**RAPTOR IDE**

**Introdução**

O RAPTOR é uma aplicação de programação baseado num ambiente de fluxogramas, desenhado especificamente para ajudar os alunos a visualizar os seus algoritmos e evitar bagagem sintática. Com o RAPTOR, a programação é criada de forma visual e executada visualmente traçando a execução por fluxogramas. Requer o mínimo de sintaxe. Os alunos preferem usar fluxogramas para expressar os seus algoritmos que, de certa forma, traz mais sucesso para os alunos do que as tradicionais linguagens de programação ou desenhar fluxogramas sem o RAPTOR.

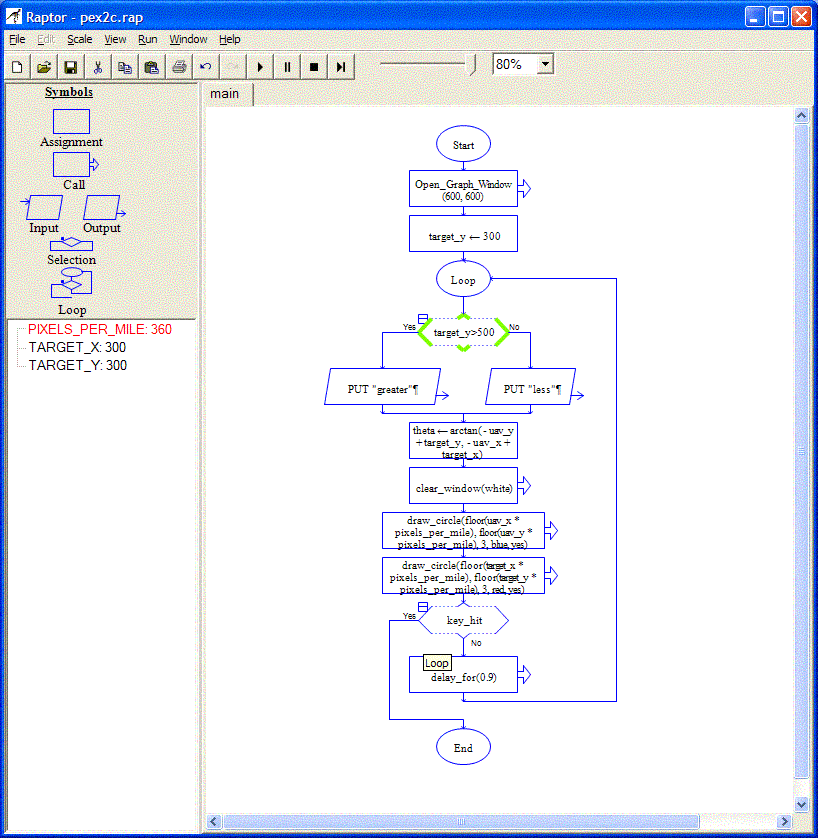


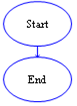
Fig1. Interface do RAPTOR

**Características do programa**

A linguagem RAPTOR tem o mínimo de sintaxe quando comparado com outras linguagens de programação. Isto faz com que seja mais fácil escrever algoritmos com RAPTOR, tornando mais eficiente e reduzindo o numero de linguagens que é necessário para aprender para se tornar produtivo.

Em segundo lugar, a linguagem RAPTOR é visual, ou seja, a programação é feita por diagramas (fluxogramas), como resultado, facilita a compreensão da programação.

**Estrutura do programa**

 A aplicação RAPTOR é executada diretamente num grafo com inicio = start ----🡪 fim = end. Quando se quer executar um programa, começa por carregar no botão START (botão com símbolo parecido ao ”play”) que está no topo, na barra de ferramentas.

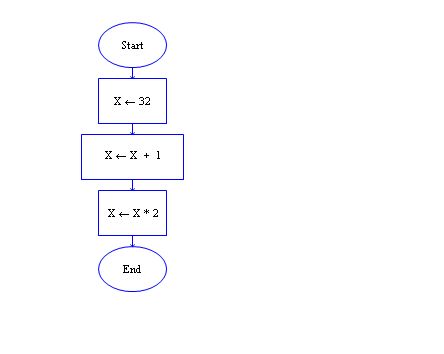
Certamente que se executar o programa só com as instruções inicio/fim, não fará nada.

Arrastando instruções entre o inicio e o fim do grafo, irá criar fluxogramas mais complexos e mais significativos.

**Variáveis RAPTOR**

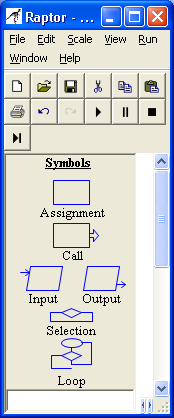
As variáveis são memórias locais que armazenam um valor. Em qualquer altura uma variável pode armazenar um valor específico de um tipo de dado, porém durante a execução do algoritmo, essa variável poderá sofrer alterações. Todas as variáveis terão de conter um nome distinto dado pelo programador, para facilitar a sua compreensão.

Os tipos de dados do RAPTOR para armazenar variáveis são:

* Inteiros
* String
* Boolean
* Float
* Arrays

Ao contrário da maioria das outras aplicações, as variáveis do RAPTOR são declaradas logo com um valor.

**Instruções RAPTOR**

 O RAPTOR tem 6 instruções básicas, Entrada (INPUT), Saída (OUTPUT), Declaração (ASSIGNMENT), Chamar função (CALL), Seleção (SELECTION) e Ciclo (LOOP). Qualquer um destas instruções é designado por um símbolo diferente.

