

Некоторые методы проектирования и оптимизации кода

Антон Зачесов

Video Group
CS MSU Graphics & Media Lab



Содержание

- **Зачем все это нужно?**
- Проектирование проекта
- Оптимизация кода
- Полезное в работе с Visual Studio

Описание проблемы (1/2)



**ТАК БЫЛО ПОСТАВЛЕНО
ТЕХНИЧЕСКОЕ ЗАДАНИЕ**



**ТАК ЕГО ПОНЯЛИ
РАЗРАБОТЧИКИ**



**ТАК ЭТУ ЗАДАЧУ
РЕШАЛИ РАНЬШЕ**



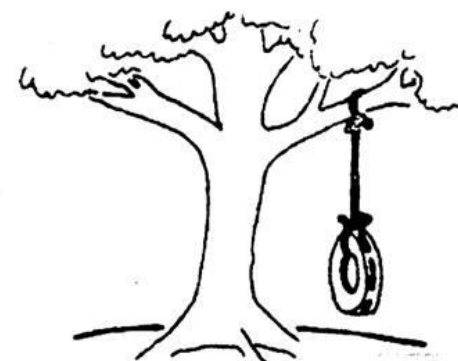
ТАК ЕЕ РЕШИЛИ ТЕПЕРЬ



**ТАКОЙ ПРОГРАММА СТАЛА
ПОСЛЕ ОТЛАДКИ**



**ТАК ЕЕ ОПИСАЛИ
В ОТДЕЛЕ РЕКЛАМЫ**



**А, СОБСТВЕННО, ТАК
ЕЕ ПРЕДСТАВЛЯЛ СЕБЕ
ЗАКАЗЧИК**

Описание проблемы (2/2)

Следствия отсутствия (либо некорректного) проектирования/оптимизации:

- Приложение работает неочевидно (делает не то, что вы ожидаете)
- Приложение работает медленнее, чем должно (или вы бы хотели)
- Код приложения настолько ужасен, что хочется его выкинуть и переписать заново

Постановка проблемы

Кажущиеся способы решения

- Увеличить сроки разработки
- Увеличить число людей в проекте
- Обеспечить программистов адекватными требованиями к разработке

В 90% случаев эти способы не сработают

Постановка требований

Пример

«Хочу в приложении кнопку сделать хорошо»,
и чтобы она всегда работала»

Или более адекватный:

«Нужно сделать плагин синхронизации
стереовидео во времени
(с учетом смещения в 0,5 кадра)»

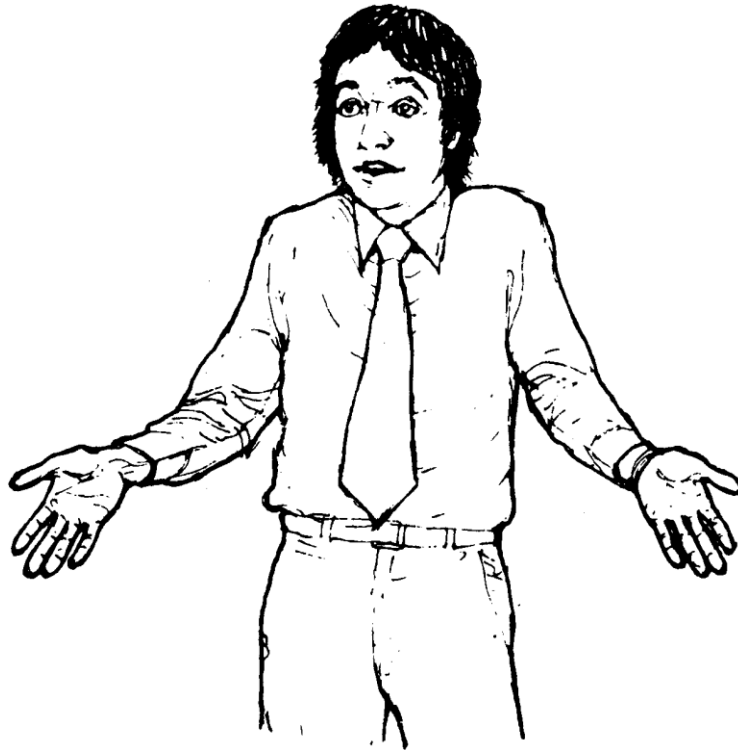


Черты идеального продукта

- Соответствие целям пользователя
- Понятность и очевидность поведения
- Стабильность
- Быстрый отклик на действия пользователя
- Адекватная производительность

Черты идеального проекта

Как приблизиться к идеалу?



