COLLEGE OF ART & TECHNOLOGY

Computer Graphics
- Final Project-

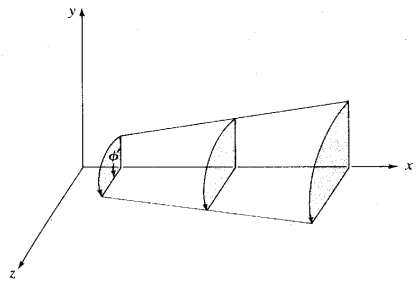
Computer Art & Technology
Sanghyun Seo
sanghyun@cau.ac.kr

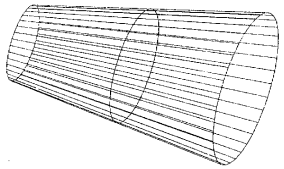
GRLab

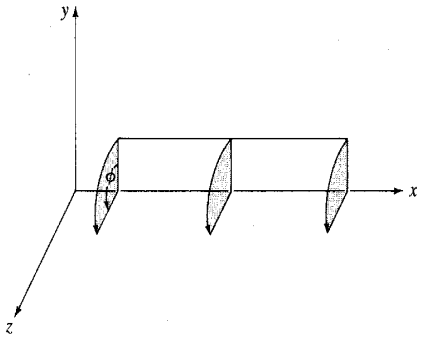
- Topic : 만든 모델링 데이터를 배치한 공간 탐험 게임
  - 팀 프로젝트 (3인 이하 구성)
  - 1. 객체 모델링
    - SOR Model Technique
    - 파일 저장
    - 모델러 SW사용 가능 (SOR 모델러 개발시 추가점)
  - 2. 미로 만들기 및 지형 만들기
    - 사각형 박스등을 이용해 간단한 미로 구성
  - 3. SOR 모델러로 생성한 데이터 미로에 배치
    - 미로와 SOR 모델링 데이터 통합 배치
  - 4. 가상 공간 탐험
    - 이동을 하며 숨겨진 모델링 데이터 체크

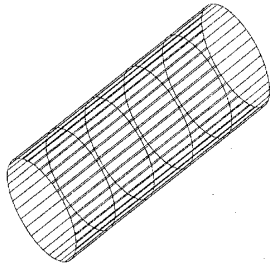
Surface of Revolution (SOR)

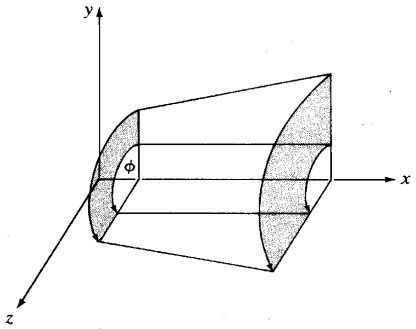
- To generate a 3-D surface, revolve a two dimensional entity, e.g., a line or plane about the axis in space.
  - ⟶called surfaces of revolution (SOR)

