# O jogo das cartas

Trata-se de um jogo de cartas para dois jogadores (um humano e o computador), que se desenrola em 10 jogadas com um baralho digital de 40 cartas para o computador e um baralho em papel de 40 cartas para o jogador humano. Semelhante ao denominado "jogo da batalha", as regras, para cada jogada, são simples:

* Cada jogador retira uma carta do seu baralho e as duas cartas são confrontadas;
* O jogador que tiver a carta mais alta ganha 2 pontos (1 ponto por carta);
* Caso as cartas tenham a mesma figura, ocorre um empate, e cada jogador retira mais uma carta. Quem ganhar recebe 4 pontos (2 pontos por cada carta).
* E no caso de ocorrerem empates seguidos, adicionar-se-ão sempre mais 2 pontos por cada duas cartas que se confrontam.

# Primeira versão do jogo

Para criarmos o jogo, necessitamos de definir algumas variáveis e literais globais.

import random

BARALHO = "23456QJK7A"

# pontuacao dos dois jogadores

pontosComputador = 0

pontosHumano = 0

# numero maximo de jogadas

jogadas = 10

# conta as cartas em caso de empate

empate = 0

A literal BARALHO contém a sequência de figuras das cartas do baralho digital, ordenadas em valor crescente, portanto, do menor valor (2) para o maior (A).

|  |  |
| --- | --- |
| **Head with gears** | Observe em que posição se coloca a carta 7. Não aparece imediatamente a seguir à carta 6!  O baralho digital só tem 10 cartas, em vez das 40 cartas de um baralho normal! Terá isto algum impacto negativo no funcionamento do jogo? |

Em seguida, definem-se as varáveis da pontuação do jogador humano e do computador, inicializadas a zero. Finalmente o número máximo de jogadas (10). Há ainda uma variável adicional (empate) que permite contabilizar a pontuação a atribuir em caso de empate ou empates seguidos.

Definiremos uma função jogada() para interagir com o jogador humano. O jogador retira uma carta do seu baralho (em papel) e é chamada a função jogada()com a figura da carta como argumento.

O algoritmo para cada jogada pode ser o seguinte:

1. O computador retira uma carta;
2. Apresentam-se as cartas de ambos os jogadores;
3. **Subproblema** logica do jogo;
4. Se terminaram as 10 jogadas então:  
   **Subproblema** apresentar o vencedor

Para simplificar, deixamos como subproblemas: 1) logica do jogo, que consiste na aplicação das regras do jogo e determinação das pontuações, que são variáveis globais; 2) apresentar o vencedor.

Observe, a implementação da função jogada():

def jogada(cartaHumano):

print("\*\*\* Jogada \*\*\*")

# 1. jogada do computador

cartacomputador = random.choice(BARALHO)

# 2. apresenta ambas as cartas

print("Computador:", cartacomputador)

print("Humano:", cartaHumano)

# 3. subproblema logica de jogo

logica\_jogo(cartacomputador, cartaHumano)

# 4. verifica se já terminaram as jogadas

if jogadas == 0:

print("O jogo terminou...")

# subproblema vencedor

vencedor()

else:

print("Tem", jogadas, "jogadas.")

|  |  |
| --- | --- |
| **Eye** | A figura da carta do computador é obtida através da função random.choice()com a cadeia de caracteres do BARALHO.  Mas esta função não garante que não saia a mesma figura várias vezes repetida... |

Antes de iniciar a lógica de jogo, é conveniente definir uma função simples que determine o valor da figura da carta.

# retorna numero de ordem da figura no baralho

def valor(carta):

return BARALHO.index(carta)

Uma vez que o literal BARALHO tem as figuras das cartas em sequência, utiliza-se o método [index()](https://www.w3schools.com/python/ref_list_index.asp) para determinar o seu número de ordem.

O algoritmo da lógica de jogo será uma estrutura de seleção múltipla com as seguintes condições:

* Se carta computador > carta humano então:
  + Atribuir pontos ao jogador humano
  + decrementar uma jogada
* Se não, se carta computador < carta humano
  + Atribuir pontos ao computador
  + decrementar uma jogada
* Se não... Empate.

Observe que a atribuição dos pontos ao vencedor inclui, não só as 2 cartas em confronto, mas também as cartas anteriores, em caso de empates prévios. No caso de empate, mantém-se a mesma jogada até ocorrer um desempate.

