



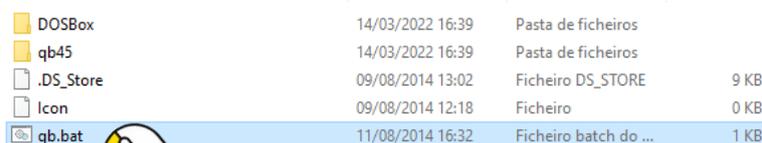
QB Neste guião irás aprender a **desenhar** em **QBasic**.

QB Os **elementos** de desenho são **importantes** para **criar personagens** ou **níveis** de um **jogo**.

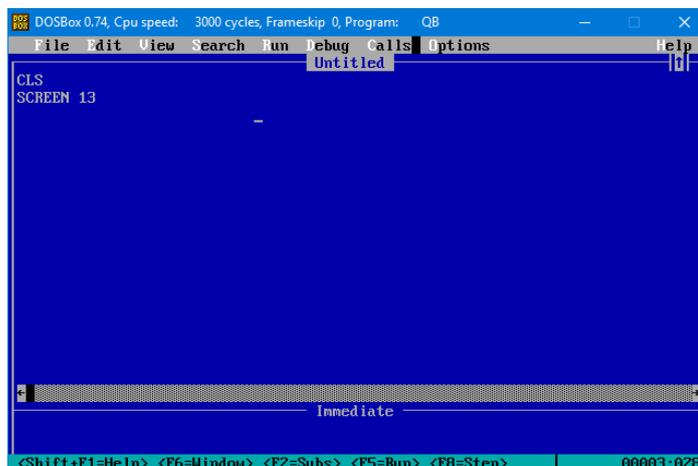


Imagem retirada de qbasic.net

QB Executa o ficheiro **QB.bat** para correres o IDE **Microsoft QuickBasic 4.50**



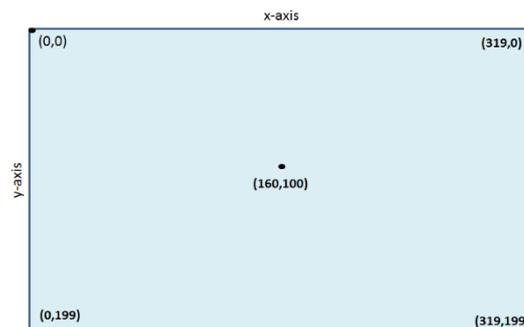
QB Digita as seguintes linhas de código:



Analisando o código:

A linha **SCREEN 13** define um ecrã de **resolução 320X200** com **256 cores**.

O **ponto mais à esquerda** e em cima é a origem **(0,0)**. O ponto **central** é metade do **SCREEN** neste caso **metade** de 320 (**160**) por metade de 200 (**100**). Os outros **extremos não** são **320** nem **200** porque ao começar no valor 0 daria ao todo mais um pixel que o ecrã, daí serem **319** e **199** os limites.





QB Vamos **pintar** todos os **pixels** dos **vértices** do ecrã.

QB Começa por **adicionar** as seguintes **linhas** de **código**:

```
SCREEN 13  
  
PSET (0, 0), 40  
PSET (0, 199), 40  
PSET (319, 199), 40  
PSET (319, 0), 40  
  
SLEEP
```

O SLEEP permite que **consigas** ver o ecrã **todo** antes do **programa** **terminar**.

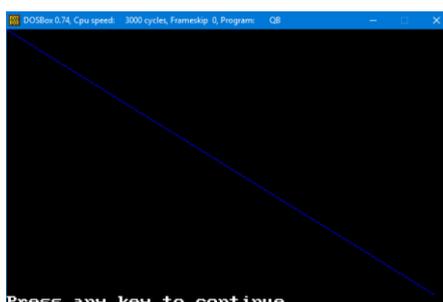
Deves ver um **pixel vermelho** em cada **canto** do ecrã devido ao PSET (Pixel SET). Modo de funcionamento:

PSET(X,Y), cor

QB Agora que **sabes** como está **definido** o ecrã e as **respetivas** **coordenadas** vamos **começar** a **desenhar**. **Acrescenta** as seguintes **linhas** de **código**:

LINE (x inicial, y inicial)-(x final, y final), cor da linha
LINE - traça uma linha entre o ponto inicial e final
vírgula
Cor da linha em numérico. Tabela simplificada de cores abaixo.

