

# **기초 컴퓨터 그래픽스 - 프로그래밍 숙제 1**

**OpenGL API 함수를 사용한 2 차원 기하 변환 연습**

**서강대학교 컴퓨터공학과  
CSE 4170 기초컴퓨터그래픽스**

**담당교수: 임인성**

**제출일: 2019 년 4 월 17 일**

**국어국문학전공 엄태경 20120085**

## 1. 프로젝트 목적

본 프로젝트의 목적은 OpenGL API 함수를 이용해 다양한 2차원 아핀 변환을 구현하고, Modeling transformation 을 이해하는 것이다.

## 2. 프로젝트 개요

다양한 2차원 아핀 변환을 적용해 게임 요소가 있는 프로그램을 제작한다. 직접 설계한 2차원 기하 물체(다각형)를 추가하고, 재미있고 복잡도가 있는 기하 변환 및 애니메이션 효과를 적용한다.

## 3. 실행

방법 1) Visual Studio 2017 에서 파일 "HW1\_S20120085.sln"을 열고, Solution Configuration 은 "Release", Solution Platform 은 "x64"로 두고 빌드한다. "디버깅하지 않고 시작(Ctrl+F5)" 기능을 이용해 프로그램을 시작한다.

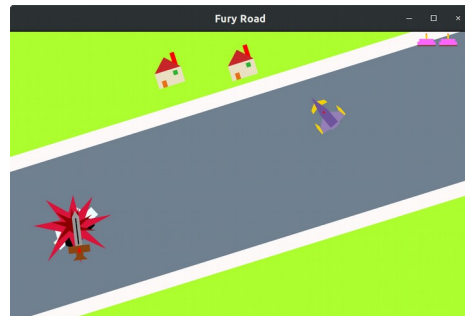
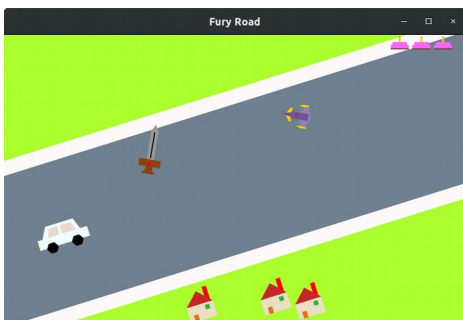
방법 2) HW1\_S120085/HW1\_S120085/HW1\_S120085.exe 파일을 실행한다.

## 4. 구성 요소

- 모든 오브젝트는 윈도우 크기 변화에 따라 스케일링한다.
- 애니메이션 효과가 있는 오브젝트는 자동차 (car), 자동차 2 (car2), 칼 (sword), 비행기 (airplane), 집 (house), 충돌효과 (boom)이다.
- 케이크(cake)는 애니메이션 효과는 없으나 프로그램 진행에 따라 개수가 변화하는 동적 요소가 있다.
- 직접 설계하여 추가한 오브젝트는 도로 (road), 충돌효과 (boom)이다.
- 충돌효과 (boom)와 자동차 (car)는 특정 조건을 만족해야 화면에 표시되며, 나머지 오브젝트는 처음부터 표시된다.
- 화면은 초당 30 프레임이 표시되며, 각 동적 효과는 매 프레임마다 계산된다.
- 자동차가 오른쪽으로 전진하는 효과를 주기 위하여, 칼 (sword)과 집 (house)은 매 프레임마다 변수 car\_speed 에 따라 왼쪽으로 일정한 정도를 이동한다.

### 1) 구성 화면

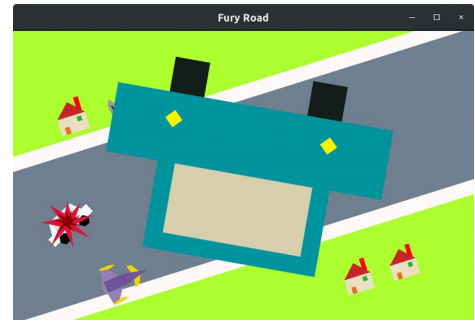
- 초기 화면
  - 처음 프로그램을 실행하면 보이는 상태.
  - 도로를 배경으로 각 물체가 배치되어있다.
  - 우측 상단에 케이크가 표시되며, 집, 자동차 2, 칼, 비행기가 각각 움직이고 있다.
- 충돌 화면
  - 칼과 자동차 2가 충돌하면 보이는 상태.
  - 두 물체 사이에 충돌효과가 표시되고, 우측 상단에 케이크가 하나 줄어든다.



- 자동차 2가 시계방향으로 조금 회전하여 충돌한 효과가 발생한다.

- 종료 화면.

