



Robolocode – Controla o teu robô e ganha a corrida!  
Manual do professor



Co-funded by the  
Erasmus+ Programme  
of the European Union

## Índice

Introdução .....	3
O propósito deste manual .....	3
Grupo alvo / Audiência.....	3
Learning objectives.....	3
Robolocode – O jogo .....	4
Bem-vindos.....	4
O que precisa de saber (como professor) - para se preparar e ajudar os seus alunos.....	4
A garagem:.....	4
A programação:.....	4
As pistas: .....	5
Obstáculos.....	5
Módulos robóticos.....	6
Funções disponíveis no jogo.....	8
Exemplo de programação, para iniciar algumas funções.....	8
Iniciando a corrida.....	10
Papel do professor.....	10
Primeira implementação.....	10
Possíveis abordagens didáticas.....	11
Abordagem individual.....	11
Abordagem de grupo.....	11
Abordagem de sala de aula.....	11

# Introdução

Este manual destina-se a professores, para os ajudar na utilização do jogo Robolocode como uma ferramenta de aprendizagem com os seus alunos. O jogo pode ser jogado em dispositivos móveis, onde os alunos irão aprender os básicos do pensamento computacional e programação.

O objetivo do jogo é que o jogador (aluno) ensine o robô (através de programação simples) como mover-se na pista, evitar obstáculos, atrasar os oponentes e chegar à meta em primeiro lugar.

O Robolocode é um jogo *mobile* no qual os jogadores montam um robô e desenvolvem o programa para que o mesmo corra numa pista em competição com outros robôs. A parte relacionada com a programação inicia-se como código simples para guiar o robô ao longo da pista por forma a garantir que o mesmo chega à meta. No entanto, vai progressivamente ficando mais complicado quando mais robôs competem e atrapalham-se uns aos outros, especialmente quando estão equipados com dispositivos ofensivos e defensivos que podem influenciar a corrida.

O objetivo do jogo é ganhar corridas, subir de nível e melhorar o robô por forma a continuar a correr. No entanto, a chave para ganhar corridas consiste em escrever bom código!

## O propósito deste manual

Este manual foi criado para professores que estão a utilizar o Robolocode como parte das suas atividades de aprendizagem, ensinando aos alunos o básico do pensamento computacional, programando comandos simples, criando ciclos, definindo condições e mais.

Este manual providencia informação teórica e prática sobre o uso do jogo Robolocode com o propósito de aprendizagem, olhando para a implementação de estratégias, papel do professor e dando exemplos sobre como o jogo pode ser utilizado como uma ferramenta de aprendizagem para o público alvo.

É aconselhado que, como professor, leia o manual cuidadosamente antes de planear a utilização do jogo na sua atividade de aprendizagem com os seus alunos.

## Grupo alvo / Audiência

Este jogo destina-se a alunos do ensino básico e início do ensino secundário.

## Objetivos de aprendizagem

- Pensamento computacional
- Linguagem de programação Python (não é um objetivo de aprendizagem no jogo mas sim como parte das atividades de aprendizagem)

# Robolocode – O jogo

## Bem-vindos

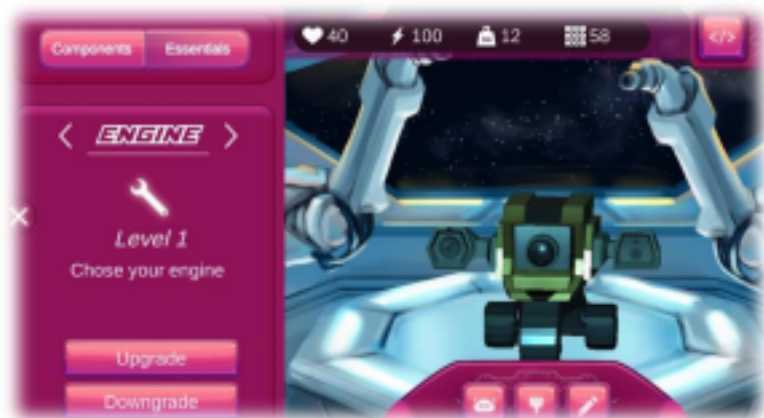
Caro professor, enquanto lê este manual, queremos que explore e veja como utilizar e implementar o jogo como parte do seu ambiente de aprendizagem.

