

전차전 시뮬레이션을 위한 3차원 그래픽 게임 제작 연구

○

박애희*, 장현아*, 공길영**, 심훈***, 송재훈***, 조창석*
*한신대학교 정보통신학과
**한국해양대학교 운항시스템공학부
*** (주)미루정보통신

A Study on the 3D Graphics Simulation Game Implementation for Tank Battle

○

Ae-hui Park*, Hyun-ah Jang*, Gil-Young Kong**,
Hoon Shim***, Jae-hoon Song***, Chang-suk Cho*
* Dept. of Information Science & Telecommunications, Hanshin Univ.
E-mail : moonlight79@orgio.net, jjangah@orgio.net, cscho@hanshin.ac.kr
** Div. of Ship Operation Systems Engineering, Korea Maritime Univ.
***Miru corporation
E-mail : harry@mirunet.com, jason@mirunet.com

요 약

본 연구에서는 3차원 그래픽 시뮬레이션 게임 제작 연구를 목적으로 하여 3차원 그래픽 전차 모델을 제작하고 이를 이용한 전차전 시뮬레이션 게임 제작을 목표로 한다. 게임 상에는 아군과 대항군의 2대의 전차가 사용되며 대항군 전차는 통계적인 함수 모델을 이용하여 자체적인 전투 수행을 할 수 있도록 하고, 아군 전차는 사용자가 움직임을 통제한다. 본 논문의 결과로 사용자 탱크가 자유자재로 지형 위를 움직이며 상대편 탱크와 전투를 하는 시스템이 개발된다.

1. 서론

최근 많은 사람들이 3차원 그래픽을 이용해서 현실과 동일한 영화나 게임의 세계에 흥미를 느끼고 있다. 또한, 하드웨어의 발달로 개인용 PC에서도 단순한 그래픽 표현을 넘어 보다 다양하고 현실감 있는 그래픽 표현이 가능하게 되었다. 이런 사회적인 흐름을 반영하여 본 논문에서는 보다 실질적이고 현실감 있는 게임제작 연구를 하려고 한다. 이를 위해, 실제감을 주기 위해 2D가 아닌 3D로 제작하며, 실세계의 전차를 모델로 하여 사실감을 부여한다. 따라서, 현실에서는 어려운 전차전을 가상의 공간에서 경험하며 즐길 수 있다. 또한, 게임 제작에 필요한 개발 라이브러리는

그래픽스 모델 자체에 높은 완성도를 지원하고 있는 OpenGL 라이브러리를 사용하며 프로그래밍 언어로는 Visual C++을 사용한다.

본 논문에서는 전차전에 사용될 오브젝트들의 렌더링과 지형 렌더링, 아군탱크의 움직임, 대항군 탱크의 움직임에 관한 애니메이션 제작을 서술한다. 또한, 두 탱크 사이에 전투를 위해 쓰이는 방법을 제시하며 적용된 카메라 및 특수효과에 대해 서술한다. 결론으로는 기대효과 및 향후 연구과제를 제시한다. 이러한 과정에 걸쳐서 사용자 탱크가 지형 위를 움직이며 대항군 탱크와 게임을 하는 시스템이 개발된다.

2. 오브젝트 렌더링

전차전에 사용될 오브젝트(아군 탱크, 대항군 탱크)를 생성하기 위해 우선, 3D Max에서 각각의 오브젝트를 모델링한다.

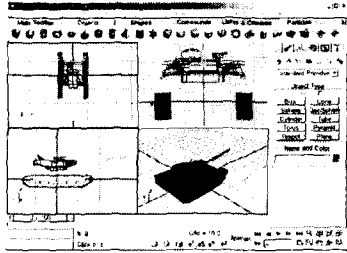


그림 1. 3D MAX에서 아군 탱크제작

다음으로 3D Max에서 모델링한 오브젝트들을 ASE(ASCII Scene Export) 파일로 Export시킨다. 그 후, ASE 파일 중 오브젝트의 좌표값 정보만을 얻는 프로그램을 작성한다. 좌표값 정보로는 정점(Vertex) 정보와 정점을 이용하여 삼각형 구성을 위해 사용되는 인덱스 정보이다. 또한, 오브젝트 부분별 애니메이션을 위해 각 부분에 관한 데이터 파일을 따로 생성하도록 한다. 데이터 파일의 이름은 3D Max에서 부여한 오브젝트 각 부분별 이름으로 생성되어지며 확장자는 .d 파일로 된다. 즉, ASE파일로부터 오브젝트 좌표값을 얻는 프로그램의 출력값인 d파일의 구성은 다음과 같다.

