

파라메트릭 디자인 XIV

Parametric Design XIV

글. 성우제 Sung, Woojae

Grimshaw Architects / Associate

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

지난호에서는 2012년 가을학기 Harvard GSD의 Digital Media and Material Practice 수업의 일환으로 진행되었던 Diffusion Limited Aggregation System에 대하여 이야기를 시작했습니다. 먼저 유한 확산 집합체가 무엇인지를 자연에서 쉽게 발견할 수 있는 눈 결정의 현미경 사진을 통하여 알아 보았고 이 현상이 어떻게 일어나는 지에 대한 고찰을 통하여 유한 확산 집합체를 디지털 모형으로 생성하기 위한 밑그림을 그려보았습니다. 이번 회에서는 이를 디지털 모형으로 재구성했던 방법을 Visual Basic 을 통하여 단계별로 알아보도록 하겠습니다.

유한 확산 집합체의 첫번째 모듈은 지난회에 살펴 보았던 것처럼 일정한 공간상에서 브라운 운동을 하는 입자를 시뮬레이션 하는 것이었습니다. 이를 구현하기 위해서 가장 먼저 정의를 해야 했던 부분은 '일정한 공간상'이라고 하는 조건이었습니다. 실제로 눈 결정을 생성하는 수분의 분자는 끝이 없이 확장되어 있는 공기중이라는 공간상에 임의의 점에 위치하며 이 특정의 분자가 다음 순간에 어느 점으로 이동할 지, 그리고 어떠한 방향

으로 얼마만큼 이동할 지에 대한 범위에 대한 제한은 없다고 볼 수 있습니다. 물론 바람과 기압, 중력과 분자간의 인력의 영향이 그 범위에 영향을 미치겠지만, 패턴을 연구하기 위한 차원의 간략화된 시뮬레이션에서 그러한 영향을 모두 고려하여 상기의 '일정한 공간 상'의 범위를 정의 하기보다는 조금 더 직관적이고 효율적으로 패턴을 형성하는 주된 알고리즘을 실험 할 수 있는 범위를 설정하는 것이 보다 현실적이라는 판단하에 임의의 원안으로 그 범위를 좁히게 됩니다. 분자들이 이동할 수 있는 범위를 제한 함으로써 반응이 일어나는 속도를 높이게 되며 반응을 일으키지 않고 사라져 버리는 분자를 시뮬레이션에서 원천적으로 제외를 함으로써 반응의 효율을 높이게 됩니다. (fig. 1)

