



Aplicativo para introdução à lógica e programação

Esse aplicativo é um jogo (game) do tipo "Flappy Bird".

O jogador usa as teclas para cima e para baixo para fazer o dragão passar por entre as colunas. Cada vez que o dragão passa, o jogador ganha um ponto. O jogo se encerra quando o dragão bate em alguma coluna ou no chão.

Esse é um código-fonte aberto, desenvolvido com fins educacionais. Pode ser usado e alterado livremente.

Julho/2021



Etapa 1





Os três arquivos do programa:

- `index.html` → Arquivo que vai executar o programa e exibir no navegador
- `script.js` → Esse é o programa propriamente dito
- `style.css` → Arquivo que contém formatações (cores, tamanhos, espaços, etc.)

O “local” do jogo:

```
<canvas id="canvas" width="400" height="600"></canvas>
```



Cor do céu

background-color: #39addd;

style.css

RGB red green blue

#39addd

#ffffff

#ff0000

#00ff00

#0000ff

#C0C0C0

#000000

Hexadecimal

0	00	16	10	32	20
1	01	17	11	33	21
2	02	18	12	34	22
3	03	19	13	35	23
4	04	20	14	36	24
5	05	21	15	37	25
6	06	22	16	38	26
7	07	23	17	39	27
8	08	24	18	40	28
9	09	25	19	41	29
10	0a	26	1a	42	2a
11	0b	27	1b	43	2b
12	0c	28	1c	44	2c
13	0d	29	1d	45	2d
14	0e	30	1e	46	2e
15	0f	31	1f	47	2f



Etapa 2





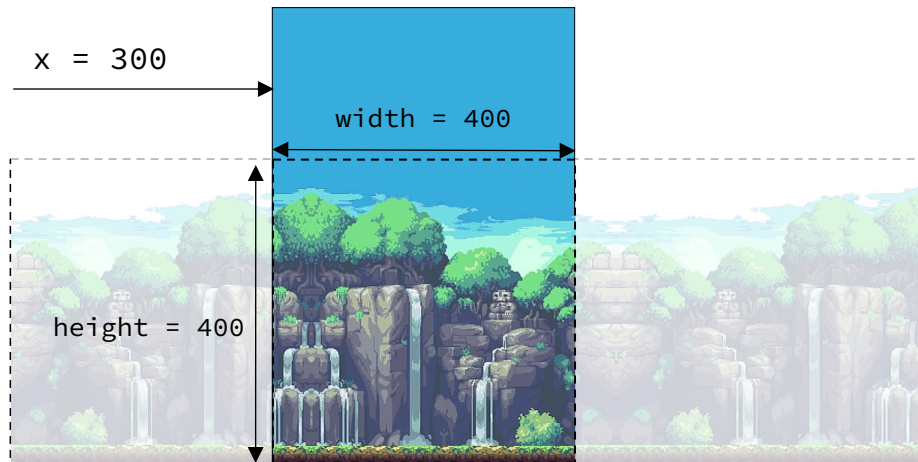
“Objeto”

```
var fundo1 = new Image();  
fundo1.src = 'img/fundo.png';
```

```
var fundo = {  
  image: fundo1,  
  x: 0,  
  y: 0,  
  width: 400,  
  height: 400,  
  top: 200, //  
  x_max: 800  
};
```



```
var fundo = {  
  image: fundo1,  
  x: 300,  
  y: 0,  
  width: 400,  
  height: 400,  
  top: 200, //  
  x_max: 800  
};
```



A imagem de fundo tem 1200 x 400 px.

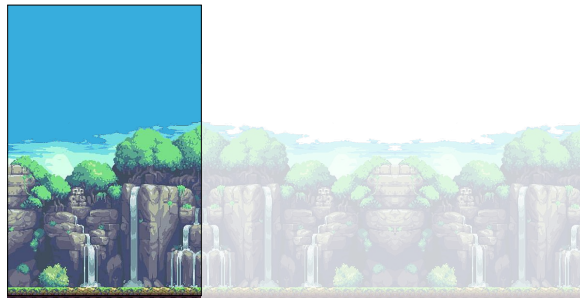
Mas apenas uma janela de 400 x 400 é exibida por vez.

A cada loop, o valor de x vai sendo aumentado, dando a sensação de movimento.

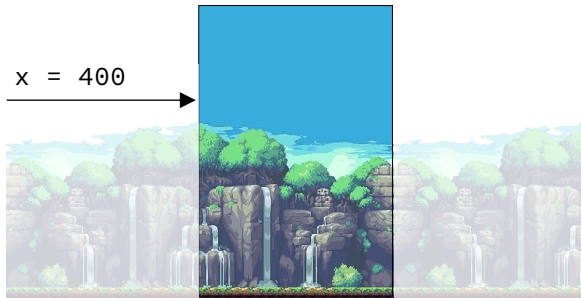
Quando x chega em 800, começa novamente.



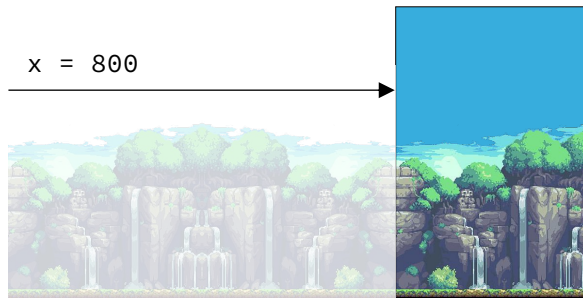
$x = 0$



$x = 400$



$x = 800$



Loop

`requestAnimationFrame`



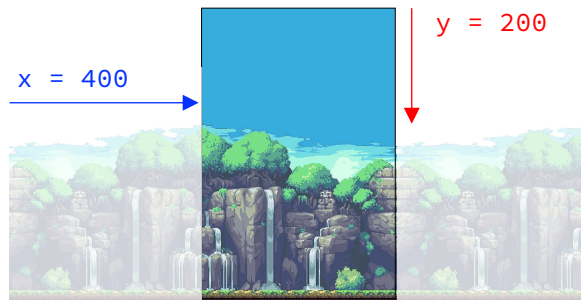
Sintaxe de recorte

```
context.drawImage(fundo, 400, 0, 400, 400, 0, 200, 400, 400);
```

Diagram illustrating the parameters for the `drawImage` method:

- Valores na imagem original** (Original image values):
 - `400`: x
 - `0`: y
 - `400`: width
 - `400`: height
- Valores no canvas** (Canvas values):
 - `0`: x
 - `200`: y
 - `400`: width
 - `400`: height

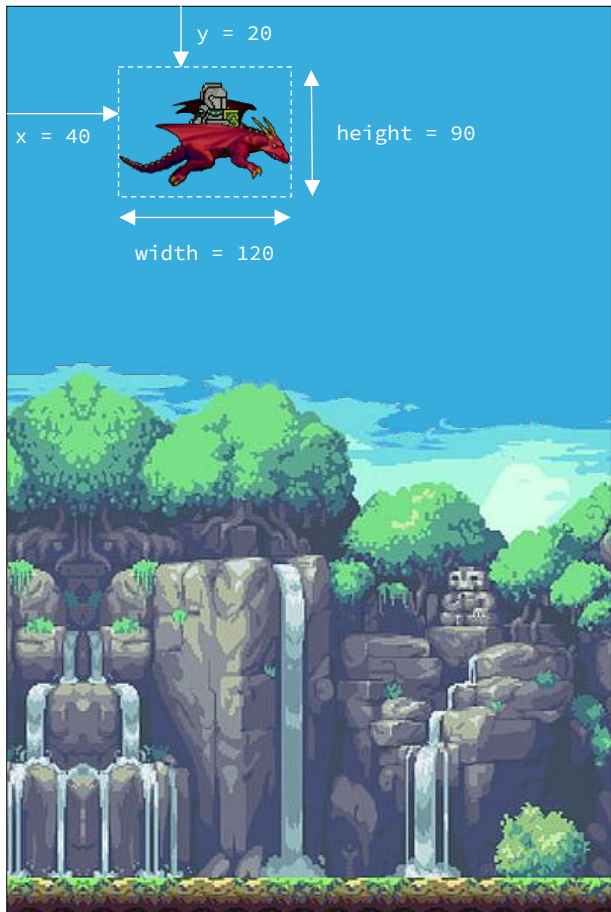
O "drawImage" é um método que permite aplicar uma imagem em um canvas.





