Insper

Design de Software

Aula 1 – Programação Visual

17/Fev/2016 – Engenharia

Fábio Ayres <FabioJA@insper.edu.br>
Raul Ikeda <RaulIGS@insper.edu.br>

Objetivos de Aprendizado (do curso)

- Essenciais:
  - Desenvolver programas de computador.
  - Identificar e implementar estratégias algorítmicas computacionais.
  - Atuar em uma equipe autogerenciada de desenvolvimento.
- Complementares:
  - Identificar requisitos com técnicas de design.
  - Desenvolver interfaces gráficas de usuários.
  - Julgar e priorizar seu aprendizado da disciplina.

Datas Importantes

- EP - Exercícios Programa
  - EP1 – Entrega 02/Mar
  - EP2 – Entrega 16/Mar
  - EPMS – Entregas 11/Mar e 23/Mar
  - EP3 – Entrega 18/Abr
- Provas
  - P1 – 28/Mar
  - P2 – 30/Mai
  - PSub – 13/Jun

- **PROJETO**
  - **Entrega dia 24/MAIO**

Informações importantes

- Office Hours (Atendimento)
  - Turma A: Seg 11h45~13h15
  - Turma B: Ter 09h45~11h15
  - Turma C: Ter 07h30~09h00
- Ninjas
  - Turma A: Eric e Gustavo
  - Turma B: Matheus e Pedro
  - Turma C: Marcelo e Vitor

Atenção: verificar periodicamente o Blackboard e o e-mail para comunicados importantes

Critérios de Avaliação

- NI - Nota Individual:

NI – NOTA INDIVIDUAL	
Instrumento	**Peso**
EP1	10%
EP2	15%
EPMS	5%
P1	35%
P2	35%

- NG - Nota em Grupo:

NG – NOTA DE GRUPO	
Instrumento	**Peso**
EP3	30%
Projeto	70%

Critérios de Avaliação

- Conceito Final do curso:

Insatisfatório (I)	Em desenvolvimento (D)	Essencial (C)	Proficiente (B)	Avançado (A)
O aluno apresentou conceito I em NI ou em NG.	O aluno apresentou um conceito D em NI e um conceito superior a I em NG. O aluno apresentou um conceito C em NI e um conceito D em NG.	O aluno apresentou um conceito C em ambos. O aluno apresentou um conceito B ou A em NI e um conceito D em NG.	O aluno apresentou um conceito C em NI e um conceito B ou A em NG. O aluno apresentou um conceito B em NI e um conceito C ou B em NG. O aluno apresentou um conceito A em NI e um conceito C em NG.	O aluno apresentou um conceito B em NI e um conceito A em NG. O aluno apresentou conceito A em NI e pelo menos B em NG.

		NG				
		I	**D**	**C**	**B**	**A**
NI	**I**	I	I	I	I	I
	D	I	D	D	D	D
	C	I	D	C	B	B
	B	I	C	B	B	A
	A	I	C	B	A	A

Insper

Um curso de programação?

Alguém já programou?

Para que programou?

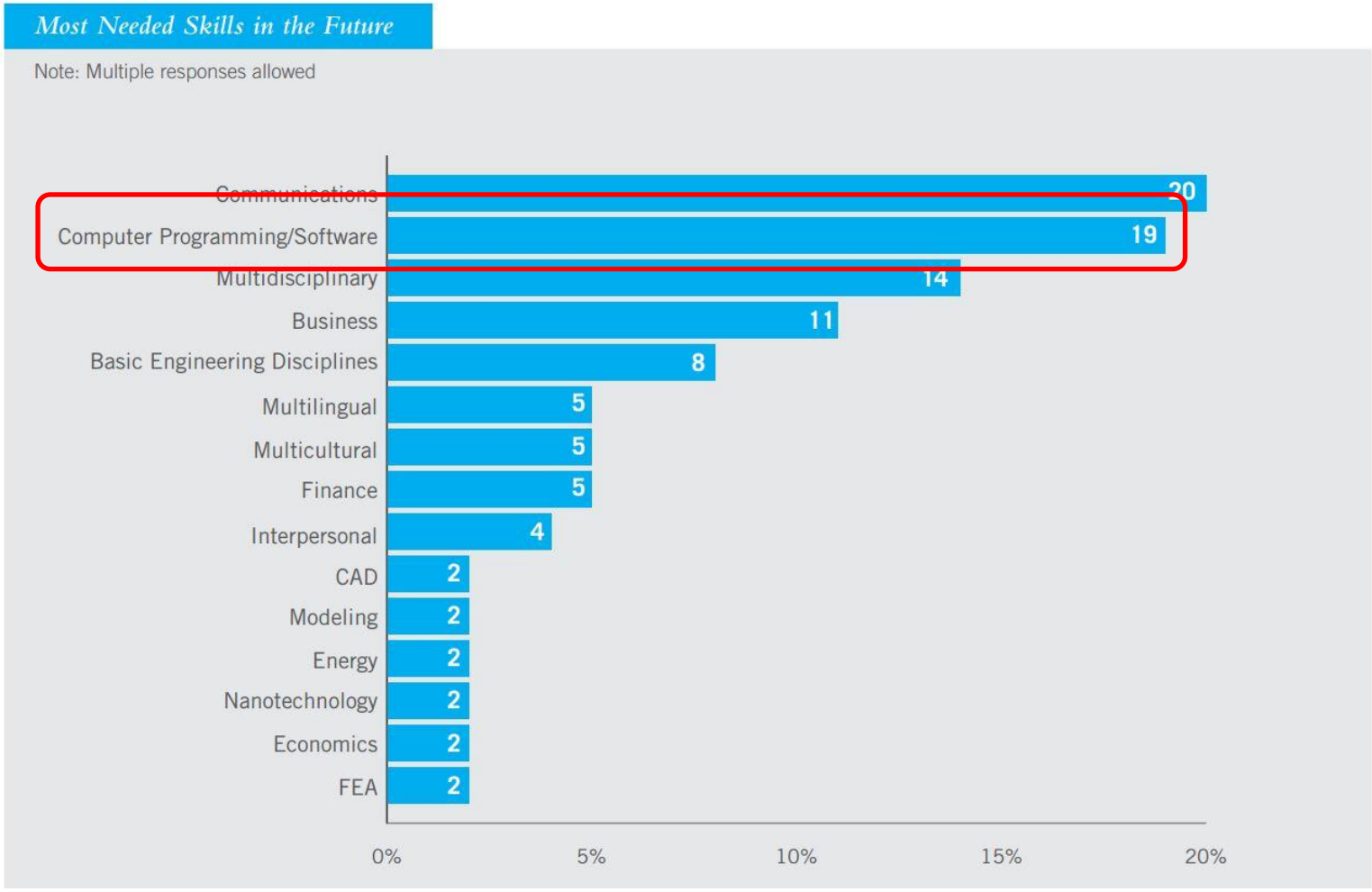
Como usou o programa?

