



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

*Departamento de Engenharia Informática*

Introdução à Inteligência Artificial  
2020/2021 - 2º Semestre

Trabalho Prático Nº2:

*D31*

*The Rise of the Ballz*

**Nota:** A fraude denota uma grave falta de ética e constitui um comportamento inadmissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude levará à anulação da componente prática tanto do facilitador como do prevaricador, independentemente de acções disciplinares adicionais a que haja lugar nos termos da legislação em vigor. Caso haja recurso a material não original, as **fontes** devem estar explicitamente indicadas.

# 1 Introdução

Depois de muito trabalho a recolher recursos, o D31 decidiu que estava na altura do merecido descanso e dar uns toques na bola. No entanto a sua tarefa não se adivinha fácil. Com tanta cabeçada que deu nas paredes, com as peças adquiridas na lixeira inter-galáctica, o D31 esqueceu-se das coisas mais básicas do futebol.

O objectivo deste trabalho passa assim por ajudar a criar controladores para o D31, de forma a que ele consiga praticar *bom* futebol. Para isso, iremos recorrer a um simulador de treino virtual, que permite o treino de várias tarefas, nomeadamente, defesa, controlo de bola, e jogar contra um adversário.

O principal objectivo é que o controlador seja capaz de conduzir o D31 a ultrapassar um conjunto de tarefas que lhe vão sendo dadas relacionadas com o futebol, de forma a conseguir ganhar ao seu adversário. A resposta do controlador que vai ser criado para o D31 será a intensidade e o ângulo da força que terá de ser aplicada para o poder mover (de forma muito semelhante ao TP1). A Figura 1 mostra um exemplo do ambiente com o qual o controlador terá de lidar.

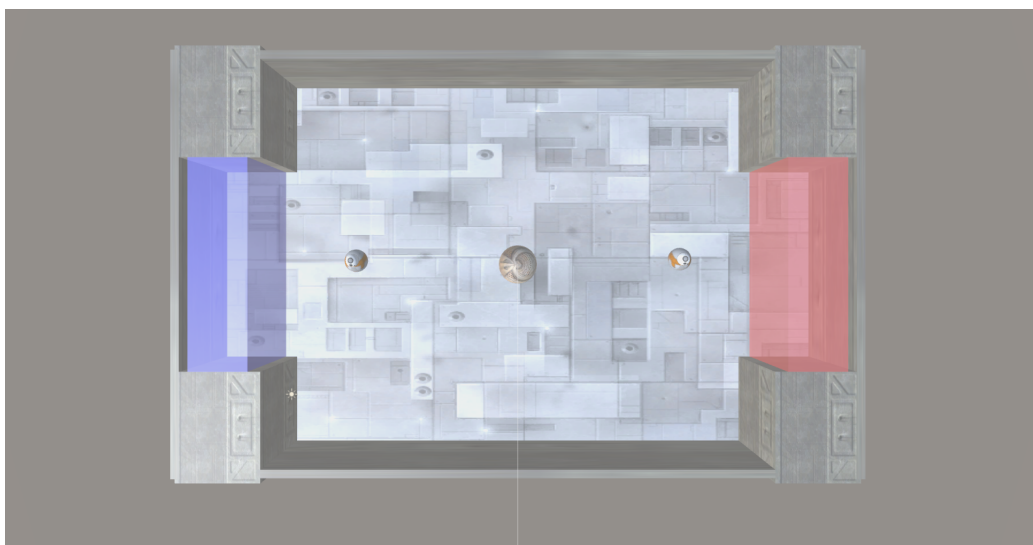


Figura 1: Exemplo do ambiente onde será inserido o controlador.

## 2 Enunciado

Este trabalho prático tem como objectivo principal a aquisição de competências relacionadas com a análise, desenvolvimento, implementação e teste de agentes adaptativos.

Assim, pretende-se desenvolver um controlador para o d31 para realizar tarefas de controlo de bola, defesa e ataque. Para isso irão ser utilizadas redes neuronais e algoritmos genéticos. A rede neuronal será responsável por, dado um conjunto de informações acerca do ambiente, tomar decisões no que toca à intensidade e ângulo da força a aplicar no D31. O algoritmo genético irá aperfeiçoar a rede, para que esta consiga *aprender* e melhorar a sua performance ao longo do tempo. São retiradas várias informações do ambiente que serão fornecidas como entradas para a rede neuronal. Desta forma deverá assumir que a arquitectura da rede neuronal é a seguinte:

- **18 inputs** <sup>1</sup>:

1. Distância do agente à bola;
2. Ângulo entre os vectores de posição do agente e da bola;
3. Distância do agente à sua própria baliza;
4. Ângulo entre os vectores de posição do agente e da sua própria baliza;
5. Distância do agente à baliza adversária;
6. Ângulo entre os vectores de posição do agente e da baliza adversária;
7. Distância do agente ao adversário (no caso de não existir o valor é -1);
8. Ângulo entre os vectores de posição do agente ao adversário (no caso de não existir o valor é -1);
9. Distância da bola à baliza do agente;
10. Distância da bola à baliza do adversário;
11. Distância à parede mais próxima;
12. Ângulo entre os vectores de posição do agente e da parede mais próxima;
13. Velocidade instantânea do agente calculada a partir da posição no ponto actual e no imediatamente anterior;

---

<sup>1</sup>Poderá ainda usar inputs adicionais. Deverá no entanto validar as suas decisões com os docentes da disciplina.

