



13 a 17 de Outubro de 2008 - ICMC - USP São Carlos

Desenvolvimento Rápido de Jogos em 3D Usando o Dark GDK com o Microsoft Visual C/C++ (Game RAD - *Rapid Application Development*)

Prof. Fernando Santos Osório

Profa. Kalinka Castelo Branco

Email: fosorio [at] { icmc. usp. br , gmail. com }

kalinka [at] { icmc. usp. br }

Material Complementar

Página Web: <http://www.icmc.usp.br/~fosorio/>

Agenda



Temas Abordados:

- 8h00-12h00 Introdução ao Desenvolvimento Rápido de Jogos
Ferramenta Dark GDK / TheGameCreators
Demonstração: Exemplos de Jogos feitos com o GDK
Introdução a Linguagem Dark GDK – Programação em C++
Hands-On: Criando Jogos em 3D!
- 14h-18h00 Projeto de um Jogo e Ferramentas Complementares
> Design Doc., Modelagem 3D, Som, Texturas, Animações
Técnicas de Desenvolvimento de Jogos
> Cenário: Skybox, Terrenos, BSP
> Atores: Movimentação, Animação, Ações, Comportamento
> Colisão: Cenário, Objetos, Atores, Gravidade
Jogos Multi-Player
> Conexão em Rede: P2P, Cliente-Servidor, MMOG



Simple DirectMedia Layer (SDL)

- SDL – *Simple DirectMedia Layer*
- API de desenvolvimento multimídia
 - Inclusive games!
- Licença GNU LGPL – pode ser usada em softwares proprietários
- Multiplataforma – Alta portabilidade
- Escrita em C, integrada com C++ e com binding para diversas linguagens: Java, Lua, PHP, Python, Ada,...
- Possui integração com a OpenGL



Simple DirectMedia Layer (SDL)

- Oferece as seguintes funcionalidades:
 - Gráficos/Vídeos
 - Tratamento de eventos (teclado/*mouse/joystick*)
 - Áudio (permite mixagem) e áudio em CDROM
 - Temporizadores
 - **Sockets – jogos em rede**
 - **Threads**



SDL – Quem utiliza??

- Bem, Quem não utiliza?
- Quase a maioria dos jogos
- Tuxracer, Chromium, glTron
- Civilization: Call To Power, HOMM3, Myth II, SMAC, SoF, Tribes
- Podemos até arriscar a dizer que você poderia inserir seu jogo aqui nesta lista.



Por que SDL??

- Ela é leve, simples e possui uma API “limpa”
- Portável para diversos Sistemas Operacionais
- Ela é simples.



SDL - Inicialização

- Inicialização Básica é feita por ***SDL_Init()***
- Pega como argumentos os subsistemas para inicialização:
 - ***SDL_Init(SDL_INIT_AUDIO | SDL_INIT_VIDEO)***;
 - ***SDL_Init(SDL_INIT EVERYTHING)***
- Retorna *flags* indicando sucessos nas operações
- Inicialização de subsistemas podem ser feitos por ***SDL_InitSubsystem()***



SDL - Bibliotecas

- Por si só SDL é apenas uma biblioteca gráfica simples.
- Para propostas diferentes (necessidades diferentes) fazem-se necessárias outras bibliotecas – Extensões da SDL:
 - **SDL_image** – utilizada para carregar outros tipos de imagens que não BMP
 - **SDL_ttf** – fontes *true type* podem ser carregadas
 - **SDL_mixer** – fornece suporte a músicas em diferentes formatos
 - **SDL_net** - *fornece API para suporte a Rede*
 - **SMPEG** – fornecer a seqüência do filme



SDL_Net – Suporte a Rede

- **SDL_net** – biblioteca portátil para ser utilizada com a SDL.
- O **SDL_net** está disponível sob a licença *GNU General Public Library*. A API pode ser encontrada no arquivo **SDL_net.h**.
- O objetivo desta parte do curso é o de ilustrar como usar **SDL_net** para permitir que suas aplicações (jogos) utilizem a capacidade de rede



