

## SISTEMAS SENSORIAIS

MIEEC & MIEBM & MIEI & MIEF

2017-2018

### 1º Trabalho de Avaliação







#### *Deteção e Reconhecimento de Jogos de Xadrez*

##### 1 Introdução

O Xadrez é um desporto, também considerado uma arte e uma ciência. Pode ser classificado como um jogo de tabuleiro de natureza recreativa ou competitiva para dois jogadores, sendo também conhecido como Xadrez Ocidental ou Xadrez Internacional para distingui-lo dos seus antecessores e de outras variantes atuais.

A forma atual do jogo surgiu no sudoeste da Europa na segunda metade do século XV, durante o Renascimento, depois de ter evoluído a partir das suas antigas origens persas e indianas.

O jogo de xadrez é realizado num tabuleiro de casas claras e escuras, sendo que, no início, cada jogador controla dezasseis peças com diferentes formatos e características:

- |   |  |
|---|--|
| •  - Rei;    | •  - Torre;  |
| •  - Rainha; | •  - Cavalo; |
| •  - Bispo;  | •  - Peão.   |

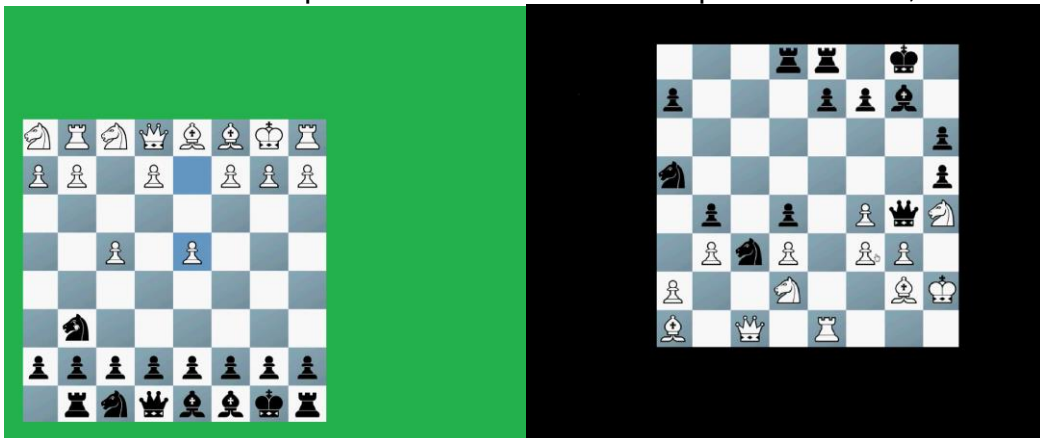
## 2 Objectivos

### 2.1 Níveis do Trabalho

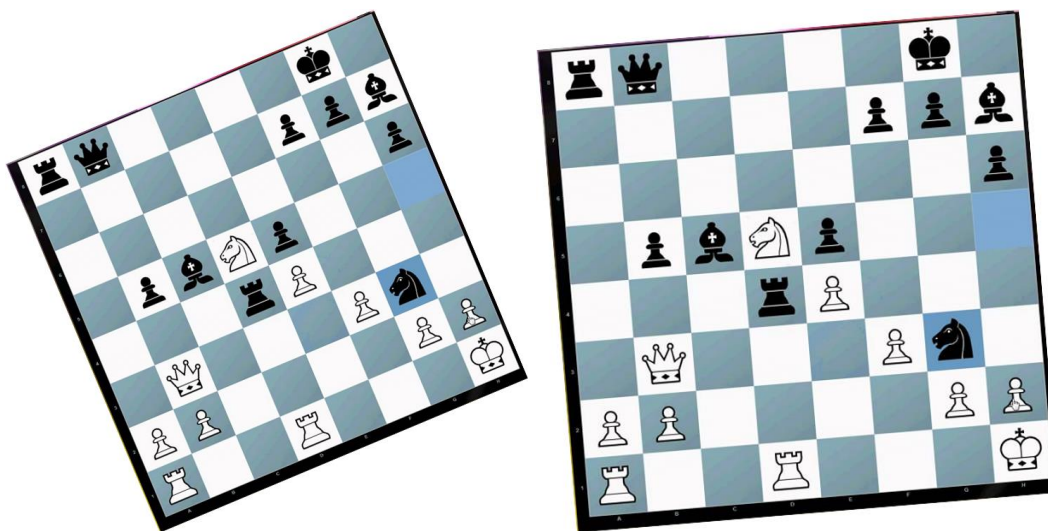
O trabalho consiste na detecção do tabuleiro de xadrez, do ângulo que este apresenta na imagem e de quais as peças que estão em cada casa do tabuleiro.