- 자동차 2와 칼이 3차례 충돌하여 케이크가 모두 없어지면 보이는 상태
- 자동차 2와 충돌효과, 집은 그대로 정지하고 비행기와 칼만 움직인다.
- 화면 중앙에 자동차가 표시되어 회전하면서 점점 커진다.
- 자동차가 화면 전체를 덮을만큼 커지면 회전을 중지한다.



## 2) 오브젝트

화면에 표시되는 순서에 따라 아래에서 위로 나열한다. 각 캡처 이미지의 회색 테두리는 오브젝트 이미지와 흰 배경을 구분하기 위한 테두리이며, 오브젝트에 포함되지 않는다.

- 도로 (road)

- 동적인 요소 없이 윈도우 대각선에 평행하게 회전한 상태로 배치되어 배경 역할을 한다.



- 집 (house)

- 도로의 양쪽 바깥으로 2개씩, 총 4개가 화면에 배치된다. 각 집의 위치는 도로 바깥 영역에 임의로 배치된다.
- 자동차가 오른쪽으로 전진하는 효과를 주기 위하여 변수 car\_speed에 따라 매 프레임마다 왼쪽으로 이동한다.
- 왼쪽으로 이동하는 집이 윈도우 영역을 벗어나면, 윈도우의 오른쪽 바깥 영역에서 다시 랜덤하게 생성되어 왼쪽으로 이동한다.
- 집이 자동차를 향해 다가오는 원근 효과를 주기 위하여, 매 프레임마다 조금씩 확대된다.



- 자동차 2 (car2)

- 윈도우 대각선에 평행하게 회전한 상태로 배치된다.
- 60 프레임 주기로 좌우로 진동한다.
- 상하 화살표 입력에 따라 위아래로 움직인다.
- 칼과 충돌하면 시계방향으로 조금 회전하여 충돌하는 효과를 보인다.



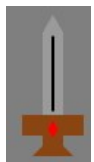
- 충돌효과 (boom)

- 16각형 두 개를 겹쳐놓은 모양으로, 칼이 자동차 2에 닿으면 자동차 2의 바론 위에 표시된다.
- 일정한 범위 내에서 커졌다가 작아지기를 반복하여 칼이 자동차에 충돌하는 듯한 효과를 준다.

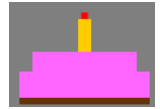


- 칼 (sword)

- 윈도우 대각선에 수직하게 회전한 상태로 배치된다.
- 매 프레임마다 좌우로 움직이며 전진하며, 움직이는 방향으로 칼 끝이 돌아간다. 움직이는 방향과 정도는 랜덤하게 결정된다.
- 이동하는 칼이 윈도우 영역을 벗어나면, 다시 랜덤한 위치에 생성되어 윈도우의 아래쪽 바깥에서 위로 올라온다.
- 자동차가 오른쪽으로 전진하는 효과를 주기 위하여 변수 car\_speed에 따라 매 프레임마다 왼쪽으로 이동한다.



- 케이크 (cake)
  - 윈도우의 오른쪽 상단에 표시되며, 프로그램이 종료될 때까지 움직임 없이 고정되어있으나 물체의 개수는 점점 줄어든다.
  - 최초에 3 개로 시작하여 칼이 자동차 2 (car2)에 닿을 때마다 하나씩 줄어든다.



- 비행기 (airplane)
  - 시야를 방해하기 위한 오브젝트로, 랜덤한 위치에 생성되어 점점 확대되면서 같은 자리에서 원을 그리며 움직인다.
  - 원을 그리며 움직이는 속도와 그리는 원의 크기도 매 프레임마다 빨라지고 커진다.



- 자동차 (car)
  - 케이크가 모두 없어지면 윈도우 중앙에 등장한다.
  - 점점 커지면서 회전하다가, 화면 전체를 채울 정도로 커지면 멈춘다.



### 3) 키보드 입력

키보드 입력은 'p'/'P', ESC 키, 그리고 상하좌우 화살표에 반응한다.

- 'p'/'P'
  - 화면에 등장하는 모든 오브젝트의 움직임이 멈춘다.
  - 다시 한번 입력하면 원래대로 오브젝트가 움직인다.
  - 케이크가 모두 없어져서 윈도우 중앙에서 자동차 (car)가 회전하고 있는 상태에서는 입력해도 반응하지 않는다.
- ESC 키
  - 프로그램을 종료한다.
- 상하 화살표
  - 자동차 2 (car2)를 윈도우 대각선에 수직한 방향으로 움직인다.
- 좌우 화살표
  - 집과 칼이 왼쪽 아래로 이동하는 속도를 조절한다.