

# JOGO DO LABIRINTO

## MAKERCODE ARCADE 2023

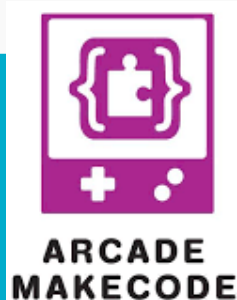
Produzido para o PROCURUMIM

Criado por Régis Caria

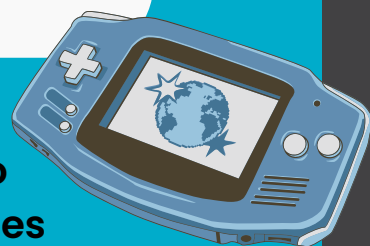
Apoio Adrio Hattori



@procurumim



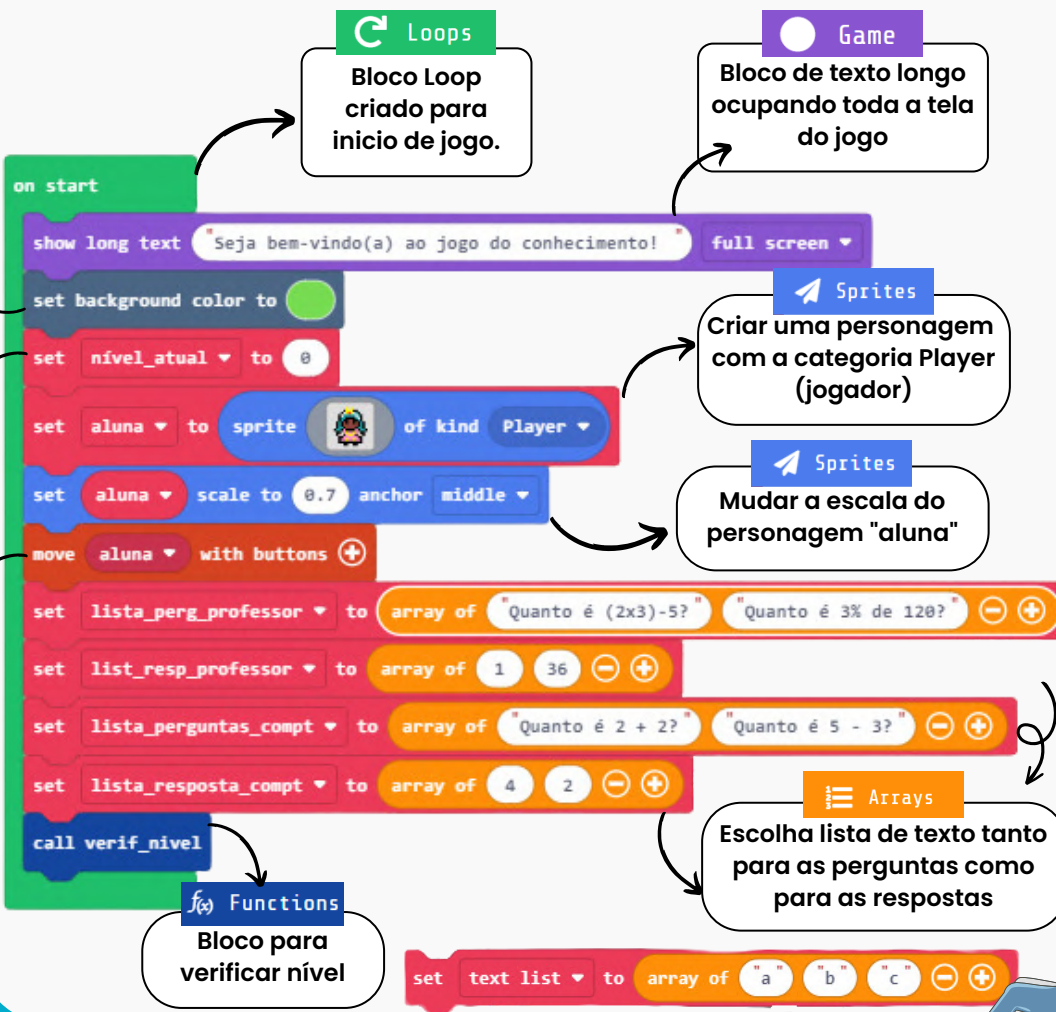
Conjunto  
de 9 Cartões





# TELA INICIAL

Você precisará criar variáveis com os seguintes nomes: "nível\_atual" para definir o nível do jogo, "lista\_perg\_professor" e "lista\_perguntas\_compt" para criar as perguntas e "lista\_resp\_professor" e "lista\_respostas\_compt" para criar a lista de respostas.



