

Guessing Game - Python Edition

Universidade de Aveiro

Bernardo Marçal, Telmo Sauce



Trabalho de Aprofundamento 2

Guessing Game - Python Edition

Departamento de Eletrónica, Telecomunicações e
Informática

Universidade de Aveiro

Bernardo Marçal, Telmo Sauce
(103236) bernardo.marcal@ua.pt, (104428) telmobelasauce@ua.pt

30 de maio de 2021

Resumo

A realização deste trabalho teve como objetivo principal, criar um servidor que consiga suportar a geração de um número totalmente aleatório, neste caso entre 0 e 100, e além disso, gerar um número de tentativas, também aleatório, para que o utilizador consiga adivinhar o tal número "secreto", neste caso entre 10 e 30.

Para efetuar este trabalho com sucesso é necessário criar um client, onde o cliente vai tentar adivinhar o número, e um servidor, onde se conecta diretamente com o client. Este servidor nunca deverá aceitar dois "clientes" com o mesmo ID que estejam a jogar simultaneamente. Ao acabar o jogo, ou seja, após ter esgotado as tentativas ou ter acertado no número, o servidor vai gerar um ficheiro report.csv, onde estará escrito todas as estatísticas do utilizador ao concluir o jogo (o seu número secreto, o número máximo de jogadas, o número de jogadas realizadas e o resultado do cliente, seja sucesso, fracasso ou desistência).

Caso tudo corra bem e o cliente consiga acertar com sucesso o número aleatório, aparecerá uma mensagem a congratular o cliente da sua conquista, com o número de tentativas efetuadas, e o mesmo acontecerá se este esgotar o número de tentativas dadas, aparecerá uma mensagem a informar que este não conseguiu descobrir o tal número.

Índice

1	Introdução	1
2	Metodologia	2
2.1	Conteúdo do Server e Cliente	2
2.1.1	Funcionamento do Servidor	2
2.1.2	Funcionamento do Cliente	3
2.2	Execução do programa	3
3	Conclusões	5
3.0.1	CODE UA	6
3.0.2	Contribuição dos Autores	6

Capítulo 1

Introdução

Este relatório tem como objetivo analisar as funcionalidades e as funções existentes, tanto do `server.py` como no `client.py`, e analisá-las passo a passo para uma melhor compreensão de cada função e o que representa para o funcionamento do código.

Além deste resumo de cada função destes dois ficheiros, também teremos uma breve mas bastante explícita, explicação de como todo o programa funciona, abordando todos os detalhes e informações necessárias para a total compreensão do programa, explicar como se consegue vencer o jogo, e o que acontece após este acontecimento.

Concluimos este trabalho explicando alguns dos poucos problemas que nos foram aparecendo ao longo do desenvolvimento do trabalho.

Capítulo 2

Metodologia

Neste capítulo vai ser explicado em detalhe a criação dos ficheiros `server.py` e `client.py`.

2.1 Conteúdo do Server e Cliente

2.1.1 Funcionamento do Servidor

- **"find_client_id"** - Esta função lê a lista gamers até encontrar a posição do cliente;
- **"new_msg"** - Esta função é a base de todo o programa, visto que esta executa outras funções para mais tarde comunicar com o Cliente;
- **"new_client"** - Esta função cria uma nova lista para o novo jogador, esta lista contém a Cipher Key, o socket, o número a adivinhar e o número de tentativas;
- **"clean_client"** - Esta função tem a simples função de eliminar o cliente da lista criada em "new_client";
- **"quit_client"** - Esta função atualiza o ficheiro csv, usando a função "update_file", com a mensagem "QUIT" e elimina o jogador da lista criada em "new_client", usando a função "clean_client";
- **"guess_client"** - Esta função retorna uma mensagem ao cliente a dizer se o numero secreto é maior, menor ou igual a tentativa do jogador;
- **"stop_client"** - Esta função é executada quando o jogador acaba as suas tentativas ou ganha o jogo. Se o jogador não acertou o número, tendo ficado sem tentativas a função atualiza o ficheiro csv, usando a função "update_file", com a mensagem "FAILURE". Se o jogador conseguir adivinhar o número, esta atualiza o mesmo ficheiro mas com a mensagem

"SUCCESS". No final o cliente é eliminado da lista chamando a função `clean_client`;

- **"create_file" e "update_file"** - Estas são funções que modificam e criam o ficheiro csv. `Create_file` é chamada ao executar `"server.py"`, esta cria um novo ficheiro chamado `"report.csv"` com o texto. `"Gamer,Secret Number,Attempts,Attempts Used,Result"`. Já a função `"update_file"` acrescenta texto ao ficheiro, como no exemplo seguinte, `"João,74,11,6,SUCCESS"` `client id`, numero a adivinhar, numero máximo de tentativas e resultado, respetivamente;
- **"encrypt_intvalue" e "decrypt_intvalue"** - Estas são funções usadas para encryptar e decryptar os valores;
- **"rec_key"** - Esta função tem como objetivo inserir a Cipher Key na lista do respetivo jogador;

2.1.2 Funcionamento do Cliente

- **"quit_action"** - Esta função manda mensagem ao Server, onde esta escuta a função `"quit_client"`
- **"run_client"** - Esta é a Interface do nosso jogo e ela que comunica com o Servidor para executar as funções existentes no Server
- **"str_Key"** - Esta função cria uma Cipher Key para o cliente.
- **"encrypt_intvalue" e "decrypt_intvalue"** - Estas são funções usadas para encryptar e decryptar os valores.

2.2 Execução do programa

Após o jogador executar o programa `"server.py"` no terminal o server cria ficheiro `"report.csv"` chamando a função **"create_file"**. Em seguida aparece no terminal as informações do jogo e uma mensagem para o jogador decidir se quer começar o jogo ou sair, para o jogo começar este tem de escrever `s/S`, para sair este tem que escrever `q/Q`. Após este começar o jogo, é enviada uma mensagem ao servidor a função `new_msg` recebe a mensagem e executa a função **"new_client"** onde é criada a lista do nosso jogador. Em seguida a interface do jogo aparece no terminal entrando num `while` até o número de tentativas máximas acabar. Dentro do `while`, é pedido ao jogador para tentar adivinhar o número secreto, cada vez que o jogador introduzir um número é enviada uma mensagem ao server, que é recebida por `new_msg`, que chama a função **guess_client**, esta após comparar o número a adivinhar e a adivinha do jogador, é retornado uma mensagem ao cliente caso esta seja mais pequeno aparece no terminal a mensagem `"Try a smaller number"` caso seja maior aparece

a mensagem "Try a larger number", estas funções e a mensagem são chamadas uma a seguir a outra, até que o numero seja igual ou as tentativas acabarem ou até se o jogador pretender desistir do jogo, se este pressionar q/Q o jogo acaba, neste caso o **new_msg** executa a função **quit_client** limpando o jogador da lista gamers e atualizando o report com a mensagem *"QUIT"*. Caso o jogador adivinhe o numero secreto a mensagem que a função **guess_client** retorna é "equals" e de seguida aparece no terminal a mensagem *"Good job! You guessed the number in only (numero de tentativas)"* e é enviada uma mensagem ao servidor, recebida por **new_msg** e que redireciona para **stop_client** como o jogador ganhou o jogo a função **update_file** executa escrevendo a mensagem de *"SUCCESS"* no report e em seguida o jogo acaba com o comando *"exit(0)"*. Por fim, caso o jogador não consiga adivinhar o número nas tentativas que lhe são oferecidas, o jogo acaba, enviando uma mensagem para o server, esta mensagem é recebida por **new_msg** e esta executa **stop_client** que chama a função **update_client** e atualiza o ficheiro csv com a mensagem *"FAILURE"* acabando com o jogo ao executar *"exit(0)"*.

Capítulo 3

Conclusões

Caso o trabalho tenha corrido perfeitamente, o jogador começaria o jogo e teria a escolha de poder encryptar a comunicação dos números entre o Cliente e o Servidor, desta forma este teria maior privacidade. No entanto tivemos problemas técnicos na descriptação do numero no servidor, por esta razão tivemos de retirar esta funcionalidade do jogo, comentando grande parte do código relacionado com a encriptação. Para além destes imprevistos, pensamos ter seguido bem as instruções no guião, realizado desta forma o jogo da maneira que era previsto.

Dados dos Autores

3.0.1 CODE UA

Aqui encontra-se o link para o repositório onde está toda a informação sobre este trabalho :

- <https://code.ua.pt/projects/labi2021-ap2-g14/repository>

3.0.2 Contribuição dos Autores

Bernardo Marçal (BM) - 50%

Telmo Sauce (TS) - 50%

Acrónimos

BM Bernardo Marçal

TS Telmo Sauce