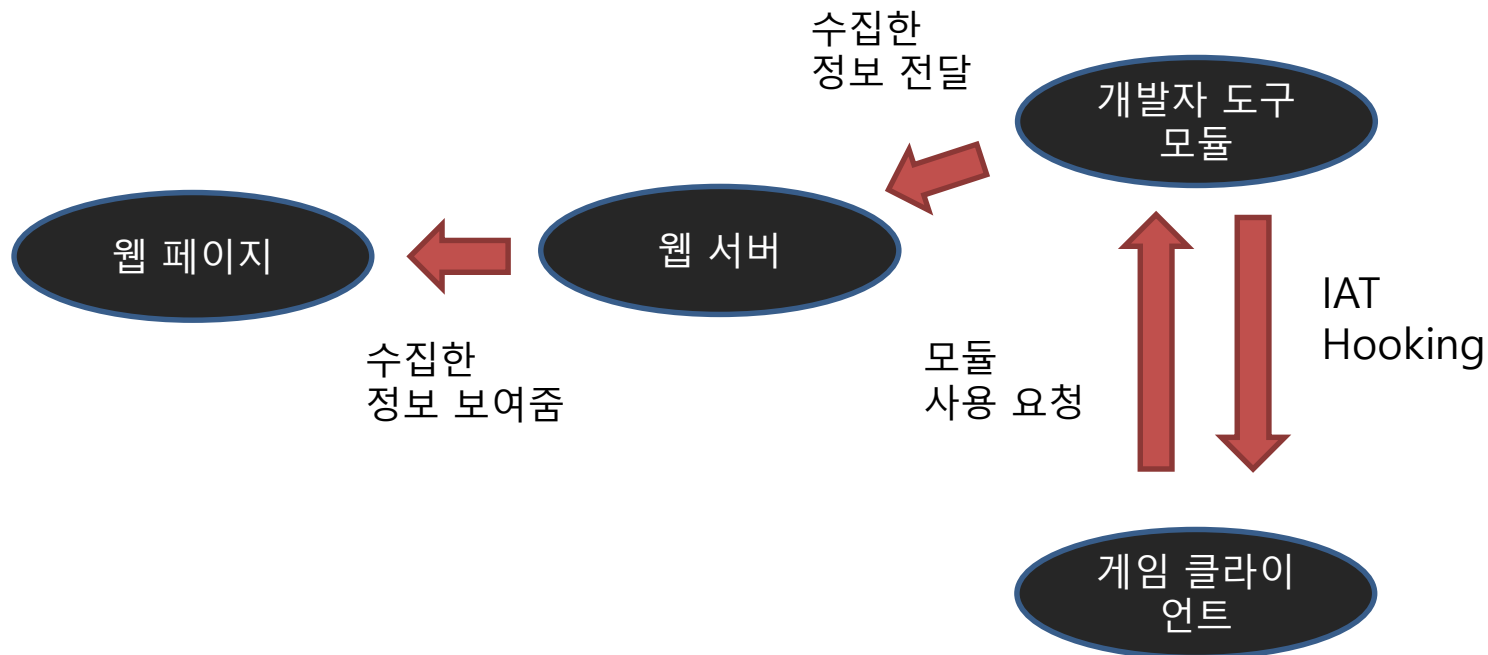


게임 개발을 쉽게 하자!

(개발자 도구 툴을 소개합니다.)

개발자 도구 툴이란?

- 클라이언트 프로파일링 툴입니다.
 - 별도 설치 없이 웹 페이지에서 확인 가능
 - 기초 프로파일링 정보 수집
 - 이벤트 발생 시 사후 디버깅 정보 추출 가능
- 기본 구조



프로파일링이란?

Profiling (computer programming)

From Wikipedia, the free encyclopedia



This article **needs additional citations for verification**. Please help [improve this article](#) by [adding citations to reliable sources](#). Unsourced material may be challenged and removed.
(January 2009) (*[Learn how and when to remove this template message](#)*)

In [software engineering](#), **profiling** ("program profiling", "software profiling") is a form of [dynamic program analysis](#) that measures, for example, the space (memory) or time complexity of a program, the [usage of particular instructions](#), or the frequency and duration of function calls. Most commonly, profiling information serves to aid program [optimization](#).

