

고급 3D 스크립팅

MAYA PYTHON SCRIPT를 활용한 아트웍

Vincent

Van

Gogh

기밀파일처리

CONTENTS

1. 주제

2. 작품 소개 및 구현 방법

3. 예술적 요소 및 차별성

4. 결과

01

주제



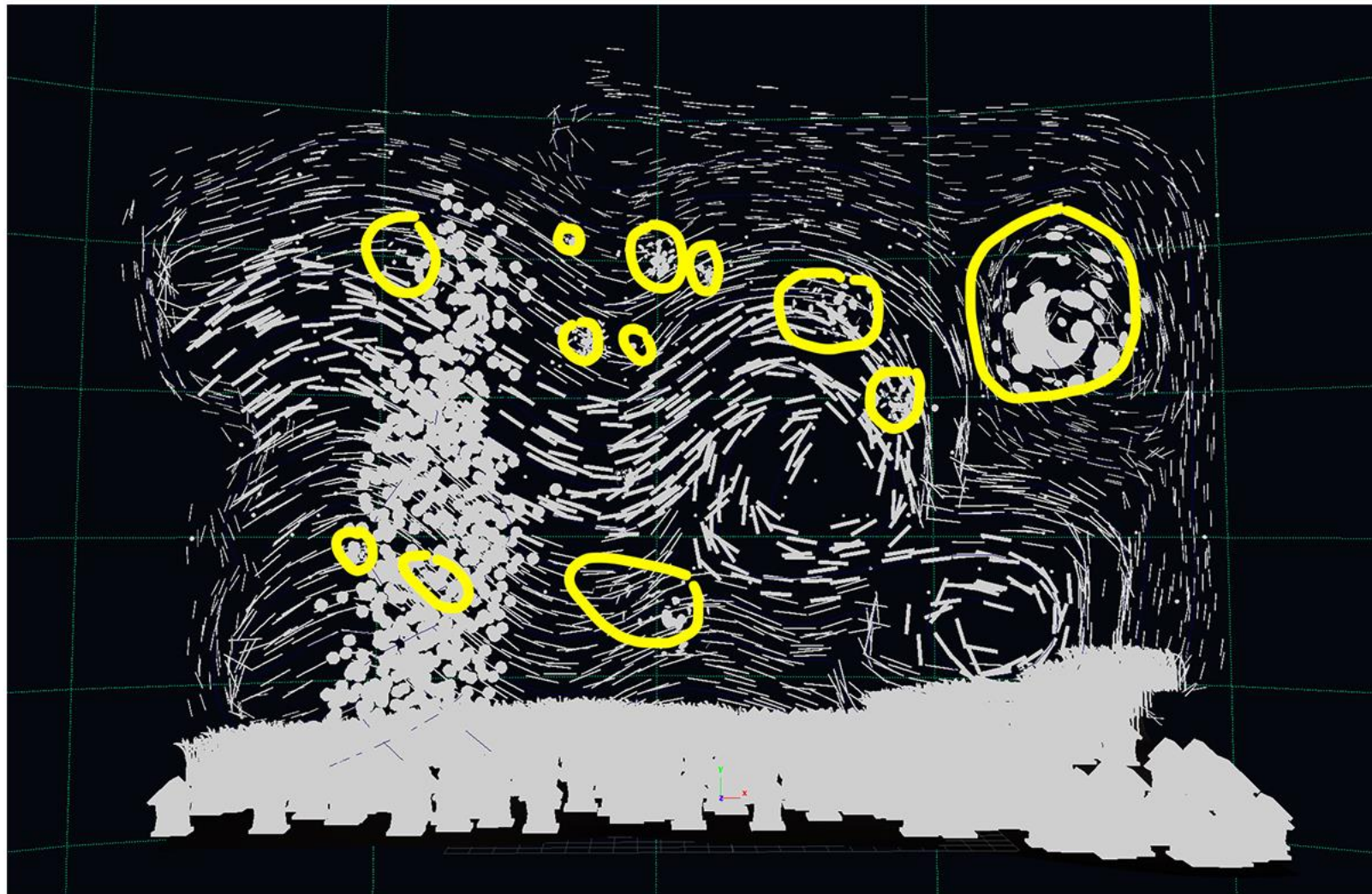
빈센트 반 고흐의 '별이 빛나는 밤' 재해석

Maya python script와 그 외의 기능을 활용해
영화 '별이 빛나는 밤'을 제작

02

작품 소개 및 구현

1. 스크립팅 11개의 별과 달 부분을 스크립트로 구성



02

작품 소개 및 기획

1. 스크립팅 11개의 별과 달 부분을 스크립트로 구성

randomInstances.py #선택한 오브젝트를 일정한 범위 안에 생성

```
1 # randomInstances.py
2 import maya.cmds as cmds
3 import random
4
5 random.seed( 1234 )
6
7 result = cmds.ls( orderedSelection=True )
8
9 print('result: %s' % (result))
10
11 transformName = result[0]
12
13 instanceGroupName = cmds.group( empty=True, name=transformName + '_instance_grp' )
14
15 for i in range(0, 50):
16     instanceResult = cmds.instance(transformName, name=transformName + '_instance#')
17     cmds.parent(instanceResult, instanceGroupName)
18     # print('instanceResult: ' + str(instanceResult))
19
20     x = random.uniform(-5, 5)
21     y = random.uniform(-36, 46)
22     z = random.uniform(82, 95)
23     # 주어진 범위 안에서 랜덤으로 생성
24
