

```

#include "InputControl.h"
#include "DxLib.h"

/*****
*   マクロ定義
*****/
#define XINPUT_BUTTON_MAX (16)                                // コントローラーのボタン数

/*****
*   型定義
*****/
enum KEY_STATE
{
    E_NONE,                // 未入力
    E_CLICK,               // 押した瞬間
    E_PRESS,               // 押し続けてる
    E_RELEASED,            // 離れた
    E_KEY_STATE_MAX
};

/*****
*   変数宣言
*****/
KEY_STATE button_state[XINPUT_BUTTON_MAX];                    // 各ボタン入力状態

/*****
*   プロトタイプ宣言
*****/

/*****
*   入力制御機能：初期化处理
*   引 数：なし
*   戻り値：なし
*****/
void InputControl_Initialize(void)
{
    int i;                // ループカウンタ

    // 入力状態の初期化
    for (i = 0; i < XINPUT_BUTTON_MAX; i++)
    {
        button_state[i] = E_NONE;
    }
}

/*****

```

\* 入力制御機能：更新処理

\* 引 数：なし

\* 戻り値：なし

\*\*\*\*\*/

void InputControl\_Update(void)

{

int i; // ループカウンタ  
XINPUT\_STATE input\_controller; // コントローラー入力情報

// コントローラーの入力情報を取得

GetJoypadXInputState(DX\_INPUT\_PAD1, &input\_controller);

// 入力状態の更新

for (i = 0; i < XINPUT\_BUTTON\_MAX; i++)

{

if (input\_controller.Buttons[i] == TRUE)

{

switch (button\_state[i])

{

case E\_NONE:

case E\_RELEASED:

button\_state[i] = E\_CLICK;

break;

case E\_CLICK:

case E\_PRESS:

button\_state[i] = E\_PRESS;

break;

default:

button\_state[i] = E\_NONE;

break;

}

}

else

{

switch (button\_state[i])

{

case E\_NONE:

case E\_RELEASED:

button\_state[i] = E\_NONE;

break;

case E\_CLICK:

case E\_PRESS:

button\_state[i] = E\_RELEASED;

break;

default:

button\_state[i] = E\_NONE;

break;

```

    }
}
}

/*****
* 入力制御機能：押されているか判定処理
* 引 数：XINPUTのボタン情報
* XINPUT_BUTTON_DPAD_UP          (0)          // デジタル方向ボタン上
* XINPUT_BUTTON_DPAD_DOWN        (1)          // デジタル方向ボタン下
* XINPUT_BUTTON_DPAD_LEFT        (2)          // デジタル方向ボタン左
* XINPUT_BUTTON_DPAD_RIGHT       (3)          // デジタル方向ボタン右
* XINPUT_BUTTON_START            (4)          // STARTボタン
* XINPUT_BUTTON_BACK             (5)          // BACKボタン
* XINPUT_BUTTON_LEFT_THUMB       (6)          // 左スティック押し込み
* XINPUT_BUTTON_RIGHT_THUMB      (7)          // 右スティック押し込み
* XINPUT_BUTTON_LEFT_SHOULDER    (8)          // LBボタン
* XINPUT_BUTTON_RIGHT_SHOULDER   (9)          // RBボタン
* XINPUT_BUTTON_A                (12)         // Aボタン
* XINPUT_BUTTON_B                (13)         // Bボタン
* XINPUT_BUTTON_X                (14)         // Xボタン
* XINPUT_BUTTON_Y                (15)         // Yボタン
* 戻り値：TRUE(押されてる), FALSE(押されていない)
*****/

int GetButton(int button)
{
    int ret = FALSE;

    if (button_state[button] == E_CLICK || button_state[button] == E_PRESS)
    {
        ret = TRUE;
    }

    return ret;
}

/*****
* 入力制御機能：押した瞬間か判定処理
* 引 数：XINPUTのボタン情報
* XINPUT_BUTTON_DPAD_UP          (0)          // デジタル方向ボタン上
* XINPUT_BUTTON_DPAD_DOWN        (1)          // デジタル方向ボタン下
* XINPUT_BUTTON_DPAD_LEFT        (2)          // デジタル方向ボタン左
* XINPUT_BUTTON_DPAD_RIGHT       (3)          // デジタル方向ボタン右
* XINPUT_BUTTON_START            (4)          // STARTボタン
* XINPUT_BUTTON_BACK             (5)          // BACKボタン
* XINPUT_BUTTON_LEFT_THUMB       (6)          // 左スティック押し込み
* XINPUT_BUTTON_RIGHT_THUMB      (7)          // 右スティック押し込み

```

```

* XINPUT_BUTTON_LEFT_SHOULDER      (8)          // LBボタン
* XINPUT_BUTTON_RIGHT_SHOULDER     (9)          // RBボタン
* XINPUT_BUTTON_A                   (12)         // Aボタン
* XINPUT_BUTTON_B                   (13)         // Bボタン
* XINPUT_BUTTON_X                   (14)         // Xボタン
* XINPUT_BUTTON_Y                   (15)         // Yボタン
* 戻り値：TRUE(押した瞬間)，FALSE(押した瞬間ではない)
*****/

int GetButtonDown(int button)
{
    int ret = FALSE;

    if (button_state[button] == E_CLICK)
    {
        ret = TRUE;
    }

    return ret;
}

/*****
* 入力制御機能：離れたか判定処理
* 引 数：XINPUTのボタン情報
* XINPUT_BUTTON_DPAD_UP             (0)          // デジタル方向ボタン上
* XINPUT_BUTTON_DPAD_DOWN           (1)          // デジタル方向ボタン下
* XINPUT_BUTTON_DPAD_LEFT           (2)          // デジタル方向ボタン左
* XINPUT_BUTTON_DPAD_RIGHT          (3)          // デジタル方向ボタン右
* XINPUT_BUTTON_START               (4)          // STARTボタン
* XINPUT_BUTTON_BACK                (5)          // BACKボタン
* XINPUT_BUTTON_LEFT_THUMB           (6)          // 左スティック押し込み
* XINPUT_BUTTON_RIGHT_THUMB          (7)          // 右スティック押し込み
* XINPUT_BUTTON_LEFT_SHOULDER        (8)          // LBボタン
* XINPUT_BUTTON_RIGHT_SHOULDER       (9)          // RBボタン
* XINPUT_BUTTON_A                   (12)         // Aボタン
* XINPUT_BUTTON_B                   (13)         // Bボタン
* XINPUT_BUTTON_X                   (14)         // Xボタン
* XINPUT_BUTTON_Y                   (15)         // Yボタン
* 戻り値：TRUE(押した瞬間)，FALSE(押した瞬間ではない)
*****/

int GetButtonUp(int button)
{
    int ret = FALSE;

    if (button_state[button] == E_RELEASED)
    {
        ret = TRUE;
    }
}

```

```

    return ret;
}

/*****
* 入力制御機能：ゲーム終了用のボタン判定処理
* 引 数：無し
* 戻り値：TRUE(押した瞬間), FALSE(押した瞬間ではない)
*****/
int GetExitButton(void)
{
    int ret = FALSE;

    if ((GetButtonDown(XINPUT_BUTTON_BACK) == TRUE) || (CheckHitKey(KEY_INPUT_ESCAPE) == TRUE))
    {
        ret = TRUE;
    }

    return ret;
}

```