HW3\_S20161595

배성현

1. 사용 물체

(1) 선택한 물체

- 집(house), 비행기(airplane), 칼(sword), 자동차(car), 모자(hat)

(2) 모델링 한 물체

- 화분(flowerpot)

|  |
| --- |
| #define FLOWERPOT\_BODY 0  #define FLOWERPOT\_BODY2 1  #define FLOWERPOT\_HEAD 2  #define FLOWERPOT\_HEAD2 3  #define FLOWERPOT\_DOWN 4  GLfloat flowerpot\_body[4][2] = { { -12.0, 0.0 },{ -12.0, 4.0 },{ 12.0, 4.0 },{ 12.0, 0.0 } };  GLfloat flowerpot\_body2[4][2] = { { -11.0, 0.0 },{ -8.0, -16.0 } ,{ 8.0, -16.0 },{ 11.0, 0.0 } };  GLfloat flowerpot\_head[4][2] = { { -1.0, 4.0 },{ -1.0, 20.0 } ,{ 1.0, 20.0 },{ 1.0, 4.0 } };  GLfloat flowerpot\_head2[9][2] = { { -2.0, 20.0 },{ -6.0, 17.0 } ,{ -10.0, 22.0 }, { -5.0, 24.0 },{ 0.0, 22.0 } ,{ 5.0, 24.0 }, { 10.0, 22.0 },{ 6.0, 17.0 } ,{ 2.0, 20.0 } };  GLfloat flowerpot\_down[4][2] = { { -9.4, -16.0 } ,{ -9.4, -17. },{ 9.4, -17.0 },{ 9.4, -16.0 } };  GLfloat flowerpot\_color[5][3] = {  { 170 / 255.0f, 125 / 255.0f, 0 / 255.0f },  { 188 / 255.0f, 80 / 255.0f, 0 / 255.0f },  { 0 / 255.0f, 254 / 255.0f, 36 / 255.0f },  { 0 / 255.0f, 174 / 255.0f, 0 / 255.0f },  { 255 / 255.0f, 255 / 255.0f, 255 / 255.0f }  }; |

- 땅(ground, 배경으로 사용)

|  |
| --- |
| #define GROUND\_MAIN 0  #define GROUND\_ROAD 1  #define GROUND\_STONE2 2  #define GROUND\_STONE3 3  GLfloat ground\_main[4][2] = { { -500.0, -180.0 },{ -500.0, -400.0 },{ 500.0, -400.0 },{ 500.0, -180.0 } };  GLfloat ground\_road[4][2] = { { -500.0, -200.0 },{ -500.0, -235.0 },{ 500.0, -235.0 },{ 500.0, -200.0} };  GLfloat ground\_stone2[4][2] = { { 120.0, -190.0 },{ 120.0, -195.0 },{ 125.0, -195.0 },{ 125.0, -190.0} };  GLfloat ground\_stone3[4][2] = { { -10.0, -290.0 },{ -10.0, -295.0 },{ -15.0, -295.0 },{ -15.0, -290.0} };  GLfloat ground\_color[4][3] = {  { 80 / 255.0f, 165 / 255.0f, 0 / 255.0f },  { 49 / 255.0f, 68 / 255.0f, 45 / 255.0f },  { 49 / 255.0f, 68 / 255.0f, 45 / 255.0f },  { 49 / 255.0f, 68 / 255.0f, 45 / 255.0f }  }; |

- 구름(cloud, 배경으로 사용)

|  |
| --- |
| #define GROUND\_TOP 0  #define GROUND\_BODY 1  #define GROUND\_BOTTOM 2  GLfloat cloud\_top[4][2] = { { -10.0, 20.0 },{ 10.0, 20.0 },{10.0, 10.0 },{ -10.0, 10.0 } };  GLfloat cloud\_body[4][2] = { { -20.0, 10.0 },{ 20.0, 10.0 },{ 20.0, -5.0 },{ -20.0, -5.0} };  GLfloat cloud\_bottom[4][2] = { { -10.0, -5.0 },{ 10.0, -5.0 },{ 10.0, -15.0 },{ -10.0, -15.0} };  GLfloat cloud\_color[3][3] = {  { 207 / 255.0f, 207 / 255.0f, 207 / 255.0f },  { 215 / 255.0f, 215 / 255.0f, 215 / 255.0f },  { 207 / 255.0f, 207 / 255.0f, 207 / 255.0f }  }; |