Разобраться в этом – цель данного доклада



Содержание

- Зачем все это нужно?
- **Проектирование проекта**
 - Проектирование взаимодействия
 - Проектирование кода
- Оптимизация кода
- Полезное в работе с Visual Studio



Проектирование

«Проектирование – то, чем программисты занимаются 20 минут перед тем, как начать писать код»
(бородатый анекдот)

Проектирование

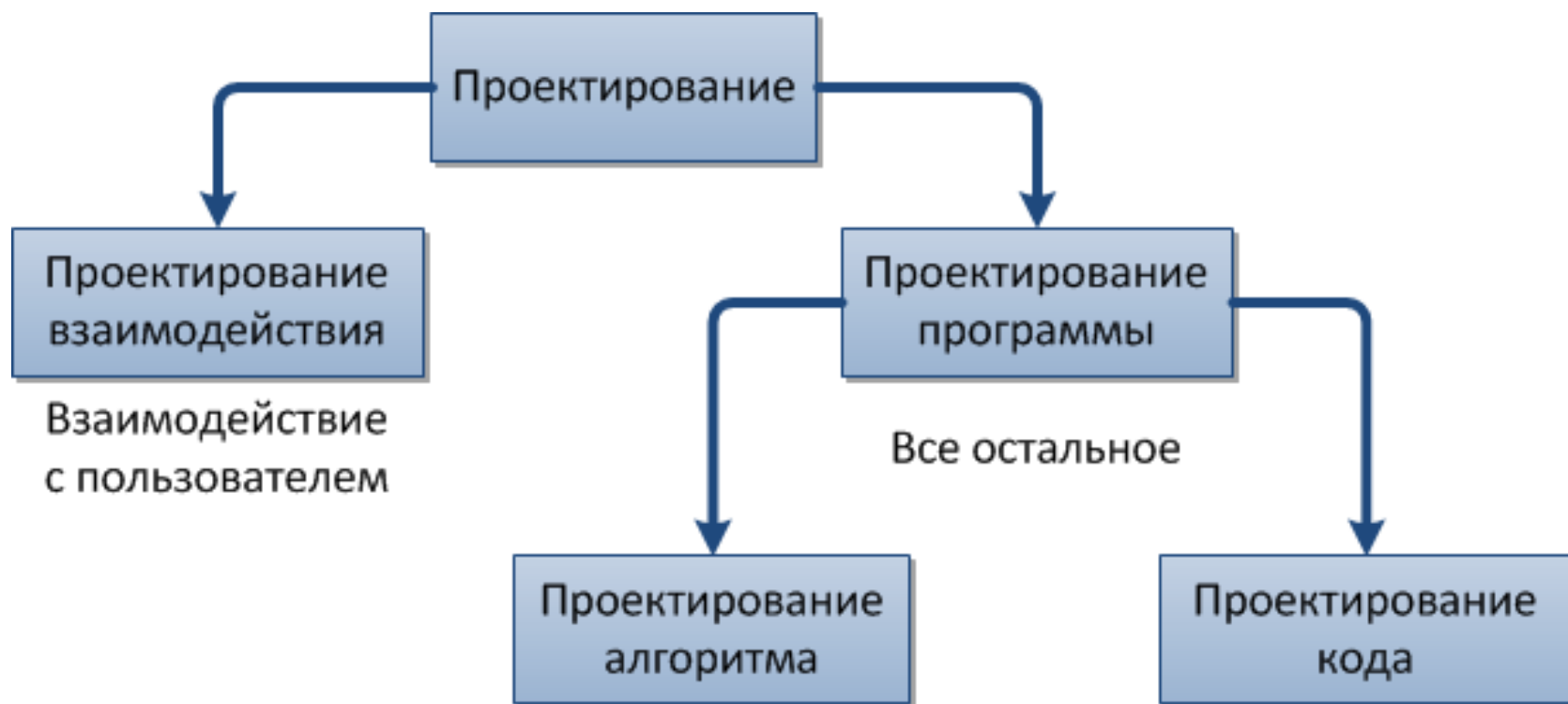
Суть

Весь процесс создания программы от постановки требований до начала написания кода

Цели:

- Определить usecases программы
- Найти контакт с пользователем программы
- Облегчить жизнь программистам-потомкам

Виды проектирования



Виды проектирования

Различия

- Проектирование кода (фактически, программирование) полностью на совести программиста
- Проектирование взаимодействия программист не в состоянии сделать один
«Программирование определяется краевыми случаями, а проектирование – центральными»
(Алан Купер)

