



# 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) .....	5
DLL .....	7
Bugs e erros do Trabalho Prático .....	8
Diagrama de classes .....	8

## **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 consola 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).

O primeiro 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.

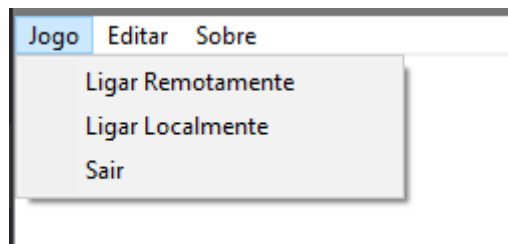
De seguida o cliente que criou a sala pode dar inicio ao jogo e todos os jogadores apenas dão indicações as suas cobras que direção estas devem seguir.

Este possui os seguintes comandos para o desenrolar do jogo:

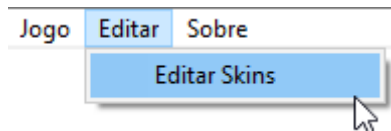
- **CREATEGAME** <Map Width [1,80]> <Map Height [1,80]> <Nr Players [1,10]> <Snake Size [1,10]> <Nr Objects [1,10]> <Nr Snakes AI [1,10]> <Player Username> - Este comando serve para criar o jogo, indicando os vários parâmetros descritos em cima.
- **JOIN** <Player Username> - Este comando serve para associar um jogador a um jogo já criado, indicando o seu nome.
- **START** – Este comando serve para iniciar o jogo. Só é aceite pelo jogador que criou a sala.
- **SETDIRECTION** <direction> - Este comando serve para indicar a direção que a sua cobra irá seguir, indicando a respetiva direção.
- **DISCONNECT** <PID> - Este comando serve para remover o jogador da sala, indicando o pid (o id que lhe define no servidor).

