



Snake Multiplayer - Relatório

Trabalho Prático – 2ª Meta

Diogo Miguel Branco Santos – 21210835

Mário Jorge Lopes da Costa – 21210385

Sistemas Operativos 2

Licenciatura em Engenharia Informática

Ano Letivo 2016/2017

Índice

Introdução.....	3
Servidor	4
Cliente (GUI).....	6
DLL.....	9
Cliente Administrador	9
Bugs e erros do Trabalho Prático	10
Diagrama de classes	10

Introdução

Este relatório, realizado no âmbito da 2ª meta do trabalho prático da unidade curricular “Sistemas Operativos 2”, pretende apresentar de forma resumida os principais mecanismos de comunicação, de sincronização e de estruturação da aplicação. Pretende também dar uma descrição detalhada dos programas que constituem o trabalho.

Servidor

O servidor consiste numa aplicação do tipo serviço NT que permite criar um jogo com vários clientes, comunicando com os mesmos através de named pipes remotos e/ou memória partilhada. A criação dos named pipes e memória partilhada é feita logo no arranque do servidor, ficando à espera dos clientes. Este cria uma thread com um named pipe por cada cliente. No caso da memória partilhada, este cria duas threads, uma para escrever na memória partilhada (através das funções da DLL) e outra para ler da memória partilhada (através das funções da DLL). Também permite que um cliente do tipo Administrador se ligue e execute pedidos para o mesmo.

O primeiro cliente a ligar-se cria a sala de jogo, indicando o tamanho do mapa (comprimento e altura), o numero de jogadores, o numero de jogadores controlados pelo servidor, o tamanho inicial das cobras, o numero de objetos no jogo e o seu nome no jogo, e os restantes associam-se a esse mesmo jogo, indicando apenas o seu nome no jogo.

De seguida o cliente que criou a sala pode dar inicio ao jogo e todos os jogadores apenas dão indicações as suas cobras que direção estas devem seguir.

Se o cliente ligado for do tipo Administrador, o servidor permite que este execute pedidos para o jogo que se encontra a decorrer.

Este recebe/envia pedidos/respostas através de uma estrutura denominada de “Message”, constituída pelas seguintes variáveis da imagem seguinte.

```
typedef struct
{
    int actualX;
    int actualY;
    TCHAR map[MAX_TAM_MAP][MAX_TAM_MAP];
}Map;

typedef struct
{
    int pid;
    int numOfPlayers;
    int code;
    char msg[BUFFER_SIZE];
    int scores[MAX_PLAYERS];
    Map map;
}Message;
```

Cada tipo de pedido/resposta é distinguido através da variável *code* e a informação necessária para esse pedido é colocada na variável *msg*, ou pelas outras variáveis existentes na estrutura. Os pedidos/respostas implementados no servidor são:

<i>code</i>	<i>Dados do(a) pedido/resposta</i>	<i>Descrição</i>
CREATEGAME	<Comprimento [1,80]> <Altura [1,80]> <Nr jogadores [1,10]> <tamanho da cobra[1,10]> <Nr de objetos [1,10]> <Nr de cobras Automáticas [1,10]> <Username>	Serve para criar o jogo, indicando os vários parâmetros descritos em cima.
JOIN	<Username>	Serve para associar um jogador a um jogo já criado, indicando o seu nome.
START	Vazia	Serve para iniciar o jogo. Só é aceite pelo jogador que criou a sala.
SETDIRECTION	<Direção>	Serve para indicar a direção que a sua cobra irá seguir, indicando a respetiva direção.
DISCONNECT	<pid>	Serve para remover o jogador da sala, indicando o id (o id que lhe define no servidor).
MAP	Variável <i>map</i> com a informação do mapa	Serve para enviar o estado do mapa para todos os clientes, para que estes atualizem o mapa. Só enviado pelo servidor.
SEED_OBJECT	<Tipo de Objeto> <Numero de Objetos>	Serve para indicar ao servidor, qual e quantos objetos mete no jogo. Só enviado pelo cliente do tipo administrador.
END	Vazio	Serve para enviar a todos os clientes que o jogo terminou
SUCCESS	Vazio	Serve para indicar que o pedido enviado foi executado com sucesso.
FAIL	Vazio	Serve para indicar que o pedido enviado não foi executado com sucesso.

