

Code Snippets - Fragmentos de códigos

O GM disponibiliza muitos Snippets para facilitar a nossa vida, vou explicar a função de cada um:

Block - Bloco

Simplemente são duas chaves para separar os códigos. São elas "**Start a block**"=**{** e "**End a block**"=**}**

If,else - Se, senão

Com o **if** você declara uma condição, e com **else** você verifica se ela é falsa. A declaração no **if** só é executada se a condição é verdadeira, ja no **else** só se o contrário do **if**.

```
if (CONDIÇÃO)

{

<DECLARAÇÃO>;

}

else

{

<DECLARAÇÃO>;

};
```

Em jogos de plataforma podemos usá-la para definir a gravidade:

```
if (place_free(x,y + 1))

{

gravity=1

}

else

{

gravity=0

};
```

Loops

Vamos aprender a usar loops em nossos códigos, em vez de repetir um mesmo código podemos fazer em até uma linha! Usaremos os snippets **for,while,do/until e repeat**:

For - Para/Durante/Devido á

O **for** executa a declaração enquanto a condição ainda é ativa:

```
for (INICIALIZAÇÃO; CONDIÇÃO; INCREMENTAÇÃO)
{
    <DECLARAÇÃO>;
};
```

Na inicialização se define uma variável, em condição você define uma condição para a incrementação.

Você pode criar instancias usando o **for**:

```
for (objetos=0; objetos<10; objetos+=1)
{
    instance_create(x,y,objeto)
};
```

A lógica é que esse código será repetido enquanto a variável "objetos" não atingir seu objetivo que é chegar até 10.

While - Enquanto

Simplesmente(ou não) executa uma declaração até que a condição é verdadeira:

```
while (CONDIÇÃO)
{
    <DECLARAÇÃO>;
};
```

Podemos usar **while** para subir uma rampa:

```
while (place_meeting(x,y,rampa))
```

```
{  
  
    y-=1;  
  
};
```

Do e Until - Fazer e Até que

Explicação retirada do tutorial do Allison do fórum Game Maker Brasil:

"As instruções do e until funcionam como um while invertido, se traduzirmos do inglês, a palavra do = fazer e until = até que, sua estrutura basicamente é:"

```
do {  
  
<EXPRESSÃO> }  
  
until  
  
(CONDIÇÃO)  
  
ou  
  
fazer {  
  
<EXPRESSÃO> }  
  
até que  
  
(CONDIÇÃO)
```

Repeat - Repetir

É o loop mais rápido de usar, mas também o menos prático pois não usa uma condição, usa uma expressão.

```
repeat (EXPRESSÃO)  
  
{  
  
    <DECLARAÇÃO>;  
  
};
```

Na expressão se coloca o tanto de vezes que a declaração irá se repetir, simples não é?

Agora saindo dos loops vamos ver uma snippet mais útil que o **if/else**.

Switch - Interruptor

De certo modo esse snippet funciona como um interruptor mesmo. Ele analisa todos os casos de uma variável e suas declarações.

```
switch (EXPRESSÃO)

{

    case EXPRESSÃO: DECLARAÇÃO; break;

    default: DECLARAÇÃO;

}
```

No parenteses do **switch**() você coloca o nome da variável.

Em **case** você põe o caso (ou o estado da variável, se ela é igual a 0,1,2,3... oque quiser), e em declaração o código que será executado se sua variável for igual ao caso definido. Você pode usar quantos casos quiser, mas evite usar iguais. E depois de todos os casos feche com o **break**; para encerrar a declaração ali, se não usar executará os outros códigos dos outros casos.

E no **default** se põe o que acontecerá se a variável tiver um valor que não está nos casos.

With - Com

Simplesmente aplica o código á certo objeto.

```
with (EXPRESSÃO)

{

    <DECLARAÇÃO>;

};
```

Onde o Expressão é um objeto ou id, e declaração é o código que se aplicará a ele. Isso é tudo que precisa saber para usar snippets, soltando a imaginação se pode fazer inumeras coisas com os loops.