp5.draw();		ant BankaroundColor	r/BaClr) //ba of blook	
		void g4ControlEvent(ControlEvent theEvent)		//restore scene		.SetBackgroundcoro.	r(BgClr) //bg of block	
{	privace	voia greeneroibvene (concretiblene enebvene)		restoreState (myview	w.scene):	.setColorBackground	d(BarClr) //bar color	
			}		,,		, , , , , , , , , , , , , , , , , , , ,	
		float v;				.setColorForeground	d(HoverClr) //bar color	on hover
		<pre>switch(theEvent.getController().getId())</pre>	// PRIVA	TE METHODS ///////	///////////////////////////////////////			
{			///////////////////////////////////////	///		.setColorLabel(Labe	elClr) //label text col	or
		case(13):					;	
		<pre>v = theEvent.getController().</pre>	=	p5 gui elements				
getValue(();		private	void setupP5Gui(){			// group 4 (camera di	
		<pre>myController.moveCameraByPhi(v);</pre>					controlP5.Group g4 =	cps.
		hh		i-+ D 20.		addCrann / Nmr.Crann /	// \	
		break;		int BarH = 20;	r(50) · //ba of block	addGroup("myGroup4		setId(4)
		case(14):		int BgClr = p.colo	r(50); //bg of block	addGroup("myGroup4'		setId(4)
getValue(();			<pre>int BgClr = p.colo int BarClr = p.colo</pre>	or(50); //bar color	addGroup("myGroup4		setLabel("view")
getValue(();	case(14):		<pre>int BgClr = p.colo int BarClr = p.colo</pre>	or(50); //bar color //bar color on hover	addGroup("myGroup4' Height(200)		
getValue(();	case(14): $v \; = \; \text{theEvent.getController()} \; . \label{eq:velocity}$		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30;</pre>	or(50); //bar color //bar color on hover			setLabel("view")
getValue(();	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30;</pre>	or(50); //bar color //bar color on hover			setLabel("view") setBackground-
getValue((); }	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col</pre>	or(50); //bar color //bar color on hover	Height (200)		setLabel("view") setBackground-
getValue(();	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10;</pre>	or(50); //bar color //bar color on hover	Height(200) .setBackgroundColo		setLabel("view") setBackground-
getValue(();	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5;</pre>	or(50); //bar color //bar color on hover plor(255);	Height(200) .setBackgroundColo	.: .: .:	setLabel("view") setBackground-
}	}	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10;</pre>	or(50); //bar color //bar color on hover plor(255);	Height(200) .setBackgroundColo: .setColorBackground	r(BgClr) //bg of block d(BarClr) //bar color	setLabel("view") setBackground- setBarHeight(BarH)
}	}	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc;	Height(200) .setBackgroundColo: .setColorBackground		setLabel("view") setBackground- setBarHeight(BarH)
} /*****	}	<pre>case(14):</pre>	11).	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide</pre>	or(50); //bar color //bar color on hover plor(255);	Height (200) .setBackgroundColor .setColorBackground	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color	setLabel("view") setBackground- setBarHeight(BarH) on hover
}	}	<pre>case(14):</pre>	11);	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc;	Height (200) .setBackgroundColor .setColorBackground	r(BgClr) //bg of block d(BarClr) //bar color	setLabel("view") setBackground- setBarHeight(BarH) on hover
} /*********	}	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 =</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc;	Height (200) .setBackgroundColor .setColorBackground	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color	setLabel("view") setBackground- setBarHeight(BarH) on hover
} /******** Gui Class	} *********** s wrapper fo	<pre>case(14):</pre>	<pre>11); p.createFont("Aria</pre>	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 =</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc;	Height (200) .setBackgroundColor .setColorBackground	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color	setLabel("view") setBackground- setBarHeight(BarH) on hover
} /****** Gui Class Gui is a provides	} ******** wrapper for user inter	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 =</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc;	Height (200) .setBackgroundColor .setColorBackground	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color	setLabel("view") setBackground- setBarHeight(BarH) on hover
} /****** Gui Class Gui is a provides	} ******** wrapper for user inter	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10);</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30",	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text colo;	setLabel("view") setBackground- setBarHeight(BarH) on hover or
} /****** Gui Class Gui is a provides	} ******** wrapper for user inter	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideW = 100; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30",	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s.	setLabel("view") setBackground- setBarHeight(BarH) on hover or
/****** Gui Class Gui is a provides http://ww	<pre>******** wrapper for user inter www.sojamo.or </pre>	<pre>case(14):</pre>		<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(":</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30",	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s.	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials **
/******* Gui Class Gui is a provides http://ww Part of:	wrapper for user interwww.sojamo.o	<pre>case(14):</pre>	p.createFont("Aria	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(":</pre>	or(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30",	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color; // ADD CONTROLLERS: s	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ***********
/******* Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J	wrapper for user interwww.sojamo.o	<pre>case(14):</pre>	p.createFont("Aria	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(":</pre>	cr(50); //bar color //bar color on hover color(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p.	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s ************************************	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ***********
/******* Gui Class Gui is a provides http://ww Part of:	wrapper for user interwww.sojamo.o	<pre>case(14):</pre>	p.createFont("Aria	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(":</pre>	cr(50); //bar color //bar color on hover color(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10)	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s. ************************************	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ***********
/******* Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201	wrapper for user interwww.sojamo.o	<pre>case(14):</pre>	p.createFont("Aria	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(":</pre>	cr(50); //bar color //bar color on hover color(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20)	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s. ************************************	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ***********
/******* Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201	wrapper for user interwww.sojamo.o	<pre>case(14):</pre>	p.createFont("Aria	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(":</pre>	cr(50); //bar color //bar color on hover color(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10)	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s. ************************************	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ***********
/******* Sui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte	wrapper for user intervented www.sojamo.or	<pre>case(14):</pre>	p.createFont("Aria	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.co int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.coo PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(":</pre>	cr(50); //bar color //bar color on hover plor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200)	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s. ************************************	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ***********
/******* Sui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte	wrapper for user intervention was of the state of the sta	<pre>case(14):</pre>	p.createFont("Aria	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = clMT-30", 10); //add footer cp5.addTextlabel(": fn1.com")</pre>	cr(50); //bar color //bar color on hover plor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200)	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s. ************************************	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ***********
/******* Sui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte	wrapper for user intervention was of the state of the sta	<pre>case(14):</pre>	<pre>p.createFont("Aria Natures Lab www. height - 20)</pre>	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = clMT-30", 10); //add footer cp5.addTextlabel(": fn1.com")</pre>	cr(50); //bar color //bar color on hover plor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200);	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color ; // ADD CONTROLLERS: s. ************************************	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ************ lers)
} /****** Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte ********* package g	wrapper for user intervented www.sojamo.or	<pre>case(14):</pre>	<pre>p.createFont("Aria Natures Lab www. height - 20)</pre>	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = clMT-30", 10); //add footer cp5.addTextlabel(": fnl.com")</pre>	cr(50); //bar color //bar color on hover plor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200);	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color // ADD CONTROLLERS: s. //add group 1 control cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setColorForeground(p	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ********** lers))) .color(50)) .color(255,100))
} /****** Sui Class Sui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte ******** package g import pr import pr	wrapper for user intervented www.sojamo.or	<pre>case(14):</pre>	<pre>p.createFont("Aria Natures Lab www. height - 20)</pre>	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(": fnl.com") // ADD GROUPS ***** // group 1 (shortcom)</pre>	cr(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200) ;	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color // ADD CONTROLLERS: s. //add group 1 control cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setColorForeground(p .setText("shortcuts:	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** *********** lers))) .color(50)) .color(255,100)) \n"
} /****** Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte ******* package g import pr import pr import co	wrapper for user intervented www.sojamo.or ures Joshua Parill ed License ***********************************	<pre>case(14):</pre>	p.createFont("Aria Natures Lab www. height - 20)	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideW = 100; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(") fnl.com") // ADD GROUPS ***** // group 1 (shortcolor) group gl</pre>	cr(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200) ;	Height (200) .setBackgroundColor .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color // ADD CONTROLLERS: s. //add group 1 control cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setColorForeground(p .setText("shortcuts:	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ********** lers))) .color(50)) .color(255,100))
Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte package g import pr import pr import co import re	wrapper for user interest www.sojamo.or ures Joshua Parill ed License *********** gui; roccessing.or user interest was a second or user interest was a second or user in the sec	<pre>case(14):</pre>	<pre>p.createFont("Aria Natures Lab www. height - 20)</pre>	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideW = 100; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(") fnl.com") // ADD GROUPS ***** // group 1 (shortcolor) group gl</pre>	cr(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200) ;	Height (200) .setBackgroundColo: .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color // ADD CONTROLLERS: s. //add group 1 control. cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setText("shortcuts: +	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ********** lers) .color(50)) .color(255,100)) \n" "f - toggle fills
Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte package g import pr import pr import co import re	wrapper for user interest www.sojamo.or ures Joshua Parill ed License *********** gui; roccessing.or user interest was a second or user interest was a second or user in the sec	<pre>case(14):</pre>	p.createFont("Aria Natures Lab www. height - 20) ***********************************	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(") fnl.com") // ADD GROUPS **** // group 1 (shortcolor) controlP5.Group g1 ")</pre>	cr(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200) ;	Height (200) .setBackgroundColor .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text color // ADD CONTROLLERS: s. //add group 1 control. cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setText("shortcuts: +	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** *********** lers))) .color(50)) .color(255,100)) \n"
Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte package g import pr import pr import co import vi	wrapper for user interest www.sojamo.co	<pre>case(14):</pre>	p.createFont("Aria Natures Lab www. height - 20)	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(") fnl.com") // ADD GROUPS **** // group 1 (shortcolor) controlP5.Group g1 ")</pre>	cr(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200) ; **********************************	Height (200) .setBackgroundColor .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text col ; // ADD CONTROLLERS: s. ************* //add group 1 control cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setText("shortcuts: +	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ********** lers)) .color(50)) .color(255,100)) \n" "f - toggle fills "e - toggle per-
Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte package g import pr import pr import co import re	wrapper for user interest www.sojamo.co	<pre>case(14):</pre>	p.createFont("Aria Natures Lab www. height - 20) ***********************************	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(") fnl.com") // ADD GROUPS **** // group 1 (shortcolor) controlP5.Group g1 ")</pre>	cr(50); //bar color //bar color on hover blor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200) ;	Height (200) .setBackgroundColor .setColorBackground .setColorForeground .setColorLabel (Labe	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text col ; // ADD CONTROLLERS: s. ************* //add group 1 control cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setText("shortcuts: +	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ********** lers) .color(50)) .color(255,100)) \n" "f - toggle fills
Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte package g import pr import pr import co import vi	wrapper for user interest www.sojamo.co	<pre>case(14): v = theEvent.getController(). myController.moveCameraByTheta(v); break; } **********************************</pre>	p.createFont("Aria Natures Lab www. height - 20) ***********************************	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(") fnl.com") // ADD GROUPS **** // group 1 (shortcolor) controlP5.Group g1 ")</pre>	cr(50); //bar color //bar color on hover plor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,p. .setFont(arial10) .setHeight(20) .setWidth(200) ; **********************************	Height (200) .setBackgroundColor .setColorForeground .setColorLabel (Label ***********************************	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text col. ; // ADD CONTROLLERS: s. ************ //add group 1 control. cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setText("shortcuts: + +	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ********** lers)) .color(50)) .color(255,100)) \n" "f - toggle fills "e - toggle per-
Gui Class Gui is a provides http://ww Part of: Aper[na]t Author: J Date: 201 Inheritte package g import pr import pr import co import vi	wrapper for user interest www.sojamo.or ures Joshua Parill ed License ***********************************	<pre>case(14): v = theEvent.getController(). myController.moveCameraByTheta(v); break; } **********************************</pre>	p.createFont("Aria Natures Lab www. height - 20) ***********************************	<pre>int BgClr = p.colo int BarClr = p.colo int HoverClr = 30; int LabelClr = p.col int slideH = 10; int slideW = 100; int slideX = 10; int slideSpc = 5; int slideDY = slide PFont arial = p.cre PFont arial10 = ilMT-30", 10); //add footer cp5.addTextlabel(") fnl.com") // ADD GROUPS **** // group 1 (shortcolor) controlP5.Group g1 ")</pre>	cr(50); //bar color //bar color on hover plor(255); eH+slideSpc; eateFont("ArialMT-30", label") .setText("by Fourth .setPosition(2,psetFont(arial10) .setHeight(20) ; **********************************	Height (200) .setBackgroundColor .setColorForeground .setColorLabel (Label ***********************************	r(BgClr) //bg of block d(BarClr) //bar color d(HoverClr) //bar color elClr) //label text col. ; // ADD CONTROLLERS: s. ************ //add group 1 control. cp5.addTextarea("txt" .moveTo(g1) .setPosition(5,5) .setSize(195,90) .setFont(arial) .setLineHeight(14) .setColor(p.color(128 .setColorBackground(p .setText("shortcuts: + +	setLabel("view") setBackground- setBarHeight(BarH) on hover or liders and dials ** ********** lers)) .color(50)) .color(255,100)) \n" "f - toggle fills "e - toggle per- "c - close controls

//add group 2 controllers	.snapToTickMarks(true)	.setNumberOfTickMarks(100)
3	.showTickMarks(false);	.snapToTickMarks(true)
cp5.addSlider("noise inc")		.showTickMarks(false);
.setId(1)	cp5.addSlider("x trim")	.Showileswarks (laise) /
.moveTo(g2)		<pre>cp5.addSlider("h offset")</pre>
-	.setId(13)	-
.setPosition(slideX,slideDY*1)	.moveTo(g3)	.setId(21)
.setSize(slideW,slideH)	<pre>.setPosition(slideX, slideDY*9)</pre>	.moveTo(g3)
.setRange(0.0f,1.0f)	.setSize(slideW,slideH)	<pre>.setPosition(slideX,slideDY*17)</pre>
.setValue(.4f);	.setRange(0,3)	<pre>.setSize(slideW, slideH)</pre>
	.setValue(0)	.setRange(-1,1)
cp5.addSlider("noise seed")	.setNumberOfTickMarks(11)	.setValue(0)
.setId(2)	.snapToTickMarks(true)	.setNumberOfTickMarks(100)
.moveTo(g2)	.showTickMarks(false);	.snapToTickMarks(true)
.setPosition(slideX,slideDY*2)	· Onowillowalko (lalbo)/	.showTickMarks(false);
	5 1101'1 (W 1 1/1)	.SHOWLICAMAIRS (laise);
.setSize(slideW,slideH)	<pre>cp5.addSlider("y scale")</pre>	
.setRange(0,50)	.setId(14)	<pre>cp5.addSlider("h trim")</pre>
.setValue(8);	.moveTo(g3)	.setId(22)
	<pre>.setPosition(slideX,slideDY*10)</pre>	.moveTo(g3)
//add group 3 controllers	.setSize(slideW,slideH)	.setPosition(slideX,slideDY*18)
	.setRange(-3,3)	.setSize(slideW,slideH)
cp5.addSlider("total radiation")	.setValue(1)	.setRange(0,3)
.setId(5)	.setNumberOfTickMarks(100)	.setValue(0)
.moveTo(g3)	.snapToTickMarks(false)	.setNumberOfTickMarks(11)
.setPosition(slideX,slideDY*1)	<pre>.showTickMarks(false);</pre>	.snapToTickMarks(true)
.setSize(slideW,slideH)		<pre>.showTickMarks(false);</pre>
.setRange(0,10)	<pre>cp5.addSlider("y offset")</pre>	
<pre>.setValue(0);</pre>	.setId(15)	//add group 4 controllers
	.moveTo(q3)	
<pre>cp5.addSlider("direct radiation")</pre>	.setPosition(slideX,slideDY*11)	cp5.addKnob("azimuth")
.setId(6)		-
	.setSize(slideW,slideH)	.setId(13)
.moveTo(g3)	.setRange(-1,1)	.moveTo(g4)
.setPosition(slideX,slideDY*2)	.setValue(.1f)	.setPosition(20,20)
.setSize(slideW,slideH)	.setNumberOfTickMarks(100)	.setRadius(35)
.setRange(0,10)	.snapToTickMarks(true)	.setAngleRange(2*PApplet.PI)
<pre>.setValue(0);</pre>	<pre>.showTickMarks(false);</pre>	<pre>.setStartAngle((float)(.5*PApplet.PI))</pre>
		.setRange(0,360)
<pre>cp5.addSlider("diffuse radiation")</pre>	<pre>cp5.addSlider("y trim")</pre>	.setValue(0)
.setId(7)	.setId(16)	.setNumberOfTickMarks(16)
.moveTo(g3)	.moveTo(g3)	.snapToTickMarks(true)
<pre>.setPosition(slideX,slideDY*3)</pre>	<pre>.setPosition(slideX,slideDY*12)</pre>	<pre>.setColorForeground(p.color(255))</pre>
.setSize(slideW,slideH)	.setSize(slideW,slideH)	.setColorBackground(p.color(40))
.setRange(0,10)	.setRange(0,3)	<pre>.setColorActive(p.color(255,255,0))</pre>
<pre>.setValue(0);</pre>	.setValue(0)	<pre>.setDragDirection(Knob.HORIZONTAL);</pre>
.55574245 (0) /	.setNumberOfTickMarks(11)	.SccDiagDirection(InnoD.noRiBoRing)
5 1101'1 (W) 1 (f)		5 13 10 1 10 1
cp5.addSlider("by grade")	.snapToTickMarks(true)	<pre>cp5.addKnob("elevation")</pre>
.setId(8)	<pre>.showTickMarks(false);</pre>	.setId(14)
.moveTo(g3)		.moveTo(g4)
.setPosition(slideX,slideDY*4)	<pre>cp5.addSlider("w scale")</pre>	.setPosition(110,20)
.setSize(slideW,slideH)	.setId(17)	.setRadius(35)
.setRange(0,10)	.moveTo(q3)	.setAngleRange(PApplet.PI)
<pre>.setValue(0);</pre>	.setPosition(slideX,slideDY*13)	<pre>.setStartAngle((float)(1.5*PApplet.PI))</pre>
.Seevarae(0)/		
	.setSize(slideW,slideH)	.setRange(1,179)
cp5.addSlider("by elevation")	.setRange(-3,3)	.setValue(0)
.setId(9)	.setValue(1)	.setNumberOfTickMarks(8)
.moveTo(g3)	.setNumberOfTickMarks(100)	.snapToTickMarks(true)
.setPosition(slideX,slideDY*5)	.snapToTickMarks(true)	<pre>.setColorForeground(p.color(255))</pre>
.setSize(slideW,slideH)	<pre>.showTickMarks(false);</pre>	.setColorBackground(p.color(40))
.setRange(0,10)		<pre>.setColorActive(p.color(255,255,0))</pre>
-	ann addelidar(Nr. offact//)	
<pre>.setValue(0);</pre>	<pre>cp5.addSlider("w offset")</pre>	<pre>.setDragDirection(Knob.HORIZONTAL);</pre>
	.setId(18)	
cp5.addSlider("noise")	.moveTo(g3)	
.setId(10)	<pre>.setPosition(slideX,slideDY*14)</pre>	//Add Sidebar Accordion: **********
.moveTo(g3)	.setSize(slideW,slideH)	******
.setPosition(slideX,slideDY*6)	.setRange(-1,1)	<pre>accordion = cp5.addAccordion("acc")</pre>
.setSize(slideW,slideH)	.setValue(0)	.setPosition(0,0)
.setRange(0,20)	.setNumberOfTickMarks(100)	.setWidth(200)
.setValue(0);	.snapToTickMarks(false)	.addItem(g1)
	.showTickMarks(false);	.addItem(g2)
cp5.addSlider("x scale")		.addItem(g3)
.setId(11)	<pre>cp5.addSlider("w trim")</pre>	.addItem(g4)
.moveTo(q3)	.setId(19)	į
.setPosition(slideX,slideDY*7)	.moveTo(q3)	
.setSize(slideW,slideH)	.setPosition(slideX,slideDY*15)	//initially open all groups
.setRange(-1.0f,1.0f)	.setSize(slideW,slideH)	accordion.open(0,1,2,3);
.setValue(1.0f)	.setRange(0,3)	
.setNumberOfTickMarks(21)	.setValue(0)	//allow multiple groups open at a time.
.snapToTickMarks(true)	.setNumberOfTickMarks(11)	accordion.setCollapseMode(Accordion.
<pre>.showTickMarks(false);</pre>	.snapToTickMarks(true)	MULTI);
	.showTickMarks(false);	
on5 addSlidor(\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	.0.00110.0.01100(10100)/	//٦٨٨
cp5.addSlider("x offset")		//Add keyboard shortcuts ***********
.setId(12)	<pre>cp5.addSlider("h scale")</pre>	*************
.moveTo(g3)	.setId(20)	<pre>cp5.mapKeyFor(new ControlKey() {public</pre>
.setPosition(slideX,slideDY*8)	.moveTo(g3)	<pre>void keyEvent() {accordion.open(0,1,2,3);}}, 'o');</pre>
.setSize(slideW,slideH)	<pre>.setPosition(slideX,slideDY*16)</pre>	cp5.mapKeyFor(new ControlKey() {public
.setRange(-1,1)	.setSize(slideW,slideH)	<pre>void keyEvent() {accordion.close(0,1,2,3);}}, 'c');</pre>
.setValue(.1f)	.setRange(-3,3)	
.setVarue(.11) .setNumberOfTickMarks(100)	- · · · · · · · · · · · · · · · · · · ·	1//ord :=:+/)
. OCCIMUMDETOTITCULATERS (TAA)	.setValue(1)	}//end init()

topic/proscene-and	on from https://forum.processing. -2d-drawing void saveState(Scene scene) {	of subcla org/ Panels are which defi space, an
	<pre>// Disable depth test to draw 2 p.hint(DISABLE_DEPTH_TEST);</pre>	about the
	// Set processing projection as	each pane nd model-
view matrices to d		Part of:
	// 1. projection matrix:	
plet.tan(PApplet.P	float cameraZ = ((p.height/2.0f) T*60 0f/360 0f)):	/ PAp- Aper[na]t: Author: Jo
piccidan (imppiccii	scene.pg3d.perspective(Date: 201
	PApplet.PI/	•
pectRatio(),	scene.camer	a().as- Inheritte
pecenacio(),	cameraZ/10.)f, ******
	cameraZ*10.	Of); package m
	// 2 model view matrix	import pr
	scene.pg3d.camera();	import mo
}		
//functi	on from https://forum.processing.	public clapray/
topic/proscene-and		
	// 1. Restore processing projec	tion ma-
trix	<pre>switch (scene.camera().type())</pre>	
	case PERSPECTIVE: scene.pg3d.perspective	e (
		cene.cam-
<pre>era().fieldOfView() era().aspectRatio(</pre>	S	cene.cam-
era().zNear(),		cene.cam-
014().211041(),	S	cene.cam-
era().zFar());	break;	
	case ORTHOGRAPHIC:	
	float[] wh = scene.came	era().
getOrthoWidthHeigh	t(); scene.pg3d.ortho(
		wh[0], data into
wh[0], -wh[1], wh[1],	
era().zNear(),	S	cene.cam-
era().2Near(),	S	cene.cam-
era().zFar());		
	break;	
	,	
	// 2. Restore processing modely	iew ma- ///////
trix	scene.pg3d.camera(
	scene.camer	a().
position().x,		- ()
position().y,	scene.camer	
position().z,	scene.camer.	
at().x,	scene.camer.	
at().y,	scene.camer.	
at().z,		
upVector().x,	scene.camer	
upVector().y,	scene.camer	
upVector().z);	scene.camer	-17.
	//Re-enble depth test	
	p.hint(ENABLE_DEPTH_TEST);	}
}		/****
}		Panel Cla
/*********	*********	
		Panel is

Hanger is data model defining panel system, composed

of panels.

Panels are basically two sets of parameters: one

```
ass skin, which is composed of panels.
                                                          which defines the panel's size and position in
                                                          space, and another that defines the size and
       re basically two sets of parameters: one
       fines the panel's size and position in
                                                          relative position of a single rectangular void
       nd another that defines the size and
                                                          in the panel. Panel also contains data from ecotect
       position of a single rectangular void
                                                           about the amount of direct and diffuse radiation
       anel. Panel also contains data from ecotect
                                                          each panel recieves.
        amount of direct and diffuse radiation
                                                          Part of
         recieves
                                                          Aper[na]tures
                                                          Author: Joshua Parker
                                                          Date: 2011
        Toshua Parker
                                                           Inheritted License: GPL, V3
       ed License: GPL, V3
                                                            model;
                                                           import processing.core.*;
       rocessing.core.PApplet;
                                                          public class Panel
       odel.Skin;
                                                                    //The parent PApplet
       lass Hanger{
                                                                   PApplet p;
       //The parent PApplet
                                                                   //panel properties
                                                                   public PVector[] coords;
       PApplet p;
                                                                   public float x,y,w,h;
                                                          //normalized
       //composit skin
                                                                  public float px, py, pd;
        public Skin myskin;
                                                                                                       //center-
                                                          line of frame
        //CONSTRUCTOR
                                                                   public float t;
        public Hanger(PApplet parent,
                                                           //thickness of frame
                                                                   public float ix, iy, id;
                                                                                                       //inside
                                                           line, ie. panel
                         float D,
                                                                   public float hx, hy, hw, hh; //perforations, ie.
                         float T,
                         float snap,
                                                                   public float snap;
                         String PATH) {
                                                           //panel subdivisions
                                                                   public int pClr;
                                                           //panel color
                p = parent;
                                                                    //imported data
                 //create skin
                 myskin = new Skin(p, U, V, D, T, snap);
                                                                   private float avgDailyTotal;
                                                                   private float avgDailyDirect;
                                                                    private float avgDailyDiffuse;
                 //pass data path to skin so it can load
                                                                    private float grade;
                 myskin.loadData(PATH);
                                                                    private float elevation; */
                 //update skin pattern
                                                                    //normalized parameters
                 update();
                                                                    private float totalNorm;
                                                                    private float directNorm;
       }//end constructor
                                                                    private float diffuseNorm;
                                                                    private float gradeNorm;
        private float elevationNorm;
        public float noiseVal;
        //set noise increment and update model
                                                                    //parameter weights
        public void setInc(float Inc) {
                                                                    private float[] xcf;
              myskin.setInc(Inc);
                                                                    //constructor
                                                                    Panel (PApplet parent,
       //set noise seed and update model
                                                                                    float px,
                                                                                     float py,
       public void setSeed(int Seed) {
                                                                                     float pd,
                mvskin.setSeed(Seed);
                                                                                     float ix,
                                                                                     float iy,
       //set pattern coeffecients and update model
                                                                                     float id,
        public void setXcf(int i, float C) {
                                                                                     float t,
                myskin.setXcf(i, C);
                                                                                     float snap) {
                                                                            p = parent;
       //update Hanger Model
        public void update(){
                                                                            //println("new panel");
                 //update skin pattern
                 myskin.update();
                                                                            this.px = px;
                                                                            this.py = py;
                                                                            this.pd = pd;
                                                                            this.ix = ix;
        ***********
                                                                            this.iy = iy;
                                                                            this.id = id;
                                                                            this.t = t;
                                                                            //hx = this.hx;
       subclass of Hanger data model defining panel
                                                                            //hy = this.hy;
                                                                            //hw = this.hw;
system, composed of subclass skin, which is composed
```

//hh = this.hh;

this.snap = snap;

			/(xcf[0] +xcf[1]	Skin Class	
	<pre>coords = new PVector[4];</pre>	+xcf[2] +xcf[3] +	xcf[4] +xcf[5] +1);		
	<pre>xcf = new float[18];</pre>		//calc normalized scale effects and off-		er data model defining panel ass skin, which is composed
}		set x, y, w, z		of panels.	
			//VALUE RANGE: 0.0 - 1.0	Panels are basically two	
// PU //////////////////////////////////	JBLIC METHODS ////////////////////////////////////	cale) + .5) + wOf	float wv = (float)((((pattern5) * wS-	which defines the panel's space, and another that d	=
,,,,,,,,,,,,,,,	,,,,,	cale) + .5) + WOI	float hv = (float)((((pattern5) * hS-	relative position of a si	
//loa	d data into panel	cale) + .5) + hOf			contains data from ecotect
publi	c void loadData(String avgDailyTotal,		float xv = (float)((((pattern-1) * xScale)	about the amount of direc	ct and diffuse radiation
	String avgDailyDirect,	+ 0) + xOff);		each panel recieves.	
	String avgDailyDiffuse,		<pre>float yv = (float)((((pattern-1) * yScale)</pre>		
	String grade,	+ 0) + yOff);		Part of:	
	String elevation, String totalNorm,		//calculate real hole dimensions and	Aper[na]tures	
	String directNorm,	displacement in m		Author: Joshua Parker	
	String diffuseNorm,	<u>1</u>	hw = wv * id;	Date: 2011	
	String gradeNorm,		hh = hv * id;		
	String elevationNorm){		hx = xv * id + t;	Inheritted License: GPL,	V3
			hy = yv * id + t;		********
* arseFloat(avgD	this.avgDailyTotal = Float.		//snap and normalize coords to subgrid	package model;	*******
arserioac (avgb	this.avgDailyDirect = Float.		<pre>w = (int) (PApplet.round(hw/snap));</pre>	package model,	
arseFloat(avgD			<pre>h = (int) (PApplet.round(hh/snap));</pre>	import processing.core.PA	Applet;
	this.avgDailyDiffuse = Float.		<pre>x = (int)(PApplet.round(hx/snap));</pre>	<pre>import model.Panel;</pre>	
arseFloat(avgD	<pre>DailyDiffuse);</pre>		<pre>y = (int)(PApplet.round(hy/snap));</pre>		
	<pre>this.grade = Float.parseFloat(grade);</pre>			public class Skin	
=1 . / 1	this.elevation = Float.		//convert back to absolute coords	{	
arseFloat(elev	ration);*/		<pre>hw = w*snap; hh = h*snap;</pre>	//The parent PA PApplet p;	applet
	this.totalNorm = Float.		hx = px + x*snap;	rappiet p,	
arseFloat(tota			hy = py + y*snap;	//registration	pt
	<pre>this.directNorm = Float.</pre>			<pre>public float x,</pre>	у;
arseFloat(dire	ectNorm);		//load coordinate array (relative to		
	<pre>this.diffuseNorm = Float.</pre>	basegrid)		//composit pane	
arseFloat(diff			//adjust order for vertice mapping	<pre>public Panel[]</pre>	panels;
arseFloat(grad	this.gradeNorm = Float.		<pre>coords[3] = new PVector(x,y); coords[2] = new PVector(x+hw, y);</pre>	//skin properti	0.5
a10011000 (9100	this.elevationNorm = Float.		coords[1] = new PVector(x+hw, y+hh);	int u;	//number of modules
arseFloat(elev	rationNorm);		<pre>coords[0] = new PVector(x, y+hh);</pre>	in U direction	
				int v;	//number of modules
}//en	d loadData()			in V direction	
		}//end	perforate	float dim;	//panel lengh/width
//	c void setXcf(int i, float C){	//		<pre>dimension, D (meters) float thick;</pre>	//thickness, T of frame (me-
publi	xcf[i]=C;		void colorize(){	ters)	//tilickless, 1 of frame (me
}		•			
			//calc smth like a weighted avg of pat-	//noise vars	
//find		tern components		float inc;	
publi	c void perforate(){		float pattern = (totalNorm * xcf[0] +	int seed;	
	//name coefficients	+	directNorm * xcf[1]	//CONSTRUCTOR	
	<pre>float totalWeight = xcf[0];</pre>	•	diffuseNorm * xcf[2]	Skin(PApplet pa	arent,
	<pre>float directWeight = xcf[1];</pre>	+			int U,
	<pre>float diffuseWeight = xcf[2];</pre>		<pre>gradeNorm * xcf[3]</pre>		int V,
	<pre>float gradeWeight = xcf[3];</pre>	+			float D,
	<pre>float elevationWeight = xcf[4];</pre>		elevationNorm *		float T,
	<pre>float noiseWeight = xcf[5]; float xScale = xcf[6];</pre>	xcf[4] +	noiseVal * xcf[5] +		float snap) {
	float xOff = xcf[7];		1)/	p = p	parent;
	<pre>//float xTrim = xcf[8];</pre>		(xcf[0] +xcf[1]	r r	
	<pre>float yScale = xcf[9];</pre>	+xcf[2] +xcf[3] +	xcf[4] +xcf[5] +1);	//noi	se vars
	<pre>float yOff = xcf[10];</pre>				0.4f;
	//float yTrim = xcf[11];		int c1 = p.color(50, 50, 50);	seed	= 8;
	<pre>float wScale = xcf[12]; float wOff = xcf[13];</pre>		<pre>int c2 = p.color(200, 255, 200); float v = pattern * 255;</pre>	thic	u = U;
	//float wTrim = xcf[14];		moat v = pattern ~ 255;		v = V;
	float hScale = xcf[15];		<pre>float deltaR = p.red(c2)-p.red(c1);</pre>		dim = D;
	<pre>float hOff = xcf[16];</pre>		<pre>float deltaG = p.green(c2)-p.green(c1);</pre>	this.	thick = T;
	<pre>//float hTrim = xcf[17];</pre>		<pre>float deltaB = p.blue(c2)-p.blue(c1);</pre>		
					tialize 1d array for u x v panels
	//calc smth like a weighted avg of pat-		int c = p.color(panel	s = new Panel[u*v];
ern components	float pattern = (totalNorm * totalWeight	(p.red(c1)+(v)*(d	eltaR/255)),	//oro	eate u x v panels in 1d array
	(Section Cocarnery in	,a(c., / (v) '(u			n = 0;
	directNorm * direct-	(p.green(c1)+(v)*	(deltaG/255)),		nt i=0; i <v; i++)="" td="" {<=""></v;>
eight +					for(int j=0; j <u; j++)="" td="" {<=""></u;>
	diffuseNorm * dif-	(p.blue(c1)+(v)*(deltaB/255))		
useWeight +);		//calc panel dimen-
eight +	gradeNorm * grade-		nclr = c.	sion and reg pts	floot vd - D
cigne T	elevationNorm * el-	1	pClr = c;		float pd = D; float px = $j*D$;
vationWeight +		1			float py = i*D;
	noiseVal * noise-	}//end class			- - -
eight +		/********	********		//calc inner frame
	1)			dimension and reg pts	

