CSED451 Computer Graphics

Assignment #2

2D Animation

알파고알파고

20140908 컴퓨터공학과 박경원(pkw1012)

20140735 컴퓨터공학과 백우정(pcsewj14)

1. 프로그램 개요

이번 과제는 Assignment 1 에서 구현했던 것을 세 가지 면에서 좀 더 발전시키는 것이었다. 첫째는 수업시간에 배웠던 shader programming을 이용하여 pipe line을 직접 건드려 그림을 출력하는 것, 둘째로는 캐릭터를 계층 구조를 이용해 관절 움직임을 구현하는 것, 그리고 마지막으로는 이 모든 것을 씬 그래프(Scene Graph)를 이용해 디자인 하는 것이다.

1. 프로그래밍 환경

Visual Studio 2015를 이용하였다.

1. 프로그램 설계 및 구현
2. Type 정의

이번 assignment에서는 씬 그래프를 이용해야했기 때문에 한 cpp파일에 모든 것을 구현했던 assignment 1번과 달리 class를 세부적으로 나누어 type을 구현하였다.

구현한 class는 다음과 같다.

class GAME : 게임 전체를 관리

class MAP : 게임의 맵, 즉 들판, 도로, 강을 다룬다.

class ROAD : 도로 한 줄과, 그 위의 자동차를 다룬다.

class RIVER : 강 한 줄과, 그 위의 통나무를 다룬다.

class FIELD : 들판 한 줄과, 그 위의 나무를 다룬다.

class CAR : 자동차에 대한 함수를 구현했다.

class LOG : 통나무에 대한 함수를 구현했다.

class TREE : 나무에 대한 함수를 구현했다.

class PLAYER : 사용자가 조작하는 캐릭터를 관리한다.

1. 함수

모든 class에 기본적으로 들어가 있는 함수는 init(\_), buffer(), draw(\_, \_) 세 가지이다.

1. void init(\_)

object의 처음 좌표를 지정하고, 이를 shader program에 넘겨준다.

1. void buffer()

vertex array에 각 object에 대한 buffer를 만들어 준다.

1. void draw(GLint pro, GLint model)

object를 출력한다. 필요할 경우 model\_veiw matrix를 변형시켜 object를 움직인다.

1. void move()

animation이 들어가는 object의 경우, 이 함수를 사용하여 변화량을 바꾼다. 좌표값을 바꾸는 것이 아니라, △x, 혹은 △y 값을 바꾸었다. class CAR, class LOG에서 사용하였으며 이를 위 쪽 class에서 호출한다.

1. void reInit()

자동차나 통나무가 무한히 나오는 것처럼 보이게 하기 위해서 일정 지점에 도달하면 좌표를 큰 폭으로 수정해야 했다. 마치 원을 무한히 도는 것과 같은데, 그것이 이 함수에 구현되어있다. move()에서 호출된다.

아래는 class GAME에서만 구현된 함수이다.

1. bool finished()

class GAME에 구현되어 있다. 게임 종결 조건을 검사하는 함수이다. 캐릭터가 화면 밖으로 나갔을 때, 자동차와 충돌했을 때, 혹은 강에 빠졌을 때 true를 반환한다. specialkeyboard와 GAME의 move에서 호출된다.

1. bool treehere()

캐릭터가 이동하려는 곳에 나무가 있는지 검사한다. specialkeyboard에서 호출된다.

다음은 assn2.cpp의 함수들이다.

1. void specialkeyboard(int key, int x, int y)

방향키 입력을 받는 함수이다. 캐릭터 조종은 방향키를 이용하도록 구현하였다. 이동할 때마다 finished를 이용해 종결 조건인지, treehere을 이용해 나무가 있는지 검사한다.

1. void phase1(int value)

void phase2(int value)

void phase3(int value)

캐릭터가 오른쪽으로 움직일 때, 걸어가는 애니메이션을 추가했는데, 걸어가는 동작을 만들어주는 함수들이다.

1. int main(int argc, char\*\* argv)

생성된 game object를 이용하여 게임을 실행시키는 함수이다.

다음은 class PLAYER에서 구현된 함수이다.

1. void quad(inti dx, int a, int b, int c, int d)

void colorcube(int index)

플레이어 캐릭터의 몸은 몸통, 머리, 팔, 다리 등의 계층 구조로 되어 있으며 이 각각의 구조를 상자 모양으로 만들었는데, 이 모양을 만들기 위한 함수이다.

1. void traverse(Node\* node)

캐릭터를 출력할 때, 트리 형태의 구조를 traverse하여 출력해야하며, 이 때 사용하는 함수이다.

1. void initNodes( )

계층 구조에서 몸통, 머리 등의 각각의 node들의 크기나 형태들을 초기화시켜주는 함수이다.

1. 추가 구현
2. 실제 게임에서는 캐릭터가 가만히 있을 경우 독수리가 날아와서 잡아간다. 이와 비슷하게 캐릭터가 가만히 있다가 화면 밖으로 사라지면 게임은 자동으로 종료되도록 구현하였다.
3. 화면의 위와 아래로는 캐릭터가 이동할 수 없도록 하였다.
4. vec.h에는 ‘==’에 대한 연산이 정의되어있지 않았다. 이를 이용하기 위해 vec.h에 ‘==’에 대한 operator overloading을 정의하여 사용하였다.
5. 프로그램 실행 방법

ctrl + F5를 누르면 된다. 이 때 헤더파일 Angel.h, vec.h, mat.h와 shader program을 불러와주는 InitShader.cpp가 반드시 있어야한다. 이 파일들은

