|  |  |  |
| --- | --- | --- |
| 이름 | 백성빈 | 손주은 |
| 학번 | 20170680 | 20170702 |
| 학과 | 컴퓨터공학과 | 컴퓨터공학과 |
| ID | sungbin0515 | jeson |

CSED 451 Computer Graphics—Assn3

**Assn 3: 피카츄 배구**

ASSN 1과 2에서 만들었던 2D 피카츄 배구 게임을 3D그래픽으로 구현한다.

# 개발 환경:

**IDE:** Visual Studio 2017 community

**Compile mode (VS 2017):** Release x86

**Dependency**: OpenGL / GLEW 2.1.0 / Freeglut 3.0.0

# **프로그램 설계**

1. **Vertex Shader와 Fragment Shader**

이번 과제에서는 앞선 과제들과 달리 Opengl의 내장함수들을 사용하지 않고 Programmable Shader를 통해 물체들을 그렸다. Application에서는 vertex의 color information이나 model-view, projection matrix 정보를 모두 곱한 ‘MVP’ matrix등을 shader로 넘겨준다. ‘ShaderLoader’ 를 통해 vertex/fragment shader를 load 할 수 있다.

Vertex Shader에서는 MVP와 vertex Position을 곱한 값을 gl\_position 내장 변수에 저장한다. 이 과정을 통해 object는 viewing coordinate상으로 나타내진다.

Fragment Shader에서는 위와 비슷한 방법으로 application에서 넘겨준 color정보를 바탕으로 vertex의 color를 지정한다.

1. **GameManager와 RenderManager**
2. **Loading Obj File**
3. **Camera Setting**
4. **Hidden Line Removal**

키보드의 ‘h’키를 입력함으로써 hidden line removal mode와 wireframe mode 사이에서 전환할 수 있게 설계했다.

1. **충돌 판정 및 점수 계산**

두 Object 사이의 물리적 거리를 반환하는 함수 ‘distance’를 이용해 충돌을 판정했다. 기준을 정해놓고 두 물체 사이의 거리가 기준치보다 가까울 때 충돌이 일어났다고 판정했고 이에 따라 공의 speed를 바꾸어 탄성 충돌을 표현했다.

Ball의 z좌표가 boundary로 정해놓은 지점을 넘을 때, 점수를 update하게 했다. (캐릭터가 공을 막지 못할 때)

