. 목표

; 자체 엔진에 Apple Metal 그래픽스 라이브러리 이식.

; Apple 사의 추후 OpenGL 미지원 선언에 따른 대응책임.

. 작업 개괄

; 기존 OpenGL 단독 사용 => OpenGL, Apple Metal 호환 처리.

; 디파인 처리(X), 가상함수 분기(O).

; 추후 Vulkan 또는 DirectX 추가 용이토록 할 것.

; 렌더러 구현 필요.

;; 인터페이스 : RenderSystem dll.

;; 구현 : RenderSystem\_OpenGL dll, RenderSystem\_Metal dll.

; OpenGL 전용 변수 일괄 변경(엔진, 게임) : ex. GLuint -> uint32\_t

; 렌더러 적용 코드와 기존 코드는 디파인으로 분리(원활한 서비스 적용 위해).

; A7 chip, iOS9 이상 Metal 구동. 나머지 조건 OpenGL 구동.

; mtlpp 사용 : C++ 호환 이유.

; Metal 셰이더 작성 필요.

; Unreal, Ogre, Irricht, Source 등 오픈소스 참조(Unreal, Ogre 다수).

. 특이 사항

; 좌표계

;; NDC 좌표계 : OpenGL xyz(-1.0 ~ 1.0), Metal xy(-1.0 ~ 1.0) z(0.0 ~ 1.0)

;; 좌표계 설정 : right handed 통일.

; 시스템 메모리 스트리밍 DRAW 함수 : glVertexAttribPointer(OpenGL) -> setVertexBytes(Metal), 4K 미만.

; 셰이더 파라미터 : 임시 렌더파이프라인 객체 생성, Reflection 객체에 저장.

; 미지원 텍스처 포맷 : Metal 은 24bit 텍스처 포맷 지원 안 함 => non transparency 32bit format.

; 파이프라인 상태 객체

;; OpenGL : 디바이스 알아서 관리.

;; Metal : 수동으로 관리. Hash key / Table..

; hang error(출력 에러 현상) : ios metal project => deployment 값을 9.0 이상으로.

; 속도 관련

;; depth buffer clear 남발 : Metal 은 depth buffer clear 시 렌더패스 새로 생성하는 상황(부하 요소).

;; cpu / gpu 병행 처리 보장토록. 버퍼 위주 활용(동기화 변수 사용 안 함).

; 셰이더 작성 유의 사항.

;; byte 수 유의 : float3( 16byte ), packed\_float3( 12byte )

;; 샘플러 객체 활용(OpenGL ES 는 없음).