2. 요구사항(창의적인 동적 효과 생성)에 대한 각각의 scene(빨간색 글씨는 기능에 관한 설명)

|  |  |  |
| --- | --- | --- |
| Scene1 | | 왼쪽과 같은 화면에서 자동차가 오른쪽에서 왼쪽으로 오게 된다. 이 때 사인함수를 통하여 실제 차가 가는 것처럼, 자동차가 약간 떨리면서 가게 된다.(자동차에 대한 translate 구현) |
| 자동차 : 오른쪽 화면 끝에서 왼쪽으로 오게 되는데, sceneframe이 timer함수에 의해 1씩 증가되게 되고, 이에 –를 붙여서 translate함수의 x좌표 이동에 사용하여 시간이 흐를수록 오른쪽에서 왼쪽으로 오게 되며, y좌표의 이동에 sinf를 이용하여 차의 미세한 떨림을 구현하였다. | | |
| Scene2 | | Scene1에서 왼쪽으로 가던 자동차가 특정 위치(왼쪽 밑의 한 지점)에 멈추어 서게 되면, 왼쪽 그림과 같이 하늘이 어둡게 변하여 밤이 되고, 비행기가 왼쪽에서 오른쪽으로 지나가게 된다(translate). 이때 비행기는 왼쪽화면에서 오른쪽화면으로 이동하면서 점점 작아지게 된다(scale). 비행기의 방향은, 비행기가 현재 가고 있는 방향(비행기에 적용된 함수의 그래프의 기울기)과 같은 방향을 가리키도록 변화된다(rotate). |
| 비행기: sceneframe이 timer함수에 의하여 1씩 증가하게 되는데, 이를 translate함수의 x좌표 이동에 사용하여 시간이 흐를수록 왼쪽에서 오른쪽으로 지나가게 되며, y좌표 이동에 powf함수와, sinf함수를 각각 이용하여 3대의 비행기가 각기 다른 이동경로를 보이도록 하였다. 또한, 해당 비행기가 이동경로를 나타내는 그래프를 미분하여 그래프의 기울기를 구한 뒤, atanf를 이용해 rotate시켜 비행기가 가리키는 방향이 비행기가 현재 위치에서 가는 방향과 같도록 하였다. 또한 비행기의 크기는 scale의 x방향, y방향 모두 1-(float)sceneframe/350으로 적용하여 sceneframe이 증가함에 따라 비행기의 크기가 점점 축소되도록 하였다. | | |
| Scene3 | | Scene2의 비행기가 지나가 화면 밖으로 사라진 이후에, 오른쪽에서 4개의 칼이 날라와 집에 꽂히게 된다(translate).이 때 칼의 방향은 각각의 칼이 가게 되는 방향을 가리키게 된다.(rotate). |
| 칼을 관리하는 clock인 swordclock이 점점 감소하게 되고 이를 translate의 x좌표 이동에 사용하여 시간이 흐를수록 칼이 오른쪽에서 왼쪽으로 날라오게 하였다. 그리고 이에 따른 y좌표 이동에는 sqrtf을 이용해, 루트 x 와 같은 그래프 모양으로 날라오도록 하였다. 또한, 칼이 이동하는 이동경로를 나타내는 그래프의 식을 미분하여 그래프의 기울기를 구한 뒤 atanf를 이용해 rotate시켜 칼이 가리키는 방향이 칼이 현재 위치에서 가는 방향과 같도록 하였다. | | |
| Scene4 | | 칼들이 집에 꽂히는 순간, 오른쪽 3개의 집들은 각자 정해진 함수에 따라 날라가게 되고(translate). 이때 가장 오른쪽에 있는 집은 원을 그리면서 날라가게 된다(rotate). 또한 왼쪽에 있는 집은 4개로 쪼개져서 점점 작아지면서 사라지게 된다(Scale). |
| 가장 왼쪽에 있는 집: Scene3의 가장 왼쪽 집의 위치에 4개의 집을 한 번에 그려주었는데 4개의 집들은 각각 sceneframe을 x좌표이동에 +, -하고, y좌표이동에 +, -하여 원래 한 개의 집이 4개의 집으로 상, 하, 좌, 우로 쪼개지는 것처럼 보이게 하였다. 또한 scale에 cosf(0~90도)을 이용하여 4개의 집들이 점점 작아지도록 하였다. 그리고 rotate에 sceneframe을 이용하여 집이 회전하면서 사라지는 것처럼 보이게 하였다.  가장 오른쪽에 있는 집 : translate한 이후에 sceneframe에 따라 rotate하게 하여 집이 원을 그릴 수 있도록 하였고, 원을 그려지는 시작점을 원래 가장 오른쪽 집이 있던 위치에 있도록 다시 translate하여 원을 그리면서 화면 밖으로 사라질 수 있도록 하였다. | | |
