Programació amb Scratch BAT_PTD1 8.3

Lloc:	Institut d'Ensenyaments a Distància de les Illes Balears	*Imprès Per:*	PERE MIQUEL ROSSELLO ESPASES
Curs:	Participant 041	*Data:*	dilluns, 15 d'abril 2024, 20:34
Llibre:	Programació amb Scratch BAT_PTD1 8.3		

Taula de continguts

1. Programació amb Scratch

Introducció a Scratch

Scratch és un **entorn de programació** dissenyat per ensenyar a nins, joves i **gent sense experiencia prèvia, els fonaments de la programació d'ordinadors**. Ha estat desenvolupat pel Laboratori de Mitjans de l'Institut Tecnològic de Massachusetts (MIT), a Boston (EUA).

Farem servir la versió en línia, accedint al web scratch.mit.edu amb qualsevol navegador que soporti HTML5 . Scratch és útil per crear animacions, contes interactius i jocs. Nosaltres l'utilitzarem per **programar videojocs senzills**. A les pràctiques següents farem **un joc de frontó, un joc d'obstacles i un joc d'asteroides**.

Ajuda de Scratch: si voleu aprofundir en l'estudi de Scratch, us recomanam que consulteu l'apartat Ajuda del web de Scratch: scratch.mit.edu/help. Hi trobareu una guia d'inici, targetes amb exercicis senzills i videotutorials.

1.1. En què consisteix programar un ordinador?

📌 Un **programa** és un **conjunt d'instruccions, disposades ordenadament, que descriuen com ha de treballar un ordinador**.

📌 Anomenam **programar** a la tasca de **dissenyar i escriure les instruccions que ha de seguir un ordinador**. Un programa s'executa quan s'insereix a la **memòria** d'un ordinador (un xip que **emmagatzema informació**) i el **microprocessador** (el xip més important de l'ordinador, per analogia es diu que fa la **funció de "cervell"**) el llegeix i comença a seguir les instruccions que hi ha a dins del programa.

📌 **Els programes estan formats per tres blocs d'instruccions**: l'entrada, el procés i la sortida.

1. **Bloc d'entrada**: són les instruccions que indiquen a l'ordinador quines dades s'han de captar dels perifèrics d'entrada (el ratolí, el teclat, una pantalla tàctil, etc.).
2. **Bloc de procés**: són les instruccions que descriuen què s'ha de fer amb les dades que s'han captat.
3. **Bloc de sortida**: són les instruccions que descriuen com s'ha de presentar el resultat en els perifèrics de sortida (la pantalla, els altaveus, la impressora, etc.).

1.2. Llenguatges de programació. Què són?

Els ordinadors només entenen les **instruccions** si els les donem com una **seqüència determinada de senyals elèctrics** que entren pels terminals del microprocessador. Aquests senyals **s'emmagatzemen a la memòria** i formen un **conjunt de 0 (no entra corrent) i 1 (entra corrent)**. És el que es coneix com a **codi màquina**.

Un **programa**, per tant, no **és res més que una seqüència de 0 i 1**. El problema és que a les persones ens resulta molt difícil escriure instruccions en forma de seqüències de 0 i 1. Per ajudar-nos amb aquesta feina hi ha programes especialitzats anomenats entorns de programació. Un **entorn de programació permet escriure programes utilitzant un codi específic molt semblant al llenguatge humà**, normalment a l'anglès, anomenat **llenguatge de programació**. Més avall podeu veure un fragment del codi d'un programa, l'anomenat **codi font** , escrit en el **llenguatge de programació C++.** Quan s'acaba d'escriure el codi font, el mateix **entorn de programació** s'encarrega de **traduir-lo al codi màquina** , perquè pugui interpretar-lo l'ordinador. Aquesta traducció d'un llenguatge a un altre s'anomena **compilació** .

```
#include <iostream.h>
using namespace std;
int main() {
   cout <<"Hello world!";
   return 0;
}
```