Cliente (GUI)

É uma aplicação win32 com interface gráfica que serve para o cliente poder comunicar com o servidor e jogar o respetivo jogo. Nela existe um menu com as abas “Novo Jogo” (com um submenu contendo as opções “Ligar Remotamente”, “Ligar Localmente” e “Sair”), “Editar” (com um submenu contendo a opção “Editar Skins”) e “Sobre”, como mostram as figuras seguintes.

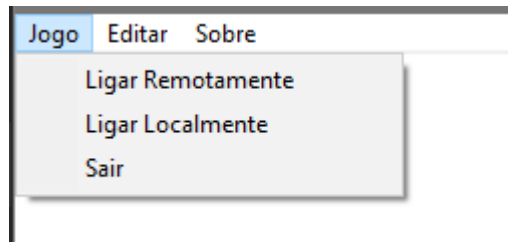


Figura 1

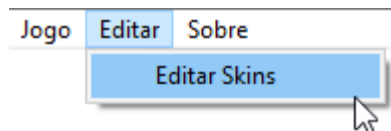


Figura 2

Ao escolher “Ligar Remotamente”, é mostrado uma Dialog dando a opção de criar jogo ou associar-se a um jogo.



Figura 3

Escolhendo “Criar jogo”, é mostrado ao cliente outra dialog com os dados de criação do jogo no lado do servidor.

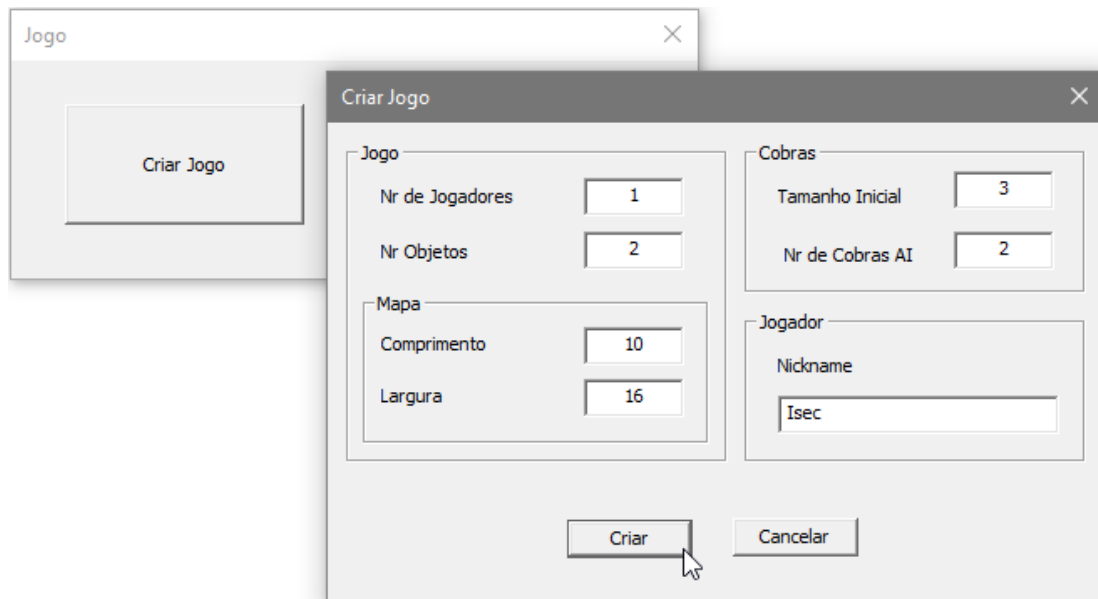


Figura 4

Escolhendo “Juntar ao Jogo”, é mostrado ao cliente outra dialog com os dados de associação a um jogo no lado do servidor.

