**3D게임1 과제02**

폰트, 텍스트, 그래픽, 로고이(가) 표시된 사진

자동 생성된 설명

2022180024 유영빈

**목표**

이전 과제에서 Windows GDI를 이용해 구현했던 게임 혹은 그래픽 프로그램을 Direct3D 12 기반으로 전환하여 렌더링과 확장 가능한 구조를 갖춘 3D 게임 엔진 구조를 구현하는 것이다.  
Direct3D 12의 렌더링 파이프라인, 루트 시그니처 설정, 명령 리스트 처리, 셰이더 변수 관리 등의 핵심 요소들을 직접 구현함으로써 로우레벨 그래픽 API에 대한 이해를 증진시키고, 동시에 실질적인 게임 오브젝트의 생성, 애니메이션 처리, 충돌 판정, 씬 전환, UI 요소 표시 등의 기능을 통합적으로 구성하는 것을 목표로 한다.

(모든 기본 필수 구현은 구현 되어있다) (그 외 세운 목표)

1. 게임 전체 흐름 및 씬 전환을 구현

시작화면->메뉴화면->Level-1->Level-2의 구조로 구성하고, 각 씬 별로 전환이 자연스럽게 되도록 한다.

첫번째 과제에서 문제가 됐던 Level-2 -> Level-1으로 이동할 때

애니메이션이 유지되어 있었던 문제를 해결한다.

1. 시작 화면 기능

OBJ확장자 파일을 사용하는 것이 아닌 직접 오브젝트 위치를 지정하여 이름을 만든다.

위치 기반 클릭 감지가 아닌 마우스 픽킹을 사용하여 오브젝트 클릭 시, 터지도록 한다.

1. 메뉴 선택 기능

첫번째 과제에서 키보드를 사용하여 메뉴를 선택하였지만, 이번에는 마우스 클릭을 사용하여 선택한 오브젝트에 따라서 해당 씬에 맞게 이동하도록 구현한다.

1. Level-1 롤러코스터 이동 구현

첫번째 과제에서 카메라 이동이 고정되어 있었지만, 카메라가 롤러코스터 (플레이어) 오브젝트 이동 방향에 맞게 움직이도록 한다.

1. Level-2 탱크 전투 구현

첫번째 과제에 구현하지 못했던 필수 구현 부분들을 전부 구현한다.

승리 글자 오브젝트가 카메라 기준으로 보이게 만든다.

**가정**

* 해당 과제는 Direct3D 12 기반으로 윈도우 11, Visual Studio 2022 Community를 기준으로 가정한다.
* 실행 환경은 기본적으로 Release 모드이어야 한다.
* 쉐이더 파일 등 실행 파일과 동일한 폴더 내에 존재해야 한다.
* 키보드 마우스를 사용한다.

**실행 결과 및 조작법**

**1. 시작 화면**

스크린샷, 텍스트, 소프트웨어, 운영 체제이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

**텍스트, 스크린샷, 그래픽 디자인, 디자인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.**

**기능** : “3D 게임 프로그래밍 1 유영빈” 이 화면 중앙을 기준으로 회전한다. 해당 3D 모델을 클릭하게 되면 3초간 폭발이 진행된 후 메뉴 화면으로 이동한다.

**조작법** :

마우스 – 3D 오브젝트를 왼쪽 클릭한다.

ESC – 종료

**실행 결과** : 폭발 이팩트와 함께 3초 후, 메뉴 화면으로 이동한다.

**구현내용** :

GameFrameWork.h 헤더에

CCamera\* m\_pStartSceneCamera, StartScene\* m\_pStartScene, CStartScenePlayer\* m\_pStartScenePlayer 변수를 사용한다.

m\_pStartScene이 씬 객체이며, void CStartSceneShader::BuildObjects()함수가 랜더링 될 오브젝트를 생성한다.

각 오브젝트는 COrbitingObject 클래스를 사용한 오브젝트이다.

COrbitingObject클래스는 { 300.0f, 0.0f, 0.0f } 를 중심으로 공전한다.

CStartSceneShader::PickObjectByRayIntersection()함수에서 마우스 피킹을 감지한 후, m\_ppExplosionObjects[i]->StartExplosion(); 부분에서 폭발 오브젝트가 실행된다.   
  
폭발 오브젝트는 CExplosionCubeObject클래스 이며, m\_bExploding와 m\_bActive가 True가 될 경우 CExplosionCubeObject::Animate()함수가 실행되어 폭발이 진행된다.  
  
텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

3초가 지난 후, 메뉴 씬으로 전환된다.

**2. 메뉴 화면**

텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.



