

SimpleExceptionHandling 샘플

*이 샘플은 Microsoft 게임 개발 키트(2020년 6월)와 호환 가능합니다.*

# 설명

이 샘플에서는 타이틀에서 발생할 수 있는 예외를 처리하는 다양한 방법을 보여 줍니다.

* 처리되지 않은 예외 필터 – [처리되지 않은 예외 필터](https://docs.microsoft.com/windows/win32/api/errhandlingapi/nf-errhandlingapi-setunhandledexceptionfilter) 를 사용하여 타이틀의 일반적인 예외를 잡아내고 처리하는 방법을 보여 줍니다.
* 구조적 예외 – [구조적 예외 처리](https://docs.microsoft.com/cpp/cpp/structured-exception-handling-c-cpp) 시스템을 사용하는 방법을 보여 줍니다.
* 벡터화된 예외 처리기는 – [벡터화된 예외 처리](https://docs.microsoft.com/windows/win32/debug/vectored-exception-handling) 시스템을 사용하는 방법을 보여 줍니다.
* C++ 언어 예외 – [C++ 언어](https://docs.microsoft.com/cpp/cpp/try-throw-and-catch-statements-cpp)에 기본 제공되는 예외 시스템을 사용하는 방법을 보여 줍니다.
* 맞춤 패턴 – 다른 시스템 조합을 사용하는 맞춤 패턴을 보여 줍니다.

# 샘플 빌드

Xbox One 개발 키트를 사용하는 경우 활성 솔루션 플랫폼을 Gaming.Xbox.XboxOne.x64로 설정합니다.

Xbox Series X|S를 사용하는 경우 활성 솔루션 플랫폼을 Gaming.Xbox.Scarlett.x64로 설정합니다.

데스크톱을 사용하는 경우 활성 솔루션 플랫폼을 Gaming.Desktop.x64로 설정합니다.

*자세한 내용은 GDK 설명서의* 샘플 실행*을 참조하세요.*

# 샘플 사용

각 데모에 대해 컨트롤러의 해당 버튼을 누릅니다. 예외가 발생할 때 코드에서 수행되는 작업 순서가 디스플레이에 표시됩니다.

참고: 처리되지 않은 예외 필터 예제는 디버거가 연결된 경우 다르게 동작하며, 주석에 추가 세부 정보가 있습니다.

# 구현 참고 사항

모든 예제는 예제 폴더에 포함됩니다. 각 시스템에 대한 세부 정보와 작동 방식이 자세히 설명되어 있습니다.

# 업데이트 기록

2021년 4월 초기 릴리스

# 개인정보처리방침

샘플을 컴파일하고 실행하는 경우 샘플 사용량을 추적할 수 있도록 샘플 실행 파일의 파일 이름이 Microsoft에 전송됩니다. 이 데이터 수집을 옵트아웃하려면 "샘플 사용량 원격 분석"으로 레이블이 지정된 Main.cpp에서 코드 블록을 제거할 수 있습니다.

Microsoft의 개인 정보 정책에 대한 자세한 내용은 [Microsoft 개인정보처리방침](https://privacy.microsoft.com/en-us/privacystatement/)을 참조하세요.