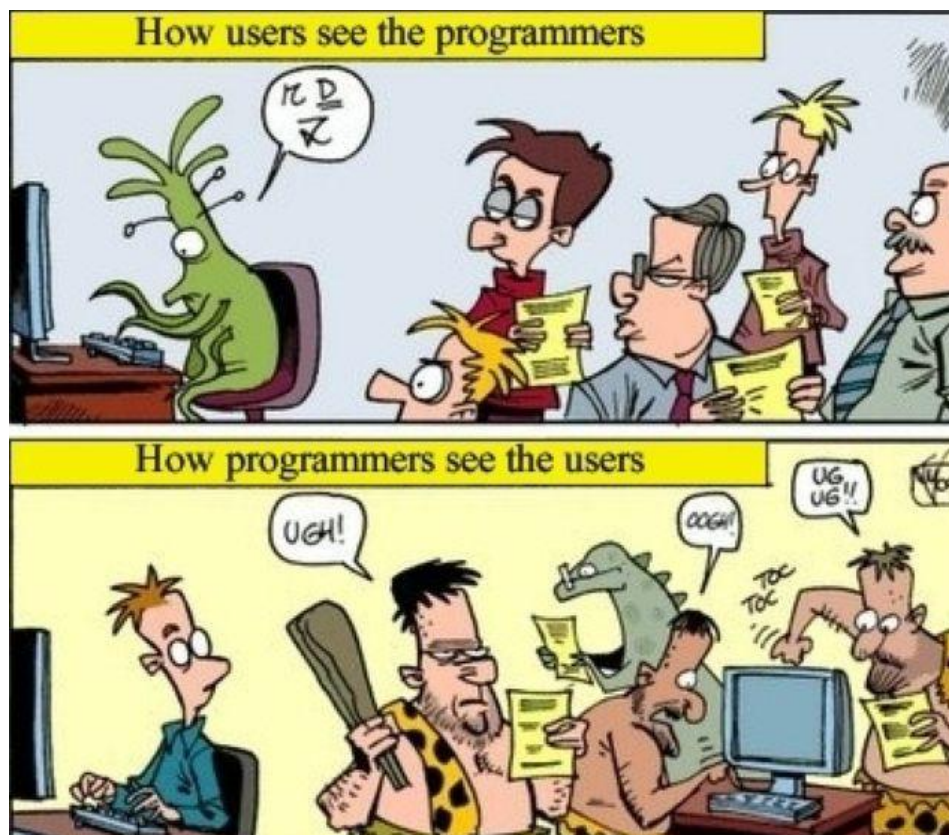


Содержание

- Зачем все это нужно?
- Проектирование проекта
 - **Проектирование взаимодействия**
 - Проектирование кода
- Оптимизация кода
- Полезное в работе с Visual Studio

Проектирование взаимодействия

Цель – преодолеть подобную ситуацию



Проектирование взаимодействия «Танцующий медведь»

«Танцует медведь просто ужасно,
но чудо не в том, что он танцует
хорошо, а в том, что он вообще
танцует»
(Алан Купер)

Суровая реальность в том, что
многие приложения похожи
на «танцующих медведей»



Проектирование взаимодействия

Основная цель

- Главное – понять цели пользователя и сделать их достижение возможным
- Выбрать, какие функции могут понадобиться пользователю для достижения цели, – не главное. Они могут не соответствовать целям

Проектирование взаимодействия Персонажи

Задача – максимально подробно описать
пользователя программы



Недостаточно подробно



Достаточно подробно

Проектирование взаимодействия

Сценарии

Цель – получить минимальное количество проблем с основными функциями

Задача – максимально детально описать процесс регулярного использования программы

Вспоминаем:

«Программирование определяется краевыми случаями, а проектирование – центральными»
(Алан Купер)



Содержание

- Зачем все это нужно?
- Проектирование проекта
 - Проектирование взаимодействия
 - **Проектирование кода**
- Оптимизация кода
- Полезное в работе с Visual Studio



Проектирование кода

«Есть два способа разработки проекта:
сделать его настолько простым, чтобы было
очевидно, что в нем нет недостатков, или
сделать его настолько сложным, чтобы в нем
не было очевидных недостатков»
(С. А. Р. Хоар)

Проектирование кода

Мое восприятие

- Проектирование кода – всегда изобретение велосипеда
- Важно не впадать в крайности:



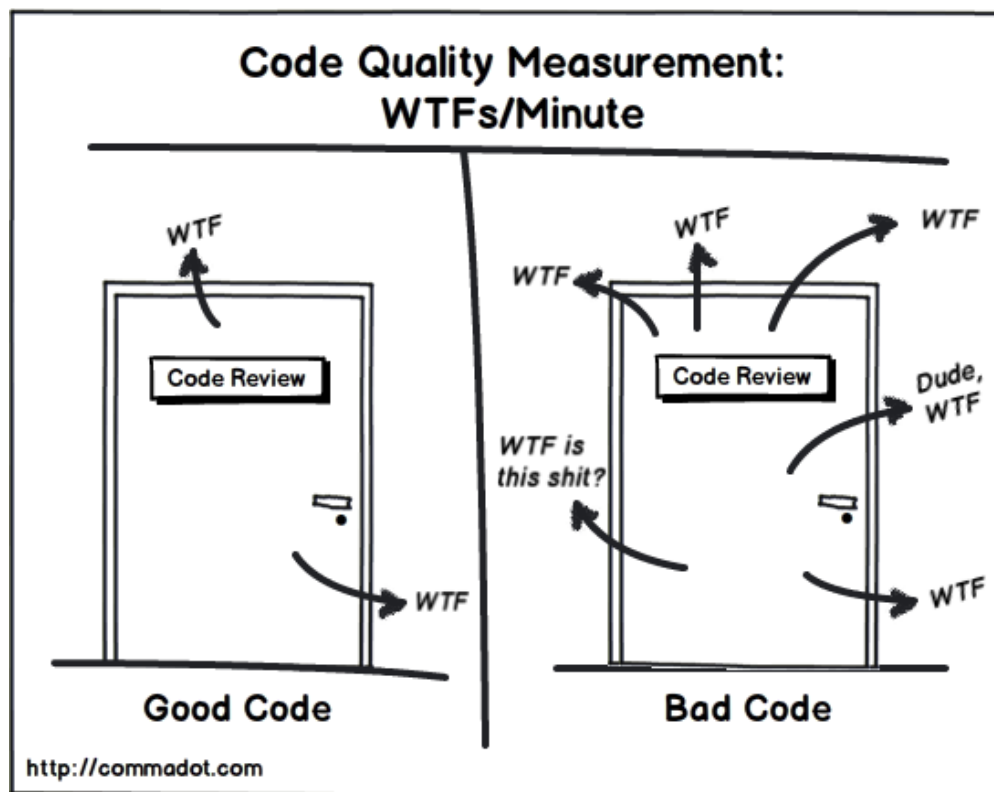
...