**기능** : 마우스 왼쪽 클릭을 한 상태로 마우스 회전에 따라 카메라 회전이 가능하고, 마우스 왼쪽 클릭을 사용하여 각 오브젝트를 선택하게 되면 해당 씬으로 화면이 전환된다.

**조작법** :

마우스 – 왼쪽 클릭한 상태로 카메라 회전, 오브젝트를 마우스 왼쪽 클릭으로 해당 씬 이동

ESC – 종료

**실행 결과** : Start, Level-1 오브젝트(그린)를 클릭 시, Level-1 씬으로 이동한다. Tutorial, Level-2를 클릭 시, Level-2 씬으로 이동한다. End 클릭 시, 종료된다.

**구현 내용** :

GameFrameWork.h에 CCamera\* m\_pMenuSceneCamera, MenuScene\* m\_pMenuScene, CMenuScenePlayer\* m\_pMenuScenePlayer를 사용한다.

CMenuSceneShader::BuildObjects()함수에서 오브젝트를 생성한다.

CMenuSceneObject클래스를 사용하고, 해당 오브젝트 클래스는 각 오브젝트에 std::string objectName;라는 변수를 사용해서 오브젝트 별 이름을 부여한다.

CMenuSceneShader::PickObjectByRayIntersection()함수에서 오브젝트에 마우스 픽킹을 감지한다.  
텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

이미지와 같이 objectName의 이름에 따라서 씬을 전환한다.

**3. Level-1 장면**

스크린샷, 라인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

**기능** : 노란색 오브젝트를 따라서 플레이어가 이동을 한다.

**조작법** :

n, N – Level-2 장면으로 전환한다.

ESC - 메뉴 장면으로 화면을 전환한다. (시간이 5초 정도 걸릴 수 있습니다, 오브젝트 해제 후 재생성하기 때문에)

**실행 결과** : 오브젝트 끝에 도달 시, Level-2 장면으로 씬을 전환한다.

**구현 내용** :

GameFrameWork의 CCamera\* m\_pLevel1SceneCamera, Level1Scene\* m\_pLevel1Scene, CLevel1ScenePlayer\* m\_pLevel1ScenePlayer 변수를 사용한다.

Level1ScenePlayer는 Level1Scene에도 존재하며 GameFrameWork:: BuildObjects()함수에서 

이미지와 같이 추가한다.

CLevel1SceneShader클래스를 사용한다.

CGameObject\*\* m\_ppObjects는 오브젝트를 담고,

std::vector<XMFLOAT3> m\_vRailPath;

std::vector<float> m\_vRailAngles;

std::vector<float> m\_vRailPitches;

각 변수는 위치, 회전, 피치값을 가지고 있다.

void CreateRailPath();는 레일을 생성하는 함수이다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

BuildObject함수에서 저장해놨던 위치, 회전, 피치값을 사용하여 생성한다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Level1Scene클래스에서 m\_pLevel1ScenePlayer를 레일에 맞게 이동시킨다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

마지막 레일에 도달 시, Level-2씬으로 전환한다.

**4. Level-2 장면**

스크린샷, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

**기능** : 플레이어는 빨간색 탱크이고, 적 탱크는 파란색이다. 대포를 발사하여 적 탱크를 전부 제거한 후, 승리하면 된다.

적 탱크

스크린샷, 텍스트, 지도, 멀티미디어 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

스크린샷, 다채로움, 그린, 그래픽 디자인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.총알 발사 적 탱크 총알 발사 (핑크색)

스크린샷, 만화 영화이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

지도, 블루, 일렉트릭 블루이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.장애물 플레이어 탱크 피격 시 폭발

포스트잇 노트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

리스폰 쉴드 기능 (1회 피격 방어)

스크린샷, 아동 미술, 다채로움, 예술이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.스크린샷, 아동 미술, 지도, 예술이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

스크린샷, 그린, 디자인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.적 탱크 폭발 승리 표시!

스크린샷, 그린, 만화 영화, 다채로움이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

적 탱크 타겟 발사 (대포알이 적 탱크를 향해서 자동 공격)

스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

스크린샷, 다채로움이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

**조작법** :

초기에 방향키를 사용해서 움직여줘야 합니다. 방향키를 입력하면 탱크가 내려가집니다.

마우스 – 마우스 왼쪽 클릭을 누른 채로 화면이동, 적 플레이어 탱크 타겟

방향키 – 키보드 방향키를 사용한 플레이어 이동

왼쪽 ctrl – 총알 발사

e – 쉴드

t – 승리 표시

**실행 결과** : ctrl 키를 사용하여 적 탱크를 제거할 수 있으며, 적 탱크의 총알(핑크색)을 맞게 되면 플레이어 탱크 폭발 후, 3초 리스폰 후 부할, e키를 눌러 쉴드가 생기며 1회 피격을 방어한다. 모든 플레이어를 제거하면 승리 표시가 뜨고, t를 눌러도 승리 표시가 뜨게된다.