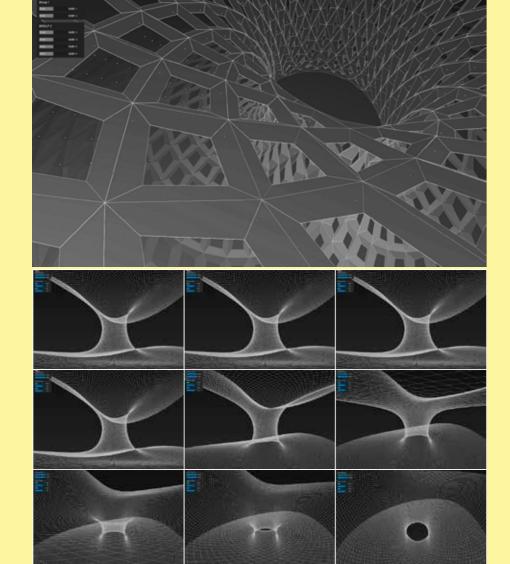
```
float id = pd-(T*2);
                                                                                                                           ize();
                                    float ix = px+T;
                                    float iy = py+T;
                                     //frame thickness
                                                                      }//end loadData()
                                                                                                                                    }//end perforate
                                    float thick = T;
                                                                       //set a coefficient
                                                                                                                            }//end class
                                                                                                                            /***************
                                     //create panel
                                                                       public void setXcf(int id, float C) {
                                     panels[n] = new
                                                                               int n = 0;
Panel(
                                                                                for(int i=0; i<v; i++) {
                                                                                                                            View Class
                                                                                         for(int j=0; j<u; j++) {
                                                                                                  panels[n].setXcf(id,
                                                                                                                           View - contains the scene(empty space and camera),
                                                             C);
                                                                                                                              imported 3d base model, and data model defining
                                                                                                                             panel system. View calls draws to screen via
                                                                                                                              viewState interface, though only one viewState
                                                                                                                             is implemented: View3d, a basic 3d view.
                                                                       //set noise increment
                                                                       public void setInc(float Inc) {
                                                                                                                            Aper[na]tures
                                                                                                                            Author: Joshua Parker
                                                                                inc = Inc;
                                                                                                                            Date: 2011
                                                                                                                            Inheritted License: GPL, V3
id,
                                                                       //set noise seed
                                                                       public void setSeed(int Seed){
                                                                               seed = Seed;
                                                                                                                            ***************
thick,
                                                                                                                            package view;
snap):
                                                                       //update pattern
                                                                                                                            import processing.core.PApplet;
                                     //incr counter
                                                                       public void update(){
                                                                                                                            import remixlab.proscene.*;
                                     n++;
                                                                                //generate noise pattern
                                                                                                                            import saito.objloader.*;
                                                                                generateNoise();
                                                                                                                            import model.Hanger;
                                                                                 //assign colors to panels
                                                                                                                            import view.View3d;
                                                                                colorize();
                                                                                                                            import view.ViewState;
                   //generate initial pattern
                                                                                //generate perforation from ecotect data
                                                                                                                            public class View{
                                                                                                                                     //parent PApplet
                                                                                                                                     PApplet p;
         //generate noise
public void generateNoise(){
                                                                                                                                     public Scene scene;
                                                                                                                                     public OBJModel model;
         //load data
                                                                                //noise constants
         public void loadData(String PATH) {
                                                                                float vx = 0.0f;
                                                                                                                                     public Hanger myhanger;
                                                                                float vy = 0.0f;
                  //holders for rows and items in a row
                                                                                //float inc = 0.4;
                                                                                                                                     //OBJModel flags - not used
                                                                                                                                     public boolean bTexture = true;
respectively
                                                                                //float k = 0;
                                                                                                                                     public boolean bStroke = false;
                  String loadedStrings[];
                                                                                p.noiseSeed(seed);
                  String splitString[];
                                                                                                                                     public boolean bMaterial = true;
                                                                                int n = 0;
                  //load data into an array of rows
                                                                                for(int i=0; i<v; i++) {
                                                                                                                                     //states
                   loadedStrings = p.loadStrings(PATH);
                                                                                         for(int j=0; j<u; j++) {
                                                                                                                                     //private ViewState myView3d;
                                                                                                                                     //private ViewState myViewUnroll;
                  int n = 0;
                                                                                                  //calc noise
                  for(int i=0; i<v; i++) {
                                                                                                   panels[n].noiseVal =
                                                                                                                                     public ViewState viewState;
                                                              p.noise(vx, vy);
                            //get data for first row, i+1
                                                                                                                                     public View(PApplet parent, Hanger Myhanger) {
b/c first row of text file
                                                                                                   //increment panel
                            //is column headers
                                                                                                  n++;
                                                                                                                                             p = parent;
                           splitString = PApplet.
split(loadedStrings[i+1], '\t');
                                                                                                  //increment noise x
                                                                                                                                              this.scene = new Scene(parent);
                           for(int j=0; j<u; j++) {
                                                                                                  vx = vx + inc;
                                                                                                                                              this.scene.setAxisIsDrawn(false);
                                                                                                                                              this.scene.setGridIsDrawn(false);
                                     //load up panel
                                                                                                                                              this.scene.enableFrustumEquationsUp-
                                                                                         //increment noise v param
                                     panels[n].loadData(
                                                                                         vv = vv + inc;
                                                                                                                           date();
                                                                                                                                             //imported 3d model; no need to use mtls
splitString[2],
                                                                       }//end generateNoise()
                                                                                                                           or specify shapemode b/c
splitString[5],
                                                                       //find holes
                                                                                                                                             //not drawing model, but just using the
                                                                       public void perforate(){
                                                                                                                            face vertices and normals
                                                                                                                                             this.model = new OBJModel(p, "bs10.
splitString[8],
                                                                               int n = 0;
                                                                                for(int i=0; i<v; i++) {
                                                                                                                            obj");
splitString[10],
                                                                                         for(int j=0; j<u; j++) {
                                                                                                                                             //this.model = new OBJModel(p, "bs10.
                                                                                                  panels[n].perfo-
                                                                                                                            obj", "relative", LINES);
splitString[12],
                                                                                                                                             //this.model.shapeMode(LINES);
                                                                                                                                              //this.model.disableMaterial();
splitString[3],
                                                                                                                                              //this.model.enableDebug();
                                                                                                                                              this.model.scale(1);
splitString[6],
                                                                      }//end perforate
                                                                                                                                              this.myhanger = Myhanger;
                                                                       //find holes
                                                                                                                                              //view state
splitString[9],
                                                                       public void colorize(){
                                                                                                                                              setViewState(new View3d(p, this));
                                                                                int n = 0;
splitString[11],
                                                                                for(int i=0; i<v; i++) {
splitString[13]);
                                                                                         for(int j=0; j<u; j++) {
```

panels[n].color-

///////////////////////////////////////	-1);	p.direction	onalLight(126, 126, 126, 0, 0,		p.beginShape();
//set view state	-1);	//p.ambie	ntLight(102, 102, 102);	vs[2].z);	p.vertex(vs[2].x, vs[2].y,
<pre>public void setViewState(ViewState newState){</pre>		p.lights()		vs[3].z);	<pre>p.vertex(vs[3].x, vs[3].y,</pre>
3			era position and target	pts[3].z);	<pre>p.vertex(pts[3].x, pts[3].y,</pre>
//get current fill	/*		= new PVector(0,0,-1);	pes[5].2//	p.vertex(pts[2].x, pts[2].y,
<pre>public boolean getFill()(</pre>			<pre>2 = new PVector(0,0,0); ene.camera().</pre>	pts[2].z);	<pre>p.endShape();</pre>
}	setUpVector(v,true)		ene.camera().lookAt(v2);*/		<pre>p.beginShape();</pre>
//turn off strokes, turn on surface fill		_			p.vertex(vs[3].x, vs[3].y,
<pre>public void enableFill() {</pre>	3d model!!! this wi	_	nt vertices so do not draw the etch way down	vs[3].z);	p.vertex(vs[0].x, vs[0].y,
}		//myview.r	model.draw();	vs[0].z);	<pre>p.vertex(pts[0].x, pts[0].y,</pre>
<pre>//turn off surface fill, turn on strokes public void disableFill() {</pre>			ces in array egment = myview.model.getSeg-	pts[0].z);	<pre>p.vertex(pts[3].x, pts[3].y,</pre>
<pre>this.viewState.disableFill();</pre>	ment(0);			pts[3].z);	
}		Face[] fac	<pre>ces = segment.getFaces();</pre>		<pre>p.endShape();</pre>
<pre>//draw the view public void draw() {</pre>		//for each	h face i = faces.length-1; i >= 0;		}//end for loop
this.viewState.draw();	i) {	TOT (INC.	1 = laces.length-1; 1 >= 0;	}//end 0	draw
}					
) /********	panel		//get face and corresponding	// PRIV	ATE METHODS ////////////////////////////////////
View3d Class			<pre>Face nextFace = faces[i]; Panel nextPanel = myview.my-</pre>	//find h	ole vertices on face from a face
View3d implements a basic 3d view	hanger.myskin.panel	ls[i];		vertices(vs),	normals(ns), and face-space coordinates(cs)
			//get vertices and normals of	private	<pre>PVector[] mapFace(PVector[] vs, PVector[]</pre>
Part of:	face and relative o		//of panel opening	n, PVector[] cs){	
Aper[na]tures Author: Joshua Parker	<pre>Vertices();</pre>		<pre>PVector[] vs = nextFace.get-</pre>		<pre>PVector[] pts = new PVector[0];</pre>
Date: 2011			<pre>PVector[] ns = nextFace.get-</pre>		//for each corner, map 2d coord to 3d pt
Inheritted License: GPL, V3	Normals();		<pre>PVector[] cs = nextPanel.co-</pre>		<pre>for (int i = vs.length-1; i >= 0; i) { PVector pt = mapPt(vs[0],</pre>
***********/	ords;			n[i], cs[i]);	<pre>pts = (PVector[])PApplet.</pre>
package view;			<pre>//get panel color int clr = nextPanel.pClr;</pre>	append(pts, pt);	1
<pre>import processing.core.*;</pre>					return pts;
<pre>import saito.objloader.*; import model.*;</pre>	to world space coor	rdinates.	//map panel space coordinates	}	
	co morra opace coor		PVector[] pts = mapFace(vs,	j	
<pre>public class View3d implements ViewState{</pre>	ns, cs);				<pre>inge 3d pt from a vertice(v), normal(n), d relative coordinate(c)</pre>
//The parent PApplet			//stroke normal		PVector mapPt(PVector v, PVector n, PVec-
PApplet p;	ve[0] z ve[0] v+n:		//line(vs[0].x, vs[0].y, 0].y+ns[0].y, vs[0].z+ns[0].z);	tor c){	
//composit view	V3[0].2, V3[0].X1N3	J[0].A, VJ[0	0].y.no[0].y, vo[0].Z.no[0].Z,		//get angle of normal
<pre>public View myview;</pre>			//set fill and stroke		<pre>float a = (float) Math.atan(n.z/n.y);</pre>
			if (bFill) {	aha (Bānnlat dagra	a = PApplet.radians(90-PApplet.
<pre>//flags boolean bFill = false;</pre>			<pre>if(bFill){ p.fill(clr); p.noStroke();</pre>	abs(PApplet.degree	
<pre>boolean bFill = false;</pre>			<pre>p.fill(clr); p.noStroke(); }else{</pre>	abs(PApplet.degre	
			<pre>p.fill(clr); p.noStroke();</pre>	abs(PApplet.degre	es(a)));
<pre>boolean bFill = false; //constructor</pre>	10);		<pre>p.fill(clr); p.noStroke();)else{ p.noFill();</pre>	abs(PApplet.degre	es(a))); float x,y,z;
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	10);		<pre>p.fill(clr); p.noStroke();)else{ p.noFill();</pre>	abs (PApplet.degre	es(a))); float x,y,z; if(n.y < 0){
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	10);		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons</pre>	abs (PApplet.degre	<pre>float x,y,z; if (n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview){</pre>	10);		<pre>p.fill(clr); p.noStroke(); }else{</pre>	abs (PApplet.degre	<pre>es(a))); float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	10); vs[0].z);		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons</pre>	abs(PApplet.degre	<pre>float x,y,z; if (n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview){</pre>			<pre>p.fill(clr); p.noStroke(); }else{</pre>	abs(PApplet.degre	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview){</pre>	vs[0].z);		<pre>p.fill(clr); p.noStroke(); }else{</pre>	abs (PApplet.degre	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	vs[0].z); vs[1].z); pts[1].z);		<pre>p.fill(clr); p.noStroke(); }else{</pre>	abs(PApplet.degre	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	vs[0].z); vs[1].z);		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons p.beginShape(); p.vertex(vs[0].x, vs[0].y, p.vertex(vs[1].x, vs[1].y, p.vertex(pts[1].x, pts[1].y,</pre>	abs(PApplet.degre	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	vs[0].z); vs[1].z); pts[1].z);		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons p.beginShape(); p.vertex(vs[0].x, vs[0].y, p.vertex(vs[1].x, vs[1].y, p.vertex(pts[1].x, pts[1].y, p.vertex(pts[0].x, pts[0].y,</pre>	}	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	<pre>vs[0].z); vs[1].z); pts[1].z); pts[0].z);</pre>		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons p.beginShape(); p.vertex(vs[0].x, vs[0].y, p.vertex(pts[1].x, vs[1].y, p.vertex(pts[1].x, pts[1].y, p.vertex(pts[0].x, pts[0].y, p.vertex(pts[0].x, pts[0].y,</pre>	} /*******	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	vs[0].z); vs[1].z); pts[1].z);		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons p.beginShape(); p.vertex(vs[0].x, vs[0].y, p.vertex(pts[1].x, pts[1].y, p.vertex(pts[0].x, pts[0].y, p.vertex(pts[0].x, pts[0].y, p.beginShape();</pre>	}	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	<pre>vs[0].z); vs[1].z); pts[1].z); pts[0].z);</pre>		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons p.beginShape(); p.vertex(vs[0].x, vs[0].y, p.vertex(pts[1].x, vs[1].y, p.vertex(pts[0].x, pts[0].y, p.vertex(pts[0].x, vs[0].y, p.vertex(pts[0].x, vs[0].y, p.vertex(pts[0].x, vs[0].y, p.vertex(pts[0].x, vs[0].y, p.endShape(); p.beginShape(); p.vertex(vs[1].x, vs[1].y,</pre>	} /************************************	<pre>float x,y,z; if(n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview){</pre>	<pre>vs[0].z); vs[1].z); pts[1].z); pts[0].z);</pre>		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons p.beginShape(); p.vertex(vs[0].x, vs[0].y, p.vertex(pts[1].x, pts[1].y, p.vertex(pts[0].x, pts[0].y, p.endShape(); p.beginShape(); p.beginShape(); p.vertex(vs[1].x, vs[1].y, p.vertex(vs[1].x, vs[1].y, p.vertex(vs[1].x, vs[1].y, p.vertex(vs[1].x, vs[1].y, p.vertex(vs[2].x, vs[2].y, p.vertex(pts[2].x, pts[2].y,</pre>	} /************************************	<pre>float x,y,z; if (n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview) {</pre>	<pre>vs[0].z); vs[1].z); pts[1].z); pts[0].z); vs[1].z);</pre>		<pre>p.fill(clr); p.noStroke(); }else{</pre>	} /************* ViewState Class ViewState is an in	<pre>float x,y,z; if (n.y < 0) {</pre>
<pre>boolean bFill = false; //CONSTRUCTOR View3d(PApplet parent, View Myview){</pre>	<pre>vs[0].z); vs[1].z); pts[1].z); pts[0].z); vs[1].z); vs[2].z);</pre>		<pre>p.fill(clr); p.noStroke(); }else{ p.noFill(); p.stroke(10, 10, } //draw four quad polygons p.beginShape(); p.vertex(vs[0].x, vs[0].y, p.vertex(pts[1].x, pts[1].y, p.vertex(pts[0].x, pts[0].y, p.endShape(); p.beginShape(); p.beginShape(); p.vertex(vs[1].x, vs[1].y, p.vertex(vs[1].x, vs[1].y, p.vertex(vs[1].x, vs[1].y, p.vertex(vs[1].x, vs[1].y, p.vertex(vs[2].x, vs[2].y, p.vertex(pts[2].x, pts[2].y,</pre>	} /******* ViewState Class ViewState is an in	<pre>float x,y,z; if (n.y < 0) {</pre>

ZHUHAI PAVILION, SOURCE CODE

A CUSTOM PANELING TOOL WAS BUILT WITH JAVA/PROCESSING USING OPEN SOURCE GEOMETRY LIBRARY (IGEO, BY SATURO SUGIHARA) TO INTEROGATE FORM AND EXPLORE THE POSSIBILITY OF PERFORATING THE SHELL IN PLACES TO CREATE VISUAL AND PHYSICAL ACCESS THROUGH VERTICAL SURFACES. TOOL DEFINES A UV DIAGRAD AND MAPS PERFORATION PATTERNS AN INPUT SURFACE USING SIMPLE ALGORITHMS AND EXPOSES DESIGN PARAMETERS. THIS ALLOWED FOR REAL-TIME COLLABORATIVE USE BY ARCHITECTS, ENGINEERS, AND CLIENT.



gradient paneling condition fedback to surface.

continuous surface subdivision.

```
/************
DiagridPaneler
                                                                              //send ui events to listener, so it can route them to control-
                                                                       ler
Author: Joshua Parker
                                                                              public void controlEvent (ControlEvent theEvent) {myListener.
Date: 2011
                                                                       controlEvent(theEvent); }
                                                                              public void mousePressed() {myListener.mousePressed(mouseX,
Uses the following processing libraries:
                                                                       mouseY);}
                                                                              public void keyPressed() {myListener.keyPressed(key);}
controlP5 - gui toolkit
- http://www.sojamo.de/libraries/controlP5/
Proscene - 3d scene library
- http://code.google.com/p/proscene/
Objloader - 3d object loader
- http://code.google.com/p/saitoobjloader/
Inheritted License: GPL, V3
                                                                       /**************
*****************
                                                                       Controller Class
import igeo.*;
                                                                       MVC Controller contains methods for updating data
import processing.core.*;
import controlP5.*;
                                                                       model & controlling view(s) *
import processing.opengl.*;
import controller.Listener;
                                                                       Part of:
import controller.Controller;
import migeo.MyIG;
                                                                       Aper[na]tures
import migeo.MyPIGraphicsGL;
                                                                       Author: Joshua Parker
import model.Box;
                                                                       Date: 2011
import model.Model;
import view.*;
                                                                       Inheritted License: GPL, V3
import gui.*;
                                                                       *************************************
                                                                       package controller;
public class DiagridPaneler extends PApplet{
       private static final long serialVersionUID = 1L;
                                                                       import igeo.IG;
                                                                       import processing.core.PApplet;
       //path to tabular data from ecotect analyses
                                                                       import processing.core.PVector;
       String PATH = "data/ecotectData.txt";
                                                                       //import model.Box;
                                                                       import model.Model;
       //objects
                                                                       import view. View;
       Listener myListener;
                                                                       import qui.Gui;
       Controller mvController;
                                                                       public class Controller{
       Model mymodel;
       View myview;
       Gui mygui;
                                                                              //The parent PApplet
                                                                              PApplet p;
       IG myig;
                                                                              //PROPS
       public void setup(){
                                                                              private Model mymodel;
        size(1920, 1200, "migeo.MyPIGraphicsGL");
                                                                              private View myview;
        //IG.bg(50, 40, 70, 90);
                                                                              private Gui mygui;
        IG.bg(255, 255, 255, 255);
                                                                              //CONSTRUCTOR
         //MODEL: Parametric Data for panel system
                                                                              public Controller (PApplet parent, Model Mymodel, Gui Mygui,
        //mybox = new Box(this);
                                                                       View Myview) {
        mymodel = new Model(this);
                                                                                    p = parent;
                                                                                  mymodel = Mymodel;
         //VIEW: draws a representation of model to screen
                                                                                 mygui = Mygui;
         //NOTE: This is a representation of the data model ONLY!
                                                                                  myview = Myview;
        // It create shapes and geometry and igeo renders to
panel panes
        myview = new View(this, mymodel);
                                                                              ///////
         //GUI: needs to access view scene for drawing 2d qui on top
of 3d scene
        //TBD: This should wrap all iu elements, ie. igeo panel,
                                                                              public void setUCount(int n){
sidebar, etc.
         mygui = new Gui(this /*, IG.current().panel*/);
                                                                                     mymodel.unum = n;
         //CONTROLLER: updates model & views in response to ui events
         myController = new Controller(this, mymodel, mygui, myview);
                                                                              public void setVCount(int n) {
                                                                                     mymodel.vnum = n;
         //LISTENER/ROUTER: recieves ui events, ids and routes them to
        myListener = new Listener(this, myController);
                                                                              public void setPatternUStart(int v) {
       public void draw() {
                                                                                     mymodel.uMin = (int) (v*.01*mymodel.unum);
        //update model, view and gui via controller
        myController.update();
       //required to run as application
                                                                              public void setPatternUStop(int v) {
       public static void main(String args[]) {
       PApplet.main(new String[] { "--present", "DiagridPaneler" });
                                                                                     mymodel.uMax = (int) (v*.01*mymodel.unum);
```

```
public void hideFill() {
                                                                                     //myview.disableFill();
       public void setPatternVStart(int v) {
              mymodel.vMin = (int) (v*.01*mymodel.vnum);
                                                                             public void topView(){
                                                                                     //IG.topView();
                                                                                     IG.current().panel.currentPane().getView().setTop();
                                                                                     //IG.view(0).setTop();
       public void setPatternVStop(int v) {
              mymodel.vMax = (int) (v*.01*mymodel.vnum);
                                                                              public void perspectiveView(){
                                                                                     //IG.perspectiveView();
                                                                                     IG.current().panel.currentPane().getView().setAxonomet-
       ///////
                                                                             public void gotoView1(){
                                                                                     myview.setView1();
      public void focusGui(){
              //myview.scene.disableMouseHandling();
              //myview.scene.disableKeyboardHandling();
                                                                              public void gotoView2(){
                                                                                     myview.setView2();
      public void focusScene(){
              //myview.scene.enableMouseHandling();
                                                                              public void update() {
              //{\tt myview.scene.enableKeyboardHandling();}
                                                                                     //{\tt mybox.update();} //update the model
                                                                                     mymodel.update();
                                                                                     myview.draw(); // draw the 3d scene
       public void toggleTexture(){
                                                                                     mygui.draw(); // then draw the gui on top
       /*if(!myview.bTexture){
              showTexture();
                                                                       /**************
              hideTexture();
                                                                       Listener Class
                                                                       Listener recieves ui events, parses them and routes
                                                                       them to controller for handling.
       public void toggleMaterial(){
              if(!myview.bMaterial) {
                                                                       Part of:
//
                     showMaterial();
                                                                       Aper[na]tures
              } else
                     hideMaterial();
                                                                       Author: Joshua Parker
//
                                                                       Date: 2011
                                                                       Inheritted License: GPL, V3
                                                                       ************************************
       public void toggleFill(){
              /*if(!myview.getFill()) {
                                                                       package controller;
                     showFill();
              } else
                                                                       import igeo.IG;
                     hideFill();
                                                                       import controlP5.ControlEvent;
                                                                       import controlP5.ControlListener;
                                                                       import processing.core.PApplet;
                                                                       public class Listener implements ControlListener{
      public void showTexture(){
              //myview.model.enableTexture();
                                                                              //The parent PApplet
              //myview.bTexture = true;
                                                                              PApplet p;
                                                                              //PROPS
                                                                              Controller myController;
      public void hideTexture(){
              //myview.model.disableTexture();
                                                                              //CONSTRUCTOR
              //myview.bTexture = false;
                                                                              public Listener(PApplet parent, Controller MyController){
                                                                                     myController = MyController;
       public void showMaterial(){
                                                                              //myview.model.enableMaterial();
              //myview.bMaterial = true;
                                                                       //////
                                                                              //route slider events to controller via private handlers
                                                                              public void controlEvent(ControlEvent theEvent) {
       public void hideMaterial(){
              //myview.model.disableMaterial();
                                                                                     switch(theEvent.getController().getParent().getId()){
              //myview.bMaterial = false;
                                                                                     case(0):
                                                                                            //route g1 (shortcut info) event!
                                                                                            glControlEvent(theEvent);
                                                                                            break;
      public void showFill(){
                                                                                     case(1):
              //myview.enableFill();
                                                                                            //route g2 (noise sliders) event!
                                                                                            g2ControlEvent(theEvent);
                                                                                     break:
```

```
myController.setPatternVStart((int)(v));
                                                                                               break;
                                                                                               case(7):
                                                                                                       v = theEvent.getController().getValue();
       //route mouse events to controller
       //this should be replaced with focus on hover
                                                                                                       myController.setPatternVStop((int)(v));
       public void mousePressed(int mx, int my) {
                                                                                               break;
              if (mx < 200) {
                      myController.focusGui();
               } else {
                      myController.focusScene();
                                                                         package gui;
       //route key press events to controller
       public void keyPressed(char key) {
                                                                         import igeo.IG;
                                                                         import igeo.IVec;
               if(key == 't') {
                                                                         import igeo.gui.IGraphicMode;
                      //toggle the texture listed in .mtl file
                                                                         import igeo.gui.IGraphicMode.GraphicType;
                      myController.toggleTexture();
                                                                         import processing.core.PApplet;
                                                                         import processing.core.PFont;
                                                                         import controlP5.ControlP5;
              else if(kev == 'w') {
                      //toggle the material listed in .mtl file
                                                                         import controlP5.Knob;
                      //myController.toggleMaterial();
                                                                         public class ControlPanel {
                      //currentMousePane.getView().mode().
setDrawMode(true, false, false);
                      IG.current().panel.currentPane().getView().
                                                                                 //parent PApplet
mode().setDrawMode(true, false, false);
                                                                                 PApplet p;
               else if(key == '0') {
                                                                                 private ControlP5 cp5;
                      //toggle the fill for new shapes
                      System.exit(0);
                                                                                 //
                                                                                 controlP5.Group[] groups;
               else if(key == '1') {
                      //toggle the fill for new shapes
                                                                                 //count elements
                      myController.gotoView1();
                                                                                 //int NumOfTabs;
                                                                                 int numOfAccordings;
              else if(key == '2') {
                                                                                 int numOfGroups;
                      //toggle the fill for new shapes
                                                                                 int numOfElements;
                      myController.gotoView2();
                                                                                 //current Elements
              else if(kev == 'a') {
                                                                                 //String curTab;
                      //toggle the fill for new shapes
                                                                                 //String curAccording;
                      myController.topView();
                                                                                 //String curGroup;
                                                                                //String curElement;
              else if(key == 's') {
                      //toggle the fill for new shapes
                                                                                 int x;
                      myController.perspectiveView();
                                                                                int y;
                                                                                int w;
                                                                                 int h;
       int[] ppt = {0,0};
//////
                                                                                 int[] gpt = {0,0};
                                                                                 int[] ept = {0,0};
       //parse group 2 (noise sliders) event!
                                                                                 int dx;
       private void g1ControlEvent(ControlEvent theEvent) {
                                                                                 int dy;
              float v;
                                                                                 //style
               switch(theEvent.getController().getId()){
                                                                                 //int CPMargin;
                                                                                 int marginTop = 40;
              case(1):
                                                                                 int marginLeft = 20;
                      v = theEvent.getController().getValue();
                                                                                 int marginBottom = 10;
                      myController.setUCount((int)(v));
                                                                                 int groupWidth = 200;
              break;
                                                                                 //int groupHeight;
              case(2):
                      v = theEvent.getController().getValue();
                                                                                 //int accordingMargin;
                      myController.setVCount((int)(v));
                                                                                 //int groupPadding;
                                                                                 int headerPaddingTop = 10;
              break;
                                                                                 int headerPaddingLeft = 5;
                                                                                 int elementPaddingTop = 5;
                                                                                 int elementPaddingLeft = 10;
       //parse group 2 (noise sliders) event!
                                                                                 int elementWidth = 120;
                                                                                 int elementHeight = 15;
              private void g2ControlEvent(ControlEvent theEvent){
                      switch(theEvent.getController().getId()){
                                                                                 int BarH;
                                                                                 int BgClr; //bg of block
                      case(4):
                                                                                 int BarClr; //bar color
                              v = theEvent.getController().getValue();
                              myController.setPatternUStart((int)(v));
                                                                                 int HoverClr; //bar color on hover
                                                                                 int LabelClr;
                      break;
                      case(5):
                                                                                 int slideH;
                              v = theEvent.getController().getValue();
                             myController.setPatternUStop((int)(v));
                                                                                int slideW;
                      break;
                                                                                 int slideX;
                      case(6):
                                                                                 int slideSpc;
                              v = theEvent.getController().getValue();
                                                                                 int slideDY:
```

```
//calc bgheight, placeholder for now <----!!!</pre>
       PFont arial:
                                                                                       int bqHeight = 50;
       PFont arial10;
                                                                                       groups[numOfGroups] = cp5.addGroup(name)
       //cp5 objs
       //private Cp5UI cPanel;
                                                                                               .setId(numOfGroups)
                                                                                               .set.Label (name)
       public ControlPanel(int X, int Y, int W, int H, PApplet parent)
                                                                                               .setPosition(gpt[0] + marginLeft, gpt[1] + mar-
                                                                         ginTop)
                                                                                               .setWidth(groupWidth)
                                                                                               //.setBackgroundHeight(bgHeight)
              p = parent;
                                                                                               .setBarHeight(BarH)
              this.x = X;
                                                                                               .setBackgroundColor(BgClr)
                                                                                               .setColorBackground(BarClr)
              this.y = Y;
                                                                                               .setColorForeground(HoverClr)
              this.w = W;
                                                                                               .setColorLabel(LabelClr);
              this.h = H;
              gpt[0] = x;
                                                                                       numOfGroups++;
              gpt[1] = y;
                                                                                       qpt[1] = qpt[1] + elementHeight + marginTop;
                                                                                       ept[0] = 0 + elementPaddingLeft;
              this.dx = 25;
                                                                                       ept[1] = 0 + elementPaddingTop;
              this.dy = 30;
              cp5 = new ControlP5(p);
                                                                                       //return this;
                                                                               } * /
              cp5.setAutoDraw(false);
              numOfAccordings = 0;
                                                                                //add group
              numOfGroups = 0;
                                                                                @SuppressWarnings("deprecation")
              numOfElements = 0;
                                                                                public void addGroup(String name) {
              groups = new controlP5.Group[2];
                                                                                       //calc bgheight, placeholder for now <----!!!
                                                                                       //int bgHeight = 50;
              BarH = 20;
              BgClr = p.color(50); //bg of block
              BarClr = p.color(50); //bar color
              HoverClr = 30; //bar color on hover
              LabelClr = p.color(255);
                                                                                       groups[numOfGroups] = cp5.addGroup(name)
                                                                                               .setId(numOfGroups)
              slideH = 10;
                                                                                               .setLabel(name)
              slideW = 100;
                                                                                               .setPosition(gpt[0], gpt[1])
              slideX = 10;
                                                                                               .setWidth(groupWidth)
                                                                                               //.setBackgroundHeight(bgHeight)
              slideSpc = 5;
              slideDY = slideH+slideSpc;
                                                                                               .setBarHeight(BarH)
                                                                                               .setBackgroundColor(BgClr)
              arial = p.createFont("ArialMT-30", 11);
                                                                                               .setColorBackground(BarClr)
              arial10 = p.createFont("ArialMT-30", 10);
                                                                                               .setColorForeground(HoverClr)
                                                                                               .setColorLabel(LabelClr)
                                                                                               .hideBar();
       numOfGroups++;
///////
                                                                                       cp5.addTextlabel(name+"-label")
       public void draw() {
              //switch to 2d draw gui, then switch back
                                                                                                     .setId(numOfElements)
              //also need to disable active pane so it
                                                                                        .setText(name)
              //doesn't draw in 1of4 grid pane in gridview
                                                                                        .setHeight(elementHeight)
                                                                                        .setPosition(ept[0]+headerPaddingLeft,
                                                                         ept[1]+headerPaddingTop)
              cp5.draw();
                                                                                        //.setColorValue(0xffffff00)
              enable3d();
                                                                                        .setFont(p.createFont("Arial-bold",12))
                                                                                        .moveTo(groups[numOfGroups-1])
       ////////
                                                                                       numOfElements++:
       private void disable3d(){
              IGraphicMode mode2D = new IGraphicMode();
                                                                                       step();
              mode2D.setGraphicType(GraphicType.J2D);
              //myview.panel.currentPane().getView().setMode(mode2D);
                                                                                       //return this;
              IG.graphicMode(mode2D);
                                                                                public void endGroup(){
       private void enable3d(){
              IGraphicMode modeGL = new IGraphicMode();
              modeGL.setGraphicType(GraphicType.GL);
                                                                                       groups[numOfGroups-1].setBackgroundHeight(ept[1]);
              //myview.panel.currentPane().getView().setMode(modeGL);
              IG.graphicMode(modeGL);
                                                                                       pop();
      //public void addTab(){}
                                                                                //add sep function for floating pt values
      public void addAccording() {
                                                                                public void addSlider(String label, int minV, int maxV, int de-
                                                                                       cp5.addSlider(label)
       /*//add group
       public void addGroup(String name) {
                                                                                               .setId(numOfElements)
                                                                                               .moveTo(groups[numOfGroups-1])
```

```
//reset element registration pt
                      .setPosition(ept[0]+elementPaddingLeft,
ept[1]+elementPaddingTop)
                                                                                         ept[0] = 0;
                                                                                        ept[1] = 0;
                      .setSize(elementWidth,elementHeight)
                      .setRange(minV, maxV)
                      .setValue(defaultV);
                                                                                 private void step(){
                                                                                        //set to next element in group
               numOfElements++;
                                                                                         ept[1] += dy;
               step();
                                                                                 private void pop(){
                                                                                         //pop out of group
                                                                                         //gpt[0] -= dx;
                                                                                         gpt[1] += ept[1] + marginBottom;
       /*//add sep function for floating pt values
              public void addSlider(String label, int minV, int maxV,
int defaultV){
                                                                          /**************
                      cp5.addSlider(label)
                                                                          Gui Class
                              .setId(numOfElements)
                              .moveTo(groups[numOfGroups-1])
                                                                          Gui is a wrapper for controlP5 library, which
                                                                          provides user interface toolkit. more at:
                              .setPosition(ept[0],ept[1])
                              .setSize(elementWidth,elementHeight)
                                                                          http://www.sojamo.de/libraries/controlP5/
                              .setRange(minV, maxV)
                              .setValue(defaultV);
                                                                          Part of:
                      numOfElements++;
                                                                          Aper[na]tures
                                                                          Author: Joshua Parker
                      gpt[1] = gpt[1] + elementHeight;
                                                                          Date: 2011
                      ept[1] = ept[1] + elementHeight + elementPad-
dingTop;
                                                                          Inheritted License: GPL, V3
                                                                          *************************************
                      groups[numOfGroups-1].setBackgroundHeight(ept[1]
                                                                          package gui;
+ elementPaddingTop);
                                                                          import igeo.IG;
                                                                          import igeo.gui.IGraphicMode;
                                                                          import igeo.gui.IGraphicMode.GraphicType;
       public void addKnob(String label, int minV, int maxV, int de-
                                                                          import processing.core.PApplet;
                                                                          import processing.core.PFont;
                                                                          import controlP5.*;
               //add to style sheet
                                                                          //import remixlab.proscene.*;
              //...
                                                                          //import view.*;
                                                                          public class Gui
              cp5.addKnob(label)
                      .setId(numOfElements)
                      .moveTo(groups[numOfGroups-1])
                                                                                 //parent PApplet
                      .setPosition(20,20)
                                                                                 PApplet p;
                      .setRadius(35)
                      .setAngleRange(2*PApplet.PI)
                                                                                 //cp5 objs
                      .setStartAngle((float)(.5*PApplet.PI))
                                                                                 private ControlPanel cPanel;
                      .setRange(minV, maxV)
                      .setValue(defaultV)
                                                                                 //CONSTRUCTOR
                      .setNumberOfTickMarks(16)
                                                                                 public Gui(PApplet parent){
                      .snapToTickMarks(true)
                      .setColorForeground(p.color(255))
                                                                                         //initialize
                      .setColorBackground(p.color(40))
                                                                                         p = parent;
                      .setColorActive(p.color(255,255,0))
                      .setDragDirection(Knob.HORIZONTAL);
                                                                                         cPanel = new ControlPanel(20,20, 200, p.height, p);
              numOfElements++;
                                                                                         cPanel.addGroup("Resample surface");
                                                                                        cPanel.addSlider("u Samples", 5, 200, 20);
cPanel.addSlider("v Samples", 5, 200, 20);
       public void addTextArea(String label, String text){
                                                                                         cPanel.endGroup();
                                                                                        cPanel.addGroup("Pattern u/v range");
               //add to style sheet
                                                                                        cPanel.addSlider("u start", 0, 100, 25);
               //...
                                                                                         cPanel.addSlider("u stop", 0, 100, 75);
                                                                                         cPanel.addSlider("v start", 0, 100, 25);
               cp5.addTextarea(label)
                                                                                         cPanel.addSlider("v stop", 0, 100, 75);
                      .moveTo(groups[numOfGroups-1])
                                                                                         cPanel.endGroup();
                      .setPosition(5,5)
                      .setSize(195,90)
                      .setFont(arial)
                      .setLineHeight(14)
                                                                                 .setColor(p.color(128))
                                                                          ///////
                      .setColorBackground(p.color(50))
                      .setColorForeground(p.color(255,100))
                                                                                 public void draw() {
                                                                                         //switch to 2d draw gui, then switch back
                      .setText(text);
                                                                                         //also need to disable active pane so it
                                                                                         //doesn't draw in lof4 grid pane in gridview
              numOfElements++;
                                                                                         disable3d();
                                                                                         cp5.draw();
       private void push(){
                                                                                         enable3d();
              //push into group
               //qpt[0] += dx;
                                                                                         cPanel.draw();
```

30

//gpt[1] += dy;

		<pre>panel.setVisible(true);</pre>
/*// PRIVATE METHODS ////////////////////////////////////		<pre>// initialize iGeo IG ig = IG.init(panel);</pre>
<pre>private void disable3d() { IGraphicMode mode2D = new IGraphicMode(); mode2D.setGraphicType(GraphicType.J2D); //myview.panel.currentPane().getView().setMode(mode2D); IGraphicMode(mode2D);</pre>	Path	<pre>ig.server().graphicServer().enableGL(); // //ig.setBasePath(parent.sketchPath("")); // not sketch</pre>
<pre>IG.graphicMode(mode2D); }</pre>		<pre>if(!parent.online){ // only when running local ig.setBasePath(parent.dataPath("")); // for defaul</pre>
<pre>private void enable3d() { IGraphicMode modeGL = new IGraphicMode(); modeGL.setGraphicType(GraphicType.GL);</pre>	path to	read/write files }
<pre>//myview.panel.currentPane().getView().setMode(modeGL); IG.graphicMode(modeGL);</pre>		<pre>ig.setInputWrapper(new PIInput(parent));</pre>
}*/ }		<pre>parent.addMouseListener(panel); parent.addMouseMotionListener(panel); parent.addMouseWheelListener(panel);</pre>
package migeo; import igeo.IG;		<pre>parent.addKeyListener(panel); parent.addFocusListener(panel); parent.addComponentListener(panel);</pre>
public class MyIG extends IG{		<pre>//igg = new IGraphics(); igg = new IGraphicsGL();</pre>
/** Processing Graphics using OpenGL to be put in size() method in Processing */		//noSmooth();
<pre>public static final String GL = "igeo.p.MyPIGraphicsGL";</pre>		<pre>if(PIConfig.drawBeforeProcessing) parent.</pre>
<pre>} package migeo;</pre>	register	rPre(this); else parent.registerDraw(this); parent.registerPost(this);
<pre>import java.awt.Color; import javax.media.opengl.GL;</pre>	}	<pre>if(PIConfig.resizable){ parent.frame.setResizable(true)</pre>
import processing.core.PApplet;		<pre>super.hints[DISABLE_OPENGL_2X_SMOOTH]=true; // super.hints[ENABLE_OPENGL_4X_SMOOTH]=true; //</pre>
import com.sun.opengl.util.j2d.Overlay;		}
<pre>import igeo.IG; import igeo.gui.IGraphicsGL; import igeo.gui.IGridPanel; import igeo.gui.IPaneLight; import igeo.gui.IPanel; import igeo.gui.IView; import igeo.p.PIConfiq;</pre>		<pre>public void setGLProperties() { gl.glEnable(GL.GL_MULTISAMPLE); // gl.glEnable(GL.GL_POINT_SMOOTH); // gl.glEnable(GL.GL_LINE_SMOOTH); // gl.glEnable(GL.GL_POLYGON_SMOOTH); // gl.glEnable(GL.GL_ALPHA_TEST); //</pre>
<pre>import igeo.p.PIGraphicsGL; import igeo.p.PIInput;</pre>		<pre>//gl.glEnable(GL.GL_BLEND); // //gl.glDisable(GL.GL_BLEND); // //gl.glBlendFunc(GL.GL_SRC_ALPHA, GL.GL_ONE_MINUS_SRC_</pre>
public class MyPIGraphicsGL extends PIGraphicsGL{	ALPHA);	<pre>// //gl.glBlendFunc(GL.GL_SRC_ALPHA, GL.GL_ONE); //</pre>
<pre>//public PIGraphicsGL() { super(); } public IPaneLight pane;</pre>		<pre>gl.glHint(GL.GL_LINE_SMOOTH_HINT, GL.GL_NICEST); // gl.glHint(GL.GL_POINT_SMOOTH_HINT, GL.GL_NICEST); // gl.glHint(GL.GL_POLYGON_SMOOTH_HINT, GL.GL_NICEST); //</pre>
<pre>/** setParent is called by Processing in the initialization process of Processing. Here the initialization proces of iGeo is also done.</pre>		<pre>//gl.glEnable(GL.GL_NORMALIZE); // //gl.glEnable(GL.GL_AUTO_NORMAL); // //gl.glShadeModel(GL.GL_SMOOTH); //</pre>
<pre>@param parent parent PApplet of Processing. */</pre>		<pre>//gl.glLightModeli(GL.GL_LIGHT_MODEL_TWO_SIDE, 1); //</pre>
<pre>public void setParent(PApplet parent) {</pre>	super	<pre>//gl.glEnable(GL.GL_DEPTH_TEST); // already enabled in</pre>
<pre>super.setParent(parent);</pre>		<pre>//gl.glDisable(GL.GL_DEPTH_TEST); // ? for transparenc</pre>
<pre>// initialize root GUI //panel = new IGridPanel(0,0,parent.getWidth(),parent. getHeight(),2,2);</pre>		<pre>//gl.glEnable(GL.GL_LIGHTING); // test! //gl.glEnable(GL.GL_LIGHT1); // test!</pre>
<pre>panel = new IPanel(0,0,parent.getWidth(),parent. getHeight());</pre>		}
<pre>IView v=null; v = IView.getDefaultPerspectiveView(0,0,parent.</pre>		<pre>public void pre() { drawIG(); } public void draw() { drawIG(); }</pre>
<pre>getWidth(),parent.getHeight());</pre>		/** Drawing all the iGeo objects through IPanel.
<pre>v.enableRotationAroundTarget(); // here? v.setTarget(0,0,0); //</pre>	3D graph	Overlay is also used to draw 2D graphics on top of OpenGL hics. */
<pre>pane = new IPaneLight(0,0,parent.getWidth(),parent. getHeight(),v,panel);</pre>		public synchronized void drawIG(){
<pre>panel.addPane(pane); panel.currentMousePane = pane;</pre>		<pre>int[] viewport=null; if(PIConfig.restoreGLViewport) { viewport = new int[4];</pre>

```
gl.glGetIntegerv(GL.GL_VIEWPORT, viewport, 0);
                                                                                  public void mousePressed(MouseEvent e) {
              gl.glMatrixMode(GL.GL_MODELVIEW);
                                                                                  public void mouseReleased(MouseEvent e) {
            gl.glPushMatrix();
                                                                                  public void mouseClicked(MouseEvent e) {
            gl.glMatrixMode(GL.GL_PROJECTION);
            gl.glPushMatrix();
                                                                                  public void mouseEntered(MouseEvent e) {
              if(PIConfig.resetGLDepthBefore) gl.glClear(GL.GL_DEPTH_
                                                                                  public void mouseExited(MouseEvent e){
BUFFER_BIT);
                                                                                  public void mouseMoved(MouseEvent e) {
              //gl.glClear(GL.GL COLOR BUFFER BIT | GL.GL DEPTH BUF-
                                                                                  public void mouseDragged(MouseEvent e) {
FER_BIT);
              //gl.glClear(GL.GL COLOR BUFFER BIT);
                                                                                  public void mouseWheelMoved(MouseWheelEvent e) {
              setGLProperties();
                                                                                  public void keyPressed(KeyEvent e) {
               if(igg.getGraphics() ==null) {
                                                                                  public void keyReleased(KeyEvent e) {
                  setOverlay();
                                                                                  public void keyTyped(KeyEvent e){
              //igg.setGraphics(overlay.createGraphics());
                                                                                  public void focusLost(FocusEvent e) {
              igg.getGraphics().clearRect(0,0,parent.
                                                                                  public void focusGained(FocusEvent e) {
getWidth(),parent.getHeight()); //
               //overlay = new Overlay(drawable); //
              //Graphics2D g = overlay.createGraphics();
                                                                                  public void componentHidden(ComponentEvent e) {
              //igg.setGraphics(g);
                                                                                  public void componentMoved(ComponentEvent e) {
              igg.setGL(gl);
                                                                                  public synchronized void componentResized(ComponentEvent e){
              panel.draw(igg);
                                                                                         int w = e.getComponent().getBounds().width;
                                                                                         int h = e.getComponent().getBounds().height;
              if(PIConfig.resetGLDepthAfter) gl.glClear(GL.GL_DEPTH_
                                                                                         setSize(w,h);
BUFFER BIT);
                                                                                  public void componentShown(ComponentEvent e) {
              gl.glMatrixMode(GL.GL PROJECTION);
            gl.glPopMatrix();
              gl.glMatrixMode(GL.GL_MODELVIEW);
            gl.glPopMatrix();
                                                                          /*************
                                                                          FramePanel Class
               // bring the original viewport back
              if(PIConfig.restoreGLViewport && viewport!=null){
                                                                          Part of:
                  \verb"gl.glViewport"(viewport[0]", viewport[1]", view-
port[2], viewport[3]);
                                                                          Aper[na]tures
                                                                          Author: Joshua Parker
                                                                          Date: 2011
               if(overlay!=null){
                  overlay.markDirty(0,0,parent.getWidth(),parent.
                                                                          Inheritted License: GPL, V3
getHeight());
                  overlay.drawAll();
                                                                          ****************
                                                                          package model;
               //q.dispose();
                                                                          import igeo. IVec;
                                                                          import processing.core.PApplet;
               //igg.getGraphics().dispose();
                                                                          public class FramePanel {
        public void setOverlay(){
              overlay = new Overlay(drawable); //
                                                                                 //The parent PApplet
               igg.setGraphics(overlay.createGraphics());
                                                                                 PApplet p;
               igg.getGraphics().setBackground(overlayBG);
                                                                                 double MAXTHICK = .2;
        public void setSize(int w, int h){
                                                                                 public IVec center;
               super.setSize(w,h);
                                                                                 public IVec normal;
               setOverlay(); // update overlay
                                                                                 public IVec[] corners = new IVec[4];
                                                                                 public double openness; //0.0 - 1.0
                                                                                 public double thinness; //0.0 - 1.0
        public void post(){
                                                                                 public IVec[] pts = new IVec[5];
              if(overwritePAppletFinish) parent.finished=finished;
                                                                                 public IVec[] normals = new IVec[5];
               if(overwritePAppletLoop) if(looping) parent.loop();
else parent.noLoop();
                                                                                 public FramePanel(IVec[] Pts, IVec[] Normals, double Openness,
                                                                          PApplet parent) {
        public void loop(){ if(!looping) looping=true; }
                                                                                        p = parent;
        public void noLoop(){ if(looping) looping=false; }
                                                                                         //pts = Pts;
        public void start() { if(finished) finished=false; }
                                                                                        //normals = Normals;
        public void stop(){ if(!finished) finished=true; }
```