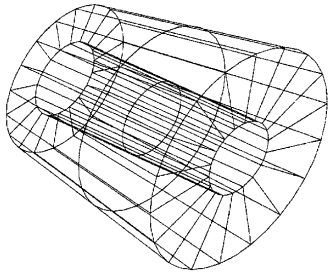




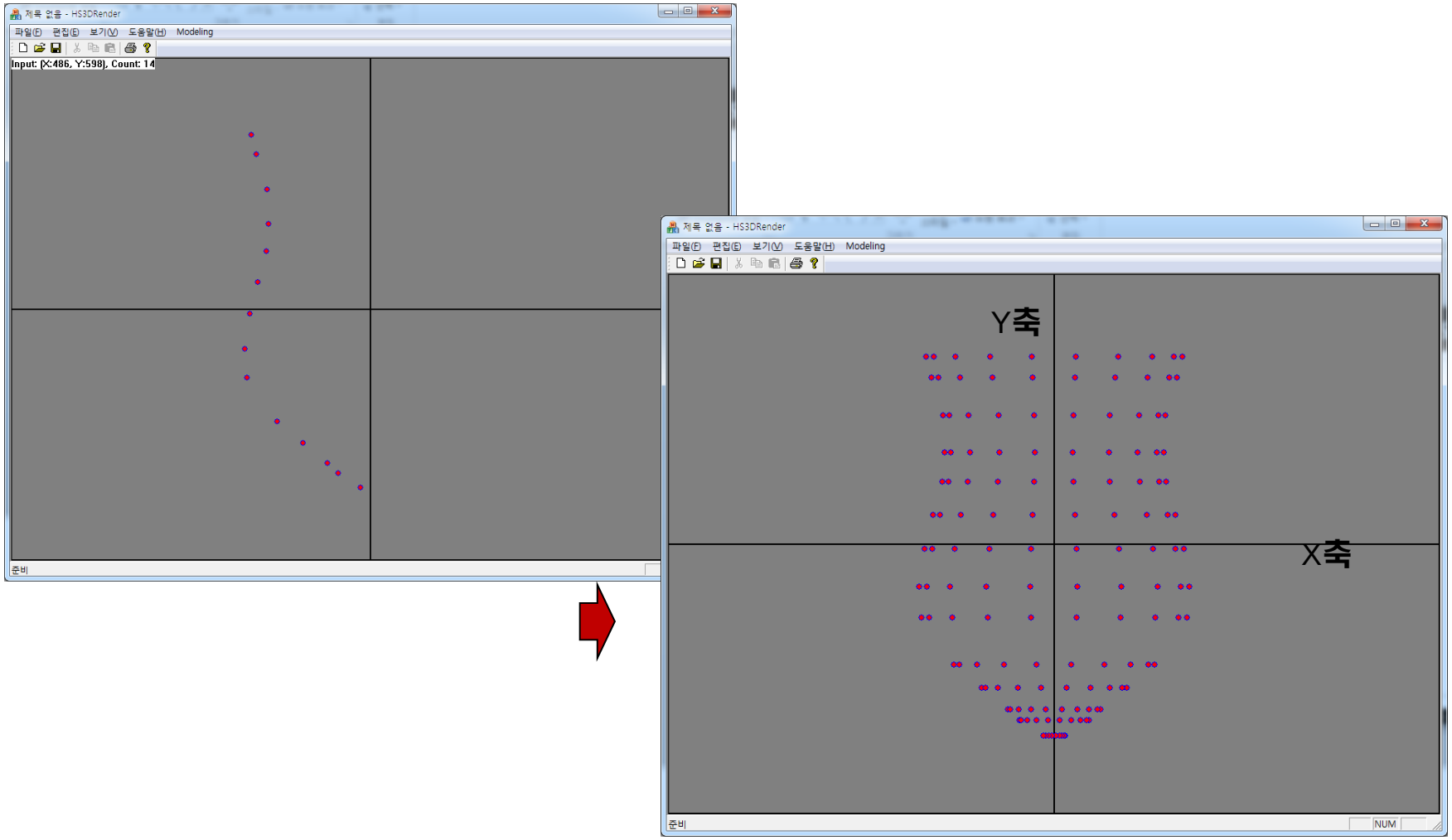








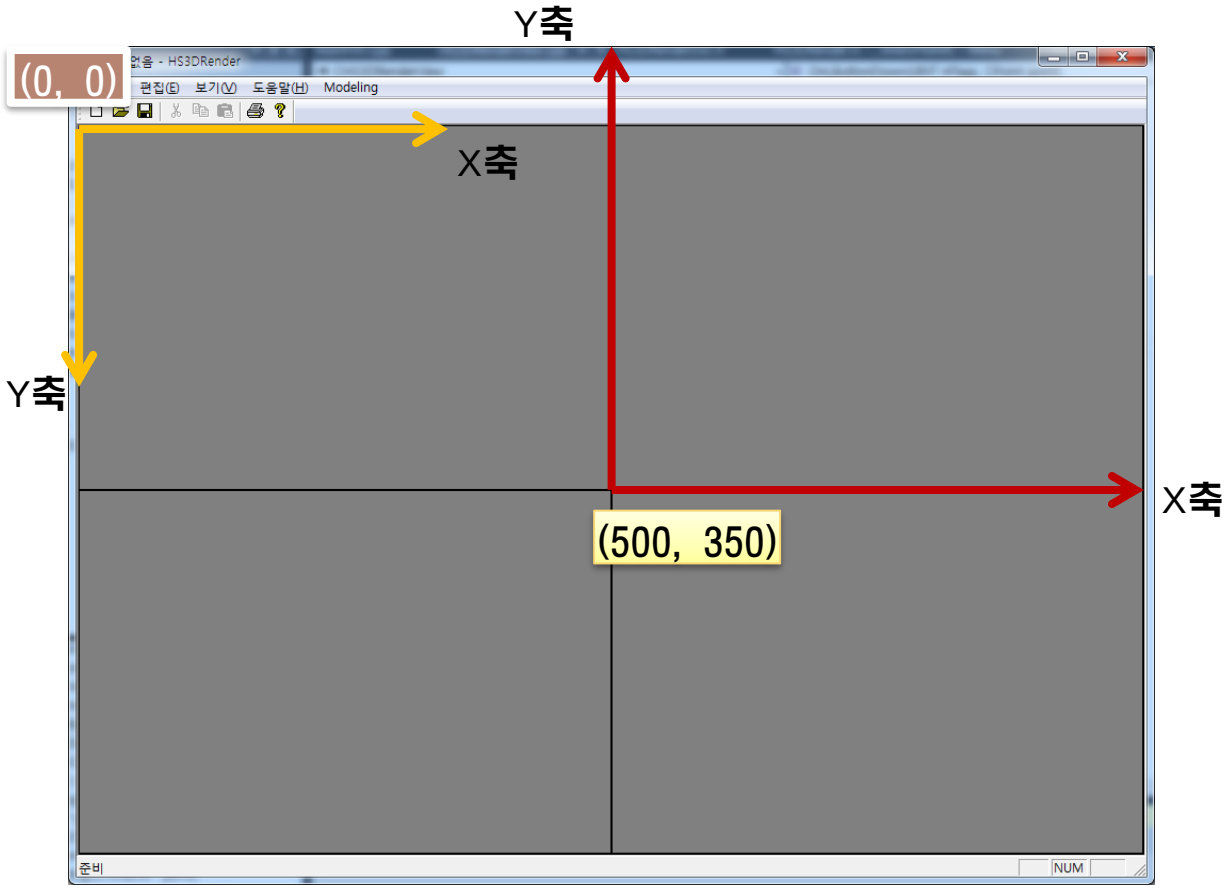
- 1. download in eclass
  - SOR_Modeller.exe



모델러 주요기능

- 주요 기능
  - 1. 입력한 점들을 특정축을 중심으로 회전(일반적: Y축)시키도록 설계
    - 회전시키는 각도를 입력할 수 있어야 한다 (scanf or menu)
      - 360의 약수로 입력하도록 설정하는 것이 좋음
        - Ex) 30도 선택 시 11번 회전시키면 됨
  - 2. 입력각도로 회전한 점들을 저장해야 함
    - 점들을 저장할 수 있는 자료구조가 필요함
    - 이러지는 슬라이드 참조
- 추가 기능 구현 가능
  - 다른 축 회전
  - 다른 모델링 기법도 가능

- Window 좌표계 => 실제 직관적인 좌표계로 수정 (지난 강의록 참조)
  - 윈도우의 크기, 뷰포트의 크기, glOrthro 파라미터에 대한 이해 필요
  - 마우스입력좌표와 World좌표를 일치시켜주는 것이 좋음 (단 y축은 뒤집힘)



- Data Structure
  - Class or Structure로 3D Point(Vertex) 표현하면 편함

```
#include<vector>
class xPoint3D
{
    public:
    float x, y, z, w;
    xPoint3D(){x = y = z = 0; w= 1;};
};
```

```
//사용 예시
std:vector<xPoint3D> arPoints;    // or  xPoint3D arPoints[10000];

xPoint3D pt;
pt.x = 100;        pt.y = 300;        pt.z = 0;
arPoints.push_back(pt);
pt.x = 100;        pt.y = 200;        pt.z = 0;
arPoints.push_back(pt);
```