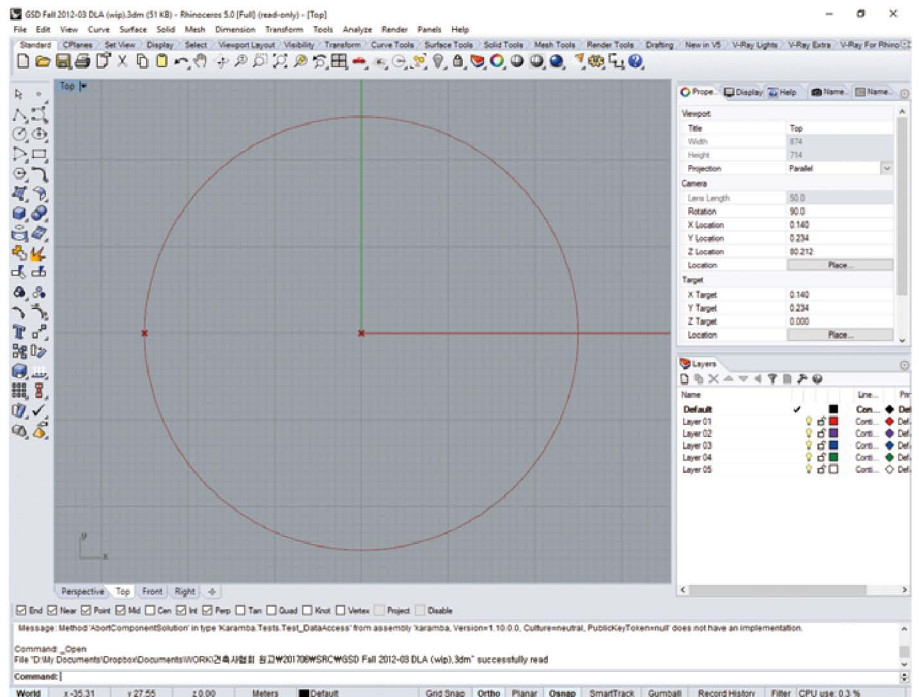


Figure 1

원을 반응이 일어날수 있는 공간의 경계로 정의를 하였다라는 것은 특정의 분자가 그 원의 한 점에서 출발을 하여야 한다는 것을 의미하므로 반응을 시작하기 위한 분자는 반드시 해당하는 원의 한점에 있어야 합니다. 이는 Visual Basic 에서 다음과 같은 코드로 표현이 될수 있습니다. (fig. 2)

```

Script Editor
Script component VB
1 Option Strict Off
2 Option Explicit On
3
4 Imports Public Class Script_Instance
36 Inherits GH_ScriptInstance
37
38 Public Functions
61
62 Members
75
76
81 Private Sub RunScript(ByVal attr As Point3d, ByVal bndry As Curve, ByVal go As Boolean, ByRef A As Object)
82
83     If go = False Then
84
85         'step 01 - get a random point on the given boundary
86         getrndpt(bndry, pt)
87
88     Else If go = True
89
90         'doesnt happen anything yet here
91
92     End If
93
94     A = pt
95
96 End Sub
97
98
99
100 Dim pt As point3d
101
102 Private Sub getrndpt(ByVal bndry As Curve, ByRef rndpt As point3d)
103
104     Dim rnd As Random = New Random()
105     Dim t As Double = rnd.Next(0, 10) / 10
106     print(t)
107     rndpt = bndry.PointAtNormalizedLength(t)
108
109 End Sub
110
111
112 End Class
Cache Recover from cache OK

```

Figure 2

위의 코드에서 104-105행은 랜덤 함수를 이용하여 0에서 10 사이의 임의의 정수를 생성하고 이를 다시 10으로 나눔으로써 0과 1사이의 유리수를 구해내는 과정입니다. 컴퓨터에서 구현되는 랜덤은 물론 우리가 패턴을 인식할 수 없는 거대한 패턴을 사용하여 구현되

는 패턴화된 수이지만 본 시물레이션에서는 충분하다고 생각되어 그대로 사용을 하게 되었습니다. 상기의 과정은 0과 1사이의 수를 10으로 나눔으로써 소수점 한자리수까지의 밀도를 가지게 되나 이는 물론 랜덤함수의 최대값과 이를 나누는 값을 바꿈으로써 유리수의 밀도를 더욱 증가 시킬 수 있습니다. 106행에서는 시물레이션의 경계를 나타내는 원, bndry라는 곡선의 길이를 1이라고 가정하였을때 그 곡선상에 위에서 구한 랜덤 값을 통해 점을 위치시키는 과정을 설명하고 있습니다. 이러한 과정은 102행에서부터 109행까지 하나의 모듈로 함수화 되어서 83행에서 코드의 실행을 위한 스위치가 비활성화가 되어 있을시 86행을 통하여 실행됩니다. 즉 본격적인 시물레이션이 시작되기 앞서서 시물레이션의 첫 점이 될 점을 원상의 한 점으로 랜덤하게 정해 놓는 것입니다. 이렇게 구해진 점은 94행을 통하여 라이노의 뷰포트에 나타나게 됩니다.

