

PENSAMENT COMPUTACIONAL A L'AULA AMB SCRATCH

PROGRAMACIÓ VISUAL



PENSAMENT COMPUTACIONAL A L'AULA AMB SCRATCH

Amb el **pensament computacional** l'alumnat desenvolupa una manera diferent de resoldre i formular problemes quotidians. Aquesta habilitat pròpia a la Matemàtica o la Informàtica es pot adquirir jugant al pati, amb jocs de taula... i també amb tecnologia.

En aquest curs donarem recursos al professorat d'Educació Secundària per a construir jocs fàcils, divertits i motivadors, que desenvolupen el pensament computacional amb un **entorn de programació visual**.

INFORMACIÓ DEL CURS

Pensament computacional a l'aula amb Scratch és un curs en línia de 30h coordinat pel CEFIRE CTEM adreçat a professorat d'Educació Secundària.

Aquest curs inicia el professorat en l'ús dels entorns de **programació visual** a l'aula. És el primer pas en la línia de formació de pensament computacional per a Secundària i es podrà aplicar a la programació de dispositius mòbils, robots o videojocs, en cursos posteriors.



PREGUNTES FREQUENTS

- **Qui pot matricular-se?** És un curs adreçat a professorat d'Educació Secundària de qualsevol especialitat.
- **Necessite saber programar?** NO, una bona forma d'introduir-se són els entorns de programació visual.
- **Fomenta la creativitat?** Sí, els problemes es poden resoldre fàcilment amb moltes solucions creatives

Curs tutoritzat per Santiago Ortega

INSCRIPCIÓ AL WEB DEL CEFIRE CTEM

<http://mestreacasa.gva.es/web/cefireambitctm/inscripcio>

Bloc 1: Introducció al pensament computacional

Aquest document forma part del curs [Pensament computacional a l'aula amb Scratch](#) del [CEFIRE CTEM](#).

Aquesta obra està subjecta a la llicència Reconeixement-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Per veure una còpia d'aquesta llicència, visiteu

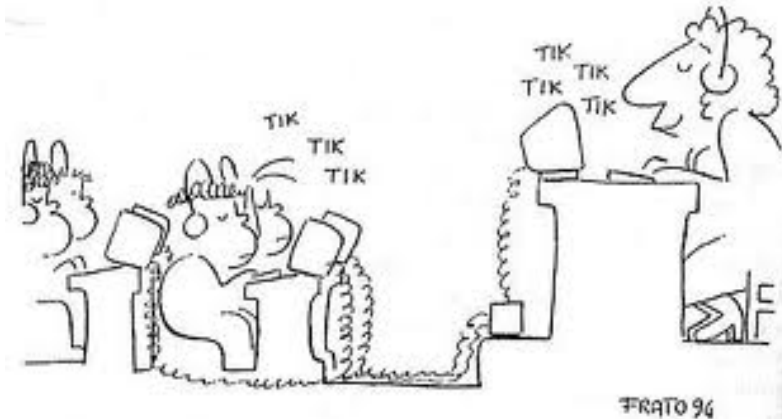
<http://creativecommons.org/licenses/by-nc-sa/4.0/>.

Autoria: Santiago Ortega, Òscar López Jiménez i Javier García Benlloch.



1. INTRODUCCIÓ

Si després de veure aquesta vinyeta de Frato penses que un aula on es treballa el pensament computacional és una classe de xics, col·locats en fila connectats al seu ordinador, amb altaveus en les orelles i abstrerts en la pantalla, anem a desmuntar-te alguns d'aquests falsos mites.



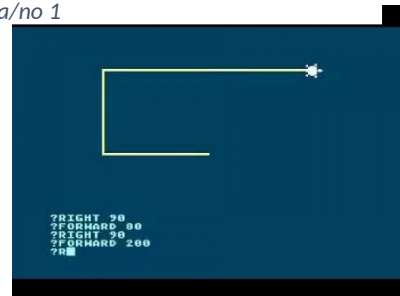
Comencem per una definició!

2. QUÈ ÉS EL PENSAMENT COMPUTACIONAL?

El pensament computacional com a concepte apareix per primera vegada amb les idees de Seymour

Papert, les quals, basant-se en les idees constructivistes de l'aprenentatge desenrotllades per Piaget radicaven a utilitzar els ordinadors per a integrar l'alumne en el mateix centre del que es pretenia aprendre, per a aconseguir així, un aprenentatge verdaderament significatiu. Papert, va plasmar les seues idees en un llibre anomenat "Mindstorm Children, Computers and Powerful Ideas"; assentant així les bases del pensament computacional. Posteriorment, per a portar les seues idees a un pla més pragmàtic va desenvolupar un llenguatge de programació anomenat Logo (veure imatge a la dreta) . El qual, podríem dir que seria el besaiio de l'actual Scratch, i encara que va estar carregat de bones intencions, no arribé a tindre èxit entre la comunitat educativa.

<http://maximaonline.com.ar/multimedia/no 1>



No obstant això, no va ser Papert qui realment va encunyar el terme "Pensament computacional" sinó que va ser Jeannette Wing, la qual, al març de 2006, va publicar un article desengranant els múltiples beneficis que podia portar la implantació d'este mètode de resolució de problemes en les classes escolars.



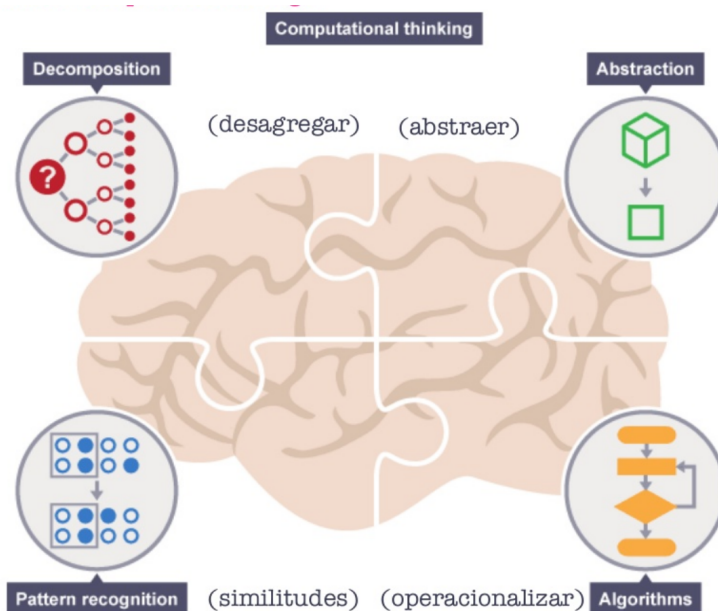
"Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."

"Pensament computacional: procés mental utilitzat per a formular problemes i les seues solucions de manera que les solucions es plasmen de manera adequada per a ser dutes a terme per qualsevol agent que processe informació"

Aquesta definició, a partir de la qual anem a treballar, implica que el pensament computacional és un procés de pensament independent de la tecnologia. Mitjançant el pensament computacional som capaços de dissenyar solucions per a ser executades per un ordinador, una tableta, un humà, o qualsevol combinació d'ells.