25     cmds.move(x, y, z, instanceResult)
26
27     xRot = random.uniform(0, 360)
28     yRot = random.uniform(0, 360)
29     zRot = random.uniform(0, 360)
30     # 주어진 범위 안에서 랜덤으로 회전
31
32     cmds.rotate( xRot, yRot, zRot, instanceResult )
33
34     scalingFactor = random.uniform(0.3, 1.5)
35
36     cmds.scale( scalingFactor, scalingFactor, scalingFactor, instanceResult )
37
38     cmds.hide( transformName )
39
40     cmds.xform( instanceGroupName, centerPivots=True )
41
42
43
```

aimAtFisrt.py #선택된 오브젝트들 중 제일 먼저 선택된 오브젝트를 향하게 함

```
1 # aimAtFisrt.py
2 import maya.cmds as cmds
3
4 selectionList = cmds.ls(orderedSelection=True)
5
6 if len(selectionList) >= 2:
7     print('Selected items: %s' % (selectionList))
8
9     targetName = selectionList[0]
10     selectionList.remove(targetName)
11     for objectName in selectionList:
12         print('Constraining %s towards %s' % (objectName, targetName))
13         cmds.aimConstraint(targetName, objectName, aimVector=[0,1,0])
14
15 else:
16     print('Please select two or more objects.')
17
18
```