O que é programar?

Porque programar?

Como programar?

Survey Eng. Mecânica



Fonte: The American Society of Mechanical Engineers.
"The State of Mechanical Engineering: Today and Beyond", Technical Report, 2011

Insper

Linguagens de programação?

http://oldcomputers.net/altair-8800.html e

Como começamos a programar

Hello World!: programa de computador que imprime "Hello, World!”, usualmente seguido de uma quebra de linha. Utilizado como um exemplo de código minimalista para introduzir as pessoas a uma linguagem de programação.

Linguagens e mais linguagens

"Hello, world" em várias linguagens

[ASSEMBLY]

```
MODEL SMALL
IDEAL
STACK 100H

DATASEG
        MSG DB 'Hello, world!', 13, '$'

CODESEG
Start:
        MOV AX, @data
        MOV DS, AX
        MOV DX, OFFSET MSG
        MOV AH, 09H
        INT 21H
        MOV AX, 4C00H
        INT 21H
END Start
```

[JAVA]

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

[COBOL]

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.
PROCEDURE DIVISION.
    DISPLAY "Hello, world!"
    STOP RUN.
```

[C]

```
#include <stdio.h>
int main(void) {
    printf("Hello, World!\n");
    return 0;
}
```

[BASIC]

```
10 PRINT "Hello, world!"
20 END
```

[PASCAL]

```
Program Hello;
begin
    writeln('Hello, world.')
end.
```

[FORTRAN]

```
00    program hello
      write(*,*) 'Hello World!'
      stop
      end
```

[PROLOG]

```
:- write('Hello, world!'),nl.
```

[PYTHON]

```
print("Hello, world!")
```

[BATCH]

```
@echo off
echo Hello World!
pause
exit
```

[VBA]