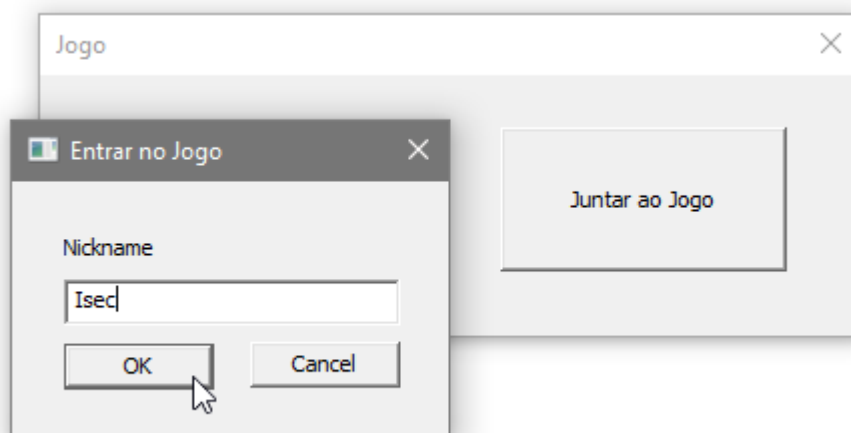


Figura 5

Após clicar no OK ou Criar das dialogs anteriores, é mostrado uma outra para dar o começo do Jogo.

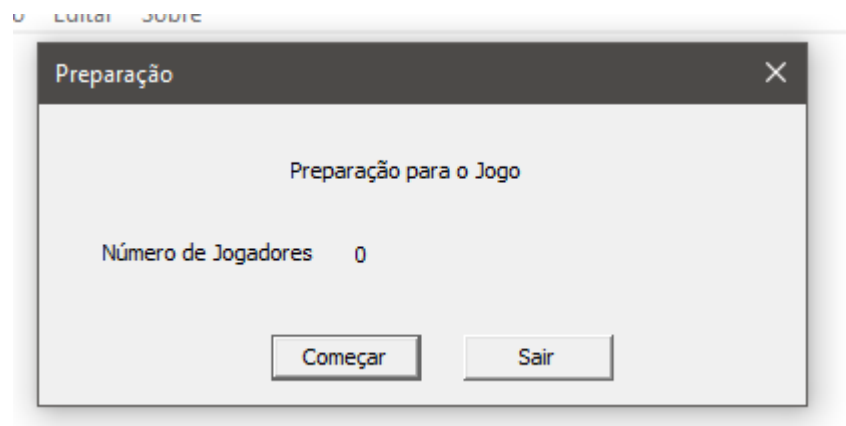


Figura 6

Escolhendo “Editar Skins” na aba “Editar”, é mostrado um dialog com a lista das imagens que o jogador pode editar no jogo, através do Paint.

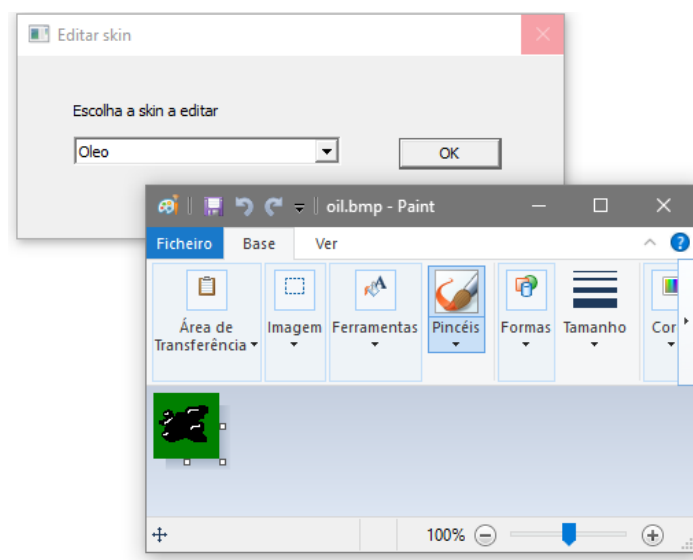


Figura 7

Escolhendo a opção “Sobre”, é mostrado uma dialog com um breve resumo dos criadores da aplicação.

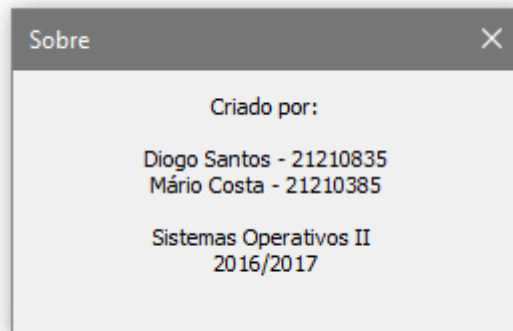


Figura 8

Nesta aplicação também a criação do mesmo numero de threads que existem no servidor para a respetiva comunicação, estando só ativas threads que fazem parte do seu tipo de ligação, remoto (através dos named pipes) ou local (através da memória partilhada).

DLL

A DLL possui apenas o mecanismo de comunicação preciso para a interação entre o cliente e o servidor, através da memória partilhada. Nela existem duas funções, uma para ler da memória partilhada e outra para escrever na memória partilhada.

Cliente Administrador

O cliente do tipo administrador, é um cliente que funciona como administrador do jogo que está a decorrer. Este liga-se ao servidor e pode colocar objetos dentro do jogo que está a decorrer.

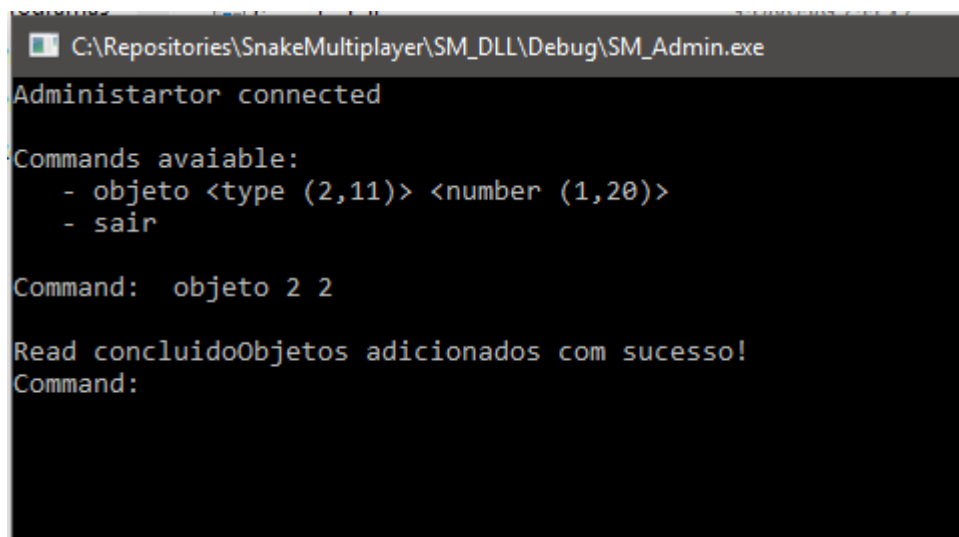


Figura 9

Bugs e erros do Trabalho Prático

O trabalho não está completo, assim existem algumas falhas na comunicação e na logica do jogo.

Diagrama de classes

Diagrama do Servidor:

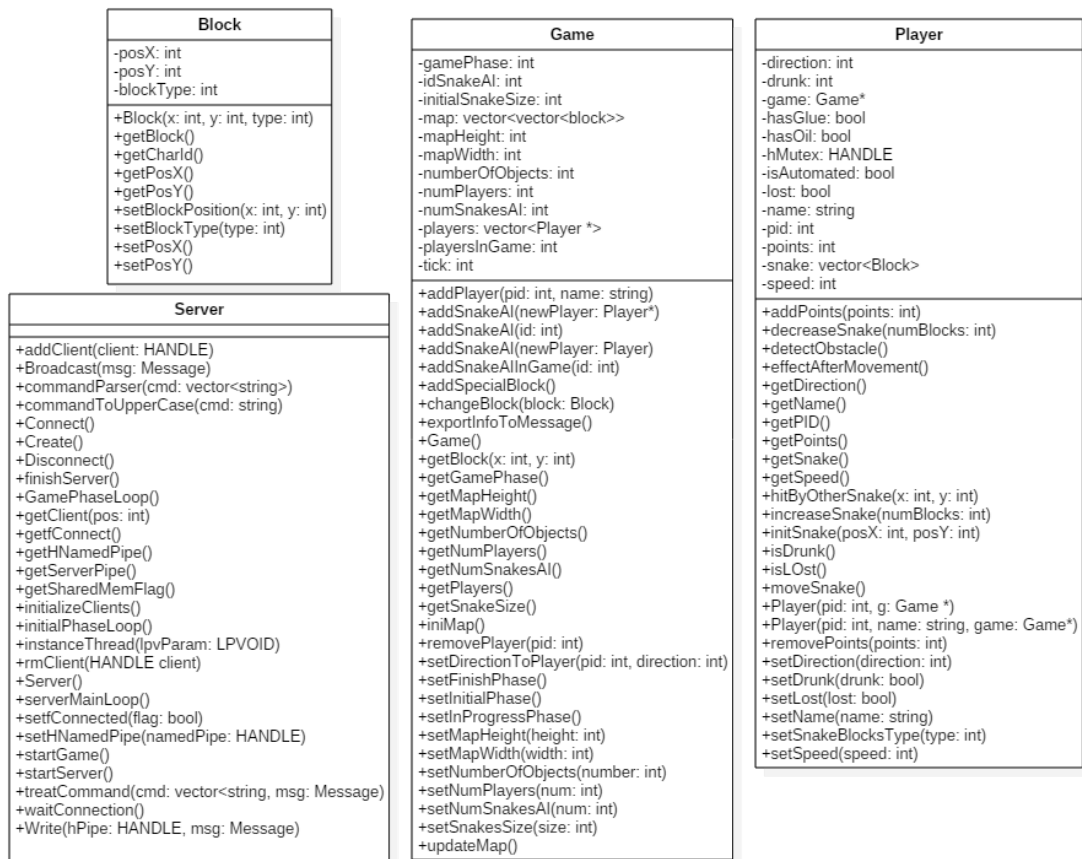


Diagrama da DLL:

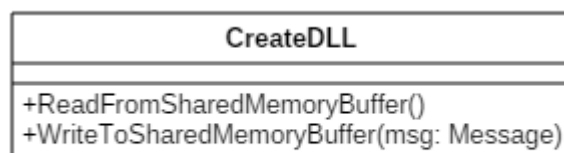


Diagrama do Cliente:

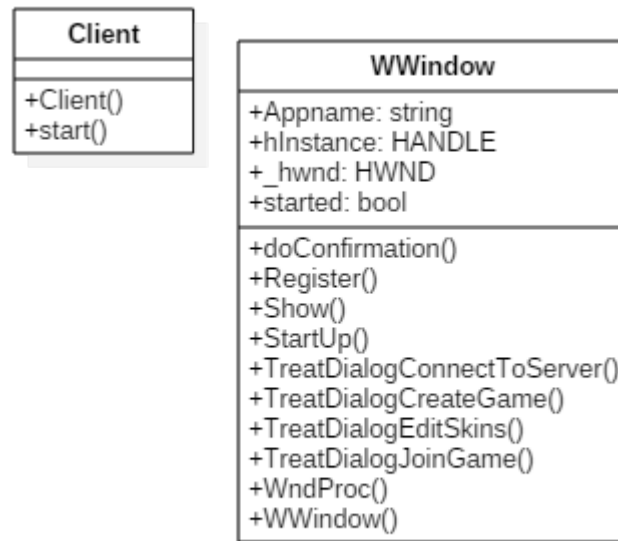


Diagrama do Cliente administrador:

