```
#pragma once
#ifdef _DEBUG
#include <windows.h>
#include <memory>
#include <tuple>
#include <chrono>        // 精度は1msec程度

#define _dbgSetup(A,B,C)                    _DebugDispOut::GetInstance().Setup(A,B,C)
#define _dbgSetAlpha(A)                       _DebugDispOut::GetInstance().SetAlpha(A)
#define _dbgStartDraw()                       _DebugDispOut::GetInstance().StartDrawDebug()
#define _dbgAddDraw()                        _DebugDispOut::GetInstance().AddDrawDebug()
#define _dbgDrawGraph(fmt, ...)               _DebugDispOut::GetInstance().DrawGraph(fmt, __VA_ARGS__)
#define _dbgDrawBox(fmt, ...)                _DebugDispOut::GetInstance().DrawBox(fmt, __VA_ARGS__)
#define _dbgDrawLine(fmt, ...)                _DebugDispOut::GetInstance().DrawLine(fmt, __VA_ARGS__)
#define _dbgDrawCircle(fmt, ...)             _DebugDispOut::GetInstance().DrawCircle(fmt, __VA_ARGS__)
#define _dbgDrawPixel(fmt, ...)               _DebugDispOut::GetInstance().DrawPixel(fmt, __VA_ARGS__)
#define _dbgDrawString(fmt, ...)             _DebugDispOut::GetInstance().DrawString(fmt, __VA_ARGS__)
#define _dbgDrawFormatString(fmt, ...)        _DebugDispOut::GetInstance().SetScreen(); ¥
                                            DxLib::DrawFormatString(fmt, __VA_ARGS__);¥
                                            _DebugDispOut::GetInstance().RevScreen()
#define _dbgSetDrawPosFps(fmt, ...)           _DebugDispOut::GetInstance().SetDrawPosFps(fmt, __VA_ARGS__)
#define _dbgDrawFPS()                        _DebugDispOut::GetInstance().DrawFPS()

using ChronoSysClock = std::chrono::system_clock::time_point;

enum class FPS_SIDE
{
    LEFT,
    RIGHT
};


enum class FPS_VER
{
    TOP,
    BOTTOM
};

class _DebugDispOut
{
public:
    static _DebugDispOut &GetInstance()
    {
        return (*s_Instance);
    }
    int DrawGraph(int x, int y, int GrHandle, int TransFlag);
    int DrawBox(int x1, int y1, int x2, int y2, unsigned int Color, int FillFlag);
    int DrawString(int x, int y, char* String, unsigned int Color);
```

```
//    int DrawFormatString(int x, int y, unsigned int Color, std::string FormatString, ...);
    int DrawLine(int x1, int y1, int x2, int y2, unsigned int Color);
    int DrawCircle(int x, int y, int r, unsigned int Color, int FillFlag);
    int DrawPixel(int x, int y, unsigned int Color);
    void DrawFPS(void);
    void SetDrawPosFps(FPS_SIDE side, FPS_VER ver);
    bool StartDrawDebug(void);
    bool AddDrawDebug(void);
    bool SetAlpha(int alpha);
    bool Setup(int screenSizeX,int screenSizeY, int alpha);
    bool SetWait(double timeCnt);
    void SetScreen(void);
    void RevScreen(void);
    void WaitMode(void);
private:
    struct _DebugDispOutDeleter
    {
        void operator()(_DebugDispOut* _debugContOut) const
        {
            delete _debugContOut;
        }
    };

    _DebugDispOut();
    ~_DebugDispOut();
    static std::unique_ptr<_DebugDispOut, _DebugDispOutDeleter> s_Instance;
    int _alpha;
    ChronoSysClock fpsStartTime_;
    ChronoSysClock fpsEndTime_;
    ChronoSysClock  startTime_;
    ChronoSysClock  endTime_;
    double waitTime_;
    bool dispFlag_;
    int ghBefor_;
    bool clsFlag_;
    int endKey_[2];
    int pouseKey_[2];
    int homeKey_[2];
    int f1Key_[2];
    int backSp_[2];
    int DbgScreen_;
    int fpsView_;
    int fpsCount_;
    int fpsPosX;
    int fpsPosY;
    int screenSizeX_;
    int screenSizeY_;
};
```

```
#else

#define _dbgSetup(A)
#define _dbgSetAlpha(A)
#define _dbgStartDraw()
#define _dbgAddDraw()
#define _dbgDrawGraph(fmt, ...)
#define _dbgDrawBox(fmt, ...)

#endif    // _DEBUG
```