center = Pts[0].dup();

```
normal = Normals[0].dup();
                                                                                          for(int i=0; i <= unum; i++){
                                                                                            for(int j=0; j < vnum; j++) {
  if( (i+j) %2 == 0 ) {</pre>
              corners[0] = Pts[1].dup();
              corners[1] = Pts[2].dup();
              corners[2] = Pts[3].dup();
                                                                                                IVec c = surf.pt( i*uinc, j*vinc );
              corners[3] = Pts[4].dup();
                                                                                               IVec nc = surf.normal( i*uinc, j*vinc ).len(.2);
               openness = Openness;
                                                                                                IVec p1 = surf.pt( (i-1)*uinc, j*vinc );
              thinness = .5;
                                                                                                IVec n1 = surf.normal( (i-1)*uinc, j*vinc
                                                                         ).len(.2);
                                                                                                IVec p2 = surf.pt( i*uinc, (j-1)*vinc );
                                                                                                IVec n2 = surf.normal( i*uinc, (j-1) *vinc
                                                                         ).len(.2);
/**************
                                                                                                IVec p3 = surf.pt( (i+1)*uinc, j*vinc );
                                                                                                IVec n3 = surf.normal( (i+1)*uinc, j*vinc
Model Class
                                                                         ).len(.2);
Part of:
                                                                                                IVec p4 = surf.pt( i*uinc, (j+1)*vinc );
                                                                                                IVec n4 = surf.normal( i*uinc, (j+1)*vinc
Aper[na]tures
                                                                         ).len(.2);
Author: Joshua Parker
Date: 2011
                                                                                                IVec[] pts = {c,p1,p2,p3,p4};
Inheritted License: GPL, V3
                                                                                                IVec[] normals = \{nc, n1, n2, n3, n4\};
*******************
                                                                                                /*IVec xaxis = new IVec(1,0,0);
package model;
                                                                                                IVec yaxis = new IVec(0,1,0);
                                                                                                double[] ref = nc.projectTo2Vec(xaxis, yaxis);
import java.util.ArrayList;
                                                                                                IVec vRef = new IVec(ref[0], ref[1], ref[2]);
                                                                                                double openness = (nc.angle(vRef)/PApplet.
import processing.core.PApplet;
                                                                          PI) * (openMax-openMin) +openMin; */
import igeo.IBox;
import igeo.ICurve;
                                                                                                double openness;
import igeo.IFieldVisualizer;
                                                                                                if(i<uMax & i>uMin & j<vMax & j>vMin){
                                                                                                      openness = map.get( i*uinc, j*vinc
import igeo.IG;
import igeo.IImageMap;
                                                                          ) * (openMax-openMin) +openMin;
import igeo.IPoint;
                                                                                                }else{
import igeo.IRandom;
                                                                                                      openness = 1;
import igeo. I Surface;
import igeo. ISurface Normal Field;
import igeo.IVec;
                                                                                                panels.add(new FramePanel(pts, normals, open-
                                                                          ness, p));
//import igeo.IVec;
//import processing.core.PApplet;
public class Model {
       //The parent PApplet
       PApplet p;
       public ISurface surf;
       public IImageMap map;
                                                                          /**************
       public int unum, vnum;
       public int uMax, uMin, vMax, vMin;
       public double openMax, openMin;
       public ArrayList<FramePanel> panels;
                                                                          View - contains the scene(empty space and camera),
                                                                           imported 3d base model, and data model defining
       public Model(PApplet parent){
                                                                           panel system. View calls draws to screen via
                                                                           viewState interface, though only one viewState
              p = parent;
                                                                           is implemented: View3d, a basic 3d view.
              //load assets and create data model here. SHOULD NOT
NOT BE GEOMETRY HERE!!! SHOULD CREATE LOADER CLASS FOR THAT.
                                                                         Part of:
              //IG.open("testsrf-zhuhaiPavilion-untrimTop.3dm");
              IG.open("testsrf-zhuhaiPavilion-wormhole1.2.3dm");
                                                                          Aper[na]tures
              ISurface[] surfs = IG.surfaces();
                                                                         Author: Joshua Parker
              map = new IImageMap("map1.jpg");
                                                                         Date: 2011
              surf = surfs[0];
                                                                          Inheritted License: GPL, V3
              unum = 20;
              vnum = 20;
                                                                          ************************************
              uMax = 15;
                                                                          package view;
              uMin = 5;
              vMax = 15;
                                                                          import processing.core.PApplet;
                                                                          import model.Model;
               vMin = 5;
              openMax = .8;
                                                                          import view.View1;
              openMin = .2;
                                                                          import view.ViewState;
                                                                          public class View{
       //update Model
                                                                                 //parent PApplet
       public void update(){
                                                                                 PApplet p;
              double uinc = 1.0/unum, vinc = 1.0/vnum;
                                                                                 //data model
              panels = new ArrayList<FramePanel>();
                                                                                 public Model mymodel;
```

```
//CONSTRUCTOR
       //states
      private ViewState view1;
                                                                         View1(PApplet parent, View Myview){
      private ViewState view2;
                                                                               p = parent;
       //current state
      public ViewState curState;
                                                                               this.view = Myview;
       public View(PApplet parent, Model Mymodel){
                                                                               //moved to model...
             p = parent;
                                                                         this.mymodel = Mymodel;
                                                                  ///////
                                                                         //draw the 3d view
             view1 = new View1(p, this);
             view2 = new View2(p, this);
                                                                         public void draw()
                                                                               //clear geometry in the scene
             //view state
             setViewState(view1);
                                                                               IG.cur().server().clear();
                                                                               //IG.current().panel.currentPane().getView().mode().
      setDrawMode(true, false, false);
///////
                                                                               //draw
       //set view1 state
                                                                               for(FramePanel panel : view.mymodel.panels){
      public void setView1(){
             setViewState(view1);
                                                                                      //outer and inner corner pts of panel
                                                                                      IVec[] outer = new IVec[4];
                                                                                      IVec[] inner = new IVec[4];
       //set view2 state
       public void setView2(){
                                                                                      //locate inner corners (already have outer)
             setViewState(view2);
                                                                                      for(int i=0; i<outer.length; i++){</pre>
                                                                                             outer[i] = panel.corners[i].dup();
                                                                                             inner[i] = panel.center.dup()
       //draw the view
                                                                                                             .sub(outer[i])
      public void draw() {
                                                                         //get miter vector
             this.curState.draw();
                                                                                                             .scale(panel.openness)
                                                                  //scale by openness
                                                                                                             .add(outer[i]);
      //position at corner
////////
       //set view state
                                                                                      if(panel.openness==1){
      private void setViewState(ViewState newState){
                                                                                             new ISurface (outer[0], outer[1], out-
                                                                  er[2], outer[3]);
             this.curState = newState;
                                                                                      }else{
                                                                                             //create mitered frame faces
/**********************************
                                                                                             for(int j=0; j<outer.length; j++) {</pre>
View3d Class
                                                                                                          new ISurface(outer[j],
                                                                  outer[(j+1)%4], inner[(j+1)%4], inner[j])
View3d implements a basic 3d view
                                                                                                          .clr(panel.openness, 0,
Part of:
Aper[na]tures
Author: Joshua Parker
                                                                         }//end draw
Date: 2011
Inheritted License: GPL, V3
                                                                         *****************
package view;
                                                                  /
/******************
import java.util.ArrayList;
                                                                  View3d Class
import model.FramePanel;
                                                                  View3d implements a basic 3d view
import model. Model;
import processing.core.*;
import igeo.IBox;
                                                                  Part of:
import igeo. IBrep;
import igeo.ICurve;
                                                                  Aper[na]tures
import igeo.IG;
                                                                  Author: Joshua Parker
import igeo.IMesh;
                                                                  Date: 2011
import igeo.IPoint;
                                                                  Inheritted License: GPL, V3
import igeo.ISurface;
import igeo.IVec;
                                                                  *************************************
public class View1 implements ViewState{
                                                                  package view;
       //The parent PApplet
                                                                  import processing.core.*;
                                                                  import saito.objloader.*;
      PApplet p;
                                                                  import migeo.*;
      //composit view
                                                                  import model.*;
                                                                  import processing.core.PVector;
       public View view;
       public Model model;
                                                                  import igeo. IBrep;
```

import igeo. ICurve;

```
import igeo.IG;
import igeo.IMesh;
import igeo.IPoint;
import igeo.ISurface;
import igeo.IVec;
public class View2 implements ViewState{
       //The parent PApplet
      PApplet p;
      //composit view
public View view;
public Model model;
       //CONSTRUCTOR
      View2(PApplet parent, View Myview){
             p = parent;
             this.view = Myview;
             //moved to model...
      //////
      //draw the 3d view
      public void draw() {
             //clear geometry in the scene
IG.cur().server().clear();
             IG.bg(25, 25, 50, 50);
             //draw
             for(FramePanel panel : view.mymodel.panels){
                    new ICurve(panel.corners[0], panel.corners[1]).
clr(150);
                    new ICurve(panel.corners[1], panel.corners[2]).
clr(150);
      }//end draw
      ///////
ViewState Class
ViewState is an interface for various views
Part of:
Aper[na]tures
Author: Joshua Parker
Date: 2011
Inheritted License: GPL, V3
************************************
package view;
public interface ViewState{
        //public void enableFill();
        //public void disableFill();
        //public boolean getFill();
        public void draw();
```

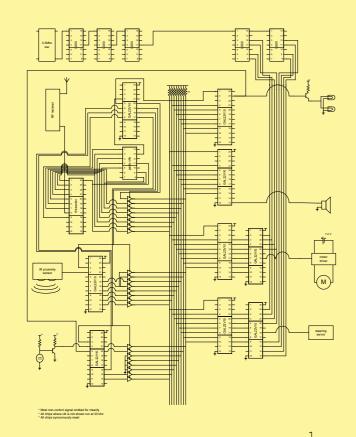
BEING THERE. SOURCE CODE

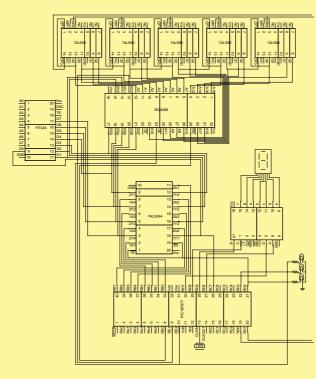
THE CAR PROJECT CONSISTS OF TWO MAIN MODULES. ONE OF THE MAIN MODULES IN THE USER INTERFACE CIRCUIT WHICH HANDLES ALL DATA COMING IN FROM THE USER AND SENDS IT OFF TO THE RF TRANSMITTER. THE OTHER MAIN MODULE RECEIVES THE INCOMING DATA FROM THREE SEPARATE SOURCES: THE RF RECEIVER, LIGHT SENSOR, AND PROXIMITY SENSOR. USING THESE SIGNALS IT CAN SUCCESSFULLY CONTROL THE STEERING, HORN, LIGHTS, AND MOTOR. THE LATTER CIRCUIT IS ACTUALLY ON THE RC CAR. THE USER WILL ALSO HAVE A REAL-TIME VIDEO FEED AVAILABLE TO BE ABLE TO WATCH WHERE THEY ARE GOING. THIS TRANSLATES TO A RC CAR CONTROLLED BY THE USER VIA VIDEO GAME CONTROLLER. THE STEERING WHEEL CONTROLLER WILL SEND DATA TO A PIC MICROCONTROLLER, WHICH WILL BE THE MAIN CONTROL UNIT ON THE USER SIDE OF THE SYSTEM. A DIGITAL READOUT OF THE CURRENT GEAR AND WELL AS STATUS LIGHTS WILL BE PRESENT. SRAM WILL BE AVAILABLE TO RECORD INPUT FROM THE USER. THE CAR WILL RESPOND TO COMMANDS BY THE USER VIA RF COMMUNICATION.

CONTRIBUTORS:
KEVIN LARSSON
JOSH PARKER
MATT ZOBEL

ADVIISORS: JAMES K. PECKOL

main board schematic
 control board schematic





The user module is controlled by the PIC microcontroller. The PIC accepts user input via the steering wheel, gas pedal, brake pedal, and buttons. Since the wheel, and pedals are potentiometer devices the PIC sample these measurements using the internal A/D converter. These measurements are then encoded internally in the PIC to compress some of the bits that are used. The PIC samples these devices continually and sends out three separate frames in sequence over RF transmission. The first frame contains speed information, the second contains turning information, and the third contains button push information.

To help detect errors a parity checker is used to calculate even parity and the parity bit is sent along with the transmission. The frames are sent to the RF encoder and then onto the RF transmitter for transmission to the car at 434MHz.

The PIC also controls some user displays according to the current state of the car. There is a gear display near the user that display the current gear the car is in. The PIC will increment or decrement the gear according to user input. The valid gears are 1,2,3,and R. In addition, the PIC controls 3 LED's that indicate the current status of the memory. The Green LED indicates that the car is under control of the user. The yellow LED indicates the memory is currently being written to. The Red LED means the memory is being read, nullifying any user input.

To control the memory the PIC uses 5 cascaded 74LS93 chips for addressing. The PIC sends out clock signals to the 74LS93 chips to increment them each time an address is written or read. The PIC is also responsible for clocking the SRAM to look at the current address and either read or write and is responsible for adding a flag bit to the last instruction written to memory. This is all done via the 8 bit parallel bus.

The car receives the data from the user module via RF transmission at 434MHz. The incoming data is sent through the RF decoder and onto the bus and into the parity checker. The Bus Master waits to get a valid transmit signal from the RF decoder. When Bus Master sees this signal he makes sure that the parity was correct. If so he strobes everyone listening to the bus to capture the new information.

Bus Master is also in charge of flipping the bus around when the proximity sensor chip or the light sensor chip have something to write to the bus. When the requests are made the Bus Master will flip the bus at the appropriate time and then flip it back when they are done.

After the Bus Master strobes the other nodes to capture the information on the bus, each node looks at the type of frame it is, as shown on Figure 5. Using bits 6 and 7 each node knows what type of frame to look at, if it is the wrong frame, it just ignores it. Speed decoder, steering decoder, horn controller, and light controller all listen to the bus continually.

Speed decoder and steering decoder decode the five-bit data stream in a speed frame or steering frame. With the input from the bus they decode the data into an eight-bit BCD value corresponding to the actual PWM signal that the generators will generate. For example, in a steering frame a 10000 sent by the user module gets decoded into a 42% duty cycle, therefore, the module puts out a 4 and 2 in BCD. The generators use 2 cascaded 74LS90 chips that are counting in BCD to do a comparison on. When the BCD input equals the BCD count, the PWM signal goes down and a PWM generation is produced.

The proximity sensor node and light sensor node have a little more intelligence. These nodes are responsible for writing to the bus to control the four nodes that are listening. If the light sensor senses a discrepancy between the status of the light and the status of the sensor, the light sensor node asks the bus master for bus control. When the control is granted the light sensor will write information to the bus to turn the light on or off appropriately.

he proximity sensor node works similarly to the light sensor. When the node receives information that the car is near a wall the node asks for control of the bus. When the control is granted, the proximity sensor node writes information to the bus, then to the speed controller, telling it to throw it into reverse. When the wall is no longer detected the proximity sensor then writes to the bus again telling the speed controller to go to idle. It must be noted that the bus is completely tied up by the proximity sensor node when the wall is near the car.

Using this implementation resulted in a successful project that operated correctly. Timing was the key ingredient and although we had some issues with timing in the beginning, in the end, it worked very

			present paritycheck /* 001 */
			default next paritycheck;
			if(proxsense)
UPL CODE			next proxdetect;
			<pre>if(!parity & !proxsense) next idle;</pre>
			if(parity & !proxsense)
usmaster.	pld		next strobe;
			present strobe /* 010 */
ame	busmstr;		default
artno	GAL22V10;		next idle;
ate evision	05/06/02;		if (proxsense)
esigner	XX; XXXXX;		next proxdetect;
ompany	XXXXX;		present proxdetect /* 011 */
ssembly	XXXXX;		default
ocation evice	XXXXX; g22v10;		next proxdetect; if(strobein)
	5,		next proxdetect2;
	*****	**********	,
*	MINDOM	V/UNKNOWN */	*/ present proxdetect2 /* 111 */ default
*		,, отпанения	*/ next proxdetect2;
*****	*****	*********	, 11 (00100011)
** Input	s **/		next proxwait;
Input	.5/		present proxwait /* 100 */
in 1	= clk;		default
in 2	= reset;		next idle;
in 4	= parity;		present lightdetect /* 101 */
in 5	= strobein;		default
in 6	= proxsense;		next lightdetect;
in 7 in 8	<pre>= lightsense; = validtransm</pre>		<pre>if(proxsense) next proxdetect;</pre>
111 0	- valideransi	arc,	if(strobein & !proxsense)
** Outpu	ts **/		next lightwait;
in 17	= lightonable		procent lighturit /t 110 t/
	<pre>= lightenable = proxenable;</pre>		present lightwait /* 110 */ default
	= mstrenable;		next idle;
in 20	= strobeout;		}
in [21	23] = [s2s0];		
			/** Logic Equations **/
IELD busc	ntrl = [s2s0];		
DEFINE id	11.0	'b' 000	idl = !s2 & !s1 & !s0; parchk = !s2 & !s1 & s0;
	ritycheck	'b' 001	str = !s2 & s1 & !s0;
DEFINE st	robe	'b' 010	proxd = !s2 & s1 & s0;
DEFINE pr		'b' 011	proxd2 = s2 & s1 & s0;
DEFINE pr	ghtdetect	'b'100 'b'101	proxw = s2 & !s1 & !s0; lightd = s2 & !s1 & s0;
DEFINE li	=	'b'110	lightw = s2 & s1 & !s0;
DEFINE pr	oxdetect2	'b' 111	
			<pre>strobeout = str; mstrenable = !(idl # parchk # str);</pre>
** Declar	ations and Intermed	diate Variable Definitions **/	Enable */
			<pre>proxenable = !(proxd # proxw # proxd2);</pre>
0.ar	= reset;		Enable */
1.ar 2.ar	= reset; = reset;		<pre>lightenable = !(lightd # lightw);</pre>
0.sp	= 'b'0;		honker.pld
1.sp 2.sp	= 'b'0; = 'b'0;		Name honker;
	<i>~</i> ∨,		Partno GAL22V10;
0.oe	= 'b'1;		Date 05/27/02;
1.oe 2.oe	= 'b'1; = 'b'1;		Revision XX; Designer XXXXX;
2.00	ν τ,		Company XXXXX;
equence b	ouscntrl		Assembly XXXXX;
			Location XXXXX;
nre	esent idle	/* 000 */	Device g22v10;
P-1	default		/**************************************
ext idle;			/* */
ext proxd	if(proxsense)		/* WINDOW/UNKNOWN */ /* */
.cac proxo		nit & !proxsense)	/**/**********************************
ext parit	ycheck;		
ov+ 1:~		& !proxsense & !strobein)	/** Inputs **/
ext light	uctect,		Pin 1 = clk;

```
Pin 2 = reset;
Pin 3 = strobe;
                                                              s0.ar
                                                                          = reset;
                                                              s0.sp
                                                                          = 'b'0;
Pin [4..10] = [bus1..bus7];
                                                              s0.oe
/** Outputs **/
                                                              sequence buscntrl
Pin 13
            = out_freq;
Pin 22
            = s0;
                                                                    present off
Pin 23
            = honkactive;
                                                                          default
                                                              next off;
                                                                          if(strobe & bus1 & bus7 & !bus6)
FIELD buscntrl = [s0];
                                                              next on;
$DEFINE off
                                                                   present on
                         'b'1
$DEFINE on
                                                                          default
                                                              next on;
                                                                          if(strobe & bus1 & bus7 & !bus6)
/** Declarations and Intermediate Variable Definitions **/
                                                              next off;
s0.ar
            = reset;
            = 'b'0;
s0.sp
           = 'b'1;
                                                              /** Logic Equations **/
s0.oe
                                                              lightactive = s0;
sequence buscntrl
                                                              lightsense.pld
      present off
                                                              Name
                                                                       lsense;
            default
                                                              Partno GAL22V10;
                                                              Date 05/06/02;
            if(strobe & bus2 & bus7 & !bus6)
                                                              Revision XX;
                                                              Designer XXXXX;
                                                              Company XXXXX;
     present on
                                                              Assembly XXXXX;
            default
                                                              Location XXXXX;
                                                              Device g22v10;
next on;
            if(strobe & !bus2 & bus7 & !bus6)
                                                               next off;
                                                                                WINDOW/UNKNOWN
/** Logic Equations **/
                                                              /** Inputs **/
honkactive = s0 & out_freq;
lighter.pld
                                                              Pin 1
                                                                        = clk;
                                                              Pin 2
                                                                         = reset;
         lighter;
                                                              Pin 3
                                                                         = !flipenable;
                                                                        = !lightsense;
Partno GAL22V10;
                                                              Pin 4
Date 05/27/02;
                                                              Pin 5
                                                                         = lightstatus;
Revision XX;
                                                                       = honkstatus;
Designer XXXXX;
Company XXXXX;
Assembly XXXXX;
                                                              /** Outputs **/
Location XXXXX;
Device g22v10;
                                                              Pin 14
                                                                         = strobe;
                                                              Pin 15
                                                                        = bus1;
                                                                       = bus2;
= fliprequest;
                                                              Pin 16
                                                              Pin 17
                  WINDOW/UNKNOWN
                                                              Pin [18..20] = [s0..2];
                                                             Pin 21 = bus6;
Pin 22 = bus7;
/** Inputs **/
Pin 1 = clk;
Pin 2 = reset;
Pin 3 = strobe;
                                                              FIELD lightcntrl = [s2..s0];
                                                              $DEFINE idle
                                                                                       'b'000
Pin [4..10] = [bus1..bus7];
                                                              $DEFINE flipreq
                                                                                       'b'001
                                                              $DEFINE datasend
                                                                                       'b'010
/** Outputs **/
                                                              $DEFINE strobey
                                                                                            'b'011
                                                              $DEFINE wait
                                                                                       'b'100
            = s0;
Pin 23
           = lightactive;
                                                              /** Declarations and Intermediate Variable Definitions **/
FIELD buscntrl = [s0];
                                                              s0.ar = reset;
                                                              s1.ar = reset;
                                                              s2.ar = reset;
$DEFINE off
$DEFINE on
                         'b'1
                                                              s0.sp = 'b'0;
                                                              s1.sp = 'b'0;
```

s2.sp = 'b'0;

/** Declarations and Intermediate Variable Definitions **/

s0.oe = 'b' s1.oe = 'b' s2.oe = 'b'	1;	\$DEFINE rev_send 'b'010 \$DEFINE rev_strobey 'b'011 \$DEFINE rev_wait 'b'100 \$DEFINE idle send 'b'101
sequence lighto	entrl	\$DEFINE idle_strobey b'110 \$DEFINE idle_wait b'111
present	idle	<pre>/** Declarations and Intermediate Variable Definitions **/</pre>
next idle;	default	s0.ar = reset;
next rare,	<pre>if((lightsense \$ lightstatus))</pre>	s1.ar = reset;
next flipreq;		s2.ar = reset;
present	flipreq default	s0.sp = 'b'0; s1.sp = 'b'0;
next flipreq;		s2.sp = b'0;
next datasend;	if(flipenable)	s0.oe = 'b'1;
		s1.oe = 'b'1;
present.	datasend	s2.oe = b'1;
F	default	sequence proxcntrl
next strobey;		{
present	strobey	present idle default
	default	next idle;
next wait;		<pre>if (proxsense) next flipreq;</pre>
present	wait	
next idle;	default	present flipreq default
nene lale,		next flipreq;
}		<pre>if(flipenable) next rev_send;</pre>
/** Logic Equa	ations **/	
		present rev_send
	= !s2 & s1 & s0; = 'b'1;	default next rev strobey;
	= honkstatus;	
	= 'b'0; = 'b'1;	nresent rev strokev
	= !s2 & !s1 & s0;	<pre>present rev_strobey default</pre>
		next rev_wait;
Proximity Sense	e.pld	present rev wait
		default
Name pser	nse;	<pre>next rev_wait;</pre>
Partno GAL2	22V10;	next idle_send;
Date 05/0 Revision XX;	06/02;	present idle_send
Designer XXXX	XX;	default
Company XXXX		<pre>next idle_strobey;</pre>
Assembly XXXX Location XXXX		present idle strobey
Device g22v	710;	default
/*********	**************	next idle_wait; /
/*	*	· · · · · · · · · · · · · · · · · · ·
/* /*	WINDOW/UNKNOWN */	default / next idle;

/** Inputs **	*/	}
Pin 1	= clk;	/** Logic Equations **/
	= reset;	
	<pre>= !flipenable; = proxsense;</pre>	strobe = (!s2 & s1 & s0) # (s2 & s1 & !s0); bus1 = 'b'0;
/** Outputs *		bus2 = (!s2 & s1 & !s0) # (!s2 & s1 & s0) # (s2 & !s1 & !s0); bus3 = (!s2 & s1 & !s0) # (!s2 & s1 & s0) # (s2 & !s1 & !s0);
Pin 14	= strobe;	bus4 = 'b'0; bus5 = (s2 & !s1 & s0) # (s2 & s1 & !s0) # (s2 & s1 & s0);
Pin [1519] Pin [2022]	= [bus15]; /* Tie Bus 6 & 7 to Pin 18 */	fliprequest = !s2 & !s1 & s0;
FIELD proxcntrl	l = [s2s0];	Speed Generator.pld
-		
\$DEFINE idle \$DEFINE flipreq	`b'000 `b'001	Name spedpw90; Partno GAL22V10;

```
05/27/02;
Date
Revision XX;
Designer XXXXX;
        xxxxx:
Company
Assembly
        XXXXX;
Location XXXXX;
Device
        g22v10;
WINDOW/UNKNOWN
/****************/
/** Inputs **/
Pin 1
           = clk;
           = reset;
Pin 2
Pin [3..6] = [MSB3..MSB0];
Pin [7..10] = [LSB3..LSB0];
Pin [13..16] = [clkMSB0..clkMSB3];
Pin [20..23] = [clkLSB0..clkLSB3];
/** Outputs **/
           = !a0;
                        /\star this is the PWM output to the motor
Pin 18
Pin 19
           = temp;
FIELD pwm = [a0];
/** Declarations and Intermediate Variable Definitions **/
a0.ar
            = reset;
a0.sp
           = !reset;
a0.oe
                  (LSB0 $ clkLSB0) #
temp =
                  (LSB1 $ clkLSB1) #
                  (LSB2 $ clkLSB2) #
                  (LSB3 $ clkLSB3) #
                  (MSB0 $ clkMSB0);
sequence pwm
      present 'b'0
            default
next 'b'0;
                  (MSB1 $ clkMSB1) #
                  (MSB2 $ clkMSB2) #
                  (MSB3 $ clkMSB3) ))
next 'b'1;
      present 'b'1
           default
next 'b'1;
            if(!clkLSB0 & !clkLSB1 & !clkLSB2 & !clkLSB3 &
              !clkMSB0 & !clkMSB1 & !clkMSB2 & !clkMSB3)
next. 'b'0:
Speed Decoder.pld
         speeder;
Partno
         GAL22V10;
         05/27/02;
Revision
         XX;
Designer
         XXXXX;
         XXXXX;
Company
Assembly XXXXX;
        XXXXX;
Location
Device
        q22v10;
/*
/*
                  WINDOW/UNKNOWN
```

```
/** Inputs **/
Pin 1
                                    /* this is the Strobe off the
          = clk;
bus, bus bit 0 */
Pin 2
             = reset;
Pin [4..10] = [bus1..bus7];
/** Outputs **/
Pin [14..17] = [LSBdutycycle0..LSBdutycycle3];
Pin [18..21] = [MSBdutycycle0..MSBdutycycle3];
/* intermediates */
MSBdutycycle3.ar = reset;
MSBdutycycle2.ar = reset;
MSBdutycycle1.ar = reset;
MSBdutycycle0.ar = reset;
LSBdutycycle3.ar = reset;
LSBdutycycle2.ar = reset;
LSBdutycycle1.ar = reset;
LSBdutycycle0.ar = reset;
MSBdutycycle3.sp = 'b'0;
MSBdutycycle2.sp = 'b'0;
MSBdutycycle1.sp = 'b'0;
MSBdutycycle0.sp = 'b'0;
LSBdutycycle3.sp = 'b'0;
LSBdutycycle2.sp = 'b'0;
LSBdutycycle1.sp = 'b'0;
LSBdutycycle0.sp = 'b'0;
MSBdutycycle3.oe = 'b'1;
MSBdutycycle2.oe = 'b'1;
MSBdutycycle1.oe = 'b'1;
MSBdutycycle0.oe = 'b'1;
LSBdutycycle3.oe = 'b'1;
LSBdutycycle2.oe = 'b'1;
LSBdutycycle1.oe = 'b'1;
LSBdutycycle0.oe = 'b'1;
speed frame = !bus6 & !bus7; /* Speed Frame, 00 */
{\tt MSBdutycycle3.d = (speed\_frame \& `b'0) \# (!speed\_frame \& MSBdutycy-}
MSBdutycycle2.d = (speed_frame & 'b'1) # (!speed_frame & MSBdutycy-
MSBdutycycle1.d = (speed frame & ( (bus5 & bus4 & (bus3 # bus2)))) #
(!speed frame & MSBdutycycle1);
MSBdutycycle0.d = (speed frame & ( (bus5 & (!bus4 \# (bus4 & !bus3 &
!bus2))))) # (!speed frame & MSBdutycycle0);
LSBdutycycle3.d = (speed\_frame \& ( (bus5 \& bus4 \& !bus3 \& !bus2))) \ \#
(!speed_frame & LSBdutycycle3);
LSBdutycycle2.d = (speed_frame & ( (bus3 & (!bus4 \# bus2)))) \#
(!speed_frame & LSBdutycycle2);
LSBdutycycle1.d = (speed_frame & ( ((!bus4 & bus2) \# (bus4 & bus3 &
!bus2)))) # (!speed_frame & LSBdutycycle1) ;
LSBdutycycle0.d = (speed_frame & ( (bus1))) # (!speed_frame & LSBdu-
tycycle0);
Steering Decoder.pld
Name
          steerer;
          GAL22V10;
Partno
          05/27/02;
Date
Revision XX;
Designer XXXXX;
        xxxxx:
Company
Assembly XXXXX;
```

```
Location XXXXX;
                                                                                                       (bus5 & !bus4 & bus2)
                                                                                                       (!bus5 & bus4 & bus3 & !bus2) #
Device q22v10;
                                                                                                       (!bus5 & !bus3 & (bus4 $ bus2))))
# (!steering frame & LSBdutycycle1);
                                                                                                             (bus1))) # (!steering
                    WINDOM/UNKNOWN
                                                                     LSBdutycycle0.d = (steering_frame & (
                                                                     frame & LSBdutycycle0);
/************************
                                                                     Steering Generator.pld
/** Inputs **/
                                                                               sterpw90;
Pin 1
            = clk;
                                 /* this is the Strobe off the
                                                                              GAL22V10;
                                                                     Partno
bus, bus bit 0 */
                                                                              05/27/02;
                                                                     Date
            = reset;
                                                                     Revision
                                                                              XX;
Pin 2
                                                                              XXXXX;
                                                                     Designer
Pin [4..10] = [bus1..bus7];
                                                                     Company
                                                                              XXXXX;
                                                                     Assembly XXXXX;
                                                                    Location XXXXX;
/** Outputs **/
                                                                              g22v10;
                                                                    Device
                                                                     Pin [14..17] = [LSBdutvcvcle0..LSBdutvcvcle3];
Pin [18..21] = [MSBdutycycle0..MSBdutycycle3];
                                                                    /*
                                                                                                                              * /
                                                                                         WINDOW/UNKNOWN
                                                                     MSBdutycycle3.ar = reset;
MSBdutycycle2.ar = reset;
                                                                    /** Inputs **/
MSBdutycycle1.ar = reset;
MSBdutycycle0.ar = reset;
                                                                    Pin 1
                                                                                  = clk;
LSBdutycycle3.ar = reset;
                                                                    Pin 2
                                                                                  = reset;
LSBdutycycle2.ar = reset;
LSBdutycycle1.ar = reset;
                                                                     Pin [3..6] = [MSB3..MSB0];
LSBdutycycle0.ar = reset;
                                                                    Pin [7..10] = [LSB3..LSB0];
                                                                    Pin [13..16] = [clkMSB0..clkMSB3];
MSBdutycycle3.sp = 'b'0;
                                                                    Pin [20..23] = [clkLSB0..clkLSB3];
MSBdutycycle2.sp = 'b'0;
MSBdutycycle1.sp = 'b'0;
                                                                     /** Outputs **/
MSBdutycycle0.sp = 'b'0;
                                                                    Pin 18
                                                                                 = !a0;
                                                                                               /* this is the PWM output to the servos
                                                                    Pin 19
LSBdutvcvcle3.sp = 'b'0;
                                                                                 = temp;
LSBdutycycle2.sp = 'b'0;
LSBdutycycle1.sp = 'b'0;
                                                                    FIELD pwm = [a0];
LSBdutycycle0.sp = 'b'0;
                                                                    /** Declarations and Intermediate Variable Definitions **/
MSBdutycycle3.oe = 'b'1;
MSBdutycycle2.oe = 'b'1;
                                                                    a0.ar
                                                                                  = reset;
MSBdutycycle1.oe = 'b'1;
                                                                    a0.sp
                                                                                  = 'b'0;
MSBdutycycle0.oe = 'b'1;
                                                                     a0.oe
                                                                                  = !reset;
LSBdutycycle3.oe = 'b'1;
LSBdutycycle2.oe = 'b'1;
LSBdutycycle1.oe = 'b'1;
LSBdutycycle0.oe = 'b'1;
                                                                                         (LSB0 $ clkLSB0) #
                                                                     temp =
                                                                                         (LSB1 $ clkLSB1) #
                                                                                         (LSB2 $ clkLSB2) #
steering frame = bus6 & !bus7; /* Directional Frame, 01 */
                                                                                         (LSB3 $ clkLSB3) #
                                                                                         (MSB0 $ clkMSB0);
MSBdutycycle3.d = (steering frame & 'b'0) # (!steering frame & MSBdu-
tycycle3);
                                                                     sequence pwm
MSBdutycycle2.d = (steering_frame & bus5) # (!steering frame & MSBdu-
                                                                           present 'b'0
tycycle2);
                                                                                  default
                                                                     next 'b'0;
{\tt MSBdutycycle1.d = (steering\_frame \& !bus5) \# (!steering frame \& {\tt MSB-}
                                                                                  if(!( temp
                                                                                         (MSB1 $ clkMSB1) #
dutycycle1);
                                                                                         (MSB2 $ clkMSB2) #
{\tt MSBdutycycle0.d} = ({\tt steering\_frame \& ((!bus5 \& (bus4 \# (bus3 \& bus2))))\#}
                                                                                         (MSB3 $ clkMSB3) ))
                                  (bus5 & bus4 & (bus3 # bus2)))) #
                                                                    next 'b'1;
(!steering_frame & MSBdutycycle0);
                                                                           present 'b'1
LSBdutycycle3.d = (steering_frame & ( (bus5 & bus4 & !bus3 & !bus2)
                                                                                  default
                                                                    next 'b'1;
                                  (!bus5 & !bus4 & bus3 & !bus2)
                                                                                  if(!clkLSB0 & !clkLSB1 & !clkLSB2 & !clkLSB3 &
                                                                                     !clkMSB0 & !clkMSB1 & !clkMSB2 & !clkMSB3)
                                  (!bus5 & bus4 & bus3 & bus2))) #
                                                                    next 'b'0;
(!steering frame & LSBdutycycle3);
LSBdutycycle2.d = (steering_frame & ((bus5 & bus4 & bus3 & bus2) #
                                  (bus5 & !bus4 & bus3)
                                 (!bus5 & !bus4 & !bus3)
                                  (!bus5 & bus4 & (bus3 $ bus2))))#
(!steering frame & LSBdutycycle2);
LSBdutycycle1.d = (steering_frame & ( (bus5 & bus4 & bus3 & !bus2) \ \ \#
                                                                    #define MAXGEAR 3
```

