Direct 3D
포트폴리오 소개서

정다경

Athas(아서스)

소스 코드 (github) : https://github.com/jdk211/Athas
영상 링크 (youtube) : https://youtu.be/sFxwIgGIIa0

개발 환경 : visual studio 2015, Direct, C++

[특징 및 기능]

- 팀(2인) 개발, 맵툴 제작, 기타
- World of Warcraft에 나오는 아서스를 키우는 게임
- Picking을 이용한 높이맵 제작
- 구조물 설치와 Scale, Roate 변경 가능
- OBB충돌을 이용한 건물과 플레이어의 충돌체크

How To Map

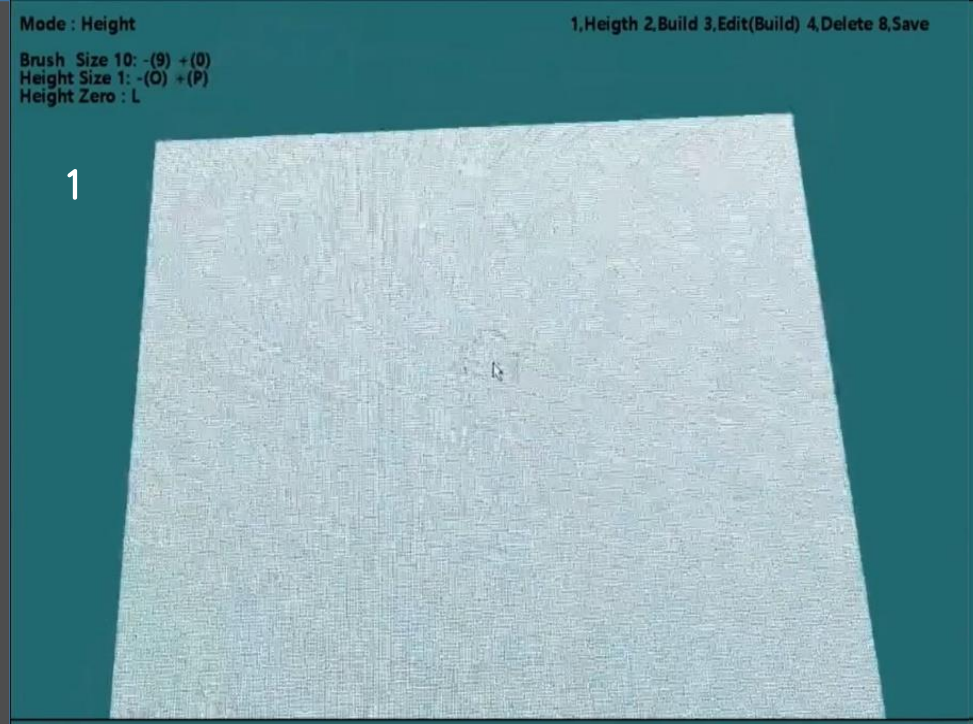
01

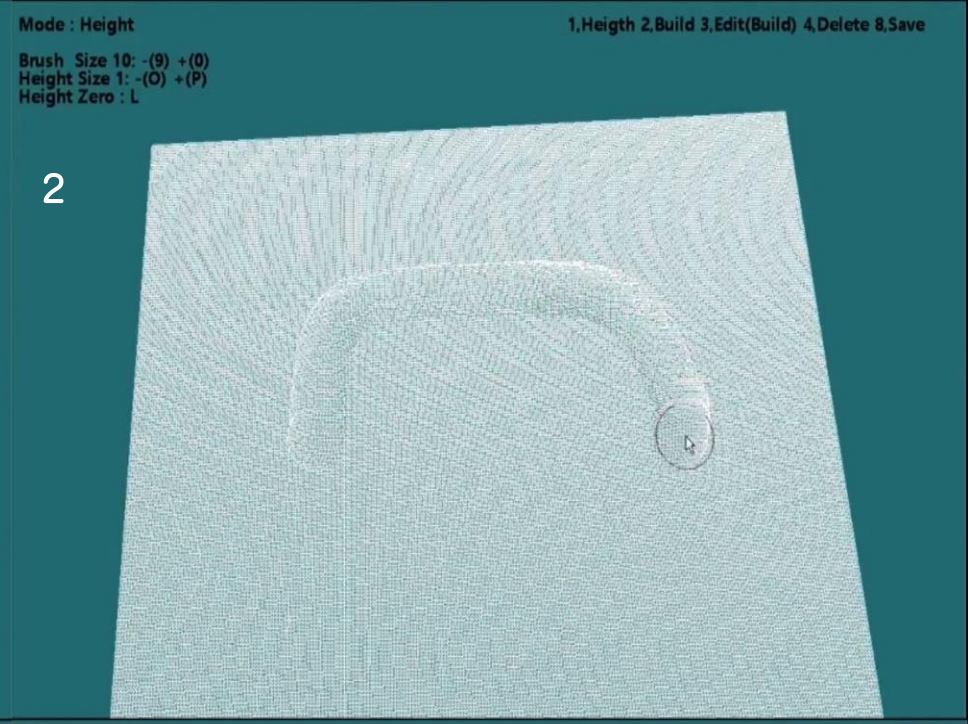
1, 맵의 크기에 따른 Vertex를 만들어 면을 만들어 줍니다.

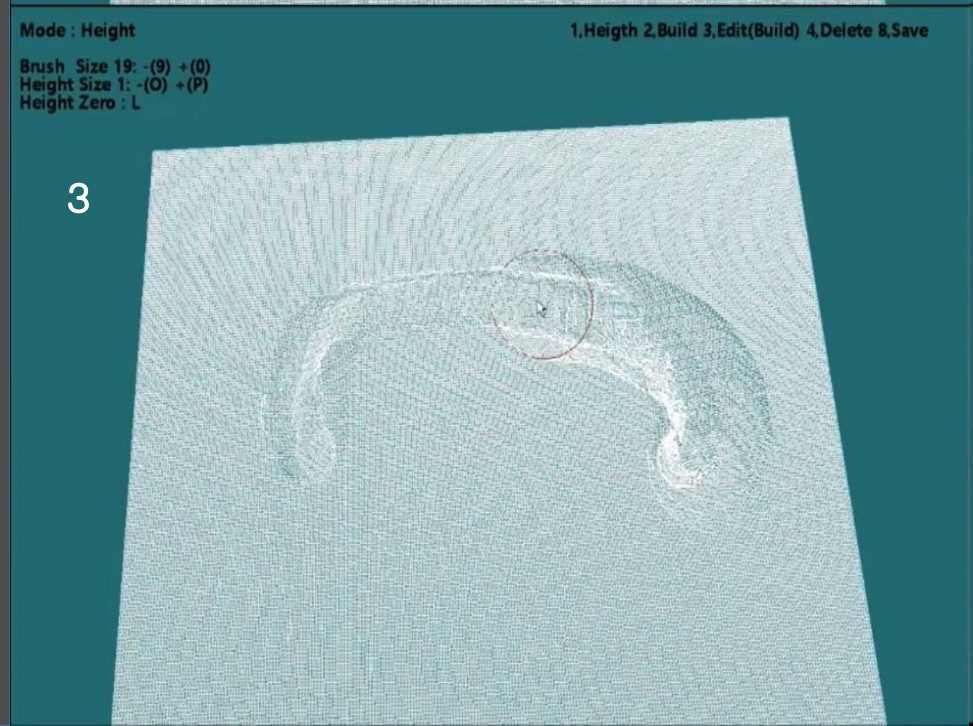
2, Picking을 중심으로 Brush Size안에 있는 Vertex들의 Y값을 변경해 맵의 높이를 수정해줍니다.