3. QUÈ APORTA?

La introducció del pensament computacional en les aules permet dotar a l'alumnat d'una manera de resoldre problemes i enfrontar-se a les tasques quotidianes amb un nombre major d'habilitats. Aquestes habilitats van a ser clau a l'hora de tenir una societat PRODUCTORA de tecnologia en lloc de només CONSUMIDORA.



<https://www.slideshare.net/cristobalcobo/cultura-maker-pensando-en-el-pensamiento-computacional-coding-diy> // CC BY-SA 4.0



Dins d'aquestes habilitats anem a destacar la descomposició, l'abstracció, el pensament algorítmic i els patrons, encara que com veurem en els pròxims temes les habilitats que es poden treballar són encara majors com per exemple la lògica.

És important dir, que no sempre es treballen totes les habilitats, dependrà en part del tipus de problema a què ens enfrontem, i de les nostres pròpies capacitats. Per exemple, poguera ser que una persona determinada puga identificar la majoria dels patrons d'un problema concret sense necessitat de descompondre-ho. No obstant això, en major o menor mesura, l'últim pas serà sempre la creació d'un algoritme que resolga el problema.

3.1 L'abstracció

En un plànol de metre, sabem com arribar d'una estació a una altra seguint una línia de color. No ens importa l'arquitectura dels edificis, ni la temperatura, ni si hi haurà molta gent. Ens oblidem de tot açò i ens centrem en l'essencial. Açò és l'abstracció. En el plànol es representa la informació necessària perquè el viatger puga anar d'un lloc a un altre eliminant tota la informació no rellevant.



http://www.levante-emv.com/especiales/transportes-valencia/2012/09/plano-zonal-metrovalencia-n66_12_1793.html

L'abstracció és el procés pel qual anem a eliminar els aspectes innecessaris deixant solament els importants per a dur a terme la resolució d'un problema.

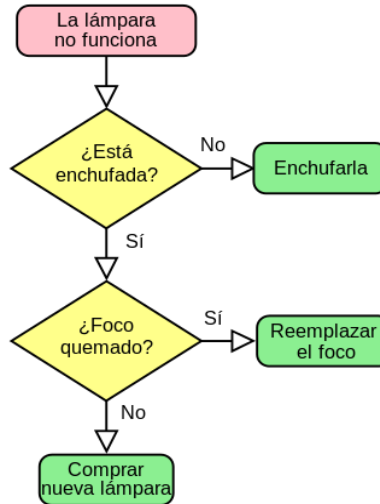
3.2 El pensament algorítmic

L'abstracció és el procés pel qual anem a eliminar els aspectes innecessaris deixant solament els importants per a dur a terme la resolució d'un problema. La Real Acadèmia Espanyola (RAE) defineix algorisme com "un conjunt ordenat i finit d'operacions que permet trobar la solució d'un problema".

Per exemple, podem usar un algorisme per a trobar la ruta més curta entre dues ubicacions en un mapa; el mateix algorisme es pot usar per a qualsevol parell de punts d'inici i finalització, per la qual cosa la solució depèn de l'entrada a l'algorisme.

Si coneixem l'algorisme per a resoldre un problema, llavors podem resoldre aqueix problema fàcilment, quan vulguem, sense haver de pensar! Solament hem de seguir els passos.

Un exemple gràfic d'algorisme quotidià.



upload.wikimedia.org/wikipedia/commons/thumb/b/bd/LampFlowchart-es.svg/330px-LampFlowchart-es.svg.png

Cridem pensament algorítmic al procés de creació d'algorismes, és a dir, una forma d'arribar a una solució definint els seus passos.

El pensament algorítmic ajuda a l'alumnat a resoldre problemes de la vida quotidiana o acadèmics d'una manera estructurada i creativa.

Un exemple de pensament algorítmic portat al terreny de l'humor és el vídeo "Sheldon Cooper l'algorisme de l'amistat" de la sèrie Big Bang Theory"

<https://youtu.be/h3z3hdbl5qu>

3.3 La descomposició

La descomposició consisteix en el procés de dividir un problema complex o sistema en parts més xicotetes que són més senzilles i més fàcils d'entendre. Cada part es pot examinar i resoldre, o dissenyar individualment, ja que són més fàcils de treballar.

La descomposició és una habilitat important per a crear algorismes i processos que es poden implementar en un dispositiu informàtic, perquè els ordinadors necessiten instruccions molt específiques. Necessiten que els diguen cadascun dels xicotets passos que han de seguir per a fer les coses.

Per exemple, la tasca general de fer unes galetes de mantega es pot descompondre en diverses tasques més xicotetes, cadascuna de les quals es pot realitzar fàcilment.



- Tamisem la farina amb la sal i el llevat. Reservem.
- Batem el sucre amb la mantega fins que es forme una crema blanca.
- Afegim l'ou i l'essència. Batem fins que s'incorpore.
- Ara anem afegint a poc a poc la mescla de farina, a cullerades, deixant que s'integre la primera abans de tirar la següent cullerada.
- La massa estarà llesta quan resulte una massa blana però no apegalosa.
- Amb una espàtula, arrepleguem les galetes i les posem en una safata de forn. Estirem entre dos papers de forn. Hem de deixar la massa d'uns 5-6 mm de grossor. Tallem amb els motles.
- Fornejar uns 5- 10 min, o fins que comencen a torrar-se les vores.
- Traiem del forn, deixem refredar uns minuts i passem a una reixeta perquè refreden del tot.

Recepta obtinguda de <http://unaenfermeraenlacocina.blogspot.com.es/2013/12/galletas-de-mantequilla-para-decorar.html>

3.4 Els patrons. Generalitzacions

La **generalització** s'associa amb la identificació de patrons, similituds. És una forma de resoldre nous problemes a partir de la base de problemes ja resolts anteriorment.



Què tenen en comú aquestes imatges?

Observant les imatges veiem que tenen en comú que totes giren al voltant d'un eix. Aquest seria el patró.

Quan pensem en els problemes, podem reconèixer similituds entre ells i que es poden resoldre de manera similar. Açò es diu coincidència de patrons, i és una cosa que fem naturalment tot el temps en la nostra vida diària.

3.5 Exemple

Passem ara a explicar un **exemple** senzill del nostre dia a dia, **bullir pasta**.

Amb la **descomposició**, **separem** un problema major en problemes més xicotets, de manera que la resolució de cada un d'ells per separat acabe portant a la resolució del major. En el nostre exemple, separarem el problema en problemes més xicotets:

1. Preparació.
2. Ebullició.
3. Servir.

Els **patrons**, ens permeten identificar aquells aspectes que tenen en comú les diferents parts del problema. Tant en la seua divisió de problemes més xicotets, com en el problema en el seu conjunt. En el nostre exemple, **identificarem** aquells aspectes que marcaran cada xicotet problema:

1. Preparació.
 - a. Grandària adequada del recipient
 - b. Quantitat d'aigua
 - c. Sal
2. Ebullició.
 - a. Temperatura del foc
 - b. Temps d'ebullició
3. Servir.
 - a. Quantitat de racions

Amb l'**abstracció**, eliminem del problema tots aquells aspectes que resulten irrelevants per a la resolució del mateix, quedant-nos únicament amb l'essencial.

Potser no ens adonem, que se'ns ha pogut oblidar un patró determinat, no hi ha problema perquè, per a tornar arrere i afegir-ho. Vegem com en el nostre exemple hem considerat superflu el clima, la temperatura a què està la pasta, ... Però ens hem adonat de que no hem considerat la quantitat de pasta, afegim-ho doncs:

1. Preparació.
 - a. Grandària adequada del recipient
 - b. Quantitat de pasta
 - c. Quantitat d'aigua
 - d. Sal
2. Ebullició.
 - a. Temperatura del foc
 - b. Temps d'ebullició
3. Servir.
 - a. Quantitat de racions

En **crear l'algoritme**, presentarem una sèrie d'instruccions concretes que, partint dels nostres passos anteriors, descomponen el problema en xicotets passos concrets, els quals se centren en l'essencial del mateix, per a poder resoldre-ho de la manera més eficient. En fer-ho, ens plantejarem preguntes per a determinar si hem de passar a la fase següent de la resolució del problema o de si al contrari hem de tornar a realitzar la fase anterior.

Hi ha diverses formes de presentar un algoritme, una de les més comunes, intuïtives i gràfiques és la de l'ús d'un diagrama de flux. És per això que s'ha decidit presentar-lo en aquest format. No obstant això, més avant mostrarem diverses formes de plasmar gràficament un algoritme.

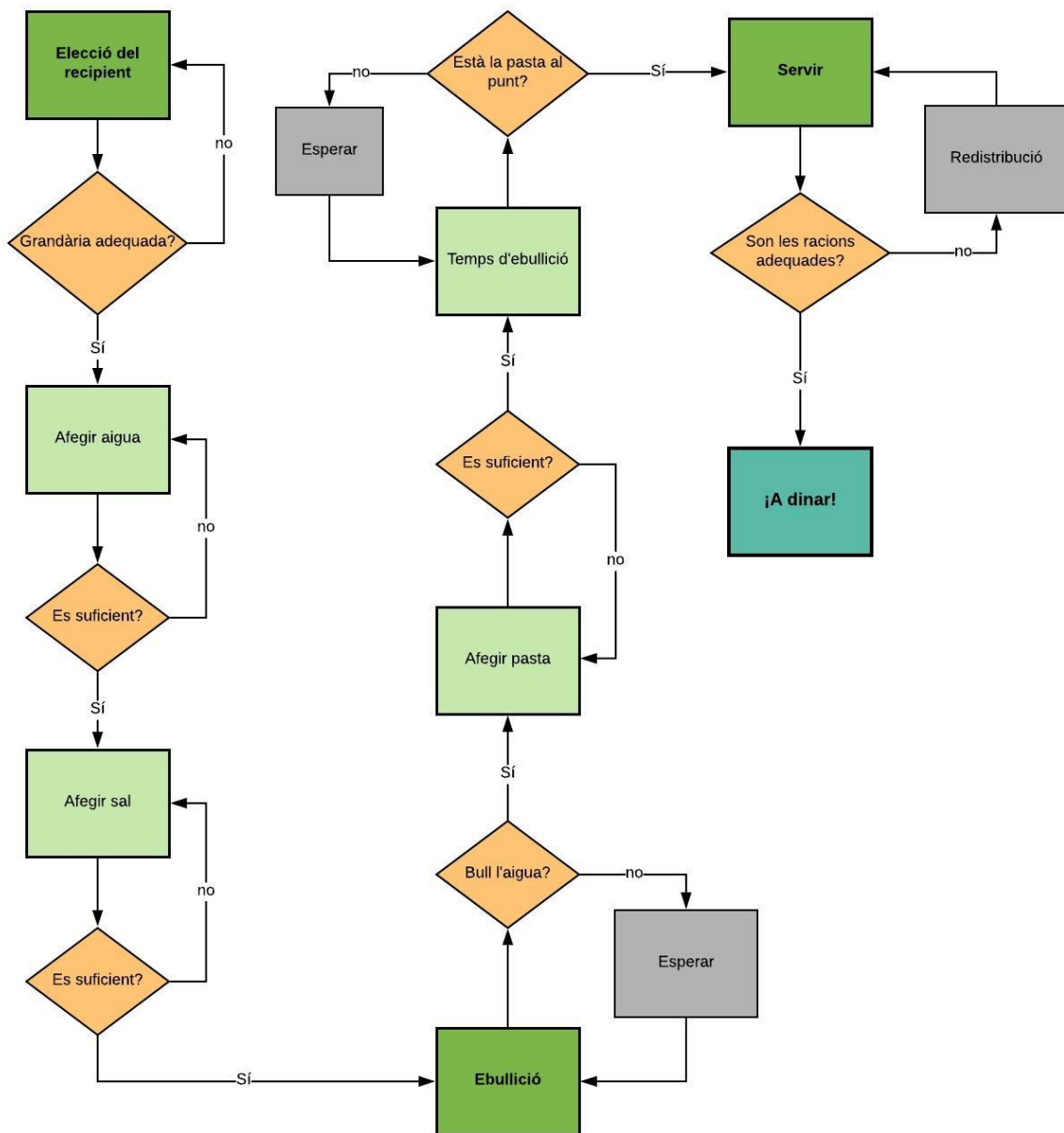


Diagrama de flux creat amb la versió gratuïta de Lucidchart (www.lucidchart.com)

Òbviament, este algoritme pot ser més complex incloent més passos i opcions, o més senzill, obviant alguns passos. Depenent del grau d'exactitud requerit, l'algoritme haurà de ser més o menys precís.

Si esteu interessats en conèixer més sobre com funcionen i quines normes tenen el diagrames de flux, en el següent enllaç podreu trobar-ne més informació:

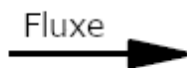
https://www.iramonet.com/sites/default/files/adjuntos/diagramas_flujo_jrf_v2013.pdf

4. REPRESENTACIÓ D'ALGORITMES

Amb la creació d'un algoritme, es fa patent la necessitat de plasmar una idea abstracta en una representació objectiva que guie al lector a través del procés de resolució del problema. Per a això, hi ha multitud de representacions que serveixen com un mapa que represente els diferents camins que es poden prendre. Explicarem perquè algunes d'estes representacions:

4.1. Diagrama de flux.

El diagrama de flux també conegut com fluxograma o diagrama d'activitats, representa per mitjà de figures geomètriques codificades, fletxes i text el procés que s'ha de seguir per a la resolució d'un problema concret. Facilitant així diversos camins que poden ser circulars (quan tornen al punt previ) o unidireccionals (quan comencen un procés).

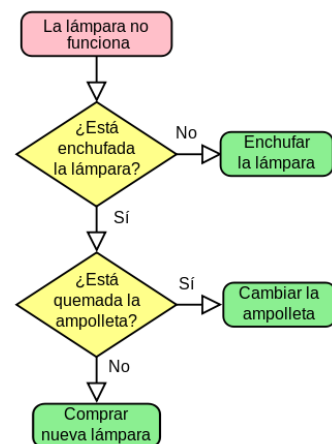


Com a símbols bàsics utilitzats tenim:

El **rectangle** o caixa: representa una activitat, operació o tasca.

Rombe: representa una condició, pregunta o decisió plantejada. Sempre s'ha de respondre únicament amb "Sí" o "No".

Línia amb fletxa: Representa el sentit de les activitats o flux.



És important conèixer, que en els diagrames de flux existeix **NOMÉS** un punt d'INICI i **NOMÉS** un punt FINAL. A més d'esta hi ha altres normes tècniques que normalitzen el seu ús.

Com és obvi, els diagrames de flux poden ser molt més complexos, amb símbols més complexos, auxiliars... Però ja que no és el nostre camp d'estudi, ens detindrem ací.

https://es.wikipedia.org/wiki/Diagrama_d_1

4.2. Pseudocodi

El pseudocodi és una **descripció INFORMAL** i resumida, d'una sèrie **d'instruccions**, del principi operatiu d'un programa o d'un algoritme. Vegem-ho amb un exemple com cridar per un telèfon mòbil:

INICI

Desbloquejar mòbil
Entrar en aplicació cridades
Marcar número desitjat
Pulsar la tecla cridada
Esperar contestació
Parlar
Pulsar la tecla finalització de cridada
Bloquejar mòbil

FI

Amb este mètode podem esbossar de forma **ràpida i senzilla** les diferents ordes que desitgem que posteriorment es plasmen en una altra representació, a més de treballar amb els alumnes la utilització del llenguatge com a instruccions.

4.3. Programació amb codi

La programació amb codi fa referència a aquella que utilitza un llenguatge o codi propi per a dur a terme una sèrie d'accions en un programa determinat. Es fa necessari doncs, conèixer el llenguatge del programa per a poder escriure les ordes necessàries. Hi ha multitud de llenguatges informàtics a l'hora de programar, alguns exemples són: JAVA, C, C++, Python y C#.

4.4. Programació visual

La programació visual és aquella que **simplifica** enormement la **creació de l'algoritme** de resolució, ja que la codificació del codi ja ha sigut prèviament creada i codificada, en una sèrie d'elements visuals que l'editor simplement deu d'anar seleccionant en l'orde que el desitge, per a crear el seu codi. Generalment, estos elements visuals estan situats dins d'un **IDE** (Entorn integrat de desenvolupament) , prou amigable, que ajuda a seleccionar-los, dividint-los per categories, pintant-los de diferents colors, atorgant-los diferents formes segons la seua aplicació...



En el cas de **Scratch**, estos elements visuals es coneixen amb el nom de "**blocs**" i fa referència a la seua forma.

A més de Scratch, hi ha altres plataformes de programació molt utilitzades en educació, que fan ús dels blocs com:

- App Inventor
- Code.org
- Makeblock

Els llenguatges de programació són la via per a comunicar-nos amb els ordinadors, amb ells, podem construir una sèrie d'instruccions, que, executats en un orde o un altre prèviament determinats, donaran forma als nostres jocs, animacions, cançons...

Estos llenguatges donen una sèrie d'ordes, les quals acaben conformant el codi font d'un programa. Per tant, este codi font serà el llistat d'accions ordenades que el nostre programa executarà a l'iniciar-se. En concret, en Scratch, estes ordes prenen la forma de blocs, que units d'una manera o d'una altra per l'usuari/programador, donaran forma al nostre joc o programa.




Generalment, programar és, òbviament, un procés molt més complex, per a fer-ho solem necessitar:

1. **Un editor de codi.** “Un editor de codi font és un editor de text dissenyat específicament per a editar el codi font de programes informàtics.”
https://es.wikipedia.org/wiki/Editor_de_c%C3%B3digo_fuente
2. **Un compilador.** “Un compilador és un programa informàtic que tradueix un programa que ha sigut escrit en un llenguatge de programació a un llenguatge comú”, de manera que “reuneix diversos elements o fragments en una mateixa unitat” i ho transforma en un programa executable.
https://es.wikipedia.org/wiki/Editor_de_c%C3%B3digo_fuente

No obstant això, una altra opció que existeix a l'hora de programar és que estos dos elements, l'editor i el compilador, s'ajunten en un sol, en la qual cosa es coneix com IDE, “Integrated Development Environment”, o en castellà Entorn Integrat de Desenvolupament. Este tipus d'opció simplifica molt les coses a l'hora de programar, i com no podia ser d'una altra manera, Scratch és un IDE molt intuïtiu.

5. RELACIÓ AMB EL CURRÍCULUM

L'Institut Nacional de Tecnologies Educatives i de Formació del professorat (INTEF) va presentar en 2015 l'informe [Computhink](#), el pensament computacional en l'ensenyament obligatori. El treball realitzat en aquest informe, la lectura del qual recomanem, presta especial atenció a la manera en el qual s'hauria d'incloure el pensament computacional en l'ensenyament obligatori per a aconseguir:

- El desenvolupament d'habilitats de Pensament Computacional en l'alumnat perquè puguem pensar de manera diferent, expressar-
 se  a 

través d'una varietat de mitjans, resoldre problemes del món real i analitzar temes quotidians des d'una perspectiva diferent.

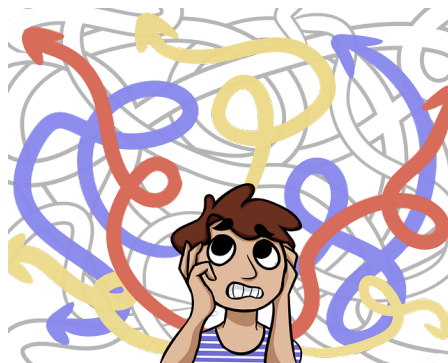
- El foment del Pensament Computacional per a impulsar el creixement econòmic, cobrir llocs de treball TIC i preparar-se per a futures ocupacions.

A Espanya ja hi ha comunitats autònomes pioneres que han introduït en el seu [currículum de primària](#) i secundària el pensament computacional com la Comunitat de Navarra que fins i tot per al curs escolar 2017-2018 ho inclouran en la prova diagnòstica.

Però la major implantació s'ha realitzat a [Regne Unit](#), de fet està suposant un model per a la resta d'Europa.

En el currículum de la Comunitat valenciana no hi ha una menció expressa al pensament computacional com a part del contingut de les diferents matèries que ho conformen. Però sí, podem fer referència a ell en les nostres programacions en diferents matèries que anomenen en els seus continguts o justificació aspectes com: **el desenrotllament del pensament i raonament, la resolució de problemes, el pensament alternatiu, el pensament mitjans-fi, el pensament abstracte o el pensament de perspectiva** entre altres. Sens dubte, estos aspectes apareixen amb especial rellevància en matèries com a **Tecnologia, Informàtica o Tecnologies de la Informació i la Comunicació**, encara que també es pot aplicar a altres matèries com a Matemàtiques, Física i Química, ...

6. RESUM



La formació del pensament computacional ajuda als estudiants a **aprendre a descriure** un **problema**, **identificar** quins són els **detalls importants** que necessiten per a resoldre aquest problema, **dividir-ho** en xicotets **passos lògics** (perquè puguem crear un procés que solucione el problema) i després **avaluar** aquest **procés**.

Aquestes **habilitats** són **transferibles** a qualsevol altra àrea de currículum i, per descomptat, per a desenvolupar sistemes digitals i resoldre problemes usant les capacitats dels ordinadors.

Bloc 2: La web de Scratch. Primer projecte.

Aquest document forma part del curs [Pensament computacional a l'aula amb Scratch](#) del [CEFIRE CTEM](#).

Aquesta obra està subjecta a la llicència Reconeixement-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Per veure una còpia d'aquesta llicència, visiteu

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.

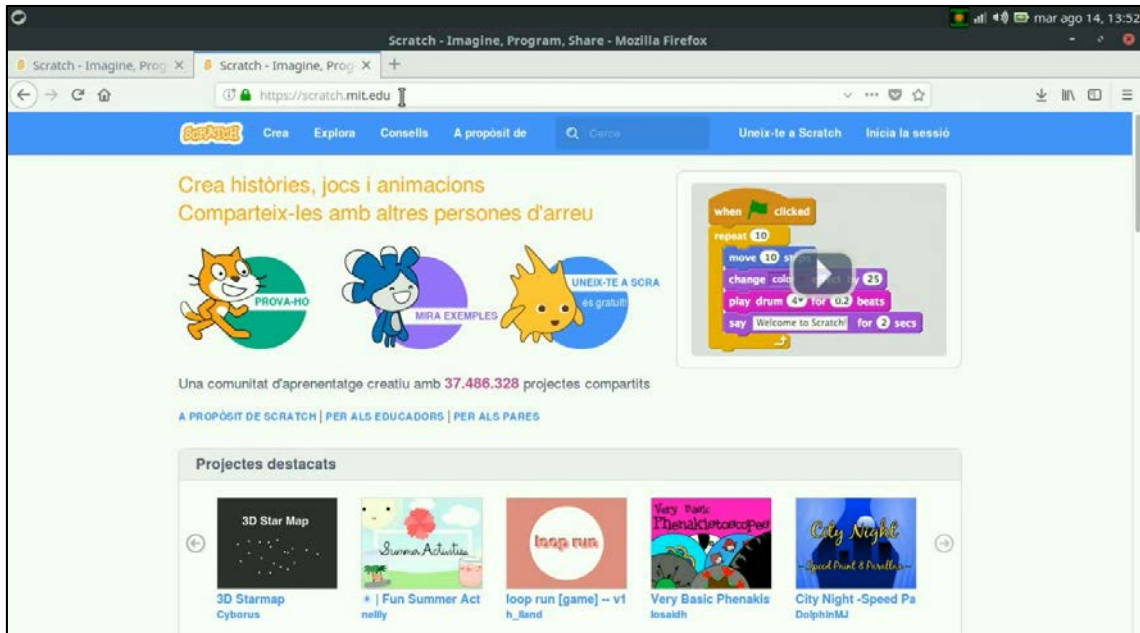
Autoria: Santiago Ortega



PÀGINA PRINCIPAL DE SCRATCH

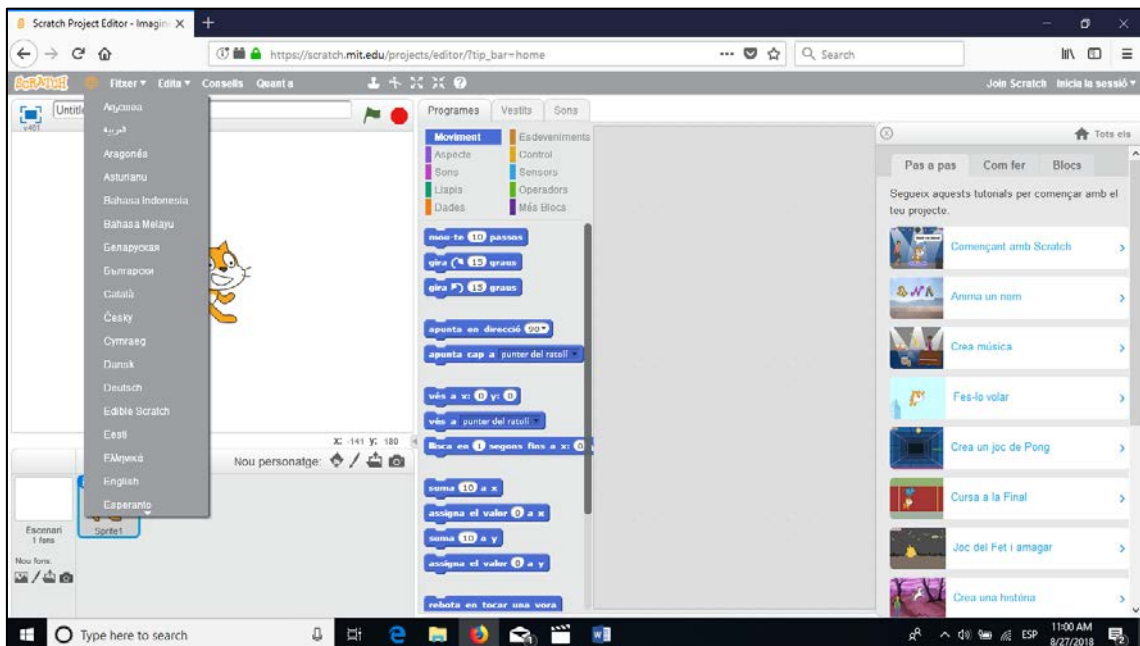
Podem accedir a la pàgina principal de Scratch de dos maneres:

1. Escrivint al nostre buscador: "Scratch" i fent clic al primer enllaç
2. Escrivint la direcció: <https://scratch.mit.edu/> a la barra de direccions del nostre navegador.

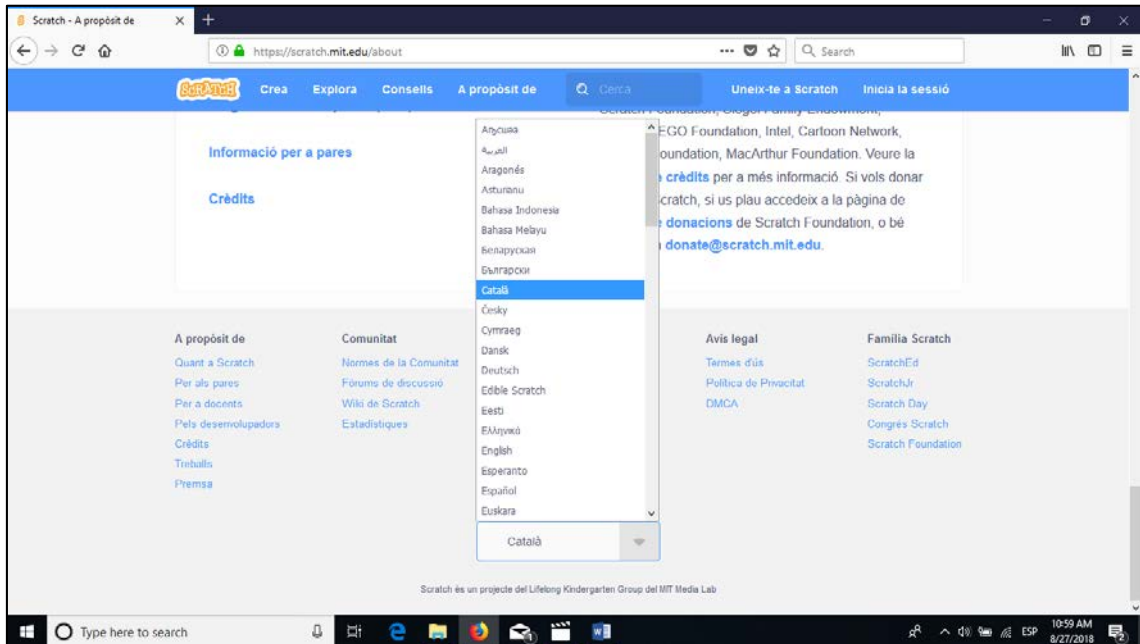


Per canviar el idioma predeterminat de la pàgina de Scratch podem triar dos opcions:

1. Des de la pestanya de "Crea".
 - a. Fem clic en el globus del mon i triem el idioma pertinent.
 - b. Tornem a la pàgina inicial.



2. Des de la pestanya “A propòsit de”.
 - a. Anem al final de la pàgina i canviem l’idioma des de el desplegable.



Seccions

La barra blava de Scratch:

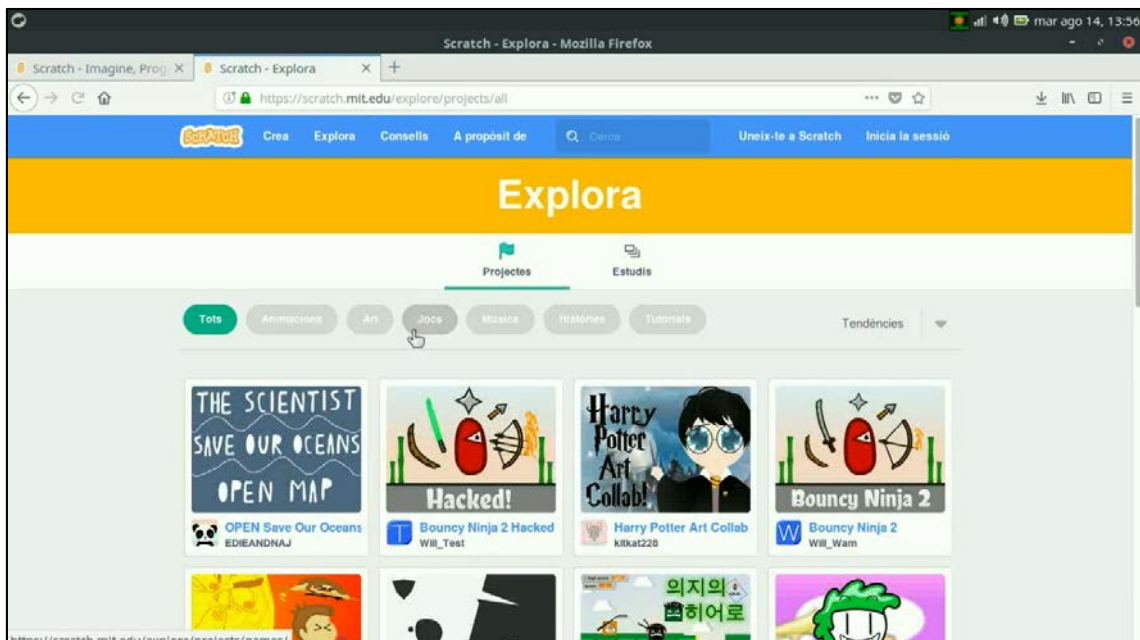


- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

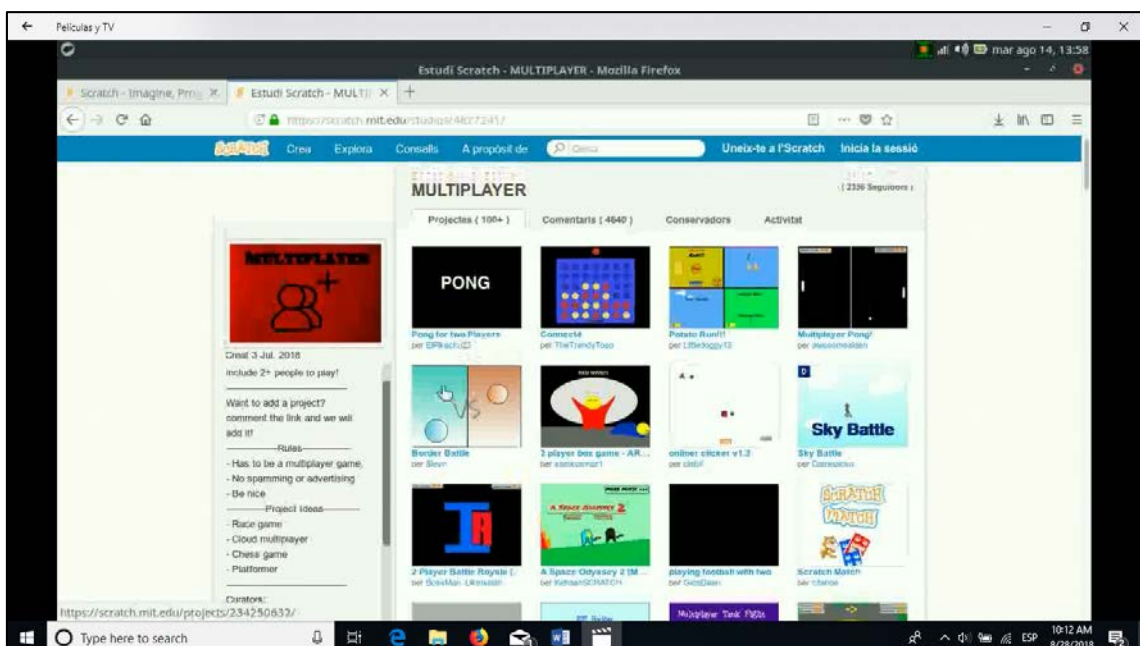
1. El botó “Scratch”: retorna a la pàgina inicial.
2. La pestanya de “Crea”: ens porta al IDE de Scratch per crear nous projectes.
3. La pestanya d’ “Explora”: ens permet navegar a través de projectes i estudis de Scratch creats per altres usuaris.
4. La pestanya de “Consells”: ens facilita fitxes, guies didàctiques, tutorials, etc...
5. La pestanya “A propòsit de”: ens lliura informació tant per a pares i docents, com per a usuaris nous. Així com dades e informació sobre l’ús de Scratch arreu del món.
6. La barra de “Cerca”: ens permet buscar projectes creats per altres usuaris, ja siga per nom o per descripció del projecte.
7. Uneix-te a Scratch: podem crear un usuari nou.
8. Inicia la sessió: per iniciar sessió amb un usuari ja creat.

Explora.

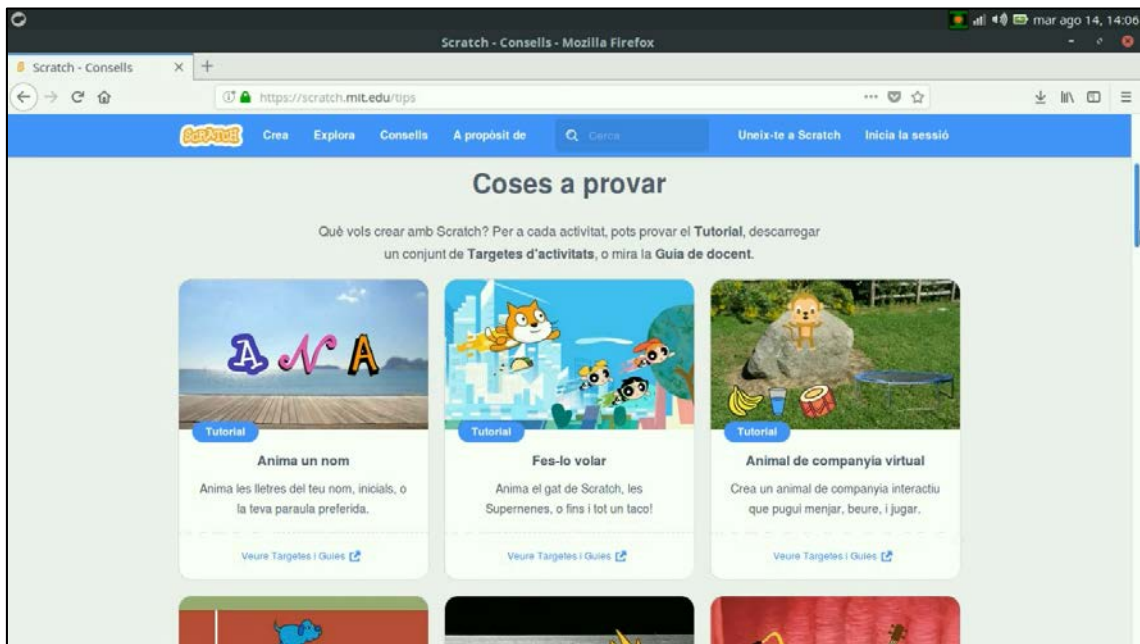
A la pestanya d'“explora” podem trobar una sèrie de projectes i estudis creats per altres usuaris dins de Scratch.



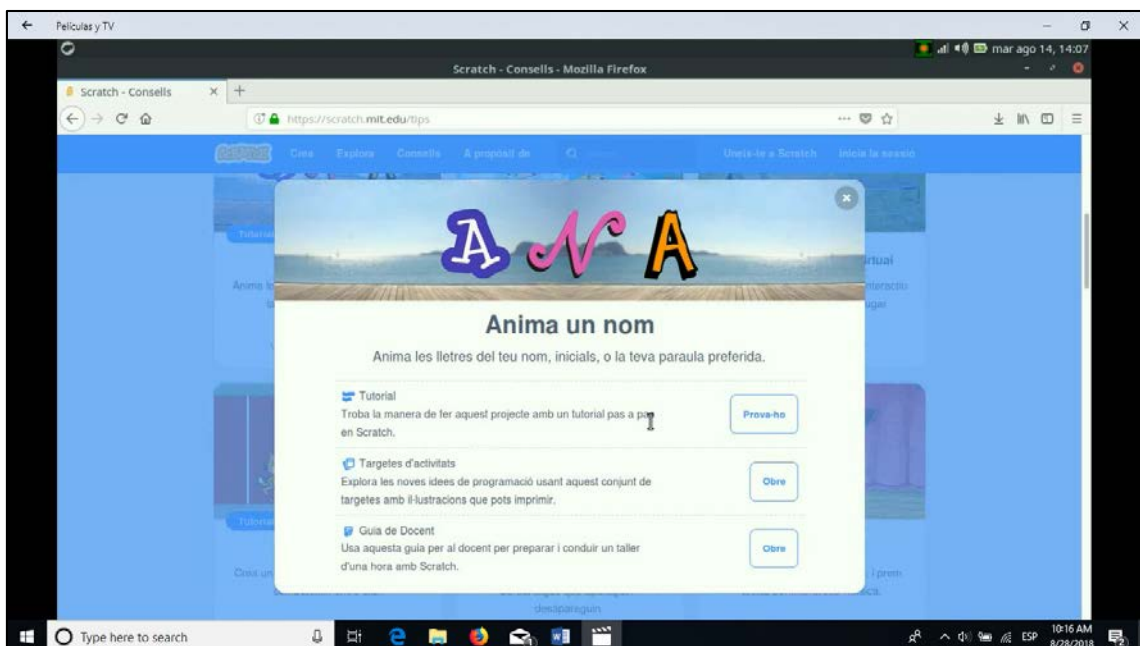
La diferència principal entre projectes i estudis és que els **projectes** es realitzen **sense seguir cap tipus de regla**, és a dir, de manera totalment **lliure**, i en els **estudis** els projectes han de realitzar-se **seguint** una sèrie de **pautes o normes** creades per un usuari o usuaris concrets. A més a més, als estudis apareixen dos tipus d'usuaris nous: els “**curators**”; que ajuden a millorar els projectes ja creats, i els **managers**; que decideixen entre altres coses quins projectes son dignes d'entrar en el estudi.



Consells



A la pàgina de consells podem trobar guies didàctiques, fitxes per als alumnes, tutorials, targetes d'activitats,... que ens ajudaran a gaudir d'una experiència més enriquidora, des de el punt de vista educatiu, de Scratch.

