# Surface Dial 実装方法について

CLIP STUDIO PAINTは弊社独自開発のアプリケーションフレームワークTriglavで実装しています。 そのため Surface Dial の組み込みは単純ではありません。 まずTriglavフレームワークにSurface Dialを対応することから始まり、そしてアプリケーションに実装します。

## 構成

CLIP STUDIO PAINTへのSurface Dial組み込みは次の4部分で構成されます。

### triglavwinrt.dll

Surface Dialで使用するWinRT APIをCインターフェースで実装したDLLです。 TriglavはVC2010のツールセットでビルドされており、安易にツールセットは変更できません。 そこでSurface DialのWinRTを呼び出すためのCインターフェースのDLLを実装しました。

### ウインドウ(Triglav)

ウインドウに対してSurface Dialのメニューを設定します。 CLIP STUDIO PAINTはクラシックのWindowsアプリケーションのため複数のウインドウを持ちます。 一方で、Surface Dialは1つのウインドウに対してメニューを設定します。 アプリケーション全体を通してひとつのSurface Dialメニューとして動作させるためには この部分でうまく対応する必要があります。

CLIP STUDIO PAINTを実際に使用して、アプリケーション全体で1つのSurface Dialメニューがあるかのように 動作していることを確認してみてください。ただし、Triglavがアプリケーション固有のSurface Dialメニューを 表示するウインドウの種類を限定しているので、不要なウインドウに対しては固有のメニューを設定しません。

### アプリケーション(Triglav)

アプリケーションクラスでSurface Dialメニューを設定する機能を追加し、 またそのSurface Dialからイベントを受け取る機能を実装しています。

### アプリケーション(CLIP STUDIO PAINT)

アプリケーション起動時、アプリケーションクラスにSurface Dialメニューを設定します。 Surface Dialからイベントを受け取ると、アプリケーションクラスでイベントを受信し、 そのイベントの種類のパラメーターに応じてキャンバスに対する動作を実行します。

## 実装

### triglavwinrt.dll

添付した triglavwinrtフォルダの プロジェクトを見てください。 完全なプロジェクトを収録しています。

### ウインドウ(Triglav)

Triglavのウインドウに対する拡張は２つの部分で構成されます。

#### DLLの呼び出し

　これらの2つのクラスはtriglavwinrt.dllを呼び出すための実装です。 　単純なDLLのラッパーではなくオリジナルSurface DialのWinRTに近いインターフェースを備えます。

* PWLegacyWindowRuntimeModuleWin
* PWLegacyDialModuleWin

#### ウインドウの実装

PWLegacyWindowWinはもっとも重要な実装部分です。 Surface Dialに関するソースコードのみ抜粋していきます。

アプリケーションの開始時、初期化を実行します。

if ( PWLegacyWindowRuntimeModuleWin::IsAvailable() ) {  
 PWLegacyWindowRuntimeModuleWin::Initalize();  
 PWLegacyDialModule::Initialize();  
}

アプリケーション終了時、終了処理を実行します

// サーフェスダイヤルの終了  
if ( PWLegacyWindowRuntimeModuleWin::IsAvailable() ) {  
 PWLegacyWindowWin::SetDialMenu( PWDialMenu::DefaultMenu );  
 PWLegacyDialModule::Terminate();  
 PWLegacyWindowRuntimeModuleWin::Terminate();  
}

ウインドウを作成したとき、ウインドウに対してApplyDialMenuを呼び出し ウインドウに対してアプリケーションに設定されているSurface Dialメニューを設定します。

PWLegacyWindowWin::ApplyDialMenu( windowObject );

ウインドウが WM\_SETFOCUS を受けとたっとき、そのウインドウに対してApplyDialMenuSelectionを呼び出し アプリケーションが選択しているSurface Dialメニューをフォーカスウインドウに設定します。

if( message == WM\_SETFOCUS ) {  
 PWLegacyWindowWin::ApplyDialMenuSelection( windowObject );  
}