keyRotationUI.py #키프레임 값을 설정하여 회전하도록 하는 것을 UI로 구현

```
1 import maya.cmds as cmds
2 import functools
3
4 def createUI(pWindowTitle, pApplyCallback):
5     windowID = 'myWindowID'
6
7     if cmds.window(windowID, exists=True):
8         cmds.deleteUI(windowID)
9
10    cmds.window(windowID, title=pWindowTitle, sizeable=False, resizeToFitChildren=True)
11    cmds.rowColumnLayout(numberOfColumns=3, columnWidth=[(1,75), (2,60), (3,60)], columnOffset=[(1,'right',3)])
12    cmds.text(label='Time Range:')
13
14    startTimeField = cmds.intField(value = cmds.playbackOptions(q=True, minTime=True))
15    endTimeField = cmds.intField(value = cmds.playbackOptions(q=True, maxTime=True))
16    cmds.text(label='Attribute:')
17    targetAttributeField = cmds.textField(text='rotateY')
18
19    cmds.separator(h=10, style='none')
20    # GUI를 생성
21    cmds.separator(h=10, style='none')
22    cmds.separator(h=10, style='none')
23    cmds.separator(h=10, style='none')
24
25    cmds.button(label='Apply', command=functools.partial(pApplyCallback,
26                                                         startTimeField,
27                                                         endTimeField,
28                                                         targetAttributeField))
29
30    def cancelCallback(pArgs):
31        if cmds.window(windowID, exists=True):
32            cmds.deleteUI(windowID)
33
34    cmds.button(label='Cancel', command=cancelCallback)
35    cmds.showWindow()
36    # 지정된 객체에 대해 특정 속성의 키프레임을 설정
37
38    def keyFullRotation(pObjectName, pStartTime, pEndTime, pTargetAttribute):
39        cmds.cutKey(pObjectName, time=(pStartTime, pEndTime), attribute=pTargetAttribute)
40        cmds.setKeyframe(pObjectName, time=pStartTime, attribute=pTargetAttribute, value=0)
41        cmds.setKeyframe(pObjectName, time=pEndTime, attribute=pTargetAttribute, value=160)
42        cmds.selectKey(pObjectName, time=(pStartTime, pEndTime), attribute=pTargetAttribute, keyframe=True)
43        cmds.keyTangent(inTangentType='linear', outTangentType='linear')
44
45    def applyCallback(pStartTimeField, pEndTimeField, pTargetAttributeField, pArgs):
46        # print 'Apply button pressed.'
47
48        startTime = cmds.intField(pStartTimeField, query=True, value=True)
49        endTime = cmds.intField(pEndTimeField, query=True, value=True)
50        targetAttribute = cmds.textField(pTargetAttributeField, query=True, text=True)
51
52        print ('Start Time: %s' % (startTime))
53        print ('End Time: %s' % (endTime))
54        print ('Attribute: %s' % (targetAttribute))
55        # 사용자가 지정한 속성(attribute)에 대해
56        # 선택한 모든 오브젝트에 대해 키프레임을 설정
57
58        selectionList = cmds.ls(selection=True, type='transform')
59        for objectName in selectionList:
60            keyFullRotation(objectName, startTime, endTime, targetAttribute)
61
62    createUI('My Title', applyCallback) # cmds.keyTangent를 사용하여 키프레임의 탄젠트
63
```


02

작품 소개 및 환경

1. 스크립팅 모든 공간에 존재하는 작은 별들

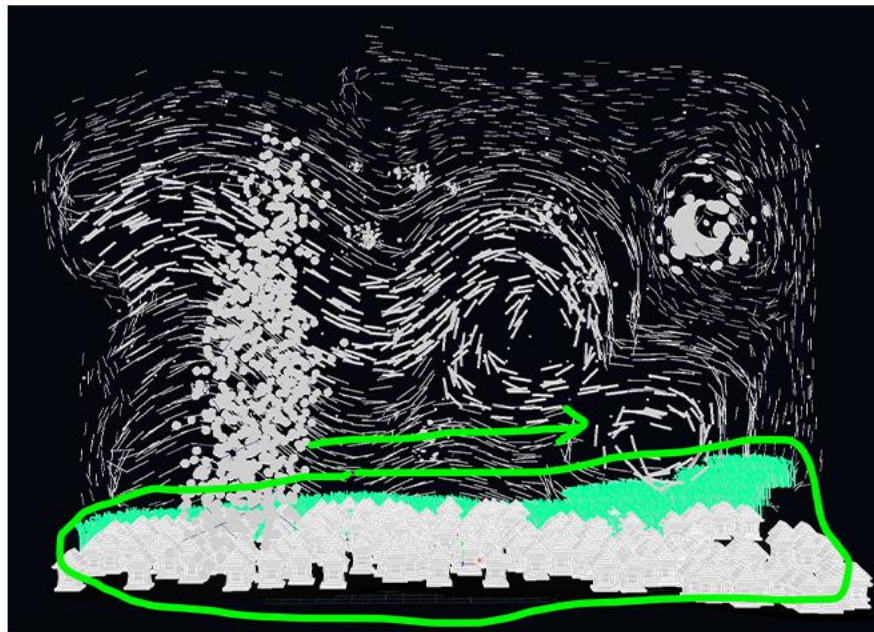
star.py # CSV 파일을 사용 - x, y, z값에 각각 랜덤 변수인 (=RANDBETWEEN(,))을 사용

```
MEL x randomInstances.py x aimAtFirst.py x keyRotation.py x keyRotationWithUI.py x star.py x +
1 from csv import reader
2 import maya.cmds as cmds
3
4 f = open('/Users/eungyeulsssss/Desktop/data/star.csv', 'r')
5 data = reader(f)
6 next(data)
7
8 for i, row in enumerate(data):
9     x = float(row[1])
10    y = float(row[2])
11    z = float(row[3])
12    sphere = cmds.polySphere(r=0.1)[0]
13    cmds.move(x, y, z, sphere)
14
```