Corre o código. Resultado:



Ficam alguns códigos de cor para saberes:

0 Black	8 Dark gray
1 Dark blue	9 Light blue
2 Dark green	10 Light green
3 Dark cyan	11 Light cyan
4 Dark red	12 Light red
5 Dark magenta	13 Light magenta
6 Dark yellow	14 Yellow
7 Light gray	15 White

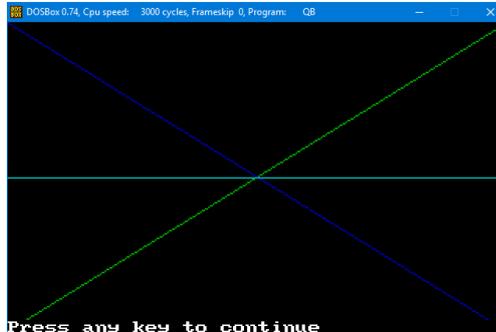
QB Acrescenta a seguinte **linha** de **código** ao teu **programa**:

```
LINE (0, 199)-(319, 0), 2
```



Este código traça uma linha na outra diagonal do ecrã.

Sabendo que a coordenada do centro do ecrã é (160,100) traça uma linha horizontal que passe no centro das outras duas (cor 3):



QB Altera as coordenadas dos 3 lines para as seguintes:

```
LINE (50, 50)-(50, 150), 40  
LINE (50, 150)-(150, 150), 100  
LINE (150, 150)-(50, 50), 60
```

QB Irás obter um triângulo.

QB O comando **LINE** também permite fazer retângulos. Começa por comentar as tuas 3 linhas de código para que não sejam lidas pelo programa:



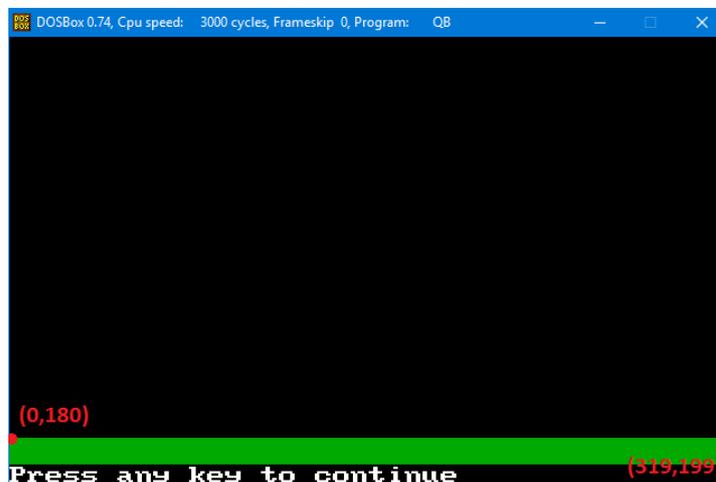
Para isso basta introduzir uma plica (apóstrofo) antes de cada linha de código.

Agora vamos fazer um retângulo que pareça o chão de um jogo de plataformas:

```
LINE (0, 180)-(319, 199), 2, BF
```

Neste caso temos mais um elemento o **BF**. **B** significa **BOX** que significa fazer um retângulo.

F quer dizer **FILL** quer dizer que o retângulo deve ser preenchido com a cor. Experimenta.



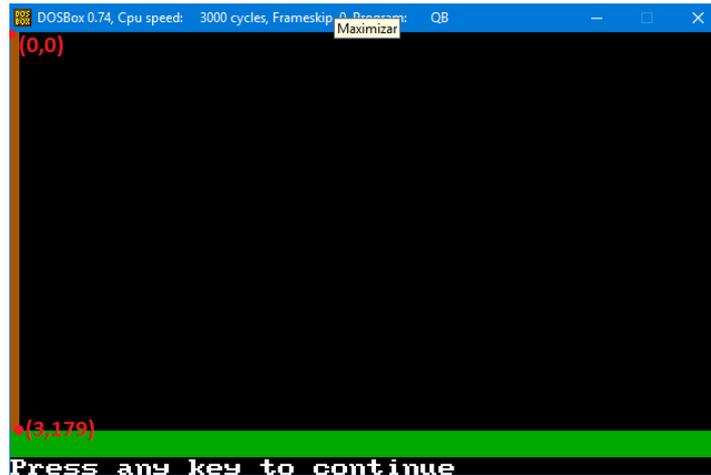


Como podes verificar o **retângulo** foi **pintado** entre a **coordenada inicial** e **final** que são **dois vértices** do **retângulo**. O inicial canto **superior esquerdo** e o mais **afastado** deste o **canto inferior direito**.