Etapa 3





Sintaxe básica

```
context.drawImage(dragao, 40, 20, 120, 90);
```

Valores no
canvas

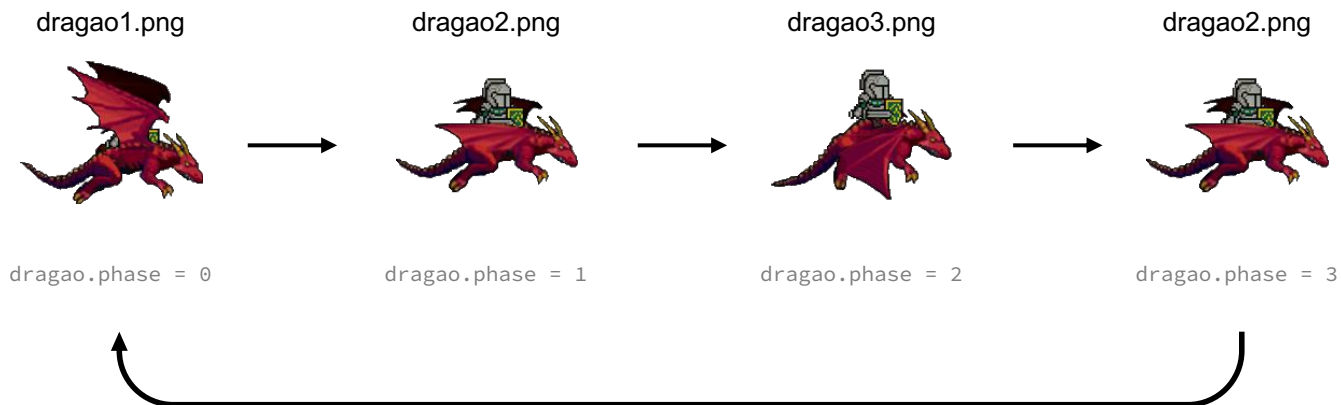
↓ ↓ ↓ ↓

x y width height



Etapa 4





Loop

`requestAnimationFrame`

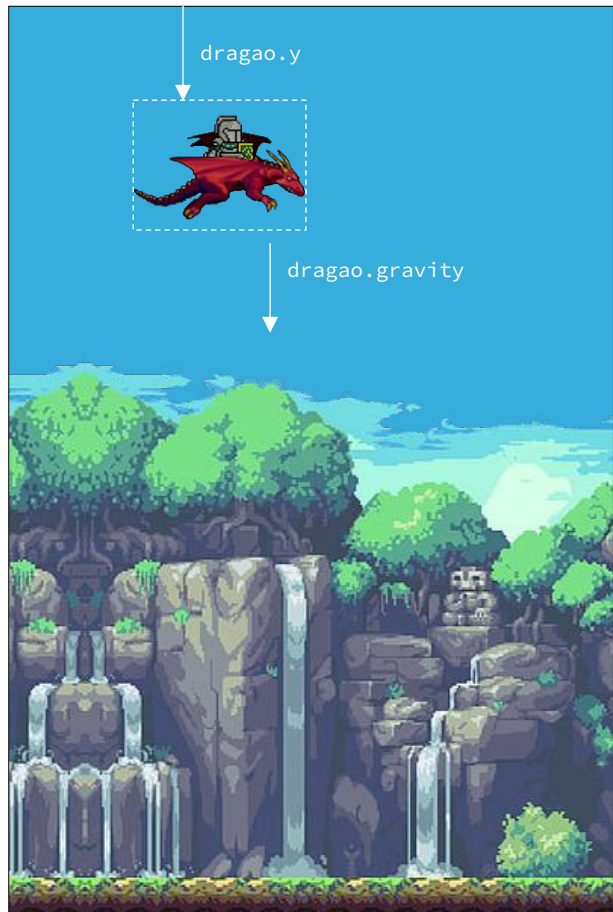
`dragao.flap = 3`

A propriedade (“flap”) define que o dragão vai bater as asas a cada “n” loops. No exemplo, vai bater as asas a cada 3 loops. Quanto maior o valor de “flap”, mais lentamente o dragão bate as asas.



Etapa 5





```
dragao.y += dragao.gravity;
```

Pode-se aumentar o valor de `dragao.gravity` para deixar o jogo mais difícil.



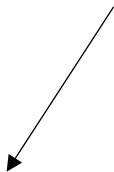
Etapa 6





Movimentação pelo teclado

```
document.addEventListener('keydown', ResponderTecla);
```



```
function ResponderTecla(tecla) {  
  if (tecla.keyCode == 38) {  
    dragao.y = dragao.y - 20;  
    if (dragao.y < 0) dragao.y = 0;  
  } else if (tecla.keyCode == 40) {  
    dragao.y = dragao.y + 20;  
  }  
}
```

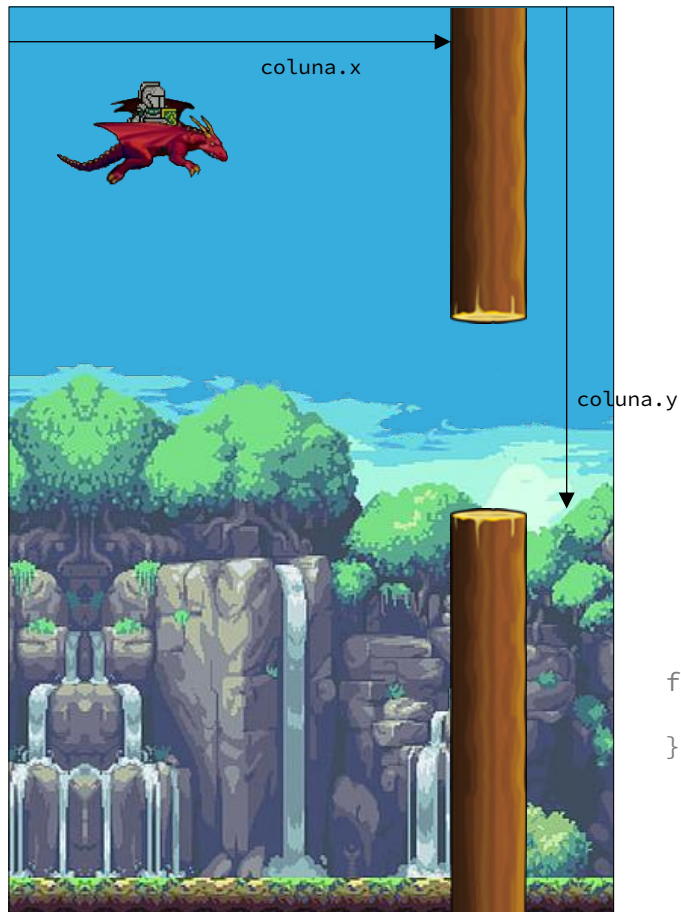
Exemplos de keyCodes:

- Enter = 13
- Escape = 27
- Seta para esquerda = 37
- Seta para cima = 38
- Seta para direita = 39
- Seta para baixo = 40



Etapa 7





Sorteia posição horizontal

```
coluna.x = AleatorioEntre(400, 800);
```

Sorteia posição vertical

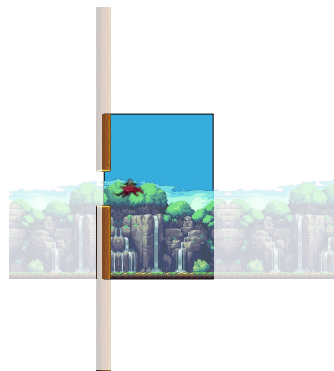
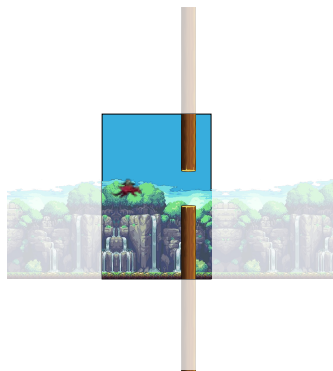
```
coluna.y = AleatorioEntre(220, 570);
```

```
function AleatorioEntre(min, max) {  
    return Math.floor(Math.random() * (max - min + 1) + min);  
}
```



Pode-se aumentar o valor de `coluna.speed` para deixar o jogo mais difícil.

Pode-se diminuir o valor de `coluna.space` para deixar o jogo mais difícil.





Etapa 8





Colisão

Análise horizontal

`dragao.x + dragao.width >= coluna.x`





Colisão

Análise vertical (coluna inferior)

`dragao.y + dragao.height >= coluna.y`

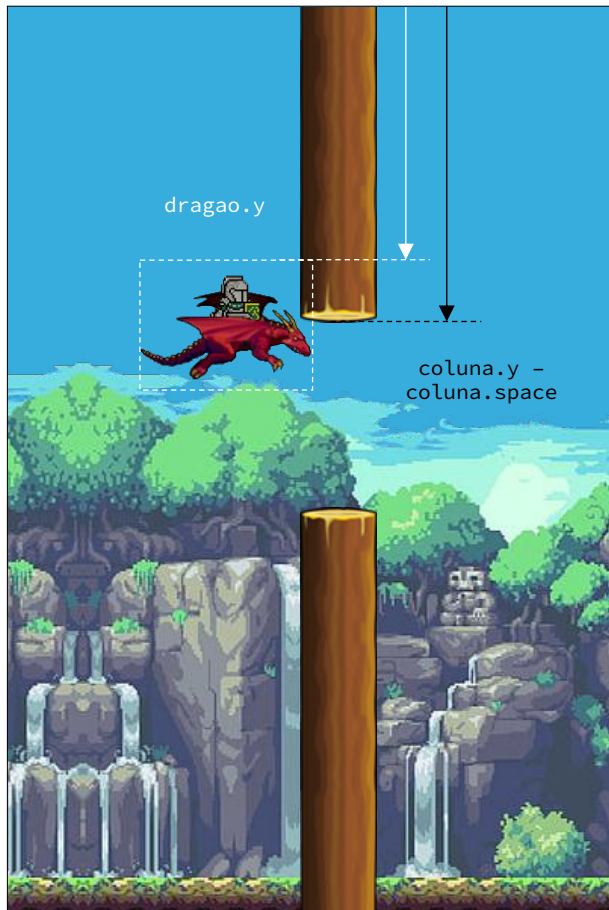




Colisão

Análise vertical (coluna superior)

`dragao.y <= coluna.y - coluna.space`





Colisão

Chão

`dragao.y + dragao.height >= canvas.height`





Etapa 9





2

```
<!DOCTYPE html>
<html>
  <head>
    <title>Game Etapa 9</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <canvas id="canvas" width="400" height="600"></canvas>
    <div id="placar">0</div>
    <script src="script.js"></script>
  </body>
</html>
```

document.getElementById('placar').innerHTML = jogo.score;



Objeto para registrar
o placar (score)

```
var jogo = {  
  score: 0,  
  scored: 0  
}
```

Ainda não pontuou
nessa rodada*

As colunas já passaram
pelo dragão nessa rodada*

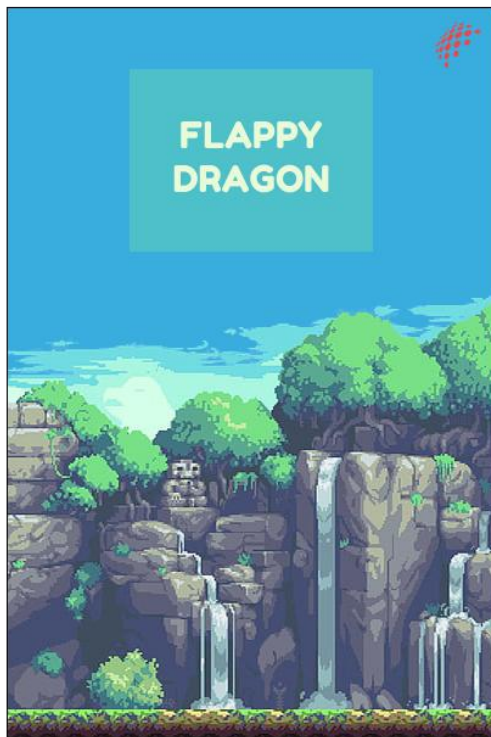
```
if (jogo.scored == 0 && coluna.x + coluna.width <= dragao.x) {  
  jogo.scored = 1;  
  jogo.score++;  
}
```

*Cada vez que aparecem as
colunas (obstáculos), o aplicativo
considera com uma "rodada"



Final

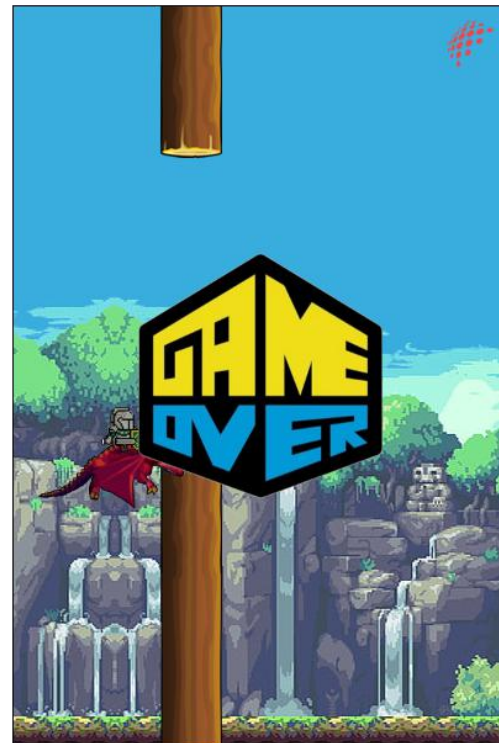




0

pressione ENTER para iniciar

Inteli - Instituto de Tecnologia e Liderança



5

pressione ENTER para reiniciar

Inteli - Instituto de Tecnologia e Liderança