| Scene5 | | Scene4에서 집들이 다 날라가서 화면에서 사라지게 되면, 다시 하늘이 하늘색으로 바뀌고, 차가 원래 있던 왼쪽지점에서, 차가 중앙으로 이동하게 된다. |
| 자동차 : 왼쪽에서 중앙으로 오게 되는데, sceneframe이 timer함수에 의해 1씩 증가하므로 이를 translate함수의 x좌표 이동에 사용하여 시간이 흐를수록 왼쪽에서 중앙으로 오게 되며, y좌표의 이동에 sinf를 이용하여 차의 미세한 떨림을 구현하였다. | | |
| Scene6 | | 차가 중앙에 멈추게 되면, 두 개의 모자가 왼쪽에서 오른쪽으로 지나가게 되는데, 모자들은, 왼쪽에서 특정 그래프 모양처럼 오다가, 화면의 중간쯤에서 롤러코스터를 타듯이 원을 한 바퀴 그린 후에, 오른쪽으로 특정 그래프 모양처럼 지나가게 되는데, 지나가면서 계속 크기변화 하게 된다(translate, scale, rotate). 이때 마우스 왼쪽 버튼으로 윈도우 창의 한 부분을 누르고 있게 되면 움직이던 모자는 마우스를 눌렀을 때의 위치에 멈춰있게 된다. 그리고 왼쪽 버튼에서 손을 다시 떼게 되면 모자가 원래대로 다시 움직이게 된다. 또 화면 하단에 보면, 화분, 집, 비행기 등이 특정 위치에 위치하여 계속 회전하고 있게 된다(translate, rotate). 또한 칼들의 경우에는 차 위에서 하트를 그리고 있는데 이때 마우스 오른쪽 버튼으로 윈도우 창의 한 부분을 누르고 있게 되면 칼들은 움직이던 방향의 반대방향으로 움직이고 오른쪽 버튼에서 손을 다시 떼게 되면 칼들은 원래 움직이던 방향으로 다시 움직이게 되며 나중에 이 칼들은 차를 따라다니게 된다. (translate) |
| 모자 : 모자의 이동은 hat\_flag를 설정하여 3개 단계로 나누어서 구현하였다. 모자는 hat\_timer에 의해 움직이게 되는데 hat\_timer는 timer 함수에 의해 1씩 증가되게 된다. hat\_flag가 0일 때는 모자가 왼쪽에서 중앙으로 이동하게 된다. 이때 hat\_timer를 translate의 x좌표 이동에 사용하게 되고, y좌표 이동에는 powf를 이용하여 두 개의 모자가 각각 x^2그래프와 –x^2그래프와 같은 형태로 왼쪽에서 중앙으로 이동하도록 하였다. 그리고 모자가 중앙에 위치하게 되었을 때 hat\_flag=1, hat\_timer를 0으로 바꾸어 주게 된다. hat\_flag가 1일 때에는 모자가 롤러코스터를 타듯이 원을 한 바퀴 그리게 되는데, 여기서는 모자를 먼저 y좌표 -100만큼 translate한 이후에 hat\_timer에 따라 360도 rotate하게 하여 모자가 원을 그릴 수 있도록 하였고, 원을 그려지는 시작점이 hat\_flag가 0일 때의 마지막 좌표에 있도록 translate하여 이어지는 것처럼 보이도록 하였다. 그리고 원을 그린 이후에는, hat\_timer=0, hat\_flag=2로 바꾸어 hat\_timer가 증가함에 따라 translate을 이용하여 중앙에서 오른쪽으로 빠져나갈 수 있도록 하였다. 그리고 오른쪽으로 빠져나간 이후에는 hat\_timer=0, hat\_flag=0으로 바꾸어 이를 계속 반복할 수 있도록 하였다. 그리고 이 모든 경우에서, scale로 abs(cosf())을 이용하여, 모자가 커졌다, 작아졌다를 반복할 수 있도록 하였다. abs를 붙여준 이유는 모자가 뒤집히지 않도록 하기 위해서이다. 또한 마우스 왼쪽버튼을 눌렀을 때 hat\_timer를 1씩 감소 시켜줌으로써, hat\_timer가 시간에 따라 변화하지 않고 유지되도록 해주어, hat\_timer에 따른 translate에 의해 모자가 멈출 수 있게 하였다.  칼 : 칼은 sword\_timer에 의하여 움직이게 되는데, sword\_timer는 timer함수에 의해 1씩 증가되게 된다. 따라서 이 sword\_timer에 따라 theta가 증가하게 하고 16sin3t와, 13cost-5cos2t-2cos3t-cos4t를 사용하여 하트모양을 그릴 수 있게 하였다. 이 때 translate에 차의 x좌표를 더해줌으로써 칼들이 차와 함께 움직일 수 있도록 하였다. 그리고 마우스 오른쪽 버튼을 눌렀을 때 sword\_timer에 –를 붙여 theta가 -값으로 변하게 하므로써 칼들이 반대방향으로 하트를 그릴 수 있도록 하였다.  집, 화분, 비행기: 랜덤한 특정 위치에서 house\_timer, flower\_timer, airplane\_timer에 따라 rotate하고 있게 된다. 이때 scale함수에 각각의 housescale, flowerscale, airplanescale들이 각각의 timer들에 따라 랜덤하게 생성되어 들어감에 따라 새로 생성될 때마다 크기가 바뀌게 된다. | | |
| (차로 비행기를 쳤을 때 비행기가 날라가는 모습)    (비행기만 계속 쳤을 때, 차의 크기 작아짐)    (화분만 계속 쳤을 때, 차의 크기가 커짐)    (집을 쳤을 때에, 초기 차의 크기로 바뀜) | 6번 scene 상태에서, 키보드로 ‘a’ 또는 ‘A’키를 누르게 되면 차가 왼쪽으로 이동하게 되고, ‘d’ 또는 ‘D’키를 누르게 되면 차가 오른쪽으로 이동하게 된다. 그리고 차가 움직일 때 칼들도 하트를 그리면서 같이 움직이게 된다. 이때 차를 이동시키면서, 회전하고 있는 물체를 치게 되면, 물체들이 물체들마다 정해진 그래프의 형태대로 회전하면서 날라가게 된다. 이때 화분의 경우에는 찌그러지면서 커지다가 사라지게 되고(rotate, scale), 집의 경우에는 4개로 쪼개져서 사라지게 되며(translate, rotate, scale) 비행기는 하늘로 날라가 버리게 된다(translate, rotate). 날라간 물체가 사라지게 되면, 도로 위의 랜덤한 위치에 사라진 물체가 새로운 크기로 나타나 다시 회전하게 된다. 또한 만약에 차가 화분을 치게 되면, 차의 크기가 원래 차의 크기보다 점점 커지게 되고, 비행기를 치게 되면 원래 크기보다 점점 작아지게 된다. 또한 집을 치게 되면 차의 크기가 초기 크기의 차로 돌아오게 된다. 즉 화분만 계속해서 치게 되면 차가 점점 커지다가 어느 정도 커졌을 때 차의 크기를 유지하고, 비행기를 계속해서 치게 되면 차가 점점 작아지다가 어느 정도 작아졌을 때 차의 크기를 유지하며, 집을 치게 되면 맨 처음의 차의 크기로 바뀌게 된다(translate, scale, rotate). | |
| 키보드로 ‘a’, ‘A’를 눌렀을 때 carx를 2씩 감소시켜, ‘d’, ‘D’를 눌렀을 때 carx를 2씩 증가시켜, 이를 translate에 이용하여 차가 키보드에 따라 움직일 수 있도록 하였다. 또 차의 x좌표인 carx와 회전하고 있는 비행기의 x좌표가 일치하였을 때, 현재 차의 가고 있는 방향을 나타내는 carflag에 따라 비행기의 flag를 바꾸어 주었고 각각의 flag에서 비행기에 대한 translate, scale, rotate를 다르게 설정하여 서로 다른 모션을 취할 수 있도록 하여 주었다.(왼쪽에서 물체를 치면 물체가 오른쪽으로 날라감, 오른쪽에서 물체를 치면 물체가 왼쪽으로 날라감) 차가 화분을 쳤을 때에는 carscale을 0.1씩 증가시켜주고 이를 draw\_car할 때 scale에 적용시킴으로써 화분을 쳤을 때 차가 점점 커질 수 있도록 하였고, 비행기를 쳤을 때에는 0.1씩 감소시켜 줌으로써 차가 점점 작아질 수 있도록 하였다. 마지막으로 집을 쳤을 때에는 carscale=1로 바꿈으로써 최초의 차의 크기로 나타날 수 있도록 하였다. 또 회전하고 있다가 날라가게 될 때 각각의 timer변수가 0으로 바뀌기 때문에 이전의 회전 각도를 저장해주기 위해 last~angle이라는 변수를 도입하여, 이전 물체의 마지막 회전각도를 저장해 last~angle+날라갈 때의 회전각도를 rotate에 넣어 모션이 자연스럽도록 하였다. | | |