정점 개수(n)
인덱스 개수(m)
0번째 정점의 x,y,z 좌표값
1번째 정점의 x,y,z 좌표값
.
n번째 정점의 x,y,z 좌표값
0번째 인덱스의 3개의 버텍스 번호
1번째 인덱스의 3개의 버텍스 번호
.
m번째 인덱스의 3개의 버텍스 번호

그림 2. 오브젝트 좌표값 정보인 d파일의 구성

각 부분별로 얻어진 데이터 파일을 읽어들이 Visual C++에서 OpenGL 라이브러리를 이용하여 렌더링한

다. 이 때 오브젝트 렌더링은 각 오브젝트를 Class 별로 제작하여 재사용이 가능하도록 한다. 먼저, 파일에서 읽어들이는 오브젝트 좌표데이터와 인덱스를 메모리에 저장한 후 인덱스 별로 데이터를 언어와 삼각형별로 구성한다. 그리고 인덱스 개수만큼의 삼각형으로 구성된 오브젝트에 재질 색상값을 주고 비트맵을 입힌다. 비트맵의 크기는 256*256이며, 맵과 재질 색상과, 빛에 영향을 받아 오브젝트 색이 결정되도록 한다. 오브젝트 각 좌표점이 비트맵 해당 pixel을 얻도록 오브젝트의 가로, 세로 최대 최소값을 구하여 좌표점에 대한 pixel위치를 계산해서 맵을 오브젝트에 입힌다. 또한, 오브젝트의 각 면과 빛이 이루는 각도에 따라 빛의 양을 달리하여 사실감을 부여하기 위해 면에 대한 표면 법선 벡터를 계산하여 적용한다. 표면 법선 벡터는 식(1)에 의해 구해진다.

3차원 상의 세 점 $P_1=[x_1,y_1,z_1]$, $P_2=[x_2,y_2,z_2]$,

$P_3=[x_3,y_3,z_3]$ 에 대해,

V_1 은 점 P_1 과 점 P_3 를 잇는 벡터이고, V_2 는 점 P_1 과 점 P_2 를 잇는 벡터라 하면, 표면 법선 벡터N은 다음과 같다.[1]

$$\therefore N = V_1 \times V_2 \quad (\times: \text{외적}) \quad (1)$$

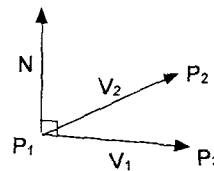


그림 3. Normal Vector

그리고, 광원의 위치와 광도를 설정해 주어 사실감 있게 오브젝트를 렌더링한다. 이렇게 렌더링한 아군탱크 오브젝트의 모습이다.

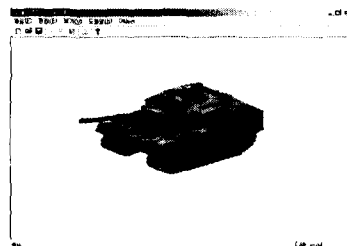


그림 4. 아군 탱크 렌더링

3. 지형 렌더링

지형은 전체 약 60000 pixel × 60000 pixel로 구성되어 있으며 페이지 클리핑(clipping)을 위해 30×30개의 파일로 이루어져 있다. 그리고 한 개의 Object에는 10000개의 정점을 갖도록 제작한다. 지형은 실제 크기에 대해 6.9cm : 1 pixel로 축소된 것이다. 따라서 전체 지형의 실제 크기는 가로 4.14km × 세로 4.14km이며 넓이는 17.1396km²로 울릉도(73km²)의 약 $\frac{1}{4}$ 이 된다. 또한 한 파일은 약 138m × 130m로 19.044m²이 된다. 지형도 오브젝트 모델링과 같이 3D Max로 제작 후, 모델링 된 지형에 대해 ASE 파일 변환 후 좌표 데이터 파일을 생성한다. 데이터의 용량을 고려하여 3D Max의 Quad Patch 기능을 사용하여 일정 크기의 한 면을 생성한다. 생성된 면의 4개의 점을 이용하여 지형의 필요로 한 형태를 갖추도록 한다. 지형의 전체적인 모습을 고려하여 언덕, 산, 평지를 구성한다. 모델링한 전체지형을 부분별로 렌더링하기 위해 페이지 클리핑을 사용한다. 페이지 클리핑(clipping)은 지형을 부분별로 나누어 탱크의 움직임에 따라 지형의 한 부분에서 다른 부분으로 넘어가면서 메모리에 올려지고 화면에 그려지게 된다. 전체 지형은 30×30개의 파일로 만들어 총 900개의 페이지로 만든다. 그 중 9개의 페이지만을 메모리에 올려 그려지게 된다. 3×3의 9개의 페이지는 탱크가 위치한 페이지를 중심(4번 페이지)으로 사방 8개의 페이지가 선택된다.