**구현 내용** :

GameFrameWork의 CCamera\* m\_pCamera, CScene\* m\_pScene, CPlayer\* m\_pPlayer를 사용한다.

CScene.cpp에서 std::vector<CEnarmyTankPlayer\*> m\_vEnemyTankPlayers;는 적 탱크 플레이어를 담고 있다. 위치는 편의를 위해 모든 적 탱크는 플레이어 바로 아래에 생성된다. (플레이어 탱크가 적 총알을 맞기 힘들어서 모아서 생성함)

CPlayer\* m\_pPlayer클래스는

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

이미지에서 CTerrainPlayer를 사용하여 생성하였고, CTankMesh를 사용하여 플레이어 오브젝트를 나타내었다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

적 탱크도 동일하게 CTankMesh를 사용하였다.

CEnarmyTankPlayer클래스는 적 탱크 플레이어 클래스이고, CEnarmyTankPlayer::Update()함수에서 이동 방향을 결정한다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

포신이 진행 방향의 앞을 항상 바라보게 하기 위해서 랜덤한 시간에 방향을 번경하게 되면 앞 방향을 유지하도록 했다.

지형 크기를 terrain 맵 크기로 하고 싶었지만, 편의성을 위해서 지형 크기를 줄여놨다.

CEnarmyTankPlayer클래스의 m\_bActive변수는 적 탱크 상태를 나타내며 false시 사망 처리한다.



두 변수는 적 플레이어와 적 탱크 총알을 담는다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

총알을 미리 생성한 후, 상태값을 false로 변경하여 비활성화 한다. 위치를 숨겨 놓으려 했지만 상태값 기준으로 처리하는게 랜더링 면에서 나을 거 같아서 CBulletObject의 m\_bActive변수를 사용하여 관리한다. (적 탱크도 동일)

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

총알은 바라보는 방향으로 나아가며, 일정 거리 이상으로 날아가게 되면 상태 값을 false로 변경한다. (SetPosition함수는 상태값을 추가했기 때문에 의미는 없다.)

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

CScene::fire()함수를 사용하여 총알이 발사되고,

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

발사된 총알은 CScene::CheckBulletEnemyCollisions()함수를 사용하여 충돌 검사한다.

CScene::CheckBulletEnemyCollisions()함수는 CScene::AnimateObjects()함수에서 호출되고 있다.

텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

CScene::FireEnemyBullet()함수는 적 탱크가 총알을 발사하는 함수이며,

옆 이미지 함수를 사용하여 적 탱크 총알과 플레이어에 충돌을 검사한다.

shield\_active 변수를 사용하여 쉴드가 생성 되어 있다면 1회 방어한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.  


폭발할 오브젝트들을 담는 변수이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

100 조각으로 생성을 한 후,

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다. CScene::CreateExplosionAt()함수에서 플레이어 또는 적 탱크가 폭발 시, 해당 위치로 이동하여 폭발 애니메이션을 실행한다. (SetActive()함수로 true가 되면 애니메이션 실행됨)

CGameObject\*\* m\_ppShield; 변수는 쉴드 오브젝트를 담고 있으며, shield\_active변수를 사용해서 true이면 오브젝트가 랜더링 된다. True일 경우 1회 피격을 방어해준다. (위 사진에 있음)



CEnarmyTankPlayer\* m\_pTargetTank; 변수는 마우스 왼쪽 클릭 시 적 플레이어 탱크를 타겟한다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

CScene::PickObjectPointedByCursor()함수에서 적 탱크 플레이어를 m\_pTargetTank 변수에 담아둔다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

CScene::fire()함수에서 타겟된 적 탱크가 있다면 적 탱크를 향해서 총알을 발사한다.



CObjectsShader \*m\_pShaders 변수는 장애물이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.  
CScene::CheckPlayerObstacleCollision()함수는 플레이어 탱크가 해당 오브젝트에 겹치게 되면 이동을 멈춘다.

모든 충돌 검사는 구 형태로 진행했다.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

두 객체를 구 형태로 받아와서 중심위치와 반지름을 가져와 충돌을 비교한다.



std::vector<CGameObject\*> m\_vWinObjects; 변수는 승리를 표현하기 위한 글자 를 담아둔 오브젝트이다. m\_vWinObjectLocalOffsets변수는 카메라 기준으로 글자 오브젝트의 위치를 담아두는 위치 변수이다.

텍스트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Animate함수에서 현재 카메라의 위치, 방향 벡터를 가져온 후, 카메라의 앞쪽 10.0f 만큼 떨어진 지점에 오브젝트 위치값을 가져온 후, 각 축에 대한 기준으로 오프셋 벡터를 생성하여 위치를 계산하여 카메라 앞에 넣는다.