Profiling is achieved by [instrumenting](#) either the program [source code](#) or its binary executable form using a tool called a *profiler* (or *code profiler*). Profilers may use a number of different techniques, such as event-based, statistical, instrumented, and simulation methods.

Program execution

General concepts

- [Code](#)
- [Translation](#)
 - [Compiler](#)
 - [Optimizing compiler](#)
- [Intermediate representation \(IR\)](#)
- [Execution](#)
 - [Runtime system](#)

어떤 정보를 수집하나요?

- 개발 로그
 - (OutputDebugString(), WriteConsole()...)
- 메모리
 - (commit 메모리, 사용 가능한 메모리..)
- CPU
 - (cpu 사용량)
- GPU
 - (Commit, Dedicated, System GPU 메모리, GPU 사용량)
- DirectX
 - (fixel count, fps)
- NETWORK
 - (send / receive traffic, 네트워크 상태(비정상 종료, 네트워크 혼잡 등..))
- Disk
 - (IO 사용량..)

설정

로그 다운로드

파일 선택

로그 업로드

검색 기준 : 로그 약

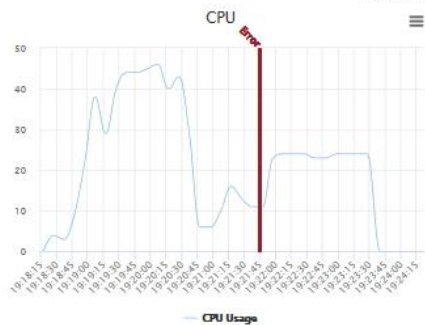
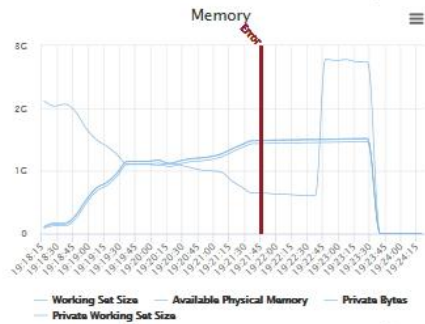
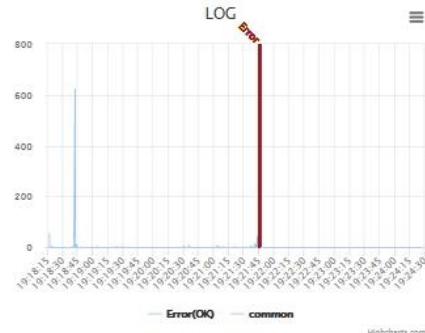
검색어 입력 (필수사항)

검색

로그 : 1750개

SESSION ID : 159

선택조기화



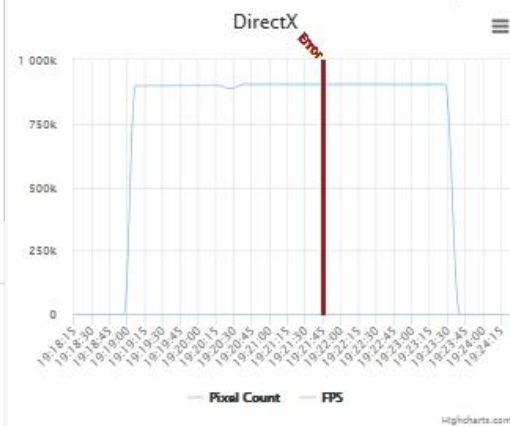
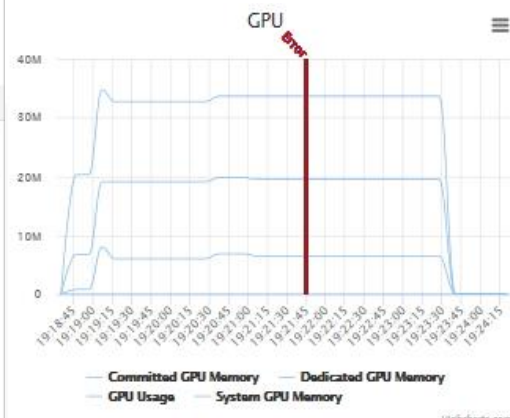
검색된 로그

일자

로그

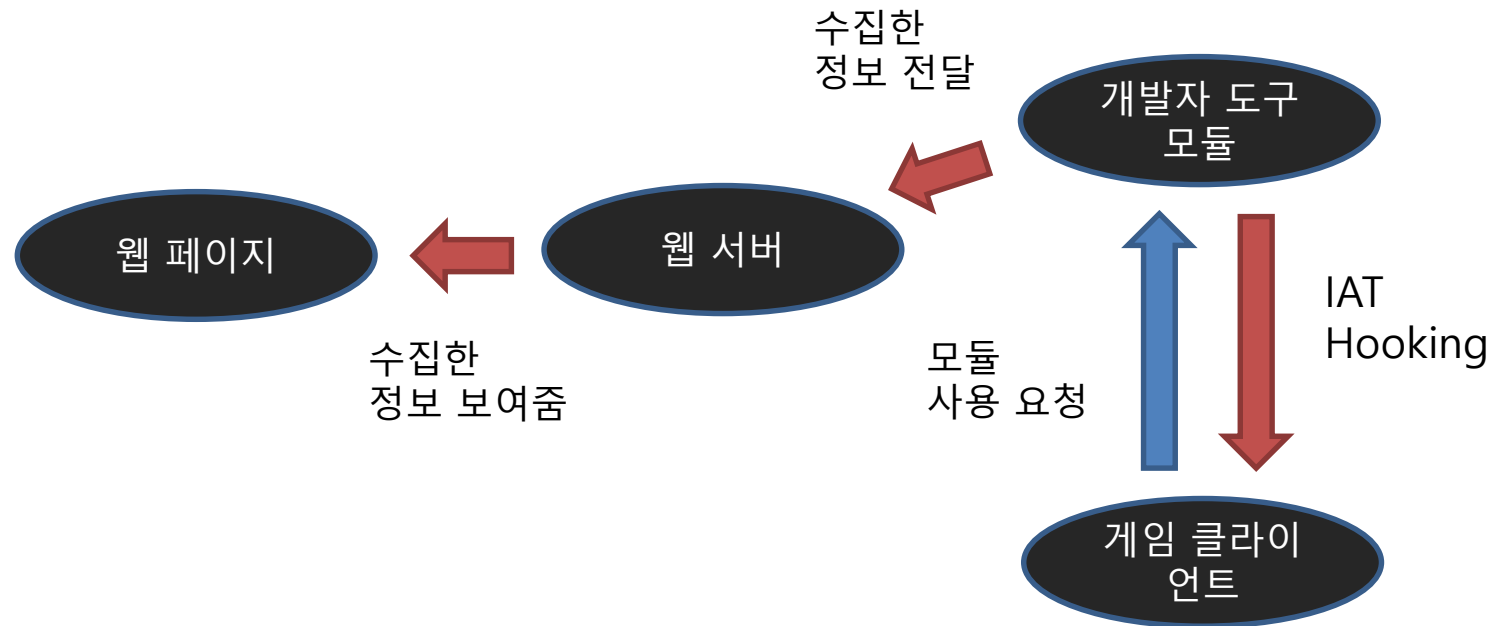
이 곳은 수집된 개발 로그가 출력되며
보안 문제로 해당 부분은 삭제했습니다.

< 1 2 3 4 5 6 7 8 9 10 >



그래서 이걸 왜 사용해야 하나요?

- 1. 프로파일링 정보가 디폴트로 수집됩니다.
 - 별도 프로파일러를 띄울 필요 없이
 - 라이브러리 등록 및 사용 요청만 하면 즉시 프로파일링 정보가 수집됩니다.



그래서 이걸 왜 사용해야 하나요?

- 2. 재현이 힘든 상황에 대한 측정이 가능합니다.

일반적인 상용 프로파일러로는..		
측정 범주	원하는 것	상용 프로파일러
코드 범위	“게임의 전반적인 Frame Rate 를 높이고 싶어”	○ 측정가능
코드 조각	“특정 모듈의 성능과 영향을 확인하고 싶어” (String, DB, 충돌체크 ...)	○ 측정가능
특정상황	“특정 지역에만 가면 느려지는 현상을 잡고 싶어” “특정몹만 만나면 느려지는 현상을 잡고 싶어”	△ 해당 시나리오를 재연하도록 구현후 측정가능
불특정상황	“가끔 알수없이 발생하는 랙을 잡고 싶어” “어느 컴퓨터에서는 가끔 엄청 느려진다던데..”	✗ 측정불가

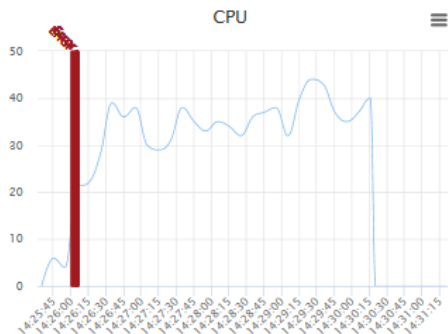
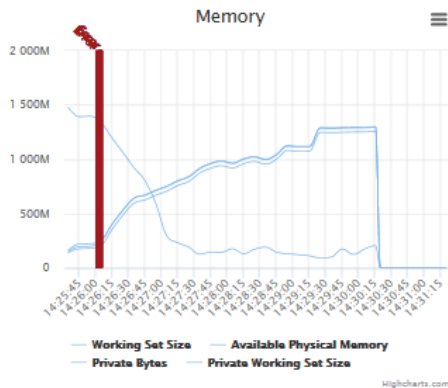
이 부분이
측정 가능합니다.

그래서 이걸 왜 사용해야 하나요?

- 3. 과거 정보와 비교가 가능합니다.
 - ex. 71번째 클라이언트에서 성능 최적화 변경점이 들어갔는데, 성능이 얼마나 좋아졌는지에 대해 확인하고 싶어요.

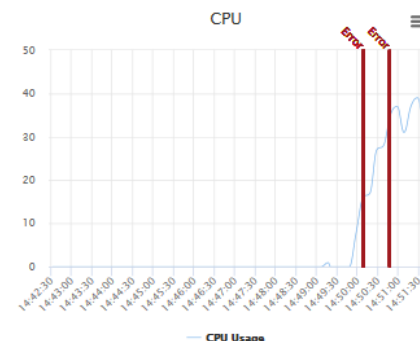
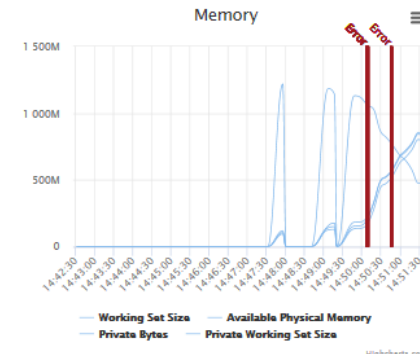
로그 : 2109개

SESSION ID : 70



로그 : 805개

SESSION ID : 71



그래서 이걸 왜 사용해야 하나요?

- 4. 알람 기능을 사용할 수 있습니다.
 - 알람기능이란?
 - 특정 프로파일링 또는 로그 값이 발생한 경우 이벤트를 발생시키는 기능입니다.
 - 알람은 어떨 때 사용하나요?
 - 문제 상황을 확인할 수 있는 기능(ex. 스크린샷, 덤프 등)을 지정할 수 있습니다.
- 예시 상황 :
 - 1) 건물 안으로 들어가면 fps가 떨어져요.
 - 2) 클라이언트가 굳었고, 아무 키도 먹지 않아요.
- 대처 방법
 - 1) fps < 10 일 때 스크린샷 수집 설정
 - 2) Cpu 사용량 > 90 이상 미니덤프 10회 + 풀덤프 1회 추출 설정

설정

로그 프로파일러 스크린샷 덤프 네트워크

☐ 프로파일러 (ON / OFF 설정)

- ☒ 메모리
- ☒ GPU
- ☒ DirectX
- ☒ CPU
- ☒ Network
- ☐ Disk

프로파일링 주기: 10000 ms

알람 설정

입력

Tag	SubTag	Operation	알람을 발생시킬 기준치를 입력하세요.	+
-----	--------	-----------	----------------------	---

타입	기준	제거
CPU	"usage" > 90	-

적용 취소

설정

로그 프로파일러 스크린샷 덤프 네트워크

덤프 기능

☒ ON/OFF

덤프 발생 방법 설정

Alarm	덤프 유형	+
-------	-------	---

타입	기준	덤프 발생 방법	제거
CPU	"usage" > 90	미니 덤프 10회 + 풀 덤프	-

적용 취소

CPU 사용률이 90% 이상일 때
덤프 추출



지원 범위

- 개발자 도구 툴 지원 범위
 - c++ 기반 PC 게임 클라이언트
- 추후 지원 예정 범위
 - PC / 모바일 게임 서버
 - 모바일 게임 클라이언트 지원 예정

시연