02

작품 소개 및 구현

2. 비스크립팅



지형

soft selection 옵션으로 vertex 이동
→ 오른쪽으로 높아지는 지형의 고도 표현

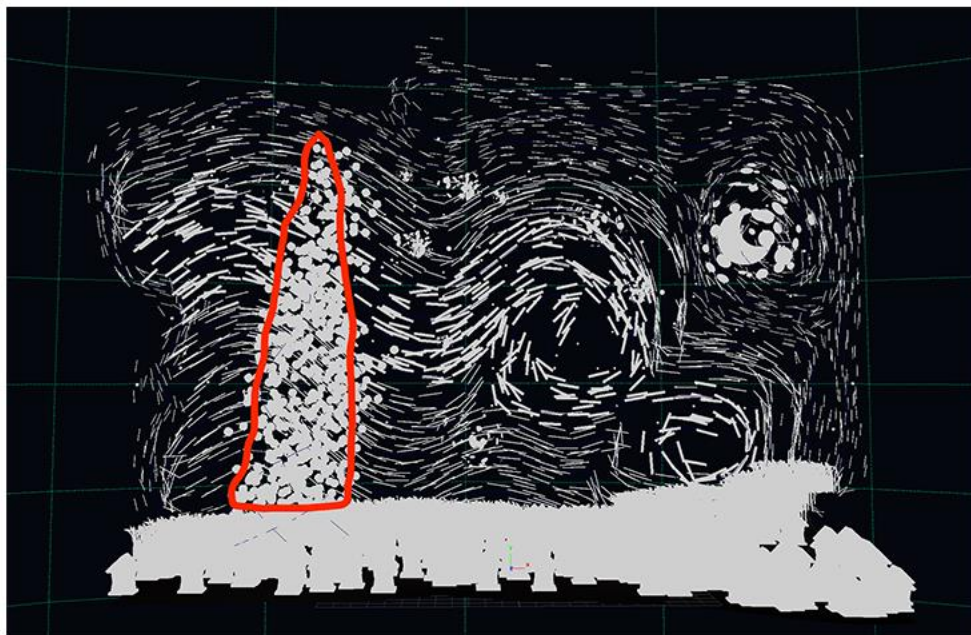
집, 언덕

제작된 지형의 앞: 집 배치 / 제작된 지형의 뒤: 언덕 배치
모델링 된 house 오브제 →
MASH 적용 →
Distribute-Distribution Type: Mesh
→ Input Mesh에 제작한 지형 연결
→ 방향 설정: Calculate Rotation 체크 해제
→ Number of point: 원하는 개수만큼 설정

02

작품 소개 및 기획

2. 비스크립팅



사이프러스 나무

정면 기준 가장 앞부분에 배치

위치에 맞춰 기준이 될 세 개의 라인을 EP curves tool로 그림

Sphere를 기본 요소로 정하고 MASH를 적용

Distribute에서 sphere의 개수를 높여 700개 정도의 폴리곤 생성

MASH에 Curve Node를 추가 -> 세 개의 라인을 연결하면 라인에 sphere들이 붙음

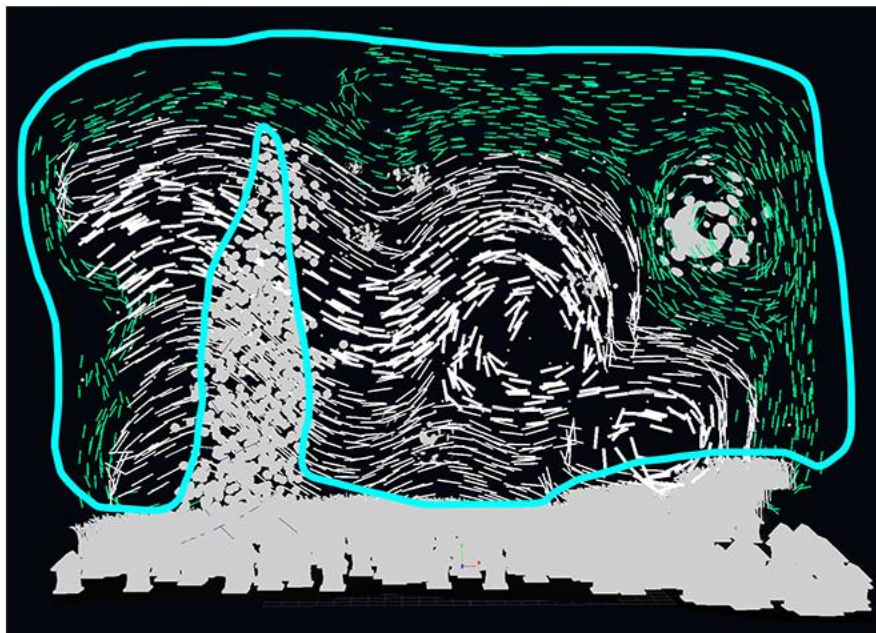
step 옵션으로 간격을 조정, Animation speed를 모션속도를 조정, Clip End로 끝나는 지점을 조정

Signal Node를 추가해서 position Y값을 높임 -> 더 풍성한 표현 가능

02

작품 소개 및 구현

2. 비스크립팅



바람, 배경 별빛의 흐름

바람의 흐름, 배경 별빛의 흐름을 각 위치와 모양에 맞게 라인 생성

기본 요소 : 얇고 긴 Cylinder -> 작품에서 사용된 스트로크 기법을 표현

(이후 작업은 사이프러스 나무 제작과 동일하여 생략)

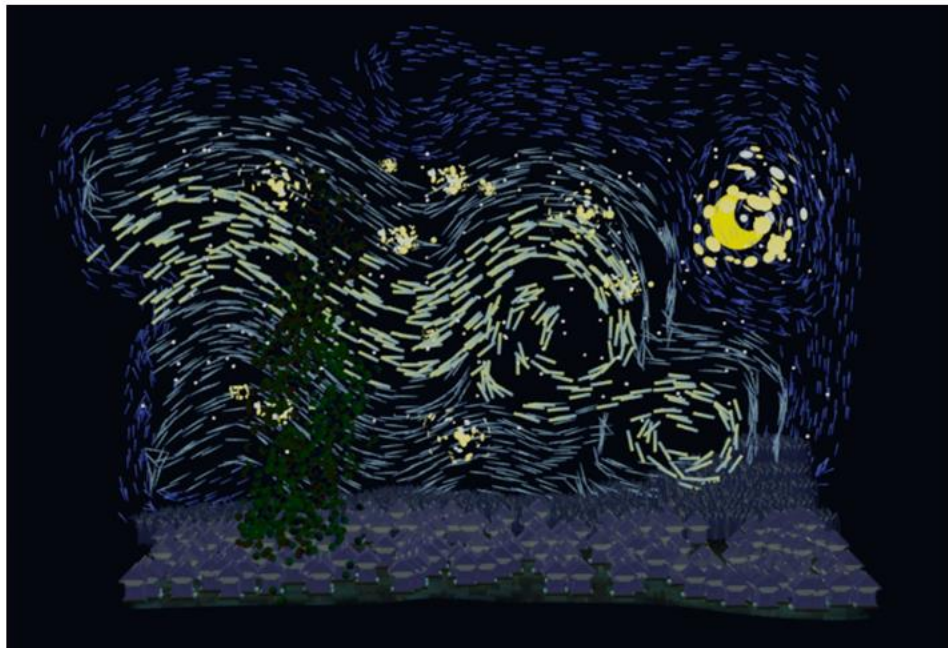
Animation Speed 변화(뒷배경으로 갈수록 느리게) + Cylinder의 두께(뒤로 갈수록 얇게)