## 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.

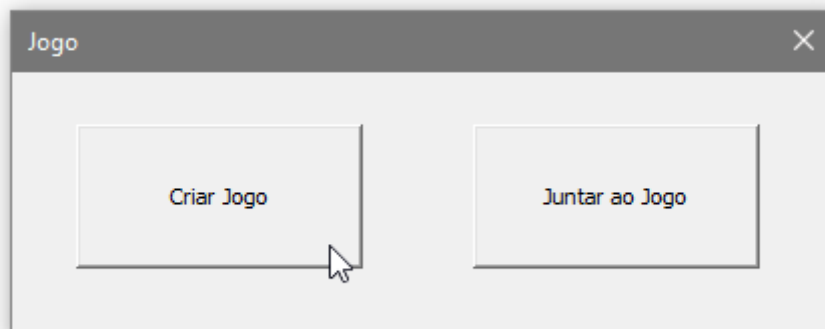


*Figura 1*



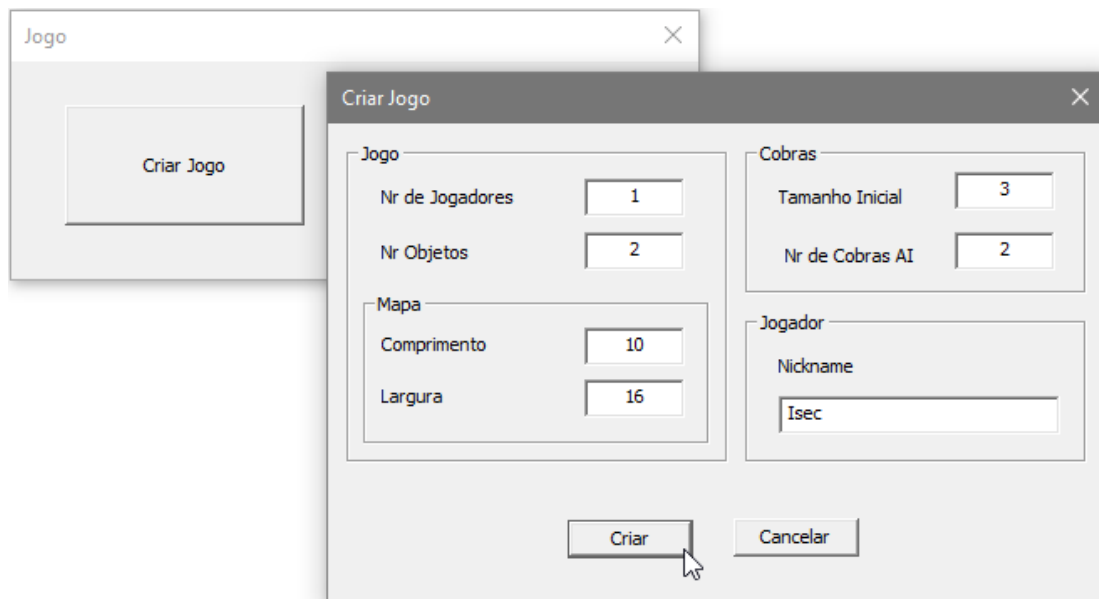
*Figura 2*

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



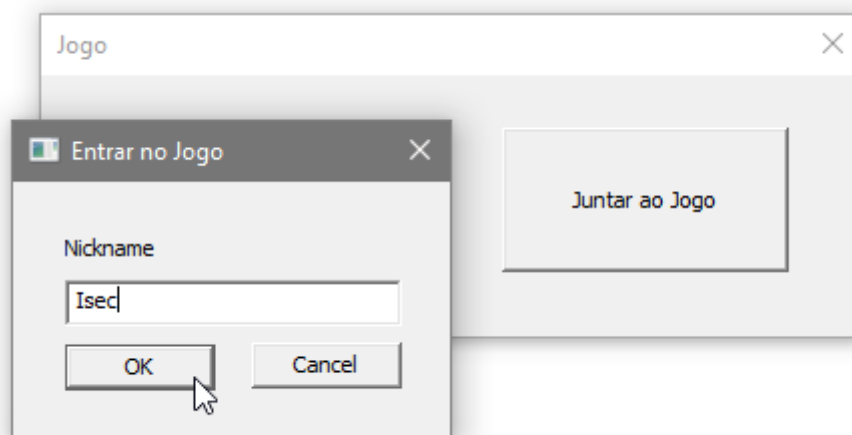
*Figura 3*

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



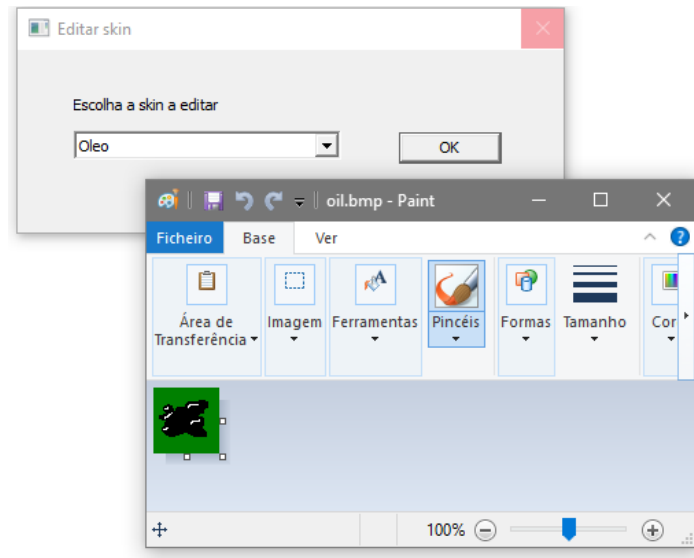
**Figura 4**

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



**Figura 5**

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 6*

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



*Figura 7*

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.

## Bugs e erros do Trabalho Prático

O trabalho não está completo, assim existem algumas falhas na comunicação, bem como a falta da visualização do jogo a decorrer.

## Diagrama de classes

Diagrama do Servidor:

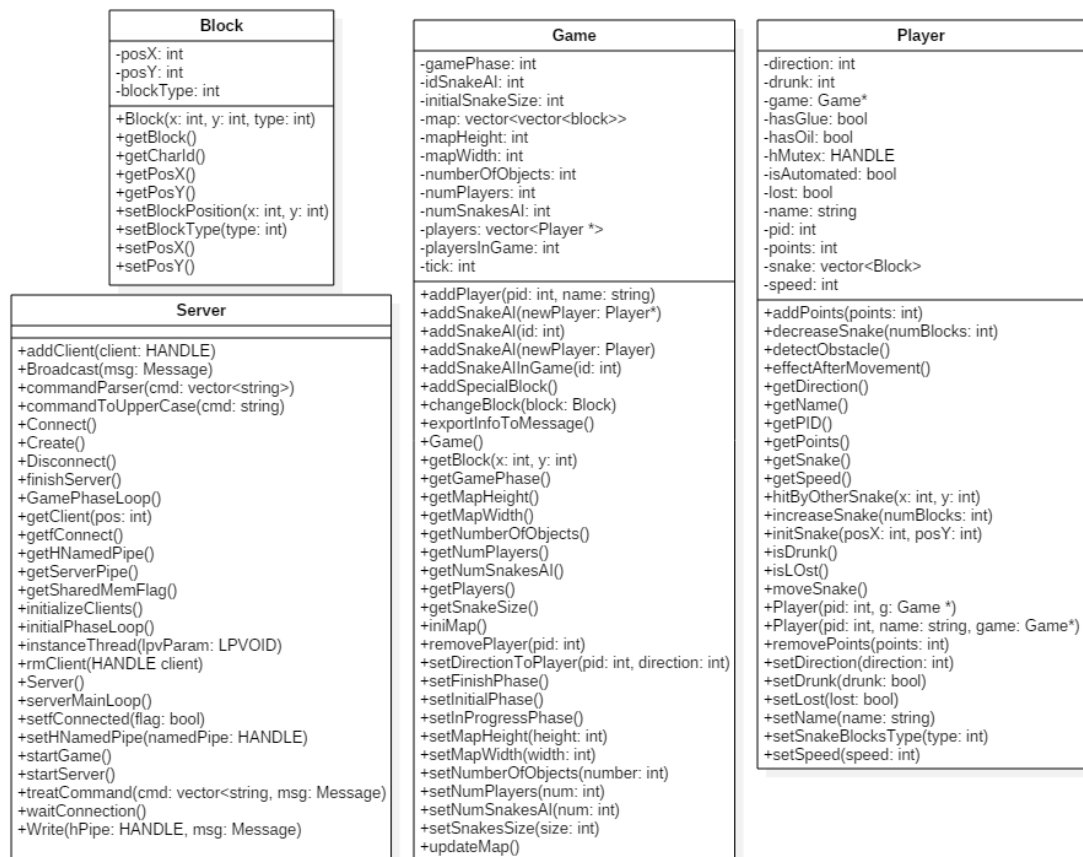


Diagrama da DLL:

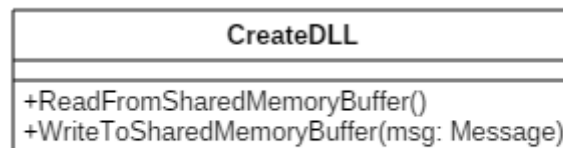




Diagrama do Cliente:

