

파라메트릭 디자인 XX

Parametric Design XX

글. 성우제 Sung, Woojae

세종대학교 건축학과 조교수

www.wojsung.com, www.selective-amplification.net

지난 회에서는 종이접기 시뮬레이션의 전체적인 Grasshopper definition의 전체적인 분석과 이중 가장 중요한 부분을 담당하고 있는 Visual Basic의 내용(아래 그림의 초록색 부분)을 살펴 보았습니다(fig.01). 그 연장선상에서 이번호에는 지난회에서 마치지 못했던 Visual Basic scripting part의 나머지 부분을 이야기 하겠습니다.

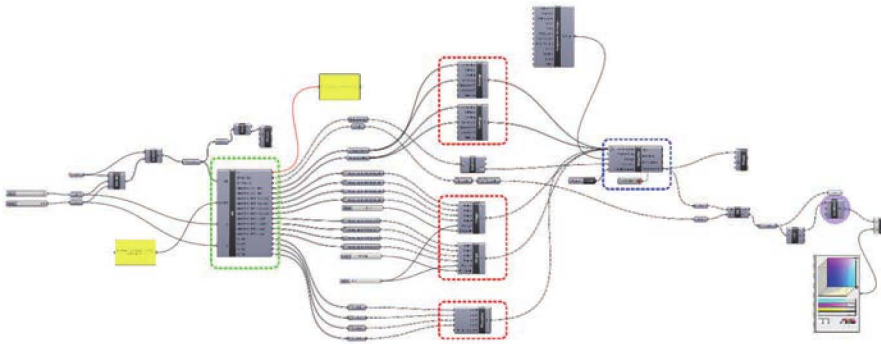


Figure 1

지난회에서 다루었던 Visual Basic의 내용은 접혀서는 안되는 면 즉, 아래 그림(fig.02)의 패턴 A에 해당하는 면들을 주어진 패턴문자를 통해 인식하고 Kangaroo가 이들을 강제적으로 평평하게 유지하기 위해 해당 패널들의 대각선과 네 꼭지점을 추출하는 것입니다. 이는 추후 Kangaroo의 컴포넌트에 입력되어야 합니다. 부연하자면 Visual Basic을 통해 이뤄지는 작업은 실제적으로 시뮬레이션하는 것이 아닌 시뮬레이션을 위해 주어진 지오메트리를 sorting 하는 작업이라고 생각하시면 됩니다.

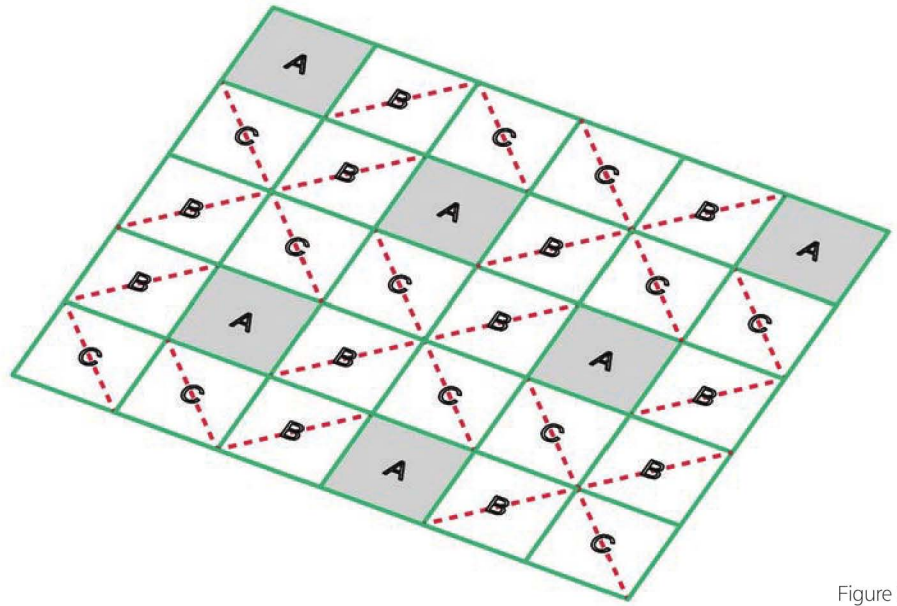


Figure 2

이전 회에서 설명했듯이 면이 접히는 것은 Kangaroo의 hinge 컴포넌트로 시뮬레이션이 가능합니다. 이를 위해서는 접히는 선을 정의하는 두개의 점과 이에 접하는 두개의 접히는 면위에 놓이는 또 다른 두개의 점을 정의해야 합니다(fig. 03/fig. 04).

