Programação I

Universidade de Évora

Curso de Engenharia Informática

Projeto: Pigs and Bulls Reverse

Trabalho realizado por:

Yaroslav Kolodiy nº39859

Eduardo Medeiros nº39873

Eduardo Eloy nº40129

Introdução

Este trabalho foi realizado a pedido da professora Teresa Gonçalves da cadeira de Programação I e tem como objetivo a criação, por parte dos alunos, de uma réplica do jogo “Pigs and Bulls” inversa ao trabalho feito anteriormente. A linguagem de programação utilizada é a lecionada durante o semestre, sendo esta Python.

Este relatório consiste na explicação detalhada do programa, apresentação das funções e variáveis utilizadas.

Breve resumo

Pigs and Bulls é um jogo para 2 jogadores cujo objetivo de um dos jogadores é adivinhar o código pensado pelo outro jogador. O código em que o jogador pensa é composto por 4 algarismos diferentes pertencentes ao intervalo de 0 a 9.

O jogador pensa num código; o computador propõe um código e o jogador indica o número de algarismos certos. Existem duas indicações distintas que o jogador pode fornecer: se os algarismos estão nas posições certas, são" touros", se estiverem no código, mas em posições diferentes, são" porcos". Por exemplo, se o código for 4271 e a tentativa for 1234, a resposta deve ser 1 touro e 2 porcos (o touro é "2" e os porcos são "4" e "1").

Deste modo, de uma forma ainda mais sucinta, o nosso programa irá funcionar da seguinte forma: o jogador pensa num código e o computador tenta adivinhá-lo consoante as pistas fornecidas pelo jogador.

Funções utilizadas no programa

Função:	**Argumentos e retorno:**
analise_resultado(*resultado*): esta função analisa o resultado fornecido pelo jogador aproveitando apenas os valores dos touros e dos porcos.	**Argumentos:** resultado - input do jogador, string **Valor de retorno:** geral - numero de touros e porcos, tuplo de inteiros
lista(): esta função gera uma lista com todas as hipóteses possíveis.	**Valor de retorno:** lista – lista que armazena todas as hipóteses
calculador_ten(*possiveis,escolhido*): esta função devolve o número de porcos e de touros do código gerado pelo computador e compara este resultado com o resultado do código anterior.	**Argumentos:** possiveis – elemento da lista de hipóteses, lista escolhido - código fornecido na jogada anterior pelo computador, string **Valor de retorno:** tuplo de 2 variáveis, os touros e os porcos em comparação com os números
retirar_lista(*lista,escolhido,score*): esta função consiste na remoção dos códigos que não fazem sentido na lista de hipóteses através de uma comparação entre o código fornecido anteriormente e o seu resultado.	**Argumentos:** lista - lista total de hipóteses, lista de strings escolhido- código anterior, string score- resposta do utilizador, tuplo de inteiros **Valor de retorno:** lista_nova - lista atualizada com os códigos que podem ser que o utilizador quer, lista
random_num(): função responsável pela geração de um numero aleatório.	**Valor de retorno:** num - número de quatro algarismos, string.

Variáveis Globais: lista_total = lista(), tent_comp = [], respostas= []

Descrição do programa

O programa começa por declarar as variáveis globais: lista_total que é inicializada com o valor da função lista(), esta função através de sucessivos ciclos for gera uma lista na qual estarão todos os códigos possíveis de 4 algarismos e sem repetições, isto é feito através de condições if existentes entre cada loop for, que ocorrem até a lista ter o tamanho desejado que para o nosso objetivo é 5040. De seguida dá-se a inicialização da variável tent_comp que é inicializada como uma lista vazia, na qual mais tarde se irão armazenar todas as tentativas executadas pelo computador, a variável "respostas", é inicializada como uma lista vazia e nesta lista serão armazenadas todas as respostas fornecidas pelo jogador.

Dá-se o print de uma frase introdutória que indica ao jogador que pode começar a jogar.

A partir deste ponto dá-se inicio do ciclo chile, este ciclo irá correr ate que o computador adivinhe o código que o utilizador está a pensar e este responder "4t 0p" ou até não existerem mais códigos possíveis, tal pode acontecer se o jogador se enganou o tentou fazer "batota" contra o computador.

Dentro do loop principal o programa começa por escolher um elemento da lista aleatoriamente (isto é feito utilizando a função random() do modulo random) e inicializar a variável escolha_pc com esse valor, depois adiciona tal elemento á lista onde estão a ser armazenadas todas as tentativas realizadas pelo computador. De seguida através de uma formatação de strings é imprimido o código que o computador escolheu e é pedida a resposta ao utilizador, a qual é armazenada na nova variável "pontuacao" e na variável onde são armazenadas as respostas do utilizador (a variável "respostas"). Seguidamente, a variável "pontuacao" será igualada ao resultado da função "analise_resultado(pontuacao)" a qual devolve um tuplo apenas com 2 dígitos, o primeiro será o número de "touros" e o segundo o número de "porcos"

A partir deste ponto dá-se inicio á próxima parte do programa, na qual será testada a inteligência do computador. Através da condição "*if* pontuacao == (4, 0):" é testada se o código escolhido pelo computador é o código que o jogador pensou, se tal for verdade o programa imprime "ACERTEI!" e terminara o ciclo. Mas se tal não se realizar o programa passará para a fase seguinte a qual consiste em apagar todos os códigos que não podem ser o pensado pelo utilizador.

Isso é feito através da atribuição de um novo valor á variável lista_total, o valor da função retirar_lista() utilizando lista_total , escolha_pc e “pontuação” como argumentos (retirar_lista(lista_total,escolha_pc,pontuacao)). Esta função ira criar a variável “lista_nova”, uma lista na qual serão armazenados os possíveis códigos pensados pelo utilizador.

De seguida dá-se o inicio de um ciclo for, que corre a lista total de códigos (a variavel lista_total) um por um, e através da condição “*if* (calculador_ten(*lista*[c], *escolhido*)==*score*):”, que recorre a função “calculador_ten()” para verificar através de dois ciclos for se o número de porcos e touros é igual ao número de porcos e touros inserido pelo utilizador. E assim a lista_nova é sempre atualizada com os códigos que o computador determina serem possíveis candidatos para o código que o jogador está a pensar.

Apos tal acontecer existem duas hipóteses ou o computador adivinha o número que o jogador estava a pensar, ou a lista estará vazia e executar, no primeiro caso o jogador insere “4t 0p” e o loop while termina, no segundo caso o programa indicará ao jogador que este fez “batota” ou que se enganou ao inserir os touros e porcos.

Dado por terminado o ciclo principal o programa imprime para o ecrã todas as tentativas do computador e as respetivas respostas do jogador através de um ciclo for que percorre o tamanho da lista com todas as tentativas executadas pelo computador(a variável tent_comp).

Assim termina o programa.

Conclusão e comentários críticos

Foram realizados vários testes para encontrar a melhor opção de como gerar a lista de todos os códigos apresentados, mas tais demoravam muito tempo.

Para alem disso foram também executadas varias tentativas ao programa para verificar se este realmente funciona, este apenas não adivinhou o código certo quando era introduzido pelo jogador um valor errado de porcos e touros.

Foram também realizadas outras tentativas para verificar outros métodos para retirar os códigos que não podiam ser os códigos pensados pelo jogador, mas tais foram malsucedidos sendo o método apresentado na nossa opinião o mais fácil e eficaz.