Client 구조 정리

1) 전체 구조

- **CGameFramework::FrameAdvance()**
  - scene의 Animate, render 수행
  - Tick() 함수 수행
- **CTimer::Tick**
  - frameRate에 맞게 반복문 수행하여 deltaTime 계산
  - 싱글톤 패턴 적용

2) 클래스 구조 표

클래스	역할	주요 특징
CObject	기본 게임 오브젝트	- Mesh 보유 - world_matrix 변경으로 이동·회전·스케일 반영 - right, up, look, position 벡터 기반 이동 및 회전. world_matrix를 참조
CPlayer (CObject 상속)	플레이어 오브젝트	- 임시로 육면체 mesh 사용 - 카메라 객체를 소유하고 제어
CShader	렌더링 파이프라인 관리	- Pipeline State 관리 - 게임 오브젝트 렌더링 처리 - **BuildObjects()** 함수에서 오브젝트 생성
CScene	게임 전체 관리	- Shader 객체 관리 - Player 객체 관리 - Billboard용 임시 카메라 보유(삭제 예정) - **BuildObjects()** 함수에서 오브젝트 생성

● CObject vs CPlayer

- **CObject**
  - 단순한 오브젝트
  - world_matrix만 수정하여 위치/회전/스케일을 적용
- **CPlayer**
  - 방향 벡터(right, up, look)를 직접 회전시키는 능동적 오브젝트

- 방향 벡터를 기반으로 world_matrix를 재구성
- 카메라를 소유하여 1인칭/3인칭 시점 구현 가능

● 카메라와 플레이어 관계

- CPlayer는 카메라를 소유
- 카메라는 플레이어의 위치·방향을 참조
- 1인칭: 플레이어의 look 방향 그대로 사용
- 3인칭: offset을 회전시켜 플레이어 뒤에서 따라오는 구조

● CShader

- 렌더링 전용 클래스
- PSO(Pipeline State Object) 설정
- Scene에서 Shader를 통해 오브젝트를 그리는 구조

● CScene

- 입력 처리
- Player 업데이트
- Camera 업데이트
- Shader를 통한 렌더링
- 게임 루프의 중심 역할