

# A PENSAR EM **tic**

7.º/8.º ANOS

Carlos Alberto Nunes  
Fernanda Ledesma

Tecnologias  
de Informação  
e Comunicação

Manual de  
Aplicações  
Alternativas

# 8



# Vais utilizar a linguagem...

## Kodu Game Lab

### 1 KODU

É um dos personagens que se podem mover pelo cenário, podendo ser programados para interagirem com outros personagens, com objetos ou com o próprio cenário.

### 2 OBJETOS

São objetos imóveis que podem interagir com os personagens.

### 3 TERRENO

Local onde decorre a ação, podendo ser facilmente editado pelo programador.

### 4 FERRAMENTAS

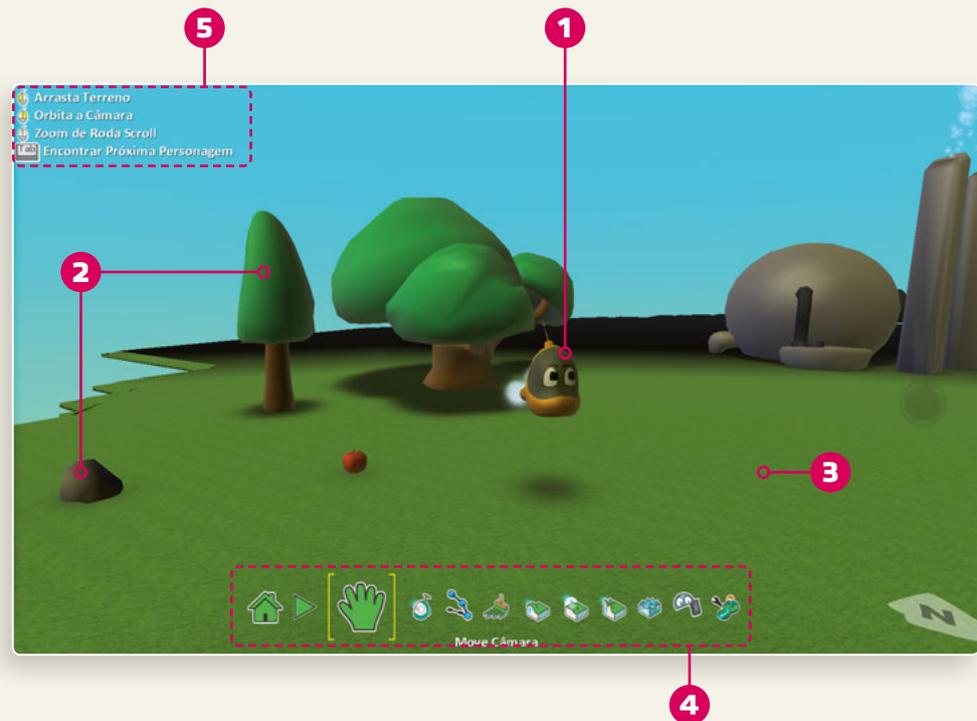
Permitem editar todos os elementos da aplicação.

### 5 ÁREA DE AJUDA

Apresenta funções que podem ser utilizadas com a ferramenta selecionada.

#### 20 AULA DIGITAL

Que mais posso saber acerca do ambiente de trabalho do Kodu Game Lab?





# Como começar?

## 1 Criar um novo projeto e editar terreno de jogo

### PASSOS A PERCORRER:

- 1 Após abrir a aplicação, **escolhe** a opção **Novo Mundo**;
- 2 **Altera** as definições do mundo;
- 3 **Aumenta** a área do jogo;
- 4 **Manipula** o relevo do terreno;
- 5 **Adiciona e manipula** a água.

**NOTA:** Podes mover a câmara pelo terreno escolhendo  ou sair em .



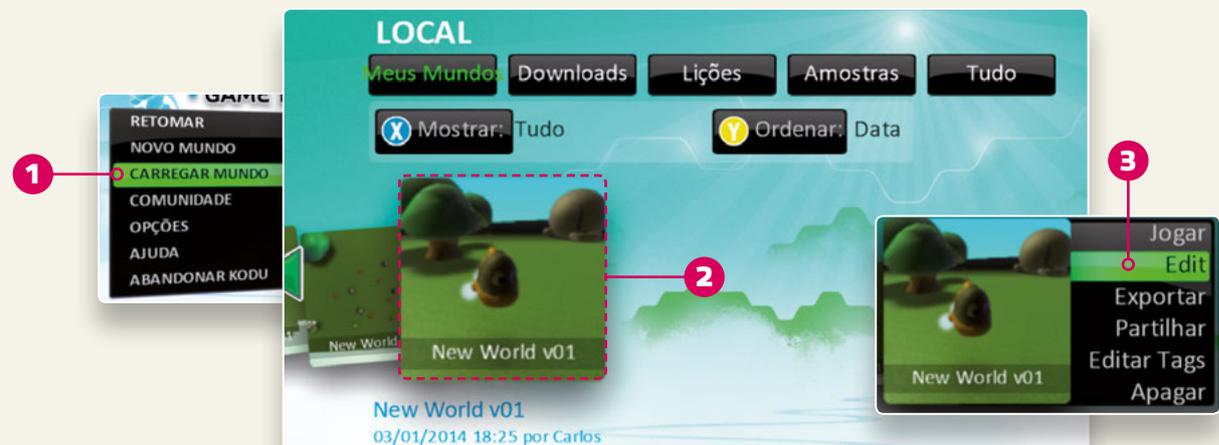
## 2 Abrir um projeto

### PASSOS A PERCORRER:

- 1 Após abrir a aplicação, **escolhe** a opção **Carregar Mundo**;
- 2 **Escolhe** o mundo a abrir;
- 3 **Escolhe** a opção pretendida (editar ou jogar).

**NOTA:** Ao clicar em exportar, podes criar um ficheiro que pode ser carregado noutra computador.

Ao executar um ficheiro exportado, esse jogo surgirá na categoria "Downloads".



### 3 Adicionar novos personagens e objetos

**PASSOS A PERCORRER:**

- 1 Após iniciares ou carregares um mundo, **clica** em  ;
- 2 **Clica** no local onde o objeto ou personagem deverá ser posicionado;
- 3 **Escolhe** o personagem ou objeto na lista.

**NOTA:** Podes alterar a cor de um personagem ou objeto, mantendo o apontador do rato em cima dele e pressionando as setas direita e esquerda.



### 4 Programar personagens e objetos

**PASSOS A PERCORRER:**

- 1 Após iniciares ou carregares um mundo, **clica** em  ;
- 2 **Move** o rato para cima do objeto ou personagem pretendido e **clica** no botão direito do rato;
- 3 **Clica** em programar;
- 4 **Define** a condição (“Quando”) para iniciar uma determinada ação;
- 5 **Define** a ação (“Faz”) a realizar pelo objeto ou personagem programado.





# Como funciona a programação de um personagem ou objeto?

Tanto os personagens (*bots*) como objetos podem ser programados de uma forma simples e intuitiva. Para os programar, necessitamos de definir uma condição (“Quando”) e uma ação (“Faz”), sendo que as opções disponibilizadas podem variar de acordo com o objeto ou personagem programado. Em baixo temos as opções iniciais do “Quando” e do “Faz” para o personagem Kodu.



1. Condições do personagem Kodu.



2. Ações do personagem Kodu.

**NOTA:** As opções que não têm um quadrado branco, contêm um grupo de opções relacionadas com uma temática.

## ESCOLHA DA CONDIÇÃO

Ao escolher uma condição poderá ser possível especificá-la melhor, através de um segundo nível de condições (a quantidade de níveis varia consoante as condições).

Em baixo temos um exemplo da utilização da condição **Mouse** (Rato), que foi especificada até ao 4.º nível (Quando clicar com o botão esquerdo numa maçã amarela).



3. Exemplo de condição (“Quando”).

## ESCOLHA DA AÇÃO

Tal como nas condições, também é possível especificar melhor uma ação através de vários níveis.

Em baixo temos um exemplo de uma ação **Move** (movimentar), aplicada à condição “Clicar numa maçã amarela”, que foi especificada até ao 6.º nível (Mover em círculos rapidamente à volta da maçã, “muito” próximo dessa maçã e pela esquerda).

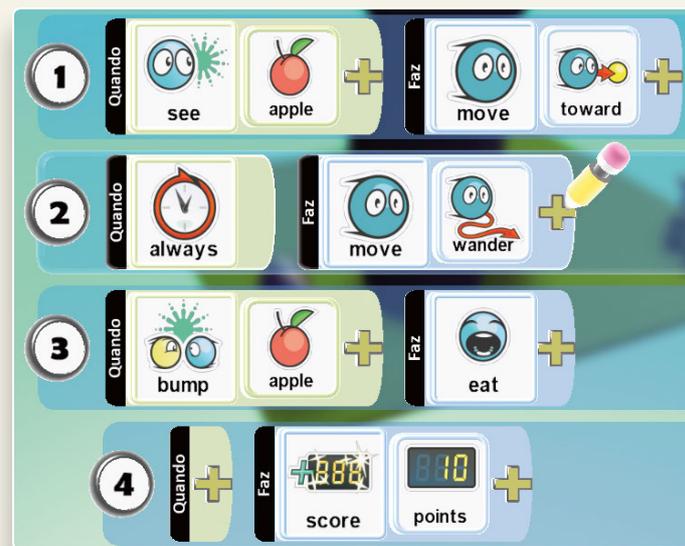


4. Exemplo de ação (“Faz”).

# Exemplo de programação de um personagem

Em baixo podemos ver o exemplo de um personagem programado. Se dessemos as instruções ao personagem utilizando uma linguagem normal, seria algo como:

- 1** Quando vês uma maçã, **move-te** na direção dela.
- 2** (Para sempre) **Move-te** sem destino.
- 3** Quando **chocas** com uma maçã, **come-a**.
- 4** Quando **chocas** com uma maçã, **adiciona 10 pontos** à tua pontuação.



5. Exemplo de programação de um personagem.

Neste exemplo podemos constatar algumas regras importantes da programação no Kodu, nomeadamente:

- Podem ser utilizadas várias condições e ações para um mesmo personagem;
- Cada opção do “Quando” e do “Faz” pode ser especificada em vários níveis. O símbolo + significa que podemos especificar melhor a condição (Por exemplo, na linha 1 poderíamos ter “Quando vês uma maçã vermelha” ou algo completamente distinto como “Quando não vês uma maçã amarela”);
- As condições têm uma prioridade de verificação e execução, sendo as condições mais em cima as que têm maior prioridade (se não fosse assim, as linhas 1 e 2 entrariam em conflito);
- Podem existir ações sem condição (como podemos ver na linha 2). Até poderia ser retirada a condição *Always* (sempre), que o efeito continuaria a ser o mesmo;
- A mesma condição pode dar origem a duas ações distintas (a indentação da linha 4 significa que a sua ação depende da condição da linha 3).



# Algumas combinações de condições

Como constatamos nos exemplos anteriores as condições têm um leque alargado de possibilidades de escolha. Deixamos aqui alguns exemplos de condições que podem ser utilizadas (apenas até ao 2.º nível), bem como a sua explicação.

## TECLADO



Quando é pressionada uma das **setas**



Quando é pressionado a tecla **"A"**

## RATO



Quando é pressionado o **botão direito** em cima de algo (especificado no 3.º nível)



Quando o ponteiro do rato **passa por cima** de algo (especificado no 3.º nível)

## TEMPORIZADOR



Quando passarem **60 segundos**



Quando passar um **número aleatório** de segundos (entre 0 e um número definido no 3.º nível)

## PONTUAÇÃO



Quando a pontuação obtida for **superior** a algo (especificado no 3.º nível)



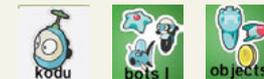
Quando a pontuação do **quadro de pontuação** vermelho for algo (especificado no 3.º nível)

Em várias situações irão surgir condições que se referem a objetos e personagens (além do próprio personagem programado).

Alguns exemplos deste tipo de condições são *Bump* (choca), *See* (vê), *Got* (obtém) ou *Shot hit* (acerta disparo).



Nestas situações, surgirão opções como **Kodu**, **Bots I** ou **Boots II** (personagens além do Kodu) e **Objects** (objetos inanimados).



Alguns exemplos de condições que podemos ter nestas situações são:

- Quando **choca** com um **Kodu**;
- Quando **vê** um **avião**;
- Quando **obtém** uma **maçã**;
- Quando **acerta disparo** numa **fábrica**.

# Algumas combinações de ações

Tal como existem diversas combinações de condições, também existem muitas combinações de ações. Deixamos aqui alguns exemplos de ações que podem ser utilizadas (apenas até ao 2.º nível), bem como a sua explicação.

## MOVIMENTO



Move-te num **caminho** (definido pelo programador)



Move-te em **frente**

## DISPARO



Dispara um **missil**



Dispara para **cima** algo (especificado no 3.º nível)

## VIRAR



Vira para a **esquerda**



Vira para **norte**

## PONTUAÇÃO



Adiciona **20 pontos** ao quadro de pontuação



Adiciona algo (especificado no 3.º nível) ao **quadro de pontuação azul**

## EXPRESSÕES



Mostra-te **magado** (vê estrelas)



Mostra-te **zangado**

## SALTO



Salta **alto**



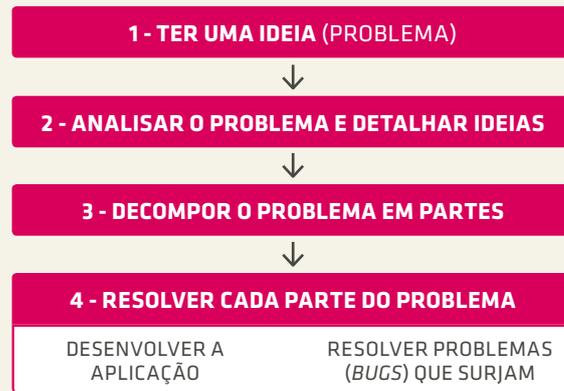
Salta **baixo**



# O desenvolvimento do meu primeiro projeto no Kodu

Para criar um projeto interessante no Kodu, apenas necessitamos de usar a nossa imaginação e engenho. A parte mais difícil da programação é ter uma boa ideia.

Para criar a nossa aplicação no Kodu, os passos essenciais que devemos seguir são:



Vamos então criar uma aplicação, seguindo a sugestão de passos referida anteriormente:

## 1 Ideia

Um jogo em que o personagem (Kodu branco) tem de apanhar maçãs.

## 2 Detalhes da ideia

- Existem dois tipos de maçãs (vermelhas e amarelas);
- O jogador é perseguido por outro personagem (Kodu preto);
- Existem obstáculos a evitar (pedras);
- O jogo deve ser visto de cima;

- O jogador tem 60 segundos para jogar;
- O jogador move o Kodu branco com o teclado;
- Quando apanhada, a maçã é comida;
- Deve ser feita uma contagem dos pontos obtidos pelas maçãs apanhadas (as maçãs amarelas valem mais pontos);
- São descontados pontos quando o Kodu preto toca no Kodu branco;
- A pontuação final deve refletir as maçãs apanhadas, o tempo utilizado e as penalizações por toque do adversário;
- O jogador perde quando o tempo termina ou o jogador bate numa pedra;
- O jogador ganha quando não há mais maçãs para apanhar.

## 3 Decompor o problema em partes

- Preparar os personagens e objetos;
- Fixar câmara no topo;
- Inicializar o temporizador;
- Kodu branco (jogador) a mover-se;
- Kodu preto (adversário) a mover-se;
- Penalizar jogador quando é apanhado pelo adversário;
- Contabilizar pontos ao apanhar maçãs;
- Atualizar o temporizador e descontar 1 ponto por cada segundo perdido;
- Ganhar o jogo quando acabam as maçãs;
- Perder o jogo quando termina o tempo ou o jogador bate numa pedra.

### 20 AULA DIGITAL

Como criar esta aplicação?

# Como fazer?

## 4 Resolver cada parte do problema

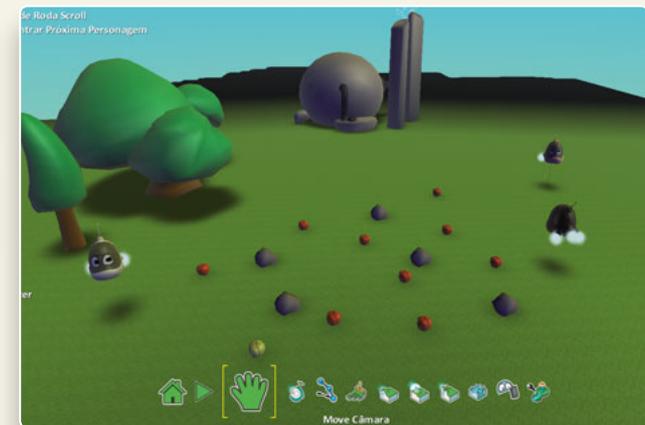
### A - Preparar os personagens e objetos

#### PERSONAGENS NECESSÁRIOS:

- **Kodu branco** – Controlado pelo jogador;
- **Kodu preto** – Controlado pelo computador;
- **Kodu vermelho** – Personagem que fará a contagem do tempo.

#### OBJETOS NECESSÁRIOS:

- **Maçã vermelha** – Irá valer 10 pontos;
- **Maçã amarela** – Irá valer 50 pontos;
- **Pedra** – Obstáculo que pode fazer perder o jogo.

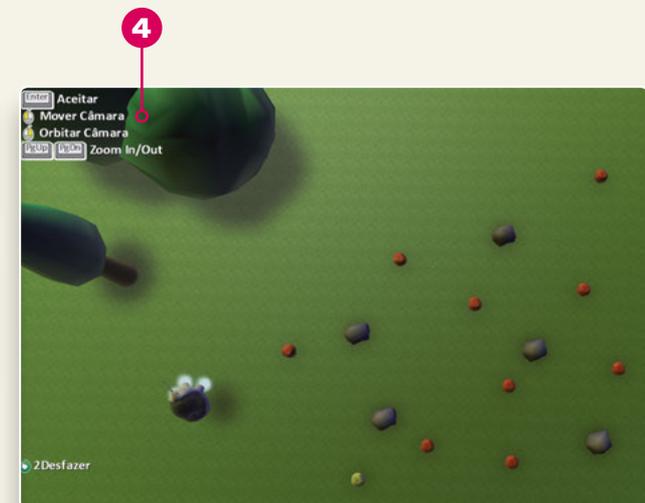


### B - Fixar câmara no topo

#### PASSOS NECESSÁRIOS:

- 1 Na barra das ferramentas, **accede** às definições do mundo;
- 2 **Procura** as opções do modo de câmara e **escolhe** a opção de câmara fixa;
- 3 **Clica** em **Definir câmara** (tecla X);
- 4 **Ajusta** a posição da câmara e **pressiona** a tecla **Enter**;

**NOTA:** Para ajustar a câmara, podes consultar a área de ajuda.





## C - Inicializar o temporizador

### O QUE DEVEMOS TER EM ATENÇÃO?

- Esta ação será feita uma única vez (*once*) e sem condições;
- O quadro de pontuação preto servirá de temporizador;
- Serão adicionados 50 + 10 pontos (serão contabilizados como 60 pontos).

### PERSONAGEM PROGRAMADO:



Kodu vermelho (Juiz)  
Inicializa (uma vez) o marcador preto, com 60 pontos.

## D - Kodu branco a mover-se

### O QUE DEVEMOS TER EM ATENÇÃO?

- O Kodu será controlado através das teclas de direção;
- Não é necessário especificar as direções individualmente.

### PERSONAGEM PROGRAMADO:



Kodu branco (Jogador)  
Quando são usadas as teclas de direção do teclado, move-te.

## E - Kodu preto a mover-se

### O QUE DEVEMOS TER EM ATENÇÃO?

- O Kodu preto deverá perseguir constantemente o Kodu branco.

### PERSONAGEM PROGRAMADO:



Kodu  
preto  
(Advers.)

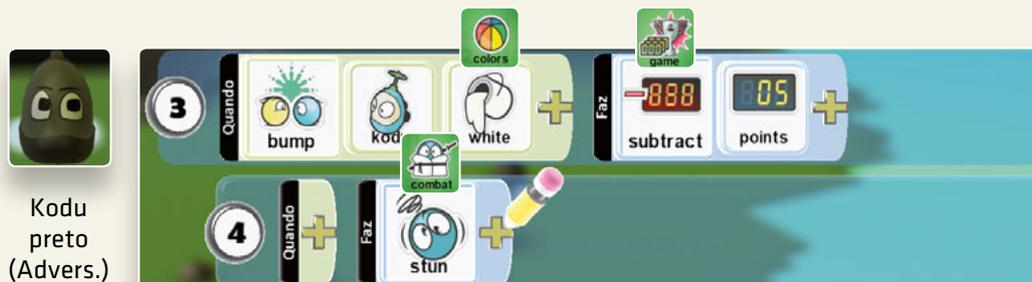
Quando vês o Kodu branco, move-te na sua direção.

## E - Penalizar jogador quando é apanhado pelo adversário

### O QUE DEVEMOS TER EM ATENÇÃO?

- O Kodu preto é o responsável por “pedir” a subtração dos pontos;
- Os pontos são descontados ao quadro de pontuação predefinido (vermelho);
- Quando os pontos são subtraídos, o Kodu branco deve ter a oportunidade de se afastar em segurança. Para isso, o Kodu preto fica atordoadado momentaneamente.

### PERSONAGEM PROGRAMADO:



Kodu  
preto  
(Advers.)

Quando chocas com um Kodu branco, subtrai 5 pontos (e ficas atordoadado).



## F - Contabilizar pontos ao apanhar maçãs

### O QUE DEVEMOS TER EM ATENÇÃO?

- O Kodu deverá comer todas as maçãs em que tocar (independentemente da sua cor).
- A maçã vermelha vale 10 pontos e a amarela 50;
- Os pontos são adicionados ao quadro de pontuação predefinido;
- Quando apanha uma maçã amarela, o Kodu pode ficar contente (não essencial para o desenrolar do jogo).

### PERSONAGEM PROGRAMADO:

The image shows the programming interface for the white Kodu character. It consists of four horizontal tracks, each representing a different behavior. The character's name 'Kodu branco (Jogador)' is listed on the left.

- Track 2:** Starts with a 'Quando' (When) block containing 'bump' and 'apple'. This is followed by a 'Faz' (Do) block containing 'eat'.
- Track 3:** Starts with a 'Quando' block containing 'bump' and 'apple'. This is followed by a 'Faz' block containing 'score' and 'points' with a value of 10. A 'colors' icon is shown above the 'red' block.
- Track 4:** Starts with a 'Quando' block containing 'bump' and 'apple'. This is followed by a 'Faz' block containing 'score' and 'points' with a value of 50. A 'colors' icon is shown above the 'yellow' block.
- Track 5:** Starts with a 'Quando' block containing a plus sign. This is followed by a 'Faz' block containing 'express' and 'happy'. An 'actions' icon is shown above the 'express' block.

Quando chocas com uma maçã, come-a.

Quando chocas com uma maçã vermelha, adiciona 10 pontos.

Quando chocas com uma maçã amarela, adiciona 50 pontos (e expressas felicidade).

## H - Atualizar o temporizador e descontar 1 ponto por cada segundo perdido

### O QUE DEVEMOS TER EM ATENÇÃO?

- O temporizador (quadro de pontuação preto) será atualizado a cada segundo;
- A pontuação (quadro de pontuação predefinido) será atualizada a cada segundo que passa (e não no final do jogo).

### PERSONAGEM PROGRAMADO:



Kodu  
vermelho  
(Juiz)

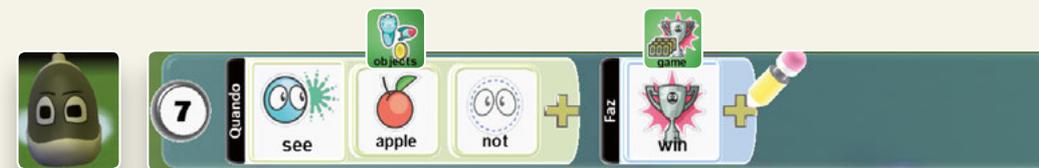
Quando o temporizador contar 1 segundo, subtrai ao quadro de pontuação preto 1 ponto.  
Quando o temporizador contar 1 segundo, subtrai 1 ponto.

## I - Ganhar o jogo quando acabam as maçãs

### O QUE DEVEMOS TER EM ATENÇÃO?

- É usada a negação à condição “quando vês uma maçã”.

### PERSONAGEM PROGRAMADO:



Kodu  
branco  
(jogador)

Quando não vês uma maçã, ganhas.



## J - Perder o jogo quando termina o tempo ou quando o jogador bate numa pedra

### O QUE DEVEMOS TER EM ATENÇÃO?

- O Kodu branco deteta a colisão com as pedras;
- O Kodu vermelho controla o temporizador.

### PERSONAGEM PROGRAMADO:



Kodu  
branco  
(jogador)

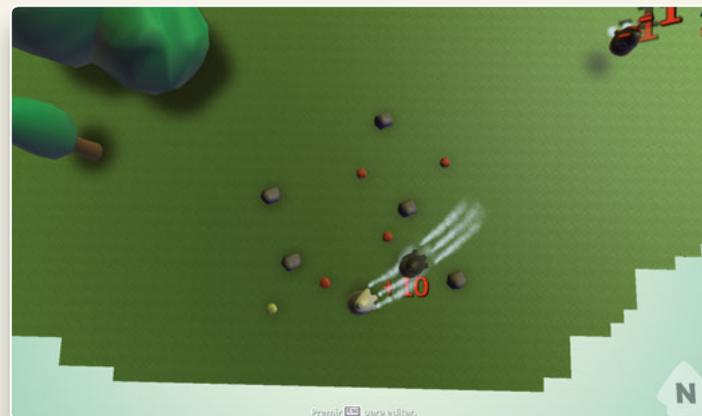
Quando chocas com uma pedra, o jogo acaba.



Kodu  
vermelho  
(juíz)

Quando o quadro de pontuação preto chega a um ponto, o jogo acaba.

### Aplicação completa



# Exercícios para praticares

Resolve os seguintes desafios, utilizando o teu engenho e imaginação:

## Desafio 1

Faz algumas melhorias e ajustes à aplicação utilizada nas páginas anteriores:

1. Faz com que o Kodu preto fale quando o jogo comece;
2. Faz com que o Kodu preto também apanhe maçãs, sendo o jogo ganho se o Kodu branco tiver apanhado mais maçãs;
3. Faz com que o Kodu preto possa ser controlado por um segundo jogador;
4. Adiciona itens ao terreno de jogo que, ao serem apanhados, fazem com que o jogador fique mais rápido (ou mais lento).

## Desafio 2

Cria um jogo de corridas, em que seja possível competir contra personagens controlados pelo computador ou por outros jogadores.

### 20 AULA DIGITAL

Como resolver estes exercícios?



# Neste subcapítulo, atingi os seguintes resultados:

<b>OBJETIVO:</b> Criar um produto original de forma colaborativa.	Já consigo	Ainda tenho algumas dificuldades	Ainda tenho muitas dificuldades
<b>1</b> Identificar um problema a resolver ou conceber um projeto. _____ Conceber um projeto desenvolvendo perspectivas interdisciplinares e contribuindo para a aplicação do conhecimento e pensamento computacional em outras áreas disciplinares. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>2</b> Analisar o problema e decompô-lo em partes. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>3</b> Explorar a utilização de variáveis, disponíveis no ambiente de programação. _____ Explorar a utilização de estruturas de decisão, disponíveis no ambiente de programação. _____ Explorar a utilização de estruturas de repetição, disponíveis no ambiente de programação. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>4</b> Implementar uma sequência lógica de resolução do problema, com base nos fundamentos associados à lógica da programação e utilizando componentes estruturais da programação. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>5</b> Efetuar a integração de conteúdos (texto, imagem, som e vídeo) com base nos objetivos estabelecidos no projeto. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>6</b> Respeitar os direitos de autor e a propriedade intelectual da informação utilizada. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>7</b> Analisar e refletir sobre a solução encontrada e a sua aplicabilidade. _____ Reformular a sequência lógica de resolução de problemas, de forma colaborativa. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>8</b> Partilhar o produto produzido na internet. _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