0	1	2
3	4	5
6	7	8

그림 6. 3×3 페이지 클리핑 윈도우

탱크가 움직일 때 8개의 페이지 방향 중 어느 방향으로 움직일 것인가를 감지하여 그 페이지를 중심으로 새롭게 9개의 페이지 클리핑 윈도우에 올라오게 된다. 이렇게 클리핑 윈도우를 하나가 아닌 9개의 블록으로 잡은 것은 탱크가 움직임에 따라 자연스럽게 지형이 넘어가기 위해서다.

4. 아군 탱크 움직임 엔진

아군 탱크의 움직임은 사용자가 키를 조작하여 회전

하거나 진행한다. 진행 시 지형의 좌표점을 비교하여 3D 이동을 하게 된다. 또한 사실적인 움직임을 주기 위해 실제 전차와 같은 애니메이션을 부여한다. 이를 위해, 탱크는 포신이 상하로 움직이며, 포신과 함께 포탑이 360°로 회전하도록 한다. 포신이 상하로 움직이게 하기 위해 움직임의 중심축이 되는 좌표값을 구한다 이 좌표값을 기준으로 하여 상하로 1.0°씩 이동시키며, 이 때 일정 각도 이상 올라가거나 내려가는 것을 방지하기 위해 값의 제한을 준다. 또한, 포탑은 포탑 중심을 구하여 이를 축으로 5.0°씩 이동시켜 360°회전시킨다. 포탑의 이동시 포신도 같이 움직임을 준다.



그림 7. 포신과 포탑 애니메이션

탱크의 움직임 애니메이션은 계층형 애니메이션으로 구성된다. 계층형 애니메이션은 오브젝트 각 부분이 모두 부모와 자식간의 트리 형태로 구성된다. 부모와 자식간의 관계는 자식이 부모의 어느 부분인지를 나타내는 연결 매트릭스로 구성되며 어떤 부분의 애니메이션을 만들기 위해서 연결 매트릭스와 그 부분 상태를 나타내는 매트릭스를 계속 곱해나가면 된다. 탱크의 경우 각 부분의 트리는 다음과 같다.

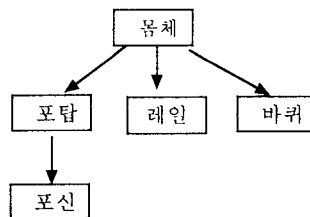


그림 8. 아군 탱크의 트리구조

예를 들어 포신의 움직임에 대한 매트릭스를 구할 때 포신의 부모 노드인 포탑의 영향을 받는다. 또한 포신은 포탑의 부모 노드인 몸체의 영향도 받는다. 따라서 우선, 몸체의 움직임에 대한 매트릭스에 포탑 연결 매트릭스를 곱하고 여기에 포탑의 움직임 매트릭스를 곱한다. 그런 다음에 포신의 연결 매트릭스와 포신의 상대적인 움직임을 나타내는 로컬 매트릭스를 곱하여 구한다.

5. 대항군 탱크 움직임 엔진

대항군 탱크는 통계적인 함수 모형을 이용하여 자체적으로 움직여 아군탱크와 대항한다. 대항군 탱크는 지형에 처음 출현하는 장소를 Random하게 한다. 그리고 대항군 탱크는 Random하게 움직여 그 방향을 사용자가 예측할 수 없게 한다. 또한, 대항군 탱크의 Random한 움직임에 확률을 달리 적용하여 대항군 탱크가 아군탱크로 전투를 위해 접근할 수 있게 한다. 이를 위해 대항군 탱크의 위치를 중심으로 아군 탱크가 어느 방향에 있는지를 감지한다. 그리고 통계적 함수 모형 중 정규분포함수를 사용하여 대항군 탱크의 Random한 움직임에 있어 아군 탱크의 방향으로 더 많은 확률을 부여한다.

우선, 대항군 탱크가 아군탱크의 이동 방향 감지를 보면 다음과 같다. 대항군 탱크는 자신의 위치를 중심으로 12방향 중 아군 탱크가 어느 방향에 있는지를 파악한다. 이를 위해, 아군 탱크의 위치에서 대항군 탱크의 위치를 뺀 값의 부호로 4방향을 설정한다.

즉, x 좌표값의 차이는 dx , z 좌표값의 차이는 dz 라고 하여, 아래의 방식으로 방향을 정한다.

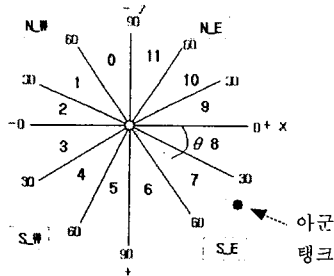


그림 9. 아군탱크 방향 감지

그림 9에서 보면, $(dx, dz) = (+,+)$ 이면 S_E를, $(+,-)$ 면 N_E를, $(-,-)$ 이면 N_W를 선택하고, $(-,+)$ 이면 S_W 방향을 선택하게 된다. 또한, 각도로서 세부적인 방향을 설정하게 된다. 각도 θ 는 식(2)에 의해 계산되어진다.

$$\theta = \tan\left(\frac{|z|}{|x|}\right)^{-1} \quad (2)$$

각도 θ 는 각 방향마다 0에서 90도 사이의 값을 가지게 되며 30°씩 나누어져 있다. 부호와 각도에 의해 0부터 11까지의 방향 중 아군탱크의 방향을 감지하게 된다.

다음으로, 대항군 탱크는 12방향 중 감지된 아군 탱

크 방향에 더 많은 확률을 부여하고 아군탱크의 방향과 멀어질수록 적은 확률을 가지고 임의로 움직이게 된다. 여기서 사용된 통계적 확률 분포는 정규분포이다. [2]

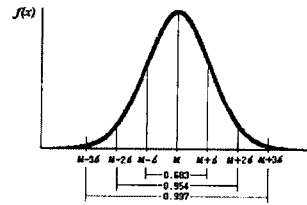


그림 10. 정규 분포의 확률밀도 곡선

정규분포는 확률적인 특성을 볼 때 그림 10의 특성을 지닌다. 즉 $P(\mu - \sigma \leq x \leq \mu + \sigma) = 0.683$ 이 되며, $P(\mu - 2\sigma \leq x \leq \mu + 2\sigma) = 0.954$ 가 된다. 또한, $P(\mu - 3\sigma \leq x \leq \mu + 3\sigma) = 0.997$ 의 값을 갖는다. 이러한 정규분포의 확률적인 특성을 대항군탱크 이동 경로를 위한 각도 설정에 이용한다.

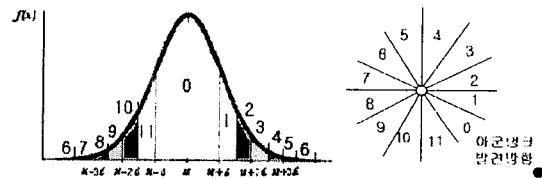


그림 11. 대항군탱크 움직임에 정규분포 확률 적용

즉, 정규분포 곡선의 확률적 특성을 고려하여 분포 면적을 확률로 사용한다. 또한 아군 탱크가 발견된 방향을 평균값에 가까이 뚫으로써 정규분포의 특성에 따라 대항군 탱크가 아군 탱크가 있는 방향을 선택할 확률이 커지게 된다. 그림 11은 각 방향을 나타내는 번호에 부여되는 확률 면적이 된다. 이 때 각 번호는 아군탱크 발견 방향을 0으로 시작하여 반 시계방향으로 부여된 순서다. 이러한 정규분포 확률 면적은 표준 정규분포로 변환하여 이용하게 된다. 여기서, 대항군 탱크가 아군탱크로 올 확률은 약 68.3%가 된다.

정규분포의 확률에 의해 임의적으로 움직이는 대항군 탱크는 아군탱크가 대항군 탱크의 일정 사격거리 안에 들어오면 아군탱크를 향해 전투를 한다.

6. 전투

아군 탱크는 포를 쏘아 대항군 탱크를 맞추게 된다.

대항군 탱크 역시 아군 탱크를 향해 포를 쏘게 된다. 포 사격은 포가 나가는 포신의 끝점의 중심에서 시작하여 일정 거리 안의 탱크를 맞추게 된다. 포 경로는 그림 12와 같다.

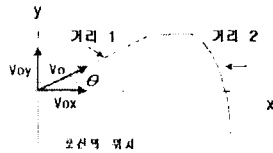


그림 12. 포 사격 경로

포 사격 경로는 '거리1'에 의해 일정거리만큼 올라간 후에 '거리2'에 의해 속도가 줄어 낙하 운동을 하게 된다. 그림 12에서 수평에 대하여 각 θ 로 초속도 V_0 로 던져 올린 것으로 V_0 성분은 식(3)과 식(4)과 같다.

$$V_{0x} = V_0 \cos \theta \quad (3)$$

$$V_{0y} = V_0 \sin \theta \quad (4)$$

그리고 '거리1'은 식(5)과 식(6)와 같다.

$$x = V_{0x}t \quad (5)$$

$$y = V_{0y}t \quad (6)$$

위의 식에서 t 는 시간을 나타낸다. V_0 속도로 t 의 시간동안의 위로 이동하게 된다. 여기서 x 는 앞으로 나아가는 직선 거리이며 y 는 포가 위로 올라가는 높이를 나타낸다. 그 후, 점점 속도가 줄어 '거리2'에 의해 포가 아래로 떨어지게 된다. 여기서 '거리2'는 식(7)과 식(8)과 같다.

$$x = V_{0x}t \quad (7)$$

$$y = V_{0y}t - \frac{1}{2}gt^2 \quad (8)$$

'거리2'는 '거리1'에서와 같이 식(7)의 x 는 포의 직선 거리이다. 식(8)의 y 는 높이이고, g 는 중력 값으로 약 9.8을 사용한다. 식(3)에서 식(8)까지의 상위 수식에서는 단순화를 위하여 가속도는 나타내지 않았다.[3]

포가 경로를 따라 이동하다가 목표 오브젝트와 충돌

이 되었는지 여부를 판단하게 된다. 이 때 충돌은 한 점과 삼각형과의 충돌을 사용하게 된다. 이는 포가 포 발사 경로를 따라 이동할 때 점의 이동이 되며, 목표 오브젝트는 삼각형으로 구성되어 있기 때문이다.

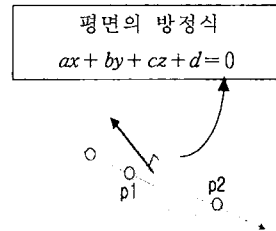


그림 13. 점과 삼각형의 충돌 검색

그림 13에서 $p1$ 은 충돌 바로 이전의 포의 위치가 되며 $p2$ 는 현재 포의 위치가 된다. $p1$ 과 $p2$ 점을 삼각형의 평면의 방정식에 대입하여 서로 부호가 다르거나, 둘 중 하나가 0이나오면 충돌로 판단한다. 이는 포가 오브젝트 면에 바로 닿을 수도 있지만 면을 뚫고 지나가는 경우도 충돌로 보아야 하기 때문이다. 삼각형의 평면의 방정식은 식(9)와 같다.

$$ax + by + cz + d = 0 \quad (9)$$

식(9)에서 a 는 삼각형 벡터의 x 성분이 되며, b 는 벡터의 y 성분이 된다. 그리고 c 는 벡터의 z 성분이 된다. 이들 벡터는 식(1)을 이용하여 구한다. d 는 식에서 삼각형 내부의 임의의 점(x_0, y_0, z_0)을 x, y, z 에 대입하여 유도하면 된다[4].

그리고, 전투 중 두 탱크간의 3D충돌검출은 경계구 충돌을 이용한다. 경계구 충돌은 객체의 중점을 찾고 그 중점과 가장 먼 거리가 있는 객체의 정점을 찾은 후, 객체의 중점을 구의 중심으로 하고, 가장 먼 정점과의 거리를 구의 반지름으로 하면 두 객체에 대해 각각 경계구가 계산되어 진다. 그 결과, 각 경계구의 중점들 사이의 거리가 각 경계구들의 반지름의 합보다 작으면 두 경계구는 충돌된다고 판단된다. 경계구 충돌은 충돌 판단에 있어 가장 많이 사용된다. 그러나 경계구 충돌 판단 후 세부적 충돌 검출이 더 필요하다.

7. 카메라 및 특수효과

3D VR 시뮬레이션의 장점중의 하나인 게임 속의 한

부분이 된 것처럼 느끼는 사실감을 나타내기 위해 아군탱크와 함께 카메라를 이동시킨다. 카메라는 원근법을 나타낼 수 있는 Perspective카메라를 사용하며, 카메라 위치를 설정하기 위해서 OpenGL에서 지원하는 gluLookAt 이라는 함수를 사용 한다. gluLookAt 인자값으로 카메라의 위치와 카메라가 바라보는 벡터 방향을 설정한다. 카메라의 움직임을 위해 카메라의 위치, 화면의 중심, 그리고 관측자가 바라보는 방향의 위쪽을 향하는 벡터에의 값을 조정하면서 카메라의 위치를 변화시킨다[5].

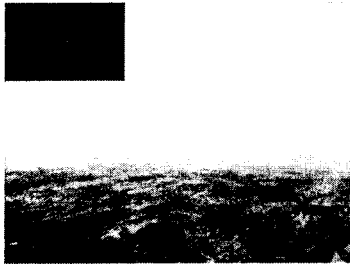


그림 14. 카메라 시점(3인칭)



그림 15. 카메라 시점(1인칭)

카메라의 시점은 1인칭 시점과 3인칭 시점에서 사용자가 번갈아 가며 보게 된다. 3인칭 시점은 배경이나 대항군 탱크, 아군탱크 등 전반적인 모습을 볼 수 있다. 1인칭 시점은 마치 자신이 탱크에 타고서 운전하고 조준해서 대항군 전차를 사격하는 등 좀더 사실적인 느낌을 준다[6].

특수효과로는 사운드와 안개 효과를 사용한다. 사운드는 Direct Sound를 사용한다. 이는 멀티 사운드나 3D 사운드를 이용하는 것으로 Direct X에서 제공되는 하나의 Class이다. Direct Sound는 wave파일 형식으로 로드하는 데 걸리는 시간을 감소시킬 수 있으며, 사운드 카드 및 스피커와 관련된 메모리 환경을 제어할 수 있다. 사운드는 아군탱크가 키 입력을 받아 앞, 뒤

로 움직일 때, 소리가 나도록 한다[7-8]. 또한 안개효과는 OpenGL에서 제공하는 glFog 함수 인자값 중 GL_EXP를 사용하여 전체적인 화면에 효과를 줌으로써 실제 지형 같은 느낌을 더해준다.

8. 결론

3차원 그래픽 시뮬레이션 게임 제작 연구를 통해 게임 프로그램에서 사용되는 기본적인 이론이나 알고리즘을 익히고 적용할 수 있었다. 또한, 사실감 있는 표현이 가능하도록 하는 빛과 카메라, 재질, 맵핑 같은 3차원 그래픽 표현을 연구할 수 있었다. 연구 결과, ASE파일에서 오브젝트 좌표값만을 얻어낼 수 있는 프로그램을 설계함으로써, 게임 프로그램 제작에 쉽게 접목시킬 수 있었고, 2대의 탱크가 서로 전투를 하는 부분은 Class를 이용하여 제작하기 때문에 차후에 비슷한 게임 제작시 변형을 쉽게 할 수 있을 것이다. 아쉬운 점이 있다면, Direct X에 비해, OpenGL에서 지원하는 게임 요소가 부족하였기 때문에, 연구하는데 어려움이 있었으며, 고속화 알고리즘의 필요성을 느꼈다. 그리고 본 연구에서는 다양한 게임환경 등의 게임적인 요소가 많이 부족하기 때문에 흥미를 유발할 수 있는 게임적인 요소와 그에 따르는 알고리즘 및 고속화 알고리즘 등에 대한 후속연구가 필요하다. 따라서 본 연구를 기초로 여러 가지 요소를 추가하여 완성도 높은 게임 제작할 수 있도록 해야 한다.

[참고문헌]

- [1]Donald Hearn and M.Pauline Baker, "Computer Graphics", Prentice Hall, pp408~pp430, 1997
- [2]박명섭, 박광태 지음, "통계학 개론", 홍문사, pp141~150, 1996
- [3]Jerry B. Marion, 김정구(외), "대학 기초 물리학", 탐구당, pp51~55, 1991
- [4]Mark Deloura 외 공저, "GAME PROGRAMMING Gens ", 정보 문화사, pp502~507, 2001
- [5]Richard S. Wright Jr, Michael Sweet, "OpenGL Superbible"(Second Edition), 인포북, pp213, 2001
- [6]Mark Deloura 외 공저, "GAME PROGRAMMING Gens ", 정보 문화사, pp480~490, 2001
- [7]Andre LaMothe, "Tricks of the WINDOWS Game Programming", MINPRESS, pp590~621,
- [8]Clayton Walum, "윈도우 95용 GAME SDK 전략 가이드", 정보 문화사, pp1249~306