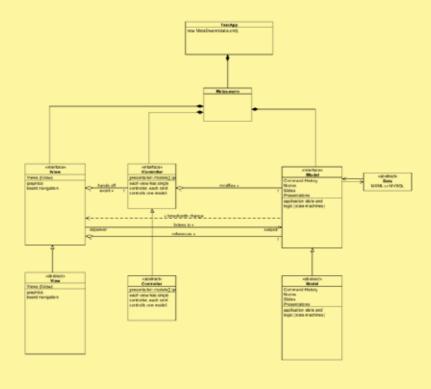
```
#define IDLE 16
                                                                                                     (CurrentGear > 0))
                                                                                                 AdcSpeedState = AdcGasState;
                                                                                             else if((AdcBrakeState < IDLE) && (CurrentGear == 0))
char TRISC@0x87;
char TRISD@0x88:
                                                                                                 AdcSpeedState = AdcBrakeState;
char TRISE@0x89;
                  // Button inputs from steering wheel
                                                                                                 AdcSpeedState = IDLE;
char PORTC@0x07;
char PORTD@0x08;
                   // Gear output display and PIC output enable
Tristate
                                                                                             PORTB = fOut(AdcSpeedState,iter);
char PORTE@0x09; // Enable signals to 512kx8 memory
char ADCON0@0x1f;
char ADCON1@0x9f;
char ADRESH@0x1e;
                                                                                         else if(iter == 1)
                                                                                             AdcWheelState = fAdc(0);
int AdcWheel, AdcGas, AdcBrake;
                                                                                             PORTB = fOut(AdcWheelState,iter);
char AdcWheelState,AdcGasState,AdcBrakeState,AdcSpeedState;
char CurrentGear, GearUp, GearDn;
char CurrentButtons, ButtonChange;
                                                                                         else
char MemWrite, MemRead, MemLast;
                                                                                             fButtons();
fAdc(char Channel);
                                                                                             PORTB = fOut(CurrentButtons,iter);
                                                                                             if(MemLast == 1)
                                                                                                                       // Send one last data frame
void fGear(char Current);
                                                                             with
                                                                                                 set bit(PORTE,1);
                                                                                                                      // a flag to determine end of
void fButtons (void);
                                                                             dat.a
                                                                                                 clear bit(PORTE,1);
fOut (char Data, char State);
fStrobe(void);
                                                                                         if((MemWrite == 1) \mid \mid (MemLast == 1) \mid \mid (MemRead == 1))
bit_set(char Data, char Index);
                                                                                             clear bit(PORTD,1);
                                                                                                                        // Send a Clock pulse to the
main()
                                                                             74LS93s
                                                                                             set bit(PORTD,1);
                                                                                                                        // for a new address
                                                                                             clear_bit(PORTD,0);
                                                                                                                        // Count Addresses on
                                                                             74LS93s
    TRISB = 0x00;
                        //Set PORTB to all output
    PORTB = 0x00;
    TRISD = 0x00;
                        //Set PORTD to all output
                                                                                         else
                                                                                             set_bit(PORTD,0);
    PORTD = 0x03;
                                                                                                                        // Reset Address on 74LS93s
                        //b00000011
    TRISE = 0 \times 00;
                        //Set PORTE to all output
                                                                             for new
    PORTE = 0 \times 05;
                        //b00000101
                                                                                                                        // sequence
    TRTSA = 0xff;
                        //Set PORTA to all input
                                                                                         delay_ms(50);
    TRTSC = 0xff:
                        //Set PORTC to all input
                                                                                                                        // Shortest amount of delay
                                                                             for RF
                                                                                                                        // Transmission
    ADCON0 = 0x81;
                        //Configure and turn on A/D Module
                        //Set PORTA to analog and RIGHT justify result
                                                                                          if((MemWrite == 1) || (MemLast == 1))
    ADCON1 = 0 \times 02;
                         //with PORTE on I/O
                                                                                             clear_bit(PORTE,0);
                                                                                                                        // Send a Write enable sig-
    CurrentGear = 1;
                                                                             nal to
                                                                                             set_bit(PORTE,0);
    GearUp = 0;
                                                                                                                        // the 512kx8
    GearDn = 0;
    MemRead = 0;
    MemWrite = 0;
                                                                                         fStrobe();
                                                                                                                        // Data is valid, send to
    MemLast = 0;
                                                                             car
    ButtonChange = 0;
    CurrentButtons = 0;
    int iter;
    delay_ms(2);
    while(1)
        iter = 0;
                                                                             fAdc(char Channel)
        for(iter;iter<3;iter++)</pre>
                                                                                 char Adc0 = 0;
            fGear(PORTC);
                                                                                 char Adc1 = 0;
                                                                                 char Adc2 = 0;
            if(iter == 0)
                                                                                 switch (Channel)
                                                                                                            //Select which ADC channel
                AdcGasState = fAdc(1);
                AdcBrakeState = fAdc(2);
                                                                                     case 0:
                                                                                         ADCON0 = 0x81;
                                                                                                           //ADC 1 Enable AtoD
                                                                                        break;
                if((AdcBrakeState < IDLE) && (CurrentGear > 0))
                                                                                     case 1:
                                                                                        ADCON0 = 0x89;
                                                                                                           //ADC 2 Enable AtoD
                    AdcSpeedState = IDLE;
                                                                                        break;
                    CurrentGear = 1;
                                                                                     case 2:
                                                                                                          //ADC 3 Enable AtoD
                                                                                         ADCON0 = 0 \times 91;
                else if((AdcGasState > IDLE) && (AdcBrakeState ==
                                                                                        break:
                                                                                     default:
JDLE) &&
```

```
ADCON0 = 0x81; //ADC 1 Enable AtoD
                                                                                   CurrentGear = CurrentGear + 1;
           break;
                                                                                  GearUp = 1;
                                                                              else if((bit_set(Current,4) == 1) && (GearUp == 1))
    delay_us(100);
                                                                                  GearUp = 0;
    set_bit(ADCON0,2);
                                                                              if(((bit_set(Current,5) != 1) && (CurrentGear > 0)) && (GearDn ==
    while(bit_set(ADCON0,2) == 1);
                                                                                   CurrentGear = CurrentGear - 1;
                                                                                   GearDn = 1;
    switch(Channel)
                                                                              else if((bit_set(Current,5) == 1) && (GearDn == 1))
        case 0: // Calculate ADC for Wheel position
                                                                                  GearDn = 0;
            Adc0 = ADRESH;
                                                                               if(CurrentGear == 0)
                                                                                                          //Use upper 4 bits of PORTD
            if(Adc0 >= 96)
                                                                                   set bit(PORTD,7);
                                                                                                         // 1100XXXXb Output a 12 for Reverse
                                                                                   set bit (PORTD, 6);
               return 31;
                                                                                  clear_bit(PORTD,5);
            else if((Adc0 >= 68) && (Adc0 < 96))
               return ((Adc0 - 34)/2);
                                                                                  clear_bit(PORTD,4);
            else if((Adc0 >= 60) && (Adc0 < 68))
               return IDLE;
                                                                              else if(CurrentGear == 1)
            else if((Adc0 \geq= 12) && (Adc0 < 60))
               return ((Adc0 - 12)/3);
                                                                                   clear_bit(PORTD,7);
                                                                                                         // 0001XXXXb Output a 1 for 1st
            else if (Adc0 < 12)
                                                                                  clear_bit(PORTD,6);
               return 0;
                                                                                  clear_bit(PORTD,5);
            else
                                                                                   set_bit(PORTD,4);
               return IDLE;
            break;
                                                                               else if(CurrentGear == 2)
        case 1: // Calculate ADC for Gas pedal position
                                                                                   clear_bit(PORTD,7); // 0010XXXXb Output a 2 for 2nd
                                                                                  clear bit(PORTD, 6);
            Adc1 = ADRESH;
                                                                                   set bit(PORTD,5);
                                                                                  clear_bit(PORTD, 4);
            if(CurrentGear > 0)
                                                                              else if(CurrentGear == 3)
               if((Adc1 >= 220) && (CurrentGear <= 1)) // Max 224
Min 181
                                                                                  clear bit(PORTD,7); // 0011XXXXb Output a 3 for 3rd
                   return (IDLE + (4 * (CurrentGear - 1)));
                                                                                  clear bit (PORTD, 6);
                                                                                   set bit (PORTD, 5);
               else if((Adc1 \geq= 220) && (CurrentGear \geq 1))
                   return (17 + (4 * (CurrentGear - 1)));
                                                                                  set bit(PORTD,4);
               else if((Adc1 >= 208) && (Adc1 < 220))
                   return (18 + (4 * (CurrentGear - 1)));
                                                                              else
               else if((Adc1 >= 195) && (Adc1 < 208))
                   return (19 + (4 * (CurrentGear - 1)));
                                                                                   clear_bit(PORTD,7);
                                                                                                         // 0001XXXXb Output a 1 for 1st
                                                                                  clear_bit(PORTD,6);
                    return (20 + (4 * (CurrentGear - 1)));
                                                                                  clear bit(PORTD,5);
                                                                                   set_bit(PORTD,4);
            else
               return IDLE;
        case 2: // Calculate ADC for Brake pedal position
                                                                           void fButtons(void)
            Adc2 = ADRESH;
            if(Adc2 < 110) // Up 80 Dn 200
                                                                              CurrentButtons = ~PORTC;
                                                                              clear bit(CurrentButtons, 4);
               return IDLE;
            else if((Adc2 >= 110) && (Adc2 < 155))
                                                                              MemLast = 0;
               return 6;
            else if((Adc2 >= 155) && (Adc2 < 175))
                                                                              if (MemRead == 0) // WRITE TO MEMORY
               return 5;
            else if(Adc2 >= 175)
               return 4;
                                                                                  if((bit\_set(CurrentButtons, 2) == 1) && (ButtonChange == 0))
            else
               return IDLE;
                                                                                      ButtonChange = 1;
           break;
                                                                                       MemWrite = 1;
                                                                                       set bit(PORTD,3);
        default:
                                                                                   else if((bit_set(CurrentButtons,2) == 0) && (ButtonChange ==
            return IDLE;
                                                                          1))
            break;
                                                                                   else if((bit_set(CurrentButtons,2) == 1) && (ButtonChange ==
                                                                          2))
                                                                                       ButtonChange = 3;
                                                                                  else if((bit set(CurrentButtons,2) == 0) && (ButtonChange ==
                                                                           3))
                                                                                      ButtonChange = 0;
                                                                                      MemLast = 1;
void fGear(char Current) // Calculates which gear the car should be
                           // in according to paddle pushes
                                                                                       MemWrite = 0;
                                                                                       clear bit(PORTD,3);
   if(((bit_set(Current,4) != 1) && (CurrentGear < MAXGEAR)) && (Gea-</pre>
rUp == 0))
```

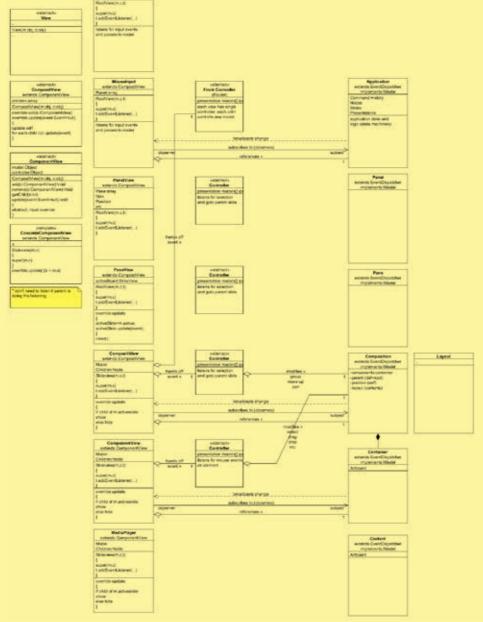
```
if((MemWrite == 0) && (MemLast == 0)) // READ FROM MEMORY
       if((bit set(CurrentButtons,3) == 1) && (ButtonChange == 0))
           set_bit(PORTD,2);
                                    // Disable PIC data output
           delay_us(25);
           ButtonChange = 1;
           MemRead = 1;
           clear_bit(PORTE,2);
                                    // Enable Mem data output
       else if((bit_set(CurrentButtons,3) == 0) && (ButtonChange == 1))
           ButtonChange = 2;
       else if((bit set(CurrentButtons,3) == 1) && (ButtonChange == 2))
           ButtonChange = 3;
       else if((bit_set(CurrentButtons,3) == 0) && (ButtonChange == 3))
           set bit(PORTE,2);
                                    // Disable Mem data output
           delay_us(25);
           ButtonChange = 0;
           MemRead = 0;
           clear_bit(PORTD,2);
                                    // Enable PIC data output
       if((MemRead == 1) && (bit set(CurrentButtons,7) == 0))
           ButtonChange = 3;
                                    // Reached end of memory
           set_bit(PORTE,2);
                                    // return to User control
fOut(char Data, char State)
                            //Formats the Data char to be ready
                             //for output by placing an address
    char tempData = Data << 1; //at the MSB and shifting left 1 to
    switch(State)
       case 0:
           clear_bit(tempData,7);
           clear_bit(tempData,6);
           return tempData;
           break;
           clear_bit(tempData,7);
           set_bit(tempData,6);
           return tempData;
           break;
           set_bit(tempData,7);
           clear bit(tempData,6);
           return tempData;
           break;
fStrobe(void)
    set bit(PORTB,0);
    clear_bit(PORTB,0);
bit_set(char Data, char Bit) //Checks to see if the selected
                            //bit in a char is high or low
    switch(Bit)
       case 0:
           if((Data % 2) == 1)
              return 1;
           else
              return 0;
           break;
       case 1:
           if(((Data % 4) / 2) == 1)
```

META-SWARM, SOURCECODE

WORKING PROTOTYPE SOURCE CODE FOR METASWARM VISUAL COMMUNICATION PLATFORM. INITIALLY DEVELOPED UI IN JAVA, THEN REBUILT IN ACTIONSCRIPT 3. IMPLEMENTED WITH VARIENT OF MODEL-VIEW-CONTROLLER DESIGN PATTERN: VIEW CONTAINS GRAPHIC ELEMENTS. EACH VIEW HAS A SINGLE CONTROLLER; EACH CONTROLLER CONTROLS ONE MODEL. MODEL HOLDS APPLICATION STATE AND LOGIC (STATE MACHINES). FLOW IS AS FOLLOWS: VIEWS HAND UI EVENTS TO CONTROLLER, WHICH MODIFIES MODEL. VIEW LISTENS FOR MODEL UPDATES AND UPDATES UI IN TURN.



47



```
public function set router(value:Router):void
package {
    import flash.display.DisplayObject;
    import flash.display.Graphics;
                                                                                                  router = value;
    import flash.display.Shape;
       import flash.display.Sprite;
                                                                                          public function get builder():Builder
       import flash.events.Event;
       import flash.events.KeyboardEvent;
       import flash.events.MouseEvent;
                                                                                                  return _builder;
       import flash.ui.Keyboard;
       import flash.net.*;
       import flash.utils.*;
                                                                                          public function set builder(value:Builder):void
       import flash.text.TextField;
       import metaswarm.ui.hoverlabel.HoverLabel;
       import metaswarm.ui.sidebar.Sidebar;
       import metaswarm.swarm.Swarm;
       import metaswarm.build.Builder;
                                                                                          public function get preloader():Preloader
       import metaswarm.control.Controller;
       import metaswarm.build.Preloader;
                                                                                                  return preloader;
       import metaswarm.control.Router;
       import metaswarm.ui.Ui;
                                                                                          public function set preloader(value:Preloader):void
       [SWF(width='1920',height='1200',backgroundColor='\#F7F7F7',frame
Rate='25')]
                                                                                                  preloader = value;
    public class Metaswarm extends Sprite
                                                                                          public function get ui():Ui
               //declarations
              private var _preloader:Preloader;
                                                                                                  return _ui;
               private var _builder:Builder;
              private var _router:Router;
              private var _controller:Controller;
                                                                                          public function set ui(value:Ui):void
              private var _ui:Ui;
                                                                                                  ui = value;
       public function Metaswarm():void
                      ui = new Ui();
                      builder = new Builder(this);
                                                                               }//end class
                      preloader = new Preloader(builder);
                                                                           }//end pkg
                                                                           ***********
                      controller = new Controller(ui); //removed hov-
erLabel param
                                                                           package
                      router = new Router(controller);
                                                                                  import flash.display.DisplayObject;
                                                                                   import flash.display.Sprite;
                                                                                   import metaswarm.node.components.dot.Dot;
                      init():
                                                                                   import metaswarm.node.components.icon.Icon;
                                                                                   import metaswarm.node.nodestates.DraggingState;
              private function init():void
                                                                                   import metaswarm.node.nodestates.DroppedState;
                                                                                   import metaswarm.node.nodestates.IdleState;
                      //this.stage.quality = "low";
                                                                                   import metaswarm.node.nodestates.PickedState;
                                                                                   import metaswarm.node.nodestates.PoppedState;
                      //register ui listeners
                                                                                   import metaswarm.node.nodestates.ThumbDraggingState;
                      addEventListener(Event.ENTER_FRAME,
                                                                                   import metaswarm.node.components.databox.DataBox;
                              router.onEnterFrame);
                                                                                   import metaswarm.node.components.tag.Tag;
                      stage.addEventListener(KeyboardEvent.KEY_DOWN,
                                                                                   import metaswarm.node.components.thumb.Thumb;
                             router.onkeyDown);
                                                                                   import metaswarm.node.NodeState;
                      stage.addEventListener(KeyboardEvent.KEY UP,
                             router.onkeyUp);
                                                                                  public class Node extends Sprite{
                      addEventListener (MouseEvent.CLICK,
                             router.onMouseClick);
                                                                                          //omit constants for now
                      addEventListener (MouseEvent.MOUSE DOWN,
                                                                                          //var nodeLoader:NodeLoader;
                             router.onMouseDown);
                      addEventListener (MouseEvent.MOUSE UP,
                                                                                          public var dat:Array; //again using key word
                             router.onMouseUp);
                                                                                          //var dat:Array;
                      addEventListener(MouseEvent.MOUSE OVER,
                             router.onMouseOver);
                                                                                          public var xpos:Array;
                      addEventListener(MouseEvent.MOUSE OUT,
                                                                                          public var ypos:Array;
                              router.onMouseOut);
                                                                                          //var xtmp:Array;
                                                                                          //var ytmp:Array;
                      //add ui stage
                                                                                          //internal composite objects
                      addChild(ui);
                                                                                          public var thumb:Thumb;
                                                                                          public var icon:Icon;
               \verb"public function get controller"(): Controller"
                                                                                          public var dot:Dot;
                                                                                          public var tag:Tag;
                      return _controller;
                                                                                          public var dataBox:DataBox;
                                                                                          //var img:Image;
              public function set controller(value:Controller):void
                                                                                          //internal states
                                                                                          public var idle:NodeState;
                      controller = value;
                                                                                          public var picked:NodeState;
                                                                                          public var dragging:NodeState;
                                                                                          public var dropped:NodeState;
                                                                                          public var tdragging:NodeState;
               public function get router():Router
                                                                                          public var popped:NodeState;
                      return _router;
                                                                                          //internal state holder
                                                                                          public var nodeState:NodeState;
```

	_	<pre>var prevState:NodeState; ructor:</pre>		public	<pre>function isPopped():Boolean{ if(nodeState == popped){</pre>
	, ,	function Node(){			return true; }else{
		//rediculous but don't know how else to initial-			return false;
ize an exact-s:	ized 2d			}	1
- 0		<pre>xpos = new Array(new Array(), new Array(), new</pre>		4.5	
ray(), new Arra		new Array(), new Array(), new Array(), new Ar-		/*funct	ion loadData(dat:Array):void{ tag.dat = dat;
		<pre>ypos = new Array(new Array(), new Array(), new new Array(), new Array(), new Array(), new Ar-</pre>			<pre>meta.dat = dat; meta.format();</pre>
ray(), new Arra	_	new Milay(), new Milay(), new Milay(), new Mi			<pre>var path:String = String(dat[0] + dat[1]);</pre>
		<pre>//nodeLoader = new NodeLoader(this);</pre>		}	<pre>//img.loadImg(path);</pre>
		idle - new Talle Charte (Fbie).			an localTrans()
		<pre>idle = new IdleState(this); picked = new PickedState(this);</pre>		Tunctio	on loadImg():void{ var path:String = String(meta.dat[0] + meta.
		<pre>dragging = new DraggingState(this);</pre>	dat[1]);		inn leadTun(math).
		<pre>tdragging = new ThumbDraggingState(this); dropped = new DroppedState(this);</pre>		} * /	<pre>img.loadImg(path);</pre>
		<pre>popped = new PoppedState(this);</pre>			5 1
		<pre>dot = new Dot();</pre>		public	<pre>function select():void{ //trace("node delegate select");</pre>
		<pre>icon = new Icon();</pre>			<pre>nodeState.select();</pre>
		<pre>thumb = new Thumb(); tag = new Tag();</pre>		}	
		<pre>dataBox = new DataBox();</pre>		public	<pre>function deSelect():void{</pre>
		<pre>//img = new Image();</pre>		}	<pre>nodeState.deSelect();</pre>
		<pre>init();</pre>			
	} public	<pre>function init():void{</pre>		public	<pre>function toggleSelect():void{ nodeState.toggleSelect();</pre>
	-	<pre>nodeState = idle;</pre>		} public	
	} /*funct	ion load():void{			<pre>function drag():void{</pre>
		//trace("node.load()");			<pre>//trace("node delegate drag");</pre>
} func	}	<pre>nodeLoader.load();</pre>		}	<pre>nodeState.drag();</pre>
	functio	n make():void{			
		<pre>//trace("node.make()"); nodeLoader.make();</pre>		public	<pre>function drop():void{ //trace("node delegate drop");</pre>
	}				<pre>nodeState.drop();</pre>
	functio	<pre>n go():void{ //trace("node.go()");</pre>		}	
	/	<pre>nodeLoader.go();</pre>		_	ge to setTarget
	}*/			public	<pre>function setTarget(pid:uint, sid:uint):void{ //trace("node set Target");</pre>
		d this be delegated???no!	المائماً .		<pre>nodeState.setTarget(xpos[pid][sid], ypos[pid]</pre>
	public	<pre>function popout():void{ nodeState.popout();</pre>	[sid]);	}	
	}			nuhlia	<pre>function rePosition():void{</pre>
	public	function popin():void{		public	<pre>//trace("node delegate rePosition");</pre>
	}	<pre>nodeState.popin();</pre>		ı	<pre>nodeState.rePosition();</pre>
	1			1	
	public	<pre>function close():void{ nodeState.close();</pre>		public	<pre>function reSize():void{ //trace("node delegate reSize");</pre>
	}				nodeState.reSize();
	public	<pre>function isIdle():Boolean{</pre>		}	
	_	<pre>//trace("isidle?");</pre>		public	<pre>function startFocus():void{</pre>
		<pre>if(nodeState == idle) { return true;</pre>			<pre>//trace(""); nodeState.startFocus();</pre>
		<pre>}else{ return false;</pre>		}	
		}		public	<pre>function stopFocus():void{</pre>
	}				<pre>//trace(""); nodeState.stopFocus();</pre>
	public	<pre>function isPicked():Boolean{</pre>		}	
		<pre>if(nodeState == picked) { return true;</pre>		public	<pre>function startHover():void{</pre>
		}else{		1	//trace("");
		<pre>return false; }</pre>		}	<pre>nodeState.startHover();</pre>
	}				Emplies starts and all M
	public	<pre>function isDropped():Boolean{</pre>		public	<pre>function stopHover():void{ //trace("");</pre>
		<pre>if(nodeState == dropped){</pre>		1	<pre>nodeState.stopHover();</pre>
		return true; }else{		1	
		return false;		public	<pre>function incrTag():void{ tag.incr();</pre>
		I .			cay.inci(),

```
var loader:URLLoader = new URLLoader();
              public function decrTag():void{
                      tag.decr();
                                                                                          // Instruct the loader to read the file as plain text -
                                                                           This line is not
                                                                                          // necessary because the dataFormat is DataFormat.TEXT
                                                                           by default.
                                                                                          loader.dataFormat = URLLoaderDataFormat.TEXT;
*************
package metaswarm.build {
                                                                                          // Register an event handler for when the data is fin-
    import flash.display.*;
                                                                           ished downloading
       import flash.events.*;
                                                                                          loader.addEventListener(Event.COMPLETE, loadComplete);
       import flash.ui.Keyboard;
       import flash.net.*;
                                                                                          // Listen for the progress event to check download
       import flash.utils.*;
                                                                           progress
                                                                                  loader.addEventListener(ProgressEvent.PROGRESS, han-
       import flash.text.*;
       import metaswarm.ui.hoverlabel.LabelMaker;
                                                                           dleProgress);
       import metaswarm.ui.sidebar.SidebarMaker;
       import metaswarm.swarm.patternmaker.PatternMaker;
                                                                                          // Load the HTML text from the example.html file
       import metaswarm.swarm.SwarmLoader;
                                                                                          loader.load(new URLRequest("index7.txt"));
       import metaswarm.ui.UiLoader;
       import metaswarm.*;
                                                                                   private function handleProgress(event:ProgressEvent):void
       //import fl.controls.Label;
                                                                                   // Calculate the percentage by multiplying the loaded-to-total \,
    public class Builder extends Sprite
                                                                                   // ratio by 100
                                                                                   var percent:Number = Math.round(event.bytesLoaded
                                                                                                                 / event.bytesTotal * 100 );
              private var _gnurbwurk:Metaswarm;
              private var _uiLoader:UiLoader;
                                                                                   _loadProg.text = " Loaded: " + event.bytesLoaded + "\n"
                                                                                                   + " Total: " + event.bytesTotal + "\n"
       public function Builder(_gnurbwurk:Metaswarm):void
                                                                                                   + "Percent: " + percent;
                      this._gnurbwurk = _gnurbwurk;
                      this._uiLoader = new UiLoader(_gnurbwurk.ui);
                                                                                          //trace(loadProg.text);
              private function init():void
                                                                               private function loadComplete(event:Event):void
                                                                                          var loader:URLLoader = URLLoader(event.target);
              public function load(dat:Array):void
                                                                                          // assign to output. The data property of the URLLoader
                                                                           is the file contents.
                      //load parsed data into objects
                                                                                          parseData(loader.data);
                      _uiLoader.load(dat);
              public function make():void
                                                                                   private function parseData(loadedData:String):void
                      //make views
                                                                                          var parsedData:Array = new Array();
                      _uiLoader.make();
                                                                                          //var parsedMenu:Array = new Array();
               public function go():void
                                                                                          \ensuremath{//} Split the string into an array of words using a
                                                                           space as the delimiter.
                      //add grandchildren to display
                                                                                          var lines:Array = loadedData.split("\n");
                                                                                          var words:Array = new Array();
                      _uiLoader.go();
                                                                                          // Loop through the array and do something with each
    }//end class
                                                                           word.
}//end pkg
                                                                                          \ensuremath{//} In this example, just output the values.
package metaswarm.build {
                                                                                          for ( var i:int = 0; i < lines.length; i++ ) {
                                                                                                 words = lines[i].split("\t");
  import flash.display.*;
  import flash.text.*;
                                                                                                  parsedData.push(words);
                                                                                                  //trace(parsedData[i][1]);
  import flash.events.*
  import flash.net.*;
                                                                                          //gnurbwurk.parsedData = parsedData;
  public class DataLoader extends Sprite {
                                                                                          _preloader.preloadComplete(parsedData);
                                                                                          //gnurbwurk.main.load(parsedData);
       private var _loadProg:TextField;
                                                                                          //gnurbwurk.main.make();
       private var _preloader:Preloader;
                                                                                          //gnurbwurk.main.go();
                                                                                          //gnurbwurk.main.makePatterns();
    public function DataLoader(preloader:Preloader)
              this._preloader = preloader;
                                                                             }//end class
              this._loadProg = new TextField();
                                                                           }//end pkg
                                                                           package metaswarm.build {
                                                                               import flash.display.DisplayObject;
               init();
                                                                               import flash.display.Graphics;
                                                                               import flash.display.Shape;
                                                                                   import flash.display.Sprite;
    private function init():void
                                                                                   import flash.events.Event;
       //loadProg.width = stage.stageWidth;
                                                                                   import flash.events.KeyboardEvent;
       //loadProg.height = stage.stageHeight;
                                                                                   import flash.events.MouseEvent;
       addChild(_loadProg);
                                                                                   import flash.ui.Keyboard;
                                                                                   import flash.net.*;
                                                                                   import flash.utils.*;
       public function preload():void {
```

public class Preloader extends Sprite

50

{				{	
		<pre>var _dataLoader:DataLoader;</pre>			<pre>main.uiState.onCtrlClick(node, button);</pre>
		ain:Main;		}	5
	private	<pre>var _builder:Builder;</pre>	h++ on . Ob - o a+ \	-	function onShftClick(node:Node,
nuhlic	functio	n Preloader(builder:Builder):void{	button:Object)	. vora	
public	Idiiocio	this. builder = builder;		· ·	<pre>main.uiState.onShftClick(node, button);</pre>
		this. dataLoader = new DataLoader(this);		}	, , , , , , , , , , , , , , , , , , , ,
		<pre>init();</pre>		public	function onAltClick(node:Node,
	}		button:Object)	:void	
				{	
		external data			//can't use alt b/c window uses it to for
	private	function init():void{	toolbar	1	
	ı	_dataLoader.preload();) nublic	function onCtrlShftClick(node:Node,
	J		button:Object)	-	Tunction onecrionicories (node, node,
	public	function preloadComplete(parsedData:Array):void{	,	{	
	1	<pre>//trace("gw.preloadComplete");</pre>		•	main.uiState.onCtrlShftClick(node, button
		_builder.load(parsedData);		}	
		_builder.make();			<pre>function onMouseClick(node:Node,</pre>
		_builder.go();	button:Object)	:void	
	}			{	'
}//end cla				ı	<pre>main.uiState.onMouseClick(node, button);</pre>
//end pkg	33			ı	
package metasw	arm.cont	rol {		//publi	ic function onDataBoxClick(node:Node,
_	flash.dis		button:Object)		, , , , , , , , , , , , , , , , , , , ,
import	metaswa	rm.ui.hoverlabel.HoverLabel;		public	function onMouseDown(node:Node,
import	metaswa	m.swarm.Swarm;	button:Object)	:void	
import	metaswa	rm.ui.Ui;		{	
import	metaswa	rm.ui.UiState;			<pre>main.uiState.onMouseDown(node, button);</pre>
1.1	1 0			}	5
public	class Co	ontroller extends Sprite{		public	function onMouseUp(node:Node):void{
	//nri 172	te var _uiState:UiState;		ı	<pre>main.uiState.onMouseUp(node);</pre>
		var main:Ui;		ı	
	F			//over	
	public	function Controller(main:Ui){		public	function onMouseOver(node:Node,
		<pre>//_uiState = main.uiState;</pre>	button:Object)	:void{	
		<pre>this.main = main;</pre>			<pre>main.uiState.onMouseOver(node, button);</pre>
				}	
	1	<pre>init();</pre>	h		function onCtrlOver(node:Node,
) nrivate	<pre>function init():void{</pre>	button:Object)	: ٧٥١α (<pre>main.uiState.onCtrlOver(node, button);</pre>
	private	//		}	main.uistate.onctitover(node, button),
	}			public	<pre>function onShftOver(node:Node,</pre>
			button:Object)		,
	//deleg	ated meth-			<pre>main.uiState.onShftOver(node, button);</pre>
ds********	*****	******		}	
	public	function onEnterFrame():void{			function onCtrlShftOver(node:Node,
	1	<pre>main.uiState.onEnterFrame();</pre>	button:Object)	:vold{	'
	} nublia	function onRightKey():void{		1	<pre>main.uiState.onCtrlShftOver(node, button)</pre>
	public	main.uiState.onRightKey();		ſ	
	}	main alboado formignonoj (//		//out	
	public	function onLeftKey():void{		public	<pre>function onMouseOut(node:Node):void{</pre>
		<pre>main.uiState.onLeftKey();</pre>			<pre>main.uiState.onMouseOut(node);</pre>
	}			}	
	public	function onUpKey():void{		public	<pre>function onCtrlOut():void{</pre>
	1	<pre>main.uiState.onUpKey();</pre>		1	<pre>main.uiState.onCtrlOut();</pre>
	} nublia	function on Down More () erroid (} nuhlia	function on Chft Out () evoid
	PUDITC	<pre>function onDownKey():void{ main.uiState.onDownKey();</pre>		PUDITC	<pre>function onShftOut():void{ main.uiState.onShftOut();</pre>
	}	main. albeate. on bownie y (//		}	main: albeate. Onone code (//
	public	function onSpaceKey():void{		public	<pre>function onCtrlShftOut():void{</pre>
	_	<pre>main.uiState.onSpaceKey();</pre>		-	//
	}			}	
	public	function onKey1():void{			
		<pre>trace("Controller -> onKey1");</pre>		-	
	1	<pre>main.uiState.onKey1();</pre>	}//end	class	
) nublic	function onKey2():void{	<pre>}//end pkg package metaswa</pre>	arm cont	rol (
	PUDITC	trace("Controller -> onKey2");	import flas		
		main.uiState.onKey2();	=	flash.ev	_
	}	<u>-</u>	_		.Keyboard;
	public	function onKey3():void{		flash.ne	
		<pre>main.uiState.onKey3();</pre>	=	flash.ut	
	}		import	flash.te	xt.*;
	public	function onKey4():void{	//:	. . - 1	trala Jahal.
	ì	<pre>main.uiState.onKey4();</pre>	//impor	L II.CON	trols.Label;
	niiplic 1	function onKey5():void{	nublic cla	ss Route	er extends Sprite
	Pantic	//swarm.showUniqueTags();	fantic cta	oo noute	. Chochas opitic
	}	1		private	e var controller:Controller;
	public	function onCtrlClick(node:Node,			dropped:Subset;
outton:Object)	:void			//var p	picked:Subset;

```
//var idle:Subset;
                                                                           Pressed) {
                                                                                                                        controller.onCtrlShft-
              //these should be scoped to methods
                                                                           Over(_focusedNode, _focusedButton);
              private var _ctrlPressed:Boolean;
                                                                                                                }else{
              private var _shftPressed:Boolean;
                                                                                                                       //trace("neither ctrl nor
              private var _altPressed:Boolean;
                                                                           shift was pressed and yet here we are");
              private var _mousedOver:Boolean;
              private var _focusedNode:Node;
                                                                                                 }else{
              private var _focusedButton:Object;
                                                                                                        //trace("you pressed an invalid key: " +
                                                                           event.keyCode);
       public function Router(controller:Controller):void
                      this._controller = controller;
                                                                                          public function onkeyUp(event:KeyboardEvent):void
                      ctrlPressed = false;
                       shftPressed = false;
                                                                                                 //reset ctrl and shft key states
                      altPressed = false;
                                                                                                 ctrlPressed = event.ctrlKey;
                      mousedOver = false;
                                                                                                 shftPressed = event.shiftKey;
                       focusedNode = new Node();
                                                                                                 //retest for focus conditions
                       _focusedButton = new Object();
                                                                                                 if(_mousedOver){
                      init();
                                                                                                        //just clear for now
                                                                                                         _controller.onShftOut();
              private function init():void{
                      //can't access this.stage until main is added to
                                                                                                        if(!_shftPressed && _ctrlPressed){
stage
                                                                                                                _controller.onCtrlOver(_focused-
                      //addEventListener(Event.ADDED_TO_STAGE, onAd-
                                                                           Node, _focusedButton);
dedToStage);
                                                                                                         }else if(_shftPressed && !_ctrlPressed){
                                                                                                                _controller.onShftOver(_focused-
                                                                           Node, _focusedButton);
              //delegated methods ***********************
                                                                                                         }else if(_shftPressed && _ctrlPressed){
******
                                                                                                                controller.onCtrlShftOver( fo-
              public function onEnterFrame(event:Event):void{
                                                                           cusedNode, focusedButton);
                       _controller.onEnterFrame();
                                                                                                         }else{
                      //cmdTag.x = mouseX;
                                                                                                                //trace("neither ctrl nor shift
                      //cmdTag.y = mouseY;
                                                                           was pressed and yet here we are");
                                                                                                                controller.onMouseOver( focused-
                                                                           Node, focusedButton);
              public function onkeyDown(event:KeyboardEvent):void
                      if (event.keyCode == Keyboard.RIGHT) {
                              _controller.onRightKey();
                      }else if (event.keyCode == Keyboard.LEFT) {
                                                                                          //maybe get rid of this since it competes with mouseUp
                              _controller.onLeftKey();
                                                                                          public function onMouseClick(event:MouseEvent):void{
                      }else if (event.keyCode == Keyboard.UP) {
                                                                                                 //trace(getQualifiedClassName(event.target));
                              _controller.onUpKey();
                                                                                                 var grandparentClass:String =
                      }else if (event.keyCode == Keyboard.DOWN) {
                                                                           getQualifiedClassName(event.target.parent.parent);
                              _controller.onDownKey();
                      }else if (event.keyCode == Keyboard.SPACE) {
                                                                                                 if(grandparentClass == "Node"){
                              _controller.onSpaceKey();
                                                                                                        if(_ctrlPressed && _shftPressed){
                      }else if (event.keyCode == 49) {
                                                                                                                _controller.
                             trace("pressed 1");
                                                                           onCtrlShftClick(event.target.parent.parent, event.target);
                                                                                                         }else if( ctrlPressed == true) {
                              controller.onKey1();
                      }else if (event.keyCode == 50) {
                                                                                                                controller.onCtrlClick(event.
                             trace("pressed 2");
                                                                           target.parent.parent, event.target);
                              controller.onKey2();
                                                                                                         }else if( shftPressed == true) {
                      }else if (event.keyCode == 51) {
                                                                                                                 controller.onShftClick(event.
                                                                           target.parent.parent, event.target);
                             trace("pressed 3");
                              controller.onKey3();
                                                                                                        }else{
                      }else if (event.keyCode == 52) {
                                                                                                                controller.onMouseClick(event.
                             trace("pressed 4");
                                                                           target.parent.parent, event.target);
                              controller.onKey4();
                      }else if (event.keyCode == 53) {
                                                                                                 }else{
                             trace("pressed 5");
                                                                                                        //trace("uh oh, grandparent is not a
                              _controller.onKey5();
                                                                           Node but a: " + grandparentClass);
                      }else if (event.ctrlKey == true || event.shift-
Key == true) {
                              _ctrlPressed = event.ctrlKey;
                                                                                          public function onMouseDown(event:MouseEvent):void
                              _shftPressed = event.shiftKey;
                                                                                                 //var parentClass:String =
                              if(_mousedOver){
                                                                           getQualifiedClassName(event.target.parent);
                                     //just clear for now
                                                                                                var grandparentClass:String =
                                     controller.onShftOut();
                                                                           getQualifiedClassName(event.target.parent.parent);
                                     if(! shftPressed && ctrlPressed)
                                                                                                 if(!_ctrlPressed && !_shftPressed && !_alt-
                                                                           Pressed && grandparentClass == "Node") {
                                                                                                         _controller.onMouseDown(event.target.
                                            _controller.onCtrlOver(_
focusedNode, _focusedButton);
                                                                           parent.parent, event.target);
                                     }else if(_shftPressed && !_ctrl-
                                                                                                 }else{
                                                                                                         //break if ctrl, shft, or alt is pressed
Pressed) {
                                            _controller.onShftOver(_
                                                                                                         //trace("don't drag");
focusedNode, _focusedButton);
                                     }else if(_shftPressed && _ctrl-
```