1.3. Llenguatges de programació. Programació textual i programació visual

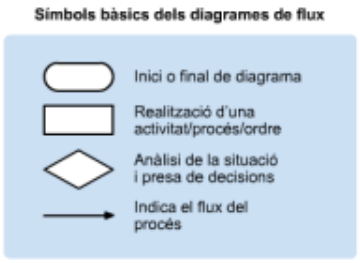
Hi ha molts **llenguatges de programació**: 📌 **Visual basic, C++, Java, JavaScript, PHP, etc**. Aquests llenguatges, que són utilitzats pels programadors professionals, són **llenguatges textuals** . S'escriuen línia a línia de forma similar a com escrivim un text (veure més avall). Cada línia de codi és una instrucció que ha d'executar l'ordinador. Són llenguatges molt potents, amb poc codi es poden escriure moltes instruccions. Malgrat això, es **necessiten moltes hores per aprendre a utilitzar-los**. Per a què persones sense experiència puguin començar a programar de forma senzilla, s'han inventat els 📌 **llenguatges visuals**, com l'**Scratch** . En els llenguatges visuals es programa unint blocs, que són dibuixos que tenen un codi associat.

```
var age = 15;
if ( age > 18 ) {
   document.write("Qualifies for driving");
else {
   document.write("Doesn't qualify for driving");
}
```

1.4. Algorismes

Les instruccions que formen un programa s'han d'escriure seguint un ordre lògic perquè el programa funcioni correctament. Abans de començar a escriure el codi s'ha de **descriure exactament, pas a pas, què volem que faci el programa i en quin ordre**. És el que es coneix com a **algorisme** .

La **representació gràfica d'un algorisme** es fa mitjançant un **diagrama de flux** .



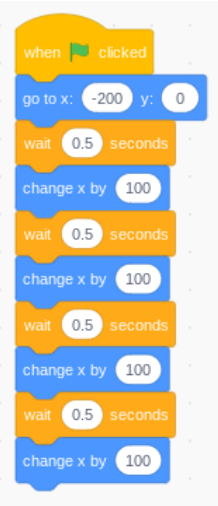
2. Estructures bàsiques de programació

L'ordinador llegeix les instruccions que hi ha en un programa d'una a una i seguint un ordre preestablert, és el que s'anomena el **flux del programa**. Per definir quin flux tindrà un programa que estem escrivint, podem combinar diferents **estructures de programació**. A continuació, veurem les estructures de programació més comunes, són les que utilitzarem per fer jocs en Scratch. **Aprendre a utilitzar aquestes estructures és fonamental per saber programar** .

2.1. Estructura seqüencial

L'estructura bàsica de tots els programes és la seqüencial . Això vol dir que les instruccions s'executen una després de l'altra, seguint l'ordre en el que s'han escrit. L'ordinador llegeix la primera línia que troba (la de dalt) i l'executa. **Cada línia de codi és una instrucció**, una ordre que ha de complir. Després, llegeix la línia següent i l'executa. **Així successivament fins que arriba a l'última línia (la de sota), on el programa s'atura**.

En aquest exemple, el programa s'inicia quan es clica la **bandera verda (línia 1)**. A continuació, la mascota de Scratch se **situa en el punt x=–200, y=0 (línia 2)**. Després, **espera 0,5 segons (línia 3)**. De la mateixa manera, es van llegint les línies següents **fins a arribar a l'última (línia 10)**. Com podeu veure, **les línies 5 a 10 són tres repeticions de les línies 3 i 4**.

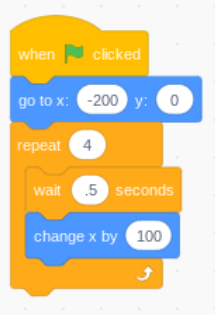


S'està carregant el projecte

S'estan pujant els sons ...

2.2. Estructures de repetició. Bucle repeat

Tot i que l'estructura seqüencial és la base de tots els programes, també són molt freqüents els bucles. Els **bucles són repeticions d'una o diverses instruccions** . S'utilitzen per no haver d'escriure el mateix codi diverses vegades. En aquest exemple s'utilitza el **bucle repeat** per ordenar al programa que **repeteixi les línies de codi 4 i 5**. El **nombre que hi ha després de la paraula "repeat" indica les vegades que el bucle s'ha de repetir** abans de continuar amb la línia següent de la seqüència principal del programa. Se sol utilitzar **la lletra "i" per indicar el nombre de la iteració (repetició)**. Aquest programa és equivalent al de l'exemple anterior, però amb un codi una mica més breu.

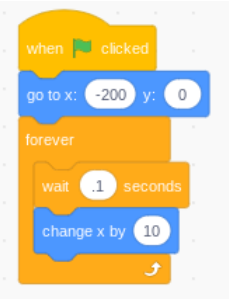


S'està carregant el projecte

Creant blocs ...