14. Ângulo entre os vectores de posição actual do agente e a sua posição no instante anterior;
15. Velocidade instantânea da bola calculada a partir da posição no ponto actual e no imediatamente anterior;
16. Ângulo entre os vectores de posição actual da bola e a posição no instante anterior;
17. Velocidade instantânea do adversário calculada a partir da posição no ponto actual e no imediatamente anterior;
18. Ângulo entre os vectores de posição actual do adversário e a sua posição no instante anterior;

- **3 Neurónios na Camada Escondida**

1. Recomenda-se a utilização de 3 neurónios nesta camada. Este valor representa um compromisso em tempo de aprendizagem e funcionalidade. Note que quanto maior for o número de neurónios nesta camada, mais tempo será necessário para a aprendizagem. Poderá, no entanto, durante a fase de experimentação verificar a influência que este parâmetro tem nos resultados.

- **2 outputs:**

1. Intensidade de força no intervalo  $[-1.0, 1.0]$ .
2. Ângulo de aplicação da força no intervalo de  $[-1.0, 1.0]$  (convertido para  $-180^\circ$  a  $180^\circ$ ).

É importante referir ainda que todos os 18 inputs da Rede Neuronal estão normalizados no intervalo  $[0.0, 1.0]$ . Deverá garantir que, qualquer que seja a topologia da sua rede, tenha pelo menos as entradas e saídas acima referidas.

O presente trabalho prático encontra-se dividido em 2 metas distintas:

1. Meta 1 – Modelação e desenvolvimento do algoritmo genético.
2. Meta 2 – Experimentação e análise.

## **2.1 Meta 1 – Modelação e desenvolvimento do algoritmo genético**

A representação escolhida, os operadores genéticos, o mecanismo de selecção, a atribuição de aptidão (*fitness*), são componentes essenciais para o bom funcionamento de um algoritmo genético. Desta forma, a etapa de **modelação**

desempenha um papel fundamental no sucesso do seu algoritmo. Nesta primeira fase do trabalho, deverá fixar a topologia da rede neural. Por exemplo, poderá considerar a seguinte arquitectura:

1. **Camada de Entrada** - 18 Nós
2. **Camada Escondida** - 3 Nós
3. **Camada de Saída** - 2 Nós

Nesta fase deverá desenvolver as funcionalidades básicas do Algoritmo Genético. Os requisitos mínimos são os seguintes:

**Recombinação** Deve desenvolver um operador de recombinação de **1-ponto**. Poderá, caso ache adequado, desenvolver e testar outros operadores.

**Mutação** Desenvolver o operador de mutação guassiana descrito pelo Algoritmo 1. Caso ache adequado poderá desenvolver outros operadores de mutação.

**Seleccção** Deve implementar o mecanismo de **seleccção por torneio**, descrito pelo Algoritmo 2. **Lembre-se que é um problema de maximização.**

**Parametrização** Deve ser possível alterar facilmente os parâmetros do algoritmo evolucionário – p.ex. especificar a probabilidade de mutação por gene, probabilidade de recombinação, tamanho do torneio, etc – através do inspector do Unity, sem que seja necessário alterar o código.

**Aptidão** A aptidão de um indivíduo está relacionada com a capacidade do agente marcar mais golos que o adversário para ganhar. É por isso um **componente essencial** ao sucesso do algoritmo. Devem ser exploradas várias funções de aptidão, tendo em conta o conjunto de informações do ambiente. Note que soluções em que os agentes sofram golos deverão ser penalizadas. Para conseguir desenvolver a sua função de aptidão terá ao dispôr, no fim de cada simulação, as seguintes informações:

1. distanceToBall - Lista com as diferentes distâncias do agente à bola;
2. distanceToMyGoal - Lista com as diferentes distâncias do agente à sua baliza;
3. distanceToAdversaryGoal - Lista com as diferentes distâncias do agente à baliza adversária;

4. distanceToAdversary - Lista com as diferentes distâncias do agente ao adversário;
5. distancefromBallToMyGoal - Lista com as diferentes distâncias da bola à baliza do agente;
6. distancefromBallToAdversaryGoal - Lista com as diferentes distâncias da bola à baliza do adversário;
7. distanceToClosestWall - Lista com as diferentes distâncias do agente à parede mais próxima;
8. agentSpeed - Lista com os diferentes valores de velocidade instantânea do agente.
9. ballSpeed - Lista com os diferentes valores de velocidade instantânea da bola.
10. advSpeed - Lista com os diferentes valores de velocidade instantânea do adversário.
11. distanceTravelled - Total de distância percorrida;
12. hitTheBall - Número de vezes que o agente tocou na bola;
13. hitTheWall - Número de vezes que o agente tocou na parede;
14. GoalsOnMyGoal - Número de golos sofridos;
15. GoalsOnAdversaryGoal - Número de golos marcados.

Note que nos cenários em que o agente está sozinho, as informações sobre o adversário estarão vazias.

```

1 !
2 Function GaussianMutation(probability):
3   Mean  $\leftarrow$  0 ;
4   Stdev  $\leftarrow$  0.5 ;
5   i  $\leftarrow$  0 ;
6   for i < GenotypeSize do
7     if Random(0, 1) < probability then
8       | genotype[i]  $\leftarrow$  genotype[i] + NextGaussian(Mean, Stdev) ;
9     end
10    i  $\leftarrow$  i + 1 ;
11  end

```

**Algoritmo 1:** Pseudocódigo do Algoritmo de Mutação Gaussiana. A função NextGaussian já se encontra implementada, e deve ser chamada de forma como se encontra no pseudo-código. Os valores das variáveis Mean e Stdev podem ser ajustados.

```

1 !
2 Function TournamentSelection(population, tournamentSize):
3   best  $\leftarrow$  null ;
4   i  $\leftarrow$  0 ;
5   for i < tournamentSize do
6     ind = population[RandomInteger(0, PopulationSize - 1)] ;
7     /* Maximização */;
8     if (best == null) or ind.Fitness > best.Fitness then
9       | best  $\leftarrow$  ind.Clone() ;
10    end
11    i  $\leftarrow$  i + 1 ;
12  end
13  return best

```

**Algoritmo 2:** Pseudocódigo do Algoritmo de Selecção por Torneio. Este algoritmo permite seleccionar o melhor indivíduo num torneio de tournamentSize indivíduos.

Após implementadas, é vital **testar** as funcionalidades do algoritmo evolucionário por forma a garantir o seu bom funcionamento. Para isso faça uso do cenário onde apenas tem um controlador para evoluir, nomeadamente o cenário *Evolving-ControlTheBallToAdversaryGoal*. Aqui deverá conseguir desenvolver uma função de fitness que permita encontrar um controlador que permita controlar a bola, e marcar golos na baliza adversária. Após estabilizar as componentes do algoritmo genético devem testar as implementações no cenário de co-evolução onde teremos duas populações distintas a evoluir

controladores dentro do mesmo ambiente (e.g. Evolving-OnevsOne).

## 2.2 Meta 2 – Experimentação e análise

Nesta meta deve utilizar a aplicação desenvolvida para encontrar uma solução para os ambientes disponibilizados. Deve utilizar o código desenvolvido na meta anterior para evoluir controladores. Os controladores podem ser evoluídos usando diferentes funções de aptidão adotando assim comportamentos distintos. Deve criar diversas funções de aptidão utilizando a informação disponibilizada pelo ambiente. São disponibilizadas vários ambientes, cujo o nome indica o objectivo que o agente deve atingir. Após a evolução dos controladores, estes são gravados na pasta do projeto. Deve usar o cenário “MatchMaker” para validar e testar os controladores evoluídos. Deve testar o(s) agentes(s) nos cenários apresentados na Fig. 2.

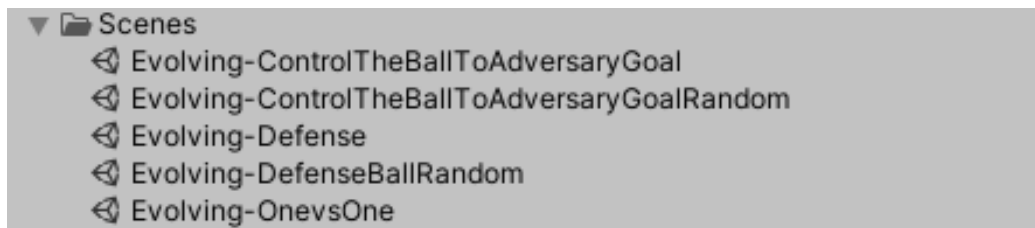


Figura 2: Cenários com as tarefas que o seu agente deve ser capaz de resolver.