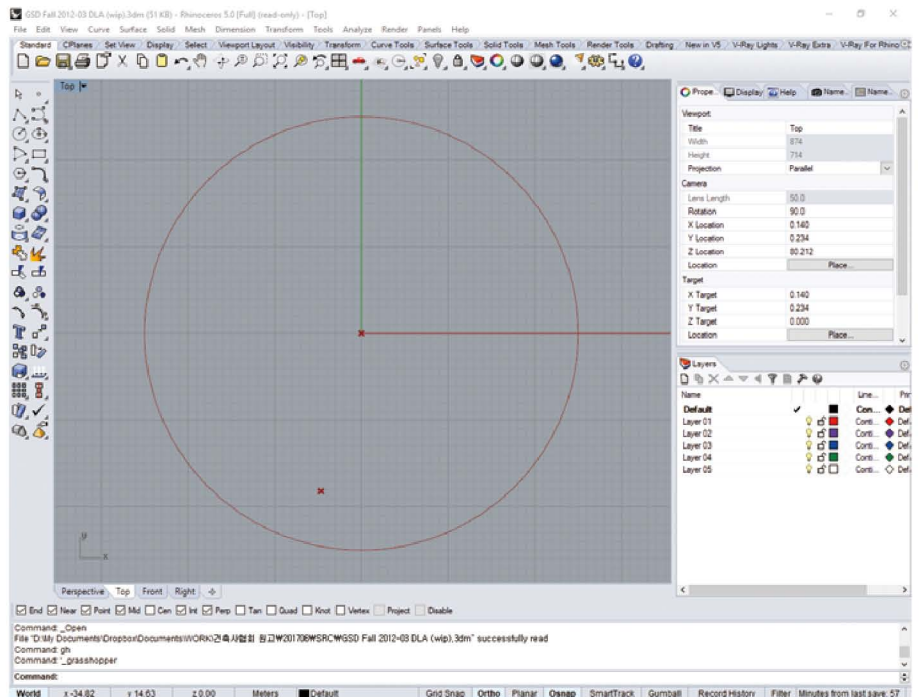


Figure 3

위의 과정을 통하여 시뮬레이션의 경계가 되는 영역을 원을 통하여 설정하였으며 시뮬레이션의 첫 단추가 되는 첫번째 분자의 위치를 그 경계가 되는 영역상의 한점으로 정의를 했습니다. 또한 첫번째 그림에서 보는 바와 같이 원의 중심이 되는 곳에 seed가 되는 점을 위시시켜 이 점이 결정의 핵이 되도록 설정해 놓았습니다. 이제 두번째 과정에서는 경계상의 임의의 점이 원의 중심을 향하여 접근하는 브라운 운동을 간략하게 시뮬레이션해 보도록 합니다. (fig. 3)

브라운 운동은 물론 이렇게 간단히 구현되지는 않으나 간략하고 효율적인 패턴의 시뮬레이션을 위해서 고안된 방법은 다음과 같습니다. 점의 현재 위치에서 seed점까지를 잇는

```

Script Editor
Script component VB
1 Option Strict Off
2 Option Explicit On
3
4 Imports Public Class Script_Instance
5 Inherits GM_ScriptInstance
6
7 Utility Functions
8
9 Members
10
11
12 Private Sub RunScript(ByVal attr As Point3d, ByVal bndry As Curve, ByVal go As Boolean, ByRef A As Object)
13
14     If go = False Then
15
16         'step 01 - get a random point on the given boundary
17         GetRndPt(bndry, pt)
18
19     Else If go = True
20
21         'doesn't happen anything yet here
22
23     End If
24
25     A = pt
26
27 End Sub
28
29
30 Dim pt As point3d
31
32 Private Sub GetRndPt(ByVal bndry As Curve, ByRef rndpt As point3d)
33
34     Dim rnd As Random = New Random()
35     Dim t As Double = rnd.Next(0, 10) / 10
36     print(t)
37     rndpt = bndry.PointAtNormalizedLength(t)
38
39 End Sub
40
41
42 End Class

```

Figure 4

벡터를 정의하고 이 벡터를 좌우 45도 방향으로 랜덤하게 회전시켜서 이 벡터를 이용하여 현재의 점을 다음 순간의 점으로 이동시키도록 합니다. (fig. 4)

위의 코드에서 113행에서부터 124행까지가 간략한 브라운 운동을 시뮬레이션하는 함수의 모듈입니다. 115행 - 116행에서 - 45에서 45까지의 임의의 정수를 랜덤 함수를 통하여 생성하고 117 - 118 행에서 현재의 점에서 seed가 되는 점사이의 벡터를 구합니다. 119 - 120 행을 통하여 - 45에서 45도 사이의 임의의 값으로 회전시킨 후 이 회전된 벡터를 이용하여 122행에서 주어진 점을 이동시키게 됩니다. 이러한 브라운 운동의 모듈은 88행에서 스위치가 활성화 되었을 때 91행을 통하여 실행되며 이 과정에서 스위치가 비활성화가 되었을시 생성된 원 위의 임의의 점을 이동시키고 이를 92행 및 96행을 통하여 라이노의 화면에 나타나게 해 줍니다.

다음회에서는 유한 확산 집합체의 나머지 부분에 대해서 살펴보도록 하겠습니다. 참고로 오늘 이야기한 내용은 다음 주소에서 더욱 자세하게 살펴볼 수 있습니다.

<https://woosung.com/2012/10/14/grasshopper-vb-workshop-harvard-gsd-fall-2012/>