**기초 컴퓨터 그래픽스**

**HW2 README**

20161663 허재성

**1. [환경 명세]**

1) 본인 프로그램의 실제 구동 환경을 명시 할 것 (OS, CPU, GPU, Compiler 등등)

Microsoft Windows 10 Home 64bit, i3-5005U, Intel(R) HD Graphics 5050,

Visual Studio 2019 Win 64 release

**2. [요구사항]**

(a) 다음 왼쪽 그림의 9개의 물체 중 5개를 선택한 후, 각 물체에 대하여 서로 다른 창의적인 동적 효과를 생성할 것. (각 물체당 최대 15점 (최대 총 75점))

1) 사용한 물체: 집



- 부여한 창의적인 동적 효과:

배경인 지구에서 집이 솟구쳐 올랐다가 정점에 정점에 이른 후 다시 떨어지는 것을 표현하였다. 올라갈 때는 관찰자 입장에서 가까이 오는 것으로 보이도록 점점 크기가 커졌다가 떨어질 때에는 지구로 떨어지며 관찰자로부터 멀어지게 하기 위해 올라갈 동안에는 크기가 점점 증가했다가 내려갈 때에는 크기가 점점 감소하도록 했다. 또한 정점에서 떨어질 때는 뒤집혀서 떨어지도록 했다.

이를 위해 sine 함수의 그래프를 이용했다. Sine 함수에서 독립변수인 각도가 0 ~ 180도일 때 정점까지 증가했다가 감소하는 것을 이용하여 시간 변수를 각도인 x좌표, 시간 변수에 sine 값을 취한 것을 y 좌표로 하여 translation을 하였다. 그러면 시간에 따라 sine 함수의 궤적으로 집이 솟구쳤다 떨어지는 것을 구현할 수 있다. 또한 sine 함수(적절한 크기의 상수를 곱하여)의 크기만큼 scale해주면 자연스럽게 X가 0 ~ 90까지는 크기가 증가하고 90 ~ 180에서는 크기가 감소하도록 구현하였으며, 90 ~ 180에서는 scale할 때 y 성분에 -1을 곱하여 집이 x 축 기준으로 뒤집힌 채 떨어지도록 하였다.

- 확인 방법: 실행하면 집의 기준 시계 hc가 1/8씩 증가한다.(집의 상승 및 하강 속도를 적절히 느리게 하기 위해서이다.) hc % 180를 x값으로 하여 180 주기마다 집이 올라갔다 내려오는 것을 알 수 있다.

2) 사용한 물체: 셔츠



- 부여한 창의적인 동적 효과:

1의 집이 정점에서 떨어질 때 집에 있던 옷가지가 떨어진다는 설정으로 셔츠가 집보다 먼저 떨어지기 시작한다. 1의 집의 시간인 hc가 90에서 180 사이일 때 나타난다. 화면 위쪽에서 포물선 궤적을 그리면서 떨어진다. 이 때 셔츠의 떨어지는 궤적은 포물선이지만 모자 자체가 흔들리며 떨어지므로 rotate을 이용하여 모자가 원 궤도의 일부분 모양으로 흔들리며 떨어진다. 흔들리는 크기(회전 반경)를 위해 translation을 사용했다.

포물선을 구현하기 위해 이차 함수의 그래프를 이용했다. hc가 90에서 180 사이일때 모자의시간 hatc가 0 ~ 90이 되도록 하고 hatc에 적절한 상수를 곱하여 x 좌표로, hatc의 제곱에 0.1을 곱하여 y 좌표로 하여 translation하여 최고차항의 계수가 음수일 때 4사분면에서의 이차 함수 모양으로 떨어지도록 구현하였다. 또한 떨어지는 중에 원 궤도의 일부분으로 흔들리는 것을 구현하기 위해 rotate을 이용했다.

- 확인 방법: 확인 방법: 실행하면 집의 기준 시계 hc가 1/8씩 증가한다. hc % 180가 90에서 180 사이일 때 화면에 나타난다.

3) 사용한 물체: 모자



- 부여한 창의적인 동적 효과:

2)와 마찬가지로 1)의 집이 정점에서 떨어질 때 집에 있던 옷가지가 떨어진다는 설정으로 모자가 집보다 먼저 떨어지기 시작한다. 1의 집의 시간인 hc가 90에서 180 사이일 때 나타난다. 화면 위쪽에서 포물선 궤적을 그리면서 떨어진다. 이 때 모자의 떨어지는 궤적은 포물선이지만 모자 자체가 흔들리며 떨어지므로 rotate을 이용하여 모자가 원 궤도의 일부분 모양으로 흔들리며 떨어진다. 흔들리는 크기(회전 반경)를 위해 translation을 사용했다. 2)에서 추가적으로 모자가 떨어지면서 뒤집히면서 떨어지도록 구현했다.