Este trabalho poderá ser desenvolvido em 3 níveis:

- Nível 1 – Tabuleiro digital, em fundo uniforme, com um ângulo, entre o canto inferior esquerdo e o canto superior direito, de  $45^\circ$ .



- Nível 2 – Tabuleiro digital com moldura simples, em fundo uniforme, com ângulo que pode ser diferente de  $45^\circ$ .



- Nível 3 – Com dois tipos de imagens: tabuleiro digital com ângulo de  $45^\circ$  em fundo não-uniforme; e imagem fotográfica de Tabuleiro em fundo irregular, com diferentes ângulos.



## 2.2 Requisitos da Aplicação a desenvolver

O programa deverá:

- disponibilizar ao utilizador:
  - uma interface que permita aplicar à imagem sobre a qual se está a trabalhar todas as funções desenvolvidas nas aulas práticas; e
  - uma interface adequada à deteção e reconhecimento do tabuleiro e das peças de xadrez.
- identificar o tabuleiro (sua localização, dimensão e ângulo);
- identificar a peça que está em cada casa do tabuleiro (quando está);
- tolerar alguma variação de:
  - dimensão entre imagens;
  - iluminação entre imagens;
  - deformidade nas imagens fotográficas (tabuleiro deformado ou oval - nível 3).

A nota do trabalho terá em conta a implementação de requisitos extra que melhorem o trabalho.

O programa será avaliado com imagens que contêm apenas um tabuleiro.

## 3 Avaliação

O trabalho será objeto de uma avaliação semi-automática que incluirá a análise de um conjunto de imagens fornecidas pelos docentes e de um outro conjunto de imagens idênticas, mas que não serão disponibilizadas aos alunos.

Na maioria dos casos não haverá discussão presencial do trabalho. Contudo caso os docentes vejam necessidade irão contactar os alunos para marcar uma discussão presencial, a realizar em Dezembro, na semana de Avaliação Contínua.

João Pedro Carvalho, Filipe Moutinho & José Manuel Fonseca ©

Última atualização: 26 de outubro de 2017

A avaliação semi-automática vai ter em conta a existência ou não das funções, o seu tempo de execução e os resultados obtidos.

O trabalho (projecto, código fonte, executável e relatório em PDF) deverá ser entregue na plataforma moodle na data fixada para o efeito.

O relatório (obrigatório) deverá descrever os algoritmos utilizados para o reconhecimento do tabuleiro e das peças, bem como das suas localizações e apresentar sucintamente todos os métodos utilizados. O relatório deverá ser estruturado de forma a que o leitor compreenda como foi pensado e criado o sistema, para que no futuro, caso se pretenda, se possa vir melhorar ou alterar o projeto realizado.

**Data de entrega do trabalho:** 15 de Dezembro de 2017.

Nota: Considera-se uma penalização na nota máxima de 2 valores por cada dia de atraso na sua entrega. Exemplo: 2 dias de atraso = nota máxima possível 16v.

## 4 Operações – Métodos *ImageClass*

### 4.1 Métodos obrigatórios

1	<b>Negative</b>	Negativo de uma imagem
void Negative(Image<Bgr, byte> img) - recebe a imagem a alterar		
2	<b>BrightContrast</b>	Ajuste de brilho e contraste.
void BrightContrast(Image<Bgr, byte> img, int bright, double contrast) - recebe a imagem a alterar, o valor de brilho e de contraste		
3	<b>RedChannel</b>	Copia a componente Red para a Green e Blue.
void RedChannel(Image<Bgr, byte> img) - recebe a imagem a alterar		
4	<b>Translation</b>	Desloca a imagem de (Dx,Dy) =(-10 , -10), preenchendo as partes sem pixeis a preto.
void Translation(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, int dx, int dy) - Recebe a imagem a alterar, uma cópia da imagem e o deslocamento em x e y.		
5	<b>Rotation</b>	Roda a imagem (angulo em radianos) , preenchendo as partes sem pixeis a preto.
void Rotation(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float angle) - Recebe a imagem a alterar, uma cópia da imagem e o angulo de rotação (radianos).		
6	<b>Scale</b>	Aplica um factor de escala a partir de (x,y) = (0,0), preenchendo as partes sem pixeis a preto.
void Scale(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float scaleFactor) - Recebe a imagem a alterar, uma cópia da imagem e o fator de escala.		

