Usos de las directivas del preprocesador en los “samples” de You.i engine

En la mayoría de archivos header utilizan un “switch” de compilación llamado DOXY (documentación en línea de las clases),

Ejem (StateMachineApp.h):

#if defined(DOXY)

// This namespace provides a unique path to each class in the samples documentation.

**namespace** yi**::**documentation**::**statemachine

**{**

#endif

//Codigo de la clase

#if defined(DOXY)

**}**

#endif

En todos los archivos de cabecera utilizan guardas con directivas #ifndef y #define para evitar la duplicación de nombres en los archivos de cabecera,

Ejem (HelloWorldApp.h):

#ifndef \_HELLO\_WORLD\_APP\_H\_

#define \_HELLO\_WORLD\_APP\_H\_

//Codigo de la clase

#endif //\_HELLO\_WORLD\_APP\_H\_

Uso de la directiva #pragma para deshabilitar un warning causado por el problema diamante en la herencia,

Ejem (IAPSampleApp.h):

// Disable the inheritence warning

#if !defined(\_\_apple\_build\_version\_\_) && !defined(\_\_ORBIS\_\_)

#pragma warning(push)

#pragma warning(disable : 4250)

#endif

//Codigo clase

#if !defined(\_\_apple\_build\_version\_\_) && !defined(\_\_ORBIS\_\_)

#pragma warning(pop)

#endif

Inclusión condicional de archivos de cabecera, usando directivas condicionales,

Ejem (IAPSampleApp.cpp):

#if YI\_PS4

#include <network/YiNetworkConfigurationPS4.h>

#endif

Definición condicional de constantes con las directivas condicionales,

Ejem (LocalizationAppFactory.cpp):

#if defined(YI\_PS4) || defined(YI\_XBOX\_360) || defined(YI\_PS3)

#define APP\_WIDTH (1920)

#define APP\_HEIGHT (1080)

#else

#define APP\_WIDTH (640)

#define APP\_HEIGHT (480)

#endif

Compilación condicional sujeta a la definición previa de constantes,

Ejem ( \samples\IAP\src\app\linux\main.cpp):

#ifdef YI\_PROCESS\_COMMAND\_ARGS

void ProcessCommandArgs**(**int argc**,** char **\*\*** argv**);**

#else

void ProcessCommandArgs**(**int argc**,** char **\*\*** argv**){** YI\_UNUSED**(**argc**);** YI\_UNUSED**(**argv**);** **}**

#endif

#ifdef YI\_OSX

// On OSX, the current working directory of an application when run from 'Finder' is NOT the directory that the app is in. We need to find what that path is in order to access the assets.

YI\_CHAR pExecutablePath**[**YI\_MAX\_PATH**];**

YI\_UINT32 uSize **=** **static\_cast<**YI\_UINT32**>(sizeof(**pExecutablePath**));**

**if** **(**\_NSGetExecutablePath**(**pExecutablePath**,** **&**uSize**)** **==** 0**)**

**{**

YI\_CHAR pLinkPath**[**YI\_MAX\_PATH**];**

**if** **(**realpath**(**pExecutablePath**,** pLinkPath**)** **!=** 0**)**

**{**

const CYIString executablePath **=** CYIFile**::**ExtractParentFromPath**(**pLinkPath**);**

path **=** executablePath**;**

**}**

**}**

#endif

#if !defined(YI\_DRAW\_SKIP)

// force redraw when draw skip is undefined

bShouldDraw **=** **true;**

#endif

Creando aliases para cadenas de caracteres con la directiva #define,

Ejem (PlatformViewApp.cpp)

#define LOG\_TAG "PlatformViewApp"

Inclusión de archivos de cabecera con #include, tanto archivos de la implementación estándar del engine como archivos del mismo directorio donde está el archivo que los incluye,

Ejem (PlatformViewApp.h)

#include "ButtonAndLabelController.h"

#include "ConnectivityController.h"

#include "WebViewController.h"

#include <framework/YiApp.h>

#include <signal/YiSignalHandler.h>