```
Private Sub Form_Click()
    Form1.Hide
    Dim HelloWorld As New Form1
    HelloWorld.Width = 2500: HelloWorld.Height = 1000: HelloWorld.Caption = "Hello, world!":
    HelloWorld.CurrentX = 500: HelloWorld.CurrentY = 75: HelloWorld.Show: HelloWorld.Font = "Tahoma":
    HelloWorld.FontBold = True: HelloWorld.FontSize = 12: HelloWorld.Print "Hello, world!"
End Sub
```

Insper

http://en.wikibooks.org/wiki/Computer_Programming/Hello_world

Abstração

Linguagens de alto nível

- Python
- C
- Fortran

Linguagens Assembly

- ARM assembler
- X86 assembler
- Multiple target assemblers

Linguagens de Máquina

- 01010101010100101010110

Começando de forma visual

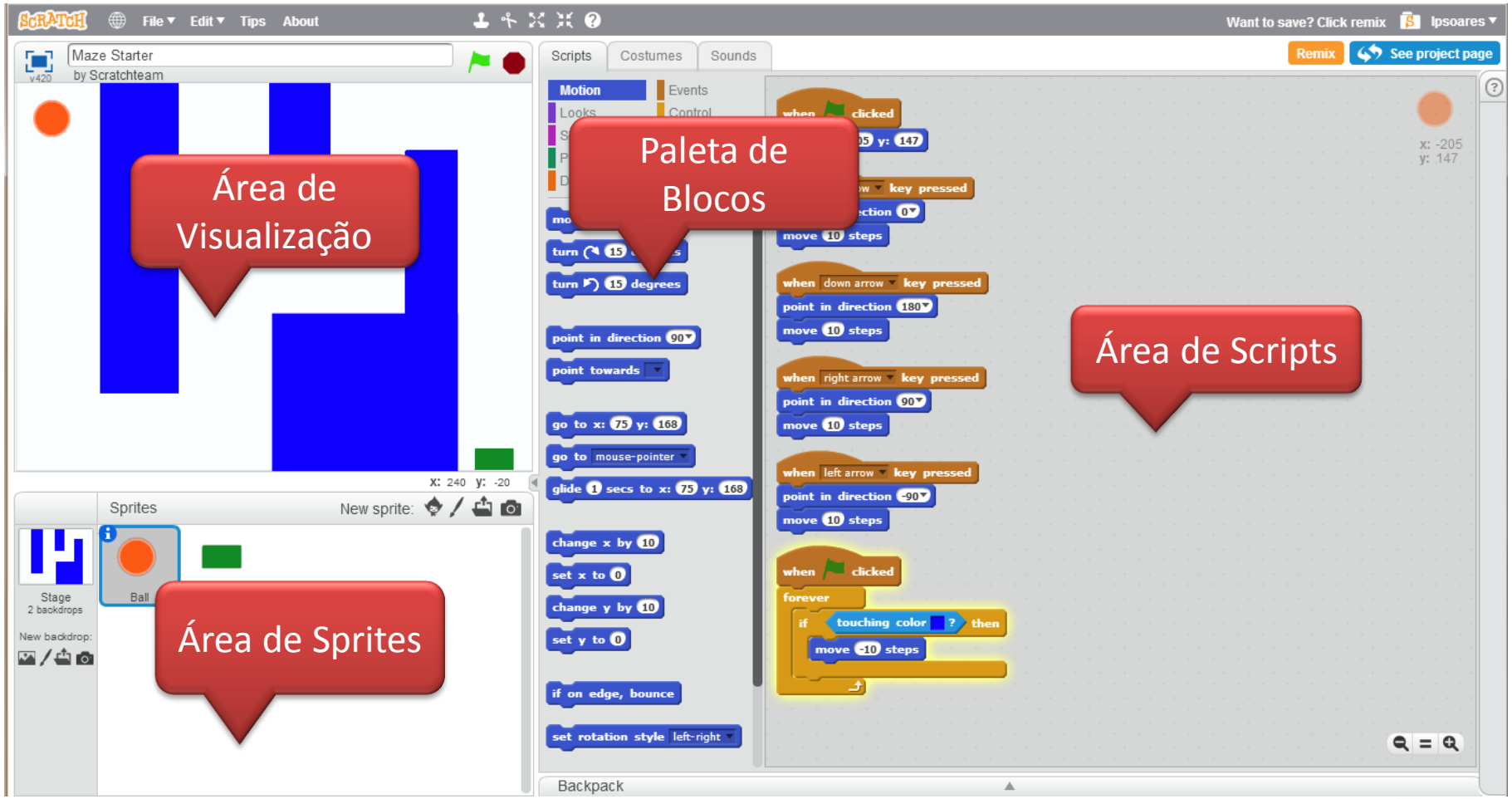
http://scratch.mit.edu/

Programando em pares

Em duplas vamos explorar a ferramenta: http://scratch.mit.edu/

Como isso funciona?

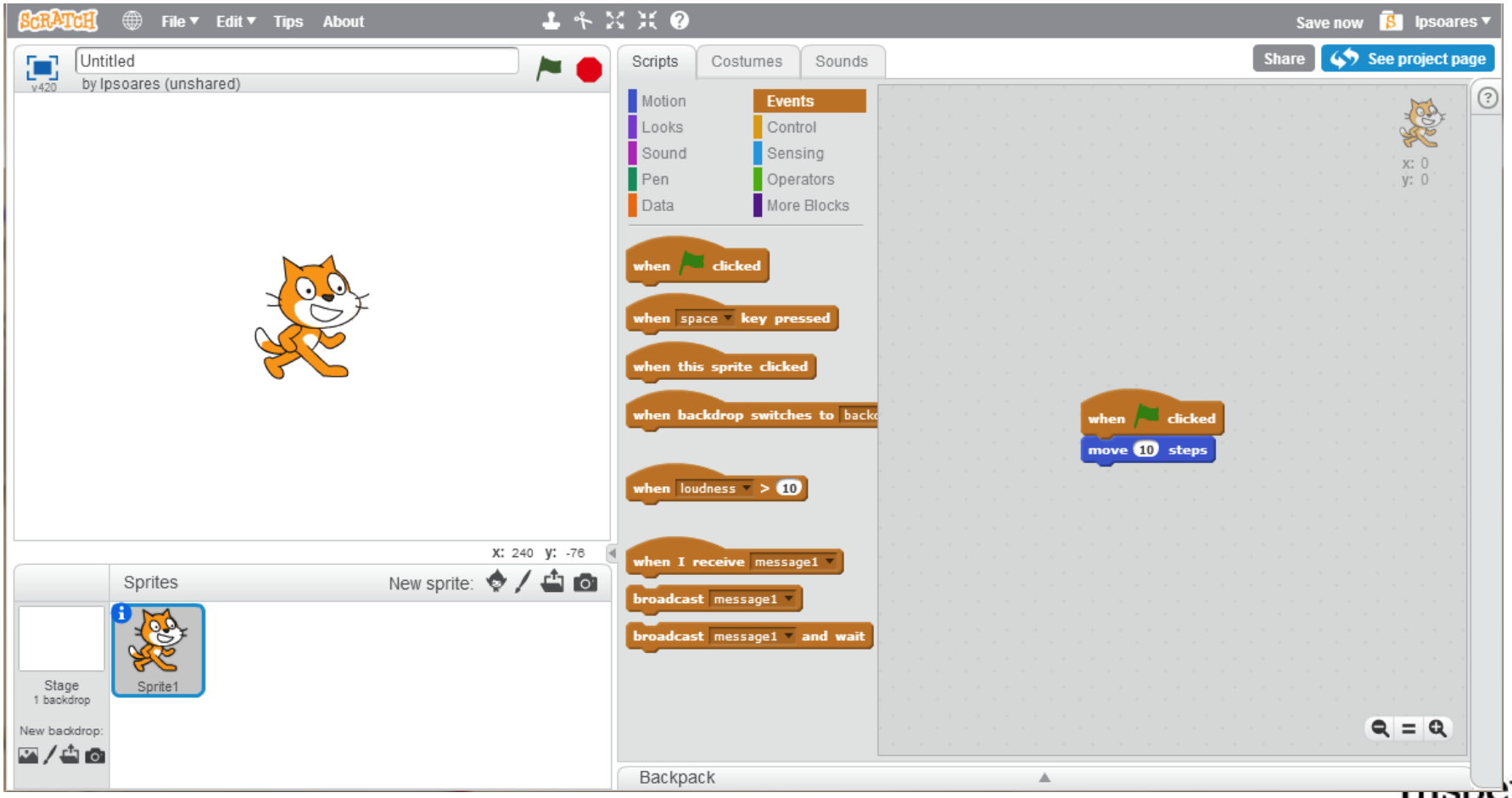
Conecte os blocos para fazer seu programa:



Insper

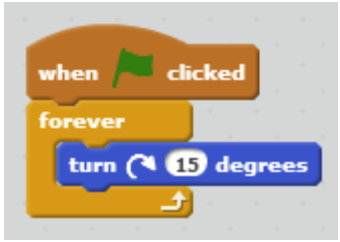
Testando algo simples

Mova o gato?

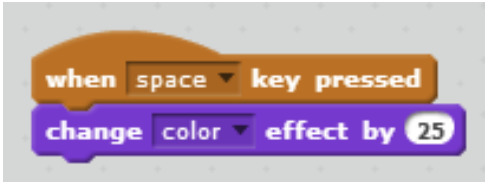


Brincando com o gato

Faça o gato girar:



Troque a cor com a barra de espaço:



Faça o gato andar e miar:

Faça o gato seguir o mouse:

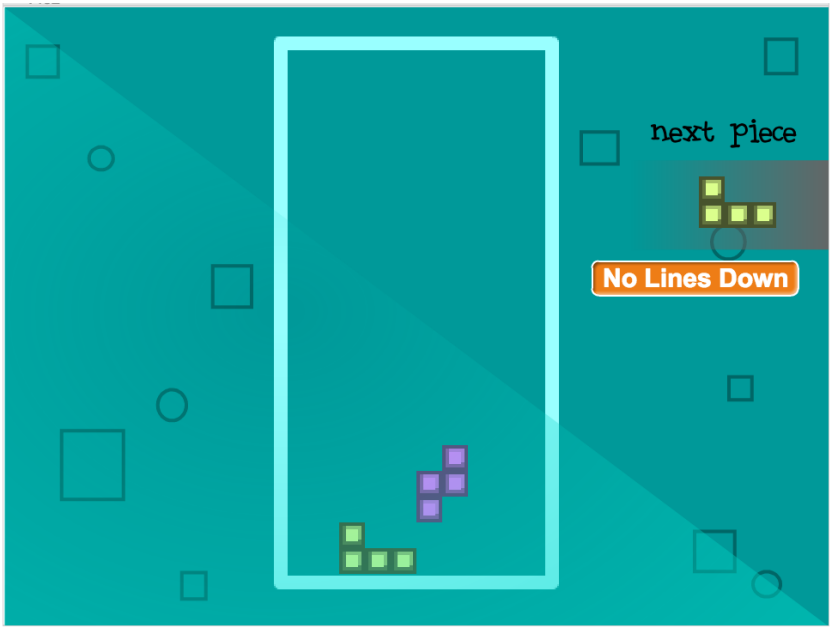
Insper

Experimente um pouco

Baseado no que você sabe até agora tente fazer outras coisas.

Isso mesmo brinque com a ferramenta livremente.

Alguns projetos com Scratch



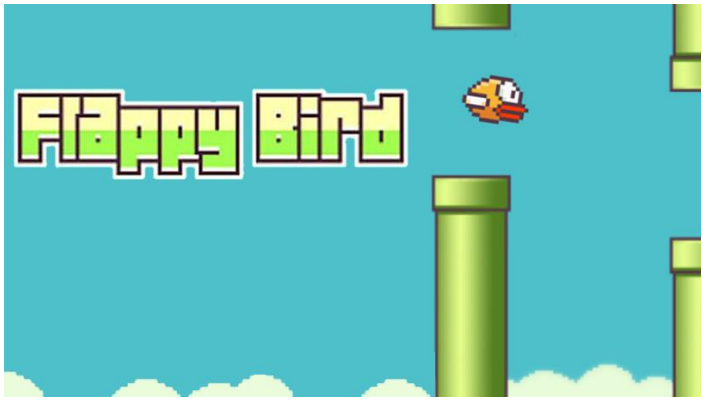
http://scratch.mit.edu/projects/41627584

http://scratch.mit.edu/projects/44645380

Insper

Flappy Bird

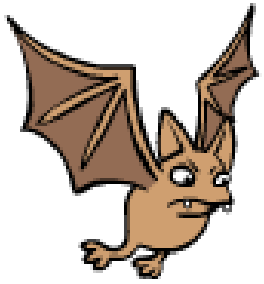
