Yeungnam University

1. Conceptualization

21911722 이현동
st.linsio@gmail.com

Mini Ray-Tracer

Yeungnam University

[Revision history]

Revision date	Version #	Description	Author
2024/03/24	1.0	포괄 개요 작성	이현동

= Contents =

1. Business purpose

1) Project Background

수업 시간에 무언가를 배우고, 배운 내용을 과제를 통해 구현할 때 지금까지는 대부분 특정한 문제 상황이 주어지고, 해당 상황을 해결하는 특정한 알고리즘, 함수를 구현하거나 간단한 동작을 하는, input-output이 정해진 상황에서의 프로그램을 만드는 것이 대부분이였다. 또한, 외부 소스나 라이브러리(혹은 API)의 참조 없이 순수히 구현하는 경우가 대부분이였다.

실제로 만들어지는 프로그램들을 본다면 하나의 목표를 달성하기 위해 매우 복잡한 연산과정이 이루어지며 내용물 또한 한명이 전체를 만드는 것도 아니며 여러 팀원이서 파트를 담당하며 만드는 것이 보통이다.

따라서 외부 라이브러리를 활용하여 프로그램을 만드는 방법을 익히는 것도 좋은 경험이 될 것이라 생각하여 이번 과제에서는 minilibx라는 외부 그래픽 라이브러리를 활용하여 phong 조명 모델을 기반으로 한 Ray-Tracing을 구현해 보고자 한다.

2) Motivation

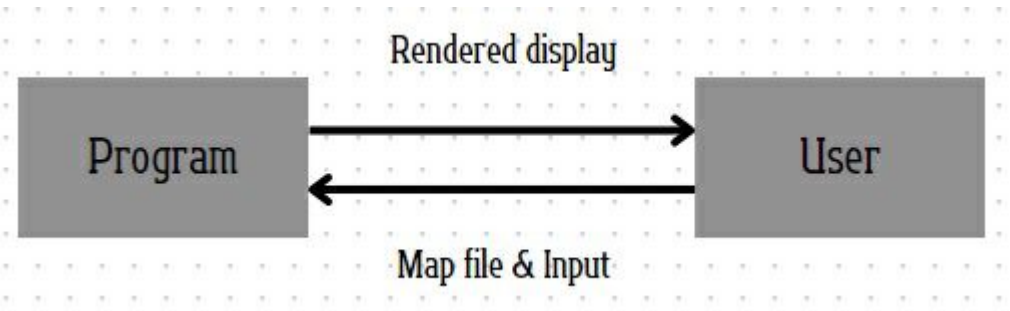
컴퓨터 그래픽스 시간에 배운 내용을 토대로 공부하던 중, 3차원을 2차원에 투영하는 방법에 대해 관심이 생기게 되었고, 컴퓨터로 생성된 3차원 이미지를 렌더링 할 때는 두 가지의 접근 방식이 있음을 알게되었다. 바로 Rasterization과 Ray Tracing이다. 전자의 경우는 효율성때문에 거의 모든 그래픽 엔진에서 사용한다고 하며 후자의 경우는 더 높은 수준의 시각적 사실감을 만들어 낸다고 한다. 그 이유는 더 높은 계산비용 때문이다. 하지만 수업시간에는 raster 방식만을 다루는 것 같아 ray trace에 관해 스스로 더 공부해보고 싶어 해당 내용을 구현하는 것을 과제로 정하기로 했다.

3) Goal

Minimal
- minilibx를 참고하여 화면에 3D 그래픽 장면을 투영하기
- Phong lightning 모델을 적용하기
- 3개 정도의 도형을 구현하기

Optional
- 키보드 및 마우스 입력을 통해 화면 종료, 이동 기능 구현하기
- 반사 구현
- Bump 매핑 구현하기

2. System context diagram



3. Use case list

1) View

Actor	Program
Description	현재 시점의 Camera에서 연산한 View를 화면에 출력함

2) Mod control

Actor	Program
Description	유저의 입력에 따라 조명 모델의 유무, 해상도의 크기를 조절하여 현재 화면을 재연산함

3) File control

Actor	Program, User
Description	제공된 파일을 토대로 Map을 만들거나 User의 영향으로 Map이 수정된 경우 저장함

4) I/O control

Actor	Program, User
Description	User의 Input에 맞게 지정된 행동을 함 (카메라 이동, 파일 저장, 오브젝트 추가, 모드 변경 등)

5) Camera control

Actor	Program, User
Description	카메라의 이동 및 회전을 처리함

6) Object control

Actor	Program, User
Description	생성 가능한 물체를 현재 Map에 추가하거나 제거 가능한 물체를 현재 Map에서 제거함

7) Light control

Actor	Program
Description	조명에 따른 색상 및 반사 연산을 처리하여 최종 연산 결과의 색상을 결정함

4. Concept of operation

1) View

Purpose	최종 연산 결과를 화면에 출력
Approach	초기 맵 로딩 혹은 각 입력에 의해 화면의 변경점이 발생했을 시, 해당 변경점을 토대로 재연산한 결과를 모니터에 출력한다
Dynamics	어떠한 입력이 주어졌을 시
Goals	성공적으로 현재 화면을 출력한다

