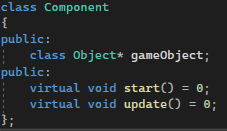
1. 메인 프로젝트 클래스 컴포넌트화

이전의 Engine Project에 있는 컴포넌트 구조를 메인 프로젝트에 병합한다.

Component Class

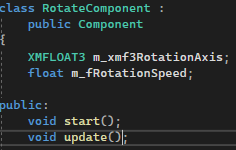


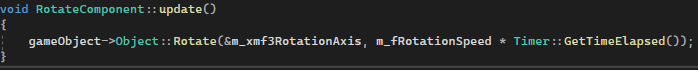
컴포넌트 기본 클래스이며 추상 클래스로 구현했다.

Start() – 컴포넌트가 시작할 때 실행되는 함수 (ex: 초기화)

Update() – 컴포넌트의 실질적인 기능을 담당하는 부분

컴포넌트 클래스 생성 예시 – RotateComponent(조금씩 회전하는 기능)





Object Class

텍스트이(가) 표시된 사진

자동 생성된 설명

오브젝트는 컴포넌트의 리스트를 갖고 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Start(), Update() – 오브젝트가 갖는 컴포넌트들의 start()와 update()를 호출한다.

AddComponent() – 컴포넌트 추가로, 템플릿을 이용해 컴포넌트 클래스를 추가한다.

GetComponent() – 컴포넌트 리스트에 찾는 컴포넌트가 있는지 확인하고 반환한다.

오브젝트 생성 및 컴포넌트 추가 예시

텍스트이(가) 표시된 사진

자동 생성된 설명

GameScene Class

게임을 진행하다 보면 많은 오브젝트들이 생성되고 파괴될 것이다.

그런데 생성과 파괴가 이루어진 상태에서 오브젝트의 상태를 바꾸려고 한다면 오브젝트가 원하는 값으로 행동하지 않게 되거나 파괴된 오브젝트를 행동하려고 할 수 있을 것이다.

이러한 문제를 해결하기 위해 생성과 파괴, 오브젝트 갱신을 따로 두어 서로 꼬이지 않도록 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

creationQueue – GameScene에 추가할 오브젝트 포인터를 갖는 큐 (생성 큐)

gameObjects – GameScene에서 관리하는 오브젝트들 (오브젝트 리스트)

deletionQueue – GameScene에서 삭제할 오브젝트 포인터 (삭제 큐)

MainScene – 정적 객체 포인터로 만들어 현재 GameScene을 쉽게 부를 수 있게 한다.

CreateEmpty() – 동적 할당으로 Object를 생성하고 포인터를 반환하는 함수이다.

PushDelete() – 삭제할 오브젝트 포인터를 deletionQueue로 추가하는 함수이다.

Update() –

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 생성 큐에서 오브젝트를 하나 가져와 start()를 호출하고, 오브젝트 리스트로 추가한다.
2. 오브젝트 리스트에 있는 모든 오브젝트의 update()를 호출한다.
3. 삭제 큐에 있는 오브젝트를 오브젝트 리스트에서 찾고 삭제한다.

후에 메모리 해제 과정에서 문제되지 않도록 포인터를 스마트 포인터로 바꿔 RAII 구조로 구현하면 예외가 발생했을 때 메모리를 제대로 해제할 수 있을 것이고 delete를 사용하는 코드들을 적게 만들 수 있어 다음 계획으로 추가할 것이다.