Flappy Bird foi, ou ainda é, um jogo desenvolvido em 2013 pelo vietnamita Nguyễn Hà Đông para plataformas Android e iOS. Notável pela sua difícil jogabilidade, rapidamente se tornou viral e foi o jogo gratuito mais instalado na AppStore chinesa e americana em Jan/2014. Apesar de altamente rentável, e também sob polêmicas em torno da produção e divulgação, o jogo foi removido no dia 09/Fev/2014. O criador alegou que o jogo era "viciante demais".



https://pt.wikipedia.org/wiki/Flappy_Bird

Como começamos?

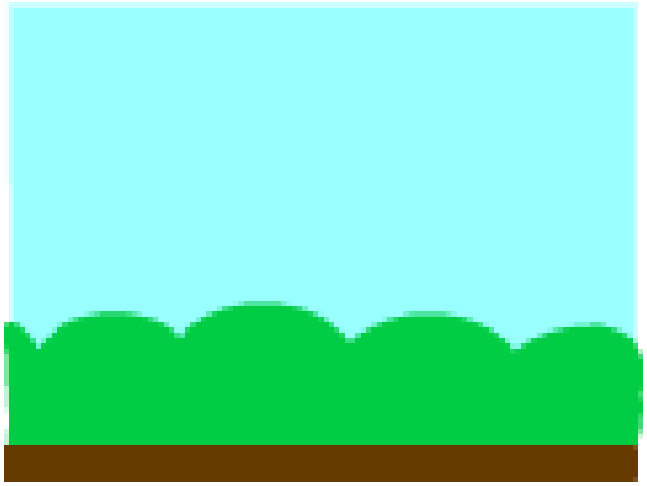
Escolha um ator para voar



Bat1

Como começamos?

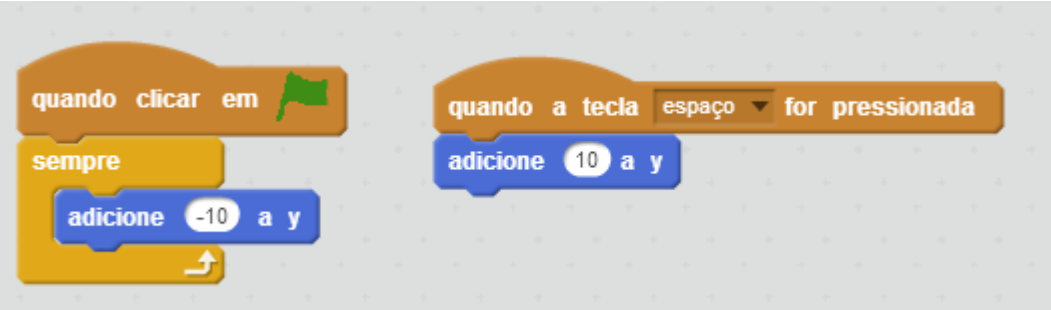
Escolha um pano de fundo



blue sky

Como começamos?

Faça o morcego cair a medida que o tempo passa e subir quando apertar espaço

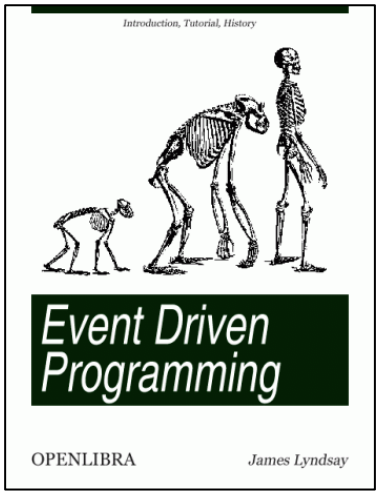


Obs: O uso do “sempre” é necessário ou então ele somente irá se mover uma vez. Não se preocupem com a dinâmica, o jogo ainda está incompleto. Perceberam que ele não retorna ao ponto inicial? Faltou colocar o bloco para posicioná-lo no início:

Programação orientada a eventos
(Event-driven programming)

Programação orientada a eventos é um paradigma de programação em que o fluxo do programa é determinado por eventos, como as ações do usuário (cliques do mouse, teclas pressionadas), saídas de sensores, ou mensagens de outros programas / threads.

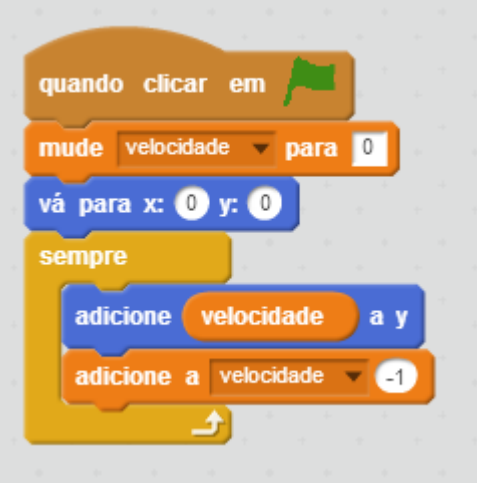
Este é o paradigma dominante usado em interfaces gráficas.



Desenvolvendo jogo Flappy Bat

Vamos tornar mais realístico: Um objeto cai com uma velocidade constante? Como implementar um decaimento que aumenta com o passar do tempo?

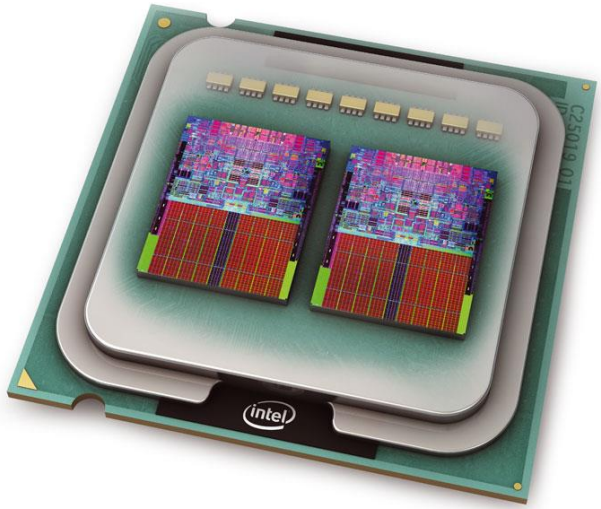
Dica: usar uma variável para guardar a velocidade atual.



Obs: Não se esqueça de zerar a velocidade no início, ou ele usará o último valor da última execução.

Multitarefa (Multitasking)

Multitarefa é a capacidade que o computador possui de executar várias sequencias de instruções simultaneamente. Esse paralelismo pode ser alcançado pela divisão do tempo da CPU ou rodando em “cores” independentes.



