

SimpleExceptionHandling Sample

*このサンプルは Microsoft Game Development Kit (2020 年 6 月) と互換性があります。*

# 説明

このサンプルでは、タイトルで発生する可能性のある例外を処理するいくつかの方法を示します。

* ハンドルされない例外フィルターは - [ハンドルされない例外フィルター](https://docs.microsoft.com/windows/win32/api/errhandlingapi/nf-errhandlingapi-setunhandledexceptionfilter)を使用してタイトルの一般的な例外をキャッチして処理する方法を示します。
* 構造化例外 – [構造化例外処理](https://docs.microsoft.com/cpp/cpp/structured-exception-handling-c-cpp)システムを使用する方法を示します。
* ベクトル化例外 – [ベクトル化例外処理](https://docs.microsoft.com/windows/win32/debug/vectored-exception-handling)システムを使用する方法を示します。
* C++ 言語の例外 – [C++ 言語](https://docs.microsoft.com/cpp/cpp/try-throw-and-catch-statements-cpp)に組み込まれている例外システムの使用方法を示します。
* 推奨パターン – 他のシステムの組み合わせを使用する推奨パターンを示します。

# サンプルのビルド

Xbox One 開発キットを使用している場合は、アクティブ ソリューション プラットフォームを Gaming.Xbox.XboxOne.x64 に設定します。

Xbox Series X|S を使用している場合、アクティブ ソリューション プラットフォームを Gaming.Xbox.Scarlett.x64 に設定します。

デスクトップを使用している場合には、アクティブ ソリューション プラットフォームを Gaming.Desktop.x64 に設定します。

*詳細については、GDK のドキュメントの*「サンプルの実行」*を参照してください。*

# サンプルの使用方法

各デモについては、コントローラーの対応するボタンを押します。ディスプレイには、例外が発生したときにコード内で発生する操作の順序が表示されます。

注意:ハンドルされない例外フィルターの例は、デバッガーがアタッチされている場合は動作が異なります。コメントに追加の詳細が記載されています。

# 実装上の注意

すべての例は Examples フォルダーに含まれています。各システムの詳細とその機能について詳しく説明されています。

# 更新履歴

2021 年 4 月の初期リリース

# プライバシーに関する声明

サンプルをコンパイルして実行する場合、サンプルの使用状況を追跡するために、サンプル実行ファイルのファイル名が Microsoft に送信されます。このデータ コレクションからオプトアウトするには、Main.cpp の「Sample Usage Telemetry」というラベルの付いたコードのブロックを削除します。

全般的な Microsoft のプライバシー ポリシーの詳細については、「[Microsoft プライバシー ステートメント](https://privacy.microsoft.com/en-us/privacystatement/)」を参照してください。