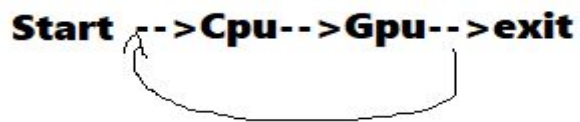


최적화

Cycle



cpu(start, update, awake 등등)
gpu(fixedupdate...)

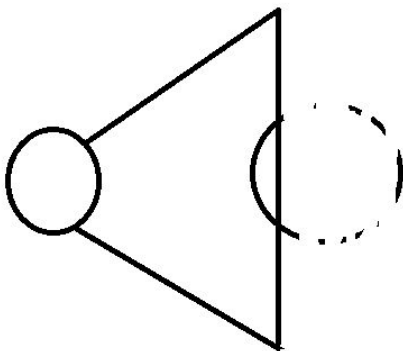
FixedUpdate는 호출빈도가 높다. 따라서 내용을 최소화 시켜야함.

audio 부분

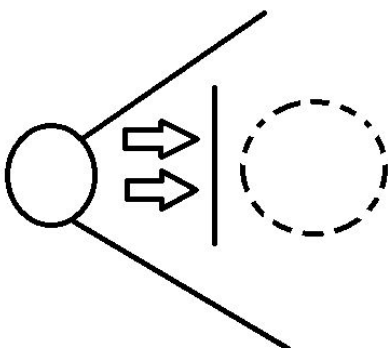
cpu 사용률에 따라 오디오 병목 현상이 일어날 수 있음. 모바일 플랫폼에서 고려해야할 상황임.

컬링

컬링: 필요한 부분만 그리는 기법
프러스텀 컬링 => 영역내 정보만 렌더링하는 기법(공간 베이스)



오클루전 컬링: 영역이 아닌 시야내 정보만 렌더링하는 기법



draw call

최소화 시켜야된다. draw call 자체가 자원을 많이 잡아먹기 때문임.
draw call은 shader 패스 당 하나씩 일어난다.(정점?) 재질의 수 만큼 콜이 일어난다.
해결법은 성질이 동일한 물체들은 하나의 메쉬와 재질을 사용해야됨.(combine children)

이 때 각각 나누어져 있는 걸 부를 때는 아틀라스 텍스처를 이용해 하나로 합치면 줄이기 가능하다.

ugui에서는 canvas가 있고 canvas는 drawcall의 기준이다. 따라서 atlas 처리가 되어야한다.
text는 font 단위로 draw call이 이루어진다. 따라서 font의 수도 줄여야함.

모바일 최적화

gpu 제조사에서 권장하는 확장이 있음.
tegra dxt
adreno atc
공동 etc

물리엔진 관련

픽셀을 겹치게 되면 오버드로우가 발생한다.(큰 거 부터 콜해라)
충돌체는 기본 충돌체 (히박스) 권장.
히박스가 있으면 그걸 옮길 땐 rigidbody 권장한다. 아니면 cpu에 부하가 큼.

라이브러리 사용 주의

게임 엔진 기능 사용하면 c++의 영역을 참조하기 때문에 추가 연산이 발생함.(해결법은 그래서?)

나눗셈 보다 *
삼각함수 => 상수
const, read only 잘 사용하기.