O código da lógica de jogo é o seguinte.

def logica\_jogo(comp, hum):

global pontosComputador, pontosHumano, empate, jogadas

if valor(comp) > valor(hum):

print(2 + empate \* 2, "pontos para o computador.")

pontosComputador += 2 + empate \* 2

jogadas -= 1

empate = 0

elif valor(comp) < valor(hum):

pontosHumano += 2 + empate \* 2

print(2 + empate \* 2, "pontos para o jogador humano.")

jogadas -= 1

empate = 0

else:

empate += 1

print("Empate... Jogue novamente...")

# pontuação

print("Computador", pontosComputador, "vs", pontosHumano, "Humano")

|  |  |
| --- | --- |
| **Balloon animal** | Experimente esta primeira versão do jogo, através da linha de comando.  *Para cada jogada chame a função* jogada() *com a sua carta:*  >>> jogada ("7") |

# Uma versão melhorada!

|  |  |
| --- | --- |
| **Head with gears** | Certamente achou que a necessidade de chamar a função jogada() a cada jogada quebrava um pouco a sensação de "imersão" no jogo. Seria possível que o próprio computador realizasse a iteração entre jogadas, de forma automática?  A resposta é afirmativa... Podemos utilizar um ciclo [while](https://www.w3schools.com/python/python_while_loops.asp). Abordaremos esta estrutura de controlo no capítulo seguinte. Para já fique com a noção que esta estrutura repete sempre o seu bloco de código enquanto a expressão condicional for avaliada como verdadeira.. |

Observe que o facto de utilizarmos variáveis globais complica a leitura do algoritmo. Nesta nova versão, iremos apenas utilizar variáveis locais, como princípio de boa programação. Apenas a literal BARALHO se manterá

A função jogo() que vamos criar, engloba todas as variáveis de jogo e também a lógica de jogo

Antes de passarmos à função jogo() , relembraremos a função valor()e apresentaremos a função vencedor().

A função vencedor segue o seguinte algoritmo:

1. Cabeçalho indicando o final do jogo
2. Estrutura condicional de seleção múltipla para determinar resultado final do jogo:
   1. Vitória do jogados humano;
   2. Vitória do computador;
   3. Empate

import random

BARALHO = "23456QJK7A"

# retorna numero de ordem da carta no baralho

def valor(carta):

return BARALHO.index(carta)

# mostra o vencedor

def vencedor(pontosComputador, pontosHumano):

print("")

print("O jogo terminou...")

print("---- Resultado final ----")

print("Computador", pontosComputador, "vs", pontosHumano, "Humano")

if pontosHumano > pontosComputador:

print ("Voce ganhou!")

elif pontosHumano < pontosComputador:

print("Voce perdeu")

else:

print("Empatamos...")

# jogo

def jogo(jogadas):

# pontuacao dos dois jogadores

pontosComputador = 0

pontosHumano = 0

pontos = 2

while jogadas>0:

print("\*\*\* Jogada \*\*\*")

print("Computador", pontosComputador, "vs", pontosHumano, "Humano")

# input - jogada do computador e do jogador humano

cartaHumano = input("Humano: ")

cartacomputador = random.choice(BARALHO)

print("Computador:", cartacomputador)

# logica de jogo

if valor(cartacomputador) == valor(cartaHumano):

# empate

print("Empate... Jogue novamente...")

pontos += 2

elif valor(cartacomputador) > valor(cartaHumano):

print(pontos, "pontos para o computador.")

pontosComputador += pontos

jogadas -= 1

pontos = 2

else:

print(pontos, "pontos para o jogador humano.")

pontosHumano += pontos

jogadas -= 1

pontos = 2

# final da jogada

print("Tem", jogadas, "jogadas.")

vencedor(pontosComputador, pontosHumano)

|  |  |
| --- | --- |
| **Head with gears** | Observe a função jogo() e determine as principais alterações face à versão anterior do jogo. |

Confira as principais alterações ocorridas:

* As variáveis globais pontosComputador e pontosHumano passaram a variáveis locais da função jogo();
* A variável global jogadas passou a ser um parâmetro da função jogo(). Isto permite configurar o jogo em função do número de jogadas com que prefere jogar;
* A variável global empate foi substituída pela variável local pontos;
* A interação com o jogador passou a ser feita com a função input().
* A condição de término do jogo passou para o ciclo while, que repete as jogadas indefinidamente até a variável jogadas atingir o valor 0.

# Desafio final

Altere o jogo de forma a que o jogo não termine após um determinado número de jogadas, mas quando um dos jogadores atingir um determinado valor de pontos (10 pontos, por exemplo, que seria definido como parâmetro da função jogo()).

|  |  |
| --- | --- |
| **Share with person** | Partilhe o **seu** jogo! |