Rotation Matrix

X axis Rotation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$y' = y \cdot \cos\theta - z \cdot \sin\theta$$
$$z' = y \cdot \sin\theta + z \cdot \cos\theta$$
$$x' = x$$

Y axis Rotation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$z' = z \cdot \cos\theta - x \cdot \sin\theta$$
$$x' = z \cdot \sin\theta + y \cdot \cos\theta$$
$$y' = y$$

Z axis Rotation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$x' = x \cdot \cos\theta - y \cdot \sin\theta$$
$$y' = x \cdot \sin\theta + y \cdot \cos\theta$$
$$z' = z$$

x축 rotation 예시

$$y' = y \cdot \cos\theta - z \cdot \sin\theta$$
$$z' = y \cdot \sin\theta + z \cdot \cos\theta$$
$$x' = x$$

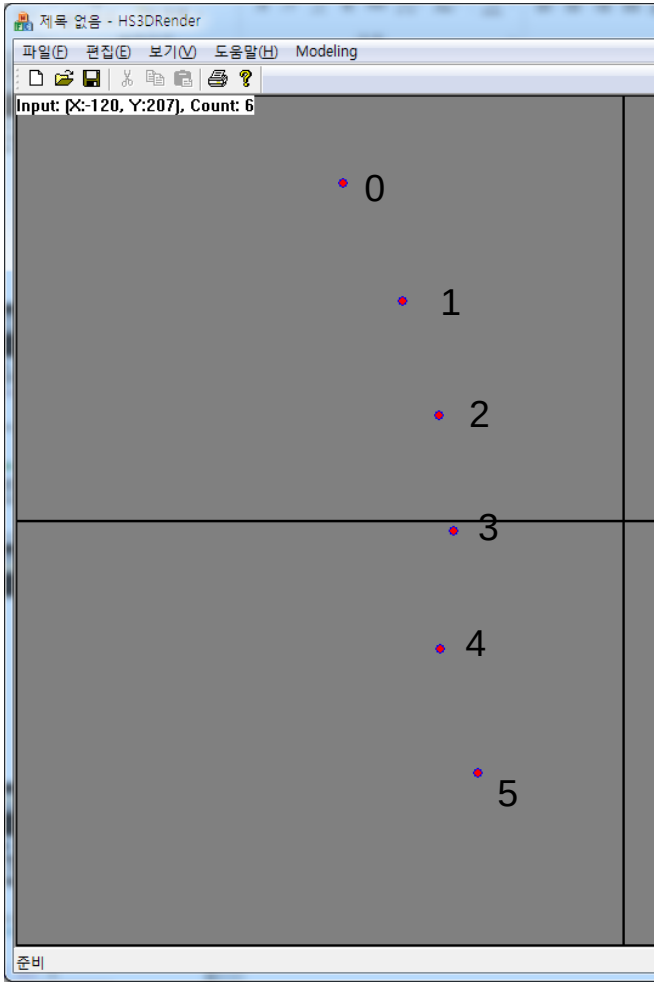
```
float fRotAngle = 30;
std:vector<xPoint3D> arInputPoints;
std:vector<xPoint3D> arRotPoints;

xPoint3D pt;
pt.x = 100;        pt.y = 300;        pt.z = 0;
arPoints.push_back(pt);
pt.x = 200;        pt.y = 200;        pt.z = 0;
arPoints.push_back(pt);

float radian = fRotAngle*(M_PI/180.);

for(int i=0; i<arPoints.size(); i++)
{
      xPoint3D newpt;
      newpt.x = arPoints[i].x;
      newpt.y = arPoints[i].y * cos(radian) – arPoint[i].z * sin(radian);
      newpt.z = arPoints[i].y * sin(radian)  +  arPoint[i].z * cos(radian);
      rotPoints.push_back(newpt);
}
```

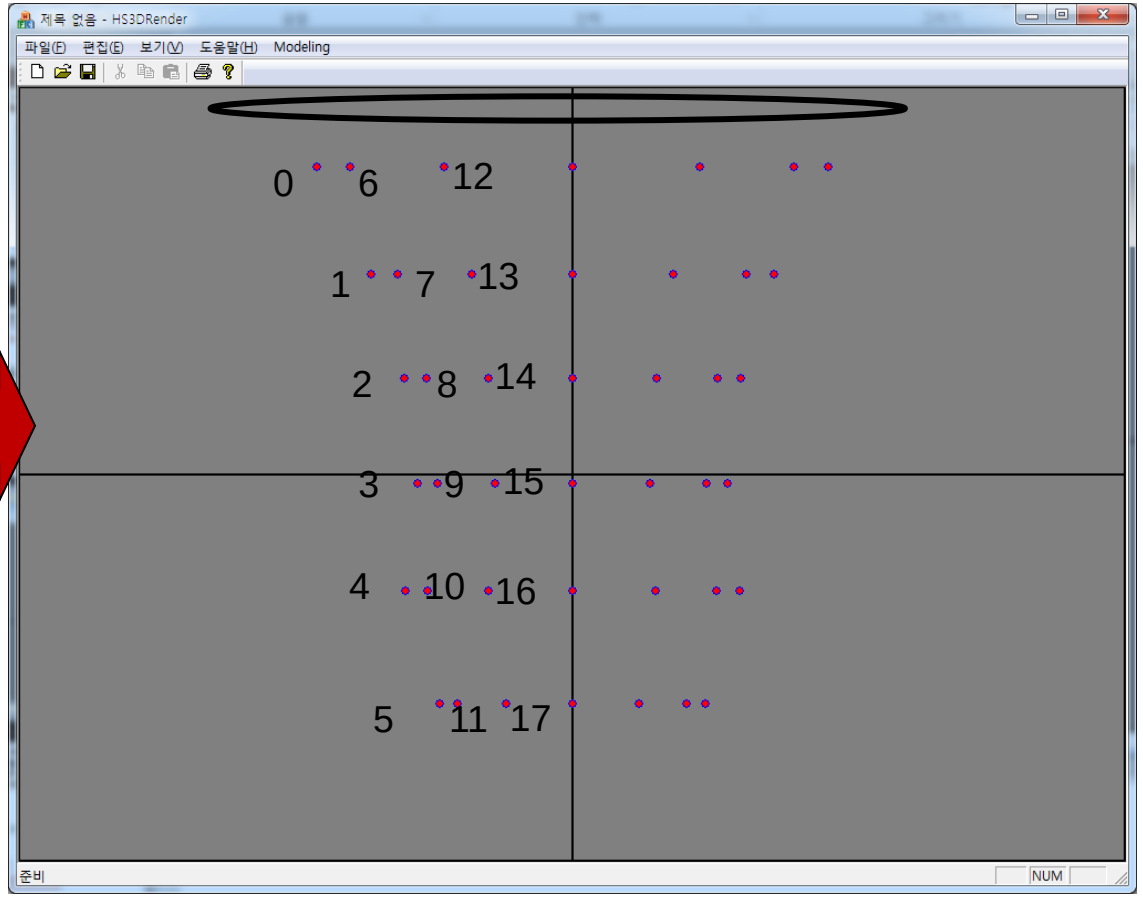
SOR 모델러 데이터 저장

- SOR 모델링 데이터 저장 함수 (지난 실습을 활용해서 저장할 경우)

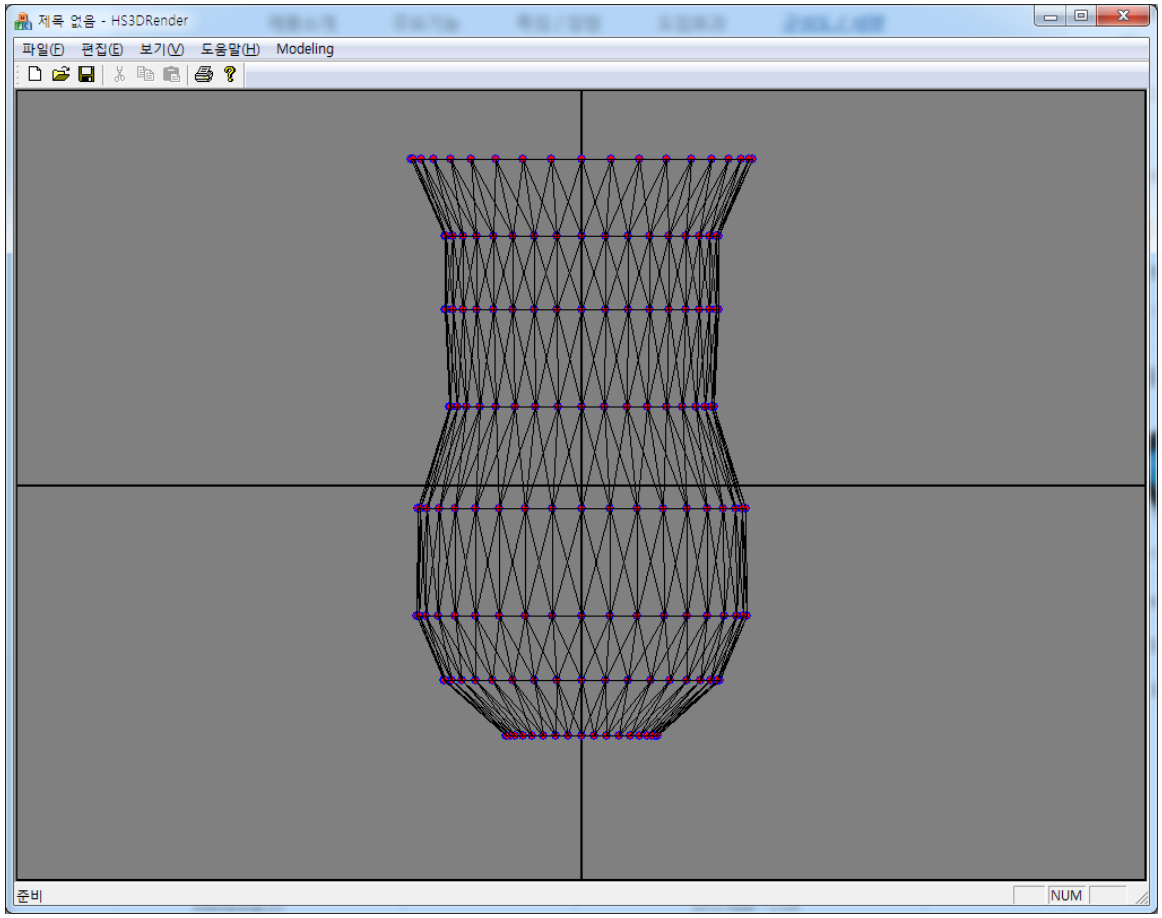
```
void SaveModel()
{
	FILE* fout;

	fout = fopen("c:\\data\\myModel.dat", "w");


	fprintf(fout, "VERTEX =  %d\n", pnum);
	for (int i = 0; i < pnum; i++)
	{
	fprintf(fout, "%.1f %.1f %.1f\n", mpoint[i].x, mpoint[i].y, mpoint[i].z);
	}

	fprintf(fout, "FACE = %d\n", fnum);
	for (int i = 0; i < pnum; i++)
	{
	fprintf(fout, "%d %d %d\n", mface[i].ip[0], mface[i].ip[1], mface[i].ip[2]);
	}
	fclose(fout);
}
```

Drawing 힌트

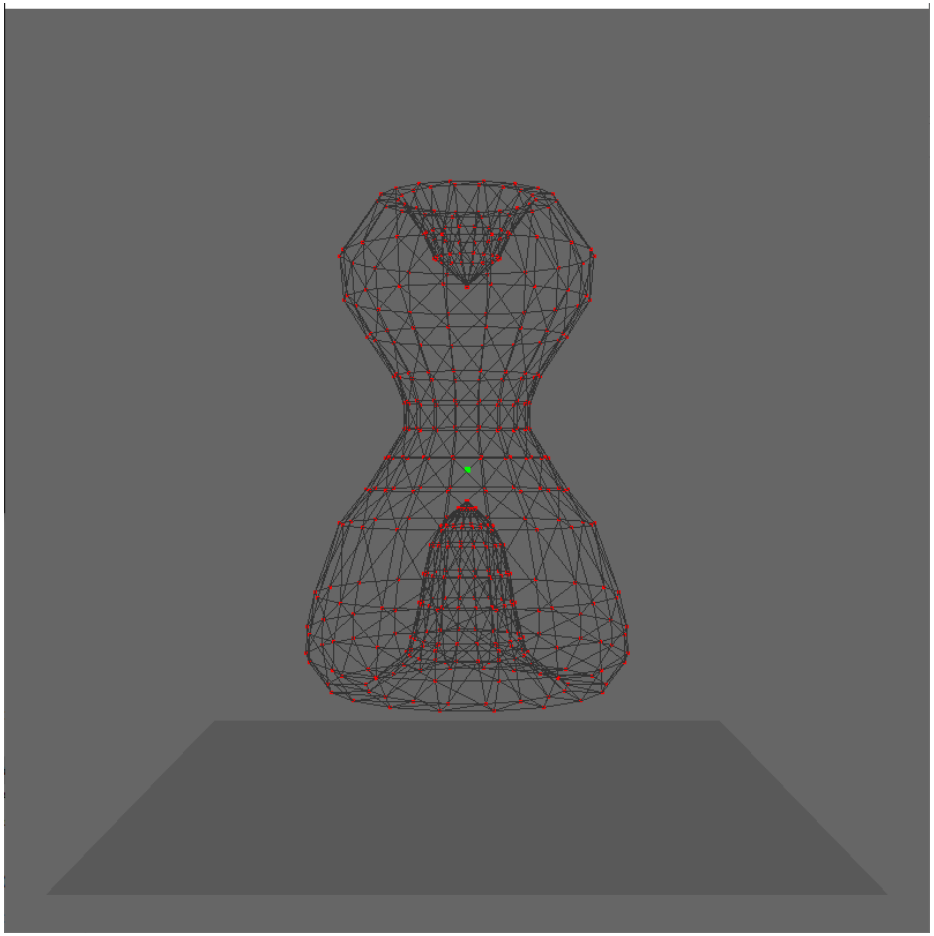

제목 없음 - HS3DRender
파일(F) 편집(E) 보기(V) 도움말(H) Modeling
Input: (X:-120, Y:207), Count: 6
0
1
2
3
4
5
준비

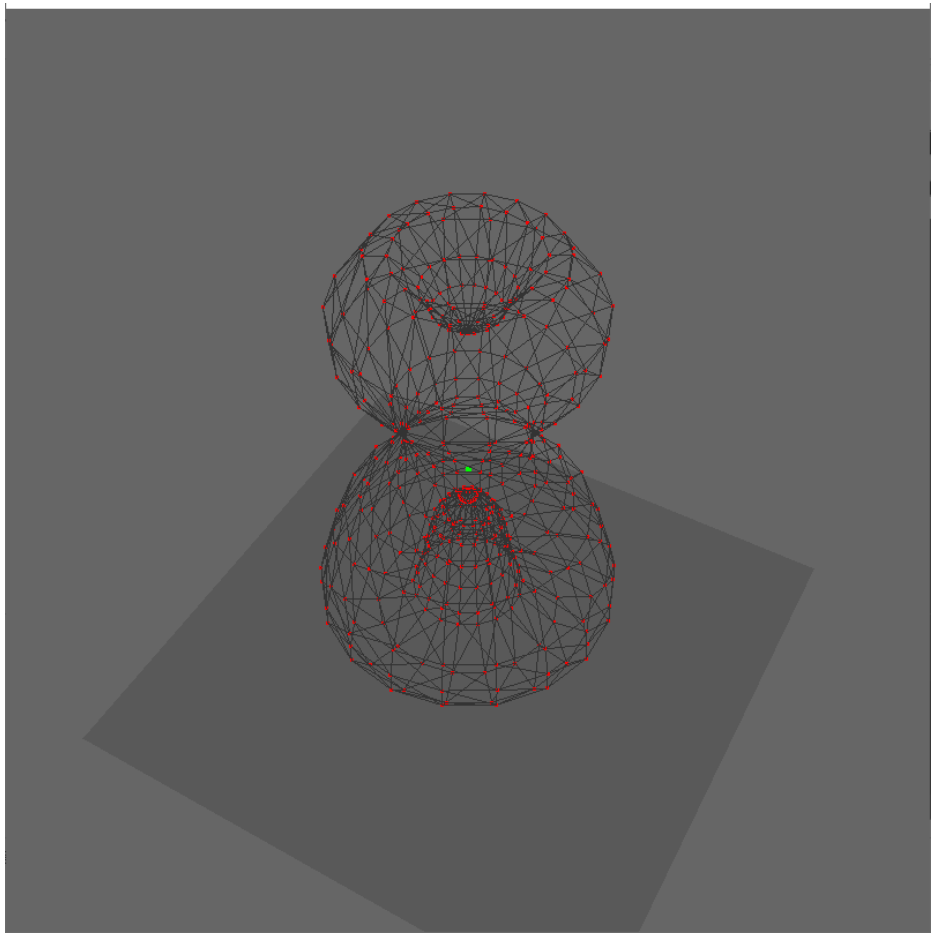

제목 없음 - HS3DRender
파일(F) 편집(E) 보기(V) 도움말(H) Modeling
0 6 12
1 7 13
2 8 14
3 9 15
4 10 16
5 11 17
준비
NUM

- 삼각형 mesh 형태로 표현
  - 회전하여 만들어진 점들을 GL_LINE /GL_TRIANGLE/GL_POLYGON 으로 표현하면 됨

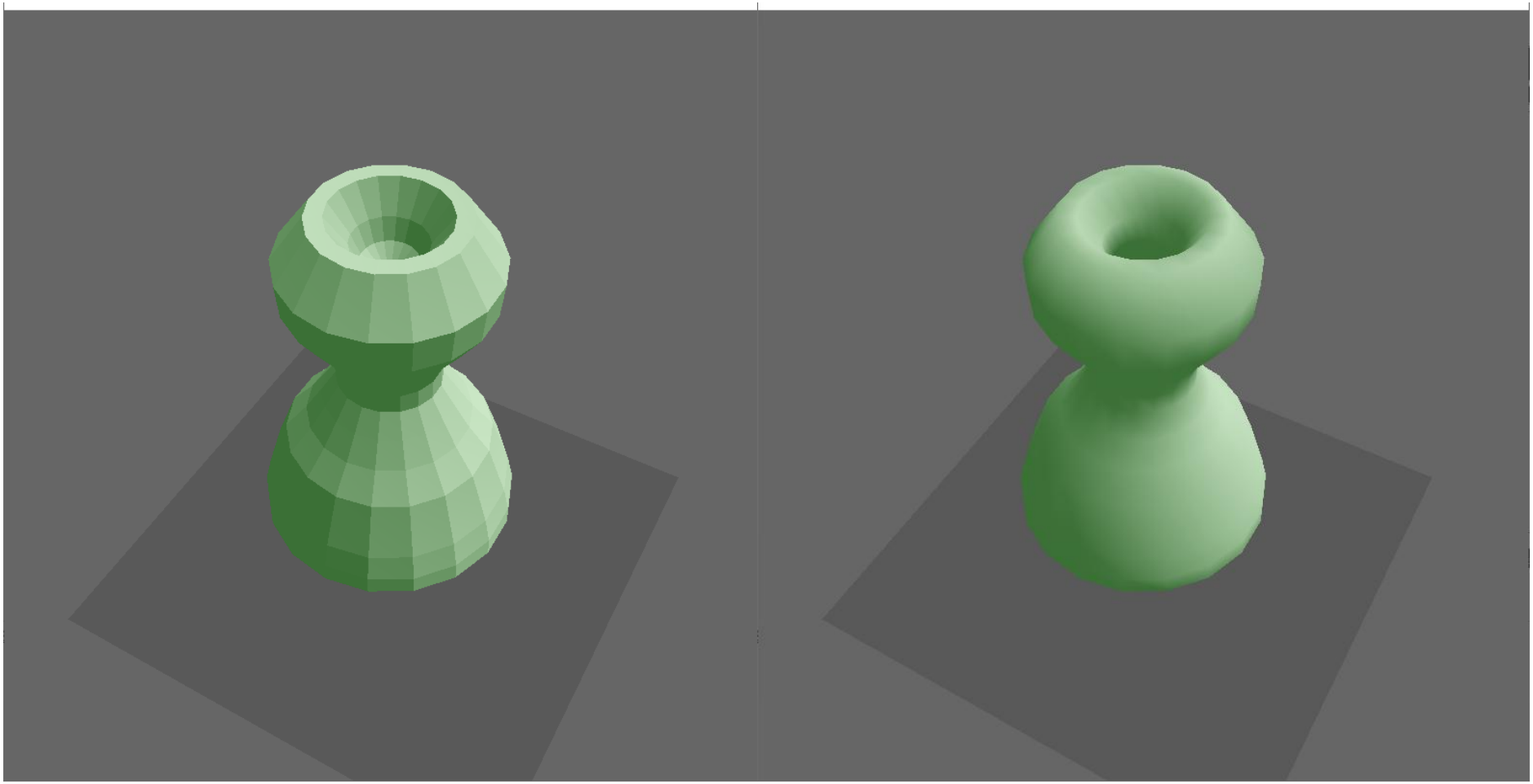


Modeler Preview



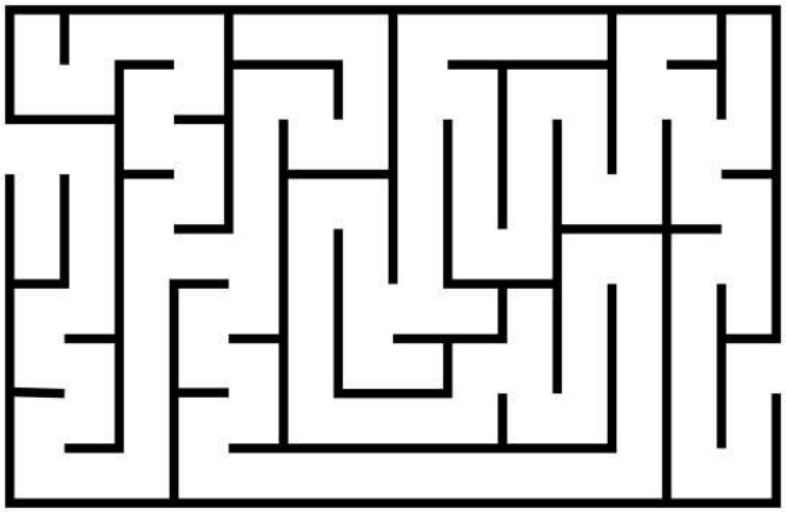


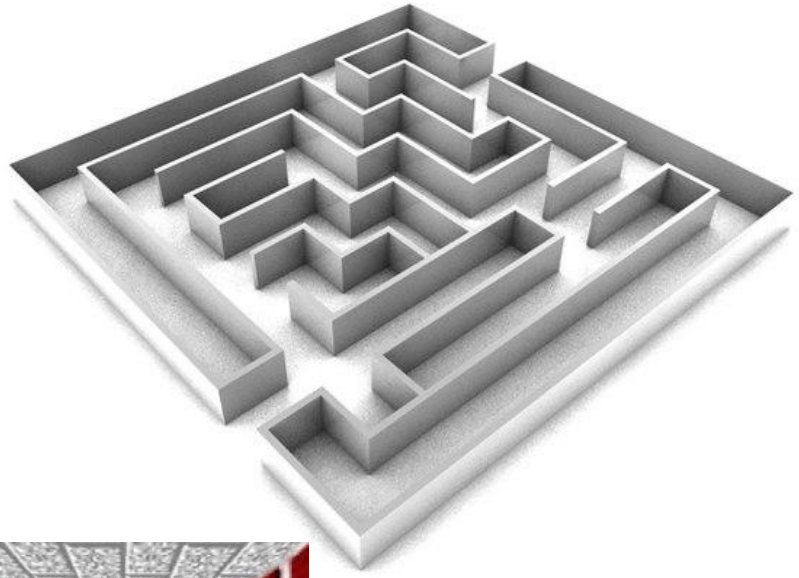
Modeler Preview