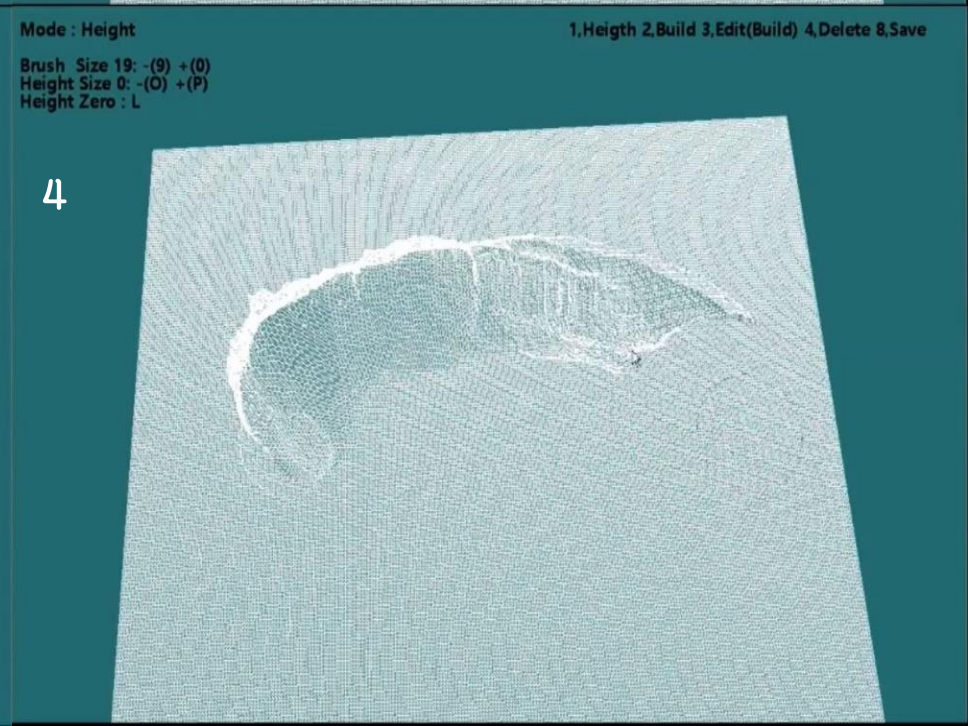
3, Height Size 를 변경해서 -값으로 땅을 내릴수도 있고 +갑스로 다시 올릴수도있고 0으로 초기화 할 수 있습니다.