## O que precisa de saber (como professor) - para se preparar e ajudar os seus alunos

O jogo está dividido em três partes principais: A **Garagem**, o ambiente de **Programação** e a **Corrida** (modo *sandbox*).

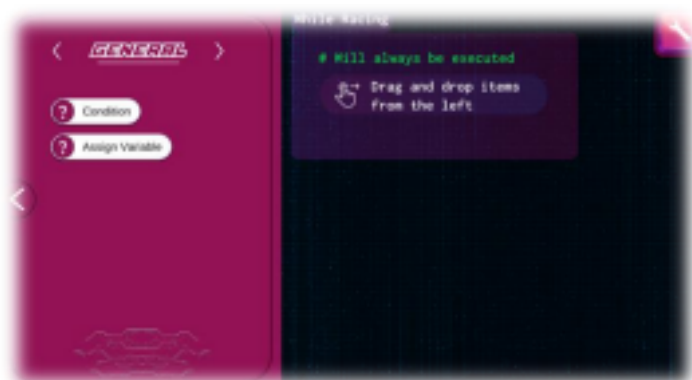
### A Garagem:

Nesta parte do jogo, irá conceber o seu robô e adicionar módulos ao mesmo. Os módulos que podem ser adicionados ao robô são **Baterias**, **Motores** e **Chassis**. As propriedades dos diferentes módulos são descritas mais abaixo..



### A Programação:

Nesta parte do jogo, o ambiente de programação permite-lhe gerar o código que irá ajudar o seu robô a manobrar na pista, evitando obstáculos, competir com os oponentes e mais.



## As Pistas:



Nesta parte do jogo pode entrar nas pistas e competir contra robôs IA ou contra outros jogadores (no modo multijogador). À medida que progride e compete com outros, conseguirá adicionar novos componentes ao seu robô, tornando-o mais rápido, forte, competitivo e mais.

Durante a corrida, o robô irá enfrentar diferentes tipos de obstáculos. Estes são descritos abaixo:

## Obstáculos

Ao correr, o robô irá enfrentar vários obstáculos. Como dono do robô, tu - o jogador, necessita de adicionar código ao robô para garantir que ele consegue evitar os obstáculos na pista. A tabela abaixo mostra a lista de obstáculos e a forma como estes se “comportam”.

### Barril

Pode conter óleo (aleatório). Quando atingido, causa movimentos aleatórios numa faixa. Se o robô estiver em chamas e atingir um barril com óleo, irá causar uma explosão e deixar óleo para trás, que será consumido pelo fogo deixando a faixa limpa.

### Chamas

Ativação intermitente. Dispara diretamente para cima e causa danos instantâneos. O robô ficará em chamas durante alguns segundos.

### Óleo

Pode ser incendiado por um robô em chamas. Irá arder por aproximadamente 5 segundos. Ao arder irá causar o mesmo efeito que as chamas. Quando as chamas terminarem, o óleo também irá desaparecer. Numa curva fará com que o robô deslize para uma faixa externa (a não ser que o robô tenha correntes ou omni rodas). Se o robô estiver na faixa mais afastada irá deslizar e embater na parede, sofrendo danos. Aleatoriamente o óleo também pode aumentar a velocidade do robô se este estiver numa reta.

### Espigões

Causam danos às rodas e têm uma ativação intermitente. Um robô com correntes destrói os espigões. Um robô com omni rodas irá abrandar mas não recebe danos ou destrói os espigões.

## Módulos robóticos

O robô pode ser atualizado adicionando diferentes tipos de módulos ao mesmo. Alguns módulos irão aumentar as capacidades do robô, outros darão capacidades ofensivas e outros capacidades defensivas.

Os diferentes módulos serão desbloqueados à medida que o jogador progride no jogo, competindo com outros jogadores (no modo multijogador).

Os módulos podem ser divididos em categorias: **Movimento**, **Atualizações**, **Sensores** e **Robot Ferramentas** (**O**fensivas e **D**efensivas).