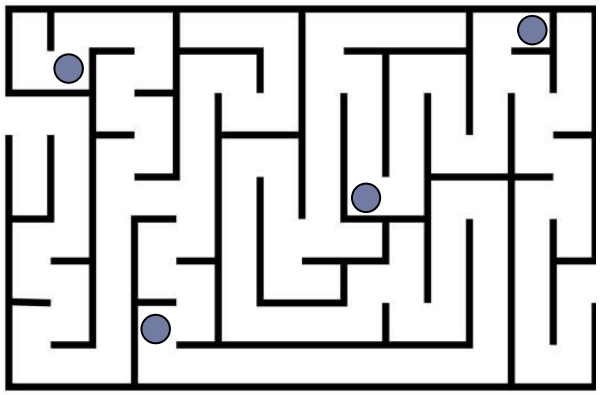
3D 미로 만들기

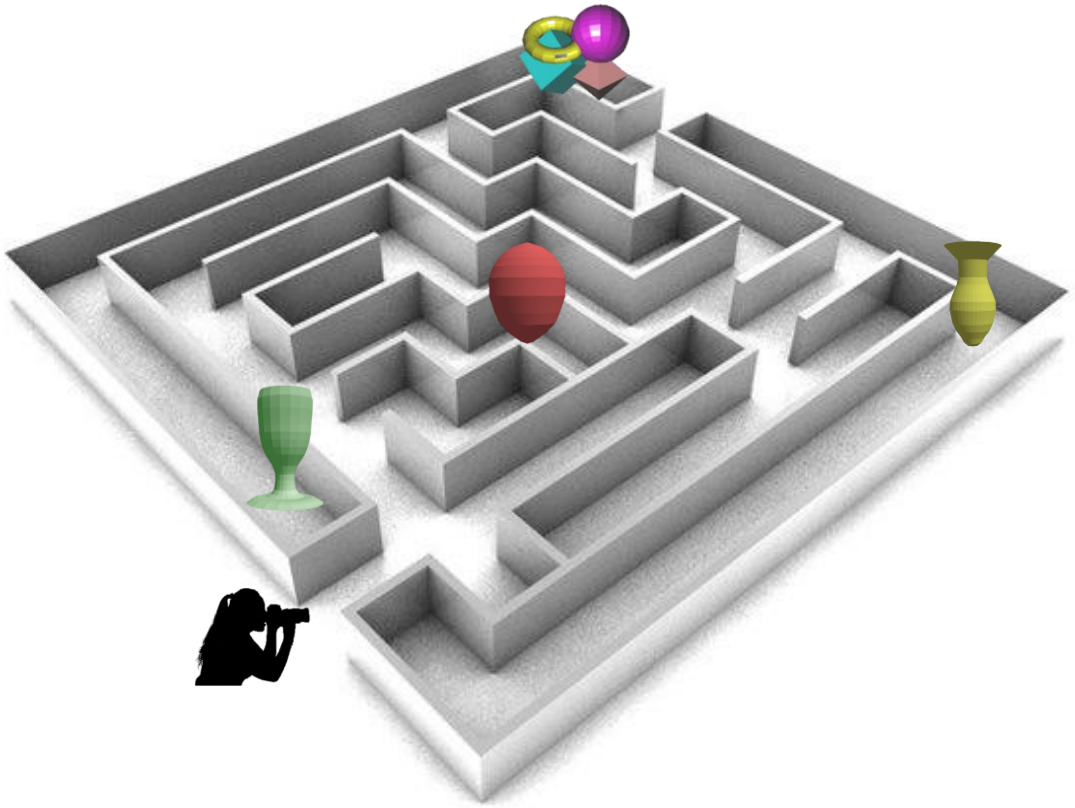
- SOR 모델링 데이터를 배치할 미로 만들기
  - 저장된 객체와 OpenGL의 기본 도형을 이용해 가상공간을 구성
  - 미로 모델러를 따로 만들어도 됨
  - 순수 데이터를 수작업 가능
  - 3D Tool을 이용해 제작 가능