-> 원근감 표현

02

작품 소개 및 구현

2. 비스크립팅



색

어두운 공간에서 작품이 보일 수 있도록 그래데이션 발광체로 제작
aistandardsurface의 Emission Weight(발광)를 1로 높임
→ aiRampRGB(그래데이션)의 out color를 emission color 노드에 연결
→ Type은 v로 선택하고 Ramp에서 각각의 요소에 적합한 색을 조합하여 적용

03

예술적 요소

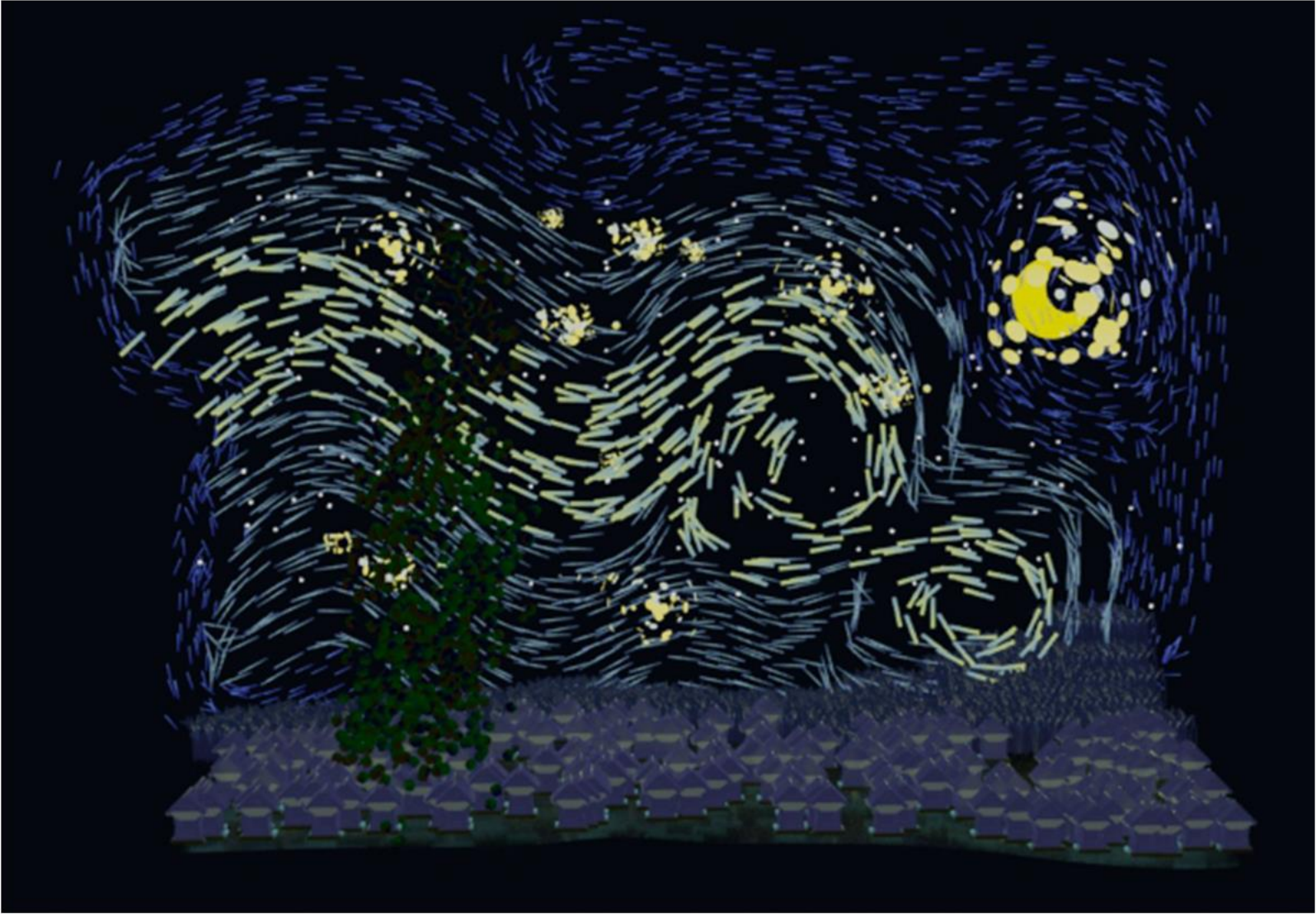
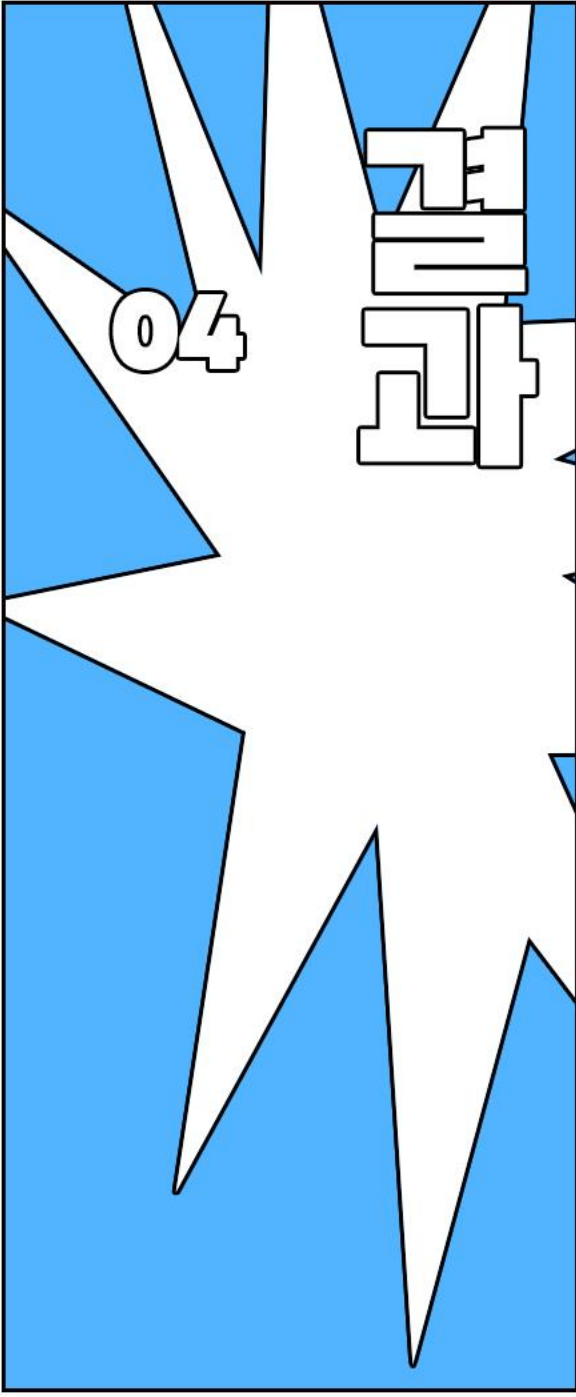
예술적 요소

‘별이 빛나는 밤’ 작품 이름에 걸맞게 오브젝트가 스스로 빛나도록 함
입체감을 느낄 수 있도록 배치

별의 궤도, 별이 반짝이는 표현, 바람의 흐름을 애니메이션을 통해 동적 표현

차별점

애니메이션과 입체감



THANK YOU