Проектирование кода

Цели (1/3)

Одна из целей – оказаться слева



Проектирование кода

Цели (2/3)

Основные цели:

- реализация всех требований к проекту
- реализация всех требований проектирования взаимодействия
- ничего больше

Побочные цели:

- понятность и читабельность кода
- устойчивость к модификациям

обычно способствуют основной

Проектирование кода

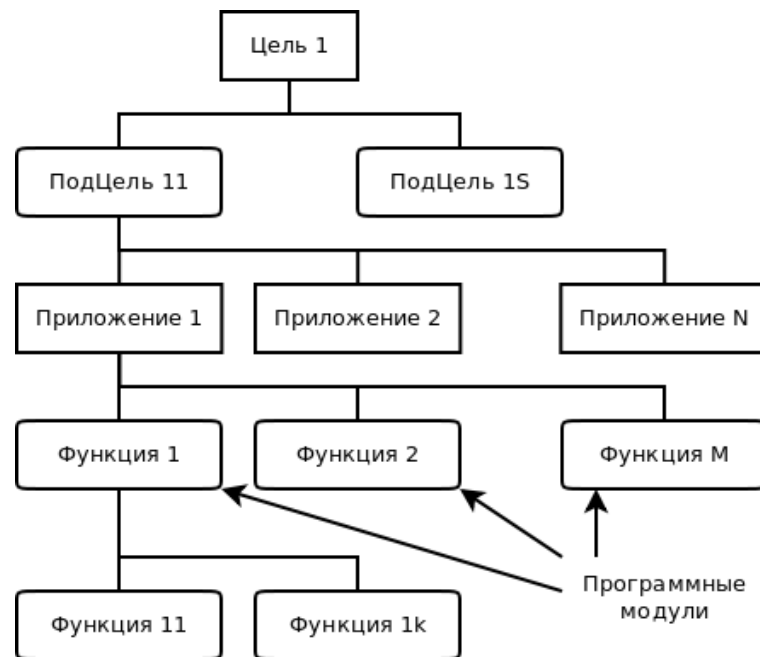
Цели (3/3)

«Книга закончена не тогда,
когда в нее нечего добавить,
а тогда, когда из нее нечего выбросить»
(Алан Купер)

Проектирование кода

Модели

- **Нисходящая**
Проектирование от идеи к реализации
- **Восходящая**
Проектирование от деталей конструктора к результату
- **Смешанная**
Тоже работает



Нисходящее проектирование




Проектирование алгоритма

Из личного опыта:

- Простые идеи работают чаще, чем сложные
- Очень полезным является прототипирование

«Прототипирование – написание абсолютно минимального объема подлежащего выбрасыванию кода, нужного для ответа на отдельный вопрос проектирования»
(Алан Купер)

Прототипирование

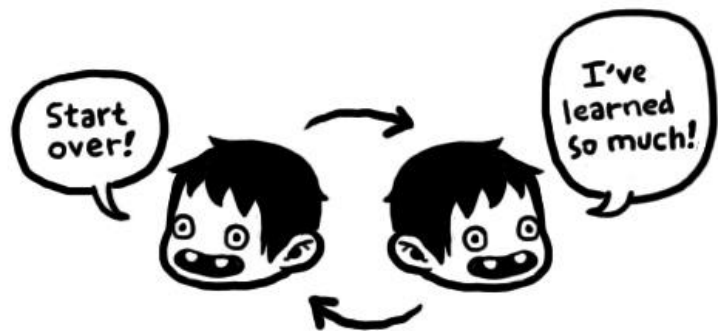
- Лучше использовать не ту среду, в которой пишется финальный продукт
 - Чаще всего это MATLAB 
 - В своей практике я использовал Adobe After Effects  и Photoshop 

Если что-то пошло не так:

«Если сомневаетесь, используйте грубую силу» (Батлер Пэмпсон)

Повторное использование кода

- Очень полезно
- Тем не менее, крайности опасны



...