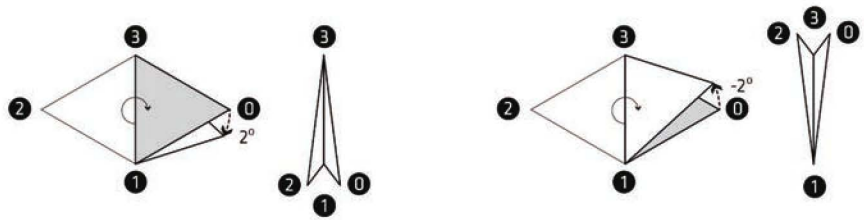


Figure 3

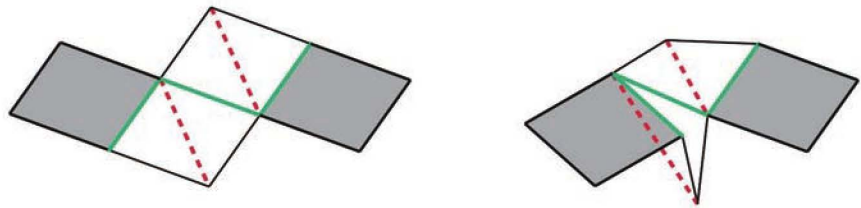


Figure 4

그림 2, 3, 4에서 유추할 수 있듯이 패널의 내부에서는 항상 면이 위로 접히고(접힌 선분이 아래로 향하고 - 붉은 점선), 패널과 패널사이에는 항상 면이 아래로 접히게 됩니다(접힌 선분이 위로 향하고 - 초록 실선).

면이 접히는 방향은 입력된 점의 순서에 따라 오른손 법칙에 의해 정의되므로 점이 hinge 컴포넌트에 입력되는 순서를 잘 정리하는 것이 중요합니다. 즉 그림 3에서 1번과 3번의 점이 접히는 선으로 정의가 되고 resting angle이 2도로 정의되면 오른손 법칙에 의해 2번 점이 0번 점에 다가가는 방향으로 면이 접히게 됩니다.

이제 Visual Basic을 살펴보면 가장 먼저 접히는 방향에 따라 두개의 별도 hinge component를 사용하기 위한 두개의 point set을 정의합니다(fig. 05).

```

190
191
192 '////////// define sets of points for hinge action
193
194
195 Dim fold_dn_pt0 As New List(Of Point3d)
196 Dim fold_dn_pt1 As New List(Of Point3d)
197 Dim fold_dn_pt2 As New List(Of Point3d)
198 Dim fold_dn_pt3 As New List(Of Point3d)
199
200 Dim fold_up_pt0 As New List(Of Point3d)
201 Dim fold_up_pt1 As New List(Of Point3d)
202 Dim fold_up_pt2 As New List(Of Point3d)
203 Dim fold_up_pt3 As New List(Of Point3d)
204
    
```

Figure 5

이는 앞서 이야기한 바와 같이, 패널과 패널사이의 접힘과 패널 내부에서의 접힘의 그 방향이 다르다는 것을 착안하여 별도로 2가지의 point set을 정의하고 이를 추후 hinge component에서 사용하기 위함입니다.

그리고 먼저 패널과 패널사이의 면의 접힘을 위한 point set을 추출하기 위해 현재의 패널의 패턴과 이웃한 다음 패널의 패턴을 분석합니다. 아래의 그림에서 A패턴의 패널이 그림상에서 아래쪽에 위치한 C패턴을 가진 이웃한 패널과 만날때에는 C패턴이 내부적으

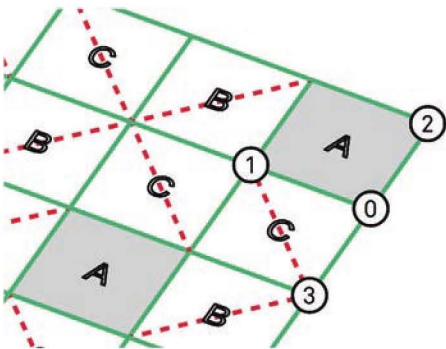


Figure 6

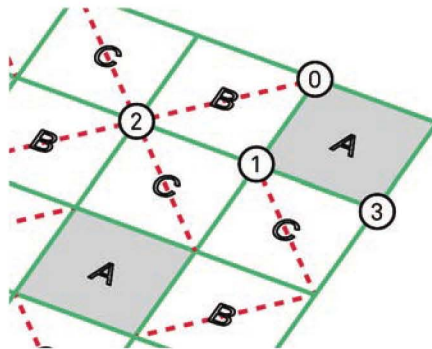


Figure 7

로 접힐 것을 고려하여 0-1-2-3의 점을 추출해야 함을 알 수 있습니다(fig. 06). 마찬가지로 A패턴의 패널이 그림상에서 좌측에 위치한 B패턴을 가진 이웃한 패널과 만날 때에는 B패턴이 내부적으로 접힐 것을 고려하여 0-1-2-3의 점을 추출해야 함을 알 수 있습니다 (fig. 07).

즉, 가로와 세로 방향을 모두 고려한 현재 패널의 패턴과 이웃하고 있는 패턴의 패널이 가질 수 있는 모든 인접의 시나리오를 고려하여 각 상황별 추출해야 하는 점을 미리 지정해야 합니다. 이는 아래 그림과 같이 for ~ next 및 if 문을 통하여 구현되었습니다(fig. 08/fig. 09 - 지면상 모든 script를 capture하지는 못했습니다).

```

207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
*/
branch-direction(x direction in rhino viewport) iteration for every possible hinge actions
*/
For j As Integer = 0 To v - 1
  For i As Integer = 0 To u - 2
    Dim pat_current As String = pat_tree.Item(pat_tree.Path(i), j)
    Dim pat_next As String = pat_tree.Item(pat_tree.Path(i + 1), j)
    Dim srf_current As Surface = srf_tree.Item(srf_tree.Path(i), j)
    Dim srf_next As Surface = srf_tree.Item(srf_tree.Path(i + 1), j)

    If pat_current = "A" And pat_next = "B" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 0)
      Dim pt1 As point3d = srf_current.PointAt(1, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 1)
      Dim pt3 As point3d = srf_next.PointAt(1, 1)

      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)

    Else If pat_current = "A" And pat_next = "C" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 0)
      Dim pt1 As point3d = srf_current.PointAt(1, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 0)
      Dim pt3 As point3d = srf_next.PointAt(1, 0)

      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)

    Else If pat_current = "B" And pat_next = "A" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 0)
      Dim pt1 As point3d = srf_current.PointAt(1, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 0)
      Dim pt3 As point3d = srf_next.PointAt(1, 0)

      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)

    Else If pat_current = "C" And pat_next = "A" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 0)
      Dim pt1 As point3d = srf_current.PointAt(1, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 1)
      Dim pt3 As point3d = srf_next.PointAt(1, 1)

      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)

    Else If pat_current = "B" And pat_next = "C" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 0)
      Dim pt1 As point3d = srf_current.PointAt(1, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 0)
      Dim pt3 As point3d = srf_next.PointAt(1, 0)

```

Figure 8

```

301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
*/*****
*/**** treestem-direction(-y direction in rhino viewport) iteration for every possible hinge actions
*/*****
For i As Integer = 0 To u - 1
  For j As Integer = 0 To v - 2
    Dim pat_current As String = pat_tree.Item(pat_tree.Path(i), j)
    Dim pat_next As String = pat_tree.Item(pat_tree.Path(i), j + 1)
    Dim srf_current As Surface = srf_tree.Item(srf_tree.Path(i), j)
    Dim srf_next As Surface = srf_tree.Item(srf_tree.Path(i), j + 1)
    If pat_current = "A" And pat_next = "B" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 1)
      Dim pt1 As point3d = srf_current.PointAt(0, 1)
      Dim pt2 As point3d = srf_current.PointAt(1, 0)
      Dim pt3 As point3d = srf_next.PointAt(1, 1)
      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)
    Else If pat_current = "A" And pat_next = "C" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 1)
      Dim pt1 As point3d = srf_current.PointAt(0, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 0)
      Dim pt3 As point3d = srf_next.PointAt(0, 1)
      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)
    Else If pat_current = "B" And pat_next = "A" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 1)
      Dim pt1 As point3d = srf_current.PointAt(0, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 0)
      Dim pt3 As point3d = srf_next.PointAt(0, 1)
      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)
    Else If pat_current = "C" And pat_next = "A" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 1)
      Dim pt1 As point3d = srf_current.PointAt(0, 1)
      Dim pt2 As point3d = srf_current.PointAt(1, 0)
      Dim pt3 As point3d = srf_next.PointAt(1, 1)
      fold_dn_pt0.Add(pt0)
      fold_dn_pt1.Add(pt1)
      fold_dn_pt2.Add(pt2)
      fold_dn_pt3.Add(pt3)
    Else If pat_current = "B" And pat_next = "C" Then
      Dim pt0 As point3d = srf_current.PointAt(1, 1)
      Dim pt1 As point3d = srf_current.PointAt(0, 1)
      Dim pt2 As point3d = srf_current.PointAt(0, 0)

```

Figure 9

이와 마찬가지로 패널내의 접힘을 위해 패널내부의 점들을 추출해야 합니다. 즉 패널이 접히는 방향에 따라 점의 순서도 달라지게 됩니다(fig. 10).

이는 Visual Basic에서 for ~ next 및 if 문을 사용하여 두가지 패턴에 대하여 그림 11과 같이 정의 됩니다(fig. 11).