# **시연**

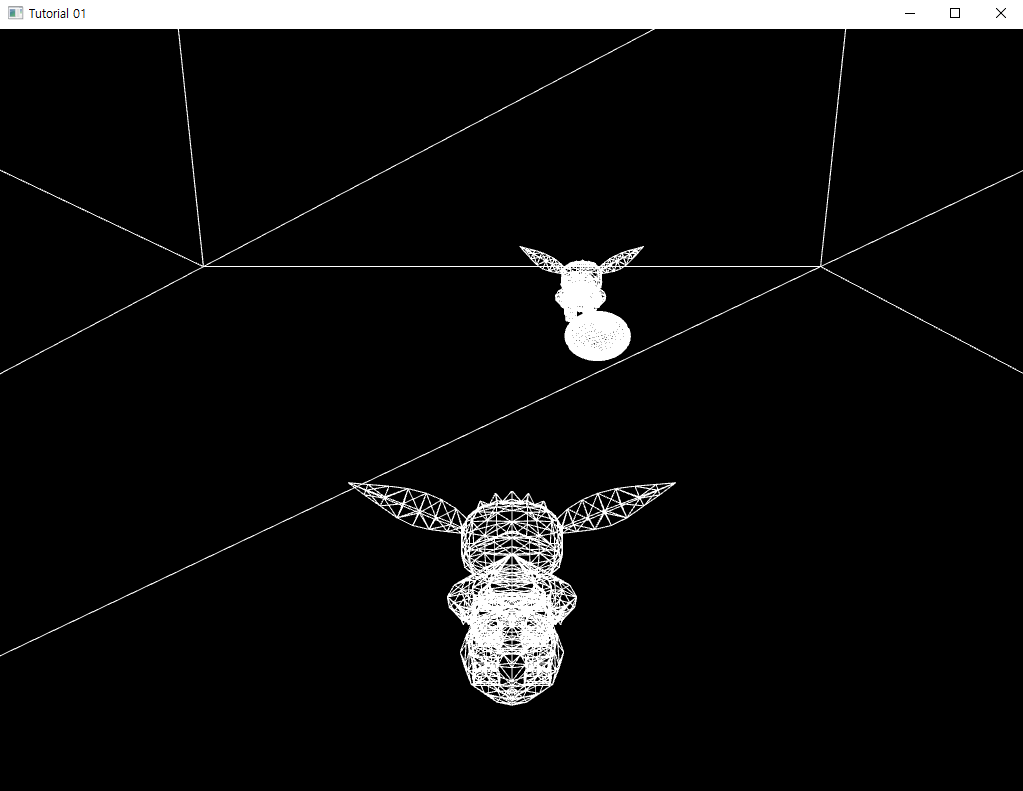


Figure 1 게임 화면이다. 공과 두 플레이어가 있다.

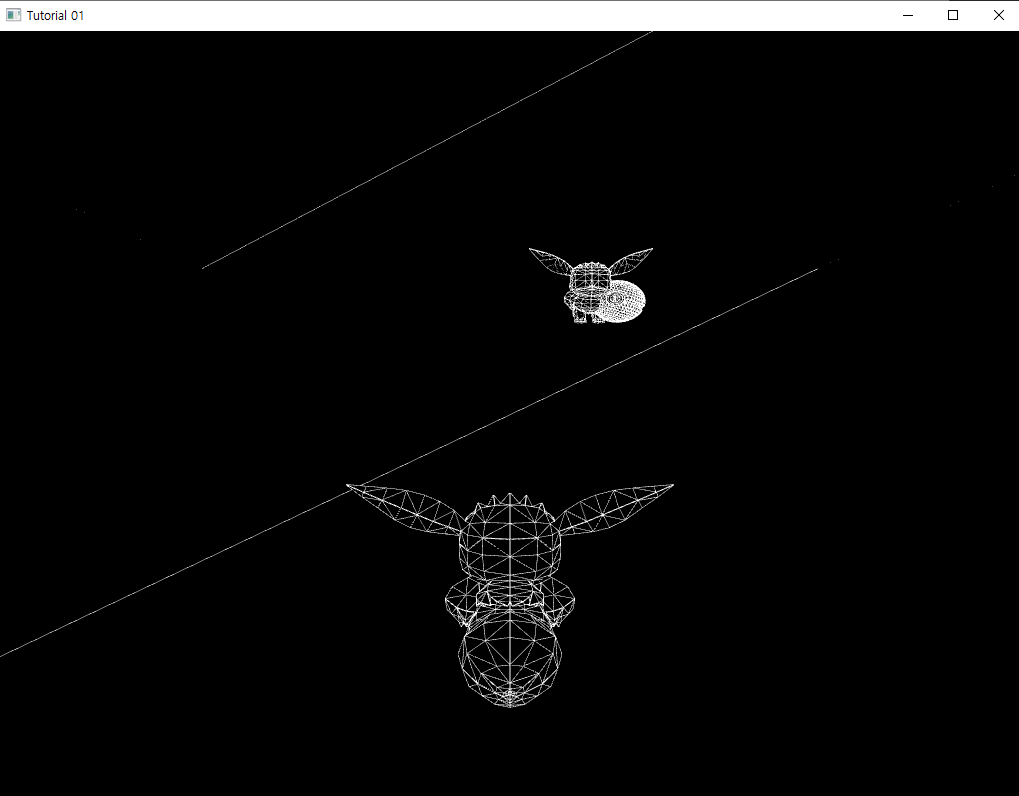


Figure 2 ‘h’키를 눌러 hidden line removal을 적용한 상태이다

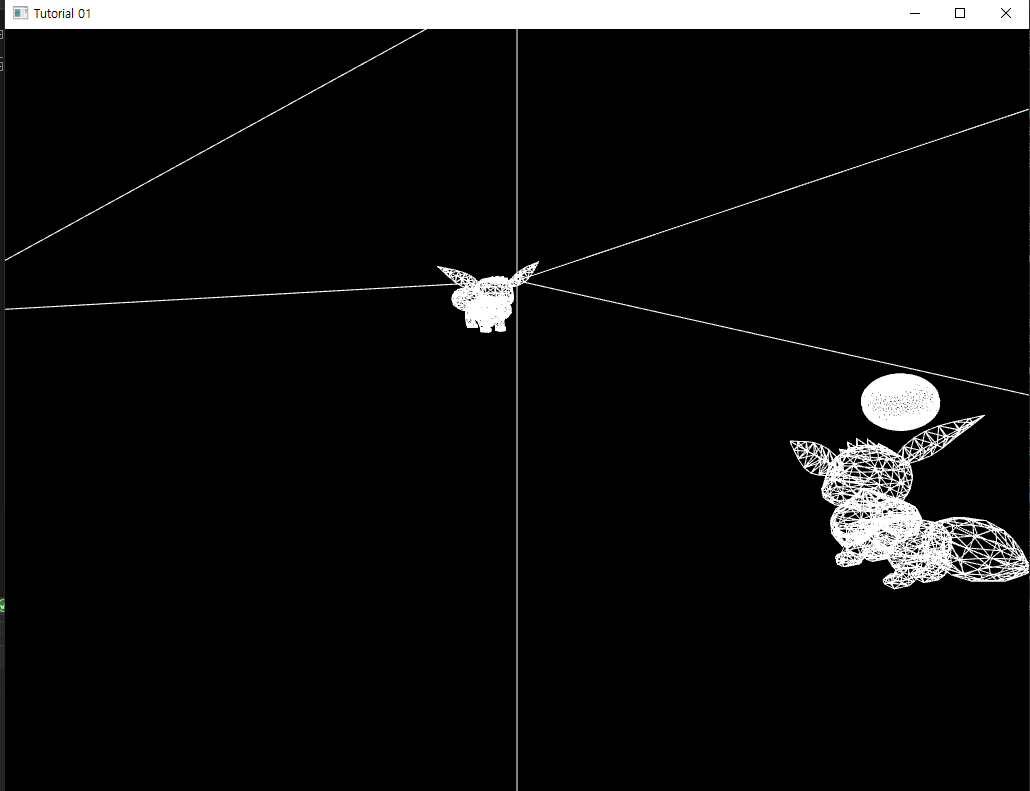


Figure 3 ‘d’키를 눌러 처음과 다른 카메라 모드에서 본 모습이다.

# **토론**

ㅇ

# **결론**

이번 과제에서는 Graphics Pipeline의 Vertex Shader와 Fragment Shader를 직접 program해 Game 화면을 그려보았다. 이전과는 달리 transformation matrix를 직접 관리함으로써 여러 카메라 모드를 구현할 수 있었다. 또한 wireframe mode와 hidden line removal을 적용한 mode를 구현해보았다.

# **개선 방향**

이전에 OpenGL의 내장 함수를 사용해서 그렸을 때는 `glutBitmapCharacters’와 같은 함수를 이용해 비교적 쉽게 점수를 화면에 출력할 수 있었는데 내장 함수를 사용하지 않고 구현하는 것이 꽤 어려웠다. 다음 과제 때 Texture Mapping을 배우고 하면 비교적 쉽게 구현할 수 있을 것이다.

# **참고자료**

<http://www.opengl-tutorial.org/kr/>