

3D게임프로그래밍1

과제 01 설명

제출일: 2022.05.13

전공: 게임공학과

학번: 2019184020

성명: 윤은지

목차

1. 조작법(키)에 대한 설명
2. 플레이어가 움직일 때 마다 바닥에 타일 깔기
3. 플레이어의 시작위치부터 이어지는 레일을 따라 플레이어가 움직이도록 하기
4. 비트맵을 로드 하여 배경설정
5. 플레이어가 롤러코스터를 탈 때에는 카메라를 고정,

롤러코스터가 꺾일 때마다 카메라 좌표 변경

1. 조작법(키)에 대한 설명

w: 앞으로 이동

s: 뒤로 이동

a: 왼쪽으로 이동

d: 오른쪽으로 이동

z: 위로 이동

x: 아래로 이동

ctrl: 총알 발사

상자를 오른쪽 버튼으로 선택 후 ctrl: 조준을 하지 않아도 상자가 총알에 맞아서 폭파됨.

숫자 키 1부터 9까지: 해당되는 상자가 폭파됨.

g: 모든 상자 폭파

p: 게임 종료

1. 플레이어가 움직일 때 마다 바닥에 타일 깔기

텍스트이(가) 표시된 사진

자동 생성된 설명

-Scene.h의 Scene의 private멤버 변수로 타일이 될 객체의 포인터, 타일이 그려질 주기를 저장할 변수, 그려질 타일들의 정보를 저장할 컨테이너 객체를 선언한다.

CGameObject\* mpObj = NULL;

Float mpTime = 0.f;

std::deque<XMFLOAT4X4> mpObjDeq;

앞서 언급한 순서대로이다.

그리고 deque을 사용하기 위해 stdafx.h에다가 #include<deque>을 추가한다.

- Scene.cpp의 void CScene::BuildObjects()에 타일을 만들기 위한 메쉬로 CCubeMesh\* m\_pCubeMesh = nullptr; 이렇게 선언하고, 형태는 m\_pCubeMesh = new CCubeMesh(4.0f, 4.0f, 0.01f); 이렇게 설정한다.

mpObj = new CGameObject;

//그리고 mpObj를 CGameObject로 동적할당 한다.

mpObj->SetMesh(m\_pCubeMesh);

//mpObj의 형태를 m\_pCubeMesh로 설정한다.

mpObj->SetColor(RGB(255, 255, 125));

// mpObj의 색을 RGB(255, 255, 125)으로 설정한다.

void CScene::Animate(float fElapsedTime)에서 0.4f가 지날 때마다 m\_pPlayer의 매트릭스 정보를 얻어 deque에 저장하고 시간 변수를 0.f로 만든다.

mpTime += fElapsedTime;//끊임없이 시간 변수에 fElapsedTime를 더한다.

if (mpTime > 0.4f)

{

mpObjDeq.push\_back(m\_pPlayer->getMat());

//XMFLOAT4X4 getMat()는 CPlayer의 public 멤버함수이고 m\_xmf4x4World를 반환한다.

mpTime = 0.f;

}

void CScene::Render(HDC hDCFrameBuffer, CCamera\* pCamera)에서 deque에 저장된 타일을 그린다.

for (const XMFLOAT4X4& mat : mpObjDeq)// mpObjDeq에 저장된 정보를 mat로 가져온다.

{

mpObj->setMat(mat);//mat의 matrix를 mpObj에 넘겨준다.

//XMFLOAT4X4 setMat(XMFLOAT4X4 xmatrix)는 CGameObject의 public 멤버함수이고 //m\_xmf4x4World에 xmatrix를 대입한다.

mpObj->Render(hDCFrameBuffer, pCamera);//mpObj를 랜더링 한다.

}

void CScene::ReleaseObjects()에서 동적할당한 mpObj의 메모리를 해제한다.

if (mpObj)

delete mpObj;

1. 플레이어의 시작위치부터 이어지는 레일을 따라 플레이어가 움직이도록 하기

- Scene.h에서 CScene의 private 멤버 변수로 레일을 구성할 상자의 개수, 왼쪽 레일, 오른쪽 레일을 차례로 선언한다.

int railNObject = 0; //상자의 개수

CGameObject\*\* railObjectL = nullptr;//왼쪽 레일

CGameObject\*\* railObjectR = nullptr;//오른쪽 레일

CCubeMesh\* railMesh = new CCubeMesh(0.4f, 0.4f, 0.2f);

//레일을 구성할 상자의 모양을 설정하고,

railNObject = 300; //한쪽 레일에 들어갈 상자의 개수를 정하고,

railObjectL = new CGameObject \* [railNObject];

railObjectR = new CGameObject \* [railNObject];