Desenvolvendo jogo Flappy Bat

Aplique o mesmo conceito de velocidade para quando o espaço for apertado. Como é a física do movimento? Preciso usar o “sempre”?

Pense em como os dois blocos agirão simultaneamente.

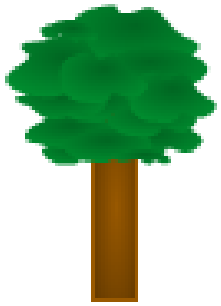
Obs: Calibrem os valores que utilizarão, lembrando que isso é um dos fatores que influenciam na dificuldade do jogo.

Orientação a objetos
(Object-orientation)

Modelo de programação em que são definidos os tipos de dados em conjunto com os tipos de operações (métodos). Desta forma é criado um objeto que inclui ambos os dados e as funções. Além disso, os programadores podem criar relações entre um objeto e outro.

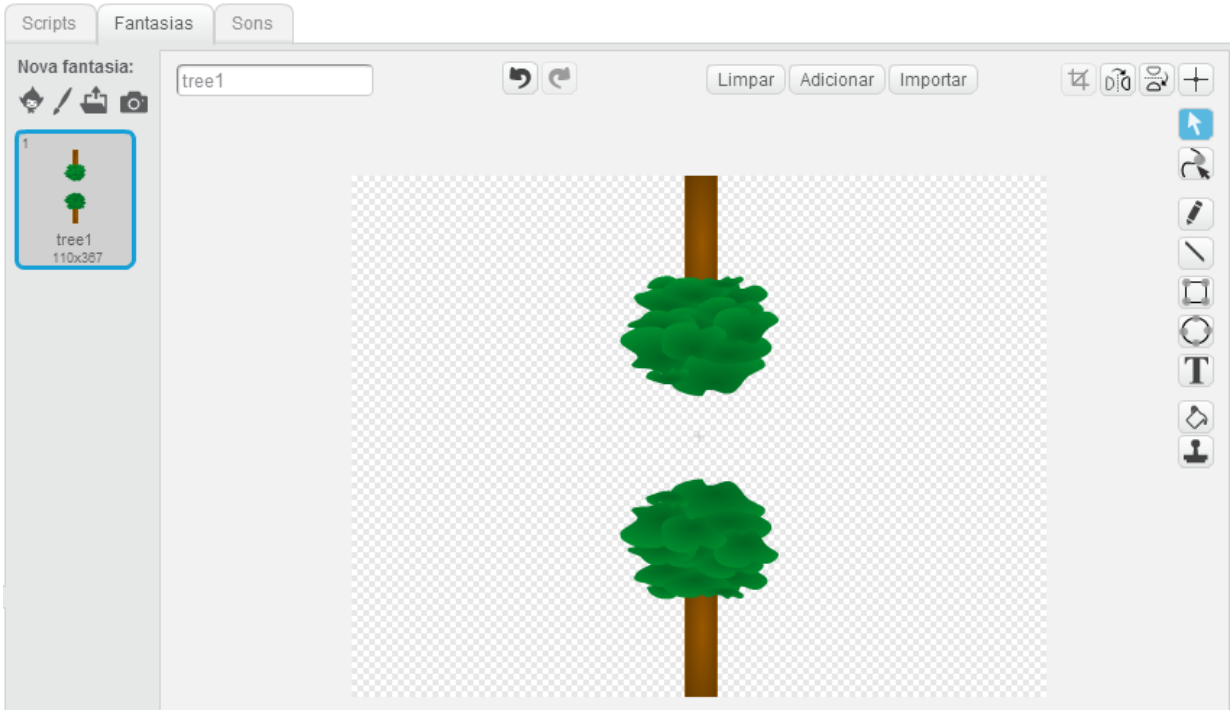
Vamos colocar um novo ator (objeto)

Escolha algum objeto que seja comprido verticalmente (escolha um desenho e não uma foto, pois ele é vetorizado)



Tree1

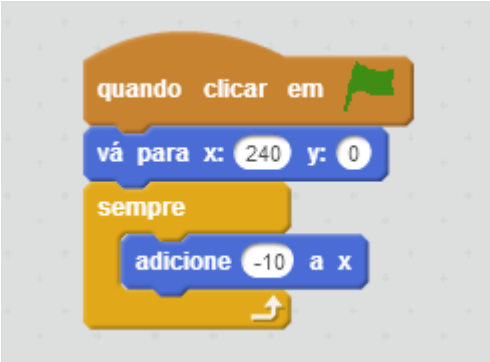
Vamos editar o desenho. Na aba fantasias é possível realizar algumas transformações. Crie uma cópia e espelhe verticalmente para criar a seguinte figura:



O espaço entre as árvores também é um fator que influencia na dificuldade do jogo.

Desenvolvendo jogo Flappy Bat

Vamos mover a árvore da direita para esquerda. Agora o bloco de programa estará dentro do segundo ator e não do primeiro. Posicione em (240,0) e mova gradualmente para a esquerda.



Condicionais

If (se algo faça)

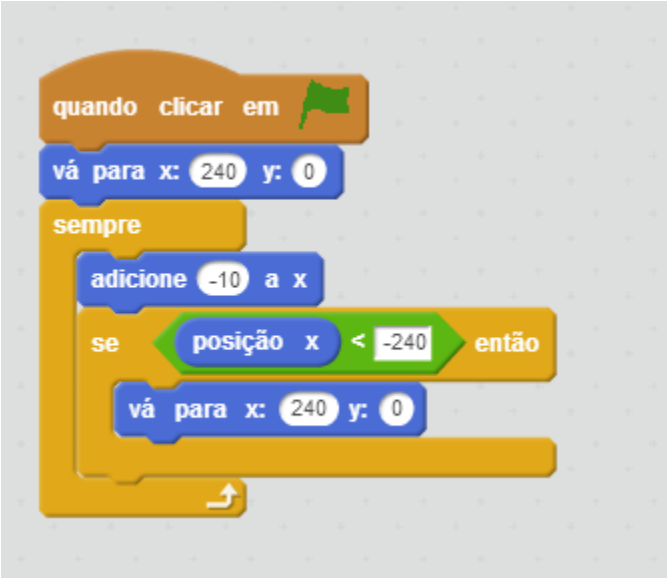
Else (senão faça)

Exemplo:

Se chover vou de carro, senão vou andando.

Desenvolvendo jogo Flappy Bat

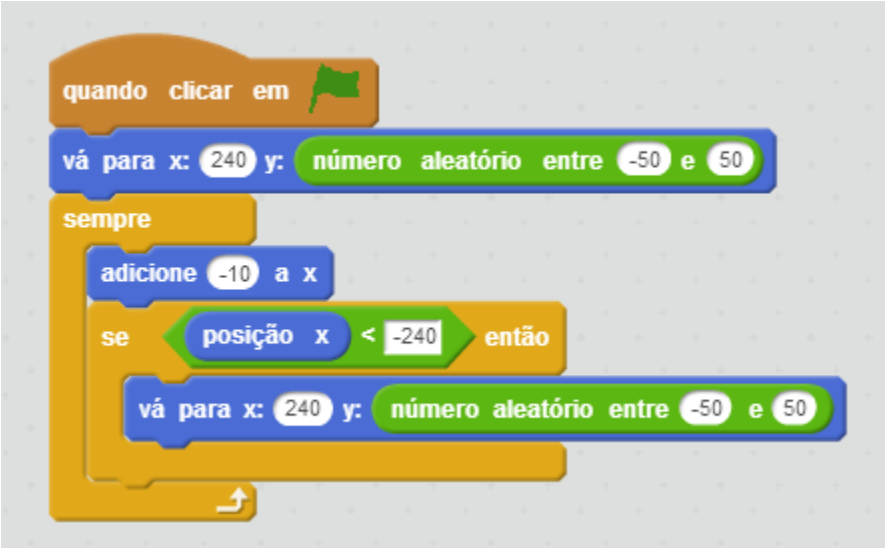
Se a árvore chegar em (-240,0) temos que retorná-la a posição inicial. Utilize o "Se" para testar a condição, caso seja verdadeiro, volte para a posição inicial. Aonde colocamos o bloco do "Se"?



Obs: Alguns computadores apresentam medidas e arredondamentos diferentes. Por isso a utilização do "<" ao invés de apenas "="

Desenvolvendo jogo Flappy Bat

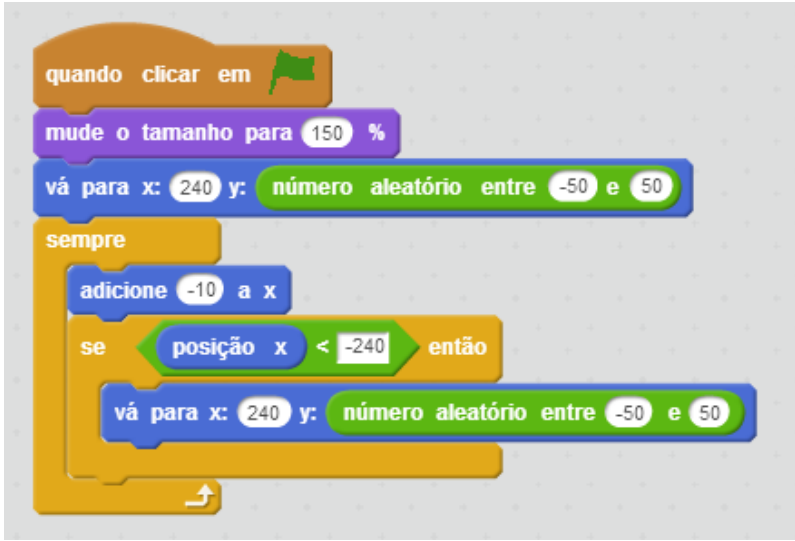
No bloco anterior, árvore sempre estará na mesma posição vertical. Vamos sortear a posição inicial em y para que o buraco se mova verticalmente cada vez que voltar à direita.



Desenvolvendo jogo Flappy Bat

Antes de criarmos o critério de parada, vamos arrumar alguns detalhes na aparência:

1. Diminuir um pouco o tamanho do morcego no início do programa.
2. Aumentar um pouco o tamanho da árvore.
3. Mudar a fantasia quando ele começar a subir. Não esquecer de mudar de volta quando acabar. Por que aparentemente não mudou nada?



Condicionais

Repeat until (repita enquanto)

Exemplo:

Repita enquanto não aprender esse exercício.

Desenvolvendo jogo Flappy Bat

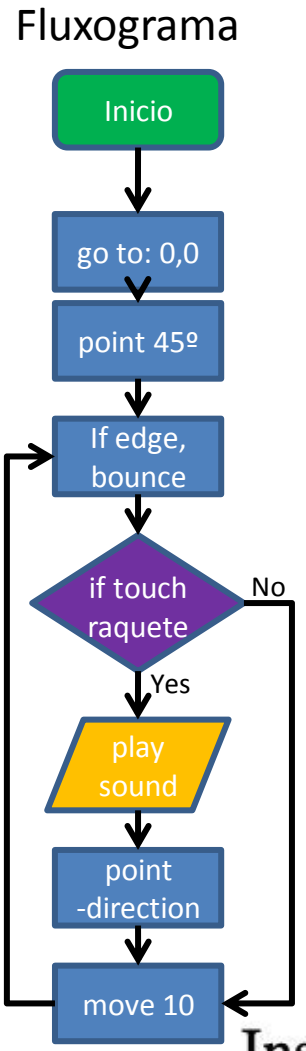
Critério de parada: o jogador perde quando o morcego encostar na árvore. Como programar esse critério? Lembre-se que as árvores devem parar e o morcego não pode mais subir.

Dica: substituir o “sempre” e utilizar “se” com uma variável indicativa de final de jogo.

Programação estruturada

Programação estruturada é uma forma de programação que preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: sequência, decisão e repetição.

Desenvolvida por Michael A. Jackson no livro "Principles of Program Design" de 1975.



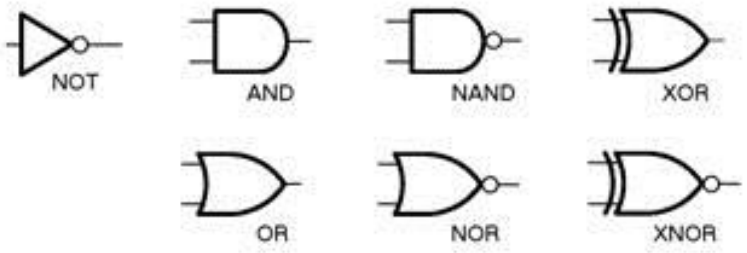
Insper

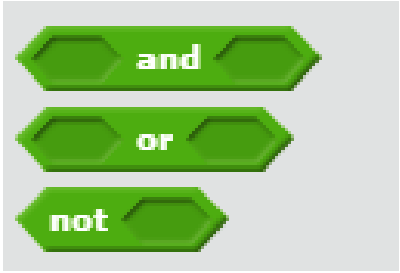
Desenvolvendo jogo Flappy Bat

Modularize a sequência de subida:

Álgebra (Lógica) booleana

A Álgebra Booleana é uma álgebra binária, pois se aplica a variáveis que podem assumir apenas dois valores distintos, permitindo avaliar uma proposição lógica como verdadeira ou falsa.





