

## Problema 1: Super Trunfo – Uma nova versão

### Problema

Lançado pela Grow na década de 70, Super Trunfo foi um dos jogos de maior sucesso feito pela empresa. Ainda que popularizado nos anos 80, hoje o jogo ainda é comercializado e é considerado um bom passatempo para todas as idades. Uma característica marcante do Super Trunfo é o lançamento periódico de novos temas, agradando fãs e colecionadores diversos. Dentre alguns dos temas estão carros, aviões, cães de raça, dinossauros, personagens Disney, entre outros.

Ao longo dos anos, empresas tentaram produzir jogos semelhantes como o Super Coluna e o 4Match. Contudo, apesar de rivalizarem com o Super Trunfo, não fizeram tanto sucesso e também não são mais comercializados.

Buscando se lançar no mercado de jogos, a empresa Asuka Productions viu grande potencial neste jogo clássico de cartas e decidiu lançar uma nova versão do mesmo. Como uma empresa fomentadora do conhecimento, a Asuka Productions estabeleceu uma parceria com o Departamento de Ciências Exatas (DEXA) do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS).

Foi então solicitado aos alunos do MI de Algoritmos que desenvolvessem a nova versão do jogo Super Trunfo, de acordo com os seguintes critérios exigidos pela empresa:

1. O desenvolvedor do jogo deverá oferecer para a empresa uma sugestão interessante de tema para a nova versão do Super Trunfo e implementar o jogo com este tema.
2. O jogo deve possuir uma tela inicial oferecendo a opção de abrir uma tela com informações sobre o jogo, a opção de sair do jogo e a opção de iniciar uma nova partida.
3. A plataforma deve comportar exatamente 2 jogadores que devem informar inicialmente seus nomes ou *nicknames*.
4. Os jogadores devem informar o número máximo de rodadas para a partida. O número mínimo de rodadas deve ser 3.
5. Cada jogador deverá receber três cartas por rodada. Cada carta deve possuir um identificador como no seguinte exemplo: *Monstro 1 - Jogador1*, *Monstro 2 – Jogador 1* e *Monstro 3 – Jogador 1*. Da mesma forma para o jogador 2. (Substituir a palavra *Jogador* pelo nome ou *nickname* do respectivo jogador)
6. As cartas ainda devem conter pelo menos 3 atributos e no máximo 4. Estes atributos devem ser escolhidos pelo desenvolvedor, a partir do tema proposto. Caso o desenvolvedor opte por 3

atributos, todas as cartas de sua versão do jogo deverão conter 3 atributos. Da mesma forma, caso escolha 4 atributos, todas as cartas de seu jogo deverão conter 4 atributos.

7. Os pesos (valores numéricos) dos atributos devem ser atribuídos aleatoriamente a cada carta em cada rodada. Além disso, cada atributo deve ter uma pontuação associada, diferente dos outros atributos (definido pelo desenvolvedor).
8. O jogo deve escolher, também aleatoriamente, o jogador que irá iniciar a partida. Após esta definição, o fluxo do jogo se mantém alternando a vez de cada jogador em cada rodada até encerrar o número de rodadas definidas.
9. Em uma rodada, o jogador que possui a vez deve ser capaz de visualizar suas cartas e os atributos de suas cartas, mas não as de seu oponente. O jogador ainda deve ser capaz de escolher a opção de atributo que lhe for mais conveniente e a carta a que lhe dará a maior probabilidade de vencer. Ao realizar sua escolha, suas cartas devem sair da tela e o jogo aguarda o segundo jogador confirmar que está na frente da tela para exibir seus dados.
10. Na mesma rodada, o segundo jogador visualiza qual atributo foi escolhido pelo jogador 1 e escolhe uma de suas 3 cartas, da maneira que lhe for mais conveniente.
11. Ao final da rodada o jogo deve apresentar os valores dos atributos de cada jogador e verificar qual jogador apresentou o maior valor de atributo, contabilizando a pontuação do atributo ao jogador vencedor da rodada.
12. Uma nova rodada é iniciada até que se encerre o número de rodadas limite escolhido pelos jogadores e, portanto, se encerre a partida.
13. Ao encerrar a partida, o nome do jogador com maior pontuação e seu score deve ser exibido na tela.
14. Caso o score do jogador vencedor seja maior que o atual recorde, ele ganhará o status de novo recorde. O nome ou *nickname* do jogador com o novo recorde deve ser registrado e exibido.

### Produto, Fluxograma e Relatório

Você deve construir um fluxograma do algoritmo deste jogo e enviá-lo até ao meio dia do dia **07/11/2019** (A entrega impressa pode ser solicitada pelo tutor).

Além disso, você deverá também desenvolver o código fonte do sistema em Python e um relatório final, no formato de artigo da SBC, conforme modelo e instruções disponibilizados no site do MI. O código deve ser entregue até ao meio dia do dia **19/11/2019** e o relatório final até ao meio dia do dia **21/11/2019** (a entrega impressa do relatório final pode ser solicitada pelo tutor). O relatório final só será aceito mediante entrega do código fonte.

O desempenho nas sessões tutoriais equivale a 30% da nota no Problema. O relatório, que inclui o fluxograma, equivale a 30% da nota e o código fonte (produto) equivale a 40%. Haverá penalidade de 1 ponto por descumprimento do prazo de entrega e 1 ponto por dia de atraso na entrega, até o máximo de cinco dias. Após este prazo, o trabalho não será mais aceito.

Tanto o código fonte quanto o relatório devem ser desenvolvidos individualmente. Deve constar no código fonte declaração de ausência de plágio.

## Recursos para Aprendizagem

ECOMP/UEFS. Site de MI – Programação.  
<http://sites.ecomp.uefs.br/mia-20192/tutorial>

MANZANO, J. A. N. G.; OLIVEIRA, J. F. Algoritmos: Lógica para Desenvolvimento de Programação. São Paulo: Érica, 1996.

FORBELLONE, A. V. L., EBERSPACHER, H. F. Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados. 2. ed. Makron Books, 2000.

WAZLAWICK, R. S. Introdução a Algoritmos e Programação com Python. Elsevier, 2018.

BORGES, L. E. Python para Desenvolvedores. Novatec, 2014.

SUMMERFIELD, M. Programação em Python 3. Elsevier / Altabooks, 2015.

DIERBACH, C. Introduction do Computer Science Using Python: A Computational Problem-Solving Focus. Wiley, 2012.

BEAZLEY, D.; JONES, B. K. Python Cookbook. O'Reilly, 2013.

BARRY, P. Use a Cabeça! Python. Elsevier / Alta Books, 2013.

## Cronograma

Data	Sessão Tutorial
17/10/19	SIECOMP – Integração dos estudantes
24/10/19	Sessão Tutorial Inicial - Problema 1
31/10/19	Sessão Tutorial - Problema 1
07/11/19	Entrega do Fluxograma.
14/11/19	Sessão Tutorial - Problema 1
19/11/19	Entrega do Código-Fonte.
21/11/19	Entrega do Relatório.
21/11/19	Apresentação do Problema 2