**Loops**  
Bloco Loop criado para início de jogo.

**Game**  
Bloco de texto longo ocupando toda a tela do jogo

**Scene**  
Bloco para definir a cor do plano de fundo de jogo

**Variables**  
Bloco para definir nível atual do jogo

**Sprites**  
Criar uma personagem com a categoria Player (jogador)

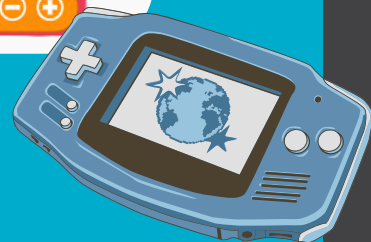
**Sprites**  
Mudar a escala do personagem "aluna"

**Controller**  
Bloco para movimentar a personagem

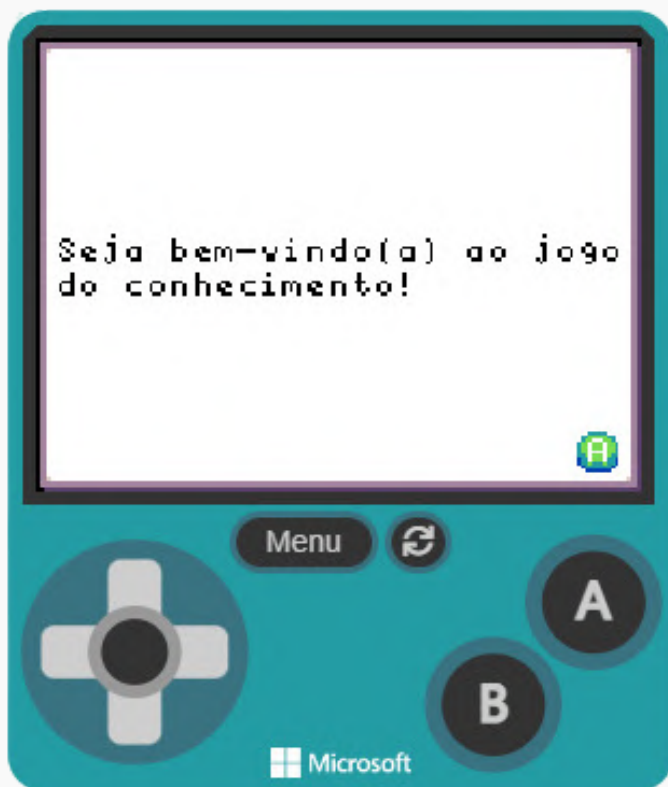
**Arrays**  
Escolha lista de texto tanto para as perguntas como para as respostas

**Functions**  
Bloco para verificar nível

```
on start
  show long text "Seja bem-vindo(a) ao jogo do conhecimento!" full screen
  set background color to #00FF00
  set nível_atual to 0
  set aluna to sprite of kind Player
  set aluna scale to 0.7 anchor middle
  move aluna with buttons
  set lista_perg_professor to array of "Quanto é (2x3)-5?" "Quanto é 3% de 120?"
  set list_resp_professor to array of 1 36
  set lista_perguntas_compt to array of "Quanto é 2 + 2?" "Quanto é 5 - 3?"
  set lista_resposta_compt to array of 4 2
  call verific_nivel
  set text list to array of "a" "b" "c"
```



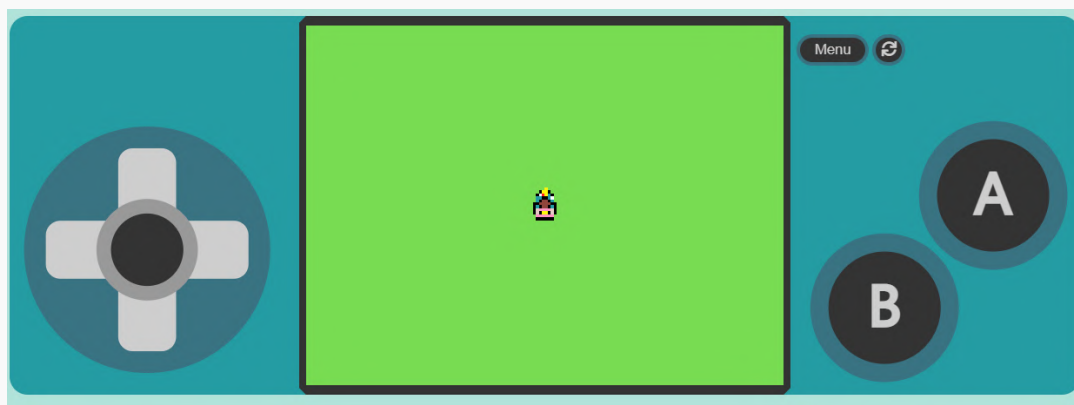
# TELA INICIAL



Finalizando os blocos anteriores você vai visualizar essa tela.

Nesta parte você pode criar a narrativa do jogo ou também criar regras do jogo que você está desenvolvendo.

Não esqueça que todo o jogo precisa de uma tela com as informações iniciais para auxiliar o jogador em sua jornada.

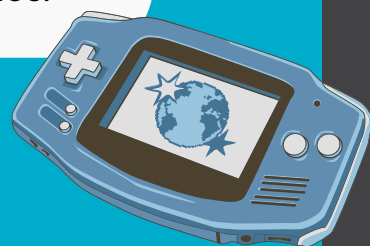


Na segunda tela do jogo você vai poder visualizar o cenário com o fundo verde e a personagem bem pequena na tela, vai poder movimentar a personagem para todos os lados.

Jogo  
do Labirinto

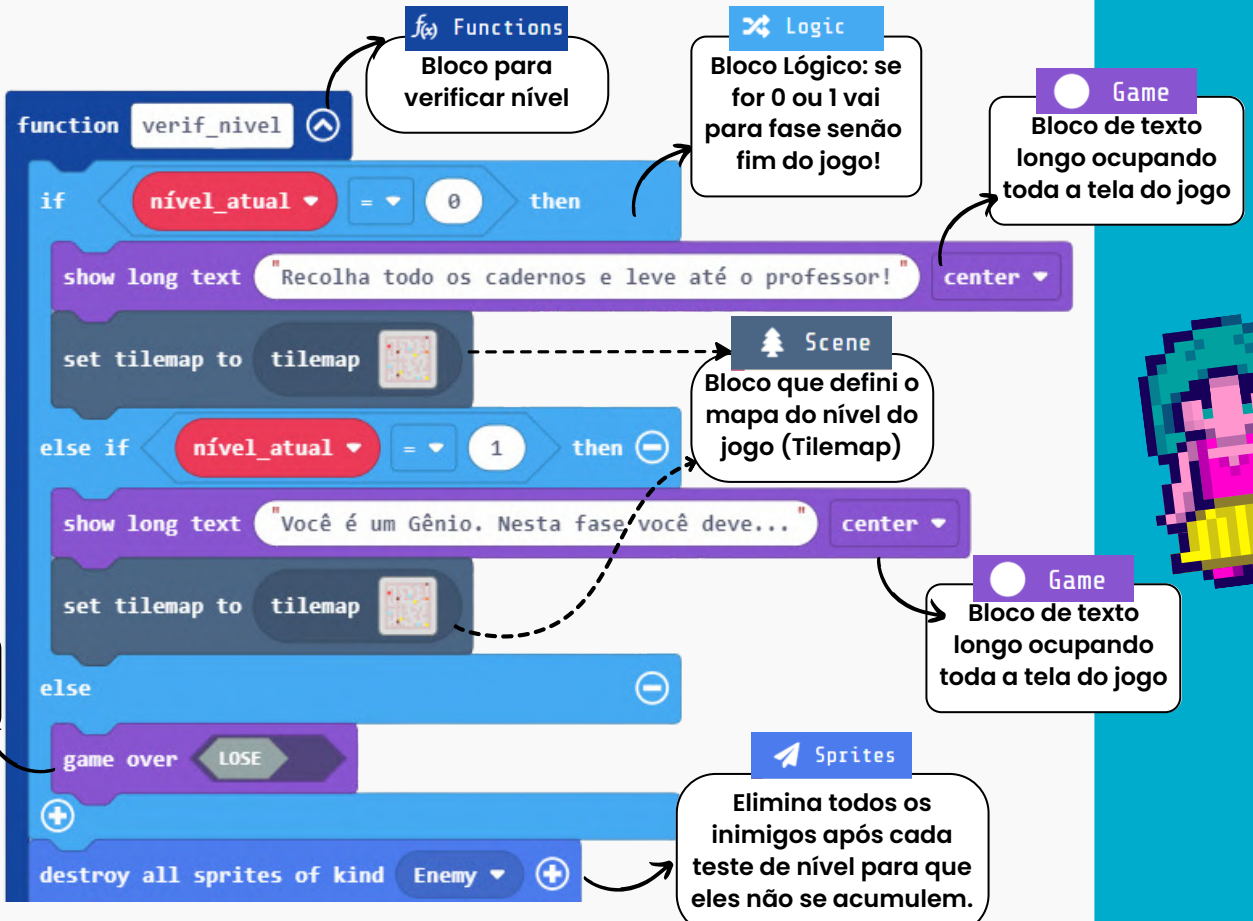
2

Arcade  
Makercode



# TILEMAP

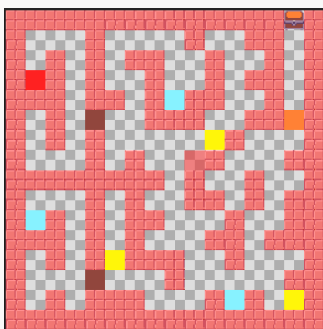
Nesta parte da programação será verificado o nível do jogo. Conforme a fase vai modificando também aumenta a dificuldade do labirinto por meio das "tilemaps" (mapa de blocos) onde serão construídos os blocos do labirinto, a posição dos elementos etc.



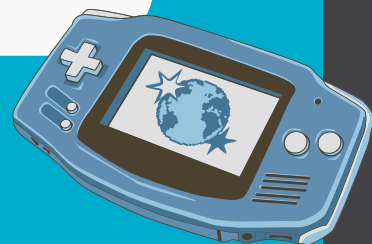
The image shows a Scratch script for a function named 'verif\_nivel'. The script is annotated with callouts explaining various blocks:

- Functions:** Bloco para verificar nível (referring to the function block).
- Logic:** Bloco Lógico: se for 0 ou 1 vai para fase senão fim do jogo! (referring to the 'if' block).
- Game:** Bloco de texto longo ocupando toda a tela do jogo (referring to the 'show long text' blocks).
- Scene:** Bloco que defini o mapa do nível do jogo (Tilemap) (referring to the 'set tilemap to' blocks).
- Game:** Bloco de texto longo ocupando toda a tela do jogo (referring to the second 'show long text' block).
- Sprites:** Elimina todos os inimigos após cada teste de nível para que eles não se acumulem. (referring to the 'destroy all sprites of kind' block).

```
function verif_nivel
  if nível_atual = 0 then
    show long text "Recolha todos os cadernos e leve até o professor!"
    set tilemap to tilemap
  else if nível_atual = 1 then
    show long text "Você é um Gênio. Nesta fase você deve..."
    set tilemap to tilemap
  else
    game over LOSE
  destroy all sprites of kind Enemy
```

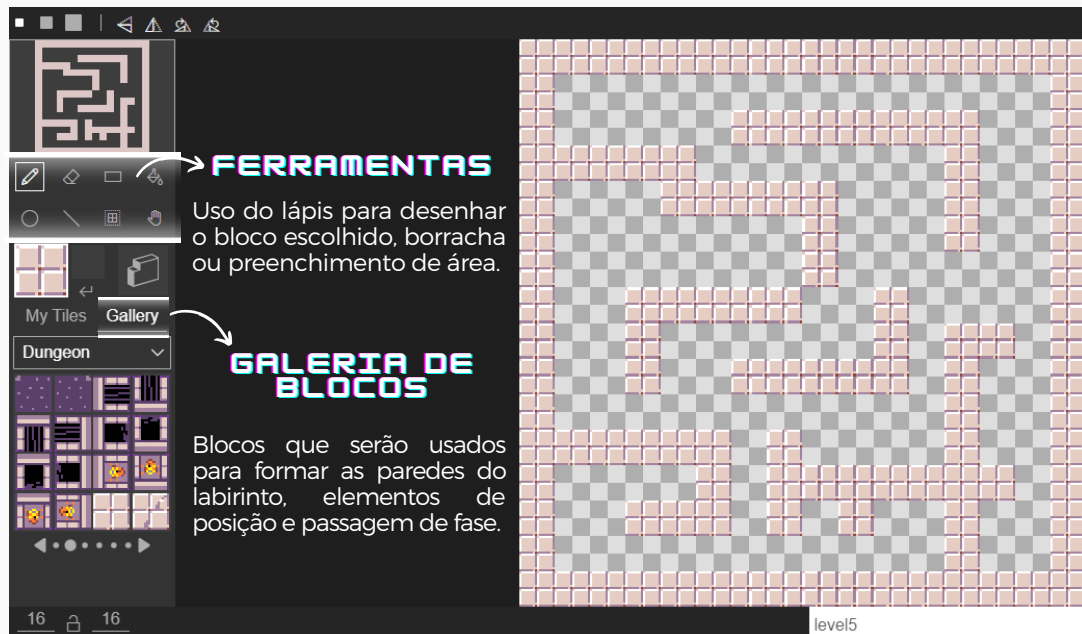


Tilemap funciona dividindo o cenário em uma grade de pequenos blocos chamados "tiles" os quais são organizados em um mapa que define a aparência e a disposição do cenário. Usaremos para criar o labirinto e definir a posição dos personagens.

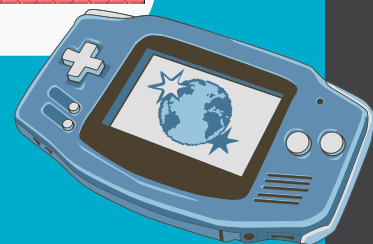
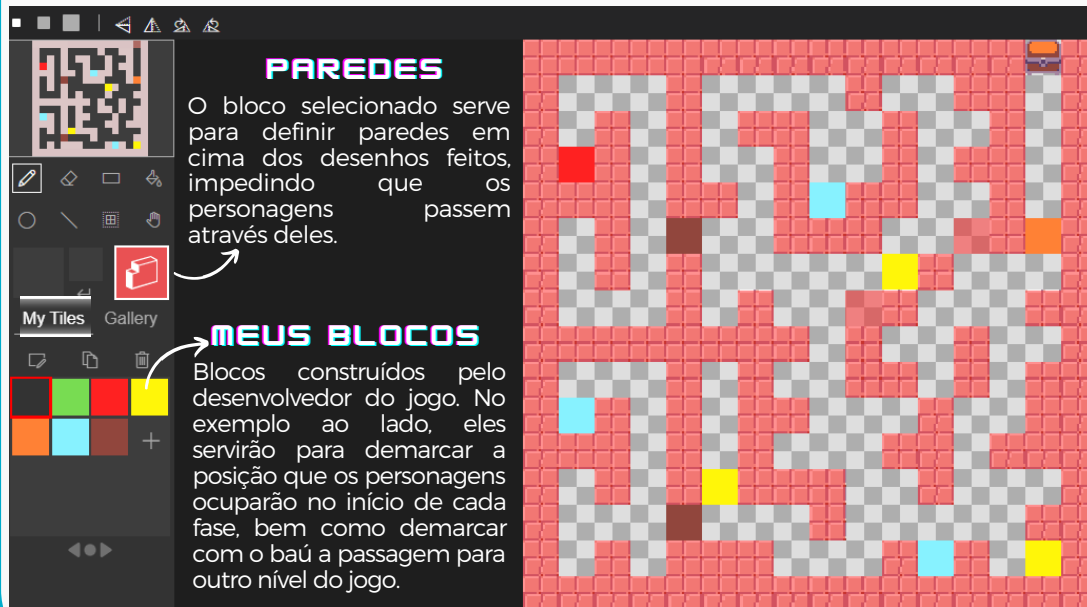


# TILEMAP

Para utilizar a tilemap você tem de escolher os blocos na Galeria (Gallery) e usar as ferramentas para preencher ou desenhar o cenário, como no exemplo abaixo:



## Criando as paredes da Tilemap



# MAPA E SPRITES

Continue essa programação abaixo da anterior. Com ela pretendemos posicionar os personagens nos seus respectivos locais de início de fase na Tilemap, bem como, diminuir a criação individual de inimigos.

```
for element value of array of all [red square] locations
do
  place aluna on top of value
  set [background] at value
camera follow sprite aluna
set life to 3
for element value2 of array of all [yellow square] locations
do
  set robo_ia to sprite [robot] of kind Enemy
  set robo_ia bounce on wall ON
  set robo_ia velocity to vx 30 vy 30
  place robo_ia on top of value2
  set [background] at value2
```

**Loops**

O que fazer com todos os locais (value) do quadrado vermelho

**Scene**

Põe a personagem aluna nesse quadrado vermelho

**Scene**

Determina fundo transparente para o lugar

**Loops**

O que fazer com todos os locais (value) do quadrado amarelo

**Info**

Determina 3 vidas iniciais para a aluna

**Sprites**

Cria o personagem robo\_ia, tipo inimigo, que irá percorrer o labirinto

**Sprites**

Determina a velocidade do robo\_ia

**Scene**

Põe o personagem robo\_ia em cima de todos os quadrados amarelos

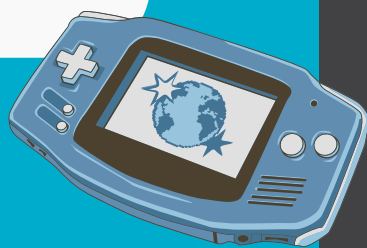
**Scene**

Determina fundo transparente aos quadrados amarelos

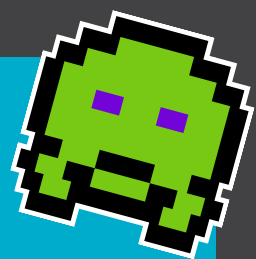


**Scene**  
A câmera segue o personagem aluna.

**Sprites**  
Determina que o robo\_ia ricocheteie na parede



# PERSONAGENS



**Loops**

O que fazer com todos os locais (value) do quadrado alaranjado

**Sprites**

Cria o personagem guardião professor que irá ajudar no labirinto

**Sprites**

Diminui o tamanho do personagem guardião

**Scene**

Determina fundo transparente aos quadrados amarelos

**Scene**

Põe o personagem em cima de todos os quadrados alaranjados

**Loops**

O que fazer com todos os locais (value) do quadrado Marrom

**Sprites**

Diminui o tamanho do personagem guardião

**Scene**

Determina fundo transparente aos quadrados amarelos

**Sprites**

Cria o personagem computador, servirá como caixa de perguntas e bloqueará o caminho o sprite responder de forma correta

**Scene**

Põe o personagem em cima de todos os quadrados marrons

**Sprites**

Cria o personagem Livros, servirá como item de coleta no jogo

**Scene**

Põe o personagem em cima de todos os quadrados marrom

**Loops**

O que fazer com todos os locais (value) do quadrado Azul Claro

**Scene**

Determina fundo transparente aos quadrados amarelos

```
for element value3 of array of all [orange] locations
do
  set guardião to sprite [guardião] of kind Professor
  set guardião scale to 0.8 anchor middle
  place guardião on top of value3
  set [checkered] at value3

for element value4 of array of all [brown] locations
do
  set computador to sprite [computador] of kind bloqueador
  set computador scale to 0.8 anchor middle
  place computador on top of value4
  set [checkered] at value4

for element value32 of array of all [light blue] locations
do
  set livros to sprite [livros] of kind leitura
  place livros on top of value32
  set [checkered] at value32
```





# COLISÃO

Este bloco detecta a colisão entre o jogador (aluna) com os cadernos (tipo leitura). Quando a aluna toca no caderno ele é apagado (destroy) e é acrescentado um ponto para a aluna.

```
on sprite of kind Player overlaps otherSprite of kind leitura  
  destroy otherSprite  
  change score by 1
```

**Sprites**  
Detecta a colisão entre objetos

**Sprites**  
Apaga o caderno (leitura)

**Info**  
Aumenta 1 ponto no Score

Este bloco detecta se a personagem aluna (Player) está sobre o local onde está a imagem do baú dentro da Tilemap. Se estiver sobrepondo o baú será determinado uma pausa de 1 seg., será determinado 1 para o nível atual e chamará a função de verificação de nível.

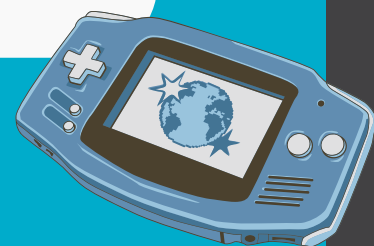
```
on sprite of kind Player overlaps [baú] at location  
  pause 1000 ms  
  change nível_atual by 1  
  call verif_nível
```

**Loops**  
Pausa 1 segundo

**Scene**  
Quando a aluna (Player) estiver em cima do baú

**Variables**  
Troca o valor da variável "nível\_atual" para 1

**Functions**  
Chama a função que verifica em que nível o jogo está



# COLISÃO

A primeira parte deste código verifica quando a personagem "aluna" está tocando o personagem "computador". Em seguida calcula a quantidade de itens dentro da lista de perguntas cadastradas para o computador.

**Sprites**  
Quando a personagem aluna toca o computador (bloqueador)

**Arrays**  
calcula a quantidade de itens dentro da "lista\_perguntas\_compt"

**Math**  
Sorteia um desses itens

**Game**  
Mostra a pergunta sorteada

**Variables**  
Troca o valor da variável "nível\_atual" para 1

**Game**  
Pergunta do usuário qual a resposta em formato de número

**Logic**  
Verifica se a resposta escrita pelo usuários (jogador) está correta (igual a da lista)

**Info**  
Acrescenta uma vida ao jogador

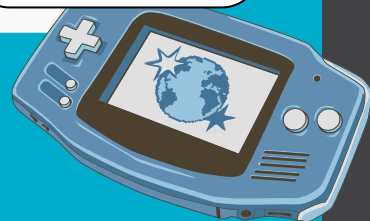
**Game**  
Mostra o texto junto ao personagem do professor

**Arrays**  
Remove a pergunta e a resposta selecionada para que elas não se repitam.

**Loops**  
Gera uma pausa de 1 segundo

**Sprites**  
Apaga o personagem professor para deixar a aluna ir até o baú.

```
on sprite of kind Player overlaps otherSprite of kind bloqueador
  set index_prof to pick random 0 to length of array lista_perguntas_compt - 1
  set pergunta to lista_perguntas_compt get value at index_prof
  set resposta to lista_resposta_compt get value at index_prof
  show long text pergunta center
  set resposta_usuario to ask for number "Qual o número?"
  if resposta_usuario = resposta then
    change life by 1
    splash "Parabéns! Pode passar..."
    pause 1000 ms
    lista_perguntas_compt remove value at index_prof
    lista_resposta_compt remove value at index_prof
    destroy otherSprite with smiles effect for 100 ms
  else
```



# TELA INICIAL

O próximo conjunto de blocos inicia com um ELSE, que é a continuação do Bloco IF. Este bloco é executado caso o usuário erre a resposta à pergunta (cálculo). Será retirada uma vida da personagem "aluna", aparecerá a mensagem ";(", a personagem "aluna" será movida para uma posição à esquerda na Tilemap e a pergunta e respostas sorteadas serão apagadas.

**Logic**  
O else será executado caso o usuário erre a resposta à pergunta (cálculo)

**Info**  
Retira uma vida da "aluna"

**Loops**  
Gera uma pausa de 1 segundo

**Sprites**  
Mostra a mensagem com cara triste

**Scene**  
Move a "aluna" para uma posição à esquerda na Tilemap

**Arrays**  
Remove a pergunta e a resposta selecionada para que elas não se repitam.

## DESAFIO



Terminado o tutorial, você irá programar a colisão (contato) da "aluna" com o "Professor". Caso faltem cadernos ele emite uma mensagem, senão, ele realizará perguntas mais difíceis que a dos "computadores". Será o desafio mais difícil antes de a "aluna" chegar ao baú.



Jogo  
do Labirinto

9

Arcade  
Makercode