			"rotati	on".	
pub.	lic functio	on onMouseUp(event:MouseEvent):void	1		override public function getAffectedProperties():Array
1		parentClass:String =	ι		<pre>return ["rotation"];</pre>
etQualifiedClassNa		arget.parent); andparentClass:String =			}
etQualifiedClassNa	=	arget.parent.parent);			<pre>// Override initInstance() method.</pre>
	1.5.41				override protected function initInstance(inst:IEffectI
ressed && grandpa	_	trlPressed && !_shftPressed && !_alt- == "Node"){	stance)	:vold {	<pre>super.initInstance(inst);</pre>
rebbea aa granapa	rencorabb	_controller.onMouseUp(event.target.par-			//RotationInstance(inst).angleFrom = angleFrom
nt.parent);		_			<pre>//RotationInstance(inst).angleTo = angleTo;</pre>
	}else{	//break if ctrl, shft, or alt is pressed		1	}
		//trace("don't drag");	}	ſ	
	}				arm.node {
}			=		<pre>display.*; text.*;</pre>
pub	lic functio	on onMouseOver(event:MouseEvent):void	=		events.*
{			impor	t flash.	net.*;
o+OualifiedClassNa		<pre>andparentClass:String = arget.parent.parent);</pre>	nuhli	a alaga	Nodel ander extends Sprite (
ecquarrineucrassna	me (event.co	riget.parent.parent,,	publi	C Class	NodeLoader extends Sprite {
	if(gra	ndparentClass == "Node") {		private	e var node:Node;
		_mousedOver = true;		14. 6	atian MadaTandan/anda Mada
nt;		_focusedNode = event.target.parent.par-	dud	{	ction NodeLoader(node:Node)
		_focusedButton = event.target;			this.node = node;
		if(!_ctrlPressed && !_shftPressed){			<pre>//this.loadProg = new TextField();</pre>
arget.parent.pare	nt, event.t	_controller.onMouseOver(event.			<pre>init();</pre>
	,	}else if(_ctrlPressed && !_shftPressed){	}		
		_controller.onCtrlOver(event.tar-		private	e function init():void{
et.parent.parent,	event.targ	get); }else if(! ctrlPressed && shftPressed){		}	//
		_controller.onShftOver(event.tar-		,	
et.parent.parent,	event.targ			public	<pre>function load(dat:Array):void{</pre>
		<pre>}else if(_ctrlPressed && _shftPressed) {</pre>			<pre>//trace("nodeLoader load") node.dat = dat;</pre>
arget.parent.pare	nt, event.t	-			node.dot.load();
		}			<pre>node.tag.load(dat);</pre>
	}else{	//trace("uh oh, grandparent is not a			<pre>node.icon.load(); //node.tag.load(node.dat);</pre>
ode but a: " + gr	andparentCl				node.dataBox.load(dat);
	}				<pre>//node.thumb.load(node.dat[0] + node.dat[1]);</pre>
}				hublic	<pre>function make():void{</pre>
pub	lic functio	on onMouseOut(event:MouseEvent):void		Public	//trace("nodeLoader make")
{					<pre>node.dot.make();</pre>
etOualifiedClassNa	_	<pre>andparentClass:String = arget.parent.parent);</pre>			<pre>node.tag.make(); node.icon.make();</pre>
	(,,			node.dataBox.make();
	if(gra	ndparentClass == "Node") {		,	<pre>//node.thumb.make();</pre>
		_mousedOver = false; //just clear for now) bublic	<pre>function go():void{</pre>
		_controller.onShftOut();		1	//nodes are initially idle
		_controller.onMouseOut(event.target.par-			<pre>//node.nodeState = node.idle;</pre>
nt.parent);	}else{				<pre>//node.addChild(node.thumb); node.addChild(node.dot);</pre>
	,0100(//trace("uh oh, grandparent is not a			node.addChild(node.icon);
ode but a: " + gr	andparentCl	ass);			node.addChild(node.tag);
}	}				<pre>node.addChild(node.dataBox); //node.addChild(node.thumb); //just added this so I ca</pre>
,			remove	bitmap	loader from display list
}//end class				}	
<pre>//end pkg ackage metaswarm.</pre>	effects		1//en	d class	
aonago modaonaim.	0110000		}//end		
		s/myEffects/Rotation.as	package	metasw	arm.node {
import mx.e		enEffect; ectInstance;		//fsm i	interface
=		fectInstance;			interface NodeState{ //< public
			breaks	program	
<pre>public clas {</pre>	ss ZoomIn e	xtends TweenEffect			//internal interface methods
// :	Define param	meters for the effect.			<pre>//function filter():void;</pre>
_	_	<pre>yleFrom:Number = 0;</pre>			<pre>function select():void;</pre>
pub	lic var ang	<pre>gleTo:Number = 360;</pre>			<pre>function deSelect():void; function toggleSelect():void;</pre>
//	Define const	ructor with optional argument.			function drag():void;
	lic functio	on Rotation(targetObj:* = null) {			function drop():void;
	=	targetObj);			//function dropThumb():void;
}	//inst	<pre>anceClass = RotationInstance;</pre>			<pre>function popout():void; function popin():void;</pre>
•					<pre>function close():void;</pre>
//	Override ge	etAffectedProperties() method to return			<pre>function setTarget(tgx:Number,</pre>

```
tgy:Number):void;
               function rePosition():void;
                                                                                         //use move(x,y) instead
               function reSize():void;
                                                                                        /*public function position(xNew:Number,
               function startFocus():void;
                                                                          yNew:Number):void
               function stopFocus():void;
                                                                                                this.x = xNew;
               function startHover():void;
              function stopHover():void;
                                                                                                this.y = yNew;
                                                                                        } * /
               //function incrTag():void;
              //function decrTag():void;
                                                                                         public function goToTarget():void {
                                                                                                //trace("x,y before: " + dx + " " + dy);
                                                                                                //trace("tgs before: " + tgx + " " + tgy);
package metaswarm.node.components {
       import flash.display.DisplayObject;
                                                                                                dx = tgx - x;
       import flash.display.Graphics;
                                                                                                dy = tgy - y;
       import flash.display.Shape;
                                                                                                if (Math.sqrt(dx*dx + dy*dy) < SNAP) {
       import flash.display.Sprite;
                                                                                                       x = tgx;
                                                                                                       y = tqy;
       public class Particle extends Sprite{
                                                                                                       //removeEventListener(Event.Enter FRAME,
                                                                          onEnterFrame);
               //public bc used in subclass plot functions
                                                                                                }else{
               //public static const DEF_CLR:Color = #000000;
                                                                                                       vx = dx * EASE;
                                                                                                       vy = dy * EASE;
              //public static const FOC_CLR:Color = #FF0000;
              private static const EASE:Number = .25;
                                                                                                       x += vx;
                                                  = 1;
              private static const SNAP:Number
                                                                                                       y += vy;
              //Target Coordinates
                                                                                                //trace("after: " + dx + " " + dy);
                                                                                                //trace("tgs after: " + tgx + " " + tgy);
              internal var tgx:Number;
              internal var tgy:Number;
              //Previous Coordinates
                                                                                         public function snapToTarget():void{ //<-----</pre>
              public var px:Number;
                                                                          wtf
              public var py:Number;
                                                                                                x = tgx;
                                                                                                y = tgy;
              //distance to target
              private var dx:Number;
              private var dy:Number;
                                                                                        public function setTarget(a:Number, b:Number):void
                                                                          {//<----wtf
              //velocity
                                                                                                tqx = a;
              private var vx:Number;
                                                                                                tgy = b;
              private var vy:Number;
              //atTarget:Boolean; unnec. used expression instead
                                                                                        public function setPrev(a:Number, b:Number):void {//<--
                                                                          ----wt.f
               //hidden:Boolean; use inherited prop visible instead
                                                                                                px = a;
              //infocus:Boolean; not needed because mouse events
                                                                                                py = b;
               \ensuremath{//\mathrm{are}} object oriented. that is they test for focus
              //internally
                                                                                         //getFocus():Interactive Object
              //x
                                                                                         //hitTestPoint(x:Number, y:Number, shapeFlag:Boolean =
                                                                          false):Boolean
              //visible : Boolean
                                                                                         //startDrag()
                                                                                         //stopDrag()
              //still unclear about constructors, see pg 180 of FofEd
anim.
              public function Particle(tgx:Number=0,
                                                                          package metaswarm.node.components.databox {
                                                                                  import flash.display.*;
                                                                                  import flash.text.*;
tgy:Number=0,
dx:Number=0,
                                                                                 public class ButtonDisplayState extends Sprite {
                                                                                         private var typeOfButton:String;
dv:Number=0,
                                                                                        private var bgColor:uint;
                                                                                         private var lineColor:uint;
vx:Number=0,
                                                                                         private var size:uint;
vy:Number=0):void
                                                                                        public function ButtonDisplayState(typeOfButton:String,
                      this.x = 0;
                                                                          lineColor:uint, bgColor:uint, size:uint) {
                      this.y = 0;
                                                                                                this.typeOfButton = typeOfButton;
                      this.tgx = tgx;
                                                                                                this.bgColor = bgColor;
                      this.tgy = tgy;
                                                                                                this.lineColor = lineColor;
                      this.dx = dx;
                                                                                                this.size = size;
                      this.dy = dy;
                                                                                                draw();
                      this.vx = vx;
                      this.vy = vy;
                      this.px = 0;
                                                                                         private function draw():void {
                                                                                                graphics.beginFill(bgColor);
                      this.py = 0;
                                                                                                graphics.drawRect(0, 0, size, size);
                      //init();
                                                                                                graphics.endFill();
                                                                                                graphics.lineStyle(0, lineColor);
              /*public function init():void {
                                                                                                //graphics.beginFill(lineColor);
                     //nothing for now
                                                                                                //. . . . .
                                                                                                //.a.b.
              public function isVisible():Boolean {
                                                                                                //. . . . .
                                                                                                //. c . d .
                      return this.visible;
                                                                                                //. . . . .
```

lacksquare

```
var ax:Number = size*(1/4);
                                                                                                  hitTestState.v = hitTestState.x;
                      var av:Number = size*(1/4);
                                                                                                   useHandCursor = true;
                      var bx:Number = size*(3/4);
                      var by: Number = size*(1/4);
                      var cx:Number = size*(1/4);
                                                                                           public function make():void {
                      var cy:Number = size*(3/4);
                                                                                                   /*downState
                      var dx:Number = size*(3/4);
                                                                            ButtonDisplayState(downColor, size, txt);
                      var dy:Number = size*(3/4);
                                                                                                   overState
                                                                            ButtonDisplayState(overColor, size, txt);
                                                                                                                 = new ButtonDisplayState(upColor,
                      switch (typeOfButton) {
                                                                            size, txt);
                              case "close"://x
                                                                                                  hitTestState = new ButtonDisplayState(upColor,
                                     //trace(0);
                                                                            size, txt);
                                     graphics.moveTo(ax, ay);
                                                                                                   hitTestState.x = -(size / 4);
                                     graphics.lineTo(dx, dy);
                                                                                                  hitTestState.y = hitTestState.x;
                                                                                                   useHandCursor = true;*/
                                     graphics.moveTo(bx, by);
                                      graphics.lineTo(cx, cy);
                                     break;
                              case "popout"://>
                                     //trace(1);
                                     graphics.moveTo(ax, av);
                                     graphics.lineTo(dx, dy);
                                                                            package metaswarm.node.components.databox
                                      graphics.moveTo(bx, by);
                                                                                    import flash.display.*;
                                     graphics.lineTo(dx, dy);
                                                                                    import flash.text.*;
                                      graphics.lineTo(cx, cy);
                                                                                    import flash.events.*;
                                     break;
                                                                                    import flash.utils.*;
                              case "popin"://<</pre>
                                                                                   import metaswarm.node.components.Particle;
                                      //trace(2);
                                     graphics.moveTo(ax, ay);
                                                                                   public class DataBox extends Particle{
                                      graphics.lineTo(dx, dy);
                                     graphics.moveTo(bx, by);
                                                                                           //{\rm need} an idea for select and filter functions
                                      graphics.lineTo(ax, ay);
                                                                                           //e.g. select matching text with id eq to this.
                                     graphics.lineTo(cx, cy);
                                                                                           //this is id of array element no textBox
                                     break;
                                                                                           //war id:uint;
                              default:
                                     trace("Not a valid type of but-
                                                                                           public var dat:Array;
ton");
                                                                                           public var t:TextField = new TextField();
                      //graphics.endFill();
                                                                                           public var closeButton:CustomSimpleButton;
                                                                                           public var popoutButton: CustomSimpleButton;
                      /*var label:TextField = new TextField();
                                                                                           public var popinButton:CustomSimpleButton;
                      var fmt:TextFormat = new TextFormat(txt, size/2,
                                                                                           //var loadExtButton:CustomSimpleButton;
0 \times CCCCCCC):
                      //label.x = dbx.closeButton.x;
                                                                                           public var dataBoxMaker:DataBoxMaker;
                      //label.y = dbx.closeButton.y;
                      label.autoSize = TextFieldAutoSize.LEFT;
                                                                                           public function DataBox():void{
                      //label.width = size;
                                                                                                   dataBoxMaker = new DataBoxMaker(this);
                      //label.height = size;
                                                                                                   init();
                      label.setTextFormat(fmt);
                      label.text = txt;
                                                                                           public function init():void {
                      addChild(label);
                                                                                                   visible = false;
                      label.mouseEnabled = false;*/
                                                                                                   focusRect = true;
                                                                                                   tabEnabled = true;
                                                                                                   //addEventListener(MouseEvent.CLICK, onMouse-
package metaswarm.node.components.databox {
                                                                            Click);
       import flash.display.*;
                                                                                           public function load(dat:Array):void{
       import flash.text.*;
                                                                                                   this.dat = dat;
       public class CustomSimpleButton extends SimpleButton {
               /*private var upColor:uint = 0xFFCC00;
                                                                                           public function make():void {
               private var overColor:uint = 0xCCFF00;
                                                                                                   dataBoxMaker.make();
               private var downColor:uint = 0x00CCFF;
              private var black:uint = 0x231F20;*/
                                                                                           private function onMouseClick(event:MouseEvent):void{
              private static const BLACK:uint = 0x231F20;
                                                                                                   //trace(getQualifiedClassName(event.target));
               private static const WHITE:uint = 0xFFFFFF;
               private static const BRIGHTRED:uint = 0xFF3333;
                                                                                                   /*if(getQualifiedClassName(event.target) ==
               private static const BRIGHTBLUE:uint = 0x66CDFF;
                                                                            "::CustomSimpleButton"){
                                                                                                          switch (event.target.type) {
               public var size:uint;
                                                                                                                  case CLOSE:
              public var type:String;
                                                                                                                         trace("close");
                                                                                                                         break;
              public function CustomSimpleButton(type:String) {
                                                                                                                  case POPOUT:
                      this.size = 15;
                                                                                                                         trace("popout");
                      this.type = type;
                                                                                                                         parent.thumb.load(dat[0]
                      downState
                                     = new ButtonDisplayState(type,
                                                                            + dat[1]);
BRIGHTBLUE, BLACK, size);
                                                                                                                         parent.thumb.make();
                      overState
                                     = new ButtonDisplayState(type,
                                                                                                                         parent.thumb.x = this.
BRIGHTRED, BLACK, size);
                                                                            width;
                                     = new ButtonDisplayState(type,
                                                                                                                         parent.thumb.v = this.v;
                      upState
                                                                                                                         parent.addChild(parent.
WHITE, BLACK, size);
                      hitTestState = new ButtonDisplayState(type,
                                                                            thumb);
WHITE, BLACK, size);
                                                                                                                         break:
                      hitTestState.x = -(size / 4);
                                                                                                                  case POPIN:
```

```
trace("popin");
                                                                                                  //graphics.beginFill(lineColor);
                                             break;
                                     default:
                                                                                                  //. . . . .
                                             trace ("Not a valid type
                                                                                                  //. a . b .
of button");
                                                                                                  //. . . . .
                                                                                                  //. c . d .
                      } * /
                                                                                                  var ax:Number = size*(1/4);
                                                                                                  var ay:Number = size*(1/4);
       }//end class
                                                                                                  var bx:Number = size*(3/4);
                                                                                                  var by: Number = size*(1/4);
}//end pkg
package metaswarm.node.components.databox
                                                                                                  var cx:Number = size*(1/4);
       import flash.display.*;
                                                                                                  var cy:Number = size*(3/4);
                                                                                                  var dx:Number = size*(3/4);
       import flash.text.*;
                                                                                                  var dy:Number = size*(3/4);
       class DataBoxButton extends SimpleButton {
               /*private var upColor:uint = 0xFFCC00;
               private var overColor:uint = 0xCCFF00;
                                                                                                  switch (typeOfButton) {
               private var downColor:uint = 0x00CCFF;
                                                                                                          case "close"://x
               private var black:uint = 0x231F20;*/
                                                                                                                 //trace(0);
                                                                                                                 graphics.moveTo(ax, av);
               private static const BLACK:uint = 0x231F20;
                                                                                                                 graphics.lineTo(dx, dy);
               private static const WHITE:uint = 0xFFFFFF;
                                                                                                                 graphics.moveTo(bx, by);
               private static const BRIGHTRED:uint = 0xFF3333;
                                                                                                                 graphics.lineTo(cx, cy);
               private static const BRIGHTBLUE:uint = 0x66CDFF;
                                                                                                                 break;
                                                                                                          case "popout"://>
               var size:uint;
                                                                                                                 //trace(1);
               var type:String;
                                                                                                                 graphics.moveTo(ax, ay);
                                                                                                                 graphics.lineTo(dx, dy);
               public function DataBoxButton(type:String) {
                                                                                                                 graphics.moveTo(bx, by);
                      this.size = 15;
                                                                                                                 graphics.lineTo(dx, dy);
                      this.type = type;
                                                                                                                 graphics.lineTo(cx, cy);
                                     = new DataBoxButtonState(type,
                      downState
                                                                                                                 break;
BRIGHTBLUE, BLACK, size);
                                                                                                          case "popin"://<</pre>
                      overState
                                     = new DataBoxButtonState(type,
                                                                                                                 //trace(2);
BRIGHTRED, BLACK, size);
                                                                                                                 graphics.moveTo(ax, ay);
                                     = new DataBoxButtonState(type,
                                                                                                                 graphics.lineTo(dx, dy);
                      upState
WHITE, BLACK, size);
                                                                                                                 graphics.moveTo(bx, by);
                      hitTestState = new DataBoxButtonState(type,
                                                                                                                 graphics.lineTo(ax, ay);
WHITE, BLACK, size);
                                                                                                                 graphics.lineTo(cx, cy);
                      hitTestState.x = -(size / 4);
                                                                                                                 break;
                      hitTestState.y = hitTestState.x;
                                                                                                          default:
                      useHandCursor = true;
                                                                                                                 trace("Not a valid type of but-
                                                                            ton"):
               public function make() {
                      /*downState
                                                                                                  //graphics.endFill();
ButtonDisplayState(downColor, size, txt);
                      overState
                                                                                                  /*var label:TextField = new TextField();
ButtonDisplayState(overColor, size, txt);
                                                                                                  var fmt:TextFormat = new TextFormat(txt, size/2,
                                     = new ButtonDisplayState(upColor,
                                                                           0xCCCCCC);
                                                                                                  //label.x = dbx.closeButton.x;
size, txt);
                                                                                                  //label.y = dbx.closeButton.y;
                      hitTestState = new ButtonDisplayState(upColor,
                                                                                                  label.autoSize = TextFieldAutoSize.LEFT;
size, txt);
                                                                                                  //label.width = size;
                      hitTestState.x = -(size / 4);
                      hitTestState.y = hitTestState.x;
                                                                                                  //label.height = size;
                      useHandCursor = true; */
                                                                                                  label.setTextFormat(fmt);
                                                                                                  label.text = txt;
                                                                                                  addChild(label);
                                                                                                  label.mouseEnabled = false; */
package metaswarm.node.components.databox {
                                                                            package metaswarm.node.components.databox {
       import flash.display.*;
       import flash.text.*;
                                                                                   import flash.display.*;
                                                                                   import flash.text.*;
       class DataBoxButtonState extends Sprite {
               private var typeOfButton:String;
                                                                                   public class DataBoxMaker extends Sprite{
               private var bgColor:uint;
               private var lineColor:uint;
                                                                                           private static const CLOSE:String = "closeDataBox";
               private var size:uint;
                                                                                           private static const POPOUT:String = "popout";
                                                                                           private static const POPIN:String = "popin";
               public function DataBoxButtonState(typeOfButton:String,
lineColor:uint, bgColor:uint, size:uint) {
                                                                                           public var dbx:DataBox;
                      this.typeOfButton = typeOfButton;
                      this.bgColor = bgColor;
                                                                                           public function DataBoxMaker(dataBox:DataBox):void{
                      this.lineColor = lineColor;
                                                                                                  this.dbx = dataBox;
                      this.size = size;
                                                                                                  init();
                      draw();
                                                                                           public function init():void {
               private function draw():void {
                                                                                           public function make():void {
                      graphics.beginFill(bgColor);
                      graphics.drawRect(0, 0, size, size);
                                                                                                  makeTextField();
                      graphics.endFill();
                                                                                                  //makeBut.tons();
```

graphics.lineStyle(0, lineColor);

```
dbx.t.setTextFormat(h3Fmt, c2, c3);
              private function makeTextField():void {
                                                                                                   dbx.t.setTextFormat(1Fmt, c3, dbx.t.text.
                       /*h1 = new TextField();
                                                                            length);
                      h2 = new TextField();
                                                                                                   dbx.t.selectable = false;
                      h3 = new TextField();
                      l = new TextField();*/
                                                                                                   dbx.t.mouseEnabled = false;
                      //var t:TextField = new TextField();
                                                                                                  //this.id = 0;
                                                                                                   dbx.addChild(dbx.t);
                      var h1Fmt:TextFormat = new TextFormat("Technic",
                                                                                                   //initTag();
18, 0x333333);
                      //var h2Fmt:TextFormat = new TextFormat();
                      var h3Fmt:TextFormat = new TextFormat("Technic",
                                                                                           private function makeButtons():void {
14, 0x06c);
                      var lFmt:TextFormat = new TextFormat("Technic",
                                                                                                   dbx.closeButton = new CustomSimpleButton(CLOSE);
8, 0x333333);
                                                                                                   dbx.popoutButton = new
                                                                            CustomSimpleButton(POPOUT);
                      /*h1Fmt.TextFormat(font:String = null,
                                                                                                   dbx.popinButton = new CustomSimpleButton(POPIN);
                                        size:Object = null,
                                                                                                   //dbx.loadExtButton:CustomSimpleButton = new
                                         color:Object = null,
                                                                            CustomSimpleButton("e");
                                        bold:Object = null,
                                         italic:Object = null.
                                                                                                   //var mvButton:Button = new Button();
                                         underline:Object = null,
                                                                                                  //dbx.close.label = "x";
                                                                                                   //dbx.close.emphasized = true;
                                         url:String = null,
                                         target:String = null,
                                                                                                  //dbx.close.width = 15;
                                         align:String = null,
                                                                                                   //close.move(20, 20);
                                         leftMargin:Object = null,
                                         rightMargin:Object = null,
                                                                                                   /*dbx.closeButton.make();
                                         indent:Object = null,
                                                                                                   {\tt dbx.popoutButton.make();}
                                         leading:Object = null);*/
                                                                                                   dbx.popinButton.make();
                                                                                                   dbx.loadExtButton.make();*/
                      //alwaysShowSelection : Boolean
                                                                                                   dbx.closeButton.x = dbx.t.width-dbx.closeButton.
                                                                            width;
                      //dbx.t.autoSize = TextFieldAutoSize.LEFT;
                                                                                                   //dbx.closeButton.y = dbx.t.height;
                      dbx.t.height = 100;
                      dbx.t.width = 100;
                                                                                                   dbx.popoutButton.x = dbx.t.width-dbx.popoutBut-
                      //dbx.t.background = true;
                                                                            ton.width;
                      //dbx.t.backgroundColor = 0x231F20;
                                                                                                   dbx.popoutButton.y = dbx.t.height-15;
                      dbx.t.alpha = 50;
                      //dbx.t.border = true;
                                                                                                   dbx.popinButton.x = dbx.t.width-(15 * 2);
                      //dbx.t.borderColor = 0x000000;
                                                                                                   dbx.popinButton.y = dbx.t.height-15;
                      //t.height =
                      //t..width =
                                                                                                   //dbx.loadExtButton.x = dbx.t.width-(15 * 3);
                      dbx.t.mouseEnabled = false:
                                                                                                  //dbx.loadExtButton.y = dbx.t.height-15;
                      //t.mouseWheelEnabled = true;
                      //t.multiline = true;
                      dbx.t.wordWrap = true;
                                                                                                   dbx.addChild(dbx.closeButton);
                                                                                                   dbx.addChild(dbx.popoutButton);
                      //t.numLines [read-only]
                                                                                                   dbx.addChild(dbx.popinButton);
                      //t.length [read-only]
                                                                                                  //dbx.addChild(dbx.loadExtButton);
                      //t.scrollH : int
                      //t.scrollV : int
                      //t.selectable = true;
                      //t.tabEnabled = true;
                      //t.thickness : Number
                                                                            package metaswarm.node.components.dot
                      /*//t.width =
                                                                                   import flash.display.DisplayObject;
                      //t.height =
                                                                                import flash.display.Graphics;
                      t.background = true;
                                                                                import flash.display.Shape;
                      t.backgroundColor = 0xCCCCCC; //light gray
                                                                                    import flash.display.Sprite;
                                                                                    import metaswarm.node.components.Particle;
                      t.border = true;
                      t.borderColor = 0x333333; //dark gray
                      t.autoSize = TextFieldAutoSize.LEFT;
                                                                                   public class Dot extends Particle{
                      //addChild(this);*/
                      dbx.t.text = "";
                                                                                           public var dotMaker:DotMaker:
                                                                                           public var dotButton:DotButton;
                      var c0:uint = 0;
                      var c1:uint = c0 + dbx.dat[0].length + dbx.
                                                                                           public function Dot():void{
dat[1].length +1;
                                                                                                   dotMaker = new DotMaker(this);
                      var c2:uint = c1 + dbx.dat[2].length + 1;
                                                                                                   init();
                      var c3:uint = c2 + 10;
                                                                                           public function init():void {
                      dbx.t.appendText(dbx.dat[0] + dbx.dat[1] +
                                                                                                   //visible = true;
'\n');
                                                                                                   //focusRect = true;
                                                                                                   //tabEnabled = true;
                      dbx.t.appendText(dbx.dat[2] + '\n');
                      dbx.t.appendText("_____" + '\n');
                      //t.text += dat[0] + dat[1] + '\n';
                      //t.text += dat[2] + '\n';
                                                                                           public function load():void{
                      //t.text += "____
                      for(var i:uint=3; i<dbx.dat.length; i++) {</pre>
                                                                                           //might dep on load later, so leave here!
                              dbx.t.appendText(dbx.dat[i] + ", ");
                                                                                           public function make():void{
                              //t.text += dat[i] + ", ";
                                                                                                   dotMaker.make();
                      dbx.t.setTextFormat(h1Fmt, c0, c1);
                      dbx.t.setTextFormat(h3Fmt, c1, c2);
```

```
init();
package metaswarm.node.components.dot {
       import flash.display.*;
                                                                                          public function init():void{
       public class DotButton extends SimpleButton{
                                                                                          public function make():void{
               private static const BLACK:uint = 0x231F20;
                                                                                                  _dot.dotButton = new DotButton();
               private static const WHITE:uint = 0xFFFFFF;
                                                                                                  _dot.addChild(_dot.dotButton);
               private static const BRIGHTRED:uint = 0xFF3333;
               private static const BRIGHTBLUE:uint = 0x66CDFF;
               public var size:uint;
                                                                           package metaswarm.node.components.icon {
               public var type:String;
                                                                                  import flash.display.DisplayObject;
                                                                                import flash.display.Graphics;
               public function DotButton() {
                                                                                import flash.display.Shape;
                      this.size = 3;
                                                                                   import flash.display.Sprite;
                      this.type = "dot";
                                                                                   import metaswarm.node.components.Particle;
                                     = new DotButtonState(BRIGHTBLUE,
                      downState
WHITE, size);
                      overState
                                     = new DotButtonState(BRIGHTRED,
                                                                                   public class Icon extends Particle{
WHITE, size);
                                     = new DotButtonState(BLACK,
                                                                                          private static const Z EASE:Number = 1;
                      upState
WHITE, size);
                                                                                          private static const Z_SNAP:Number = 1;
                      hitTestState = new DotButtonState(BLACK,
WHITE, size); //change to overstate
                                                                                          //Target zoom
                      hitTestState.x = -(size / 4);
                                                                                          private var tgw:Number;
                      hitTestState.y = hitTestState.x;
                      useHandCursor = true;
                                                                                          //radial distance to zoom target
                                                                                          private var dw:Number;
               public function forceOver():void{
                                                                                          //zoom velocity
                      upState = overState;
                                                                                          private var vw:Number;
                                                                                          //should be a customSimpleButton not a maker
                                                                                          public var iconMaker:IconMaker;
               public function restoreOver():void{
                      upState = hitTestState;
                                                                                          public var iconButton:IconButton;
                                                                                          public function Icon(tgw:Number=0,
                                                                                                                                dw:Number=0,
                                                                                                                                vw:Number=0):void
package metaswarm.node.components.dot {
                                                                                                  this.x = 0;
       import flash.displav.*;
       import flash.text.*;
                                                                                                  this.y = 0;
                                                                                                  this.tgw = tgw;
       //should this extend shape instead????
                                                                                                  this.dw = dw;
       public class DotButtonState extends Sprite {
                                                                                                  this.vw = vw;
               //private var typeOfButton:uint;
               private var bgColor:uint;
                                                                                                  iconMaker = new IconMaker(this);
               private var lineColor:uint;
               private var size:uint;
                                                                                                  init();
               public function DotButtonState(lineColor:uint,
                                                                                          public function init():void {
bgColor:uint, size:uint) {
                                                                                                  visible = false;
                      //this.typeOfButton = typeOfButton;
                                                                                                  focusRect = true;
                      this.bgColor = bgColor;
                                                                                                  tabEnabled = true;
                      this.lineColor = lineColor;
                                                                                                  doubleClickEnabled = true;
                      this.size = size;
                                                                                                  //mouseEnabled = false;
                      draw();
                                                                                          public function load():void{
              private function draw():void {
                      //trace("draw the dot button state");
                                                                                          //might depend on load later, so keep here
                      graphics.beginFill(bgColor);
                      graphics.drawRect(-size, -size, size*2, size*2);
                                                                                          //rather than put in init
                      graphics.endFill();
                                                                                          public function make():void{
                                                                                                  iconMaker.make();
                      graphics.lineStyle(0, lineColor);
                      graphics.moveTo(x, y-size);
                      graphics.lineTo(x, y+size);
                                                                                          public function scaleToTarget():void {
                      graphics.moveTo(x-size, y);
                                                                                                  dw = tgw - this.width;
                      graphics.lineTo(x+size, y);
                                                                                                  if(Math.abs(dw - Z_SNAP)<1){</pre>
                                                                                                         this.width = tgw;
                                                                                                          this.height = tgw;
                                                                                                          //removeEventListener(Event.Enter_FRAME,
package metaswarm.node.components.dot
                                                                           onEnterFrame);
       import flash.display.DisplayObject;
                                                                                                  }else{
                                                                                                         vw = dw * Z EASE;
    import flash.display.Graphics;
    import flash.display.Shape;
                                                                                                         this.width += vw;
       import flash.display.Sprite;
                                                                                                         this.height += vw;
       public class DotMaker extends Sprite{
               private var dot:Dot;
                                                                                          public function setTargetWidth(a:Number):void {
                                                                                                  tgw = a;
               public function DotMaker(dot:Dot):void{
```

