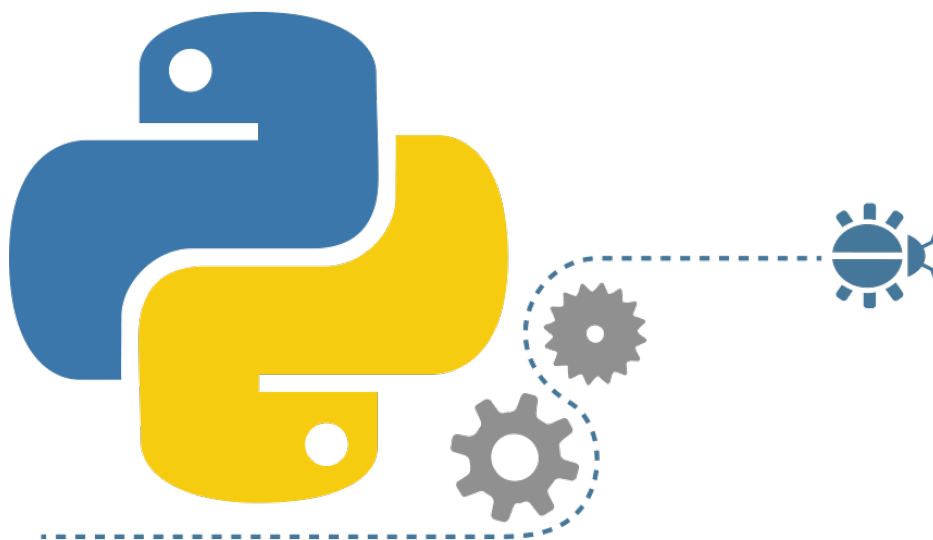




Programação I
Universidade de Évora
Curso de Engenharia Informática



Projeto: Pigs and Bulls

Trabalho realizado por:
Yaroslav Kolodiy nº39859

Introdução

Este trabalho foi realizado a pedido da professora Teresa Gonçalves de programação 1. Tem como objetivo, a criação por parte dos alunos uma réplica do jogo “Pigs and Bulls”. A linguagem de programação usada é a mesma que foi lecionada durante o semestre, sendo esta Python.

Este relatório consiste na explicação detalhada do programa, apresentar as funções e variáveis utilizadas tais como os “imports”.

Breve resumo:

Pigs and Bulls é um jogo para 2 jogadores cujo objetivo é adivinhar o código pensado por um dos jogadores. Na versão original, o código é composto por 4 algarismos escolhidos entre 0 e 9, sem repetição.

Um dos jogadores pensa num código; o outro jogador propõe um código e o primeiro indica o número de acertos. Existem duas indicações distintas para acertos: se os algarismos estão nas posições certas, são "touros"; se estiverem em posições diferentes, são "porcos". Por exemplo, se o código for 4271 e a tentativa for 1234, a resposta deve ser 1 touro; 2 porcos (o touro é "2", os porcos são "4" e "1").¹

Imports:

Nome:	Função:
random	Gerar um número aleatório

Funções utilizadas no programa:

Função:	Argumentos e retorno:
gerar_numero()	Não contem argumentos Valor de retorno: num-string de 4 digitos

<code>acertos(num_g,num,cont,tentativas)</code>	<p>Argumentos:</p> <p>num_g-número gerado pelo computador, string</p> <p>num.número introduzido pelo utilizador, string</p> <p>cont-contador de tentativa, inteiro</p> <p>tentativas-lista de todas as tentativas já realizadas, lista</p> <p>Valor de retorno:</p> <p>tupulo com as seguintes variaveis:</p> <p>acert-porcoss e touros acertados, string</p> <p>c- lista de tentativas após a soma da realizada pela função, lista</p>

Descrição do programa:

O programa começa por igual num_g a função “gerar_numero()”. A função gerar_numero acaba por inicializar uma variável “k” como uma lista vazia, a qual irá armazenar os números gerados no ciclo while que se segue. O ciclo while apresentado de seguida, irá correr até que a lista “k” não contenha 4 elementos totalmente diferentes. Esses elementos são armazenados na variável “c” e gerados através da função random.random() o qual gera um número entre [0,1[o qual tem de ser multiplicado por 10 para dar um número unitário, para retirar a parte décima, é utilizada a conversão do número para um número inteiro int(). De seguida, através da condição “if c not in k:” é testado se o número não existe na lista “k”, se esta der “True”, será executado o código no interior da condição o qual adiciona o número gerado “c” à lista “k”. Após ser,

gerada a lista “k” com 4 elementos, é criada uma variável do tipo string “num” vazia, o programa converte tal lista através de um ciclo “for”, o qual corre todo o comprimento da lista “len(k)” e armazena todos os valores de “k” em modo string na variável “num”. Deste modo, a função da return da string “num” a qual é um número de quatro dígitos todos diferentes.

O programa inicializa a variável “tentativas” como uma lista vazia, na qual serão armazenadas todas as tentativas realizadas durante o jogo, e a variável “cont” a qual funcionará como contador de tentativas realizadas pelo jogador.

Dá-se início ao ciclo principal do programa, o qual apenas terminará se o número introduzido for igual ao número gerado pelo computador, ou seja, “4T” 4 touros. Durante a execução deste ciclo, a variável “cont” será incrementada após cada número introduzido pelo jogador. É inicializada, uma variável “num” a qual irá armazenar o número introduzido pelo jogador e uma variável “x” a qual é inicializada a 0 e apenas servirá de controlo. Para se fazer o input de um número válido, dá-se início a mais um ciclo while. Neste, é solicitado ao jogador um código (número de quatro dígitos e todos diferentes). Para controlar o número introduzido pelo jogador foi utilizada a condição *”if len(num)==4 and num[0]!=num[1] and num[2]!= num[3] and num[0] != num[2] and num[1]!=num[3] and num[0]!=num[3] and num[1]!=num[2]”* a qual confirma se o número introduzido contém 4 algarismos e se cada dígito do número é diferente dos restantes. Se tal condição se verificar, a variável “x” é igualada a 1 o que faz o ciclo terminar. Após a validação do input, através da igualação das variáveis “n, tentativa” a função “acertos(num_g,num,cont,tentativas)” é chamada a função “acertos(num_g,num,cont,tentativas)”. Esta função, tende a comparar o número introduzido pelo jogador com o número gerado pelo computador, dando print do número de porcos e touros entre ambos. Assim, a função inicializa as variáveis “porcos” e “touros” a zeros de modo a poder armazenar o número de acertos

em cada uma, sendo que, a variável “porcos” irá ser incrementada quando um dígito existir, mas não estiver no lugar correto e a variável “touros” será incrementada quando o dígito coincidir com o dígito gerado e se encontrar no mesmo lugar. Tudo isto é verificado através dos ciclos “for” que se seguem. O primeiro, consiste em correr cada dígito do número introduzido “num” e comparar com cada dígito do número gerado “num_g” sendo que, se a condição “*if num[x]==num_g[x] and x==y:*” der verdadeira ou seja, o dígito do número introduzido for igual ao número na mesma posição do número gerado a variável “touros” será incrementada. De seguida, dá-se início ao segundo ciclo “for” o qual irá correr cada dígito do número introduzido e através da condição “*if num[x] in num_g:*” ou seja, o dígito do número introduzido existe no número gerado, a variável “porcos” será incrementada. Visto que, se o dígito estiver contado na variável “touros” (igual números e posição) tal não se pode encontrar contabilizado na variável “porcos” (dígito existe no número gerado mas não na posição certa) é realizada a operação de subtração entre a variável “porcos” e a variável “touros”. Deste modo, é realizada a comparação entre os dois números. De seguida, a condição “*if touros==4:*” será executada se o número introduzido for igual ao número gerado ,ou seja, “4T” (4 touros). Se tal condição for executada, a variável “acert” será igualada a “4T”(número correto), e a variável “text” através de uma formatação de strings será igualada ao valor número da tentativa (controle), ao número introduzido (num) e a variável “text”, a qual indica a quantidade de números acertados. De seguida, a lista “c” a qual contém todas as tentativas já introduzidas anteriormente será igualada a ela própria (c) mais a variável “text” convertida em lista através de “[]”, deste é introduzida uma nova tentativa a lista total de tentativas. O “print” que se segue, indica ao jogador que acertou no número e seguidamente o programa dará return das variáveis “acert” e “c” através de um tuplo as quais serão iguais as variáveis “n” e “tentativas” do programa principal. Por sua vez, se o

número introduzido não for igual ao número gerado pelo computador será executada a condição “else:” na qual a variável “acert” através de uma formatação de strings será igualada ao número de porcos (porcos) e touros (touros) acertados e a variável “text”, da mesma forma será igualada ao número da tentativa (controlo), ao número introduzido (num) e a variável “text” a qual indica a quantidade de números acertados. A condição “*if porcos==0 and touros==0:*” foi utilizada para conferir se não foi acertado qualquer número. Se tal for executado, o programa apenas adicionará a variável “text” a lista “c” e dará return das variáveis “acert” e “c” em forma de tuplo. Se tal não for executado, o programa dará print da variável “acert” a qual contém o número de porcos e touros acertados e por sua vez, também adicionará a variável “text” a lista “c” e dará return das variáveis “acert” e “c” em forma de tuplo. Deste modo, o tuplo devolvido da função “acertos” fará com que a variável “n” seja igualada a variável “acert” e a lista “tentativas” a lista “c”. Assim, se a variável “n” tiver o valor de “4T” o ciclo while terminará. Desta forma, o programa chega as últimas linhas de código as quais através de um ciclo “for” que dará print de todas as tentativas armazenadas na lista “tentativas”.

Bibliografia:

- 1- Proposta de trabalho fornecida pela professora
- 2- Powerpoints fornecidos pela professora
- 3- <https://docs.python.org/3.2/library/random.html> (consultado a 21/12/2017)