과목 : 그래픽스

1차 프로젝트

**학번 : 20151616**

**이름 : 최승환**

!!!README를 읽기에 앞서 게임을 플레이하기 위해 필요한 사항들!!!

게임은 마우스 좌측 클릭을 해야 시작됩니다.

플레이어가 조종하는 우주선은 방향키 좌측 우측에 따라 조금씩 방향을 바꾸며 앞으로 나아가며, 멈추거나 속도를 바꿀 수 없습니다.

맵 밖으로 나가거나, 차에 충돌하거나, 집에 충돌할 경우 게임은 종료됩니다!

다른 물체들은 부딪혀도 영향이 없으니, 집, 차, 벽. 이 세가지만 조심해서 플레이해주세요!

게임 시작 전에도 우주선의 방향은 바꿀 수 있습니다. 유리한 출발을 위해서는 안전한 방향으로 회전시킨 후 시작해주세요!

짧게 플레이할 수 있도록 제작했기 때문에 게임 진행도가 매우 빠릅니다. 시작하자마자 화살표를 조작할 수 있도록 키보드에 손을 얹고 시작해주세요!  
  
-----------------------------------------------------------------------------------------------------------------  
1.기본구현사항

1. 다섯 개의 기본 물체 사용

기본 물체 다섯 가지

cake - 제자리 등속 회전

car - x축 진동운동

airplane - 유저 입력 기반

cocktail - 칵테일 섞기

hat - 탄성운동

위 다섯개를 사용해 동적 표현을 했으며, 동적표현의 설명은 위와 같습니다.

이 중에서 car과 airplane은 특수효과를 부여했습니다. 특수효과에 관한 부분은 2. 추가점수구현에서 설명하겠습니다.

cake

등속도 회전운동을 합니다. frame 별로 정해진 고정값만큼 각도를 변화시킨 후, 이 각도에 맞게 rotate 함수를 통해 다시 출력합니다.

car

x축 진동운동을 합니다. x축을 따라서 초기 속도를 가진 가속도운동을 하는데, 중심점에 도달한 순간부터 반대방향으로 가속도를 받아 점차 속도가 느려지다가 반대쪽으로 움직이는 형태의 운동을 합니다.

airplane

유저 입력으로 좌측 화살표와 우측 화살표 두 종류를 받습니다. 좌측 화살표를 입력받으면 시계방향으로, 우측 화살표를 입력받으면 반시계방향으로 정해진만큼 회전합니다. 회전하는 정도는 SENSITIVITY 를 참조해 변화시킵니다. 이 방향값이 향하는 방향으로 물체는 일정한 속도(PAWNSPEED)로 나아가며, airplane의 이미지 또한 그에 맞는 각도로 rotate 합니다.

cocktail

칵테일은 칵테일을 섞는 듯한 운동을 합니다. x축과 y축 방향으로 일정한 속도로 나아가다가, 특정 위치에 도달하면 반대방향으로 움직입니다. 이 과정이 반복되면, 궤도는 일정하지 않지만, 특정 구간을 두고 도는 형태의 불규칙적인 운동을 하게 됩니다.

hat

중력을 받는 물체와 비슷한 운동을 합니다. +y축 방향의 초기속도를 가지고 움직이며, 반대 방향으로 가속을(중력가속도와 유사) 받아 점차 느려지며 나중에는 낙하합니다. 바닥에 닿는 순간 다시 튀어 오르며, 탄성 운동을 합니다.

1. 한 개의 새로운 물체

alien - rotate+scale 주기 운동

alien 은 특정 주기를 가지고 운동합니다. 게임 시작 시, 첫번째 주기의 운동을 시작합니다. scale 이 커지며, rotate 를 시작하는데, 특정 크기에 도달한 뒤, 원래와 같은 각도(0도 또는 360도)에 도달하면, 축소되기 시작합니다.

두번째 주기에서는 회전하지 않고, 크기만 점차 줄어듭니다. scale 이 줄어들다가 0에 도달하게 되면, 다시 크기가 커지며 회전하는 첫번째 주기의 운동을 시작합니다.

위 두 주기를 번갈아가며 운동합니다.

2. 추가구현사항

추가구현사항에 대해서는 glutTimerFunc, 충돌감지, 장외판정과 창 조정, 게임 시작 및 종료 로 총 4종류의 기능이 있습니다.

각 기능에 관해 항목별로 기능의 내용, 기능의 구현방법 을 간단하게 설명하겠습니다.