Dada a natureza estocástica das abordagens evolucionárias não é possível tirar conclusões a partir de uma única execução. Para cada combinação de parâmetros deverá realizar, pelo menos, **5** repetições da experiência para que a comparação efectuada tenha significado estatístico. **Desta forma, é importante reservar o tempo adequado para esta meta.**

Deve conduzir um conjunto alargado de experiências considerando, de forma sistemática, diferentes combinações de parâmetros.

Não basta enumerar resultados experimentais, deve fazer uma análise dos mesmos procurando explicar as diferenças encontradas e os comportamentos apresentados.

## 3 Datas e Modo de Entrega

Os grupos têm uma dimensão máxima de 3 alunos. A defesa é obrigatória, bem como a presença de todos os elementos do grupo na mesma.



A entrega da meta 1 é opcional, chama-se no entanto a atenção dos alunos para a importância de concluir atempadamente esta meta. Para efeitos de nota apenas será considerada a entrega final e a defesa.

### 3.1 Meta 1 – Modelação e desenvolvimento

#### Material a entregar:

- Scripts onde implementaram e/ou alteraram código, que deve estar devidamente comentado.
- Um breve documento (max. 3 páginas), em formato pdf, com a seguinte informação:
  - Identificação dos elementos do grupo (Nomes, Números de Estudante, e-mails, Turma(s) Prática(s))
  - Informação pertinente relativamente a esta meta

#### Modo de Entrega:

Entrega electrónica através do Inforestudante.

**Data Limite: 2 de Maio de 2021**

### 3.2 Meta 2 – Experimentação e análise

Tal como indicado anteriormente, esta entrega será a única que tem um impacto directo na nota. O relatório deve conter informação relativa a **todo** o trabalho realizado. Ou seja, o trabalho realizado no âmbito das metas 1 e 2 deve ser **inteiramente descrito**, por forma a possibilitar a avaliação.

#### Material a entregar:

- Scripts onde implementaram e/ou alteraram código, que deve estar devidamente comentado.
- Um relatório (max. 20 páginas), em formato pdf, com a seguinte informação:
  - Identificação dos elementos do grupo (Nomes, Números de Estudante, e-mails, Turma(s) Prática(s))
  - Informação pertinente relativamente à globalidade do trabalho realizado

Num trabalho desta natureza o relatório assume um papel importante. Deve ter o cuidado de descrever detalhadamente todas as funcionalidades implementadas, dando particular destaque aos problemas e soluções encontradas. Deve ser fácil ao leitor compreender o que foi feito e ter por isso capacidade de adaptar / modificar o código.

Conforme pode depreender do enunciado, **experimentação** e **análise** são parte fundamental deste trabalho prático. Assim, deve descrever de forma sucinta mas detalhada as experiências realizadas, os resultados obtidos, analisar os resultados e extrair conclusões.

O relatório deve conter informação relevante tanto da perspectiva do utilizador como do programador. Não deve ultrapassar as 20 páginas, formato A4. Todas as opções tomadas deverão ser devidamente justificadas e explicadas.

### **Modo de Entrega:**

Entrega electrónica através do Inforestudante.

**Data Limite: 16 de Maio de 2021**

## **4 Bibliografia**

- **Inteligência Artificial: Fundamentos e Aplicações**  
*Ernesto Costa, Anabela Simões*
- **Artificial Intelligence: A Modern Approach**  
*Stuart Russel, Peter Norvig*

## Checklist

Nesta secção fornece-se uma breve checklist que visa minimizar as probabilidades de lacunas graves no trabalho e relatório. Importa no entanto salientar que esta checklist **não substitui** a validação das opções tomadas, que deverá ser efectuada preferencialmente durante as aulas Práticas Laboratoriais, **nem garante** a obtenção de uma classificação final positiva.

- Implementação:
  - Implementou operador(es) de recombinação?
  - Implementou operador(es) de mutação?
  - Os operadores de variação preservam a validade dos indivíduos?
  - Implementou o mecanismo de selecção por torneio?
  - Teve em conta o facto de ser um problema de maximização?
  - É possível parametrizar o algoritmo sem recorrer a alterações de código?
- Experimentação:
  - As experiências realizadas têm em conta a natureza estocástica da abordagem (i.e. efectua várias repetições da experiência usando os mesmos parâmetros e seeds aleatórias distintas)?
  - Tendo em conta as opções implementadas, os resultados realizadas permitem indicar:
    - \* A melhor Rede Neuronal para o problema?
    - \* O melhor operador de mutação e respectiva taxa?
    - \* O melhor operador de recombinação e respectiva taxa?
  - Tendo as experiências realizadas consegui evoluir agentes que permitam:
    - Defender?
    - Controlar a Bola?
    - Marcar Golos?
    - Jogar contra outros agentes?
  - As respostas às perguntas anteriores constam do relatório? Estão devidamente justificadas e suportadas em resultados experimentais?
  - No relatório, descreveu o que foi feito na meta 1 e meta 2?