INF250 Assignment A1: Introduction to Matlab and Math

Note: Deliver everything in one archive-file (.zip) on Mitt.UiB.no
Deadline: 2. 2. 2018, 23:59

1 Snakes & Ladders 30%

1.) Implement a dice MatLab-function for the snakes and ladders game as `rolldice.m` that rolls a die. The output should be a single integer value between 1 and 6.

Delivery: file `rolldice.m`

2.) Finish the plot function for the game tokens in the file `moveGamePiece.m`, look for the `TODO` sections. The tokens should be clearly visible on the board, have the chosen colors and not occlude each other when the pieces are on the same board position. Make sure that the tokens move from left to right on one level and then from right to left on the next one.

Delivery: modified file `moveGamePiece.m`

Of course, the whole game should be operational.

3.) You can get extra credits for special features. You can be creative and create a rigged die. For example higher numbers for the first player, or you could check for snakes and try to avoid those. You could activate those special features only for some specific names as input. You could load images for the game tokens and display those.
Be creative and surprise us (for a bit of optional extra credit)!

2 Princess' dilemma 50%

Imagine you are a princess, the only child of your father, the King, and are looking for a suitable prince to rule your subjects with you one day. Therefore you would like to find the most charming, most intelligent, most caring, most ... well, the best husband you can find. From the kingdoms far and wide only ten potential candidates are available about which you know absolutely nothing whatsoever. Hence you will have to invite them over, one by one, to see if they are fit for you. Once you invite someone over you will discover his qualities and will have to decide whether or not to marry him, before meeting another one. If you invite a prince over and refuse to marry him you will break his heart and he will never talk to you again. Let's suppose the prince's qualities can be represented by 64 bit integer value. Using the best of k algorithm, create a simulation that will help you pick the best future king. The algorithm work's as follows:

1. You invite the princes in arbitrary order
2. The first k you will turn down.
3. You marry the next one that will be better than all of the first k
4. If it turns out there are no more candidates, you don't marry

Your task is now to figure out what k is the best for your eternal happiness...

Write a MatLab program that will for given k and a list of numbers (the princes) tell you the rank of the prince you marry. This should be done in file named `best_k.m`. The output value

are in range between -1 and 10 and the meaning is as follows: 1 the best; 10 the worst; -1 didn't marry. Run the program 1000 times for different *k*-s from the interval 1 to 10 and for different lists of princes. Plot for different *k* how many times did you wed to the best one, the second-best, and not at all. Write a short pdf where you reason which *k* would you pick as your strategy for getting the most excellent spouse. Include the charts in the pdf. The code supporting your decision should be a script file named `main.m` and should run the simulation and output the chart.

Delivery: pdf document `hand-in.pdf`, code `best_k.m`, and `main.m`

3 Vectors, 10%

Given the vector **v**= $\begin{pmatrix} 7 \\ 1 \\ 5 \end{pmatrix}$ and the vector **u** = $\begin{pmatrix} -3 \\ 1 \\ 2 \end{pmatrix}$ compute the following:

1. dot(**u**,**v**)
2. $\mathbf{u}^\top \mathbf{v}$ (consider u and u as matrices)
3. $\mathbf{u}\mathbf{v}^\top$ (consider u and u as matrices)
4. $\mathbf{u} \times \mathbf{v}$ (the cross product)
5. The angle θ between **u** and **v**

Delivery: the solution in a pdf file named `hand-in.pdf`

4 Matrices, 10%

1. Given the matrices **M1** = $\begin{pmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 5 \end{pmatrix}$ and **M2** = $\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 4 & 2 \end{pmatrix}$, compute $\mathbf{M1} \cdot \mathbf{M2}^\top$.
2. Given the matrices **A** = $\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ and **B** = $\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$, compute $\mathbf{A} \cdot \mathbf{B}$.
3. Compute the inverse of $\mathbf{M1} \cdot \mathbf{M2}^\top$
4. Compute the determinant of **A**, **B**, $\mathbf{A} \cdot \mathbf{B}$

Delivery: the solution in a pdf file named `hand-in.pdf`