QB Vamos agora **fazer o rebordo lateral** (barra vertical) no **lado esquerdo** do **ecrã**. Acrescenta o seguinte código:

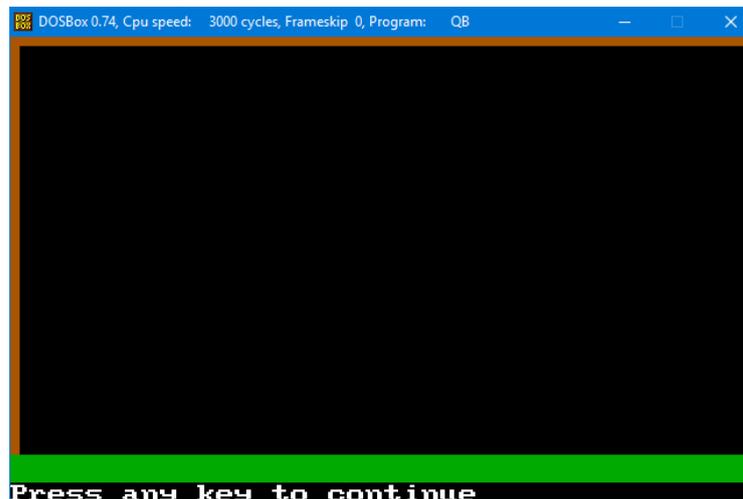
```
LINE (0, 0)-(3, 179), 6, BF
```

QB Resultado:



Como podes ver o **retângulo** foi **pintado** entre o **ponto** mais à **esquerda** do **ecrã** (**0,0**) e o **mais à direita** no **extremo oposto** (**3,179**). O valor **3** será a **espessura** do retângulo. O **179** é o valor imediatamente **anterior** ao 180 onde **começa** o **chão** verde.

QB Agora que sabes fazer **retângulos fecha** o **ecrã** do **lado direito** e em **cima** com a **mesma cor**:



QB Faz agora estas **plataformas sabendo** as **coordenadas**:



QB Vamos agora fazer **círculos**. Começa por **adicionar** a seguinte **linha de código**:

```
CIRCLE (280, 30), 20, 15
```

CIRCLE (coordenada X do centro, coordenada Y do centro), **raio**, **cor**

CIRCLE - desenha um círculo

Dimensão do raio do círculo

Número da cor da linha do círculo

QB A **diferença** aqui é que **não é possível** dizer que o **círculo** deve ser **preenchido**, tal como acontecia com o retângulo. Neste caso ainda é **necessário outra** linha de **código** para **pintar**.

```
PAINT (280, 30), 15, 15
```

PAINT (coordenada X onde começar, coordenada Y onde começar), **cor**, **cor**

PAINT – preenche a cor (tipo o balde do paint)

Nº da cor com que deve preencher

Número da cor que ao encontrar deve parar de preencher

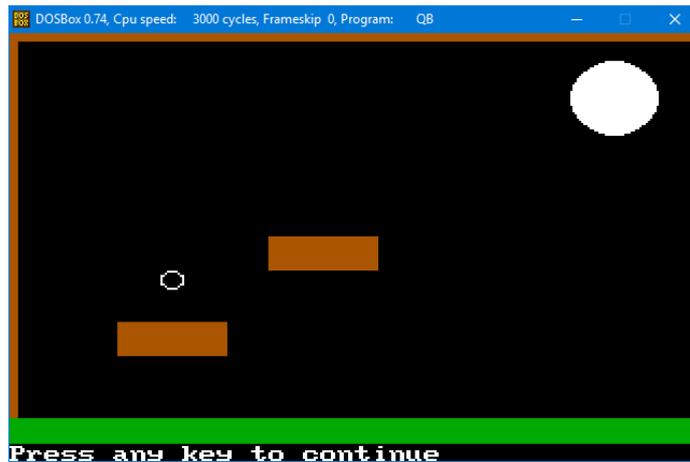
QB Se reparares a **coordenada** do **PAINT** é a **mesma** do centro do **CIRCLE** porque tem de ser uma **coordenada dentro** da **forma** que **pretendes pintar**. Para **facilitar** utilizei a **mesma** apesar de poder ser **outra** qualquer **dentro** do **círculo**.

A **primeira cor** é a cor com que **pretendes preencher** a **forma** e a **segunda** é a **cor de paragem**, isto é qual a cor que ao encontrar deve **parar de preencher**. Neste caso ao encontrar a **linha branca** (cor 15) com que o círculo foi feito **para de preencher**.

QB Outro exemplo:

```
CIRCLE (75, 115), 5, 1
```

Será a **cabeça** de um **boneco** por **cima** da **plataforma**.