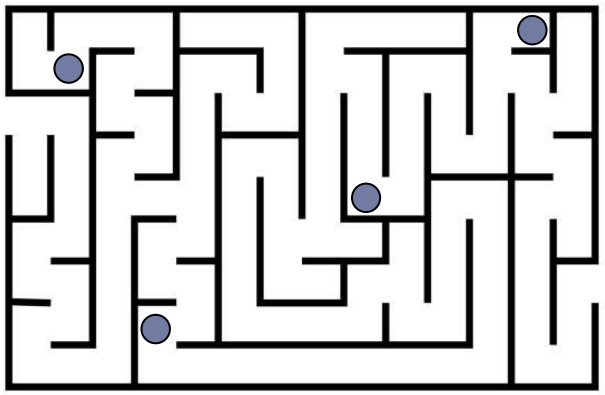


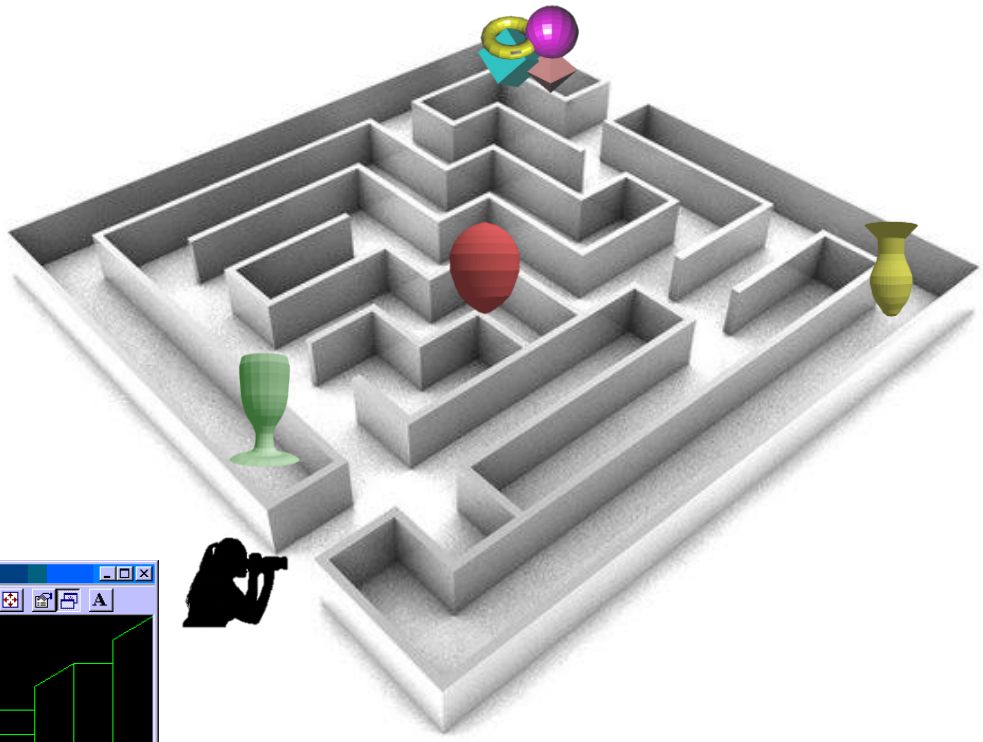
- SOR 모델링 데이터를 이용한 가상공간 렌더링/애니메이션 및 Navigation

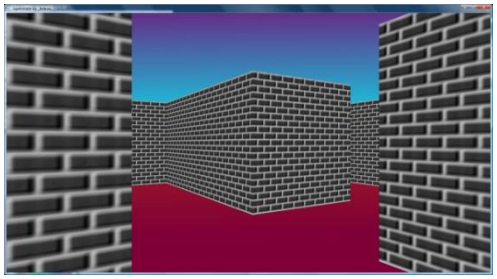


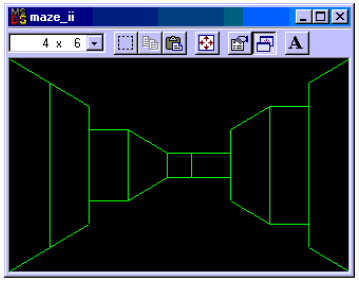


- SOR 모델링 데이터를 이용한 가상공간 렌더링/애니메이션 및 Navigation
  - 자신이 모델링한 데이터를 여러 개 배치하여 공간구성
  - 키보드 화살표 또는 이동키를 이용한 (좌우회전, 앞뒤이동 등)공간 이동



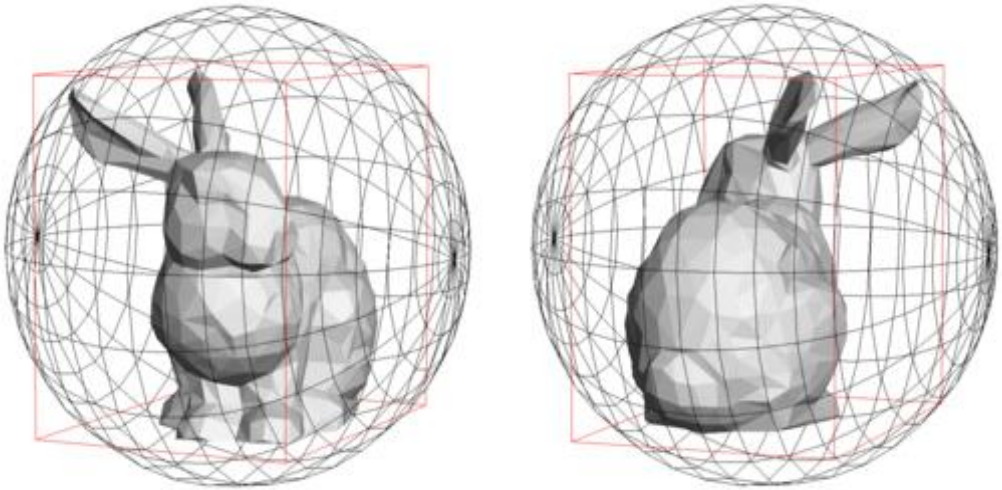






주요 기능(가산점)

- (추요 기능) 추가 구현 가능 : 자유롭게 새로운 기능을 추가하시오
  - 공간탐색 이동 시 객체와 벽과의 충돌 체크
  - 바닥과 벽에 텍스쳐 매핑 구현
  - 바닥에 그림자 넣기 구현
  - 높낮이가 다른 지형 넣어 다양한 액션 표현
  - 가상공간에 움직이는 Object 구현 등등
  - 기타 여러분의 아이디어를 이용한 추가 구현시 큰 가산점 부여



<충돌체크 : Bounding Box or Sphere>

일정 및 시연

- **최종 프로젝트 제출 기한: 2025년 12월 10일 자정**
  - 제출물
    - 1. 보고서 : Report.pdf
      - PDF로 저장하여 제출해야 함
      - 표지와 목차제외 10장이내
      - 생성형 AI/Open Source 이용시 활용내용 기입
    - 2. 첨부파일: Program.zip
      - 압축 후 실행파일을 클릭했을 경우 바로 실행되는 지 테스트 후 제출
        - 모델링 파일을 반드시 포함시킬 것 (실행파일 폴더 Bin Folder)
        - 프로젝트 전체(Debug폴더, Release 폴더 등)를 포함시키면 감점
      - 폴더구조
        - Source – 구현 원 SOURCE(cpp, h)
        - Bin : 실행파일(exe) + 모델링 데이터(dat, 확장자 변경가능)
- **최종 프로그램 시연: 2025년 12월 11일 수업시간(예정)**
  - 추후 팀내 교차평가 시행 예정

최종 팀구성

- 팀 구성 (3인 이내)