dot = dot;

```
public var icon: Icon;
package metaswarm.node.components.icon {
                                                                                           public function IconMaker(icon:Icon):void{
        import flash.display.*;
                                                                                                  this.icon = icon;
                                                                                                  init();
       public class IconButton extends SimpleButton{
                                                                                           public function init():void {
               private static const BLACK:uint = 0x231F20;
               private static const WHITE:uint = 0xFFFFFF;
               private static const BRIGHTRED:uint = 0xFF3333;
                                                                                           public function make():void {
               private static const BRIGHTBLUE:uint = 0x66CDFF;
                                                                                                  //trace("make the icon button");
                                                                                                  icon.iconButton = new IconButton();
                                                                                                  icon.addChild(icon.iconButton);
              public var size:uint;
              public var type:String;
              public function IconButton() {
                      this.size = 3;
                                                                           package metaswarm.node.components.tag {
                                                                                   import flash.display.*;
                      this.type = "icon";
                                     = new IconButtonState(BRIGHTBLUE,
                                                                                   import flash.text.*;
                      downState
BLACK, size);
                                                                                   import metaswarm.node.components.Particle;
                      overState
                                     = new TconButtonState(BRIGHTRED.
                                                                                   public class Tag extends Particle{
BLACK, size);
                      upState
                                     = new IconButtonState(WHITE,
BLACK, size);
                                                                                           //add formatting constants here
                      hitTestState = new IconButtonState(WHITE,
                                                                                           public var tagMaker:TagMaker;
BLACK, size);
                                                                                           public var tagButton:TagButton;
                      hitTestState.x = -(size / 4);
                                                                                           //zoom velocity
                      hitTestState.y = hitTestState.x;
                                                                                           //private var val:String;
                      useHandCursor = true;
                                                                                           public var t:TextField;
                                                                                           //var fmt:TextFormat;
                                                                                           public var uniqArr:Array = new Array();
               public function forceOver():void{
                                                                                           public var isUniq:Boolean = new Boolean();
                      upState = overState;
                                                                                           //need an idea for select and filter functions
                                                                                           //e.g. select matching text with id eq to this.
              public function restoreOver():void{
                                                                                           //this is id of array element no textBox
                      upState = hitTestState;
                                                                                           public var id:uint;
                                                                                           //cnt not nec, can just use array length
               /*//should be draw()
                                                                                           //var cnt:uint;
              public function make():void {
                                                                                           public var dat:Arrav;
                      icon.graphics.beginFill(0x0000ff);
                                                                                           //var dat:Array = ["cherry", "orange", "soda", "pop"];
                      icon.graphics.lineStyle(0);
                      icon.graphics.drawRect(0, 0, 10, 10);
                                                                                           public function Tag():void
                      icon.graphics.endFill();
                                                                                                  t = new TextField();
                                                                                                  tagMaker = new TagMaker(this);
                                                                                                  //fmt = new TextFormat();
package metaswarm.node.components.icon {
                                                                                                  init();
        import flash.display.*;
        import flash.text.*;
                                                                                           public function init():void {
                                                                                                  visible = false;
       //should this extend shape instead????
                                                                                                  //focusRect = false;
       public class IconButtonState extends Sprite {
                                                                                                  //tabEnabled = false;
               //private var typeOfButton:uint;
                                                                                                  //tag.mouseEnabled = false;
               private var bgColor:uint;
              private var lineColor:uint;
                                                                                           public function load(dat:Array):void{
              private var size:uint;
                                                                                                  this.dat = dat;
              public function IconButtonState(lineColor:uint,
bgColor:uint, size:uint) {
                                                                                           public function make():void{
                      //this.typeOfButton = typeOfButton;
                                                                                                  tagMaker.make();
                      this.bgColor = bgColor;
                      this.lineColor = lineColor;
                      this.size = size;
                                                                                           public function incr():void{
                      draw();
                                                                                                  if(id < dat.length-1){
                                                                                                         id++;
                                                                                                  }else{
               private function draw():void {
                                                                                                         id=0;
                      //trace("draw the icon button state");
                      graphics.beginFill(bgColor);
                                                                                                  t.text = dat[id];
                      graphics.lineStyle(0, lineColor);
                                                                                                  isUniq = uniqArr[id];
                       //graphics.drawRect(0, 0, size, size);
                                                                                                  //t.text = String(this.tgx) + ", " +
                      graphics.drawRect(-size, -size, size*2, size*2);
                      graphics.endFill();
                                                                                           public function decr():void{
                                                                                                  if(id > 0){
package metaswarm.node.components.icon {
                                                                                                         id--;
       import flash.display.DisplayObject;
                                                                                                  }else{
                                                                                                         id=dat.length-1;
    import flash.display.Graphics;
    import flash.display.Shape;
       import flash.display.Sprite;
                                                                                                  t.text = dat[id];
                                                                                                  isUniq = uniqArr[id];
       public class IconMaker extends Sprite{
                                                                                                  //t.text = String(this.tgx) + ", " +
```

```
tag.t.text = "";
String(this.tgy);
                                                                                                  tag.t.setTextFormat(fmt);
                                                                                                  tag.t.selectable = false;
                                                                                                  tag.t.mouseEnabled = false;
               /*public function getDatAt(id):void{
                      return dat[id];
                                                                                                  tag.id = 0;
                                                                                                  tag.t.x += 5;
       }//end class
                                                                                                  tag.t.y -= 25;
}//end pkg
                                                                                                  //t.width =
package metaswarm.node.components.tag {
                                                                                                  //t.height =
       import flash.display.*;
                                                                                                  tag.t.background = true
                                                                                                  tag.t.backgroundColor = bgColor; //light gray
       public class TagButton extends SimpleButton{
                                                                                                  //tag.t.border = true;
                                                                                                  tag.t.borderColor = lineColor; //dark gray
               private static const BLACK:uint = 0x231F20;
                                                                                                  tag.t.autoSize = TextFieldAutoSize.LEFT;
               private static const WHITE:uint = 0xFFFFFF;
                                                                                                  //addChild(this);
               private static const BRIGHTRED:uint = 0xFF3333;
               private static const BRIGHTBLUE:uint = 0x66CDFF;
                                                                                                  tag.t.text = tag.dat[tag.id];
               public var size:uint;
                                                                                                  tag.addChild(tag.t);
               public var type:String;
               public function TagButton() {
                      this.size = 5:
                      this.type = "tag";
                                                                            package metaswarm.node.components.tag
                      downState
                                     = new TagButtonState (BRIGHTBLUE,
                                                                                   import flash.display.*;
BLACK, size);
                                                                                    import flash.text.*;
                      overState
                                     = new TagButtonState(BRIGHTRED,
BLACK, size);
                                                                                   public class TagMaker extends Sprite{
                       upState
                                     = new TagButtonState(WHITE,
BLACK, size);
                                                                                          private var tag:Tag;
                      hitTestState = new TagButtonState(WHITE,
BLACK, size);
                                                                                          public function TagMaker(tag:Tag):void{
                      hitTestState.x = -(size / 4);
                                                                                                  this.tag = tag;
                      hitTestState.y = hitTestState.x;
                                                                                                  init();
                      useHandCursor = true;
                                                                                          public function init():void {
                                                                                                  //
               public function forceOver():void{
                       upState = overState;
                                                                                          /*public function make():void {
                                                                                                  //trace("make the tag button");
               public function restoreOver():void{
                                                                                                  tag.tagButton = new TagButton();
                                                                                                  tag.addChild(tag.tagButton);
                       upState = hitTestState;
               /*//should be draw()
                                                                                          public function make():void {
               public function make():void {
                                                                                                  //tag.tagButton = new TagButton();
                       icon.graphics.beginFill(0x0000ff);
                                                                                                  //tag.addChild(tag.tagButton);
                       icon.graphics.lineStyle(0);
                       icon.graphics.drawRect(0, 0, 10, 10);
                                                                                                  var fmt:TextFormat = new TextFormat();
                       icon.graphics.endFill();
               } * /
                                                                                                  fmt.font = "Arial";
                                                                                                  fmt.size = 5;
                                                                                                  //tag.t.embedFonts = true;
package metaswarm.node.components.tag {
                                                                                                  //tag.t.antiAliasType = AntiAliasType.ADVANCED;
                                                                                                  tag.t.text = "";
       import flash.display.*;
                                                                                                  tag.t.setTextFormat(fmt);
       import flash.text.*;
                                                                                                  tag.t.selectable = false;
       //should this extend shape instead????
                                                                                                  tag.t.mouseEnabled = false;
       public class TagButtonState extends Sprite {
                                                                                                  tag.id = 0;
               //private var typeOfButton:uint;
                                                                                                  tag.t.x += 5;
               private var bgColor:uint;
               private var lineColor:uint:
                                                                                                  tag.t.y -= 25;
               private var size:uint;
                                                                                                  //t.width =
                                                                                                  //t.height =
               public function TagButtonState(lineColor:uint,
                                                                                                  tag.t.background = true
bgColor:uint, size:uint) {
                                                                                                  tag.t.backgroundColor = 0xCCCCCC; //light gray
                      //this.typeOfButton = typeOfButton;
                                                                                                  tag.t.border = true;
                      this.bgColor = bgColor;
                                                                                                  tag.t.borderColor = 0x333333; //dark gray
                      this.lineColor = lineColor;
                                                                                                  tag.t.autoSize = TextFieldAutoSize.LEFT;
                      this.size = size;
                                                                                                  //addChild(this);
                      //draw();
                                                                                                  tag.t.text = tag.dat[tag.id];
               //need ref to tag, can't do this yet
                                                                                                  tag.addChild(tag.t);
               /*private function draw():void {
                       //tag.tagButton = new TagButton();
                      //tag.addChild(tag.tagButton);
                                                                            package metaswarm.node.components.thumb
                      var fmt:TextFormat = new TextFormat();
                                                                             import flash.display.*;
                                                                              import flash.text.*;
                      fmt.font = "Arial";
                                                                              import flash.events.*
                      fmt.size = size;
                                                                              import flash.net.*;
                      //tag.t.embedFonts = true;
                                                                              import flash.geom.Matrix;
                      //tag.t.antiAliasType = AntiAliasType.ADVANCED;
```

```
// new, scaled BitmapData object
  public class BitmapLoadr extends Sprite {
                                                                                         //loadedBitmap.bitmapData = scaledBitmapData;
       public var loadProg: TextField;
                                                                                        // Create a new Bitmap using the {\tt BitmapData}
       public var thumb: Thumb;
                                                                                 // and display it.
       private var loader:Loader; // The bitmap loader
                                                                                 //var newImage:Bitmap = new Bitmap(loadedBitmapData);
       //var parsedData:Array;
    //var output:String;
                                                                                         thumb.bmpButton.bmp.bitmapData = scaledBitmapData;
       //private static const PATH:String = "../index.txt";
                                                                                         //addChild(thumb.bmp);
    public function BitmapLoadr(thumb:Thumb) {
                                                                                         //thumb.bmp = loadedImage;
               this.thumb = thumb;
                                                                                         thumb.bmpButton.addChild(thumb.bmpButton.bmp);
               init();
    private function init():void {
                                                                                 private function handleProgress(event:ProgressEvent):void {
              loader = new Loader();
                                                                                 // Calculate the percentage loaded
              loadProg = new TextField();
                                                                                 var percent:Number = Math.round(event.bytesLoaded
       //loadProg.width = stage.stageWidth;
                                                                                                              / event.bytesTotal * 100 );
       //loadProg.height = stage.stageHeight;
                                                                                 thumb.addChild(loadProg);
                                                                                               + "Percent: " + percent;
       public function loadBmp(path:String):void {
              // Listen for the progress event to check download
progress
                                                                                         //trace(loadProg.text);
       loader.contentLoaderInfo.addEventListener(ProgressEvent.PROG-
RESS, handleProgress);
               // listen for load complete
                                                                                 private function onComplete(event:Event):void {
               {\tt loader.contentLoaderInfo.addEventListener(Event.COM-}
                                                                                         //done with loadProg so remove it
PLETE, onComplete);
                                                                                 thumb.removeChild(loadProg);
              // Register to be notified when the bitmap has been ini-
tialized
              loader.contentLoaderInfo.addEventListener(Event.INIT,
onInit);
                                                                          package metaswarm.node.components.thumb {
              //loader.load(new URLRequest(path));
                                                                                 import flash.display.*;
              var request:URLRequest = new URLRequest(path);
                                                                                  import flash.text.*;
              loader.load(request);
                                                                                 public class BmpButton extends Sprite{
       // Triggered when the bitmap has been loaded and initialized
                                                                                         public var type:String;
      private function onInit(event:Event):void {
              //trace("onInit");
                                                                                         public var bmp:Bitmap;
              var loadedBitmap:Bitmap = Bitmap(loader.content);
                                                                                        public function BmpButton() {
              var loadedBitmapData:BitmapData = loadedBitmap.bitmap-
                                                                                                this.type = "bmp";
Data:
                                                                                                bmp = new Bitmap();
              // Set the amount by which to scale the bitmap
               var scaleFactor:Number = .5;
                                                                                         public function make():void {
              /*//{\rm specify} the height here for now <<<<<<**********
*******
              var newHeight:int = 100;
                                                                          package metaswarm.node.components.thumb {
              // Calculate the new dimensions of the scaled bitmap
                                                                                 import flash.display.*;
              scaleFactor = newHeight/loadedBitmapData.height;
                                                                                 import flash.events.*;
              var newWidth:int = loadedBitmapData.width * scaleFac-
                                                                              import flash.net.URLRequest;
                                                                                 import flash.text.*;
tor; */
                                                                                  import metaswarm.node.components.Particle;
              //specify the height here for now <<<<<<**********
*********
                                                                                 public class Thumb extends Particle{
              var newWidth:int = 200;
                                                                                         public var path:String;
              // Calculate the new dimensions of the scaled bitmap
              scaleFactor = newWidth/loadedBitmapData.width;
                                                                                         public var ldr:BitmapLoadr;
              var newHeight:int = loadedBitmapData.height * scaleFac-
                                                                                         public var bmpButton:BmpButton;
tor;
                                                                                         public var bmpLoaded:Boolean;
                                                                                         public var thumbMaker:ThumbMaker;
              // Create a new BitmapData object, sized to hold the
scaled bitmap
                                                                                         public var header:TextField = new TextField();
              var scaledBitmapData:BitmapData = new
                                                                                         public var footer:TextField = new TextField();
BitmapData (newWidth, newHeight,
                                                                                         public var closeButton:ThumbButton;
                                                                                        public var popoutButton:ThumbButton;
       loadedBitmapData.transparent);
                                                                                         public var popinButton:ThumbButton;
              // Create a transformation matrix that will scale the
bitmap
                                                                                         public function Thumb():void {
                                                                                                ldr = new BitmapLoadr(this);
              var scaleMatrix:Matrix = new Matrix();
              scaleMatrix.scale(scaleFactor, scaleFactor);
                                                                                                bmpButton = new BmpButton();
                                                                                                bmpLoaded = false;
              //loadedBitmapData.draw(loadedBitmap, new Matrix())
                                                                                                thumbMaker = new ThumbMaker(this);
              // Draw the scaled bitmap into the new BitmapData ob-
                                                                                                init();
ject
              scaledBitmapData.draw(loadedBitmapData, scaleMatrix);
                                                                                        public function init():void {
                                                                                                //addChild(ldr); //no no no
              // Replace the original BitmapData object with the
                                                                                                visible = false;
```

```
//focusRect = true;
                                                                                                  //. . . . .
                                                                                                  var ax: Number = size*(1/4);
                      //tabEnabled = true;
                                                                                                  var ay: Number = size*(1/4);
                                                                                                  var bx:Number = size*(3/4);
               public function load(path:String):void{
                                                                                                  var by: Number = size*(1/4);
                      this.path = path;
                                                                                                  var cx: Number = size*(1/4);
                      if(!bmpLoaded){
                                                                                                  var cy: Number = size*(3/4);
                              ldr.loadBmp(path);
                                                                                                  var dx:Number = size*(3/4);
                              bmpLoaded = true;
                                                                                                  var dy:Number = size*(3/4);
               public function make():void{
                                                                                                  switch (typeOfButton) {
                                                                                                          case "closeThumb"://x
                      thumbMaker.make();
                                                                                                                 //trace(0);
                                                                                                                 graphics.moveTo(ax, ay);
                                                                                                                 graphics.lineTo(dx, dy);
package metaswarm.node.components.thumb {
                                                                                                                 graphics.moveTo(bx, by);
                                                                                                                 graphics.lineTo(cx, cy);
       import flash.display.*;
       import flash.text.*;
                                                                                                                 break;
                                                                                                         case "popout"://>
       public class ThumbButton extends SimpleButton {
                                                                                                                 //trace(1);
                                                                                                                 graphics.moveTo(ax, ay);
               /*private var upColor:uint = 0xFFCC00;
               private var overColor:uint = 0xCCFF00;
                                                                                                                 graphics.lineTo(dx, dy);
               private var downColor:uint = 0x00CCFF;
                                                                                                                 graphics.moveTo(bx, by);
                                                                                                                 graphics.lineTo(dx, dy);
               private var black:uint = 0x231F20;*/
                                                                                                                 graphics.lineTo(cx, cy);
               private static const BLACK:uint = 0x231F20;
                                                                                                                 break;
               private static const LIGHTGREY:uint = 0xfcfcfc;
                                                                                                          case "popin"://<</pre>
               private static const WHITE: uint = 0xFFFFFF;
                                                                                                                 //trace(2);
               private static const BRIGHTRED:uint = 0xFF3333;
                                                                                                                 graphics.moveTo(ax, ay);
               private static const BRIGHTBLUE:uint = 0x66CDFF;
                                                                                                                 graphics.lineTo(dx, dy);
                                                                                                                 graphics.moveTo(bx, by);
               public var size:uint;
                                                                                                                 graphics.lineTo(ax, ay);
               public var type:String;
                                                                                                                 graphics.lineTo(cx, cy);
                                                                                                                 break;
               public function ThumbButton(type:String) {
                                                                                                          default:
                      this.size = 15;
                                                                                                                 trace("Not a valid type of but-
                                                                            ton");
                      this.type = type;
                      downState
                                     = new ThumbButtonState(type,
BRIGHTBLUE, BLACK, size);
                                                                                          }
                                     = new ThumbButtonState(type,
                      overState
BRIGHTRED, BLACK, size);
                                                                            package metaswarm.node.components.thumb
                      upState
                                     = new ThumbButtonState(type,
                                                                                   import flash.display.DisplayObject;
WHITE, BLACK, size);
                      hitTestState = new ThumbButtonState(type,
                                                                                import flash.display.Graphics;
WHITE, BLACK, size);
                                                                                import flash.display.Shape;
                      hitTestState.x = -(size / 4);
                                                                                    import flash.display.Sprite;
                      hitTestState.y = hitTestState.x;
                                                                                   import flash.text.*;
                      useHandCursor = true;
                                                                                   public class ThumbMaker extends Sprite{
                                                                                           private static const CLOSETHUMB:String = "closeThumb";
               public function make():void
                                                                                           private static const POPOUT:String = "popout";
                                                                                           private static const POPIN:String = "popin";
                                                                                           private var thumb:Thumb;
package metaswarm.node.components.thumb {
       import flash.display.*;
                                                                                           public function ThumbMaker(thumb:Thumb):void{
                                                                                                  this.thumb = thumb;
       import flash.text.*;
                                                                                                  init();
       public class ThumbButtonState extends Sprite {
               private var typeOfButton:String;
                                                                                           public function init():void {
               private var bgColor:uint;
               private var lineColor:uint:
                                                                                           public function make():void {
               private var size:uint;
               public function ThumbButtonState(typeOfButton:String,
                                                                                                  makeHeader();
lineColor:uint, bgColor:uint, size:uint) {
                                                                                                  //makeFooter();
                      this.typeOfButton = typeOfButton;
                                                                                                  makeButtons();
                      this.bgColor = bgColor;
                                                                                                  //thumb.bmp.y = thumb.header.height;
                      this.lineColor = lineColor;
                                                                                                  /*thumb.closeButton = new
                      this.size = size;
                                                                            ThumbButton (CLOSETHUMB);
                                                                                                  thumb.closeButton.x = thumb.width-thumb.close-
                      draw();
                                                                            Button.width;
                                                                                                  thumb.addChild(thumb.closeButton);*/
               private function draw():void {
                      graphics.beginFill(bgColor);
                      graphics.drawRect(0, 0, size, size);
                                                                                           private function makeHeader():void {
                      graphics.endFill();
                      graphics.lineStyle(1, lineColor);
                                                                                                  var fmt:TextFormat = new TextFormat("Technic",
                                                                            12, 0x66CDFF);
                      //graphics.beginFill(lineColor);
                                                                                                  //thumb.header.height = 15;
                      //. . . . .
                      //. a . b .
                                                                                                  //thumb.header.width = 125;
                      //. . . . .
                                                                                                  thumb.header.background = true;
                                                                                                  thumb.header.backgroundColor = 0x231F20;
                      //. c . d .
```

```
//thumb.header.alpha = 50;
                      //thumb.header.border = true;
                                                                                                  thumb.addChild(thumb.bmpButton);
                      //thumb.header.borderColor = 0x000000;
                                                                                                  thumb.addChild(thumb.closeButton);
                      thumb.header.autoSize = TextFieldAutoSize.LEFT;
                                                                                                  thumb.addChild(thumb.popoutButton);
                      //addChild(this);*/
                                                                                                  thumb.addChild(thumb.popinButton);
                      thumb.header.text = "123.jpg";
                                                                                                  //thumb.addChild(thumb.loadExtButton);
                      thumb.header.setTextFormat(fmt);
                      thumb.header.selectable = false;
                      thumb.header.mouseEnabled = false;
                      //this.id = 0;
                      thumb.addChild(thumb.header);
                                                                           package metaswarm.node.nodestates
                      //initTag();
                                                                                   import metaswarm.node.NodeState;
                                                                                   public class DraggingState implements NodeState{
               private function makeFooter():void {
                                                                                          private var node: Node;
                      var fmt:TextFormat = new TextFormat("Technic",
10, 0x000000);
                                                                                          public function DraggingState(node:Node)
                      thumb.footer.height = 15;
                                                                                                  //trace("construct DraggingState");
                      thumb.footer.width = 125;
                                                                                                  this.node = node;
                      thumb.footer.background = true;
                      thumb.footer.backgroundColor = 0xffffff;
                                                                                          public function select():void{}
                      //thumb.header.autoSize = TextFieldAutoSize.
                                                                                          public function deSelect():void{}
LEFT;
                      thumb.footer.text = "footer";
                                                                                          public function toggleSelect():void{}
                      thumb.footer.setTextFormat(fmt);
                                                                                          public function drag():void{}
                      thumb.footer.selectable = false;
                                                                                          public function drop():void{
                      thumb.footer.mouseEnabled = false;
                                                                                                  //trace("PickedState drag: goto dragging
                      //this.id = 0;
                                                                           state");
                                                                                                  node.icon.visible = false;
                      //thumb.footer.x = 0;
                      thumb.footer.y = thumb.footer.width;
                                                                                                  node.icon.stopDrag();
                                                                                                  node.thumb.stopDrag();
                      thumb.addChild(thumb.footer);
                      //initTag();
                                                                                                  node.thumb.x = node.icon.x;
                                                                                                  node.thumb.y = node.icon.y;
                                                                                                  node.thumb.setTarget(node.icon.x, node.icon.y);
              private function makeButtons():void {
                                                                                                  node.thumb.visible = true;
                      //this should be in thumb
                                                                                                  //goto dropped state
class************
                                                                                                  node.nodeState = node.dropped;
                      thumb.closeButton = new ThumbButton(CLOSETHUMB);
                      thumb.popoutButton = new ThumbButton(POPOUT);
                      thumb.popinButton = new ThumbButton(POPIN);
                                                                                          //public function dropThumb():void{}
                      //thumb.loadExtButton:CustomSimpleButton = new
CustomSimpleButton("e");
                                                                                          public function popout():void{}
                      var buttonSize:uint = thumb.closeButton.size;
                                                                                          public function popin():void{}
                      var thumbWidth:uint = 200;
                                                                                          public function close():void{}
                      //var myButton:Button = new Button();
                      //thumb.close.label = "x";
                                                                                          public function setTarget(tgx:Number,
                      //thumb.close.emphasized = true;
                      //thumb.close.width = 15;
                                                                           tay:Number):void{
                      //close.move(20, 20);
                                                                                                  //trace("DraggingState switch");
                                                                                                  //node.dot.setTarget(tgx, tgy);
                      /*thumb.closeButton.make();
                      thumb.popoutButton.make();
                      thumb.popinButton.make();
                                                                                          public function rePosition():void{
                      thumb.loadExtButton.make(); */
                                                                                                  //trace("DraggingState rePosition");
                                                                                                  //node.dot.goToTarget();
                      thumb.closeButton.x = thumbWidth - buttonSize;
                                                                                                  //node.thumb.x = node.icon.x;
                      //thumb.closeButton.y = thumb.header.height;
                                                                                                  //node.thumb.y = node.icon.y;
                      thumb.popoutButton.x = thumbWidth - button-
                                                                                                  //dragging icon so snap dot to icon
                                                                                                  /*node.dot.x = node.dataBox.x;
Size*2;
                                                                                                  node.dot.y = node.dataBox.y;*/
                      //thumb.popoutButton.y = (thumb.header.height +
thumb.bmp.height) - 15;
                      //thumb.popoutButton.y = buttonSize;
                      thumb.popinButton.x = thumbWidth - buttonSize*3;
                      //thumb.popinButton.y = (thumb.header.height +
                                                                                          public function reSize():void{
thumb.bmp.height) - 15;
                                                                                                  //trace("DraggingState reSize");
                      //thumb.popinButton.y = buttonSize*2;
                                                                                                  node.icon.scaleToTarget(); //???
                      //thumb.loadExtButton.x = thumb.header.width-(15
* 3);
                                                                                          public function startFocus():void{}
                      //thumb.loadExtButton.y = thumb.header.
height-15;
                                                                                          public function stopFocus():void{}
```

```
public function startHover():void{}
                                                                                           public function rePosition():void{
                                                                                                  //trace("DroppedState rePosition");
               public function stopHover():void{}
                                                                                                  //preview mainstate need this to tile
                                                                                                  node.thumb.goToTarget();
package metaswarm.node.nodestates
       import metaswarm.node.NodeState;
                                                                                           public function startFocus():void{}
       public class DroppedState implements NodeState{
                                                                                           public function stopFocus():void{}
                                                                                           public function reSize():void{} //icon is not visible
                                                                            so don't bother
               private var node: Node;
                                                                                           public function startHover():void{} //change these
               public function DroppedState(node:Node)
                                                                            names to hover or smth
                                                                                           public function stopHover():void{}
                      //trace("construct DroppedState");
                      this.node = node;
                                                                            package metaswarm.node.nodestates {
               public function select():void{}
                                                                                   import flash.geom.ColorTransform;
               public function deSelect():void{}
                                                                                   import metaswarm.node.NodeState;
               public function toggleSelect():void{}
                                                                                   public class IdleState implements NodeState{
               public function drag():void{
                      //trace("DroppedState drag: goto dragging
                                                                                           private var redTransform:ColorTransform;
state");
                                                                                           private var blkTransform:ColorTransform;
                      //start dragging
                                                                                           private var node:Node;
                      node.thumb.startDrag();
                                                                                           function IdleState(node:Node)
                       //goto to dragging state
                       node.nodeState = node.tdragging;
                                                                                                  //trace("construct IdleState");
                      node.prevState = node.dropped;
                                                                                                  this.node = node;
                                                                                                  redTransform = new ColorTransform();
                                                                                                  redTransform.color = 0xFF0000;
               public function drop():void{}
                                                                                                  blkTransform = new ColorTransform();
                                                                                                  blkTransform.color = 0xCCCCCC;
               //public function dropThumb():void{}
                                                                                           public function select():void{
               public function popout():void{
                                                                                                  //trace("IdleState select:goto picked state");
                      //trace("DroppedState popout");
                                                                                                  //node.dot.mouseEnabled = false;
                      //position dataBox next to thumb and show
                      //node.dataBox.x = node.thumb.x + node.thumb.
                                                                                                  //stopFocus();
width:
                      node.dataBox.x = node.thumb.x + 125;
                                                                                                  node.dot.visible = false;
                      node.dataBox.y = node.thumb.y;
                      node.dataBox.visible = true;
                                                                                                  //{\rm position} icon at dot and make visible
                                                                                                  node.icon.x = node.dot.x;
                      //goto popped state
                                                                                                  node.icon.y = node.dot.y;
                       node.nodeState = node.popped;
                                                                                                  node.icon.setTargetWidth(6);
                                                                                                  node.icon.setTarget(node.dot.x, node.dot.y);
                                                                                                  node.icon.visible = true;
               public function popin():void{}
                                                                                                  //position tag at dot and make visible
               public function close():void{
                                                                                                  node.tag.x = node.dot.x;
                                                                                                  node.tag.y = node.dot.y;
                      //trace("DroppedState close");
                                                                                                  //node.tag.visible = true;
                      //position dot at target
                      node.dot.snapToTarget();
                                                                                                  //goto picked state
                                                                                                  node.nodeState = node.picked;
                      //position icon at thumb position
                      node.icon.x = node.thumb.x;
                      node.icon.y = node.thumb.y;
                                                                                           public function deSelect():void{}
                      node.icon.setTarget(node.dot.x, node.dot.y);
                      node.icon.visible = true;
                                                                                           public function toggleSelect():void{
                                                                                                  select();
                      //hide thumb, but should i dispose of bitmap???
                      node.thumb.visible = false;
                                                                                           //public function filter():void{}
                      //go back to idle
                       node.nodeState = node.picked;
                                                                                           public function drag():void{}
                                                                                           public function drop():void{}
               public function setTarget(tgx:Number,
                                                                                           //public function dropThumb():void{}
tgy:Number):void{
                      //trace("DroppedState switch");
                                                                                           public function popout():void{}
                      //preview mainstate needs this to tile
                                                                                           public function popin():void{}
                      //node.thumb.setTarget(tgx, tgy);
                                                                                           public function close():void{}
                      //set dot target incase thumb is closed,
                      //dot it will get back in place in swarm
                                                                                           public function setTarget(tgx:Number,
                      node.dot.setTarget(tgx, tgy);
                                                                            tav:Number):void{
```

 $oldsymbol{\epsilon}$

an) •		<pre>//trace("IdleState setTarget: " + tgx + ", " +</pre>			<pre>//trace("PickedState deselect"); stopFocus();</pre>
gy);		<pre>node.dot.setTarget(tgx, tgy);</pre>			node.icon.visible = false;
	}				<pre>node.tag.visible = false; //not working</pre>
	nublia	<pre>function rePosition():void{</pre>			//restore dot
	public	//trace("IdleState rePosition");			node.dot.x = node.icon.x;
		<pre>node.dot.goToTarget();</pre>			<pre>node.dot.y = node.icon.y;</pre>
					<pre>node.dot.setTarget(node.icon.x, node.icon.y)</pre>
		<pre>node.tag.x = node.dot.x; node.tag.y = node.dot.y;</pre>			<pre>node.dot.visible = true;</pre>
	}	node.tag.y - node.dot.y,			<pre>node.nodeState = node.idle;</pre>
				}	
	public	<pre>function reSize():void{}</pre>		nuhlia	<pre>function toggleSelect():void{</pre>
	public	<pre>function startFocus():void{</pre>		public	deSelect();
	-	//trace("startFocus");		}	
		<pre>//node.dot.transform.colorTransform = redTrans-</pre>		. 1.1 ' .	5
orm;				public	<pre>function drag():void{ //trace("PickedState drag: goto dragging</pre>
		<pre>/*private function doDrawCircle():void {</pre>	state");		,,,,,,,,,
va	r bounds	s:Shape = new Shape();			stopFocus();
//	hounds c	<pre>var r:Rectangle = getRect(node.dot); graphics.beginFill(bgColor);</pre>			//position dot at icon so icon can find its
, ,	2041140.9	bounds.graphics.lineStyle(0, 0xFF3333); //red			//way back if thumb is closed
		<pre>bounds.graphics.drawRect(r.x, r.y, r.width,</pre>			<pre>node.dot.x = node.icon.x;</pre>
.height);		<pre>//bounds.graphics.endFill();</pre>			<pre>node.dot.y = node.icon.y;</pre>
		node.addChild(bounds);			<pre>//node.dataBox.alpha = 50;</pre>
	} * /	, , , ,			<pre>node.icon.startDrag();</pre>
					<pre>//node.dot.visible = false;</pre>
		<pre>node.dot.dotButton.forceOver(); node.tag.t.borderColor = 0xFF0000;</pre>			<pre>//node.icon.visible = false; //node.thumb.visible = false;</pre>
		nous roug (c. 2014c1cc1c)			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
		<pre>node.tag.x = node.dot.x;</pre>			//remove tag
		<pre>node.tag.y = node.dot.y; //node.tag.visible = true;</pre>			<pre>node.tag.visible = false;</pre>
	}	// Hode.tag.visible - tide,			//show meta & img
	public	<pre>function stopFocus():void{</pre>			<pre>//node.dataBox.visible = true;</pre>
		//trace("stopFocus");			<pre>//node.thumb.visible = true;</pre>
orm;		<pre>//node.dot.transform.colorTransform = blkTrans-</pre>			//load thumb
		<pre>//node.tag.visible = false;</pre>			<pre>node.thumb.visible = false;</pre>
			1		<pre>node.thumb.load("imgs/" + node.dat[0] + node</pre>
		<pre>node.dot.dotButton.restoreOver(); node.tag.t.borderColor = 0xCCCCCC;</pre>	dat[1]);		<pre>node.thumb.make();</pre>
	}				<pre>node.thumb.x = node.icon.x;</pre>
					<pre>node.thumb.y = node.icon.y;</pre>
	public	<pre>function startHover():void{}</pre>			<pre>node.addChild(node.thumb);</pre>
	public	<pre>function stopHover():void{}</pre>			//transition to drag state
					<pre>node.nodeState = node.dragging;</pre>
}				}	
ackage metasw	arm.node	e.nodestates {		public	<pre>function drop():void{</pre>
-		ents.TimerEvent;		_	//trace("PickedState drag: goto dragging
-	_	om.ColorTransform;	state");		ahan Fasus ()
-		<pre>ils.Timer; rm.node.NodeState;</pre>			stopFocus();
1		,			//position dot at icon so icon can find its
public	class P	ickedState implements NodeState{			//way back if thumb is closed
	private	e var node:Node;			<pre>node.dot.x = node.icon.x; node.dot.y = node.icon.y;</pre>
	-	e static var redTransform:ColorTransform;			<pre>//node.dataBox.alpha = 50;</pre>
	_	e static var bluTransform:ColorTransform; e static var blkTransform:ColorTransform;			<pre>//node.icon.startDrag(); //node.dot.visible = false;</pre>
	-	e static var whtTransform:ColorTransform;			<pre>//node.icon.visible = false;</pre>
					<pre>//node.thumb.visible = false;</pre>
	=	function PickedState(node:Node)			//romovo tog
	{	<pre>//trace("construct PickedState");</pre>			<pre>//remove tag node.tag.visible = false;</pre>
		this.node = node;			-
					//show meta & img
		<pre>redTransform = new ColorTransform(); redTransform.color = 0xFF0000;</pre>			<pre>//node.dataBox.visible = true; //node.thumb.visible = true;</pre>
		bluTransform = new ColorTransform();			
		bluTransform.color = 0x0000ff;			//load thumb
		<pre>blkTransform = new ColorTransform(); blkTransform.color = 0x231F20;</pre>			node.thumb.visible = true;
		whtTransform = new ColorTransform();	dat[1]);		<pre>node.thumb.load("imgs/" + node.dat[0] + node</pre>
		whtTransform.color = 0xFFFFFF;			<pre>node.thumb.make();</pre>
	}	function coloct() weight			node.thumb.x = node.icon.x;
	public	<pre>function select():void{}</pre>			<pre>node.thumb.y = node.icon.y; node.addChild(node.thumb);</pre>
	public	<pre>function deSelect():void{</pre>			,

```
//trace("PickedState drag: goto dragging
                                                                                            public function stopHover():void{
state");
                                                                                                    //trace("");
                       //node.icon.visible = false;
                                                                                                    //node.tag.transform.colorTransform = blkTrans-
                                                                             form;
                       //node.icon.stopDrag();
                       //node.thumb.stopDrag();
                                                                                                    //zoom back out
                                                                                                    node.icon.setTargetWidth(6);
                       //node.thumb.x = node.icon.x;
                                                                                                    node.tag.visible = false;
                       //node.thumb.y = node.icon.y;
                       node.thumb.setTarget(node.icon.x, node.icon.y);
                       //node.thumb.visible = true;
                                                                             package metaswarm.node.nodestates
                       //transition to drag state
                                                                                    import metaswarm.node.NodeState;
                       node.nodeState = node.dropped;
                                                                                    public class PoppedState implements NodeState{
               //public function dropThumb():void{}
                                                                                            private var node: Node;
                                                                                            public function PoppedState(node:Node)
               public function popout():void{}
              public function popin():void{}
                                                                                                    //trace("construct DroppedState");
                                                                                                    this.node = node;
               public function close():void{}
              public function setTarget(tgx:Number,
                                                                                            public function select():void{}
                                                                                            public function deSelect():void{}
tgy:Number):void{
                                                                                            public function toggleSelect():void{}
                       //trace("PickedState switch");
                                                                                            public function drag():void{
                       //set new icon target only
                                                                                                    //trace("DroppedState drag: goto dragging
                       node.icon.setTarget(tgx, tgy);
                                                                             state");
                                                                                                    //start dragging
                                                                                                    node.thumb.startDrag();
               public function rePosition():void{
                       //trace("PickedState rePosition");
                                                                                                    //goto to dragging state
                                                                                                    node.nodeState = node.tdragging;
                       //update dot position
                                                                                                    node.prevState = node.popped;
                      node.icon.goToTarget();
                                                                                            public function drop():void{}
//public function dropThumb():void{}
                       //snap tag to new dot position
                       node.tag.x = node.icon.x;
                       node.tag.y = node.icon.y;
                                                                                            public function popout():void{}
                       //snap dot to new icon position
                                                                                            public function popin():void{
                       //node.dot.x = node.icon.x;
                                                                                                    //trace("DroppedState popout");
                       //node.dot.y = node.icon.y;
                                                                                                    // {\tt position} \ {\tt dataBox} \ {\tt next} \ {\tt to} \ {\tt thumb} \ {\tt and} \ {\tt show}
                       //snap icon to new dot position
                                                                                                    node.dataBox.visible = false;
                       /*node.icon.x = node.dot.x;
                       node.icon.y = node.dot.y;*/
                                                                                                    //goto popped state
                                                                                                    node.nodeState = node.dropped;
               public function reSize():void{
                       //trace("PickedState reSize");
                                                                                            public function close():void{
                       node.icon.scaleToTarget();
                                                                                                    //trace("DroppedState close");
                                                                                                    //position dot at target
              public function startFocus():void{
                                                                                                    node.dot.snapToTarget();
                       //trace("startFocus");
                       node.icon.iconButton.forceOver();
                                                                                                    //position icon at thumb position
                       node.tag.x = node.icon.x;
                                                                                                    node.icon.x = node.thumb.x;
                       node.tag.y = node.icon.y;
                                                                                                    node.icon.y = node.thumb.y;
                                                                                                    node.icon.setTarget(node.dot.x, node.dot.y);
                       //node.tag.visible = true;
                       node.tag.t.borderColor = 0xFF0000;
                                                                                                    node.icon.visible = true;
               public function stopFocus():void{
                                                                                                    node.thumb.visible = false; //hmm should i dis-
                       //trace("stopFocus");
                                                                             pose of bitmap???
                       node.icon.iconButton.restoreOver();
                                                                                                    node.dataBox.visible = false;
                       //node.tag.visible = false;
                                                                                                    node.nodeState = node.picked;
                       node.tag.t.borderColor = 0xCCCCCC;
                                                                                            public function setTarget(tgx:Number,
               public function startHover():void{
                       //trace("");
                                                                             tgy:Number):void{
                       //node.tag.transform.colorTransform = whtTrans-
                                                                                                    //trace("DroppedState switch");
form;
                                                                                                    //preview mainstate needs this to tile
                                                                                                    //node.thumb.setTarget(tgx, tgy);
                       //zoom in icon
                       node.icon.setTargetWidth(12);
                                                                                                    //set dot target incase thumb is closed,
                       node.tag.x = node.icon.x;
                       node.tag.y = node.icon.y;
                                                                                                    //{\rm dot} it will get back in place in swarm
                       node.tag.visible = true;
                                                                                                    node.dot.setTarget(tgx, tgy);
```

```
node.dot.v = node.dataBox.v;*/
                                                                                                   //snap data box to thumb while dragging
               public function rePosition():void{
                      //trace("DroppedState rePosition");
                                                                                                   node.dataBox.x = node.thumb.x + 125;
                                                                                                   node.dataBox.y = node.thumb.y;
                      //preview mainstate need this to tile
                      node.thumb.goToTarget();
                                                                                           public function reSize():void{
                      //snap databox to thumb location
                                                                                                   //trace("DraggingState reSize");
                      node.dataBox.x = node.thumb.x + 125;
                                                                                                   node.icon.scaleToTarget(); //???
                      node.dataBox.y = node.thumb.y;
                                                                                           public function startFocus():void{}
                                                                                           public function stopFocus():void{}
               public function startFocus():void{}
               public function stopFocus():void{}
                                                                                           public function startHover():void{}
               public function reSize():void{} //icon is not visible
                                                                                           public function stopHover():void{}
so don't bother
              public function startHover():void{} //change these
names to hover or smth
              public function stopHover():void{}
                                                                            package metaswarm.scrap {
                                                                                   import flash.display.DisplayObject;
                                                                                import flash.display.Graphics;
                                                                                import flash.display.Shape;
package metaswarm.node.nodestates {
                                                                                    import flash.display.Sprite;
                                                                                    import metaswarm.node.components.Particle;
       import metaswarm.node.NodeState;
       public class ThumbDraggingState implements NodeState{
                                                                                    public class Box extends Particle{
               private var node: Node;
                                                                                           private static const Z_EASE:Number = 1;
               function ThumbDraggingState(node:Node)
                                                                                           private static const Z SNAP:Number = 1;
                       //trace("construct DraggingState");
                                                                                           //Target zoom
                                                                                           private var tgw:Number;
                      this.node = node;
                                                                                           //radial distance to zoom target
              public function select():void{}
                                                                                           private var dw:Number;
               public function deSelect():void{}
                                                                                           //zoom velocity
              public function toggleSelect():void{}
                                                                                           private var vw:Number;
              public function drag():void{}
                                                                                           public function Box(tgw:Number=0,
                                                                                                                                 dw:Number=0,
              public function drop():void{
                                                                                                                                 vw:Number=0):void
                      //trace("PickedState drag: goto dragging
state");
                                                                                                   this.x = 0;
                      //node.icon.visible = false;
                                                                                                   this.y = 0;
                                                                                                   this.tgw = tgw;
                      //node.icon.stopDrag();
                                                                                                   this.dw = dw;
                      node.thumb.stopDrag();
                                                                                                   this.vw = vw;
                                                                                                   init();
                      //node.thumb.x = node.icon.x;
                      //node.thumb.y = node.icon.y;
                                                                                           public function init():void {
                      node.thumb.setTarget(node.thumb.x, node.
                                                                                                   visible = false;
thumb.y);
                                                                                                   focusRect = true;
                      //node.thumb.visible = true;
                                                                                                   tabEnabled = true;
                                                                                                   doubleClickEnabled = true;
                      //goto dropped state
                                                                                                   //mouseEnabled = false;
                      node.nodeState = node.prevState;
                                                                                           public function load():void{
               //public function dropThumb():void{}
                                                                                           //might depend on load later, so keep here
              public function popout():void{}
                                                                                           //rather than put in init
               public function popin():void{}
                                                                                           public function make():void{
                                                                                                   this.graphics.beginFill(0x0000ff);
               public function close():void{}
                                                                                                   this.graphics.lineStyle(0);
                                                                                                   this.graphics.drawRect(0, 0, 10, 10);
               public function setTarget(tgx:Number,
                                                                                                   this.graphics.endFill();
tgy:Number):void{
                      //trace("DraggingState switch");
                                                                                           public function scaleToTarget():void {
                      //node.dot.setTarget(tgx, tgy);
                                                                                                   dw = tgw - this.width;
                                                                                                   if (Math.abs (dw - Z SNAP) <1) {
               public function rePosition():void{
                                                                                                          this.width = tgw;
                      //trace("DraggingState rePosition");
                                                                                                          this.height = tgw;
                      //node.dot.goToTarget();
                                                                                                          //removeEventListener(Event.Enter FRAME,
                      //node.thumb.x = node.icon.x;
                                                                            onEnterFrame);
                      //node.thumb.y = node.icon.y;
                                                                                                   }else{
                                                                                                          vw = dw * Z EASE;
                      // {\tt dragging \ icon \ so \ snap \ dot \ to \ icon}
                                                                                                          this.width += vw:
                      /*node.dot.x = node.dataBox.x;
                                                                                                          this.height += vw;
```

```
public function setTargetWidth(a:Number) {
                      tqw = a;
package metaswarm.scrap {
        import flash.display.DisplayObject;
    import flash.display.Graphics;
    import flash.display.Shape;
        import flash.display.Sprite;
        import metaswarm.node.components.Particle;
        public class Cross extends Particle{
               public function Cross():void{
                      init();
               public function init():void {
                      //visible = true;
                      //focusRect = true;
                      //tabEnabled = true;
               public function load():void{
               //might dep on load later, so leave here!
               public function make():void{
                      //trace("cross.make()");
                      this.graphics.lineStyle(0);
                      //this.graphics.lineStyle(1, 0xFF0000);
                       this.graphics.moveTo(x, y-3);
                      this.graphics.lineTo(x, y+3);
                      this.graphics.moveTo(x-3, y);
                      this.graphics.lineTo(x+3, y);
                      /*trace("visible: " + this.visible);
                      trace("x: " + this.x);
                      trace("y: " + this.y);*/
package metaswarm.scrap {
        import flash.display.*;
        import flash.events.*;
    import flash.net.URLRequest;
    import metaswarm.node.components.thumb.BitmapLoadr;
    import metaswarm.node.components.Particle;
        public class Image extends Particle{
               var path:String;
               var ldr:BitmapLoadr;
               var bmp:Bitmap;
               var bmpLoaded:Boolean;
               public function Image():void {
                      ldr = new BitmapLoadr(this);
                      bmp = new Bitmap();
                      bmpLoaded = false;
                      init();
               public function init():void {
                      addChild(ldr);
                      //l.loadImg("fake string");
                      visible = false;
                      //focusRect = true;
                       //tabEnabled = true;
               public function load(path:String):void{
                      this.path = path;
               public function make():void{
                      if(!bmpLoaded){
                              ldr.loadBmp(path);
                              bmpLoaded = true;
```

```
package metaswarm.scrap {
  import flash.display.*;
  import flash.text.*;
  import flash.events.*
  import flash.net.*;
  public class Loadr extends Sprite {
        var loadProg:TextField;
       var gnurbwurk:Metaswarm;
   public function Loadr(gnurbwurk:Metaswarm)
               this.gnurbwurk = gnurbwurk;
               this.loadProg = new TextField();
               init();
   private function init():void
       //loadProg.width = stage.stageWidth;
       //loadProg.height = stage.stageHeight;
       addChild(loadProg);
       public function preload():void {
               var loader:URLLoader = new URLLoader();
               // Instruct the loader to read the file as plain text -
This line is not
               // necessary because the dataFormat is DataFormat.TEXT
by default.
               loader.dataFormat = URLLoaderDataFormat.TEXT;
               // Register an event handler for when the data is fin-
ished downloading
               loader.addEventListener(Event.COMPLETE, loadComplete);
               // Listen for the progress event to check download
progress
      loader.addEventListener(ProgressEvent.PROGRESS, han-
dleProgress);
               // Load the HTML text from the example.html file
               loader.load(new URLRequest("../assets/index4.txt"));
       private function handleProgress(event:ProgressEvent):void
        // Calculate the percentage by multiplying the loaded-to-total
        // ratio by 100
       var percent:Number = Math.round(event.bytesLoaded
                                      / event.bytesTotal * 100 );
       loadProg.text = " Loaded: " + event.bytesLoaded + "\n"
                        + " Total: " + event.bytesTotal + "\n"
                        + "Percent: " + percent;
               //trace(loadProg.text);
   private function loadComplete(event:Event):void
               var loader:URLLoader = URLLoader(event.target);
               \ensuremath{//} assign to output. The data property of the URLLoader
is the file contents.
               parseData(loader.data);
       private function parseData(loadedData:String):void
               var parsedData:Array = new Array();
               //var parsedMenu:Array = new Array();
               // Split the string into an array of words using a
space as the delimiter.
               var lines:Array = loadedData.split("\n");
               var words:Array = new Array();
               \ensuremath{//} Loop through the array and do something with each
```

```
var xoff:uint = s/4;
word
               // In this example, just output the values.
                                                                                                   var yoff:uint = s/4;
               for ( var i:int = 0; i < lines.length; i++ ) {
                       words = lines[i].split("\t");
                                                                                                   for(var i:uint=0; i<cnt; i++){</pre>
                       parsedData.push(words);
                                                                                                           //trace("subset.nodes[i].thumb.x: " +
                       //trace(parsedData[i][1]);
                                                                             subset.nodes[i].meta.x);
                                                                                                           //trace("subset.nodes[i].thumb.y: " +
               //gnurbwurk.parsedData = parsedData;
                                                                             subset.nodes[i].meta.y);
               //gnurbwurk.preloadComplete(parsedData);
                                                                                                           var px:Number = subset.nodes[i].thumb.x;
               gnurbwurk.load(parsedData);
                                                                                                           var py:Number = subset.nodes[i].thumb.y;
                //gnurbwurk.main.makePatterns();
                                                                                                           subset.nodes[i].thumb.setPrev(px, py);
                                                                                                           var tgx:Number = xoff + xsp*(i%cols);
                                                                                                           var tgy:Number = yoff + ysp*Math.floor(i/
 }//end class
}//end pkg
                                                                            cols);
package metaswarm.subset
                                                                                                           subset.nodes[i].popin();
        import flash.display.DisplayObject;
                                                                                                           subset.nodes[i].thumb.setTarget(tgx,
        import flash.display.Sprite;
                                                                            tqy);
                                                                                                           //subset.nodes[i].tag.setTarget(tgx,
        import flash.display.Graphics;
    import flash.display.Shape;
                                                                             tqy);
        public class Subset extends Sprite{
              public var nodes:Arrav;
                                                                                            public function restore():void{
               public var tiler:SubsetTiler;
                                                                                                    //trace("tiler.restore");
               //var count:uint;
                                                                                                    var cnt:uint = subset.nodes.length;
                                                                                                   for(var i:uint=0; i<cnt; i++) {</pre>
               //constructor:
                                                                                                           //trace("subset.nodes[i].thumb.x: " +
              public function Subset(){
                                                                             subset.nodes[i].meta.x);
                       nodes = new Array();
                                                                                                           //trace("subset.nodes[i].thumb.y: " +
                       tiler = new SubsetTiler(this);
                                                                             subset.nodes[i].meta.y);
                                                                                                           var tgx:Number = subset.nodes[i].thumb.
                                                                            px;
              public function tile():void {
                                                                                                           var tgy:Number = subset.nodes[i].thumb.
                       //trace("tile");
                                                                            py;
                       tiler.grid();
                                                                                                           subset.nodes[i].popin();
                                                                                                           subset.nodes[i].thumb.setTarget(tgx,
                                                                            tqy);
               public function restore():void {
                                                                                                           //subset.nodes[i].tag.setTarget(tgx,
                       //trace("tile");
                                                                             tgy);
                       tiler.restore();
              public function disable():void {
                                                                                    }//end class
                       trace("disable");
                                                                             }//end pkg
                       for(var i:uint=0; i<nodes.length; i++){</pre>
                                                                             package metaswarm.swarm
                              nodes[i].visible = false;
                                                                                    import flash.display.DisplayObject;
                                                                                    import flash.display.Sprite;
                                                                                    import flash.display.Graphics;
                                                                                 import flash.display.Shape;
       }//end class
                                                                                 import metaswarm.swarm.swarmstates.BrowningState;
}//end pkg
                                                                                 import metaswarm.swarm.swarmstates.GridState;
                                                                                 import metaswarm.swarm.swarmstates.LorenzeState;
package metaswarm.subset
        import flash.display.DisplayObject;
                                                                                 import metaswarm.swarm.swarmstates.ScatterState;
                                                                                 import metaswarm.swarm.swarmstates.TypeState;
        import flash.display.Sprite;
       import flash.display.Graphics;
                                                                                 import metaswarm.swarm.swarmstates.MapState;
    import flash.display.Shape;
                                                                                 import metaswarm.subset.Subset;
       public class SubsetTiler extends Sprite{
                                                                                    public class Swarm extends Sprite{
                                                                                            //omit constants for now
               public var subset: Subset;
               //var tileSize:uint;
                                                                                            //Except for TextField and Video objects.
                                                                                            //a display object with no content has a
               //constructor:
              public function SubsetTiler(subset:Subset) {
                                                                                            // {\rm width} of 0, even if you try to set width
                                                                                            //{\hbox{to a different value, so need these}}
                       this.subset = subset;
                       //this.tileSize = 100; //later define dynamically
                                                                                            public var w:int;
                                                                                            public var h:int;
               //this is beastly for now. it sets target of both
                                                                                            //internal composite objects
databox
                                                                                            public var nodes:Array;
               //and tag, regardless of whether dropped or picked
                                                                                            public var nodeCnt:uint;
called it.
                                                                                            public var propCnt:uint;
               //and s should not be hard coded!!
                                                                                            public var sortOrder:uint;
              public function grid():void{
                       //trace("tiler.grid");
                                                                                            //internal states
                       var s:uint = 200;
                                                                                            public var lorenze:SwarmState;
                       var cnt:uint = subset.nodes.length;
                                                                                            public var browning:SwarmState;
                       //trace("cnt: " + cnt);
                                                                                            public var scatter:SwarmState;
                       //var xMargin:uint = s/4;
                                                                                            public var grid:SwarmState;
               //var yMargin:uint = s/4;
                                                                                            public var type:SwarmState;
                                                                                            public var map:SwarmState;
                       var xsp:uint = s;
                       var vsp:uint = s;
                       var cols:uint = Math.ceil(((s+ysp) * cnt) /
                                                                                            //internal state holder
1080);
                                                                                            public var swarmState:SwarmState;
                       //trace("cols: " + cols);
```

```
//var swarmLoader:SwarmLoader;
                                                                                          public function addBy(t:String):void {
               //var patternMaker:PatternMaker;
                                                                                                  for(var i:uint=0; i<nodeCnt; i++) {</pre>
                                                                                                         if(nodes[i].tag.t.text == t){
               //constructor:
              public function Swarm() {
                                                                                                                 nodes[i].select();
                      //instantiate composite objs
                      nodes = new Array();
                                                                                          public function filterBy(t:String):void {
                      lorenze = new LorenzeState(this);
                                                                                                  for(var i:uint=0; i<nodeCnt; i++) {</pre>
                      browning = new BrowningState(this);
                                                                                                         if(nodes[i].tag.t.text != t){
                      scatter = new ScatterState(this);
                                                                                                                 //trace("deSelect");
                      grid = new GridState(this);
                                                                                                                 nodes[i].deSelect();
                      type = new TypeState(this);
                      map = new MapState(this);
                                                                                          public function popin(node:Node):void{
                      init();
                                                                                                  node.popin();
              private function init():void{
                      //set intial state and show unique
                                                                                          public function popout(node:Node):void{
                      //tags for that state
                                                                                                  node.popout();
                      swarmState = grid;
                      sortOrder = 2;
                                                                                          public function close(node:Node):void{
                      showUniqueTags();
                                                                                                  node.close();
              //these need to be changed
                                                                                          public function drag(node:Node):void{
              public function preview():void {
                                                                                                  node.drag();
                      var dropped:Subset = new Subset();
                      //var undropped:Subset = new Subset();
                      for(var i:uint=0; i<nodeCnt; i++){</pre>
                                                                                          public function drop(node:Node):void{
                              if(nodes[i].isDropped() || nodes[i].
                                                                                                  node.drop();
isPopped()){
                                     dropped.nodes.push(nodes[i]);
                                                                                          public function dropSelected():void{
                                                                                                  for(var i:uint=0; i<nodeCnt; i++) {</pre>
                              }else{
                                     //undropped.push(nodes[i]);
                                                                                                         if(nodes[i].isPicked()){
                                     nodes[i].visible = false;
                                                                                                                 nodes[i].drop();
                      dropped.tile();
                      //undropped.disable();
                                                                                          public function startHover(node:Node):void{
                                                                                                  node.startHover();
              public function explore():void {
                      var dropped:Subset = new Subset();
                                                                                          public function stopHover(node:Node):void{
                      //var undropped:Subset = new Subset();
                                                                                                  //trace("swarm.stopHover");
                      for(var i:uint=0; i<nodeCnt; i++) {</pre>
                                                                                                  node.stopHover();
                              if(nodes[i].isDropped() || nodes[i].
isPopped()){
                                     dropped.nodes.push(nodes[i]);
                                                                                          public function startFocusBy(t:String):void {
                                                                                                  for(var i:uint=0; i<nodeCnt; i++) {</pre>
                                     //undropped.push(nodes[i]);
                                                                                                         if(nodes[i].tag.t.text == t){
                                     nodes[i].visible = true;
                                                                                                                 nodes[i].startFocus();
                      dropped.restore();
                      //undropped.disable();
                                                                                          public function startSelectByFocus(t:String):void {
                                                                                                  for(var i:uint=0; i<nodeCnt; i++) {</pre>
               //methods
                                                                                                         if(nodes[i].tag.t.text == t){
              public function select(node:Node):void {
                                                                                                                 nodes[i].startFocus();
                      clearSelection();
                      node.select();
               public function add(node:Node):void {
                                                                                          public function startFilterByFocus(t:String):void {
                      node.select();
                                                                                                  for(var i:uint=0; i<nodeCnt; i++){</pre>
               public function remove(node:Node):void {
                                                                                                         if(nodes[i].tag.t.text == t && nodes[i].
                      node.deSelect();
                                                                           isPicked()){
                                                                                                                 nodes[i].startFocus();
               public function toggleSelect(node:Node):void {
                      node.toggleSelect();
              public function clearSelection():void {
                      for(var i:uint=0; i<nodeCnt; i++) {</pre>
                                                                                          public function stopFocusBy():void {
                              nodes[i].deSelect();
                                                                                                  for(var i:uint=0; i<nodeCnt; i++) {</pre>
                                                                                                         nodes[i].stopFocus();
              public function selectBy(t:String):void {
                      clearSelection();
                                                                                          public function printOutNodes():void{
                      for(var i:uint=0; i<nodeCnt; i++) {</pre>
                                                                                                 if(nodes[i].tag.t.text == t){
                                                                            nodes[i].select();
                                                                                                  for(var i:uint=0; i<nodeCnt; i++) {</pre>
                                                                                                         trace("node " + i + ": ");
```

		for(v	ar j:uint=0; j <propcnt;< th=""><th></th><th>public</th><th>c class</th><th>S SwarmLoader extends Sprite {</th><th></th></propcnt;<>		public	c class	S SwarmLoader extends Sprite {	
odes[i].xpos[0][j] +	", " + nodes[<pre>trace("xpos,ypos " + i].ypos[0][j]);</pre>] + ": " +		private	e var _swarm:Swarm;	
		}			publ	lic func	nction SwarmLoader(swarm:Swarm)	
	}					{	this. swarm = swarm;	
	public	function rePo					<pre>//this.loadProg = new TextField();</pre>	
			rm rePosition"); ut=0; i <nodecnt; i++){<="" td=""><td></td><td></td><td></td><td><pre>init();</pre></td><td></td></nodecnt;>				<pre>init();</pre>	
			[i].rePosition();		}			
	}	}			priv	zate fun	unction init():void	
						{		
	public	function reSi //trace("swar					Prog.width = stage.stageWidth; Prog.height = stage.stageHeight;	
		for(var i:uir	it=0; i <nodecnt; i++){<="" td=""><td></td><td></td><td></td><td>hild(loadProg);</td><td></td></nodecnt;>				hild(loadProg);	
		nodes	[i].reSize();		}			
	}	,					<pre>function load(dat:Array):void{</pre>	
	nildua	function incr	Sort():void{				<pre>//trace("swarmloader load"); swarm.nodeCnt = dat.length;</pre>	
	F	//trace("incr	Sort");				_swarm.propCnt = dat[0].length;	
			<pre>< propCnt-1) { rder++;</pre>				<pre>//create nodes and add each to display list for(var i:uint=0; i< swarm.nodeCnt; i++) {</pre>	
		}else{	ruer , ,				var tmp:Node = new Node();	
		sort0	rder = 0;				var nodeLoader:NodeLoader = new NodeLoade:	r(tmp);
		for(var i:uir	t=0; i <nodecnt; i++){<="" td=""><td></td><td></td><td></td><td><pre>nodeLoader.load(dat[i]); //maybe call afte</pre></td><td>er push</td></nodecnt;>				<pre>nodeLoader.load(dat[i]); //maybe call afte</pre>	er push
			rm.nodes[i].tag.setTxt(? [i].tag.incr();	1);			_swarm.nodes.push(tmp);	
		}	[I].cag.Incl(),			}	ı	
		setPattern(); showUniqueTag				_	<pre>function make():void{ //swarm.patternMaker.make(); //decouple from swar</pre>	m
	}	Showoniquerag	JS () ,				for(var i:uint=0; i<_swarm.nodeCnt; i++){	111
	public	function decr			swarm.no	odes[i])	<pre>var nodeLoader:NodeLoader = new NodeLoade:</pre>	r (_
		if(sortOrder			Swarm.iic	Jues[1]/	nodeLoader.make();	
		sort0	rder;			ı	}	
			rder = propCnt-1;			public:	<pre>function go():void{</pre>	
		for (war i wir	it=0; i <nodecnt; i++){<="" td=""><td></td><td></td><td></td><td><pre>//swarm.swarmState = swarm.grid; //swarm.sortOrder = 0;</pre></td><td></td></nodecnt;>				<pre>//swarm.swarmState = swarm.grid; //swarm.sortOrder = 0;</pre>	
			rm.nodes[i].tag.setTxt(]	1);			for(var i:uint=0; i<_swarm.nodeCnt; i++){	
		nodes	<pre>[i].tag.decr();</pre>		0110 mm n	odoo[:1)	var nodeLoader:NodeLoader = new NodeLoade	r (_
		setPattern();			swarm.no	odes[1])	nodeLoader.go();	
	1	showUniqueTag	rs();				_swarm.addChild(_swarm.nodes[i]);	
	1						//swarm.setPattern();	
	public	function show	UniqueTags():void{			}		
	Public		it=0; i <nodecnt; i++){<="" td=""><td></td><td>}//end</td><td>d class</td><td>5</td><td></td></nodecnt;>		}//end	d class	5	
Lq;		nodes	<pre>[i].tag.visible = nodes</pre>	[i].tag.isU-	}//end p	_	varm.swarm {	
-9/		}			_			
	}					, , -	<pre>interface interface SwarmState{</pre>	
						pastro	Interface Swarmscate (
*****	//deleg	gated tas ******	*****				<pre>//internal interface methods //function incrSort():void;</pre>	
	public	function incr	State():void{				<pre>//function decrSort():void;</pre>	
			egate incrState");				function incrState():void;	
	}	swarmState.ir	crState();				<pre>function decrState():void; function setPattern():void;</pre>	
	public	function decr				}	,	
		<pre>//trace("dele swarmState.de</pre>	egate decrState"); ecrState();) package	metaswa	warm.swarm.patternmaker {	
	}		.,.		_	import :	<pre>flash.display.*;</pre>	
	public	function setP	attern():void{ egate swarm switch");			-	<pre>flash.utils.*; flash.events.*;</pre>	
		swarmState.se	_			import :	<pre>flash.geom.*;</pre>	
	}					-	<pre>flash.text.*; metaswarm.swarm.patterns.GeoMapper;</pre>	
}//end	swarm					import 1	<pre>metaswarm.swarm.patterns.LorenzeMaker;</pre>	
//end pkg						-	<pre>metaswarm.swarm.patterns.TypeMaker; metaswarm.swarm.Swarm;</pre>	
_						-	metaswarm.swarm.patterns.GridMaker;	
ackage metaswa import flash.						//class	s needs to be broken up	
import flash.	text.*;							
<pre>import flash.</pre>		*				public	class PatternMaker extends Sprite{	
import metas	warm.nod	de.NodeLoader;					<pre>private var _swarm:Swarm;</pre>	_
import metas	warm.*;				calize t		<pre>//var typeMaker:TypeMaker; //don't know why i don stion</pre>	't lo-

```
public function PatternMaker(swarm:Swarm):void{
                       this. swarm = swarm;
                       //this.typeMaker = new TypeMaker();
               public function make():void
                       //property count is hardcoded for now so \operatorname{can'} t
                       //add/remove patterns w/out changing
                       for(var j:uint=0; j<_swarm.propCnt; j++) {</pre>
                        sortBy(j);
                         setUniqueTags(j);
                         grid(0, j);
                         scatter(1, j);
                         browning(2, j);
                         lorenze(3, j);
                        type(4,j);
                        map(5,j);
               public function setUniqueTags(tid:uint):void{
                       var prevTag:String = new String();
                       var curTag:String = new String();
                       prevTag = "";
                       for(var i:uint=0; i<_swarm.nodeCnt; i++) {</pre>
                               curTag = _swarm.nodes[i].dat[tid];
                               if(curTag == prevTag){
                                       _swarm.nodes[i].tag.uniqArr[tid]
= false;
                               }else{
                                       _swarm.nodes[i].tag.uniqArr[tid]
= true;
                              prevTag = _swarm.nodes[i].dat[tid];
               public function type(pid:uint, sid:uint):void
                       //trace("PatternMaker->type()");
                       //generate coordinates
                       var typeMaker:TypeMaker = new TypeMaker();
                       var points:Array = typeMaker.make("me++", _
swarm.nodeCnt, _swarm.w);
                       //map coordinates to nodes positions
                       for(var i:uint=0; i<_swarm.nodeCnt; i++){</pre>
                               _swarm.nodes[i].xpos[pid][sid] =
points[i][0];
                               _swarm.nodes[i].ypos[pid][sid] =
points[i][1];
               //this is beastly for now
               public function grid(pid:uint, sid:uint):void
                       //generate coordinates
                       var gridMaker:GridMaker = new GridMaker();
                       var points:Array = gridMaker.make(_swarm.no-
deCnt, _swarm.w, _swarm.h);
                       //map coordinates to nodes positions
                       for(var i:uint=0; i<_swarm.nodeCnt; i++) {</pre>
                               _swarm.nodes[i].xpos[pid][sid] =
points[i][0];
                               _swarm.nodes[i].ypos[pid][sid] =
points[i][1];
               public function scatter(pid:uint, sid:uint):void
                       for(var i:uint=0; i< swarm.nodeCnt; i++) {</pre>
                               _swarm.nodes[i].xpos[pid][sid] = Math.
random() * swarm.w;
                               _swarm.nodes[i].ypos[pid][sid] = Math.
random()*_swarm.h;
```

```
//Lorenze Attractor Pattern-just guessed on random
range
               public function lorenze(pid:uint, sid:uint):void
                       //generate coordinates
                      var lorenzeMaker:LorenzeMaker = new LorenzeMak-
er();
                      var points:Array = lorenzeMaker.make(_swarm.no-
deCnt, _swarm.w, _swarm.h);
                       //map coordinates to nodes positions
                       for(var i:uint=0; i<_swarm.nodeCnt; i++){</pre>
                              _swarm.nodes[i].xpos[pid][sid] =
points[i][0];
                              swarm.nodes[i].ypos[pid][sid] =
points[i][1];
               /*//genBrowning
               public function browning(pid:uint, sid:uint):void{
                       var range:int = 20;
                       swarm.nodes[0].xpos[pid][sid] = Math.
random()*swarm.w;
                       swarm.nodes[0].ypos[pid][sid] = Math.
random()*swarm.h;
                       for(var i:uint=1; i<swarm.nodeCnt; i++) {</pre>
                              if(swarm.nodes[i].tag.dat[sid] == swarm.
nodes[i-1].tag.dat[sid]){
                                      swarm.nodes[i].xpos[pid][sid] =
swarm.nodes[i-1].xpos[pid][sid] + (Math.random()-.5)*range;
                                      swarm.nodes[i].ypos[pid][sid] =
swarm.nodes[i-1].ypos[pid][sid] + (Math.random()-.5)*range;
                              }else{
                                      swarm.nodes[i].xpos[pid][sid] =
Math.random()*swarm.w;
                                      swarm.nodes[i].ypos[pid][sid] =
Math.random()*swarm.h;
               } * /
               //genBrowning
               public function browning(pid:uint, sid:uint):void
                      var range:int = 25;
                       _swarm.nodes[0].xpos[pid][sid] = _swarm.w/2;
                        swarm.nodes[0].ypos[pid][sid] = _swarm.h/2;
                       for(var i:uint=1; i<_swarm.nodeCnt; i++){</pre>
                               _swarm.nodes[i].xpos[pid][sid] = _swarm.
nodes[i-1].xpos[pid][sid] + (Math.random()-.5)*range;
                               _swarm.nodes[i].ypos[pid][sid] = _swarm.
nodes[i-1].ypos[pid][sid] + (Math.random()-.5)*range;
               //genBrowning
               public function map(pid:uint, sid:uint):void
                       var geoMapper:GeoMapper = new GeoMapper();
                       //var coordinates:Array = new Array();
                      var latIndex:Number = 6;
                      var longIndex:Number = 5;
                       //for each node
                       for(var i:uint=1; i<_swarm.nodeCnt; i++) {</pre>
                              var geoPoint:Array = [_swarm.nodes[i].
dat[longIndex], _swarm.nodes[i].dat[latIndex]];
                              //coordinates.push(geoPoint);
                              //convert node.dat to screen coordinates
                               var point:Array = geoMapper.
geoToScreen(geoPoint, _swarm.h/2);
                              //assign to node pos
                              swarm.nodes[i].xpos[pid][sid] =
point[0];
                              _swarm.nodes[i].ypos[pid][sid] =
point[1];
               //bubble sort. put in util class later
```

public function sortBy(tid:uint):void

72

```
var a:Node = new Node();
                                                                                              //returns a matrix of 2d coordinates
                       var b:Node = new Node();
                       for(var i:uint=0; i<_swarm.nodeCnt; i++){</pre>
                                                                                              public function geoToScreen(geoPoint:Array,
                              for(var j:uint=0; j<_swarm.nodeCnt; j++)</pre>
                                                                              mapHeight:uint):Array
                                                                                                      var long:Number = geoPoint[0];
                                       //assign neighbooring Nodes to
tmp nodes
                                                                                                     var lat:Number = geoPoint[1];
                                      a = _swarm.nodes[i];
                                                                                                     var mapWidth:uint = mapHeight*2;
                              b = swarm.nodes[j];
                                      //if two strings not alphabetized
                                                                                                      var px:Number = (long + 180) * (mapWidth/360);
                                                                                                      var py:Number = (lat - 90) * (mapHeight/180) *
then swap them
                               if(a.tag.dat[tid] > b.tag.dat[tid]){
                                               //trace("swap: " + a.tag.
                                                                                                      var point:Array = [px, py];
dat[sid] + " with " + b.tag.dat[sid]);
                                               swarm.nodes[i] = b;
                                                                                                      return point;
                                               swarm.nodes[j] = a;
                                                                                      }//end class
                                                                              }//end pkg
                                                                              package metaswarm.swarm.patterns {
                                                                               import flash.display.*;
        }//end class
                                                                              import flash.utils.*;
                                                                              import flash.events.*;
}//end pkg
                                                                              import flash.geom.*;
package metaswarm.swarm.patterns {
                                                                              import flash.text.*;
import flash.display.*;
import flash.utils.*;
                                                                                      public class GridMaker extends Sprite {
import flash.events.*;
import flash.geom.*;
                                                                                              // {\tt minimum} \ {\tt size} \ {\tt that} \ {\tt text} \ {\tt can} \ {\tt be} \ {\tt rastered} \ {\tt at} \ {\tt to} \ {\tt provide}
import flash.text.*;
                                                                                              //adequate sampling for large display
                                                                                              //private static const FONTSIZE:Number = 42;
        public class BrowningMaker extends Sprite {
                                                                                              public function GridMaker():void{
                //lorenze constants - do not change
               //private static const XXX:Number = xxx;
                                                                                              //returns a matrix of 2d coordinates
               public function BrowningMaker():void{
                                                                                              public function make(numOfPoints:uint, w:uint,
                                                                              h:uint):Array
                //returns a matrix of 2d coordinates
                                                                                                     var points:Arrav = new Arrav();
               public function make(numOfPoints:uint, w:uint,
                                                                                                     //trace("for pattern: " + pid + ", and sor-
h:uint):Array
                                                                              tOrder: " + sid + ": ");
                       var points:Array = new Array();
                                                                                                      var xMargin:uint = 0;
                                                                                              var yMargin:uint = 0;
                                                                                                      var cols:uint = Math.round(Math.
                       return points;
                                                                              sqrt(numOfPoints));
                                                                                                      var xsp:uint = w/(cols+xMargin);
        }//end class
                                                                                                      var ysp:uint = h/(cols+yMargin);
}//end pkg
                                                                                                      var xoff:uint = (w - xsp*cols)/2;
package metaswarm.swarm.patterns {
                                                                                                     var yoff:uint = (h - ysp*cols)/2;
import flash.display.*;
import flash.events.*;
                                                                                                      for(var i:uint=0; i<numOfPoints; i++) {</pre>
import flash.geom.*;
import flash.text.*;
                                                                                                             var px:Number = xoff + xsp*(i%cols);
import flash.utils.*;
                                                                                                             var py:Number = yoff + ysp*Math.floor(i/
                                                                              cols);
        public class GeoMapper extends Sprite {
                                                                                                             var nextPoint:Array = [px, py]
                                                                                                             points.push(nextPoint);
               // map dependant
               private static const LONG INIT: Number = -180;
               private static const LAT INIT:Number = 90;
                                                                                                      return points;
               private static const LONG FINAL: Number = 180;
               private static const LAT_FINAL:Number = -90;
               private static const LONG_DELTA:Number = 360;
                                                                                      }//end class
               private static const LAT_DELTA:Number = 180;
                                                                              }//end pkg
                                                                              package metaswarm.swarm.patterns {
               public function GeoMapper():void{
                                                                               import flash.display.*;
                                                                              import flash.utils.*;
                                                                              import flash.events.*;
                //returns a matrix of 2d coordinates
                                                                              import flash.geom.*;
               public function map(geoPoints:Array,
                                                                              import flash.text.*;
mapHeight:uint):Array
                                                                                      public class LorenzeMaker extends Sprite {
                       var points:Array = new Array();
                                                                                              //lorenze constants - do not change
                       //lorenze iterative functions
                                                                                              private static const H:Number = 0.01;
                       for(var i:uint=0; i<geoPoints.length; i++) {</pre>
                                                                                              private static const A:Number = 10.0;
                               //set next point coordinate
                                                                                              private static const B:Number = 28.0;
                               var nextPoint:Array =
                                                                                              private static const C:Number = 8.0/3.0;
geoToScreen(geoPoints[i], mapHeight);
                              points.push(nextPoint);
                                                                                              public function LorenzeMaker():void{
                       return points;
```

```
//returns a matrix of 2d coordinates
                                                                                           //duplicateMovieClip(this. parent.dot, imageId, 1);
               public function make(numOfPoints:uint, w:uint,
                                                                                           this. parent.attachMovie("dot", i, i);
h:uint):Array
                                                                                           //trace(mc);
                                                                                           this._parent[i]._x = imageX;
this._parent[i]._y = imageY;
                       var points:Array = new Array();
                                                                                           //setProperty(this._parent.imageId, _x, imageX);
                       var sf:Number = h/300; //scale factor. adjust by
                                                                                           //setProperty(this._parent.imageId, _y, imageY);
steping divisor by 100
                       var xoff:uint = w/2;
                       var yoff:uint = h/2;
                       var x0:Number = (Math.random() * 10) / 10;
                                                                            package metaswarm.swarm.patterns {
                       var y0:Number = Math.random() * 20;
                                                                             import flash.display.*;
                       var z0:Number = Math.random() * 10;
                                                                             import flash.utils.*;
                                                                            import flash.events.*;
                       //lorenze iterative functions
                                                                            import flash.geom.*;
                       for(var i:uint=0; i<numOfPoints; i++){</pre>
                                                                            import flash.text.*;
                              var x1:Number = x0 + H * A * (y0-x0);
                              var y1:Number = y0 + H * (x0 * (B - z0))
                                                                                    public class TypeMaker extends Sprite {
- v0);
                              var z1:Number = z0 + H * (x0 * y0 - C *
                                                                                            //minimum size that text can be rastered at to provide
                                                                                           //adequate sampling for large display
z0);
                                                                                           private static const FONTSIZE:Number = 42;
                              //set next point coordinate
                              var px:Number = (x0 * 10) * sf + xoff;
                                                                                           public function TypeMaker():void{
                              var py:Number = (y0 * 10) * sf + yoff;
                              var nextPoint:Array = [px, py]
                                                                                           //given a text string, number of times to sample it,
                              points.push(nextPoint);
                                                                             and desired width of output
                              //feedback
                                                                                           //returns a matrix of 2d coordinates
                              x0 = x1;
                                                                                           public function make(strToSample:String,
                              y0 = y1;
                                                                            numOfSamples:uint, outputWidth:Number):Array
                                                                                                   var coords:Array = new Array();
                                                                                                   var bitmap:Bitmap = new Bitmap();
                       return points;
                                                                                                   bitmap = raster(strToSample, FONTSIZE);
       }//end class
                                                                                                   coords = sample(bitmap, numOfSamples, output-
                                                                            Width);
}//end pkg
// map dependant
var LONG INIT = -180;
                                                                                                   return coords;
var LAT INIT = 90;
var LONG FINAL = 180;
                                                                                           private function raster(strToRaster:String,
var LAT FINAL = -90;
var LONG DELTA = 360;
                                                                            sizeToRasterAt:uint):Bitmap
var LAT_DELTA = 180;
                                                                                                   //bitmap obj to draw text onto
// zoom/position dependant
                                                                                                   var bitmapData:BitmapData;
var mapX = 0;
var mapY = 0;
                                                                                                   // Create the text
var mapWidth = 800; //Stage.width;
                                                                                                   var txt:TextField = new TextField();
var mapHeight = mapWidth/2; //lock aspect
                                                                                                   txt.text = strToRaster;
                                                                                                   txt.autoSize = TextFieldAutoSize.LEFT;
        this._parent.map._x = mapX;
                                                                                                   var fmt:TextFormat = new TextFormat();
        this. parent.map. y = mapY;
                                                                                                   fmt.size = sizeToRasterAt;
        this. parent.map. width = mapWidth;
                                                                                                   fmt.font = "Arial Black";
        this._parent.map._height = mapHeight;
                                                                                                   fmt.color = 0x000000;
        this. parent.map.setMask(msk);
                                                                                                   txt.setTextFormat(fmt);
// plot dots when button is pressed
                                                                                                   // Make the BitmapData object, sized to accommo-
this._parent.generateButton.onRelease = function() {
                                                                            date the text. transp is false.
                                                                                                   bitmapData = new BitmapData(txt.textWidth, txt.
        //trace(rs.getLength());
                                                                            textHeight, false, Oxffffff);
        fields = rs.getColumnNames();
        for(var i=0; i<rs.getLength(); i++){</pre>
                                                                                                   // Draw the text into the BitmapData object
               //trace(rs.getItemAt(i)[fields[0]] + " , " +
                                                                                                   bitmapData.draw(txt);
rs.getItemAt(i)[fields[1]]);
               //imageId = rs.getItemAt(i)[fields[0]];
                                                                                                   // Associate the BitmapData object with a Bitmap
               imageLat = rs.getItemAt(i)[fields[0]];
                                                                            object
                imageLong = rs.getItemAt(i)[fields[1]];
                                                                                                   var bitmap:Bitmap = new Bitmap(bitmapData);
               //this._parent.traceTxt.text = Stage.width;
               //mapWidth = 600;
                                                                                                   return bitmap;
               //mapHeight = 298;
                                                                                           private function sample(bitmapToSample:Bitmap,
               //convert longitude & latitude to screen cordinates
               //for max precision latitude/longitude incr < lat/long
                                                                            numOfSamples:uint, outputWidth:uint):Array
delta/stage hight/width respectively
               //4 sigfigs ain't bad then for my purposes, 5 would be
                                                                                                   //matrix of 2d coordinate arrays to return
                                                                                                   var points:Array = new Array();
better though
               imageX = mapX + Math.round(((number(imageLong) + Math.
abs(LONG INIT)) * (mapWidth / LONG DELTA)));
                                                                                                   //bitmapData to sample
                                                                                                   var bitmapData:BitmapData = bitmapToSample.bit-
               imageY = mapY + Math.round((((number(imageLat) * -1) +
Math.abs(LAT INIT)) * (mapHeight / LAT DELTA)));
                                                                            mapData;
               //duplicate movieclip for each image location
```