Понятность кода

«Я никогда не встречал человека, желающего читать 17000 страниц документации, а если бы встретил, то убил бы его»
(Джозеф Кастелло)

Вывод:

Лучшая документация кода – сделать его читабельным и понятным в первую очередь для других программистов



Содержание

- Зачем все это нужно?
- Проектирование проекта
- **Оптимизация кода**
- Статический анализ кода
- Полезное в работе с Visual Studio

Оптимизация

Цель – избежать подобной ситуации:



Оптимизация

Методы

- Высокоуровневая
 - Использование кэширования
 - Предоставление быстрого отклика на запрос / прогресс-бара
 - Алгоритмические оптимизации
- Низкоуровневая

Оптимизация

Низкоуровневая

- Метод «орлиного взора»
- Использование профилировщиков

Профилировка – измерение производительности программы с целью нахождения Hot Spots (участков программы, на выполнение которых расходуется наибольшее количество времени)

Оптимизация

Примеры профилировщиков

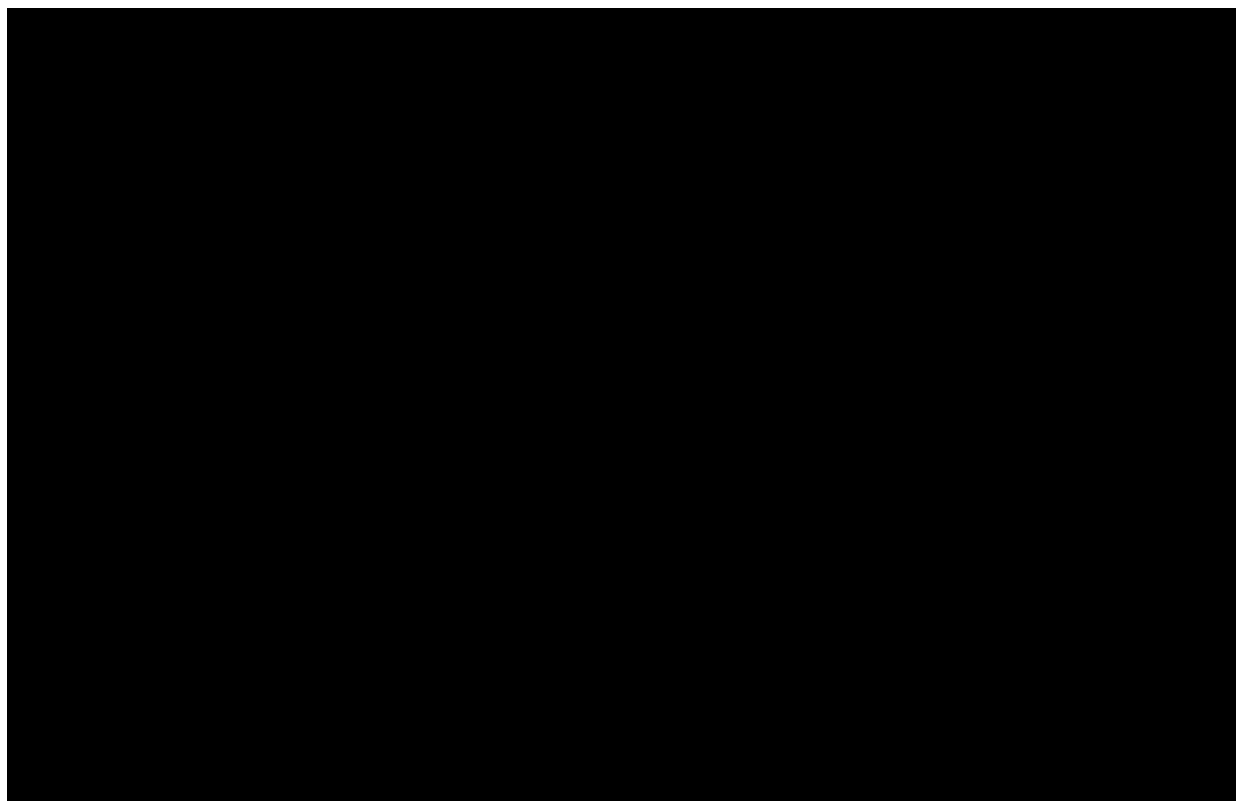
- Visual Studio Ultimate
- Intel Parallel Amplifier
- AQTime
- ...



Профилировщик

Пример использования

Профилировка библиотеки StereoToMultiview



Пример оптимизации

Результат профилировки

Hotspots

Bottom-up Top-down Tree

Function - Caller Function Tree	CPU Time:Self	Module
⊕ Erosion_Value_BYTE	9.422s	StereoToMultiviewConsole.exe
⊕ Dilation_Value_BYTE	4.834s	StereoToMultiviewConsole.exe
⊕ MakePlanes	1.727s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_16x16_H264_MMX32	1.178s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_8x8_H264_MMX32	0.677s	StereoToMultiviewConsole.exe
⊕ fastcopy_I	0.558s	StereoToMultiviewConsole.exe
⊕ malloc	0.516s	StereoToMultiviewConsole.exe
⊕ FilterMVList	0.514s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_4x4_H264_MMX32	0.400s	StereoToMultiviewConsole.exe
⊕ fastzero_I	0.358s	StereoToMultiviewConsole.exe
⊕ SwapAndCleanCands	0.334s	StereoToMultiviewConsole.exe

Пример оптимизации

Шаг 1

Проверка настроек оптимизации компилятора

Было:

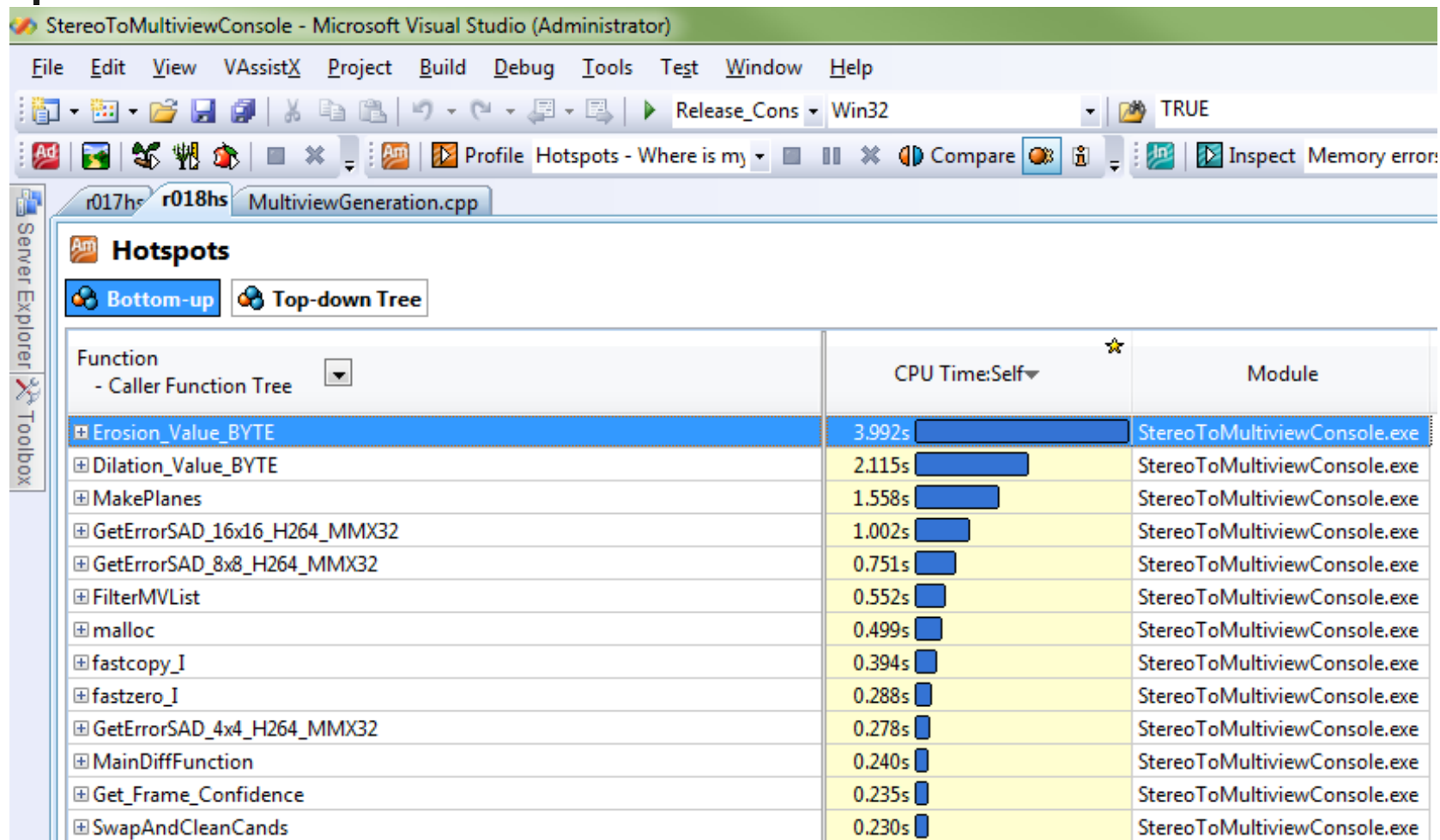
Optimization	Disabled (/Od)
Inline Function Expansion	Default
Enable Intrinsic Functions	No
Favor Size or Speed	Neither
Omit Frame Pointers	No
Enable Fiber-safe Optimizations	No
Whole Program Optimization	No

Стало:

Optimization	Maximize Speed (/O2)
Inline Function Expansion	Any Suitable (/Ob2)
Enable Intrinsic Functions	Yes (/Oi)
Favor Size or Speed	Favor Fast Code (/Ot)
Omit Frame Pointers	No
Enable Fiber-safe Optimizations	No
Whole Program Optimization	No

Пример оптимизации

Результат шага 1



Пример оптимизации

Шаг 2

Исправление указанных профилировщиком мест

374	for(i = 0; i < h; i++)	
375	for(j = 0; j < w; j++)	
376	{	
377	index0 = i * w + j;	
378		
379	min_value = src[index0];	0.040s
380	for(k = 0; k < d*d; k++)	0.274s
381	{	
382	m = i + (k / d) - r;	1.644s
383	n = j + (k % d) - r;	0.471s
384	if(INNER_POINT(m, n, h, w) && wnd_mask[k])	1.092s
385	{	
386	index = m * w + n;	0.097s
387	if(src[index] < min_value)	0.353s
388	min_value = src[index];	
389	}	
390	}	
391		