Quando o jogador adiciona um módulo ao robô, o módulo necessita de ser programado para que o robô perceba como utilizá-lo. Se o código não ativar os módulos, o módulo não será ativado. Os diferentes módulos são descritos abaixo.

### *Movimento (Velocidade)*

O robô está equipado com rodas desde o início. Estas podem ser atualizadas com correntes ou para Omni Rodas. Estes diferentes tipos de módulos de movimento têm diferentes capacidades, com vantagens e desvantagens. Estas estão listadas abaixo:

- Rodas: Movimento mais rápido. Afetadas por todos os obstáculos
- Correntes: Mais lentas que as rodas. Não afetadas pelos espigões
- Omni rodas: Mais lentas que as rodas. Não afetadas pelo óleo

### *Propriedades do robô*

O robô pode ser atualizado com vários módulos. O módulo **Chassis** irá aumentar a proteção do robô, o módulo **Motor** irá aumentar a potência do motor - permitindo ao robô deslocar-se mais rapidamente, o módulo **Bateria** está integrado com os módulos de ataque e defesa, aumentando o “tempo de vida” para a utilização destes diferentes módulos.

#### **Chassis**

1. Modo por omissão - Chassis normal
2. Modo estendido - Chassis reforçado
3. Modo Um murro só - Chassis pesado (mais peso)

#### **Motor**

1. Modo por omissão - Motor normal
2. Modo V8 - Motor mais poderoso
3. Modo foguete - Motor grande (mais peso)

#### **Bateria**

1. Modo por omissão - Bateria normal
2. Modo alcalinas - Bateria com mais capacidade
3. Modo monstro - Bateria maior (mais peso)

### *Sensores (categoria outros)*

O robô também pode ser atualizado com sensores para detectar obstáculos, calor e como os robôs dos oponentes estão equipados. Sensores de nível 1 pesam menos que sensores de nível 2 e sensores de nível 3 pesam mais que os sensores de nível 2. Nota, o peso irá afetar a velocidade do robô! Uma lista de sensores é exibida abaixo.

### Sensor de obstáculos

- Nível 1 - Apenas vê obstáculos diretamente em frente
- Nível 2 - Detecta o tipo de obstáculo
- Nível 3 - Consegue verificar as faixas em busca de obstáculos



### Sensor de calor

- Nível 1 - Detecta se um obstáculo está em chamas, diretamente me frente
- Nível 2 - Consegue prever se irá ocorrer um fogo na faixa
- Nível 3 - Consegue verificar as faixas em busca de obstáculos em chamas e prevê-los



### Sensor de oponentes (*propriedade*)

- Nível 1 - Consegue detetar a posição
- Nível 2 - Consegue detetar armamento e tipo de movimento (rodas, correntes e omni rodas)
- Nível 3 - Consegue detectar propriedades (energia, força do chassis) Pode ser instalado em qualquer lado, frente ou atrás. Consegue ver em 360º



Nota! Adicionar este sensor ao robô adiciona novas possibilidades às condições. Os níveis são cumulativos, por exemplo, o nível 3 tem as capacidades do nível 2 e 1.

### Ferramentas robóticas / engenhocas

À semelhança do sensor de oponentes os níveis são cumulativos, por exemplo, o nível 3 tem as capacidades do nível 2 e 1.

### Martelo

Apenas pode ser utilizado no lado onde está localizado

1. Martelo de plástico - Danos normais
2. Martelo de madeira - Danos extra, mais peso
3. Martelo de aço - Mais peso e mais eficiente energeticamente

### Espigões

Apenas afetam a faixa atual e os robôs que estão atrás

1. Espigões pequenos - Danos normais
2. Espigões médios - Danos extra
3. Espigões tijolo - Extremamente dolorosos

### Raio trator

Afeta o robô diretamente em frente. Requer uma certa quantidade de energia

1. Raio básico - Abrandando o oponente em frente
2. Raio estendido - Aumenta o efeito do abrandamento
3. Raio fantástico - Mais eficiente energeticamente

## Escudo

O módulo escudo protege o robô de outros robôs mas não de obstáculos.