//brute force determination number of black pix-

els					}
		<pre>var numOfBlkPixels:uint = 0; for (var i:uint=0; i<bitmapdata.width; (var="" for="" i++)="" j:uint="0;" j<bitmapdata.height;<="" pre="" {=""></bitmapdata.width;></pre>	}//end	}	
j++) {			}//end pkg		
getPixel(i,j))	;	//trace(bitmapData.	-	metaswa	rm.swarm.Swarm;
{		<pre>if(bitmapData.getPixel(i,j) == 0)</pre>	import	metaswa:	rm.swarm.SwarmState;
		<pre>numOfBlkPixels ++; }</pre>	public	class G	ridState implements SwarmState{
		}		private	e static const GRID:Number = 0;
		//calc sample step and scale factor to scale		private	e var swarm:Swarm;
output to desi	red siz	÷		public	function GridState(swarm:Swarm)
mOfSamples);		<pre>var sf:Number = outputWidth / bitmapData.width;</pre>			<pre>//trace("construct GridState"); this.swarm = swarm; init();</pre>
		//resample img every step pixels		}	
		<pre>var count:uint = 0; for (var m:Number=0; m<bitmapdata.width;< pre=""></bitmapdata.width;<></pre>		public	<pre>function init():void{ //genPattern();</pre>
m+=step){		for (var n:Number=0; n <bitmapdata.< td=""><td></td><td>} public</td><td><pre>function incrState():void{</pre></td></bitmapdata.<>		} public	<pre>function incrState():void{</pre>
height; n+=ste	p) {	<pre>if(bitmapData.getPixel(m,n) == 0)</pre>			<pre>//trace("ScatterState setPattern"); swarm.swarmState = swarm.scatter;</pre>
{		<pre>var nextPoint:Array =</pre>		}	<pre>swarm.setPattern();</pre>
[m*sf, n*sf];		//trace("next point: " +		public	<pre>function decrState():void{ //trace("ScatterState setPattern");</pre>
m*sf + ", " +	n*sf);	<pre>points.push(nextPoint);</pre>			<pre>swarm.swarmState = swarm.map; swarm.setPattern();</pre>
		count++		}	
		}			<pre>this private function setPattern():void{</pre>
		}			<pre>//trace("GridState setPattern"); /*var prev:String = new String();</pre>
		<pre>//check variables /*trace("numOfBlkPixels: " + numOfBlkPixels);</pre>			<pre>prev = "";*/ for(var i:uint=0; i<swarm.nodecnt; i++){<="" pre=""></swarm.nodecnt;></pre>
		<pre>trace("numOfSamples: " + numOfSamples); trace("step: " + step);</pre>	sortOrder);		<pre>swarm.nodes[i].setTarget(GRID, swarm.</pre>
		<pre>trace("sf: " + sf);</pre>	Bol coldel,		/*if(swarm.nodes[i].tag.t.text == prev)
		trace("count: " + count);*/	false;		swarm.nodes[i].tag.visible =
		<pre>//return coordinates of new blk pixels return points;</pre>			<pre>}else{ swarm.nodes[i].tag.visible =</pre>
	}		true;		}
}//end }//end pkg		ym gygymatatag (1	<pre>prev = swarm.nodes[i].tag.t.text;*/ }</pre>
import	metaswa	rm.swarmstates { arm.swarm.Swarm;		}	
import	metaswa	arm.swarm.SwarmState;	}//end }//end pkg	class	
public	class E	BrowningState implements SwarmState{	package metasw import		m.swarmstates { rm.swarm.Swarm;
	privat	e static const BROWNING:Number = 2;			rm.swarm.SwarmState;
	privat	e var swarm:Swarm;	public	class Lo	orenzeState implements SwarmState{
	public	<pre>function BrowningState(swarm:Swarm)</pre>		private	e static const LORENZE:int = 3;
	ι	<pre>//trace("construct GridState");</pre>		private	var swarm:Swarm;
		<pre>this.swarm = swarm; init();</pre>		public	function LorenzeState(swarm:Swarm)
	} public	<pre>function init():void{</pre>		{	<pre>//trace("construct GridState");</pre>
	}	<pre>//genPattern();</pre>			<pre>this.swarm = swarm; init();</pre>
	public	<pre>function incrState():void{ //trace("ScatterState setPattern"); swarm.swarmState = swarm.lorenze;</pre>		} public	<pre>function init():void{ //genPattern();</pre>
	}	<pre>swarm.setPattern();</pre>		} public	<pre>function incrState():void{</pre>
	public	<pre>function decrState():void{ //trace("ScatterState setPattern");</pre>			<pre>//trace("ScatterState setPattern"); swarm.swarmState = swarm.type;</pre>
		<pre>swarm.swarmState = swarm.scatter; swarm.setPattern();</pre>		}	swarm.setPattern();
	}	, ,		public	function decrState():void{
		<pre>this private function setPattern():void{</pre>			<pre>//trace("ScatterState setPattern"); swarm.swarmState = swarm.browning;</pre>
		<pre>//trace("GridState setPattern"); for(var i:uint=0; i<swarm.nodecnt; i++)="" pre="" {<=""></swarm.nodecnt;></pre>		}	<pre>swarm.setPattern();</pre>
swarm.sortOrde	r);	<pre>swarm.nodes[i].setTarget(BROWNING,</pre>			<pre>this private function setPattern():void{</pre>

```
//trace("GridState setPattern");
                                                                            }//end pkg
                      for(var i:uint=0; i<swarm.nodeCnt; i++){</pre>
                                                                            package metaswarm.swarm.swarmstates {
                              swarm.nodes[i].setTarget(LORENZE, swarm.
                                                                                    import metaswarm.swarm.Swarm;
sortOrder);
                                                                                    import metaswarm.swarm.SwarmState;
                                                                                   public class TypeState implements SwarmState{
       }//end class
                                                                                           private static const TYPE:Number = 4;
}//end pkg
package metaswarm.swarm.swarmstates
                                                                                           private var swarm:Swarm;
        import metaswarm.swarm.Swarm;
        import metaswarm.swarm.SwarmState;
                                                                                           function TypeState(swarm:Swarm)
       public class MapState implements SwarmState{
                                                                                                   //trace("construct GridState");
                                                                                                  this.swarm = swarm;
              private static const MAP:Number = 5;
                                                                                                  init();
              private var _swarm:Swarm;
                                                                                           public function init():void{
                                                                                                  //genPattern();
               public function MapState(swarm:Swarm)
                                                                                           public function incrState():void{
                      //trace("construct ScatterState");
                                                                                                  //trace("ScatterState setPattern");
                                                                                                  swarm.swarmState = swarm.map;
                      this. swarm = swarm;
                      init();
                                                                                                  swarm.setPattern();
               public function init():void{
                                                                                           public function decrState():void{
                      //do nothing
                                                                                                  //trace("ScatterState setPattern");
                                                                                                   swarm.swarmState = swarm.lorenze;
               public function incrState():void{
                                                                                                  swarm.setPattern();
                      //trace("ScatterState setPattern");
                       _swarm.swarmState = _swarm.grid;
                                                                                           //make this private
                       _swarm.setPattern();
                                                                                           public function setPattern():void{
                                                                                                   //trace("GridState setPattern");
              public function decrState():void{
                                                                                                  for(var i:uint=0; i<swarm.nodeCnt; i++) {</pre>
                      //trace("ScatterState setPattern");
                                                                                                         swarm.nodes[i].setTarget(TYPE, swarm.
                       _swarm.swarmState = _swarm.type;
                                                                            sortOrder);
                       swarm.setPattern();
              public function setPattern():void{
                      //trace("ScatterState setPattern");
                                                                                   }//end class
                                                                            }//end pkg
                      for(var i:uint=0; i< swarm.nodeCnt; i++) {</pre>
                              _swarm.nodes[i].setTarget(MAP, _swarm.
                                                                            package metaswarm.ui
sortOrder);
                                                                                import flash.display.*;
                                                                                    import flash.events.*;
                                                                                    import flash.ui.Keyboard;
       }//end class
                                                                                    import flash.net.*;
}//end pkg
                                                                                    import flash.utils.*;
package metaswarm.swarm.swarmstates
                                                                                    import flash.text.*;
        import metaswarm.swarm.Swarm;
                                                                                    import metaswarm.ui.hoverlabel.HoverLabel;
        import metaswarm.swarm.SwarmState;
                                                                                    import metaswarm.ui.log.Log;
                                                                                    import metaswarm.ui.sidebar.Sidebar;
       public class ScatterState implements SwarmState{
                                                                                    import metaswarm.swarm.Swarm;
                                                                                    import metaswarm.ui.uistates.AboutState;
               private static const SCATTER:Number = 1;
                                                                                    import metaswarm.ui.uistates.ContactState;
                                                                                    import metaswarm.ui.uistates.ExploreState;
               private var swarm:Swarm;
                                                                                    import metaswarm.ui.uistates.PreviewState;
               public function ScatterState(swarm:Swarm)
                                                                                   //import fl.controls.Label;
                      //trace("construct ScatterState");
                                                                                   //UI
                      this.swarm = swarm;
                                                                                public class Ui extends Sprite
                      init();
                                                                                           public var w:int;
              public function init():void{
                                                                                           public var h:int;
                      //do nothing
                                                                                           public var swarm: Swarm;
               public function incrState():void{
                                                                                           public var sidebar:Sidebar;
                      //trace("ScatterState setPattern");
                                                                                           public var hoverLabel:HoverLabel;
                      swarm.swarmState = swarm.browning;
                                                                                           public var cmdLog:Log;
                      swarm.setPattern();
                                                                                           //public var lastCmd:Label;
               public function decrState():void{
                                                                                           //internal states
                      //trace("ScatterState setPattern");
                                                                                           public var explore:UiState;
                      swarm.swarmState = swarm.grid;
                                                                                           public var preview:UiState;
                      swarm.setPattern();
                                                                                           //public var about:UiState;
                                                                                           //public var contact:UiState;
              public function setPattern():void{
                      //trace("ScatterState setPattern");
                                                                                           //internal state holder
                      for(var i:uint=0; i<swarm.nodeCnt; i++) {</pre>
                                                                                           public var uiState:UiState;
                              swarm.nodes[i].setTarget(SCATTER, swarm.
                                                                                    public function Ui():void
sortOrder);
                                                                                                  swarm = new Swarm();
                                                                                                  sidebar = new Sidebar();
       }//end class
                                                                                                  hoverLabel = new HoverLabel();
```

```
//lastCmd = new Label();
                                                                                                  uiState.onShftClick(node, button);
                      cmdLog = new Log();
                                                                                          public function onAltClick(node:Node,
                                                                            button:Object):void{
                      explore = new ExploreState(this);
                      preview = new PreviewState(this);
                                                                                                  uiState.onAltClick(node, button);
                      //about = new AboutState(this);
                      //contact = new ContactState(this);
                                                                                          public function onCtrlShftClick(node:Node,
                      uiState = explore;
                                                                            button:Object):void{
                                                                                                  uiState.onCtrlShftClick(node, button);
                      init();
                                                                                          /*function onDataBoxClick(node:Node,
               public function init():void{
                      //can't access this.stage until main is added to
                                                                           button:Object):void{
                                                                                                  uiState.onDataBoxClick(node, button);
                      //so listen for event then use stage
                      addEventListener(Event.ADDED TO STAGE, onAdded-
                                                                                          public function onMouseDown(node:Node,
                                                                           button:Object):void{
                                                                                                  uiState.onMouseDown(node, button);
               public function onAddedToStage(event:Event):void
                                                                                          public function onMouseUp(target:Node):void{
                      swarm.x = stage.stageWidth/6;
                      swarm.y = stage.stageHeight/6;
                                                                                                  uiState.onMouseUp(target);
                      swarm.w = stage.stageWidth*(2/3);
                      swarm.h = stage.stageHeight*(2/3);
                                                                                          public function onMouseOver(target:Node,
                                                                           button:Object):void{
               //delegate
                                                                                                  uiState.onMouseOver(target, button);
               public function onOpenState():void{
                                                                                          public function onCtrlOver(target:Node,
                      uiState.onOpenState();
                                                                            button:Object):void{
              public function onEnterFrame():void{
                                                                                                  uiState.onCtrlOver(target, button);
                      uiState.onEnterFrame();
                                                                                          public function onShftOver(target:Node,
                                                                           button:Object):void{
              public function onRightKey():void{
                                                                                                  uiState.onShftOver(target, button);
                      uiState.onRightKey();
                                                                                          public function onCtrlShftOver(node:Node,
              public function onLeftKey():void{
                                                                           button:Object):void{
                      uiState.onLeftKey();
                                                                                                  uiState.onCtrlShftOver(node, button);
              public function onUpKey():void{
                                                                                          public function onCtrlOut():void{
                                                                                                  uiState.onCtrlOut();
                      uiState.onUpKey();
              public function onDownKey():void{
                                                                                          public function onShftOut():void{
                      uiState.onDownKey();
                                                                                                  uiState.onShftOut();
              public function onSpaceKey():void{
                                                                                          public function onCtrlShftOut():void{
                      uiState.onSpaceKey();
                                                                                                  uiState.onCtrlShftOut();
              public function onKey1():void{
                                                                                          public function onMouseOut(target:Node):void{
                      uiState.onKey1();
                                                                                                  uiState.onMouseOut(target);
               public function onKey2():void{
                                                                               }//end class
                                                                            }//end pkg
                      uiState.onKev2();
                                                                            package metaswarm.ui
               public function onKey3():void{
                                                                                    import metaswarm.*;
                      uiState.onKey3();
                                                                                   import metaswarm.swarm.SwarmLoader;
                                                                                   import metaswarm.swarm.patternmaker.PatternMaker;
               public function onKey4():void{
                                                                                   import metaswarm.ui.hoverlabel.LabelMaker;
                                                                                   import metaswarm.ui.log.LogMaker;
                      uiState.onKey4();
                                                                                   import metaswarm.ui.sidebar.SidebarMaker;
                                                                                   public class UiLoader
              public function onMouseClick(node:Node,
button:Object):void{
                                                                                          private var ui:Ui;
                      uiState.onMouseClick(node, button);
                                                                                          private var labelMaker:LabelMaker;
                                                                                          private var sidebarMaker:SidebarMaker;
              public function onCtrlClick(node:Node,
                                                                                          private var patternMaker:PatternMaker;
                                                                                          private var _logMaker:LogMaker;
button:Object):void{
                                                                                          public function UiLoader(main:Ui) {
                      uiState.onCtrlClick(node, button);
                                                                                                  this. ui = main;
                                                                                                  this._patternMaker = new PatternMaker(
              public function onShftClick(node:Node,
button:Object):void{
                                                                           ui.swarm);
                                                                                                  this._labelMaker = new LabelMaker(_
```

stage

77

ToStage);

```
ui.hoverLabel);
                                                                                  import flash.display.*;
                      this. sidebarMaker = new SidebarMaker(
                                                                                  import flash.text.*;
ui.sidebar);
                      this. logMaker = new LogMaker( ui.cmdLog);
                                                                                  public class HoverLabel extends Sprite{
                      init():
                                                                                         //add formatting constants here
                                                                                         public var labelMaker:LabelMaker;
               private function init():void {
                                                                                         public var _output:TextField;
                                                                                         public var _text:String;
                                                                                         public function HoverLabel():void
               public function load(dat:Array):void{
                      //trace("main.load()");
                                                                                                 output = new TextField();
                      var swarmLoader:SwarmLoader = new SwarmLoader(_
                                                                                                 _text = new String();
                                                                                                 labelMaker = new LabelMaker(this);
ui.swarm); //<----
                      -----not sure about this
                      swarmLoader.load(dat);
                                                                                                 init();
              public function make():void{
                                                                                         public function init():void {
                      //trace("main.make()");
                                                                                                 visible = true;
                      var swarmLoader:SwarmLoader = new SwarmLoader(
                                                                                                 focusRect = false;
ui.swarm); //<----not sure about this
                                                                                                 tabEnabled = false:
                      swarmLoader.make();
                                                                                                 mouseEnabled = false;
                       labelMaker.make();
                      patternMaker.make();
                       sidebarMaker.make();
                       _logMaker.make();
                                                                                         /*public function load():void{
               public function go():void{
                      //trace("main.go()");
                                                                                         public function make():void{
                      var swarmLoader:SwarmLoader = new SwarmLoader(_
                                                                                                 labelMaker.make();
                               ----not sure about this
                      swarmLoader.go();
                                                                                         public function go():void{
                      labelMaker.go();
                                                                                                 labelMaker.go();
                       sidebarMaker.go();
                      logMaker.go();
                                                                                         public function get text():String {
                                                                                      return _text;
                      ui.addChild( ui.swarm);
                      ui.addChild(ui.hoverLabel);
                                                                                  public function set text(newText:String):void {
                       ui.addChild(_ui.sidebar);
                                                                                          text = newText;
                      __ui.addChild(_ui.cmdLog);
                      //set initial pattern
                                                                                         public function update():void{
                                                                                                 _output.x = mouseX + 15;
                      ui.swarm.setPattern();
                                                                                                 _output.y = mouseY + 5;
                                                                                                 _output.text = _text;
package metaswarm.ui {
                                                                                         public function show():void{
       //fsm interface
                                                                                                 _output.visible = true;
       public interface UiState{
               //internal interface methods
                                                                                         public function hide():void{
               function onOpenState():void;
                                                                                                 output.visible = false;
               function onEnterFrame():void;
               function onRightKey():void;
               function onLeftKey():void;
               function onUpKey():void;
                                                                                  }//end class
               function onDownKey():void;
                                                                          }//end pkg
               function onSpaceKey():void;
                                                                          package metaswarm.ui.hoverlabel {
                                                                                  import flash.display.*;
               function onKev1():void;
               function onKey2():void;
                                                                                  import flash.text.*;
               function onKey3():void;
                                                                                  public class LabelMaker extends Sprite{
               function onKey4():void;
               function onMouseClick(node:Node, button:Object):void;
               function onCtrlClick(node:Node, button:Object):void;
                                                                                         private var hl:HoverLabel;
               function onShftClick(node:Node, button:Object):void;
               function onAltClick(node:Node, button:Object):void;
                                                                                         public function LabelMaker(hoverLabel:HoverLabel):void{
               function onCtrlShftClick(node:Node, button:Object):void;
                                                                                                 this.hl = hoverLabel;
               //function onDataBoxClick(node:Node,
                                                                                                 init();
button:Object):void;
               function onMouseDown(node:Node, button:Object):void;
                                                                                         public function init():void {
               function onMouseUp(target:Node):void;
               function onMouseOver(target:Node, button:Object):void;
               function onCtrlOver(target:Node, button:Object):void;
                                                                                         public function make():void {
               function onShftOver(target:Node, button:Object):void;
                                                                                                 //var fmt:TextFormat = new TextFormat();
               function onCtrlShftOver(node:Node, button:Object):void;
                                                                                                 var fmt:TextFormat = new TextFormat("Technic",
               function onCtrlOut():void;
                                                                          10, 0xfff);
               function onShftOut():void;
               function onCtrlShftOut():void;
                                                                                                 //tag.t.embedFonts = true;
                                                                                                 //tag.t.antiAliasType = AntiAliasType.ADVANCED;
               function onMouseOut(target:Node):void;
                                                                                                 hl. output.text = "";
       }//end interface
                                                                                                 hl. output.setTextFormat(fmt);
                                                                                                hl._output.selectable = false;
}//end pkg
                                                                                                hl._output.mouseEnabled = false;
package metaswarm.ui.hoverlabel {
                                                                                                 //log.id = 0;
```

```
//t.width =
                                                                                   public class LogMaker extends Sprite{
                      //t.height =
                      hl. output.background = true
                                                                                           private var _log:Log;
                      hl._output.backgroundColor = 0xf0ff64; //black
                      //t.border = true;
                                                                                           public function LogMaker(log:Log):void{
                      //t.borderColor = 0x333333; //dark gray
                                                                                                  this._log = log;
                      hl._output.autoSize = TextFieldAutoSize.LEFT;
                                                                                                  init();
                      //addChild(this);
                                                                                           public function init():void {
                      //tag.t.text = tag.dat[tag.id];
                                                                                           public function make():void {
                                                                                                  //var fmt:TextFormat = new TextFormat();
              public function go():void {
                                                                                                  var fmt:TextFormat = new TextFormat("Technic",
                      hl.addChild(hl. output);
                                                                            10, 0xfff);
                                                                                                  //fmt.color = 0xfff;
                                                                                                  //tag.t.embedFonts = true;
                                                                                                  //tag.t.antiAliasType = AntiAliasType.ADVANCED;
                                                                                                   log.output.text = "";
package metaswarm.ui.log {
                                                                                                   _log.output.setTextFormat(fmt);
        import flash.display.*;
                                                                                                   log.output.selectable = false;
       import flash.text.*;
                                                                                                   log.output.mouseEnabled = false;
       public class Log extends Sprite{
                                                                                                  //log.id = 0;
               //add formatting constants here
                                                                                                  //t.width =
               //public var logMaker:LogMaker;
                                                                                                  //t.height =
               public var output:TextField;
                                                                                                   //_log.output.background = true
               public var entries:Array;
                                                                                                  //_log.output.backgroundColor = 0x999999; //
                                                                            light gray
              public function Log():void
                                                                                                   //t.border = true;
                                                                                                  //t.borderColor = 0x333333; //dark gray
                                                                                                   log.output.autoSize = TextFieldAutoSize.LEFT;
                      output = new TextField();
                      //logMaker = new LogMaker(this);
                                                                                                   //addChild(this);
                      entries = new Array();
                                                                                                  //tag.t.text = tag.dat[tag.id];
                      init();
                                                                                                   log.output.x=1100;
              public function init():void {
                                                                                                   log.output.y=0;
                      visible = true;
                      focusRect = false;
                                                                                           public function go():void {
                      tabEnabled = false:
                                                                                                   log.addChild( log.output);
                      //tag.mouseEnabled = false;
               public function load():void{
                                                                            package metaswarm.ui.sidebar {
                                                                                   import flash.display.*;
                                                                                   import flash.text.*;
               public function make():void{
                      //logMaker.make();
                                                                                    public class Sidebar extends Sprite{
                                                                                           public var sidebarMaker:SidebarMaker;
               public function append(txt:String):void{
                      trace("txt: " + txt);
                                                                                           public var header:TextField;
                      trace("entries[" + entries.length + "]: " +
                                                                                           public var shortcuts:TextField;
                                                                                           public var states:TextField;
entries[entries.length-1]);
                                                                                           public var nodeList:TextField;
                                                                                           public var footer:TextField;
                      trace("_");
                                                                                           public function Sidebar():void
                      entries.push(txt);
                      //output.unshift(txt + '\n');
                      output.replaceText(0, 0, txt + \n');
                                                                                                  header = new TextField();
                                                                                                  shortcuts = new TextField();
                                                                                                  states = new TextField();
                      if(txt != entries[entries.length-1]){
                                                                                                  nodeList = new TextField();
                             trace("log it");
                                                                                                  footer = new TextField();
                              entries.push(txt);
                              //output.appendText(txt + \n');
                                                                                                  sidebarMaker = new SidebarMaker(this);
                              //output.unshift(txt + '\n');
                              output.replaceText(0, 0, txt + \n');
                                                                                                  init();
                      }else{
                                                                                           public function init():void {
                              //entries.push(txt);
                              //output.appendText(entries.pop() +
'\n');
                              //output.text = entries.pop();
                                                                                           public function load():void{
                                                                                           public function make():void{
                                                                                                  sidebarMaker.make();
       }//end class
}//end pkg
package metaswarm.ui.log {
                                                                                   }//end class
                                                                            }//end pkg
        import flash.display.*;
                                                                            package metaswarm.ui.sidebar {
        import flash.text.*;
```

```
"ctrl + r - add /
       import flash.display.*;
                                                                        remove" + '\n' +
       import flash.geom.ColorTransform;
       import flash.text.*;
                                                                                                                           shft + r - se
                                                                        lect by" + '\n' +
       public class SidebarMaker extends Sprite{
                                                                                                                           "ctrl&shft + r -
                                                                        filter by" + \n';
              public var sb:Sidebar;
                                                                                              sb.states.setTextFormat(fmt);
                                                                                              sb.states.selectable = false;
              public function SidebarMaker(sidebar:Sidebar):void{
                                                                                              sb.states.mouseEnabled = false;
                     this.sb = sidebar;
                                                                                              sb.states.x = xOffset;
                                                                                              sb.states.width = sbWidth;
                     init();
                                                                                              sb.states.height = 130;
                                                                                              //sb.states.autoSize = TextFieldAutoSize.LEFT;
              public function init():void {
                                                                                              sb.states.background = true;
                                                                                              sb.states.backgroundColor = 0xffffff;
              public function make():void {
                                                                                              sb.states.border = false;
                                                                                              sb.states.borderColor = 0xE6E6E6;
                     var h1fmt:TextFormat = new TextFormat();
                                                                                              // **********
                     h1fmt.font = "technic";
                     h1fmt.size = 16;
                                                                                             sb.nodeList.text = '\n' + "nodes" + '\n\n';
                     h1fmt.leftMargin = 10;
                                                                                              sb.nodeList.setTextFormat(fmt);
                                                                                              sb.nodeList.selectable = false:
                     var fmt:TextFormat = new TextFormat();
                                                                                              sb.nodeList.mouseEnabled = false;
                     fmt.font = "technic";
                                                                                             //sb.nodeList.scrollV = 1;
                     fmt.size = 16;
                                                                                              sb.nodeList.x = xOffset;
                     fmt.leftMargin = 10;
                                                                                             sb.nodeList.width = sbWidth;
                                                                                              sb.nodeList.height = 900 - (sb.header.textHeight
                     // ******* shared
properties
                                                                        sb.shortcuts.textHeight +
                     var bg:ColorTransform;
                     bg.color = 0xFAFAFA;
                                                                        sb.states.textHeight +
                     var border:ColorTransform;
                     border.color = 0xE6E6E6;
                                                                        sb.footer.textHeight);
                                                                                             //sb.nodeList.autoSize = TextFieldAutoSize.LEFT;
                     var sbWidth:int = 180;
                                                                                             sb.nodeList.background = true;
                     var xOffset:int = 20;
                                                                                             sb.nodeList.backgroundColor = 0xffffff;
                     var yOffset:int = 50;
                                                                                              sb.nodeList.border = false;
                                                                                              sb.nodeList.borderColor = 0xE6E6E6;
                     // **********
                                                                                              // **********
                     sb.header.text = '\n' + "meta.swarm" + '\n\n';
                                                                                              sb.footer.text = '\n' + "by boot" + '\n';
                     sb.header.setTextFormat(h1fmt);
                     sb.header.selectable = false;
                                                                                             sb.footer.setTextFormat(fmt);
                     sh.header.mouseEnabled = false:
                                                                                              sb.footer.selectable = false;
                                                                                              sb.footer.mouseEnabled = false;
                     sb.header.x=xOffset;
                     //sb.header.y=0;
                                                                                              sb.footer.x = xOffset;
                     sb.header.width = sbWidth;
                                                                                              sb.footer.width = sbWidth;
                     sb.header.height = 40;
                                                                                              sb.footer.height = 40;
                     //sb.header.autoSize = TextFieldAutoSize.CENTER;
                                                                                             //sb.footer.autoSize = TextFieldAutoSize.LEFT;
                     sb.header.background = true;
                                                                                              sb.footer.background = true;
                     sb.header.backgroundColor = 0xffffff;
                                                                                              sb.footer.backgroundColor = 0xffffff;
                     sb.header.border = false;
                                                                                              sb.footer.border = false;
                     sb.header.borderColor = 0xE6E6E6;
                                                                                              sb.footer.borderColor = 0xE6E6E6;
                                                                                              // *********
                     // ***********
                     sb.shortcuts.text = " " + \n' +
                                                                                              sb.header.y = yOffset;
                                                                                              sb.shortcuts.y = sb.header.height;
                                                         "naviga-
t.ion'' + '\n\n' +
                                                                                              sb.states.y = sb.shortcuts.y + sb.shortcuts.
                                                         …^ - in-
                                                                        height;
crement sort" + \n' +
                                                                                              sb.nodeList.y = sb.states.y + sb.states.height;
                                                                                              sb.footer.y = 900 - sb.footer.height;
                                                         "v - dec-
rement sort" + '\n' +
                                                         "> - in-
crement pattern" + '\n' +
                                                                                      public function go():void {
                                                         "< - dec-
rement pattern" + \n^{\prime};
                                                                                              //sb.addChild(sb.header);
                     sb.shortcuts.setTextFormat(fmt);
                                                                                              sb.addChild(sb.shortcuts);
                     sb.shortcuts.selectable = false;
                                                                                             sb.addChild(sb.states);
                     sb.shortcuts.mouseEnabled = false;
                                                                                              //sb.addChild(sb.nodeList);
                     sb.shortcuts.x = xOffset;
                                                                                              //sb.addChild(sb.footer);
                     //sb.shortcuts.y = 0;
                     sb.shortcuts.width = sbWidth;
                     sb.shortcuts.height = 140;
                     //sb.shortcuts.autoSize = TextFieldAutoSize.CEN-
                                                                        package metaswarm.ui.uistates {
                                                                                import metaswarm.ui.Ui;
                     sb.shortcuts.background = true;
                                                                                import metaswarm.ui.UiState;
                     sb.shortcuts.backgroundColor = 0xffffff;
                     sb.shortcuts.border = false;
                                                                               public class AboutState implements UiState{
                     sb.shortcuts.borderColor = 0xE6E6E6;
                                                                                      private var main:Ui;
                     // ***********
                     sb.states.text = " " + '\n' +
                                                                                      public function AboutState(main:Ui)
                                                   "selection" +
'\n\n' +
                                                                                              //trace("construct ExploreState");
                                                   "r - select / de-
                                                                                             this.main = main;
select" + '\n' +
                                                                                             init();
```

	<pre>} private }</pre>	e function	n init():void{		package	import	metaswa	istates rm.ui.Ui; rm.ui.UiSt	{ cate;	
ods*******	//deleg	gated metl	h- *******	*		class C	ontactSt	tate imple	ements UiSta	te{
	public	function	onEnterFrame():v	oid{			var mai	n:Ui;		
	public	function	onRightKey():voi	d{			functio	n Contact	State(main:U	Ji)
	public	function	onLeftKey():void	{			t	//trace("this.main		xploreState");
	public	function	<pre>onUpKey():void{</pre>				}	init();		
	public	function	onDownKey():void	{			public	function	init():void{	
	public		onSpaceKey():voi				//deleg	ated meth	_	
	} public	function	onKey1():void{		ods****	*****			********** onEnterFrame	
	public	//main.p	picked.restore(); mainState = main.e	explore;			}		onRightKey()	
	} public	function	onKey2():void{				} public	function	onLeftKey():	void{
		_	<pre>picked.restore(); dropped.tile();</pre>				} public	function	onUpKey():vo	oid{
	}	//main.m	mainState = main.r	preview;			} public	function	onDownKey():	void{
	public		onKey3():void{ nainState = main.a	about;			} public	function	onSpaceKey()	:void{
	} public	function	onKey4():void{				}	//main.ma	inState = ma	ain.xxx;
	}	//main.m	mainState = main.c	contact;			public		onKey1():voi inState = ma	d{ ain.explore;
outton:Object):	_	function	onMouseClick(nod	e:Node,			<pre>} public</pre>		onKey2():voi inState = ma	d{ ain.preview;
outton:Object):		function	onCtrlClick(node	:Node,			} public		onKey3():voi inState = ma	
outton:Object):	_	function	onShftClick(node	:Node,			} public	function	onKey4():voi	
outton:Object):	-	function	<pre>onAltClick(node:</pre>	Node,	button:	Obiect):	-		onMouseClick	
outton:Object):		function	onCtrlShftClick(node:Node,		_	} public	function	onCtrlClick(node:Node,
outton:Object):	_	function	onCtrlShftOver(n	ode:Node,	button:	Object):	}	function	onShftClick(node:Node,
	<pre>public</pre>	function	<pre>onCtrlShftOut():</pre>	void{	button:	Object):	}	function	onAltClick(n	anda Mada
outton:Object):	_	function	onMouseDown (node	:Node,	button:	Object):	-	Tunccion	OHAT CCITCK (H	lode.Node,
	}	//target	.drag();		button:	Object):	-	function	onCtrlShftCl	ick(node:Node,
	<pre>public }</pre>		<pre>onMouseUp(targetdrop();</pre>	:Node):void{	button:	Obiect):	-	function	onCtrlShftOv	er(node:Node,
outton:Object):	_	function	onMouseOver(targ	et:Node,		,	}	function	onCtrlShftOu	ut():void{
	}	//target	.startPrev();				} public	function	onMouseDown(node:Node,
outton:Object):	_	function	onCtrlOver(targe	t:Node,	button:	Object):	:void{	//target.	drag();	
	}	_	.startPrev();				<pre>} public</pre>	function	onMouseUp(ta	rget:Node):void
outton:Object):			onShftOver(targe	t:Node,			}	//target.	_	
	}	_	.startPrev();		button:	Object):	-			target:Node,
	public		<pre>onCtrlOut():void .startPrev();</pre>	{			}	_	startPrev()	
	} public		<pre>onShftOut():void .startPrev();</pre>	{	button:	Object):	-		onCtrlOver(t startPrev()	-
	} public	_	onMouseOut(targe	t:Node):void{			} public		onShftOver(t	
	}		.stopPrev();		button:	Object):	_		startPrev()	-
}//end	class						} public	function	onCtrlOut(): startPrev()	void{
//end pkg							}			

```
public function onShftOut():void{
                                                                                                switch (button.type) {
                      //target.startPrev();
                                                                                                       case "dot":
              public function onMouseOut(target:Node):void{
                                                                                                               //dot clear and select
                                                                                                               //main.swarm.add(node);
                      //target.stopPrev();
                                                                                                               //main.cmdLog.append("add/remove
                                                                          to selection");
                                                                                                               //tag clear and selectby
       }//end class
                                                                                                               main.swarm.add(node);
}//end pkg
                                                                                                               //main.swarm.addBy(node.
package metaswarm.ui.uistates {
                                                                          tag.t.text);
       import metaswarm.*;
                                                                                                               main.cmdLog.append("add node: " +
       import metaswarm.ui.Ui;
                                                                          node.dat[0] + node.dat[1]);
       import metaswarm.swarm.
                                                                                                               break;
       import metaswarm.ui.UiState;
                                                                                                       case "icon":
       import com.greensock.TweenLite;
                                                                                                               //tag clear and selectby
                                                                                                               main.swarm.remove(node);
       public class ExploreState implements UiState{
                                                                                                               main.cmdLog.append("remove node:
                                                                          " + node.dat[0] + node.dat[1]);
              private var main:Ui;
                                                                                                               break;
                                                                                                       case "tag":
               public function ExploreState(main:Ui)
                                                                                                               //main.swarm.addBy(node.
                                                                          tag.t.text);
                      //trace("construct ExploreState");
                                                                                                               break;
                      this.main = main;
                                                                                                        default:
                      init();
                                                                                                               trace("oops, you ctrl clicked
                                                                          button of type: " + button.type);
              public function init():void{
                                                                                         public function onShftClick(node:Node,
              //delegated meth-
                                                                          button:Object):void
ods**************
              public function onOpenState():void{
                                                                                                switch (button.type) {
                      main.cmdLog.append("in explore state");
                                                                                                       case "dot":
                                                                                                               //main.swarm.filterBy(node.
              public function onEnterFrame():void{
                                                                          tag.t.text);
                     main.swarm.rePosition();
                                                                                                               //main.cmdLog.append("filter by: "
                      main.swarm.reSize();
                                                                          + node.tag.t.text);
                      main.hoverLabel.update();
                                                                                                               main.swarm.selectBy(node.
                                                                          tag.t.text);
              public function onRightKey():void{
                                                                                                               main.cmdLog.append("select by: "
                     main.swarm.incrState();
                                                                          + node.tag.t.text);
                      main.cmdLog.append("next pattern");
                                                                                                               break:
                      //main.lastCmd.text = "next pattern";
                                                                                                       case "icon":
              public function onLeftKey():void{
                                                                                                               //main.swarm.xxx(node);
                      main.swarm.decrState();
                                                                                                               main.swarm.selectBy(node.
                      main.cmdLog.append("previous pattern");
                                                                          tag.t.text);
                                                                                                               \verb|main.cmdLog.append("select by: "
              public function onUpKey():void{
                                                                          + node.tag.t.text);
                      main.swarm.incrSort();
                                                                                                               break;
                      main.cmdLog.append("increment sort");
                                                                                                        case "tag":
                                                                                                               /*main.swarm.filterBy(node.
              public function onDownKey():void{
                                                                          tag.t.text);
                      main.swarm.decrSort();
                                                                                                               main.cmdLog.append("filter by: " +
                      main.cmdLog.append("decrement sort");
                                                                          node.tag.t.text);*/
                                                                                                               break;
              public function onSpaceKey():void{
                                                                                                       default:
                      //main.dropped.tile(); //could move to preview
                                                                                                               trace("damn it, you shft clicked
state onEnterFrame
                                                                          button of type: " + button.type);
                      //main.mainState = main.preview;
                      main.swarm.dropSelected();
                      main.cmdLog.append("drop selected");
                                                                                        public function onAltClick(node:Node,
                                                                          button:Object):void
              public function onKey1():void{
                                                                                                //can't use alt b/c window uses it to focus
                                                                          toolbar
              public function onKey2():void{
                                                                                         public function onCtrlShftClick(node:Node,
                      trace("ExploreState->onKey2");
                                                                          button:Object):void
                      main.swarm.preview();
                      main.uiState = main.preview;
                                                                                                switch (button.type) {
                      //main.uiState.onOpenState();
                                                                                                       case "dot":
                      main.cmdLog.append("preview mode");
                                                                                                               break;
              public function onKey3():void{
                                                                                                        case "icon":
                      //main.mainState = main.about;
                                                                                                               main.swarm.filterBy(node.
                                                                          tag.t.text);
              public function onKey4():void{
                                                                                                               main.cmdLog.append("filter by: " +
                      //main.mainState = main.contact;
                                                                          node.tag.t.text);
                      TweenLite.to(main.swarm, .25, {scaleX:1, scal-
                                                                                                               break;
eY:1});
                                                                                                        case "tag":
              public function onCtrlClick(node:Node,
                                                                                                               break;
button:Object):void
                                                                                                        default:
```

				trace("shit, yo	u shft clicked			case	"dot":	
utton of type	: " + bu	tton.ty	pe);	orace (shrey jo	2 01110 01101100			0450	main.hoverLabel.text = "se	lect "
	,	}				+ node.dat[0]	+ node.c	dat[1];		
) nublic	functio	n onMo	useClick(node:Node	_ د			case	break; "icon":	
utton:Object)	-	Tuncero.	011110	abcorren (node: Node	,			case	main.hoverLabel.text = "dra	ag to
	{					pluck " + node	e.dat[0]	+ node.dat[1]		
		switch		on.type) { "dot":				2222	break;	
			case	//dot clear and	select.			Case	"tag": main.hoverLabel.text = "Cle	ear an
				main.swarm.sele	ct(node);	Select by: " -	+ node.ta	ag.t.text;		
				main.cmdLog.app	end("select node:				break;	
+ node.dat[0] + node	.dat[1]);	break;				case	"popout": //rename popoutData main.hoverLabel.text = "sho	
			case	"icon":		for " + node.o	dat[0] +	node.dat[1];		ow data
				//tag clear and	selectby				break;	
				//main.swarm.se	lectBy(node.			case	"popin":	
ag.t.text);				//main cmdI.oc a	ppend("select by:	for " + node.o	+ [0] +sh	node dat[1].	<pre>main.hoverLabel.text = "hic</pre>	de dat
+ node.tag.t	.text);			// main.cmalog.a	ppena (befeet by.	101 1 11040.1	aac[o] ·	node.dat[1],	break;	
-				//DO NOTHING B/	C drag will handle			case	"closeThumb":	
his case! <									main.hoverLabel.text = "clo	
			Case	break; "tag":		thumbnail and	return `	` + node.dat[<pre>0] + node.dat[1] + " to the sw break;</pre>	/arm";
			case	//				case	"bmp":	
				//main.swarm.xx	x(node);				main.hoverLabel.text = "dra	ag " +
				main.swarm.sele	ctBy(node.	<pre>node.dat[0] +</pre>	node.dat	[1];		
ag.t.text);				//main amait as a				4.5.	break;	
node.tag.t.te	ext):			//main.cmaLog.a	ppend("filter by: "			defa	uit: //trace("Not a valid type (of but
nous roug rore	0110//			break;		ton");			,, class (nee a valla e,pe .	01 240
			case	"popout": //rename	= =			}		
				//trace("popout			}			
				<pre>main.swarm.popor break;</pre>	ut (node);	button:Object)		function onC	trlOver(node:Node,	
			case	"popin":		buccon.object,).VOIQ(main.swarm.s	tartHover(node);	
				//trace("popin");				og.append("start hover");	
				main.swarm.popi	n(node);				.startFocusBy(node.tag.t.text)	
				<pre>break; "closeThumb":</pre>		ملمم ا 101 عملم	Jat [1].	main.hoverLa	bel.text = "Add/remove " + noo	de.
			Case	//		dat[0] + node	.uat[1];			
				main.swarm.clos	e(node);		public	function onSl	hftOver(node:Node,	
				break;		button:Object)):void{			
			defau		valid type of but-				tartHover(node);	
on");				//trace(Not a	valid type of but-	//should be fo	ocusDotsF		tartSelectByFocus(node.tag.t.	text);
- ,,		}				,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		-	bel.text = "Select by: " + no	de.
	}					tag.t.text;				
	//	a funat	ion on	DataBoxClick(node:	Nodo		}	function once	trlShftOver(node:Node,	
utton:Object)	_	.C Tunct.	1011 011	DataboxCIICK (Hode.	Node,	button:Object)		Tunction one	crishicover (hode. Node,	
, , , , , , , , , , , , , , , , , , ,								main.swarm.s	tartHover(node);	
		function	n onMo	useDown(node:Node,					tartFilterByFocus(node.tag.t.	text);
utton:Object)	:void					//should be fo	ocusIcons	-	hal taut - WHilton how William	ما لم -
	1	switch	(butt	on.type) {		tag.t.text;		main.noverta	bel.text = "Filter by: " + no	ae.
				"icon":		,,	}			
				//						
o dw==∓ !!				main.swarm.drag	(node); //should		//out	function :	our court (node Node)	
e dragIcon!!				main.cmdLog.app	end("drag " +		public	main.hoverLa	ouseOut(node:Node):void{	
ode.dat[0] + :	node.dat	[1]);			- ,			main.swarm.s	topHover(node);	
				break;					g.append("stop hover");	
			case	"bmp":	(1	//main.swarm	.stopFocus(target.tag.t.text)	;
e dragThumb!!				main.swarm.drag	(node); //should		public	function onCi	trlOut():void{	
o araginamo				main.cmdLog.app	end("drag " +		Public		.startHover(node);	
ode.dat[0] + 1	node.dat	[1]);							topFocusBy();	
			1 - C -	break;			1	//main.cmdLo	g.append("stopFocusBy");	
			defau	llt: //break) public	function one	hftOut():void{	
		}		, , 520an					n.startHover(node);	
	}							main.swarm.s	topFocusBy();	
	public			useUp(node:Node):v	oid{		1	//main.cmdLo	g.append("stopFocusBy");	
	}	maln.sw	varm.di	rop(node);) public	function once	trlShftOut():void{	
	ı						Puntic	//		
	//over						}			
	-	function	n onMo	useOver(node:Node,						
utton:Object)	:void{	main 1-	NTO 27 -1	oel.show();		1 / /	l class			
				oe1.snow(); tartHover(node);		}//end pkg	CIASS			
				g.append("start ho	ver");	package metas	warm.ui.u	istates {		
		//main.	swarm	.startFocus(target	.tag.t.text);		metaswa			
		ouri + ah	(bu++	on.type) {		-		rm.ui.Ui; rm.ui.UiState		
		OWILCII	(NULL)	orr.chbc) [TIIIDOLL	. mcLaSWd	±m.u±.∪±3lale	. 1	

```
import metaswarm.swarm.Swarm;
       import com.greensock.TweenLite;
       public class PreviewState implements UiState{
              private var main:Ui;
              public function PreviewState(main:Ui)
                      //trace("construct ExploreState");
                      this.main = main;
                     init();
              public function init():void{
              //delegated meth-
ods**************
              public function onOpenState():void{
                     main.cmdLog.append("in preview state");
              public function onEnterFrame():void{
                     main.swarm.rePosition();
                     main.swarm.reSize();
                     main.hoverLabel.update();
              public function onRightKey():void{
               public function onLeftKey():void{
              public function onUpKey():void{
              public function onDownKey():void{
              public function onSpaceKey():void{
                     //main.mainState = main.xxx;
                      TweenLite.to(main.swarm, 1, {scaleX:3, scal-
eY:3});
              public function onKey1():void{
                     trace("previewState->onKey1");
                     main.swarm.explore();
                     main.uiState = main.explore;
                     main.cmdLog.append("explore mode");
              public function onKey2():void{
                      //main.mainState = main.preview;
              public function onKey3():void{
                      //main.mainState = main.about;
                      TweenLite.to(main.swarm, .25, {scaleX:3, scal-
eY:3});
              public function onKey4():void{
                     //main.mainState = main.contact;
              public function onMouseClick(node:Node,
button:Object):void{
              public function onCtrlClick(node:Node,
button:Object):void{
              public function onShftClick(node:Node,
button:Object):void{
              public function onAltClick(node:Node,
button:Object):void{
               public function onCtrlShftClick(node:Node,
button:Object):void{
              public function onCtrlShftOver(node:Node,
button:Object):void{
              public function onCtrlShftOut():void{
              public function onMouseDown(node:Node,
button:Object):void{
                     //target.drag();
              public function onMouseUp(target:Node):void{
                     //target.drop();
              public function onMouseOver(target:Node,
button:Object):void{
                     //target.startPrev();
```

```
public function onCtrlOver(target:Node,
button:Object):void{
                      //target.startPrev();
              public function onShftOver(target:Node,
button:Object):void{
                      //target.startPrev();
              public function onCtrlOut():void{
                      //target.startPrev();
              public function onShftOut():void{
                      //target.startPrev();
              public function onMouseOut(target:Node):void{
                      //target.stopPrev();
       }//end class
}//end pkg
```