

```

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

// 静的メンバ変数定義
bool InputControl::now_button[16] = {};
bool InputControl::old_button[16] = {};
float InputControl::trigger[2] = {};
Vector2D InputControl::stick[2] = {};

// 入力機能：更新処理
void InputControl::Update()
{
    // XInputコントローラーの入力値を取得する
    XINPUT_STATE input_state = {};
    GetJoypadXInputState(DX_INPUT_PAD1, &input_state);

    // ボタン入力値の更新
    for (int i = 0; i < 16; i++)
    {
        old_button[i] = now_button[i];
        now_button[i] = (bool)input_state.Buttons[i];
    }

    // トリガー入力値の更新 (0.0f~1.0fに範囲を制限する)
    trigger[0] = (float)input_state.LeftTrigger / (float)UCHAR_MAX;
    trigger[1] = (float)input_state.RightTrigger / (float)UCHAR_MAX;

    // 左スティック入力値の更新 (-1.0f~1.0fに範囲を制限する)
    if (input_state.ThumbLX > 0.0f)
    {
        stick[0].x = (float)input_state.ThumbLX / (float)SHRT_MAX;
    }
    else
    {
        stick[0].x = -((float)input_state.ThumbLX / (float)SHRT_MIN);
    }
    if (input_state.ThumbLY > 0.0f)
    {
        stick[0].y = (float)input_state.ThumbLY / (float)SHRT_MAX;
    }
}

```

```

    }
    else
    {
        stick[0].y = -((float)input_state.ThumbLY / (float)SHRT_MIN);
    }

    // 右スティック入力値の更新 (-1.0f~1.0fに範囲を制限する)
    if (input_state.ThumbRX > 0.0f)
    {
        stick[1].x = (float)input_state.ThumbRX / (float)SHRT_MAX;
    }
    else
    {
        stick[1].x = -((float)input_state.ThumbRX / (float)SHRT_MIN);
    }
    if (input_state.ThumbRY > 0.0f)
    {
        stick[1].y = (float)input_state.ThumbRY / (float)SHRT_MAX;
    }
    else
    {
        stick[1].y = -((float)input_state.ThumbRY / (float)SHRT_MIN);
    }
}

// ボタン取得：押してる間
bool InputControl::GetButton(int button)
{
    return CheckButtonRange(button) && (now_button[button] &&
old_button[button]);
}

// ボタン取得：押した瞬間
bool InputControl::GetButtonDown(int button)
{
    return CheckButtonRange(button) && (now_button[button] &&
!old_button[button]);
}

```

```
}

// ボタン取得：離れた瞬間
bool InputControl::GetButtonUp(int button)
{
    return CheckButtonRange(button) && (!now_button[button] &&
old_button[button]);
}

// 左トリガー取得
float InputControl::GetLeftTrigger()
{
    return trigger[0];
}

// 右トリガー取得
float InputControl::GetRightTrigger()
{
    return trigger[1];
}

// 左スティック取得
Vector2D InputControl::GetLeftStick()
{
    return stick[0];
}

// 右スティック取得
Vector2D InputControl::GetRightStick()
{
    return stick[1];
}

// ボタン配列範囲チェック
bool InputControl::CheckButtonRange(int button)
{
    return (0 <= button && button < 16);
}
```