SDL_Net - Tipos

- A API SDL_net define e utiliza os seguintes tipos:

Tipo	Descrição
IPAddress	Endereço IP e número de porta
TCPsocket	TCP socket type
UDPsocket	UDP socket type
UDPpacket	UDP packet encapsulation
SDLNET_SocketSet	Socket set type
SDLNET_GenericSocket	Generic type for UDP and TCP sockets



SDL_Net - Funções

- A API SDL_net provê as seguintes funções:
 - Inicialização
 - **SDLNet_Init:**
int SDLNet_Init()
//Start up SDL_net functionality.
 - **SDLNet_Quit:**
void SDLNet_Quit()
//Stop SDL_net functionality.



SDL_Net - Funções

- Errors
 - **SDLNet_GetError:**
char *SDLNet_GetError()
Get the current error string.



SDL_Net - Funções

- Data helpers
 - **SDLNet_Write16**
void SDLNet_Write16(UINT16 value, void *area)
Put a 16-bit number in the network-ordered data.
 - **SDLNet_Write32**
void SDLNet_Write32(UINT32 value, void *area)
Put a 32-bit number in the network-ordered data.
 - **SDLNet_Read16**
UINT16 SDLNet_Read16(void *area)
Get a 16-bit number from the network-ordered data.
 - **SDLNet_Read32**
UINT32 SDLNet_Read32(void *area)
Get a 32-bit number from the network-ordered data.



SDL_Net - Funções

- Name resolution

Essas funções são utilizadas para resolver *hostnames* e endereço numérico IPv4 (um para o outro)

- **SDLNet_ResolveHost:**

int SDLNet_ResolveHost(IPAddress *address, char *host, Uint16 port)

Resolve the string host, and fill in the IP address pointed by the address with the resolved IP and the port number that was passed in through the port.

- **SDLNet_ResolveIP:**

char *SDLNet_ResolveIP(IPAddress *address)

Resolve the IPv4 numeric address in address->host, and return the hostname.



SDL_Net - Funções

- TCP sockets

Funções utilizadas para trabalhar com o TCP sockets. TCP é utilizado com uma conexão, enquanto UDP sem. TCP garante a chegada dos pacotes nos destinos. TCP garante o recebimento em ordem.

- **SDLNet_TCP_Open:**

TCPsocket SDLNet_TCP_Open(IPAddress *ip)

Open a TCP client or server socket.

- **SDLNet_TCP_Close:**

void SDLNet_TCP_Close(TCPsocket sock)

Close a TCP socket.

- **SDLNet_TCP_Accept:**

TCPsocket SDLNet_TCP_Accept(TCPsocket server)

Accept a connection on a server socket. This function is non-blocking.



SDL_Net - Funções

- TCP sockets

- **SDLNet_TCP_GetPeerAddress:**
IPAddress *SDLNet_TCP_GetPeerAddress(TCPsocket sock)
Get the remote host address and port number.
- **SDLNet_TCP_Send:**
int SDLNet_TCP_Send(TCPsocket sock, void *data, int len)
Send data over a connected socket.
- **SDLNet_TCP_Recv:**
int SDLNet_TCP_Recv(TCPsocket sock, void *data, int maxlen)
Receive data from a connected socket. This function does not wait to receive maximum length of bytes, it can return before the maximum length is reached. This function is blocking.



SDL_Net - Funções

- UDP sockets

Funções utilizadas para UDP Sockets.

- **SDLNet_UDP_Open:**
UDPsocket SDLNet_UDP_Open(UINT16 port)
Create a UDP socket. If the port is not 0, the bind is done in this function.
- **SDLNet_UDP_Close:**
void SDLNet_UDP_Close(UDPsocket sock)
Close and free a UDP socket.
- **SDLNet_UDP_Bind:**
int SDLNet_UDP_Bind(UDPsocket sock, int channel, IPAddress*address)
Assign an IP address number to a socket channel.
- **SDLNet_UDP_Unbind:**
void SDLNet_UDP_Unbind(UDPsocket sock, int channel)
Remove all assigned IP addresses from a socket channel.