上記の関数やこのほかの実装はPWLegacyWindowWin.cppを確認してください。 Surface Dialに関するコードの抜粋部分があります。

### アプリケーション(Triglav)

TriglavのアプリケーションクラスはSurface Dialメニューを設定する関数と イベントを受け取るコールバックを提供します。

PWApplication.hから抜粋。

//! @brief アプリケーションクラス  
class PWApplication  
{  
public:  
 //! サーフェスダイヤルメニューの設定  
 void SetDialMenu( const PWDialMenu& dialMenu );  
  
 //! サーフェスダイヤルメニューからのコールバック  
 void EventDialMenu( PWLegacyApplicationEventDialMenuParameterObject dialMenuParameterObject );  
};

PWApplication::SetDialMenuの実装は単にPWLegacyWindowWin::SetDialMenuの呼び出しです。 PWApplication::EventDialMenuはSurface Dialのイベントを受け取ったPWLegacyWindowWinが呼び出し、PWLegacyWindowWinが作成したPWLegacyApplicationEventDialMenuParameterObjectをPWDialMenuEventParameterに変換してイベントを送出します。

PWDialMenuはSurface Dialに設定するメニューを表します。

### アプリケーション(CLIP STUDIO PAINT)

CLIP STUDIO PAINT起動時、PWDialMenuを作成してPWApplication::SetDialMenuを呼び出しメニューを設定します。 CLIP STUDIO PAINTでは、メニューの内容は常に決まっているため、 メニューを設定するのはここだけです。

// サーフェスダイヤルメニューの設定.  
std::vector<PWDialMenuItem> items;  
URDialMenu::InitializeMenu(items);  
SetDialMenu( PWDialMenu( items ) );

またこのときアプリケーションクラスのイベントを接続する設定をして、OperationDialMenuのようにイベントを受け取って処理できるようにします。

Bool OperationDialMenu( PWEvent& rEvent )  
{  
 Bool result = FALSE;  
 EventKind eventKind = rEvent.GetEventKind();  
 if (eventKind == kEventKindDialMenuClick ) {  
 PWDialMenuEventParameter parameter(rEvent);  
 UInt32 menuID = parameter.GetMenuID();  
 // サーフェスダイヤルがクリックされた. 選択状態のメニューはmenuIDのものである.  
 } else if (eventKind == kEventKindDialMenuRotate) {  
 PWDialMenuEventParameter parameter(rEvent);  
 UInt32 menuID = parameter.GetMenuID();  
 double rotation = parameter.Rotation();  
 // サーフェスダイヤルを回転した. 選択状態のメニューはmenuIDのものであり, 回転角度はrotation.  
 } else {  
 // ...  
 }  
 return TRUE;  
}

このような方法でアプリケーション部分でSurface Dialのイベントとそのパラメーターを受け取り、 キャンバスに対して操作をするコードを実装します。

## 同梱物一覧

* PWDialMenu.cpp
* PWDialMenu.h
* PWDialMenuEventParameter.cpp
* PWDialMenuEventParameter.h
* PWDialMenuEventParameterType.h
* PWDialMenuType.h
* PWLegacyDialModuleTypeWin.h
* PWLegacyDialModuleWin.cpp
* PWLegacyDialModuleWin.h
* PWLegacyWindowsRuntimeModuleWin.cpp
* PWLegacyWindowsRuntimeModuleWin.h
* PWLegacyWIndowWin.cpp
* ReadMe.docx
* ReadMe.md
* triglavwinrt
* triglavwinrt/dial.cpp
* triglavwinrt/dial.h
* triglavwinrt/resource.h
* triglavwinrt/stdafx.cpp
* triglavwinrt/stdafx.h
* triglavwinrt/targetver.h
* triglavwinrt/triglavwinrt.cpp
* triglavwinrt/triglavwinrt.h
* triglavwinrt/triglavwinrt.rc
* triglavwinrt/triglavwinrt.sln
* triglavwinrt/triglavwinrt.vcxproj
* triglavwinrt/triglavwinrt.vcxproj.filters
* URDialMenu.cpp
* URDialMenu.h
* URDialMenuType.h