マウスカーソルのサンプル

*このサンプルは、2018 年 4 月の XDK と互換性があります。*

# 概要

このサンプルは、Xbox アプリとゲームにマウスコントロールを実装する方法を示します。3 つの異なるマウスコントロールが示されています。（1）windows マウスを読み取る絶対マウスコントロール、（2）windows マウスをキャプチャしてマウスルック制御を行う相対マウスコントロール、（3）windows カーソルをキャプチャしてアプリまたはゲームのウィンドウ内に留まるクリップカーソルマウスコントロール。

# サンプルの使用

このサンプルは、UI とゲームモードの相互作用にマウスコントロールを使用します。マウスの右ボタンを押して、さまざまなゲームモードを終了し、UIに戻ります。

# 実装上の注意

サンプル全体を通して、3 つの異なるマウス制御モードが共通して参照されています。

## 絶対モード（デフォルト）

このモードでは、マウスカーソルは OS によって制御され、アプリケーションウィンドウから自由に移動できます。これは、ほとんどのプログラムで毎日使用されるデフォルトのウィンドウ操作です。ICoreWindow イベント「PointerMoved」は、マウスの動きを監視して読み取るために使用されます。

[タッチ入力に応答する（DirectX と C++）](https://msdn.microsoft.com/ja-jp/library/windows/apps/xaml/hh994925.aspx)を参照してください。

## 相対モード

このモードは、「マウスルック」または「マウス移動」カメラモードとも呼ばれます。一部のアプリやゲームでは、マウスの位置ではなくマウスの相対的な動きに基づいて、マウスを一般的な入力デバイスとして使用する方が好ましい場合があります。これは一人称シューティングゲームでよく見られます。

サンプルでこのモードに入るには、一人称シューティングゲームのタイルを選択します。このモードでは、モードを終了して絶対マウスモードに戻る方法が必要です。このサンプルでは、「Esc」を押すとモードを終了してメニューに戻ります。

このモードを実装するには、システムカーソルを null に設定して非表示にします。マウスの動きは ICoreWindow イベントの「MouseMoved」を通じて読み取られます。

[マウスコントロールの開発（DirectX と C++）](https://msdn.microsoft.com/ja-jp/library/windows/apps/xaml/hh994925.aspx)を参照してください。

## クリップカーソルモード

クリップカーソルモードは、マウスがアプリケーションのウィンドウの境界に制限される絶対モードの変形です。このモードに入るには、リアルタイム戦略タイルを選択します。このモードでは、終了して絶対モードに戻るための方法も必要です。「Esc」を押すとモードを終了してメニューに戻ります。

このモードを実装するには、システムカーソルを null に設定して非表示にし、ICoreWindow イベント「MouseMoved」を通じてマウスの動きを読み取ります（相対モードと類似）。その後、アプリケーションは独自のマウスカーソルを描画し、手動で場所を更新するとともに、カーソルがウィンドウから離れないようにする必要があります。クリップカーソルモードを終了して絶対位置に戻ると、システムカーソルは以前は非表示だった場所に表示されます。したがって、アプリケーションが独自のカーソルの描画を停止した場合は、そのアプリケーションカーソルが以前に描画されていた場所にシステムカーソルを設定する必要があります。

# 既知の問題点

# 更新履歴

2018 年 2 月初版リリース

2018 年 6 月に 4k をサポートするように更新されました。これはマウス位置の dip からピクセルへの変換を必要とします。

# プライバシーステートメント

サンプルをコンパイルして実行すると、サンプルの実行可能ファイルの名前がMicrosoftに送信され、サンプルの使用状況の追跡に役立ちます。このデータ収集を無効にするには、Main.cppの「Sample Usage Telemetry」というラベルの付いたコードブロックを削除します。

マイクロソフトのプライバシーポリシー全般に関する詳細については、 [Microsoftのプライバシーステートメント](https://privacy.microsoft.com/ja-jp/privacystatement/)をご参照ください。