4. Brush Size를 변경해서 높이 변경할 범위를 바꿀수있습니다.









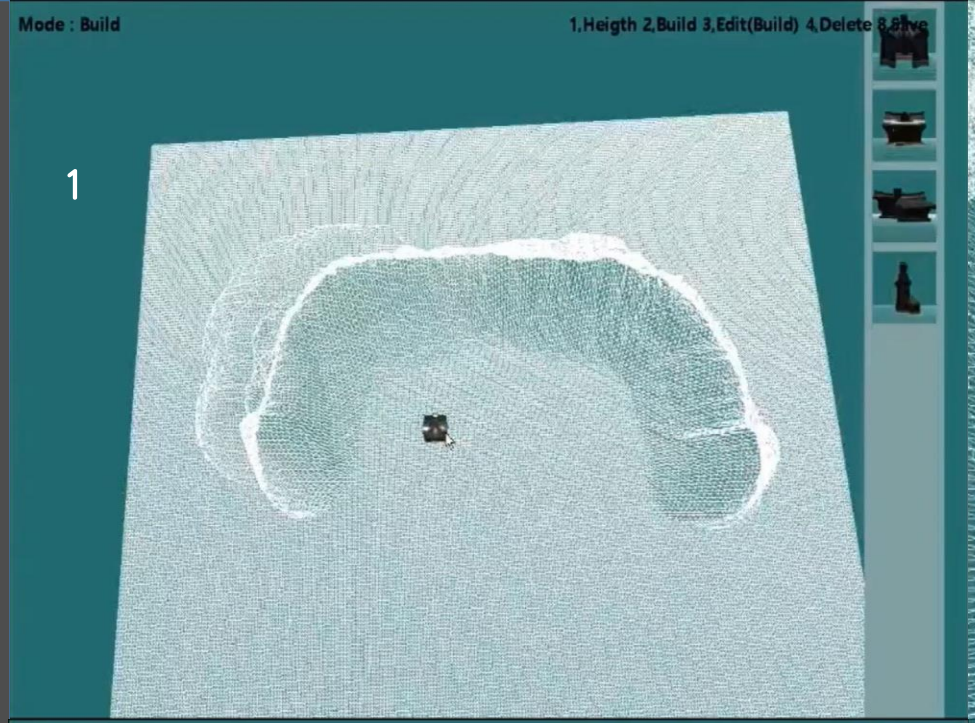
How To Map

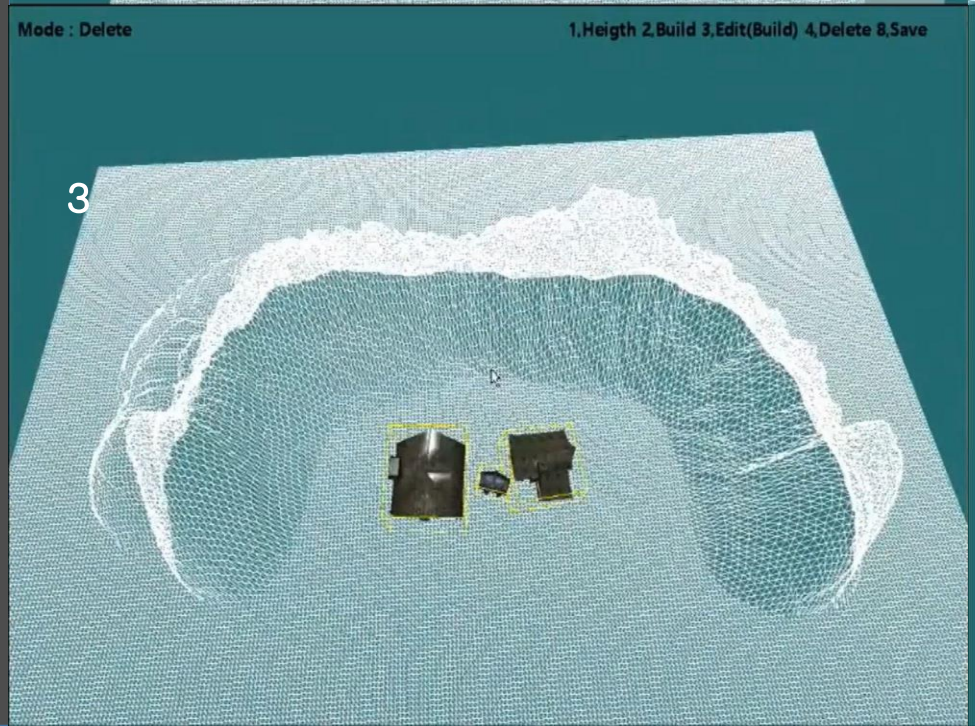
02

1~2, Mode를 변경해 건물도 설치가 가능

3, 설치한 건물의 Scale, Rotation, Delete가 가능합니다.

4, 저장 후 씬에 적용할 수 있습니다.





Map Tool

01

3d 공간에 Vertex를 찍어 면을 만들어 줍니다.

```cpp
void cScene_MapTool::CreateGround()
{
	//점을 정의
	for (int i = 0; i < m_nTotalVertex; ++i)
	{
		PNTVERTEX v;

		v.p = D3DXVECTOR3(i % m_nMapSize - (m_nMapSize / 2), 0, i / m_nMapSize - (m_nMapSize / 2));
		v.n = D3DXVECTOR3(0, 1, 0);
		v.t = D3DXVECTOR2((i % m_nMapSize) / (float)m_nMapSize, (i / m_nMapSize) / (float)m_nMapSize);

		m_pVertex[i] = v;
	}

	for (int z = 1; z < m_nTileNum; ++z)
	{
		for (int x = 1; x < m_nTileNum; ++x)
		{
			int left = (z + 0) * m_nMapSize + x - 1;
			int right = (z + 0) * m_nMapSize + x + 1;
			int up = (z + 1) * m_nMapSize + x + 0;
			int down = (z - 1) * m_nMapSize + x + 0;

			D3DXVECTOR3 leftToRight = m_pVertex[right].p - m_pVertex[left].p;
			D3DXVECTOR3 downToUp = m_pVertex[up].p - m_pVertex[down].p;
			D3DXVECTOR3 normal;
			D3DXVec3Cross(&normal, &downToUp, &leftToRight);
			D3DXVec3Normalize(&normal, &normal);

			int nIndex = z * m_nMapSize + x;
			m_pVertex[nIndex].n = normal;
		}
	}

	//점에 순서를 정해준다
	for (int z = 0; z < m_nTileNum; ++z)
	{
		for (int x = 0; x < m_nTileNum; ++x)
		{
			int _0 = (z + 0) * m_nMapSize + x;
			int _1 = (z + 1) * m_nMapSize + x;
			int _2 = (z + 0) * m_nMapSize + x + 1;
			int _3 = (z + 1) * m_nMapSize + x + 1;

			m_vecIndex.push_back(_0);
			m_vecIndex.push_back(_1);
			m_vecIndex.push_back(_2);

			m_vecIndex.push_back(_3);
			m_vecIndex.push_back(_2);
			m_vecIndex.push_back(_1);
		}
	}

	//메시를 설정
	D3DXCreateMeshFVF(
		m_vecIndex.size() / 3,
		m_nTotalVertex,
		D3DXMESH_MANAGED | D3DXMESH_32BIT,
		PNT_VERTEX::FVF,
		Device,
		&m_pMesh
	);

	//메시에 vertex, index 저장
	ResetMesh();

	DWORD* pAttr = NULL;
	m_pMesh->LockAttributeBuffer(0, &pAttr);
	ZeroMemory(pAttr, (m_vecIndex.size() / 3) * sizeof(DWORD));
	m_pMesh->UnlockAttributeBuffer();

	//메시 최적화
	vector<DWORD> vecAdj(m_vecIndex.size());
	m_pMesh->GenerateAdjacency(0.0f, &vecAdj[0]);
	m_pMesh->OptimizeInplace(
		D3DXMESHOPT_COMPACT | D3DXMESHOPT_ATTRSORT | D3DXMESHOPT_VERTEXCACHE,
		&vecAdj[0],
		NULL,
		NULL,
		NULL
	);
```

Map Tool

02

Brush Size만큼 원안에 들어있는
Vertex들의 Heigth를 변경합니다.

```
313 void cScene_MapTool::FixGroundHeight()
314 {
315         if (KEY_MGR->IsStayDown(VK_LBUTTON))
316         {
317                 POINT ptMouse = GetMousePos();
318                 D3DXVECTOR3 curVertex(0, 0, 0);   //브러쉬 안에 정점
319                 D3DXVECTOR3 tempHit;
320
321                 this->mainCamera.GetWorldRay(&this->m_ClickRay, ptMouse);
322
323                 if (IsIntersectRay(&tempHit, &this->m_ClickRay))
324                 {
325                         float fX = ((m_nMapSize / 2) - (int)tempHit.x);
326                         float fZ = ((m_nMapSize / 2) - (int)tempHit.z) * m_nMapSize;
327
328                         int PickIndex = m_nTotalVertex - (fZ + fX);
329
330                         //w * i + i / (w * 6)
331                         for (int z = -m_nBrushRadius; z < m_nBrushRadius; ++z)
332                         {
333                                 for (int x = -m_nBrushRadius; x < m_nBrushRadius; ++x)
334                                 {
335                                         //((z * m_nMapSize) + PickIndex) + x; 네모
336
337                                         int index = ((z * m_nMapSize) + PickIndex) + x;
338
339                                         //예외처리 추가
340                                         if (index < 0 || index > m_nTotalVertex) continue;
341
342                                         //원 검사
343                                         curVertex = m_pVertex[index].p;
344
345                                         m_vPickingPos = m_pVertex[index].p;
346
347                                         float len = (curVertex.x - tempHit.x)*(curVertex.x - tempHit.x) + (curVertex.z - tempHit.z)*(curVer
348
349                                         // 차후 다른 알고리즘을 적용해서 자연스럽게 증가되게 만든다.
350                                         // 중심점으로 부터 거리가 멀수록 증가값이 작아지게
351                                         //float y = 1 - 1.0f / m_nBrushRadius * len * /*m_fSlopeRate*/ + m_nBrushRadius;
352
353                                         //float y = 0.01f - (sqrt(len) / (m_nBrushRadius * m_fGapSize));
354                                         float y = cosf(sqrt(len) / m_nBrushRadius) * m_nHeightSize;
355
356                                         if (len < ((m_nBrushRadius) * (m_nBrushRadius)))  // 정점이 원안에 있다면
357                                         {
358                                                 if (y == 0) m_pVertex[index].p.y = y;
359                                                 else m_pVertex[index].p.y += y;
360                                         }
361                                 }
362                         }
363
364                         ResetMesh();
365
366                 }
367         }
368 }
```