// 왼쪽 레일과 오른쪽 레일을 구성할 상자의 배열을 동적할당한다.

for문을 이용하여 정수 i가 0부터 99까지, 100부터 199까지, 200부터 299까지의 세부분으로 두 번 꺾이는 레일을 설정한다.

for (int i = 0; i < 100; ++i)//2 0 0

{

railObjectL[i] = new CGameObject(); // 게임오브젝트 생성

railObjectL[i]->SetMesh(railMesh); //상자 모양 설정

railObjectL[i]->SetColor(RGB(255, 125, 0)); //레일 색깔 설정

railObjectL[i]->SetPosition(0.0f + i \* 0.4, 0.0f + i \* 0.4, 0.0f + i \* 0.4);

//상자 좌표 설정

//오른쪽 레일

railObjectR[i] = new CGameObject();

railObjectR[i]->SetMesh(railMesh);

railObjectR[i]->SetColor(RGB(255, 125, 0));

railObjectR[i]->SetPosition(0.0f + i \* 0.4 + 4, 0.0f + i \* 0.4, 0.0f + i \* 0.4);

}

void CScene::Animate(float fElapsedTime)에서 레일의 정보를 업데이트 한다.

for (int i = 0; i < railNObject; ++i)

railObjectL[i]->Animate(fElapsedTime);

for (int i = 0; i < railNObject; ++i)

railObjectR[i]->Animate(fElapsedTime);

void CScene::Render(HDC hDCFrameBuffer, CCamera\* pCamera)에서 레일을 그린다.

void CScene::ReleaseObjects()에서 레일의 메모리를 해제한다.

플레이어가 레일 위에 있을 때 방향키 w를 이용한 앞으로의 이동만 가능하다.

그리고 플레이어가 레일의 끝에 도달하면 모든 방향키를 사용할 수 있다.

CPlayer의 public 멤버변수를 다음과 같이 선언한다.

int onRail = 1;

//앞서 언급한 레일은 두 번 꺾이며 세부분으로 이루어져 있는데 어느 부분을 지칭할지 정하는 변수이다.

void CPlayer::Move(XMFLOAT3& xmf3Shift, bool bUpdateVelocity)에서 레일이 꺾이는 부분의 상자의 좌표를 비교해서 onRail 변수의 값을 정한다.

onRail 변수에 레일의 파트1일 때는 1, 파트2일 때는 2, 파트3일 때는 3을 저장한다. 레일을 벗어나면 0을 저장한다.

void CPlayer::Move(DWORD dwDirection, float fDistance) 에서 플레이어가 레일이 그려진 모양대로 움직이게끔 한다.

그리고 레일을 벗어났을 때에는 랩프로젝트02-1에서 움직이던 것과 같게 만든다.

1. 비트맵을 로드 하여 배경설정

void CGameFramework::ClearFrameBuffer(DWORD dwColor)에서

m\_hBitmapFrameBuffer = LoadBitmap(m\_hInstance, MAKEINTRESOURCE(129));

//리소스로 추가한 비트맵을 로드하고

SelectObject(m\_hDCFrameBuffer, m\_hBitmapFrameBuffer);

//bitmap객체를 dc에 할당한다.

void CGameFramework::OnDestroy()에서 dc와 bitmap을 삭제한다.

텍스트, 모니터, 전자기기, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

1. 플레이어가 롤러코스터를 탈 때에는 카메라를 고정,

롤러코스터가 꺾일 때마다 카메라 좌표 변경

void CGameFramework::ProcessInput()에서 비행기가 레일 위에 있을 때 카메라를 고정한다.

if (m\_pPlayer->onRail == 0)

// onRail이 0일 때, 즉 비행기가 레일 위에 있는 상태에서만

//마우스를 이용해 카메라를 움직일 수 있다.

{

if (pKeyBuffer[VK\_RBUTTON] & 0xF0)

m\_pPlayer->Rotate(cyMouseDelta, 0.0f, -cxMouseDelta);

else

m\_pPlayer->Rotate(cyMouseDelta, cxMouseDelta, 0.0f);

}

void CScene::Animate(float fElapsedTime)에서 레일이 꺾일 때마다 흔들리는 느낌을 주기 위해 카메라 좌표를 바꾼다.

// changeRot는 bool형 배열인데 false로 0번 인덱스, 1번 인덱스, 2번 인덱스 모두 초기화 되어있다. 카메라 좌표를 바꾸고 나서 true로 바꾼다. 이렇게 하지 않으면 끊임없이 카메라 좌표가 바뀌어서 화면이 빙빙 돈다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