SDL_Net - Funções

- UDP packets
The following functions are used to work with the UDPpacket type. This type is used with UDP sockets to transmit and receive data.
- **SDLNet_AllocPacket:**
UDPpacket *SDLNet_AllocPacket(int size)
Allocate a new UDP packet with a data buffer.
- **SDLNet_ResizePacket:**
int SDLNet_ResizePacket(UDPpacket *packet, int size)
Resize the data buffer in a UDP packet.
- **SDLNet_FreePacket:**
void SDLNet_FreePacket(UDPpacket *packet)
Free a previously allocated UDP packet.
- **SDLNet_AllocPacketV:**
UDPpacket **SDLNet_AllocPacketV(int howmany, int size)
Allocate a vector of UDP packets.
- **SDLNet_FreePacketV:**
void SDLNet_FreePacketV(UDPpacket **packetV)
Free a vector of UDP packets.



SDL_Net - Funções

- UDP sockets
- **SDLNet_UDP_GetPeerAddress:**
IPAddress *SDLNet_UDP_GetPeerAddress(UDPsocket sock, int channel)
Get an IP address for a socket channel or get the port with which you opened the socket.
- **SDLNet_UDP_Send:**
int SDLNet_UDP_Send(UDPsocket sock, int channel, UDPpacket *packet)
Send a UDP packet.
- **SDLNet_UDP_Recv:**
int SDLNet_UDP_Recv(UDPsocket sock, UDPpacket *packet)
Receive into a UDP packet. This function is non-blocking.
- **SDLNet_UDP_SendV:**
int SDLNet_UDP_SendV(UDPsocket sock, UDPpacket **packetV, int npackets)
Send a UDP packet vector.
- **SDLNet_UDP_RecvV:**
int SDLNet_UDP_RecvV(UDPsocket sock, UDPpacket **packetV)
Receive into a UDP packet vector.



SDL_Net - Funções

- Socket sets
Funções utilizadas para trabalhar com múltiplos sockets. Permite identificar quando um socket possui dados ou conexões esperando para processar.
 - **SDLNet_AllocSocketSet:**
SDLNet_SocketSet SDLNet_AllocSocketSet(int maxsockets)
Create a new socket set.
 - **SDLNet_FreeSocketSet:**
void SDLNet_FreeSocketSet(SDLNet_SocketSet set)
Free a socket set.
 - **SDLNet_AddSocket:**
int SDLNet_AddSocket(SDLNet_SocketSet set, SDLNet_GenericSocket sock)
Add a socket to a socket set.
 - **SDLNet_DelSocket:**
int SDLNet_DelSocket(SDLNet_SocketSet set, SDLNet_GenericSocket sock)
Remove a socket from a socket set.
 - **SDLNet_CheckSockets:**
int SDLNet_CheckSockets(SDLNet_SocketSet set, Uint32 timeout)
Check and wait for sockets in a set to have activity.
 - **SDLNet_SocketReady:**
int SDLNet_SocketReady(sock)
See if a socket has activity.



SDL_Net - Dicas

- Quando se utiliza SDL_net, o UDP por padrão é configurado para enviar pacotes em *broadcast*. Não existe nenhum modo direto exceto efetuar o bloqueio.



SDL_Net - Exemplos

- Os exemplos consistem de programas que ilustram o uso da SDL_net.



SDL_Net

- Iniciar
 - Baixar pacotes da SDL:
<http://www.libsdl.org/download-1.2.php>

The screenshot shows the SDL website's download page. A red arrow points to the link for 'SDL-devel-1.2.13-VC8.zip (Visual C++ 2005 Service Pack 1)'. The page lists various download links for different operating systems and development environments.