2) Mod control

Purpose	그래픽 품질 조절 옵션의 대체
Approach	대략적인 구조를 보여주는 연산과 모든 조명 및 반사를 적용한 연산 중 한가지 연산 방식을 선택한다
Dynamics	모드 변경 입력이 주어졌을 시
Goals	각 조명 및 반사, 그림자 처리에 대해 On/Off 기능을 제공한다

3) File control

Purpose	해당 툴을 이용하기 위한 파일의 처리
Approach	주어진 파일을 통해 현재 화면을 결정하거나, 현재 화면을 언제든지 다시 불러올 수 있도록 파일로 저장함
Dynamics	파일 관련 입력이 주어졌을 시
Goals	파일 읽기 및 저장에 대해 처리한다

4) Camera control

Purpose	카메라 이동을 올바르게 처리
Approach	3차원의 카메라의 경우 상하좌우의 이동뿐만 아니라 앞과 뒤, 회전까지 구현해야 한다. 특히, x,y,z가 아닌 yaw, roll, pitch라는 새로운 좌표 개념이 도입되므로 이를 적절히 처리해야 한다
Dynamics	카메라 이동 관련 입력이 주어졌을 시
Goals	카메라 이동을 적절하게 처리한다

5) Obejct control

Purpose	물체의 구현 및 추가 제거에 대한 처리
Approach	현재 구현하고자 하는 물체가 올바르게 연산될 수 있도록 하고, 추가 및 제거를 자유롭게 할 수 있도록 한다
Dynamics	해당하는 입력이 주어졌을 시
Goals	물체의 추가 제거 및 화면의 깨짐 없이 정상적인 출력이 되도록 한다

6) Light control

Purpose	조명에 따른 색상 및 조명 모델 적용에 대한 처리
Approach	일반적인 백색광뿐만 아니라 특정 색상을 가지는 광원에 대해서도 올바르게 색상의 반사 및 추적이 가능하도록 한다
Dynamics	연산 과정 및 조명 관련 입력이 주어졌을 시
Goals	물체의 색상 및 그림자 처리를 올바르게 한다

5. Problem statement

1. 연산 시간
사실성을 추구하는 레이 트레이싱 방식인 만큼, 연산에 매우 큰 시간이 걸릴 것으로 예상된다. 따라서 카메라 이동 입력과 장면 출력 사이에 딜레이가 발생하여 부정적인 경험이 나올 수 있다

2. 파일 양식
저장 및 출력에 사용될 파일의 경우 해당 양식을 어떻게 지정하느냐에 따라 프로그램과 유저 양쪽에 불편을 야기할 수 있다. 수치를 자세하게 준다면 프로그램이 추가적으로 연산해야 할 정보가 적어 편리하지만 유저가 직접 작성하여 수정하기에는 불편할 수 있고, 수치를 간략하게 준다면 유저가 생성하기에는 편리하지만 프로그램이 계산하는데 있어 실수부분의 오차와 추가적인 연산량이 늘어나 부담이 갈 수 있다.

3. 깊이 및 계산 복잡도, 오차 범위 설정
반사횟수, float와 double중 어떤 자료형으로 계산을 수행 할 것인지, 오차범위는 어떻게 지정할 것인가에 따라 요구되는 복잡도와 출력되는 결과물의 퀄리티의 차이가 상당할 것으로 생각된다. 특히, 오차범위의 단위를 어떻게 지정하느냐에 따라 연산이 늘어나지만 정확한 구현이 될수도, 연산이 줄어들지만 픽셀이 튀거나 매끄럽지 않은 화면이 나타날수도 있다.

NFRs
1. 기본 화면 해상도는 1280*720으로 한다
2. 입력이 없을 경우 화면의 갱신은 이루어지지 않는다
3. minilibx는 C 기반이나, 프로그램은 C++로 작성한다

6. Glossary

Minilibx :
42 Paris 에서 제작한 교육 목적 API. X-Window 시스템에서 그래픽 관련 공부를 돕기 위해 제작되었음.
Ray-Tracing :
광선 추적 기반 렌더링 알고리즘

7. References

1. Minilibx Github Repository :
https://github.com/42Paris/minilibx-linux
2. Ray-Tracing :
https://en.wikipedia.org/wiki/Ray_tracing_(graphics)
3. 3D Graphic Basics :
https://www.scratchapixel.com/
4. Phong Lighting Model Basics :
https://learnopengl.com/Lighting/Basic-Lighting
5. 레이 트레이싱에 사용되는 기본 수학 이론 강좌 :
https://www.khanacademy.org/computing/pixar/rendering/rendering1/v/rendering-1