João Pedro Carvalho, Filipe Moutinho & José Manuel Fonseca ©

Última atualização: 26 de outubro de 2017

7	<b>Scale_point_xy</b>	Aplica um factor de escala centrado em (x,y), preenchendo as partes sem pixeis a preto.
void Scale_point_xy(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float scaleFactor, int centerX, int centerY) Recebe a imagem a alterar, uma cópia da imagem, o fator de escala e o centro (x, y).		
8	<b>Mean</b>	Filtro de média 3x3 – solução A (analisa 9 pixeis)
void Mean(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
9	<b>NonUniform</b>	Aplica um filtro não-uniforme à imagem, usando os coeficientes e pesos passados como parametro.
void NonUniform(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float[,] matrix, float matrixWeight) - Recebe a imagem a alterar, uma cópia da imagem, uma matrix 3x3 contendo os coeficientes e o peso		
10	<b>Sobel</b>	Filtro de Sobel 3x3
void Sobel(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
11	<b>Diferentiation</b>	Filtro de diferenciação (2x2).
void Diferentiation(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
12	<b>Median</b>	Filtro de mediana a cores usando a avaliação 3D.
void Median(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
13	<b>Histogram_Gray</b>	Calcula o histograma apenas da escala de cinzentos.
int[] Histogram_Gray(Emgu.CV.Image<Bgr, byte> img) - Recebe a imagem a analisar - Devolve o histograma		
16	<b>ConvertToBW</b>	Binarização com valor de threshold recebido como parametro (valor = 157)
void ConvertToBW(Emgu.CV.Image<Bgr, byte> img, int threshold) - recebe a imagem a alterar e o valor de threshold		
17	<b>ConvertToBW_Otsu</b>	Binarização com valor de threshold calculado pelo método de Otsu
void ConvertToBW_Otsu(Emgu.CV.Image<Bgr, byte> img) - recebe a imagem a alterar		
40	<b>Chess_Recognition</b>	Faz o reconhecimento total do tabuleiro, devolvendo a localização/dimensão do tabuleiro, ângulo e as peças em cada casa do tabuleiro.
public static void Chess_Recognition(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, out Rectangle BD_Location, out string Angle, out string[,] Pieces)		

## 4.2 Métodos Facultativos

Lista de métodos que se forem implementados poderão ser testados no Eval:

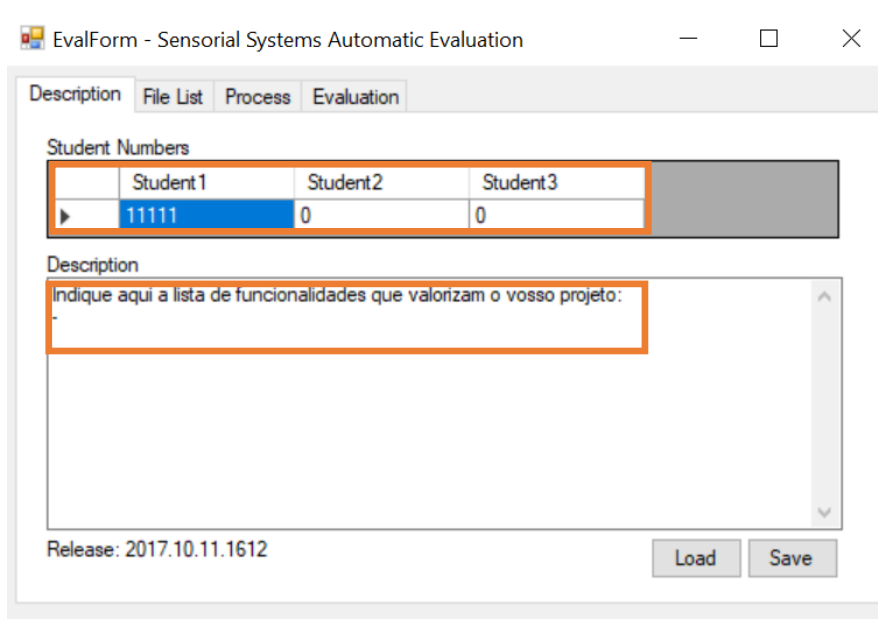
14	<b>Histogram_RGB</b>	Calcula o histograma das componentes B, G e R
int[,] Histogram_RGB(Emgu.CV.Image<Bgr, byte> img) - Recebe a imagem a analisar - Devolve o histograma das 3 componentes B + G + R numa matrix [3, 256]		
15	<b>Histogram_All</b>	Calcula o histograma das componentes Gray, B, G e R
int[,] Histogram_All(Emgu.CV.Image<Bgr, byte> img) - Recebe a imagem a analisar - Devolve o histograma das 4 componentes GRAY + B + G + R numa matrix [4, 256]		
18	<b>Mean_solutionB</b>	Filtro de média 3x3 – solução B (analisa 6 pixels)
void Mean_solutionB(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
19	<b>Mean_solutionC</b>	Filtro de média 7x7 – solução C
void Mean_solutionC(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, int size) - Recebe a imagem a alterar, uma cópia da imagem e a dimensão do filtro		
20	<b>Roberts</b>	Filtro de Roberts
void Roberts(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy) - Recebe a imagem a alterar e uma cópia da imagem		
51	<b>Rotation_Bilinear</b>	Roda a imagem de 30º (ângulo em radianos) , preenchendo as partes sem pixels a preto. Usa Interpolação Bilinear.
void Rotation_Bilinear(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float angle) - Recebe a imagem a alterar, uma cópia da imagem e o ângulo de rotação (radianos).		
61	<b>Scale_Bilinear</b>	Aplica um factor de escala de 0.5x centrado em (x,y) = (0,0), preenchendo as partes sem pixels a preto. Usa Interpolação Bilinear.
void Scale_Bilinear(Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float scaleFactor) - Recebe a imagem a alterar, uma cópia da imagem e o fator de escala.		
71	<b>Scale_point_xy_Bilinear</b>	Aplica um factor de escala de 2.2x centrado em (x,y) = (250,310), preenchendo as partes sem pixels a preto. Usa Interpolação Bilinear.
void Scale_point_xy_Bilinear (Image<Bgr, byte> img, Image<Bgr, byte> imgCopy, float scaleFactor,int centerX, int centerY) Recebe a imagem a alterar, uma cópia da imagem, o fator de escala e o centro (x, y).		

## 5 Eval - Descrição

### 5.1 Painel 1 - Description

No *Eval*, no primeiro painel (figura 1) devem colocar os vossos números de aluno e colocar na descrição uma lista de pontos contendo o que gostariam de chamar a atenção dos professores que forem avaliar o vosso projeto.

Aqui deverão referir os requisitos extras/complementares que valorizem o vosso programa, bem como referir e justificar falhas e limitações que o vosso programa possa ter.



	Student1	Student2	Student3
▶	11111	0	0

Description  
Indique aqui a lista de funcionalidades que valorizam o vosso projeto:

Release: 2017.10.11.1612

Load Save

Figura 1 – Primeiro painel - descrição.

## 5.2 Painel 2 – File List

No segundo painel (figura 2) é apresentada a lista de imagens que serão usadas para teste da aplicação e respetiva informação sobre os tabuleiros. Podem também ser incluídas imagens que não serão para fazer deteção de tabuleiros, mas apenas para testar as aulas práticas. Neste caso, a flag *is\_BD* está não assinalada.

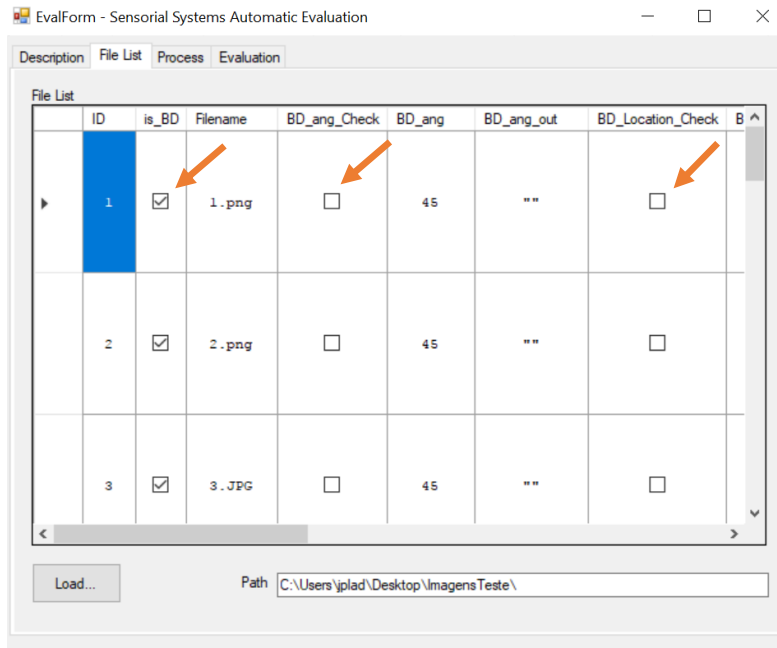
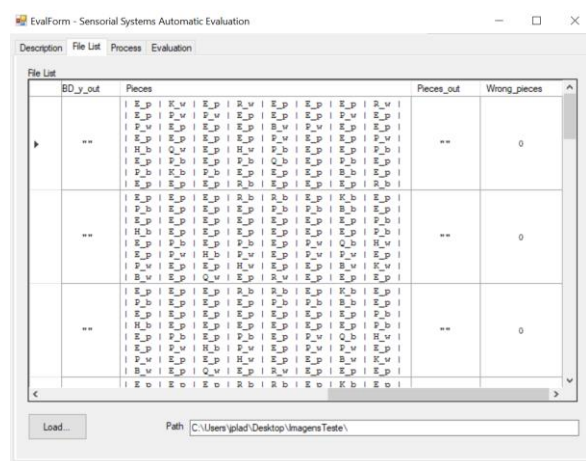


Figura 2 – Segundo painel contendo a lista de ficheiros.

Na Tabela *File List* as colunas *BD\_ang\_Check* e *BD\_Location\_Check* (assinaladas na imagem) indicam se o ângulo (do tabuleiro) e o tabuleiro foram bem localizados.

A coluna *Pieces* mostra a informação de cada casa do tabuleiro:

- E\_p = casa vazia;
- K\_w & K\_b = Rei branco e preto;
- Q\_w & Q\_b = Rainha branca e preta;
- B\_w & B\_b = Bispo branco e preto;
- H\_w & H\_b = Cavalo branco e preto;
- R\_w & R\_b = Torre branca e preta;
- P\_w & P\_b = Peão branco e preto.





Evalform - Sensorial Systems Automatic Evaluation

Description	File List	Process	Evaluation				
<b>File List</b>							
k	BD_center_x	BD_x_out	BD_center_y	BD_y_out	Pieces	Pieces_out	Wrong_pieces
▶	450	**	450	**	E_p   K_w   E_p   R_w   E_p   E_p   E_p   R_w     E_p   P_w   F_w   E_p   E_p   E_p   P_w   E_p     F_p   E_p   E_p   E_p   B_w   F_w   E_p   E_p     E_p   E_p   E_p   E_p   F_w   E_p   E_p   E_p     H_b   Q_w   E_p   H_w   P_b   E_p   E_p   E_p     E_p   P_b   E_p   P_b   Q_b   E_p   P_b   E_p     P_b   K_b   P_b   E_p   E_p   E_p   B_b   E_p     E_p   E_p   E_p   R_b   E_p   E_p   E_p   R_b	**	0
	450	1	450	1	E_p   E_p   E_p   P_b   P_b   E_p   E_p   E_p     P_b   E_p   E_p   E_p   P_b   P_b   B_b   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     H_b   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   P_b   E_p   P_b   E_p   P_w   Q_b   H_w     E_p   F_w   H_b   F_w   E_p   F_w   P_w   E_p     P_w   E_p   E_p   H_w   E_p   E_p   E_w   K_w     B_w   E_p   Q_w   E_p   R_w   E_p   E_p   E_p	E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p	27
	1530	**	1240	**	E_p   E_p   E_p   P_b   P_b   E_p   K_b   E_p     P_b   E_p   E_p   E_p   P_b   P_b   B_b   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     H_b   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   P_b   E_p   P_b   E_p   P_w   Q_b   H_w     E_p   P_w   H_b   P_w   E_p   P_w   P_w   E_p     F_w   E_p   H_w   E_p   E_p   E_p   B_w   K_w     E_p   E_p   Q_w   E_p   E_p   E_p   E_p   E_p	**	0
	1500	**	900	**	E_p   E_p   E_p   E_p   P_b   P_b   E_p   E_p     P_b   E_p   E_p   E_p   P_b   P_b   B_b   E_p     E_p   E_p   E_p   E_p   E_p   E_p   E_p   E_p     H_b   E_p   E_p   E_p   E_p   E_p   E_p   E_p     E_p   P_b   E_p   P_b   E_p   P_w   Q_b   H_w     E_p   F_w   H_b   F_w   E_p   F_w   P_w   E_p     P_w   E_p   E_p   H_w   E_p   E_p   B_w   K_w     B_w   E_p   Q_w   E_p   R_w   E_p   E_p   E_p	**	0
<					H_w   R_w   H_w   O_w   B_w   B_w   K_w   R_w  >		

Load... Path C:\Users\glad\Desktop\Imagens Teste\