Simple keyboard wrapper for games

This article presents a simple class to handle keyboard input in a game-like way.

Class declaration

```
// INCLUDES
#include <e32keys.h>
#include <e32std.h>

// CLASSES

/**
 * A simple keyboard wrapper.
 */
class TKeyboardWrapper
{
  public:

    // key codes not found in header files
    enum TKeys {
  EKey0 = 48,
  EKey1,
  EKey2,
  EKey3,
  EKey4,
  EKey5,
  EKey6,
  EKey7,
  EKey8,
  EKey9,
  EKeyShift = 127,
  EKeyStar  = 133
  };

  public:

  /**
   * Default constructor.
   */
  TKeyboardWrapper ()
  { Clear (); }


  /**
   * Clears key state.
   */
  void Clear ()
  { Mem::FillZ (iKeys, 256); }


  /**
   * Asks if some key is down.
   */
  TBool IsKeyDown (TInt aKey) const
  { ASSERT (aKey >= 0 && aKey < 256); return iKeys [aKey]; }


  /**
   * Sets a key as down.
   */
  void SetKey (TInt aKey)
  { ASSERT (aKey >= 0 && aKey < 256); iKeys [aKey] = ETrue; }


  /**
   * Sets a key as up.
   */
  void ClearKey (TInt aKey)
  { ASSERT (aKey >= 0 && aKey < 256); iKeys [aKey] = EFalse; }


  /**
   * Check if any key is down.
   */
  TBool AnyKeyDown () const
  {
    for (TInt16 i = 0; i < 256; ++i)
     {
       if (iKeys [i])
 return true;
     }

    return false;
  }

  private:
    TBool iKeys [256];
};
```

Usage

It is necessary to collect the keyboard data in some of the methods that receive keyboard input. For example, using the `HandleKeyEventL()` of the Application UI class:

```
TKeyResponse CMyAppUi::HandleKeyEventL (const TKeyEvent& aKeyEvent, TEventCode aType)
{
```


(C) Copyright Nokia 2013. All rights reserved.

```
 if (aType == EEventKeyDown)
  iKeyboard.SetKey (aKeyEvent.iScanCode);
 else
 if (aType == EEventKeyUp)
  iKeyboard.ClearKey (aKeyEvent.iScanCode);

 return EKeyWasConsumed;
}
```

In the example, `iKeyboard` is a `TKeyboardWrapper` instance. Later, to get the key state it is necessary to use the constants defined in the `TStdScanCode` enum (`e32keys.h`). Example:

```
void CMyAppUi::ProcessInput ()
{
  if (iKeyboard [EStdKeyLeftArrow] )
    // left arrow is down

  if (iKeyboard [TKeyboardWrapper::EKey0] )
    // 0 key is down

}
```

It is important not to use the constants defined in the `TKeyCode` enum, because their use leads to incorrect behavior and possibly to an invalid array access.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.