SDL-devel-1.2.13-VC8.zip (Visual C++ 2005 Service Pack 1)

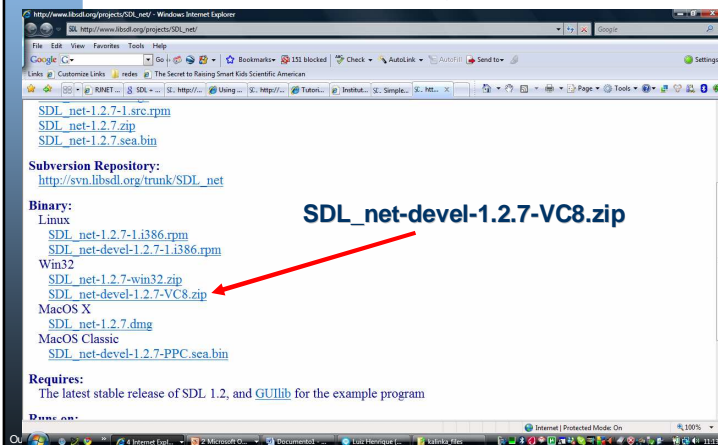
Development Libraries:

- Linux:
 - SDL-devel-1.2.13-1.036.rpm
 - SDL-devel-1.2.13-1.036.i386.rpm
 - <http://backslash.libsdl.org/0368-BSDLnet/>
- Win32:
 - SDL-devel-1.2.13-VC8.zip (Visual C++ 6.0)
 - SDL-devel-1.2.13-VC8.zip (Visual C++ 2005 Service Pack 1)
 - SDL-devel-1.2.13-vc8022.zip (MinGW32)
- MacOS (Classic):
 - SDL-devel-1.2.13-PPC.tar.gz (MPW + CodeWarrior)
- MacOS X:
 - SDL-devel-1.2.13-entire.dmg (templates and documentation)



SDL_Net

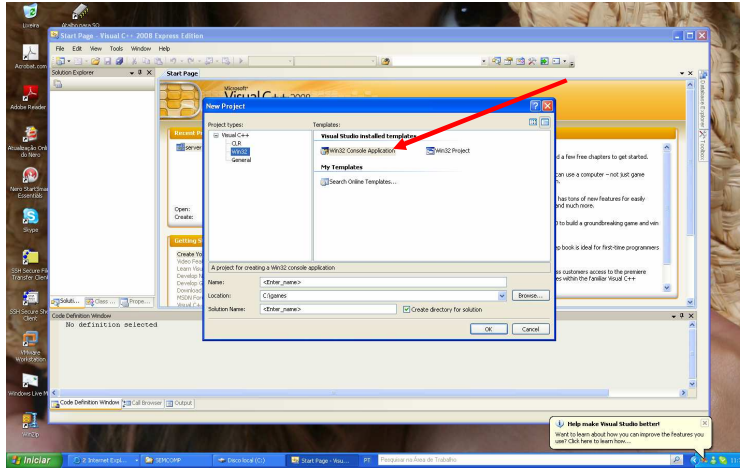
- Baixar pacote da SDL_Net
– http://www.libsdl.org/projects/SDL_net/