1. Escudo de madeira - Diminui os danos causados pelo martelo
2. Escudo de aço - Diminui os danos provocados por embates
3. Escudo de titânio - Diminui o abrandamento provocado pelo raio trator

## Funções disponíveis no jogo

O jogo tem a sua própria sintaxe para programação. A lista abaixo mostra que tipos de funções estão disponíveis no ambiente de programação do jogo:

### Geral

- Condition
- Assign Variable
- Enable autodrive

### Movimento

- Start Moving
- Stop Moving
- Move left
- Move right

### Componentes

- Hammer Left
- Hammer Right
- Tractor Beam
- Drop Spikes
- Shield Left
- Shield Right

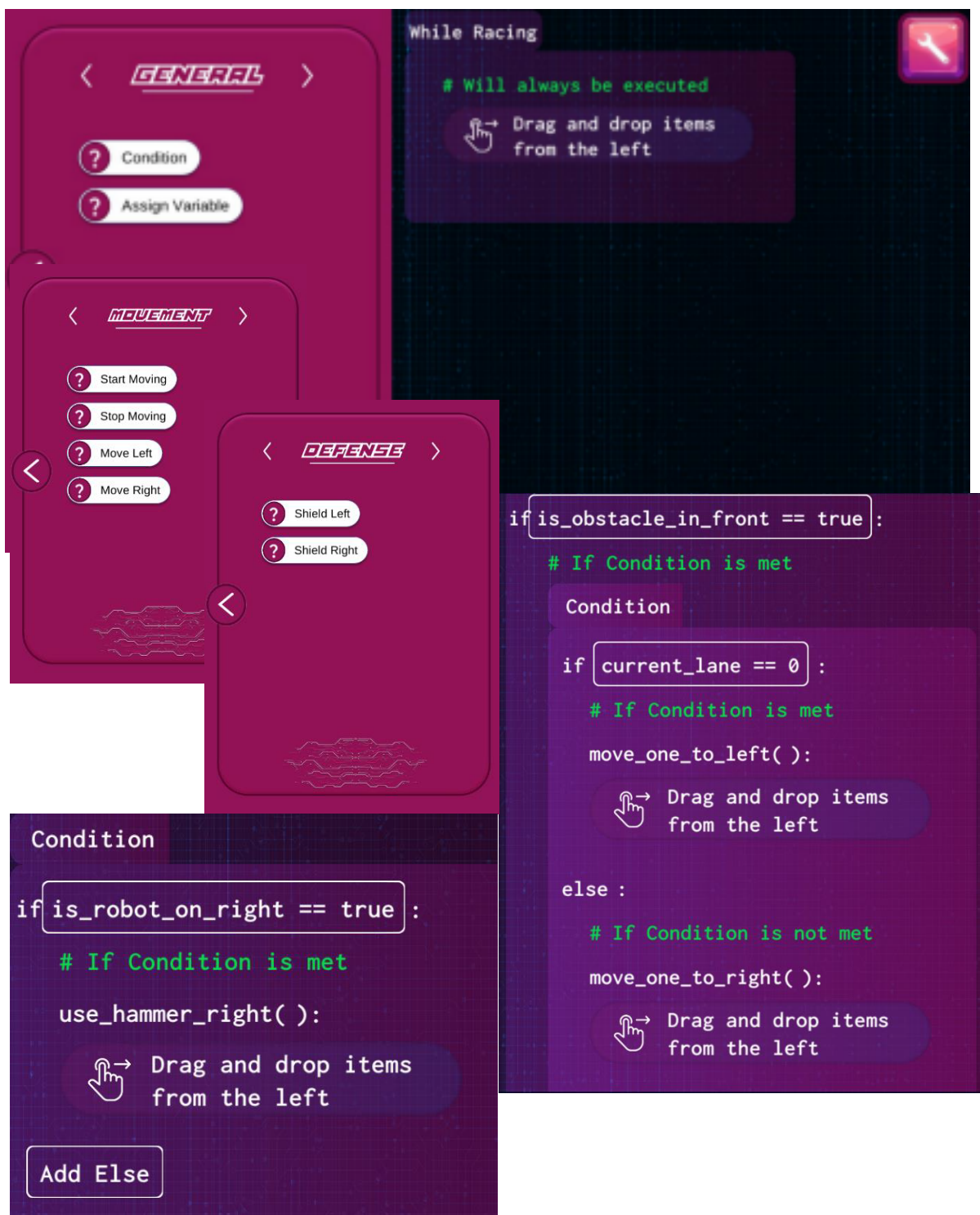
### Sensores *(funções disponíveis na edição de uma condição)*