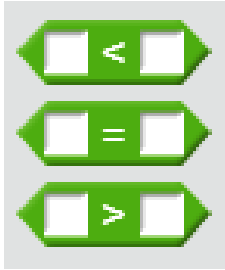


Tabela Verdade

A	B	A AND B
Falso	Falso	Falso
Falso	Verdadeiro	Falso
Verdadeiro	Falso	Falso
Verdadeiro	Verdadeiro	Verdadeiro

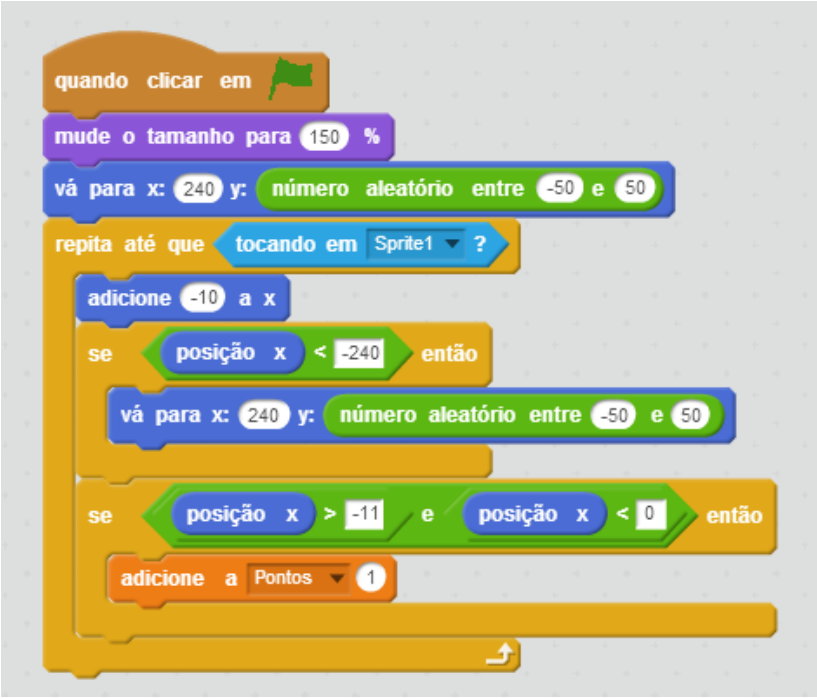
Tabela Verdade

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Desenvolvendo jogo Flappy Bat

Contagem dos pontos: Como contar os pontos? Toda vez que ele passar entre as árvores deve somar 1 em pontos.

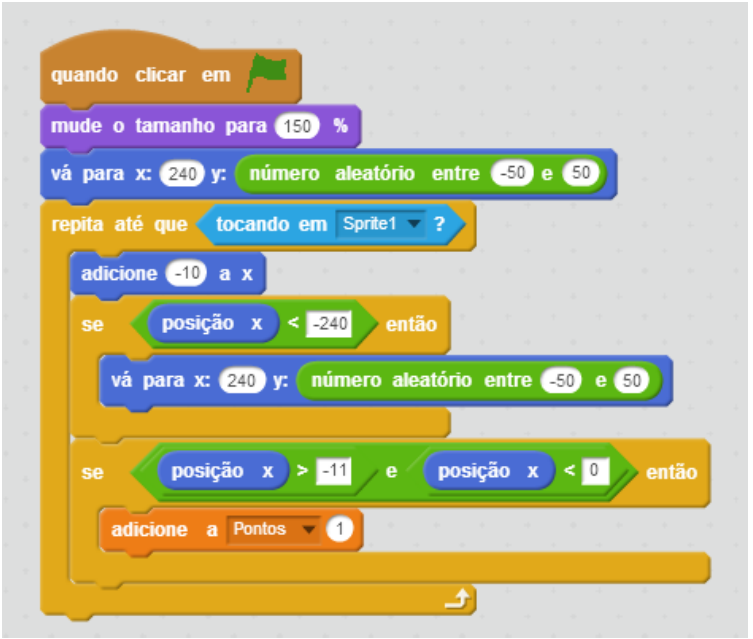
Dica: utilizar um intervalo para verificar, lembrando que podemos ter erros de arredondamento dependendo do computador que o usuário utiliza.



Insper

Desenvolvendo jogo Flappy Bat

Código final:



Insper

Desenvolvendo jogo Flappy Bat

O que mais pode ser feito:

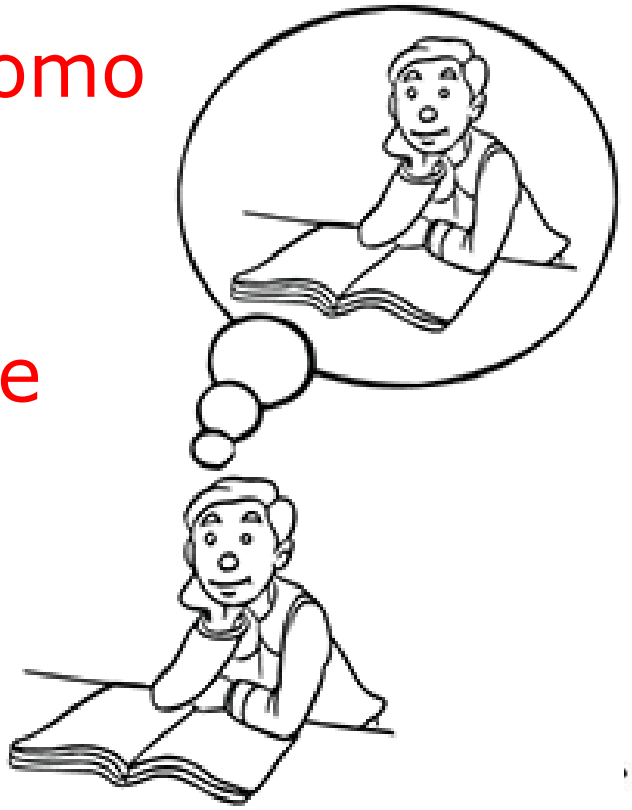
- Calibrar os valores para deixar o jogo mais fácil ou mais difícil.
- Colocar moedas em posições aleatórias para aumentar os pontos.
- Alguma outra ideia?

Vamos dar uma voltinha?

Circulem e vejam os projetos dos seus colegas.
Discutam com eles as soluções adotadas.

Reflexão

- Ajudou para a programação trabalhar em pares?
- Quando deu algo errado como vocês resolveram?
- Qual foi a maior dificuldade técnica?



Para as próximas aulas (1)

Instalar no computador o Anaconda:

http://continuum.io/downloads

Atenção: Seguir todas as instruções de instalação na página.

- Windows Users: selecionar a opção “Just Me” e não modificar as opções avançadas.

Para a próxima aula

- Ambiente Python
- Variáveis e Entrada de Dados

Capítulos 2 e 3 do livro
Introdução à Programação com Python, de Nilo Ney Menezes.

(disponível na biblioteca)

Insper

Insper

www.insper.edu.br

Fun: Pessoas codificando Hello World!

High School/Jr. High