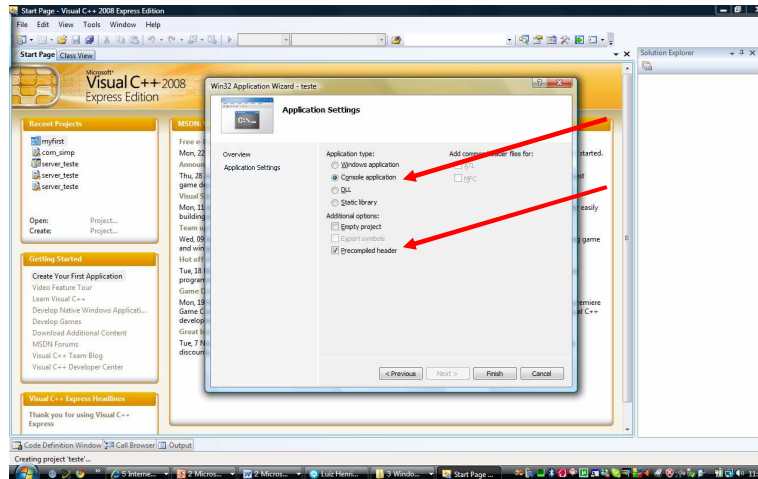
SDL_Net

- Descompacte os arquivos no diretório
– C:\
- Abra o Visual
- Escolha Criar novo projeto
- Selecione a opção WIN32 / Console Application / Precompiled Header

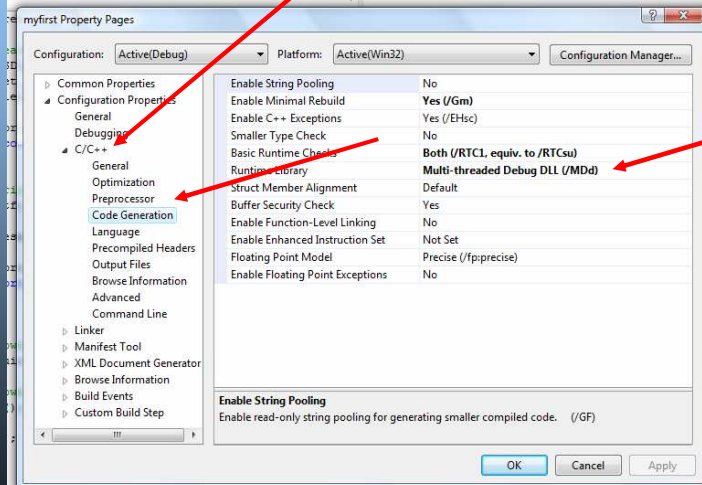
SDL_Net



SDL_Net



SDL_Net



29

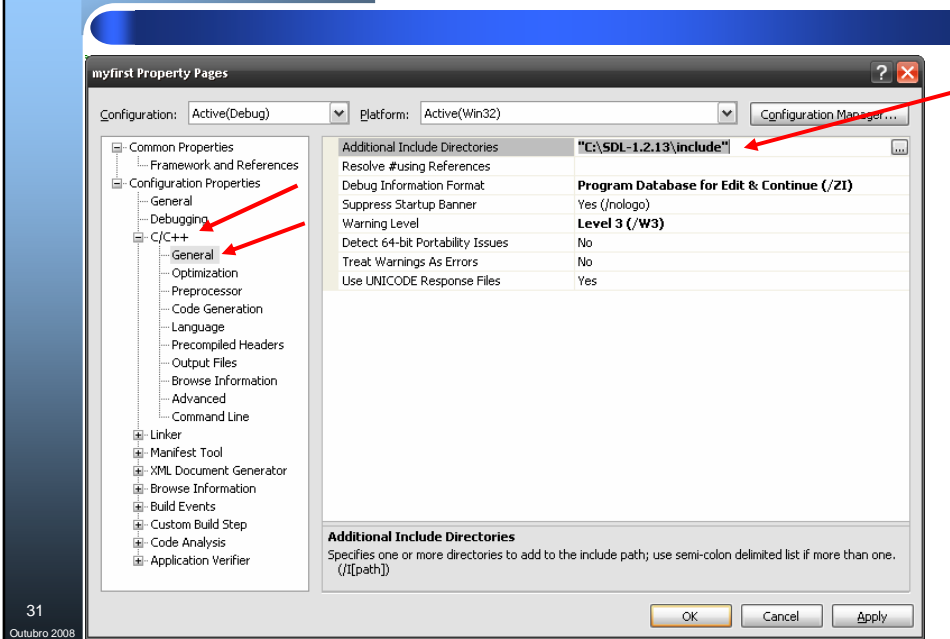
Outubro 2008

SDL_Net

- O próximo passo consiste em incluir a biblioteca SDL
 - Entre em propriedades
 - C/C++
 - General
 - Additionl Include Directories
 - Então inclua o diretório **include** que se encontra dentro do diretório SDL-1.2.13 que vc descompactou.