2.3. Estructures de repetició. Bucle forever

El **bucle forever** (en anglès, per sempre) és molt semblant al repeat, només que **no s'acaba mai**. En aquesta pàgina, s'ha modificat l'exemple anterior i s'ha substituït el bucle repeat per un bucle forever. S'ha **reduït el temps d'espera (línia 4)** i l'**increment de la x (línia 5)**. El resultat és que, en clicar la **bandera verda**, la mascota de Scratch va a la **posició x=–200, y=0** i després s'inicia un **bucle sense fi que el fa avançar poc a poc**. Seguint aquest programa, el gat avançaria indefinidament cap a la dreta. S'atura quan arriba a l'extrem dret perquè en Scratch els objectes no poden sortir de l'escenari.



S'està carregant el projecte

Creant blocs ...

Operadors

Abans de continuar, fem un parèntesi per parlar dels operadors. Són elements de programació que permeten interrelacionar dades. Aquí podeu veure els més comuns:

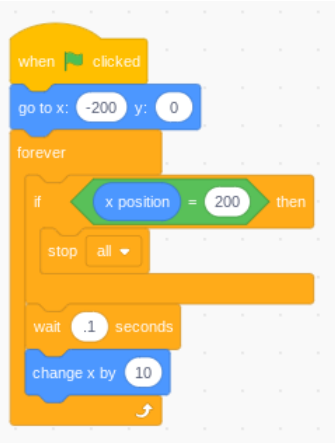
- **Suma, resta, multiplicació i divisió.** 

- **Condició "menor que", "igual que" i "més gran que".** 
- **Operador lògic AND:** torna True si els dos operands (els espais hexagonals) són certs, i False si com a mínim un dels dos operands és fals. 
- **Operador lògic OR:** torna True si com a mínim un dels dos operands és cert. 
- **Nombre a l'atzar:** genera un nombre aleatori comprès entre els dos valors que indiquem. 

2.4. Estructures condicionals. If... then...

Moltes vegades un programa ha de **prendre una decisió en funció d'una condició externa** . Una cosa així com: "Si passa això, llavors fes això altre".

Això s'implementa en els llenguatges de programació amb l'expressió anglesa "**If... then...**". Un exemple a la vida diària és un pas de vianants amb semàfor. La nostra actuació serà: "Si el semàfor està en vermell, llavors m'atur". En un programa quedaria: "If semàfor = vermell then aturar-se". En aquest exemple s'ha **afegit una estructura if... then... al programa anterior (línies 4 i 5)**. El resultat és que **el moix s'atura quan arriba a la posició x=200**, en comptes de continuar a la dreta indefinidament, com feia abans.

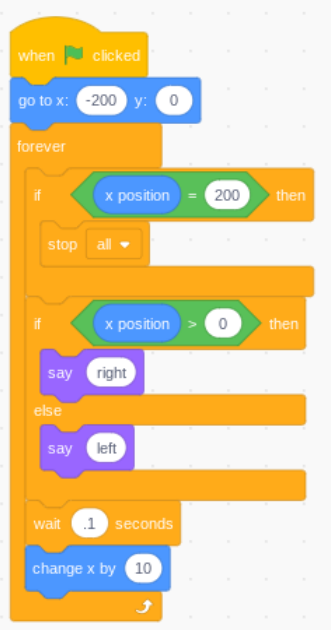


S'està carregant el projecte

S'estan pujant els sons ...

2.5. Estructures condicionals. If... then... else...

A vegades, no n'hi ha prou amb especificar quina decisió s'ha de prendre si es dóna una condició i **també interessa determinar quina altra decisió prendre en el cas que la condició no es compleixi**. Una cosa així com: "Si passa això, llavors fes allò, si no, fes això altre". En programació s'implementa amb l'expressió anglesa "**If... then... else...**". Seguint amb l'exemple del pas de vianants, la nostra actuació serà: "Si el semàfor està en vermell, llavors m'atur, si no, pas". En aquest exemple, s'ha **afegit una estructura if... then... else... al programa anterior (línies 6 a 9)**. Aquesta fa que **el moix digui right quan la seva posició x és més gran que 0 i left quan no es dóna aquesta condició**.



S'està carregant el projecte

Carregant les extensions ...