As instruções de Seleção de Ciclo são também conhecidas como instruções de estrutura de controlo. São essas instruções que controlam como o programa executa.

 Para todas as instruções, dentro da janela de cada instrução, contém uma pequena informação a explicar o que se pode fazer nessa instrução, normalmente localizada acima das áreas de texto.

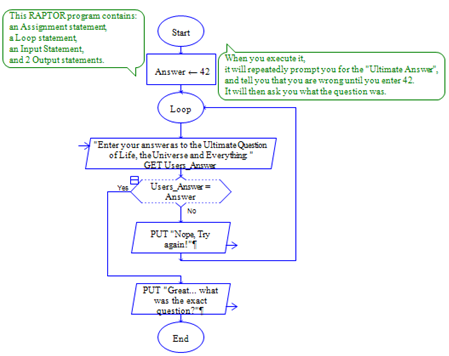
****

Fig2. Exemplo de um fluxograma em RAPTOR

**Operações e Funções**

Quando executados os operadores ou funções, esta irá devolver um resultado. O RAPTOR utiliza a seguinte construção de operadores e funções:

**Matemática básica:** +, -, \*, /, ^, \*\*, rem, mod, sqrt, log, abs, ceiling, floor

**Trigonometria:** sin, cos, tan, cot, arcsin, arcos, arctan, arccot

**Relacional:** =, !=, /=, <, >, >=, >=

**Logica:** and, or, not

**Outros:** random, Length\_of

Os operadores básicos de matemática incluem os operadores familiares (+, -, \*, /) assim como uns menos familiares.

\*\* e ^são exponenciais, exemplo 2\*\*4=16, 3^2=9.

rem (remainder = restante) e mod (module = módulo) retornam o resto (o que restou do resultado) quando o operando direito divide o operando esquerdo, exemplo: 10 rem 3 = 1, 10 mod 3 é também 1.

Sqrt retorna uma raíz quadrada, ex: sqrt(4) é 2

Log retorna um logaritmo natural, ex: log(e) é 1

Abs retorna um valor absoluto, ex: abs(-9) é 9

Ceiling arredonda para cima para um valor inteiro, ex: ceiling(3.14159) é 4

Floor arredonda para baixo para um valor inteiro, ex: floor(10/3) é 3

O + também trabalha como operação de concatenação para juntar duas strings ou um string e um número.

Length\_of retorna o numero de caracteres numa variável de strings, ex: teste é 5.

Os operadores de lógica e de relação só podem ser usados em instruções de seleção ou de ciclo.

**Constantes**

As constantes são variáveis pré-definidas cujos valores não podem ser alterados. O RAPTOR contem as seguintes variáveis:  
 **Constantes:** pi, e , true, false, yes, no

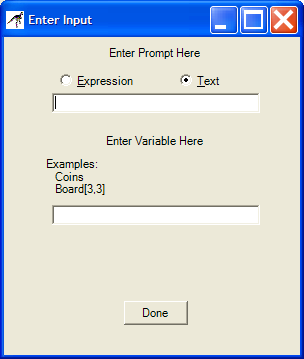
O PI tem o valor de 3.14159274101257.

O numero de nepper (e) tem o valor de 2.71828174591064

True e yes são definidos a 1

False e no são definidos a 0

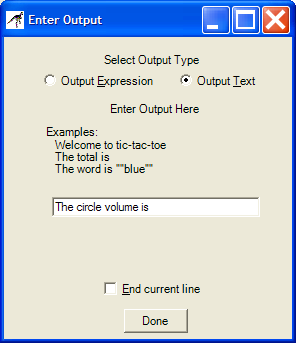
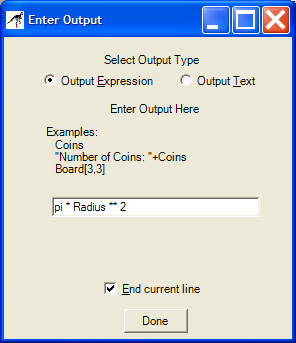
**Instruções de Input**

 Toda a linguagem de programação tem instruções que permite ao programa obter informação do utilizador do programa através do teclado, enviando essa informação para a consola.

Em RAPTOR, uma instrução de input mostra uma consola para o utilizador (para inserir um valor), e armazenar esse valor numa variável.

**Instruções de Output**

No RAPTOR, uma instrução de output é um tipo especial de procedimento que imprime o valor na consola. Uma caixa de diálogo pergunta para identificar se o utilizador está a escrever uma expressão ou texto.



Quando se imprime dados na consola, pode-se meter espaços depois do texto como o texto que está na figura acima do lado direito.

**Comentários no RAPTOR**

Como qualquer outra linguagem de programação, o RAPTOR permite adicionar comentários para o nosso algoritmo. O propósito de deixar um comentário no nosso algoritmo é explicar o algoritmo em si tornando-se mais fácil a sua compreensão.

**Outras coisas….**

Esta aplicação foi desenvolvida de duas formas, isto é, o desenvolvedor fornece a execução de instalador e uma versão portátil que pode executar a partir de uma pen, não necessitando de fazer a instalação.

Para mais informações, pode consultar a página <http://raptor.martincarlisle.com/>

Tem toda a documentação disponível (tutorial), links youtube e ainda várias versões da aplicação juntamente com as suas melhorias.