30

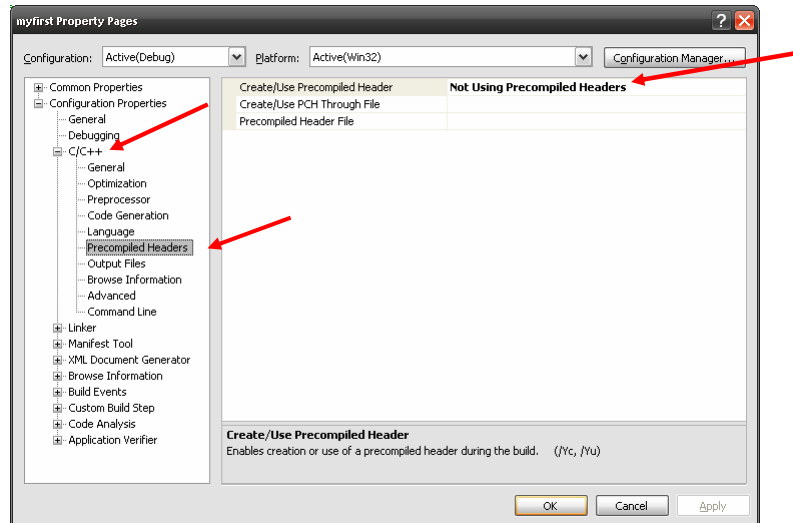
Outubro 2008



SDL_Net

- Ainda nesse menu escolha a opção **Precompiled Headers**
 - C/C++
 - Precompiled Headers
 - Create/Use Precompiled Header
 - Not Using Precompiled Headers

SDL_Net



33

Outubro 2008

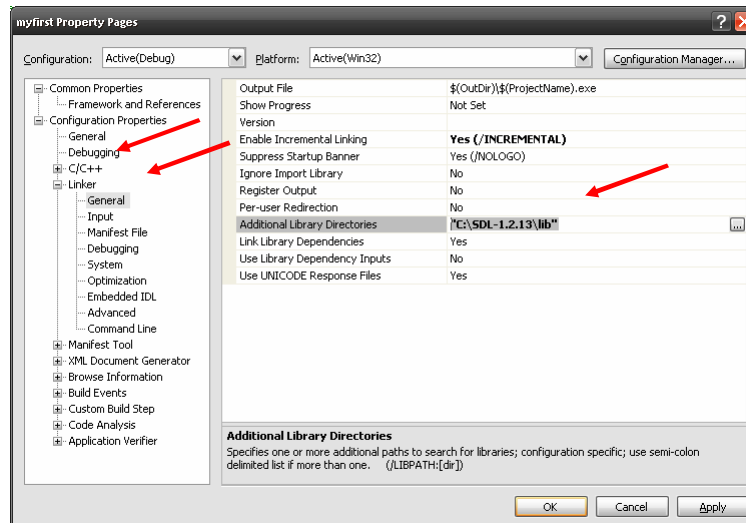
SDL_Net

- Agora:
 - Acesse a opção *linker*
 - General
 - Additional Libraries Directories
 - E inclua o diretório *lib* que você descompactou.

34

Outubro 2008

SDL_Net



35

Outubro 2008

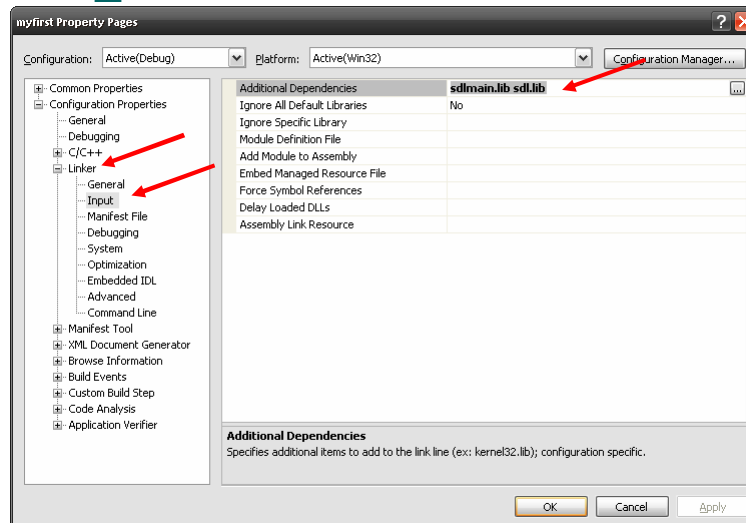
SDL_Net

- E por fim:
 - Ainda na opção *linker*
 - **Input**
 - **Additional Dependencies**
 - E inclua *sdlmain.lib* e *sdl.lib*

36

Outubro 2008

SDL_Net



37

Outubro 2008

SDL_Net

- Não esquecer de copiar o arquivo SDL.dll para o diretório de sua aplicação

38

Outubro 2008



SDL_Net

```
// myfirst.cpp : Defines the entry point for the console application.
//
#include "SDL.h"
#include "stdafx.h"
int _tmain(int argc, _TCHAR* argv[])
{
    SDL_Surface *screen;
    if (SDL_Init(SDL_INIT_VIDEO) < 0)
    {
        printf("error ao iniciar SDL: %s\n", SDL_GetError());
        exit(1);
    }
    //Antes de sair da aplicação, chama SDL_Quit
    atexit(SDL_Quit);
    screen = SDL_SetVideoMode(400, 400, 16, 0);
    if (screen == NULL)
    {
        printf("error ao selecionar o modo de video: %s\n", SDL_GetError());
        exit(1);
    }
    SDL_WM_SetCaption("Oi mundo!", "Oi Mundo!");
    SDL_Event ev;
    while(true) {
        while(SDL_PollEvent(&ev)) {
            if(ev.type == SDL_QUIT){
                printf(">O usuário quer sair.\n");
                printf(">SDL Terminado.\n");
                return 0;
            }
        }
    }
    return 0;
}
```

39

Outubro 2008



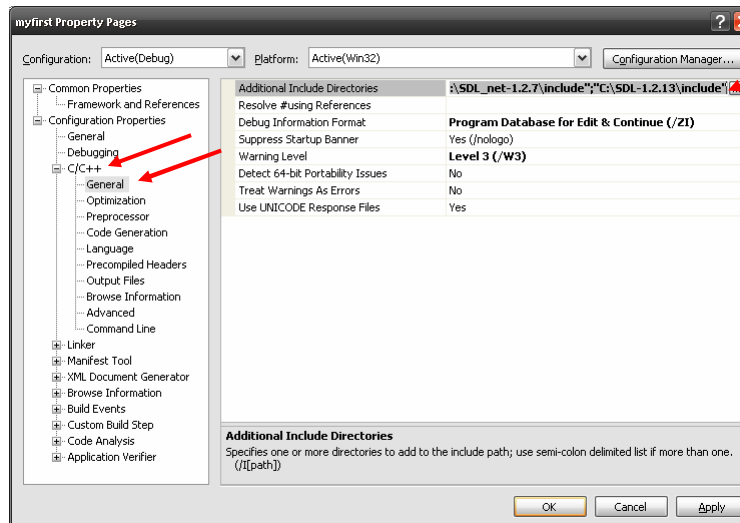
SDL_Net

- Ajustando a SDL_Net
 - Agora repetem-se alguns passos para acerto da SDL_Net
 - Entrar em propriedade novamente:
 - Menu C/C++
 - General
 - Additional Include Directories
 - (então inclui-se o diretório *include* da SDL_Net)

40

Outubro 2008

SDL_Net



41

Outubro 2008

SDL_Net

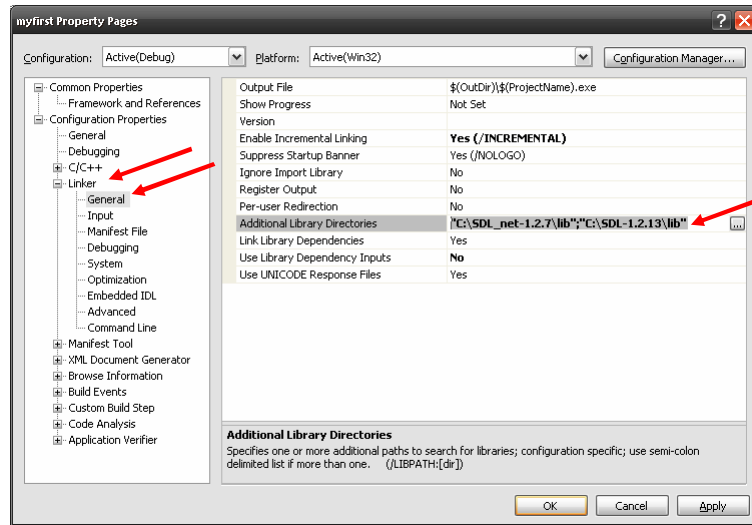
- No menu *Linker*
 - General
 - Additional Library Directories
 - Incluir diretório *lib* da SDL_Net

42

Outubro 2008



SDL_Net



43

Outubro 2008



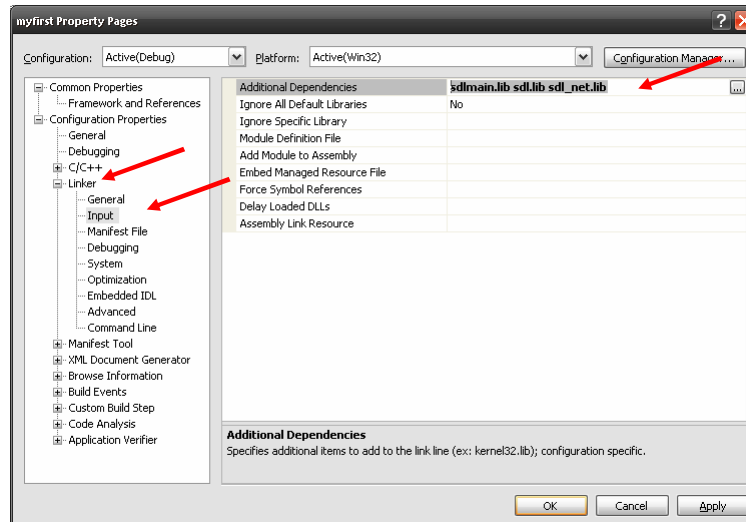
SDL_Net

- E por fim:
 - Ainda na opção **linker**
 - **Input**
 - **Additional Dependencies**
 - E inclua **sdl_net.lib**

44

Outubro 2008

SDL_Net



SDL_Net

- Mais uma vez, não esquecer de copiar o arquivo SDL_net.dll para o diretório de sua aplicação



Exemplos

- Um cliente e um servidor



Exemplos

- Um servidor e múltiplos clientes



INFORMAÇÕES SOBRE O MINI-CURSO

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO – Profa. Kalinka Castelo Branco

Web institucional: <http://www.icmc.usp.br/ssc/>

Página pessoal: <http://www.icmc.usp.br/~fosorio/>

**E-mail: [fosorio \[at\] icmc. usp. br](mailto:fosorio@icmc.usp.br) ou [fosorio \[at\] gmail. Com](mailto:fosorio@gmail.com)
[kalinka \[at\] icmc. usp. br](mailto:kalinka@icmc.usp.br)**

SEMCOMP 11 - Outubro 2008

Mini-Curso

Desenvolvimento Rápido de Jogos em 3D:

Usando o GDK com o Microsoft Visual C/C++ (Game RAD)