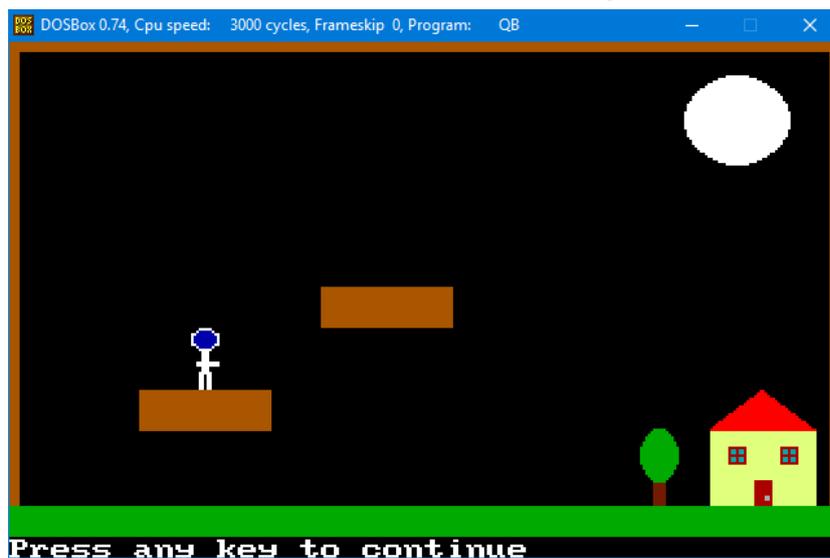


QB Achas que **consegues preencher a cabeça com a cor azul?**

QB Desafio: **faz o corpo do boneco!**



QB Extra: Tenta fazer a casa e a árvore tal como se mostra de seguida:



Dica: a copa da árvore é uma **eclipse** e faz-se da seguinte forma:

Circle(X,Y), raio, cor, , , 1.5



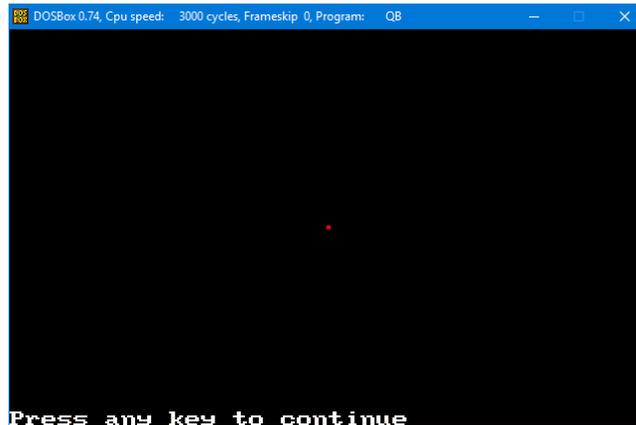
QB Guarda o teu trabalho. Para isso acede ao menu **File→Save As**

Dá-lhe o nome **formas.bas** e garante que estás no **Drive C**.

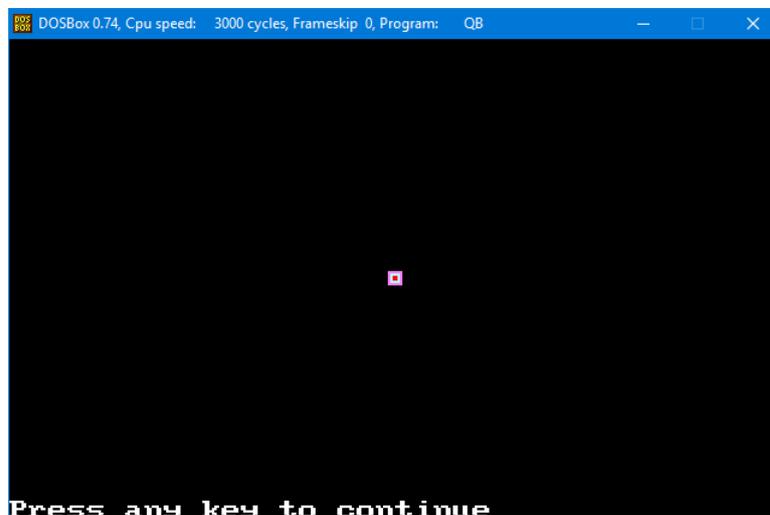
QB Cria um novo programa.

QB Cria uma caixa sem preenchimento desde as coordenadas do centro do ecrã (159,99) até (160,100) com a cor da linha 40.

QB Deves obter a seguinte forma no ecrã:



QB Sabendo que um quadrado seguinte pode se feito retirando 1 valor às coordenadas iniciais (x-1,y-1) e acrescentando 1 valor às coordenadas finais (x+1, y+1), pinta os dois próximos quadrados de cores diferentes do primeiro. Deves obter algo deste género:



QB Agora que já entendeste a lógica de construção de quadrados aplica um ciclo FOR de 160 vezes que imprima 160 quadrados sempre maiores que o anterior e sempre com cores diferentes.

Recordo a sintaxe do FOR:

FOR contador = ??? to ???

O que queres repetir aqui

NEXT contador

QB Deves obter algo do género:



QB Dá um **Step 4** ao teu **FOR** e vê a **diferença**. Atribui outro **step** e volta a ver a **diferença**.

PRO: Achas que **consegues fazer** com **círculos**?

QB Chama o teu **professor** para **avaliar**.