CSED451 Assignment #3-2

3D DRAWING REPORT

Team: GPS

CSE 20202405 문민재 csmmj4594

CSE 20202728 김진수 fusion4268

**Overview of Program**

본 프로그램은

**Programming Environment**

Visual Studio 2019

OpenGL – 4.6

GLSL – 4.60

Freeglut-MSVC-3.0.0

Glew-2.1.0

Glm-0.9.9.7

**Design and Implementation**

1. 자료구조 설계
2. **~~Color (colors.h)~~**

게임 상에서 사용할 색상 코드와 팔레트 변경을 담당한다.

1. Object (object.h)

맵 상의 오브젝트를 정의하는 클래스이다.  
멤버 변수로 type, color, x, y, z를 가지고 있다. 이들은 각각 오브젝트 타입(0일 경우 wall, 1일 경우 character), 오브젝트의 색깔, 오브젝트를 그릴 때 필요한 x, y, z좌표에 해당한다.

1. Wall: public object (Wall.h)

Object를 상속받아 맵 상의 벽을 정의하는 클래스이다.  
Wall 오브젝트들은 공통적으로 type이 0이며, x, y, z는 벽의 위치를 의미한다. 추가적으로 width, height, depth라는 멤버 변수를 가지는데, 이는 각각 직사각형의 x, y, z축 방향 길이를 의미한다.

1. Character: public object (character.h)

Object를 상속받아 맵 상의 캐릭터 오브젝트(플레이어, 도둑)를 정의하는 클래스이다.  
character 오브젝트들은 type이 1에 해당하며, x, y, z는 torso의 중심 좌표를 나타낸다. 그리고 원의 반지름을 나타내기 위한 변수인 rad와 벽과의 충돌 여부를 나타내는 변수인 isCollided가 있다. 그 외에 인간형 캐릭터 구현에 필요한 여러 멤버변수들을 가지고 있다.

1. treeNode (treeNode.h)

인간형 캐릭터의 hierarchy를 표현하기 위해 필요한 Left Child Right Sibling(LCRS) tree를 구현하는 클래스이다.

현재 modelView matrix의 오른쪽에 곱해질 기본적으로 matrix인 mtx,  
기본 mtx transform에 추가적으로 행해질 transform을 나타내는 additionalTransform,  
이 node에 해당하는 신체 부위를 그려주기 위한 함수 포인터인 draw,  
그리고 sibling과 child 정보를 가지고 있다.

1. pose (pose.h)

플레이어와 도둑의 포즈를 정의한 클래스이다.  
캐릭터의 포즈에 따른 변화는 상체에서만 이루어지므로, 왼팔과 오른팔 각각 2가지씩 총 4가지 각도와 포즈에 따른 색을 변수로 갖고있다.

1. camera(camera.h)

이번 과제에서는 3가지 시점을 구현했는데, 각 시점에 해당하는 카메라 정보(camera position, reference point, up vector))을 저장하는 클래스이다.

1. 기본 기능 구현(이전 ASSN 관련)
2. 벽의 생성 및 이동

Assn3\_1과 동일하게 구현되었다.

1. 충돌 판정

Assn3\_1과 동일하게 구현되었다.

1. 플레이어 캐릭터의 조작

Assn3\_1과 동일하게 구현되었다.

1. 치트 기능

Assn3\_1과 동일하게 구현되었다.

1. Pass/Fail 시의 action

Assn3\_1과 동일하게 구현되었다.

1. Jump 구현

Assn3\_1과 동일하게 구현되었다.

1. 3D 시점 구현

Assn3\_1과 동일하게 구현되었다.

1. 시점 변환 및 줌 인/아웃

Assn3\_1과 동일하게 구현되었다.

1. Shader 적용
2. Vertex Shader
3. Fragment Shader
4. Vertex Array를 이용한 오브젝트 구현
   1. 캐릭터
   2. 벽

**How to Run**

프로젝트 폴더의 \bin\x64 폴더로 이동하여 graphics\_assn3\_2.exe를 실행시킨다.

캐릭터의 포즈 변경은 방향키로 입력하며, 위, 아래, 왼쪽, 오른쪽 방향키는 각각 플레이어 캐릭터를 적색, 녹색, 청색, 황색으로 변경시키고, 플레이어 캐릭터는 각 색에 대응하는 포즈를 취한다. 스페이스바를 입력하여 캐릭터를 점프시킬 수 있다.

숫자키 1번을 눌러 1인칭 시점으로, 3을 눌러 3인칭 시점으로, 9를 눌러 XY 평면을 바라보는 시점으로 시점을 전환시킬 수 있다.

**Example**

기존 Assn2와 중복되는 부분은 제외하였다.

프로그램 실행

**Discussion**

1. Shader 적용
2. Vertex Shader

이번

1. Fragment Shader
2. Vertex Array를 이용한 object 구현
3. 캐릭터

이전

1. 벽

**Conclusion**

1. Shader 적용

이번

1. Vertex Array를 이용한 object 구현

**Direction of Improvement**

현재

**참고문헌**

<https://alleysark.tistory.com/260>: 이번 과제의 구현에서, 파일로 된 vertex/fragment shader를 string으로 읽어오는 코드(readShaderSource 함수)와,  
program에 vertex/fragment shader를 attach 완료한 후 program을 점검하여 문제가 있을 경우 로그를 띄워주는 CheckProgram 함수를 참고하였다.

<https://heinleinsgame.tistory.com/7>: GLFW에서 shader를 적용시켜 삼각형 하나를 출력하는 기본적인 코드로, 이번 과제에서 shader를 사용하기 위한 setup(VAO, VBO, program setting 등)을 참고하였다.

<http://www.songho.ca/opengl/gl_cylinder.html>

<http://www.songho.ca/opengl/gl_sphere.html>

Cylinder와 Sphere를 구현하는 부분의 위의 링크를 참고하여 구현하였다. 위의 링크에서는 현재 프로그램에서 구현되어 있는 vertices와 indices 뿐만 아닌 normal vectices와 texture까지 구현한다. 과제 프로그램은 이를 간소화하여 vertices와 indices를 구하는 부분만 프로그램에 추가하였다.