Пример оптимизации

Исправление кода

```
min_value = src[index0];
/*for(k = 0; k < d*d; k++)
{
    m = i + (k / d) - r;
    n = j + (k % d) - r;*/
k = 0;
for(m = i-r; m <= i+r; m++)
for(n = j-r; n <= j+r; n++)
{
    if(INNER_POINT(m, n, h, w) && wnd_mask[k])
    {
        index = m * w + n;
        if(src[index] < min_value)
            min_value = src[index];
    }
    ++k;
}
```

Пример оптимизации

Результат шага 2

Hotspots

Bottom-up Top-down Tree

Function - Caller Function Tree	CPU Time:Self	Module
⊕ Erosion_Value_BYTE	2.386s	StereoToMultiviewConsole.exe
⊕ MakePlanes	1.652s	StereoToMultiviewConsole.exe
⊕ Dilation_Value_BYTE	1.269s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_16x16_H264_MMX32	1.225s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_8x8_H264_MMX32	0.670s	StereoToMultiviewConsole.exe
⊕ malloc	0.462s	StereoToMultiviewConsole.exe
⊕ FilterMVList	0.455s	StereoToMultiviewConsole.exe
⊕ fastcopy_I	0.356s	StereoToMultiviewConsole.exe
⊕ MainDiffFunction	0.318s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_4x4_H264_MMX32	0.296s	StereoToMultiviewConsole.exe
⊕ Get_Frame_Confidence	0.248s	StereoToMultiviewConsole.exe

Пример оптимизации

Финальный вариант кода

```
for(i = 0; i < h; i++)
for(j = 0; j < w; j++)
{
    index0 = i * w + j;

    min_value = src[index0];

    if(INSIDE(i, r, h-r-1) && INSIDE(j, r, w-r-1) )
    {
        k = 0;
        for(m = i-r; m <= i+r; ++m)
        {
            for(n = j-r; n <= j+r; ++n)
            {
                //k = (m-i+r)*d + (n-j+r);

                if(/*INNER_POINT(m, n, h, w) &&*/ wnd_mask[k])
                {
                    index = m * w + n;
                    if(src[index] < min_value)
                        min_value = src[index];
                }
            }
            ++k;
        }
    }
}
```

Пример оптимизации

Результат

Hotspots

Bottom-up Top-down Tree

Function - Caller Function Tree	CPU Time:Self	Module
MakePlanes	1.864s	StereoToMultiviewConsole.exe
Erosion_Value_BYTE_Faster	1.391s	StereoToMultiviewConsole.exe
GetErrorSAD_16x16_H264_MMX32	1.023s	StereoToMultiviewConsole.exe
GetErrorSAD_8x8_H264_MMX32	0.777s	StereoToMultiviewConsole.exe
Dilation_Value_BYTE_Faster	0.714s	StereoToMultiviewConsole.exe
FilterMVList	0.490s	StereoToMultiviewConsole.exe
malloc	0.448s	StereoToMultiviewConsole.exe
GetErrorSAD_4x4_H264_MMX32	0.438s	StereoToMultiviewConsole.exe
fastcopy_I	0.278s	StereoToMultiviewConsole.exe
Get_Frame_Confidence	0.266s	StereoToMultiviewConsole.exe

Пример оптимизации

Исходный вариант

Hotspots

Bottom-up | Top-down Tree

Function - Caller Function Tree	CPU Time:Self	Module
⊕ Erosion_Value_BYTE	9.422s	StereoToMultiviewConsole.exe
⊕ Dilation_Value_BYTE	4.834s	StereoToMultiviewConsole.exe
⊕ MakePlanes	1.727s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_16x16_H264_MMX32	1.178s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_8x8_H264_MMX32	0.677s	StereoToMultiviewConsole.exe
⊕ fastcopy_I	0.558s	StereoToMultiviewConsole.exe
⊕ malloc	0.516s	StereoToMultiviewConsole.exe
⊕ FilterMVList	0.514s	StereoToMultiviewConsole.exe
⊕ GetErrorSAD_4x4_H264_MMX32	0.400s	StereoToMultiviewConsole.exe
⊕ fastzero_I	0.358s	StereoToMultiviewConsole.exe
⊕ SwapAndCleanCands	0.334s	StereoToMultiviewConsole.exe



Содержание

- Зачем все это нужно?
- Проектирование проекта
- Оптимизация кода
 - Оптимизация кода
 - **Статический анализ кода**
- Полезное в работе с Visual Studio

Статический анализ кода

Идеи



“The first step is fully admitting that the code you write is riddled with errors”
(John Karmak)

“The more I push code through static analysis, the more I’m amazed that computers boot at all”
(Dave Revell)

Статический анализ кода

- Static Code Analysis – анализ кода, проводимый без реального выполнения программы
- Поиск очевидных (и не очень) ошибок в процессе компиляции кода
- Мной используется PVS-Studio