Map Tool

03

Scene에 있는 Vertex의 높이값과
Build한 건물들을 저장합니다.

```cpp
void cScene_MapTool::SaveMap()
{
	FILE* outFile;

	fopen_s(&outFile, "./Resources/Textures/HeightValue.txt", "w");

	for (int i = 0; i < m_nTotalVertex; i++)
	{
		fprintf_s(outFile, "%f ", m_pVertex[i].p.y);
	}

	fclose(outFile);

	SaveBuild();
}

void cScene_MapTool::SaveBuild()
{
	FILE* outFile;

	fopen_s(&outFile, "./Resources/Textures/Build.txt", "w");

	if (m_vecSetBuild.size() == NULL)
	{
		fprintf_s(outFile, "noBuild");
	}
	else
	{
		for (int i = 0; i < m_vecSetBuild.size(); i++)
		{
			fprintf_s(outFile, "K %s\n", m_vecSetBuild[i]->skinPath);
			fprintf_s(outFile, "P %f %f\n", m_vecSetBuild[i]->position.x, m_vecSetBuild[i]->position.z); //포지션 저장
			fprintf_s(outFile, "S %f %f %f\n", m_vecSetBuild[i]->trans.GetScale().x, m_vecSetBuild[i]->trans.GetScale().y, m_ve
			fprintf_s(outFile, "1 %f %f %f\n", m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._11,
				m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._12,
				m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._13);
			fprintf_s(outFile, "2 %f %f %f\n", m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._21,
				m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._22,
				m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._23);
			fprintf_s(outFile, "3 %f %f %f\n", m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._31,
				m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._32,
				m_vecSetBuild[i]->trans.GetWorldRotateMatrix()._33);
			fprintf_s(outFile, "B \n");
		}
	}

	fclose(outFile);

}
```

*저장된 건물 값

```
K ./Resources/XMeshSkinned/OBJ/House2/house2.X
P 20.943321 47.554253
S 1.000000 1.000000 1.000000
1 1.000000 0.000000 0.000000
2 0.000000 1.000000 0.000000
3 0.000000 0.000000 1.000000
B
K ./Resources/XMeshSkinned/OBJ/House2/house2.X
P 34.128651 47.688957
S 1.000000 1.000000 1.000000
1 1.000000 0.000000 0.000000
2 0.000000 1.000000 0.000000
3 0.000000 0.000000 1.000000
```

*저장된 vertex height 값

```
35.781185 35.124561 34.421162 34.730396 35.532619
34.727371 34.452984 35.159382 35.318264 37.270332
30.028975 30.851500 27.986740 26.132549 26.013268
15.706925 13.993848 12.955044 10.195168 9.269377 7
0.000000 0.000000 0.000000 0.000000 0.569772 0.609
0.000000 0.000000 0.000000 0.000000 0.000000 0.000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000
0.000000 0.000000 0.000000 0.000000 0.000000 0.000
081 18.284294 20.052149 22.384903 24.168602 25.295
31.922625 0.000000 36.248779 38.399490 43.451870 4
28.568695 27.750235 30.186554 27.791903 28.452208
24.683857 25.822800 25.959867 25.998013 25.202719
```

Thank you!

감사합니다!