3. 전체기능 요약

(1)Scene 6에서 마우스 왼쪽 버튼을 누르고 있으면, 움직이던 모자가 멈추게 된다. 그리고 마우스 왼쪽 버튼에서 손을 떼면, 멈춰있던 모자가 다시 움직이게 된다.

(2)Scene 6에서 마우스 오른쪽 버튼을 누르고 있으면, 칼들이 하트를 반대방향으로 그리게 된다. 그리고 마우스 오른쪽 버튼에서 손을 떼면, 칼들이 원래 방향으로 하트를 그리게 된다.

(3)키보드 A(a)키를 누르면 차가 왼쪽으로 움직이게 되고, 키보드 D(d)키를 누르면 차가 오른쪽으로 움직이게 되며 위의 칼들도 같이 하트를 그리면서 움직이게 된다. 이 때 차가 움직이면서 물체들을 치면 물체들이 날라가거나, 찌그러지거나, 쪼개지면서 사라지게 되고 사라진 이후에는 랜덤한 위치에 랜덤한 크기로 나타나게 된다. 그리고 차가 비행기를 치게 되면 차는 점점 작아지게 되고, 차가 화분을 치게 되면 차는 점점 커지게 된다. 차가 집을 치게 되면 가장 초기의 크기로 돌아오게 된다.