Содержание

- Зачем все это нужно?
- Проектирование проекта
- Оптимизация кода
- **Полезное в работе с Visual Studio**



Visual Studio

- Позволяет эффективно разделять код на проекты (библиотечные и исполняемые модули)
- Используется в проектах лаборатории
- Чаще всего – работа с многими проектами одновременно

Настройки Visual Studio

Project Dependencies

- Некорректная настройка:
Ошибки вида `Unresolved External`
или `Duplicate Identifier`
- Глубинный смысл:
Код проектов, имеющих зависимости,
генерируется позже и может явно
ссылаться на уже сгенерированный код
проектов-зависимостей

Настройки Visual Studio

Runtime Library

- Некорректная настройка:
Ошибки вида
`... already defined in LIBCMT.lib`
и `unresolved external`
- Глубинный смысл:
При выборе `*dll` собранной программе добавляется зависимость от системных динамических библиотек, например, `MSVCR90.dll`

Настройки Visual Studio

Настройка Debug

- В исполняемом коде нужна информация об исходном коде
- Необходимо настроить компиляцию кода с информацией для дебага (Debug Information)
- Необходимо включить опцию линковки с информацией для дебага (Generate Debug Info)

Настройки Visual Studio

Link Library Dependencies

- Некорректная настройка
Ошибки вида `unresolved external`
- Глубинный смысл:
Бинарный код всех библиотек-зависимостей копируется в главную библиотеку
- Следствие:
Иногда линковка занимает много времени

Настройки Visual Studio

Compile As

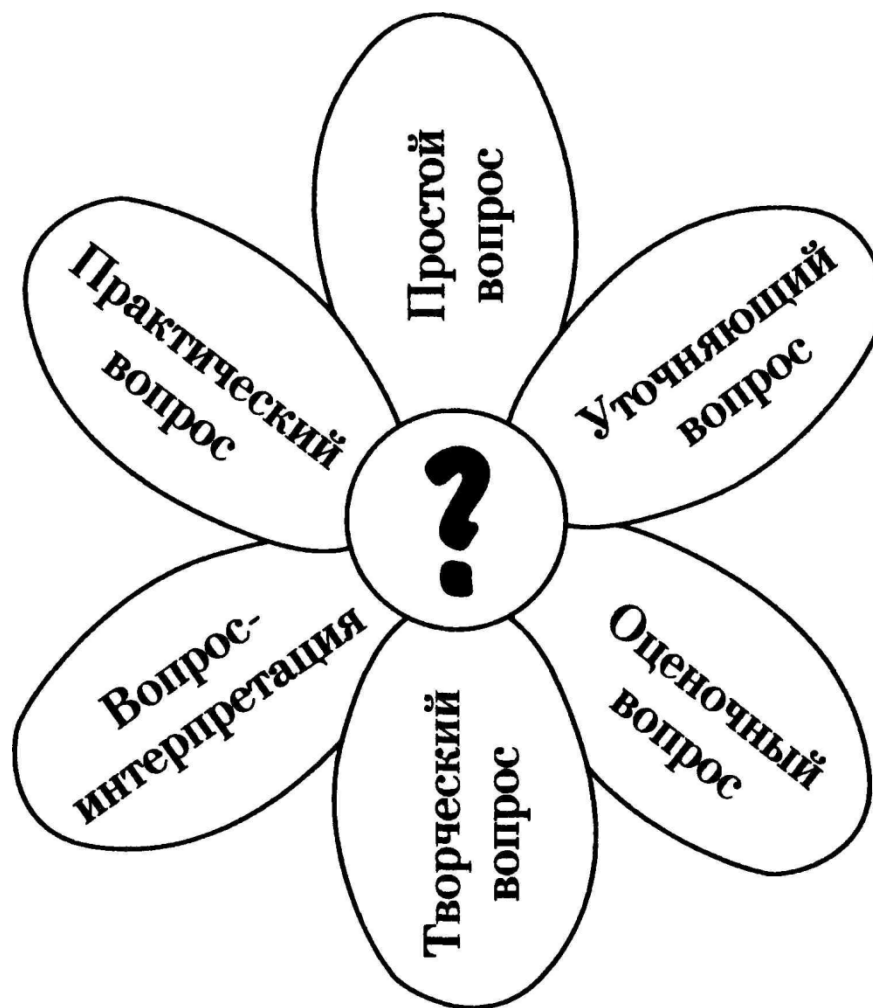
- Некорректная настройка:
Ошибки при компиляции C++ инструкций по правилам C
- Глубинный смысл:
Вообще говоря, просто выбор компилятора для конкретного файла



Литература

1. А. Купер, "Психбольница в руках пациентов".
2. С. Макконнелл, "Совершенный код".

Вопросы?



Лаборатория компьютерной графики и мультимедиа



Видеогруппа — это:

- Выпускники в аспирантурах Англии, Франции, Швейцарии (в России в МГУ и ИПМ им. Келдыша)
- Выпускниками защищены 5 диссертаций
- Наиболее популярные в мире сравнения видеокодеков
- Более 3 миллионов скачанных фильтров обработки видео