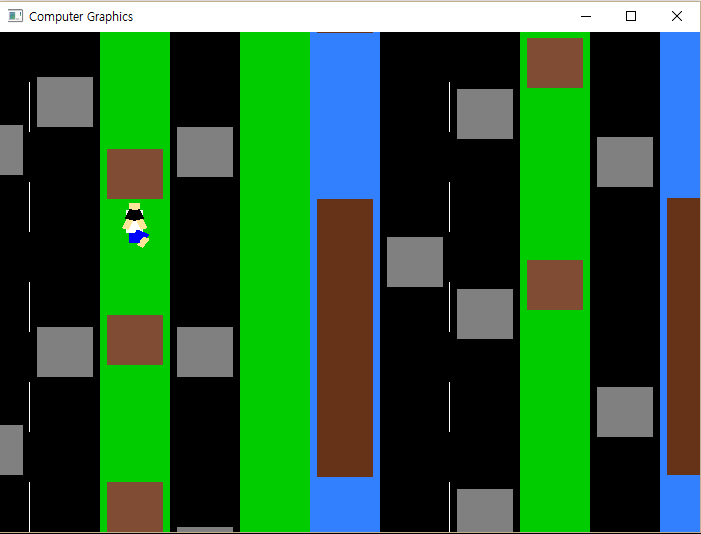
<http://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SIXTH_EDITION/CODE/include/> 에서 얻을 수 있었다.

또한 vertex shader와 fragment shader 역시 있어야 한다.

1. 예제



위는 프로그램을 실행시켰을 때, 나오는 화면이다.



캐릭터를 움직일 경우 위와 같은 애니메이션이 나오면서 오른쪽으로 이동하게 된다.

1. 토론

수업시간에 앉아 이론을 듣는 것과 실제 적용해보는 것에 많은 차이가 나서 힘들었던 과제였다. assignment 1의 경우는 비슷한 예제가 많아서 이해하지 않고도 어느 정도 할 수 있었는데, 이번에는 그런 예제가 없다보니 모두 우리의 손으로 하나하나 짜면서 이해할 수 밖에 없었다. 이 점에서 확실히 shader program, projection matrix, model-view matrix가 어떤 역할을 하는지도 알게 된 것 같다. shader에서 in과 out의 의미, 그리고 uniform하게 적용된다는 것의 의미를 직접 짜보면서 알 수 있었다.

이번 assignment에서 한 파일에 몰려있던 것을 class를 나누어 분류하고 정리했는데, 그 과정에서 shader가 이를 어떻게 받아들이며, translate함수를 썼을 때 object들을 어떻게 쪼개야 내가 원하는 대로 움직일 수 있는지 고민해야했다. 예를 들어 자동차 한 차선은 하나의 배열로 관리되고 있었다. 여기서 자동차가 순환하는 것 처럼 보이려면 자동차 하나하나마다 △y를 다르게 관리해야 한다. 때문에 △y에 대해서도 배열을 만들어 주었으며(cmove[CAR\_N]), 각 자동차에 대해 다른 buffer와 model-view matrix를 사용했다. 이와 비슷하게 다른 object에 대해서도 그들의 움직임에 관해 생각하며 프로그램을 짰다.

캐릭터가 맵이랑 상호작용을 할 때, 예를 들어 자동차와의 충돌을 할 때, 캐릭터의 정보를 밑의 clsss까지 끌고 들어가는 것은 깔끔하지 못하고, 씬 그래프와의 개념과는 맞지 않다고 판단하였다. 때문에 class GAME에서 모든 것을 해결했는데, 자동차나 통나무의 위치를 가져오는 것이 상당히 번거로웠다. 하지만 이 방법이 직관적이고 씬 그래프와 맞다고 생각하여 최대한 간단하게 구현하였다.

1. 결론

이번 assignment에서는 shader의 동작에 대해 수업시간에 듣는 것보다 더욱 자세히 이해할 수 있었고, 씬 그래프의 개념을 사용하여 그래픽 요소들을 보다 정갈하고 간단하게 다룰 수 있었다. 또한 캐릭터를 계층 구조를 사용하여 만들고 이를 이용해 간단한 애니메이션을 만들 수 있었다.

1. 개선 방향

class로 구조를 고치면서 상속을 써야할지 고민을 조금했다. 자동차와 통나무의 경우 거의 똑같았기 때문에 상속을 이용하면 좀 더 간단하게 구현할 수 있었을 것 같다.

buffer의 개수가 프로그램의 성능에 어떤 영향을 주는지는 모르겠지만 buffer를 많이 쓴 것이 마음에 들지 않았다. 배열을 조금 다른 형태로 썼더라면 buffer의 개수가 줄어들 것 같은데, 그걸 고치려면 너무 많은 것을 바꿔야해서 하지는 못했다.

Assignment 1과 같이 좌표에 약간 문제가 있다. 구현을 할 때, 가상의 격자를 만들고 캐릭터가 그 격자로 이동하도록 의도했지만 실제로 실행하면 그 격자를 벗어나 이동하는 경우가 생긴다. 이는 좌표가 int형이 아닌 float형을 써서 생기는 오차와 통나무가 계속 이동하기 때문에 발생하는 오차로 보인다. 이를 해결하기 위해서는 좌표를 int형으로 바꾸던가 캐릭터 이동을 할 때 더 정확하게 움직이도록 구현해야할 것 같다.

1. 참고문헌

<http://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SIXTH_EDITION/CODE/>

Interactive Computer Graphics - A Top-Down Approach 6e By Edward Angel and Dave Shreiner (Pearson, 2012) BBS

수업자료 5번 ppt. 05.shader