```
10 PRINT "HELLO WORLD"
20 END
```

First year in College

```
program Hello(input, output)
begin
writeln('Hello World')
end.
```

Senior year in College

```
(defun hello
(print
(cons 'Hello (list 'World))))
```

New professional

```
#include <stdio.h>

void main(void)
{
 char *message[] = {"Hello ", "World"};
 int i;
 for(i = 0; i < 2; ++i)
 printf("%s", message[i]);
 printf("\n");
}
```

Seasoned professional

```
#include <iostream.h>
#include <string.h>
class string
{
 private:
  int size;
  char *ptr;
 public:
  string() : size(0), ptr(new char('\0')) {}
  string(const string &s) : size(s.size)
  {
    ptr = new char[size + 1];
    strcpy(ptr, s.ptr);
  }
  ~string()
  {
    delete [] ptr;
  }
  friend ostream &operator <<(ostream &, const string &);
  string &operator=(const char *);
};

ostream &operator<<(ostream &stream, const string &s)
{
  return(stream << s.ptr);
}
string &string::operator=(const char *chrs)
{
  if (this != &chrs)
  {
    delete [] ptr;
    size = strlen(chrs);
    ptr = new char[size + 1];
    strcpy(ptr, chrs);
  }
  return(*this);
}
int main()
{
  string str;
  str = "Hello World";
  cout << str << endl;
  return(0);
}
```

System Administrator

```
#include <stdio.h>
#include <stdlib.h>
main()
{
 char *tmp;
 int i=0;
 /* on y va bourin */
 tmp=(char *)malloc(1024*sizeof(char));
 while (tmp[i]="Hello Wolrd"[i++]);
 /* Ooopps y'a une infusion ! */
 i=(int)tmp[8];
 tmp[8]=tmp[9];
 tmp[9]=(char)i;
 printf("%s\n",tmp);
}
```

Apprentice Hacker

```
#!/usr/local/bin/perl
$msg="Hello, world.\n";
if ($#ARGV >= 0) {
  while(defined($arg=shift(@ARGV))) {
    $outfilename = $arg;
    open(FILE, ">" . $outfilename) || die "Can't write $arg: $!\n";
    print (FILE $msg);
    close(FILE) || die "Can't close $arg: $!\n";
  }
} else {
  print ($msg);
}
1;
```

Experienced Hacker

```
#include <stdio.h>
#include <string.h>
#define S "Hello, World\n"
main(){exit(printf(S) == strlen(S) ? 0 : 1);}
```

Seasoned Hacker

```
% cc -o a.out ~/src/misc/hw/hw.c
% a.out
Hello, world.
```

Guru Hacker

```
% cat
Hello, world.
```

New Manager (do you remember?)

```
10 PRINT "HELLO WORLD"
20 END
```

Middle Manager

```
mail -s "Hello, world." bob@b12
Bob, could you please write me a program that prints "Hello, world."?
I need it by tomorrow.
^D
```

Senior Manager

```
% zmail jim
I need a "Hello, world." program by this afternoon.
```

Chief Executive

```
% letter
letter: Command not found.
% mail
To: ^X ^F ^C
% help mail
help: Command not found.
% damn!
!: Event unrecognized
% logout
```

Research Scientist

```
                PROGRAM HELLO
                PRINT *, 'Hello World'
                END
```

Older research Scientist

```
                WRITE (6, 100)
  100 FORMAT (1H ,11HHELLO WORLD)
                CALL EXIT
                END
```

Insper