- is\_robot\_on\_left
- is\_robot\_on\_right
- is\_robot\_in front
- is\_robot\_behind
- is\_obstacle\_on\_left
- is\_obstacle\_on\_right
- is\_obstacle\_in front
- is\_obstacle\_behind
- is\_driving
- current\_lane

## Exemplo de programação, para iniciar algumas funções

Conforme listado no capítulo anterior, existem várias funções que podem ser usadas e adicionadas como código ao robô. O presente capítulo irá exibir mais alguns exemplos de código que ajudaram a iniciar a utilização de componentes robóticos, por forma a evitar obstáculos, utilizar o martelo e mais.





# Iniciando a corrida

(no modo multijogador)

## Papel do professor

Ao implementar o jogo na sala de aula com os seus alunos, deverá garantir que todos têm tempo suficiente para explorar o jogo e familiarizar-se com os diferentes menus e como navegar pelos mesmos, como construir o seu próprio robô e como adicionar código ao mesmo, antes de testar o robô numa corrida. A tabela abaixo providencia uma forma de o professor planear a primeira implementação do jogo com os seus alunos

### Primeira implementação

Passo	Tarefa	Papel do professor	Duração
1	Providenciar uma introdução (apresentação) da ferramenta	Apresentar a ferramenta, exibir capturas de ecrã e mostrar rapidamente como funciona a interface, mostrar código simples para desviar-se de obstáculos e como podem começar a competir na pista	15 minutos
2	Deixar os alunos explorar a interface de utilizador (UI)	Estar disponível como facilitador enquanto os alunos exploram o jogo por eles mesmos, sem tarefas específicas	15 minutos
3	Sumário curto das experiências	O professor pergunta aos alunos sobre o jogo, o que fizeram até ao momento e ajuda a resumir a experiência (como uma sessão plenária)	15 minutos
Pausa			
4	Preparar para a corrida. Deixar os alunos programarem os seus robôs e prepararem-se para a corrida	Apoiar os alunos e ajudá-los a organizar o seu código	15 minutos
5	Organizar o evento multijogador	Ajudar os alunos a aderirem a corridas multijogador e iniciar as corridas	10 minutos
6	Efetuar uma reflexão na sala de aula onde os alunos apresentam como correu e discutem como pode o código ser melhorado.	O professor guia o processo e ajuda o grupo a aprender uns com os outros.	20 minutos

## **Possíveis abordagens didáticas**

Para implementar o jogo em atividades de aprendizagem, existem várias abordagens que o professor pode utilizar. Neste manual, sugerimos 3 abordagens, descritas abaixo.

### **Abordagem individual**

Deixar os alunos explorar o jogo por eles próprios, com o professor como facilitador, onde o professor irá apoiar cada indivíduo a programar e a iniciar as corridas multijogador, efetuando sessões plenárias regularmente.

### **Abordagem de grupo**

Organizar os alunos em grupos ( de três a quatro) e deixá-los explorar a aplicação em conjunto. O professor deverá ajudar os grupos a comunicar no grupo e de seguida ajudar as equipas/grupos a iniciar as corridas multijogador competindo uns com os outros a nível de grupo. Esta atividade deverá ser seguida por uma reflexão de grupo e uma plenária.

### **Abordagem de sala de aula**

Para esta abordagem, os professores deverão estabelecer uma colaboração com outras escolas e desta forma permitir que grupos e turmas de diferentes escolas possam competir uns com os outros, criando uma “liga escolar”, onde turmas (de diferentes escolas) entram em competição umas com as outras. Eventos desta natureza deverão ser seguidos de sessões de reflexão.