* glutTimerFunc
  + glutTimerFunc의 기능 및 필요이유
    - glutTimerFunc 는 단위시간마다 반복적으로 함수를 호출하는 함수이며, callback 함수의 한 종류입니다. 넘겨주는 인수는 다음과 같습니다.
      * unsigned int time : 반복적으로 함수를 호출하는 단위시간
      * void (\*callback)(int) : callback 함수 지정
      * int value : callback 함수에 넘겨줄 int 값  
         (주로 state를 알리거나 flag 용으로 쓰임)
* 충돌감지
  + 고정된 물체와의 충돌
    - 게임 내에는 고정된 물체의 집들이 존재합니다. 만약 이 집들과 충돌할 경우, 게임이 종료됩니다.
    - 고정된 물체와 충돌하는지를 확인하기 위해서는 고정된 물체들의 좌표를 알아야 합니다. 이를 위해서 총 9개의 좌표를 배열로 저장했습니다. 배열은 obstacle[9][2] 입니다.  
      obstacle 을 순차적으로 참조하여, 해당 좌표에서 일정 거리 이상 떨어져 있는지를 확인해야 합니다. 이를 위해 boundary 라는 변수를 선언해, boundary 안으로 들어올 경우 게임을 종료하도록 합니다.  
      obstacle의 좌표에 충돌감지가 된다고 해도, 그림이 없으면 이를 볼 수가 없습니다. 따라서 obstacle의 좌표에 따라 display 함수에서 이미지를 그려줍니다.
  + 움직이는 물체와의 충돌
    - 게임 내에는 위의 설명에 등장한, x축 진동운동을 하는 자동차가 운동을 합니다. 이 자동차에 치이면(일정 거리 이하로 들어가면) 게임지 종료됩니다.
    - 움직이는 물체와 충돌하는지를 확인하기 위해서는 물체가 움직일 때마다 그 좌표를 알고 있어야 합니다. 위에서 설명한 glutTimerFunc 를 사용하면, 몇 ms 마다 한번씩 함수를 호출할 수 있는데, 이 때마다 물체가 다음 움직일 좌표를 계산합니다.(추후에 이를 display 함수에서 출력합니다.) 이렇게 좌표가 계산된 직후에 해당 좌표가 airplane의 좌표 centerx, centery 와 부딪히지 않았는지 확인해준 후,(동일하게 boundary를 사용합니다) 부딪혔다면 게임을 종료해줍니다.
* 장외판정과 창 조정
  + 장외 판정의 기준
    - 장외판정은 창 바깥으로 나가려고 했을 때 장외판정 처리됩니다. 비행기가 창 경계에 충돌할 경우 게임은 종료됩니다.
    - 장외판정을 체크하기 위해서는 centerx, centery를 조사해야 합니다. 이 값들은 비행기의 다음 프레임에서의 좌표입니다. 이 값이 창 바깥으로 조정되어 있으면, 장외판정 처리됩니다. 이 때, 비행기의 중심좌표만 고려하면, 비행기의 일부가 창 밖으로 나가도 처리가 되지 않아 직관적으로 이해가 힘들기 때문에, 창 너비의 0.975, 창 높이의 0.95만큼 을 한계로 설정했습니다. 창 너비 적용비율이 조금 더 적은 이유는 창의 가로 비율이 대체로 더 크기 때문입니다.
  + 창 조정과 장외판정
    - 장외판정을 고정값으로 할 경우, 여러가지 문제가 발생하게 됩니다. 그 중 하나가 장외판정의 기준이 달라진다는 것입니다. 예를 들어 x 좌표가 400 을 넘어가면 장외가 되도록 설정했다고 해도, 창 크기가 800보다 작아지면, 비행기가 보이지 않는 곳까지 창을 넘어가도 게임이 종료되지 않는 현상이 발생합니다. 이 때문에, win\_height, win\_width 를 이용해, 장외판정 기준을 정합니다. win\_height와 win\_width 는 glutInitWindowSize(int , int ) 에서 초기값을 가지게 되는데, 이 때 두 int 값이 각각 가로 세로 값을 가지게 됩니다. 창의 크기에 비례해야 하는 경우는 모두 이 변수를 이용해 상대적으로 정하도록 하면 창의 크기가 바뀌어도 앞서 말한 문제가 발생하지 않습니다.
* 게임 시작 및 종료
  + 게임 시작 방법
    - 프로그램을 실행시키면, 게임이 바로 시작하지 않고, 멈춰있는 화면이 나옵니다. 마우스를 클릭한 순간부터 플레이가 시작되고, 마우스를 클릭하기 전에도 우주선 회전은 가능합니다.
    - 마우스를 클릭하면 시작하도록 하는 방법 또한, glutTimerFunc 과 관련이 있습니다. glutTimerFunc 가 호출되자마자, startflag 라는 변수를 참조하게 됩니다. startflag는 bool 형의 변수인데, 이 변수가 false 인 경우, glutTimerFunc()(실제로는 tick callback 함수 사용) 는 다음 프레임에 해야할 연산들을 시행하지 않고, 주기적으로 호출만 시행합니다. 이 과정에서 화면을 유지하기 위해서 glutPostRedisplay 함수가 계속 호출되는데, 이 함수 내에 마우스가 눌리면 값이 1로 조정되는 leftbuttonpressed라는 변수를 검사합니다. 마우스가 눌렸다면 이 때, startflag를 1로 변화시키고, 그 이후부터 호출되는 callback tick 함수에서는 다음 프레임을 위한 연산을 시작합니다.
  + 게임 종료 조건
    - 게임 종료는 앞서 말했듯이 집, 차, 벽 에 부딪힐 경우 시행합니다.
    - 게임 종료 조건은 충돌방지와 장외판정 부분에서 설명했던 것과 동일합니다. 이 경우 glutLeaveMainLoop() 를 호출합니다. 이를 호출하면, 게임이 종료되며, 프로그램 또한 종료됩니다.