포물선을 구현하기 위해 이차 함수의 그래프를 이용했다. hc가 90에서 180 사이일때 모자의시간 hatc가 0 ~ 90이 되도록 하고 hatc에 적절한 상수와 -1을 곱하여 x 좌표로, hatc의 제곱에 -0.1을 곱하여 y 좌표로 하여 translation하여 최고차항의 계수가 음수일 때 3사분면에서의 이차 함수 모양으로 떨어지도록 구현하였다. 또한 떨어지는 중에 원 궤도의 일부분으로 흔들리는 것을 구현하기 위해 rotate을 이용했다. 또한 모자가 뒤집히는 것을 구현하기 위해 특정 시간마다 x축 또는 y축 기준으로 회전 변환을 이용하였다. 떨어지면서 점점 가까워지는 것을 표현하기 위해 scale도 사용하였다.

- 확인 방법: 확인 방법: 실행하면 집의 기준 시계 hc가 1/8씩 증가한다. hc % 180가 90에서 180 사이일 때 화면에 나타난다.

4) 사용한 물체: 자동차2(측면)



- 부여한 창의적인 동적 효과:

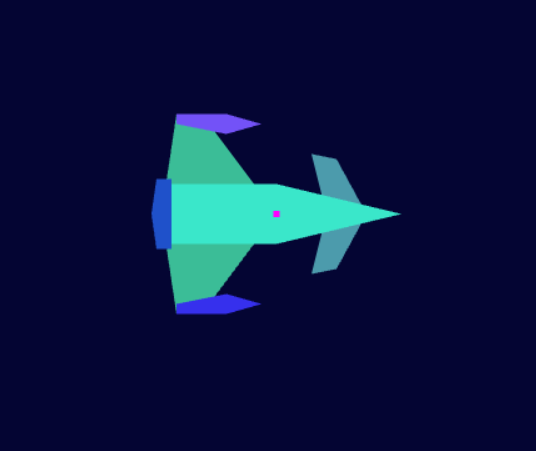
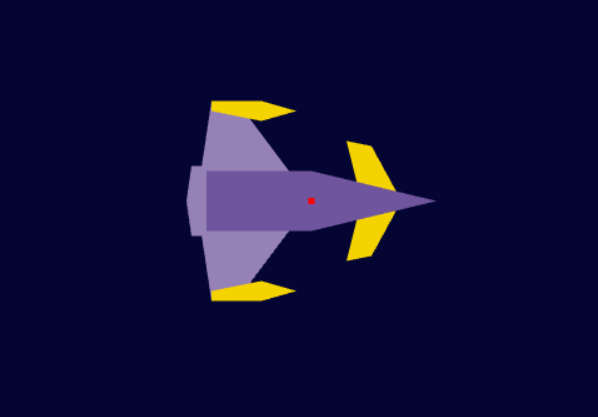
지구 상에 있던 자동차가 우주로 솟구쳐 올라왔다가 다시 지구로 떨어지고 또 다시 튀어 올라오는 것을 반복하는 것을 구현했다. 이를 구현하기 위해 cycloid 곡선의 매개 변수 방정식을 이용했다. Θ가 RADIAN 각도일 때 사이클로이드 곡선의 좌표는 다음과 같이 나타낼 수 있다.

translation을 이용해 자동차가 사이클로이드 궤도로 움직이는 것을 구현할 수 있다. 또한 사이클로이드 곡선 상에서 움직일 때 높이 올라갈수록 크기가 증가하고, 내려갈수록 크기가 감소하도록 scale을 이용했으며 사이클로이드 궤도를 따라 이동할 때 차가 공중 제비를 돌면서 이동하도록 했다. 이를 위해 rotate를 사용했다.

- 확인 방법:

timestamp1은 0부터 시작해 1씩 증가한다. 사이클로이드의 한 주기는 360도이므로 총 세 주기를 관찰하기 위해 timestamp % 1080을 이용한다. 1080마다 자동차가 왼쪽 아래에서 나오기 시작해 사이클로이드를 따라 이동하는 것을 확인할 수 있다.

5) 사용한 물체: 비행기(기본), 비행기(정예)



- 부여한 창의적인 동적 효과:

기본적으로 주어진 기본 배색 비행기(왼쪽)는 3대가 편대를 이루어 후술할 외계 비행체를 향해 돌진하는 것을 구현하였다. 단순한 직선 이동 같지만 등가속도 운동 공식을 이용하여 외계 비행체 쪽으로 갈수록 속도가 감소하도록 구현하였다. 또한 편대의 리더 비행기는 중간에 360도 회전하도록 구현하였다. 또한 편대 리더 비행기를 제외한 나머지 2개의 비행기는 비행 중간과 비행 마지막에 폭발하도록 구현하였다. 폭발 효과를 위해 후에 폭발 불꽃 물체를 따로 모델링하였다.

물체가 등가속도 운동을 할 때 시간에 따른 변위 공식은 다음과 같다. s는 변위, t는 시간, v0는 초기 속도, a는 가속도를 의미한다.

위의 공식을 이용해 비행기가 왼쪽 끝에서 이동하기 시작에 속도가 점점 감소하여 오른쪽 끝에 도달할 수 있도록 값들을 적절히 설정 후 translation을 이용하여 쉽게 구현할 수 있다. 가운데 리더 비행기의 경우 특정 시간 동안 x축 기준으로 회전하도록 rotate을 이용해 구현했다.

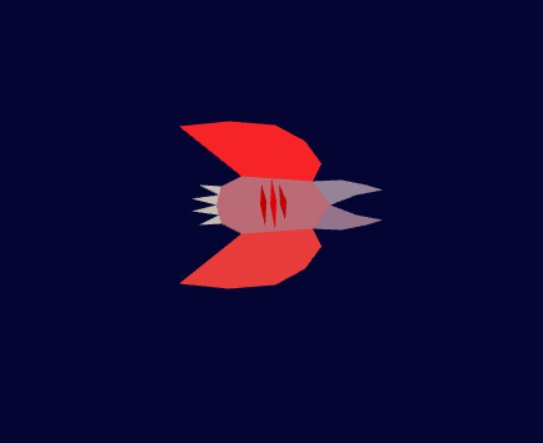
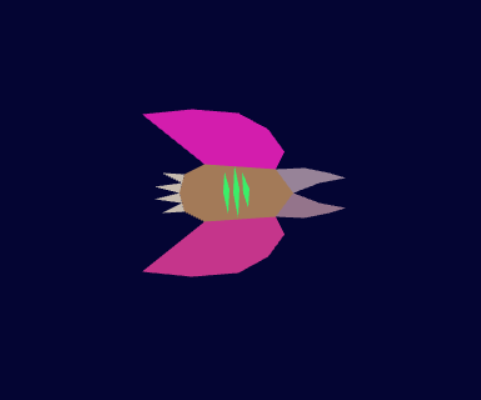
배색을 달리한 정예 비행기는 외계 비행체와 격렬한 교전을 벌이는 모습을 구현했다. 외계 비행체에게 당하고 있는 것을 구현하기 위해 외계 비행체 아래에서 크게 흔들리며 움직이는 것을 구현했다. 정예 비행기가 움직이는 궤도는 타원 궤도로 다음과 같은 타원의 매개 변수 방정식을 이용하였다.

이 때 a는 타원의 장반지름(가로 방향), b는 타원의 단반지름(세로 방향)이다. 이동 궤도는 타원의 윗부분이므로 Θ는 0부터 180까지이다. 이를 이용해 translation을 이용하면 타원 궤도로 움직이는 것을 구현할 수 있다. 움직이면서 비행기가 심하게 흔들리는 것을 구현하기 위해 z축 중심으로 rotate와 y축 기준으로 rotate를 조합하여 비행기가 격렬하게 흔들리는 것을 구현했다. 또한 비행기가 타원 정점으로 갈수록 크게 보이다 정점 이후 점점 작아지는 것도 scale을 이용해 구현했다.

- 확인 방법: 비행기의 시계는 0부터 시작에 0.5씩 증가한다. 비행기의 시계를 ac라 할 때 ac % 180이 0 ~ 180일 때 보이게 된다.

(b) 충분히 복잡도가 있는 2차원 기하 물체를 자신이 한 개 모델링 한 후, 위의 물체들처럼 창의적인 동적 효과를 생성하라. (최대 총 25점)

- 모델링한 물체 : 외계 비행체(기본, 왼쪽), 외계 비행체(정예, 오른쪽)



지구를 침공한 외계 생물체의 비행체로 기본적으로 좌우 대칭이지만 양쪽 날개를 적당히 둥글게 보이게 하기 위해 여러 개의 점으로 구현하였다. 또한 양쪽 날개를 따로 구현하여 서로 비슷하지만 다른 색을 입힐 수 있도록 하였다. 그리고 단순한 좌우 대칭 물체가 되는 것을 지양하고자 외계 비행체의 몸체에 흉터를 3개 구현하여 무늬를 만들었다. 오른쪽 정예 비행체의 경우 붉은 계통의 색상을 입혀 좀 더 흉포해 보이게 만들었다.

- 부여한 창의적인 동적 효과:

왼쪽의 기본 외계 비행체는 5개가 역삼각 형태로 움직여 위의 기본 비행기를 향해 돌진하는 역할로 기본 비행기와 다르게 등속 직선운동으로 움직이므로 크게 특별할 것은 없다. 후미에 있는 비행체가 x축을 기준으로 회전하긴 하지만 위의 비행기와 마찬가지로 이것만의 특별한 동적 효과는 없고 단순 배경용이다.

오른쪽의 정예 외계 비행체는 위에서 언급한 정예 비행기와 전투를 벌이는 것을 구현했다. 비행기를 상대로 우위를 점하는 것을 구현하기 위해 시종일관 비행기의 위에서 움직이도록 구현했으며, 비행기가 심하게 흔들리는 것과 다르게 회전하며 움직이는 것만 구현하였다. 또한 중간 즈음에 회전 방향을 반대로 바꾸어 역회전하는 것도 구현하였다.

정예 외계 비행체의 타원 궤적은 위의 공식을 이용했지만 비행기보다 위에 있게 하기 위해 a, b의 값을 더 크게 한 후 translation을 이용하였다. 비행기와 같은 각도를 이용하여 0 ~ 90일 때에는 z축 기준으로 빠르게 양의 방향으로 회전하다 90 ~ 180일 때에는 천천히 음의 방향으로 역회전하도록 구현하였다. 이를 위해 Rotate를 이용하였다. 또한 비행기와 마찬가지로 크기가 변화하도록 scale을 이용하였다.

- 확인 방법:

일반 외계 비행체의 시계 sc는 비행기와 다르게 1/3씩 증가한다. sc % 180가 0 ~ 180 사이일 때 화면에 나타난다. 비행기보다 시계의 시간 증가 속도가 느리므로 처음 속도는 좀 더 느리다.

정예 외계 비행체는 정예 비행기와 시간을 공유한다. 따라서 비행기가 나타날 때 같이 확인 가능하다.

추가로 폭발 효과를 위해 또 다른 물체를 구현하였다.



배경인 지구 곳곳에서 폭발이 일어나는 것을 구현하기 위해 rand 함수를 이용해 일정시간마다 지구에 해당하는 위치들 중 임의의 위치에 랜덤하게 여러 개의 위의 물체를 나타나도록 하였다. 이 때 폭발이 찌그러지거나 늘어날 수 있도록 적절히 scale을 이용하였으며 또한 뒤집혀 나타날 수도 있도록 rand 함수를 이용해 일정 확률로 -1을 곱하여 구현하였다. 또한 비행기가 끝에 도달하거나 중간에 폭발하는 효과를 구현하기 위해서도 사용되었다.

그 외에도 배경인 지구를 위한 물체를 정의하였으며 우주 공간의 별들을 나타내기 위한 물체를 정의하였다. 하지만 특별한 아핀 변환을 이용한 동작은 따로 없이 배경을 위해서만 정의하였다. 별의 경우 일정 시간마다 임의의 위치에 별이 나타나도록 rand 함수를 이용해 구현했다.

(c) 제출한 숙제 중 가장 재미있고 복잡도가 있는 기하 변환 및 애니메이션 효과를 생성한 학생을 적절히 선정하여 최대 25점까지 추가 점수를 부여할 수 있음 (정확한 숫자는 상황에 따라 유동적임). 공정을 기하기 위하여 컴퓨터그래픽스 연구실 대학원생들이 공동으로 심사함.

- 추가 점수를 부여 받아야할 이유   
(복잡한 기하 변환과 재미있는 애니메이션 효과 사용과 확인 방법 등을 서술)

수업 시간에 배운 아핀 변환 중 translate, scale, rotate를 이용해 단순하지 않은 동작을 구현하기 위해 여러 수학적인 공식들을 이용하였다. Sine 함수와 포물선, 사이클로이드, 타원 등 알고 있지만 실제적으로 활용할 일이 많지 않았던 것들을 이용해 복잡한 동작을 구현하였다. 기본적인 아핀 변환을 이용해 충실하게 구현하였다. 과제에서 요구하는 필수 조건 외에도 완성도와 심미적인 면을 고려하여 다른 물체들도 구현하였다.