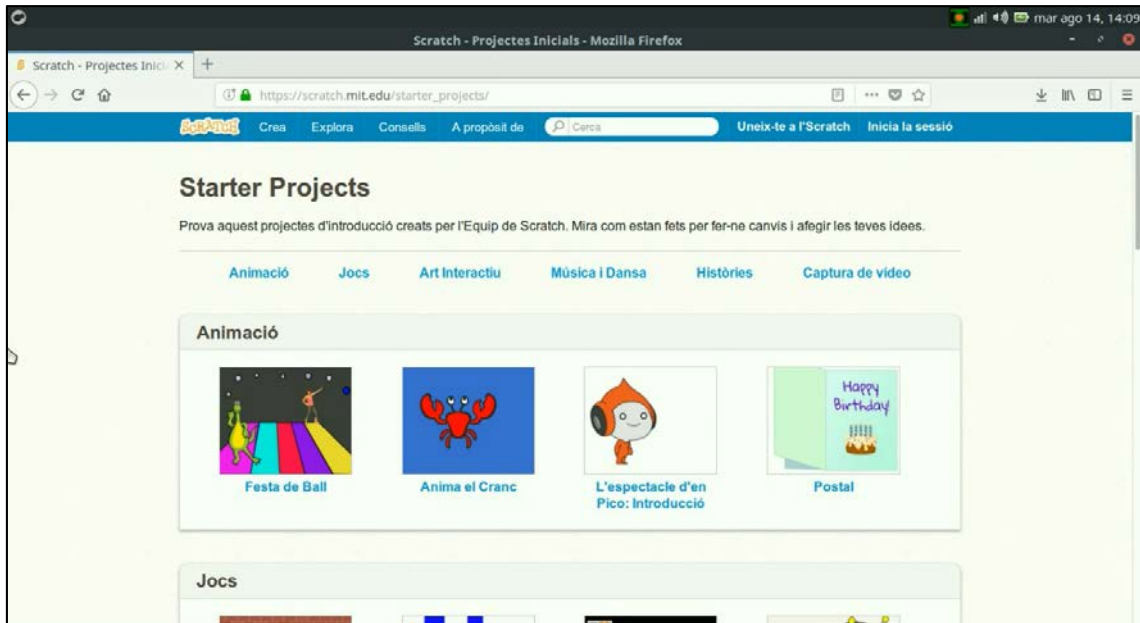


Dins de cada projecte tutoritzat, hi ha:

1. Un tutorial online, directament des de la pestanya de Crea.
2. Una sèrie de targetes d'activitats que fan referència offline al tutorial anterior.

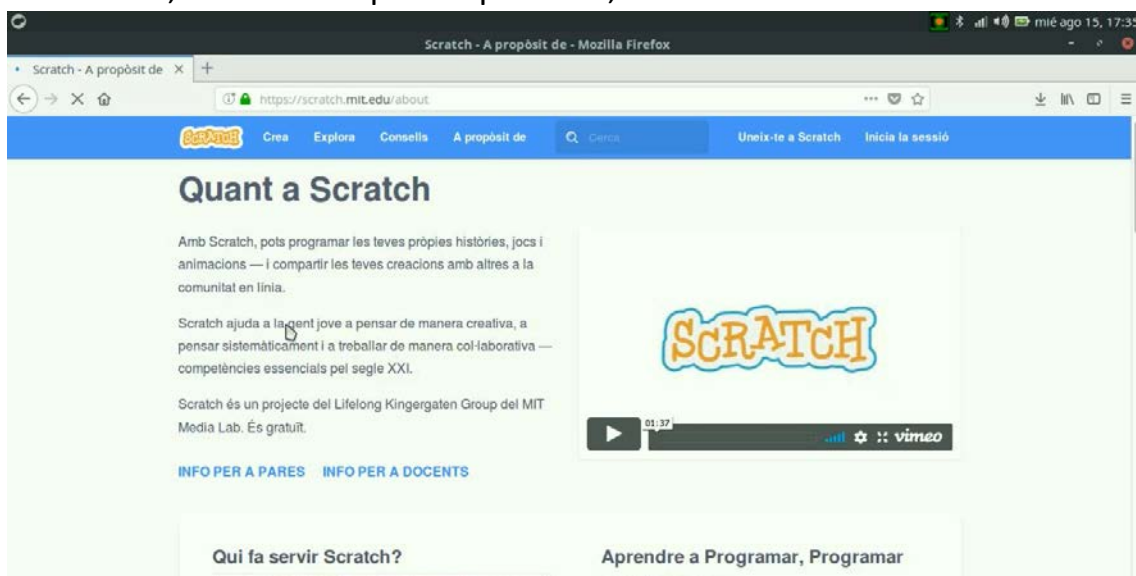
3. Una guia docent, que ajudarà al professor a conduir aquesta classe-tutorial. Amb l'inconvenient, de que aquestes guies sempre estan en anglès.

A aquesta pestanya també podem trobar els “projectes d’inici”. Aquests projectes ens ajudaran a agafar idees per tal de crear els nostres futurs projectes, programes o jocs. Podrem, entrar a veure com estan fets, i llegir en ocasions, algunes notes sobre el funcionament dels conjunts de blocs.



A propòsit de

A aquesta pestanya trobarem principalment informació sobre Scratch, per una banda, destinada a pares i per altra, destinada a docents.



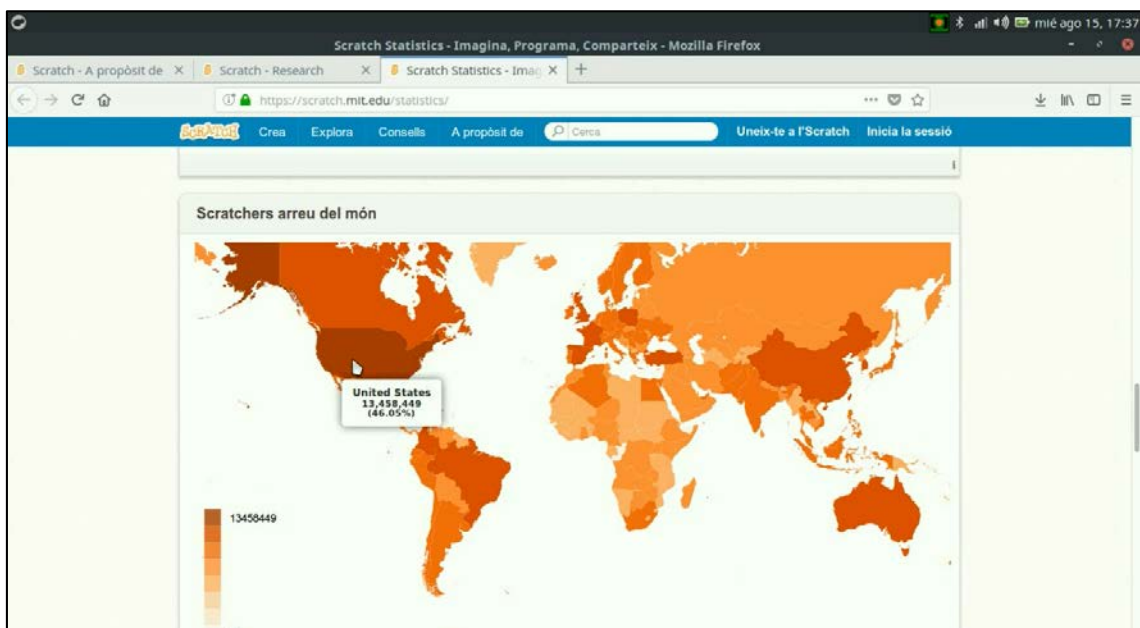
Informació per a pares



A la informació per a pares trobarem unes FAQ (Frequently Asked Questions), és a dir, unes preguntes contestades, que ens ajudaran en el cas de que tinguem que parlar als pares sobre el que fem al aula, fer una festa de l'aprenentatge, compartir-ho a un blog, etc...

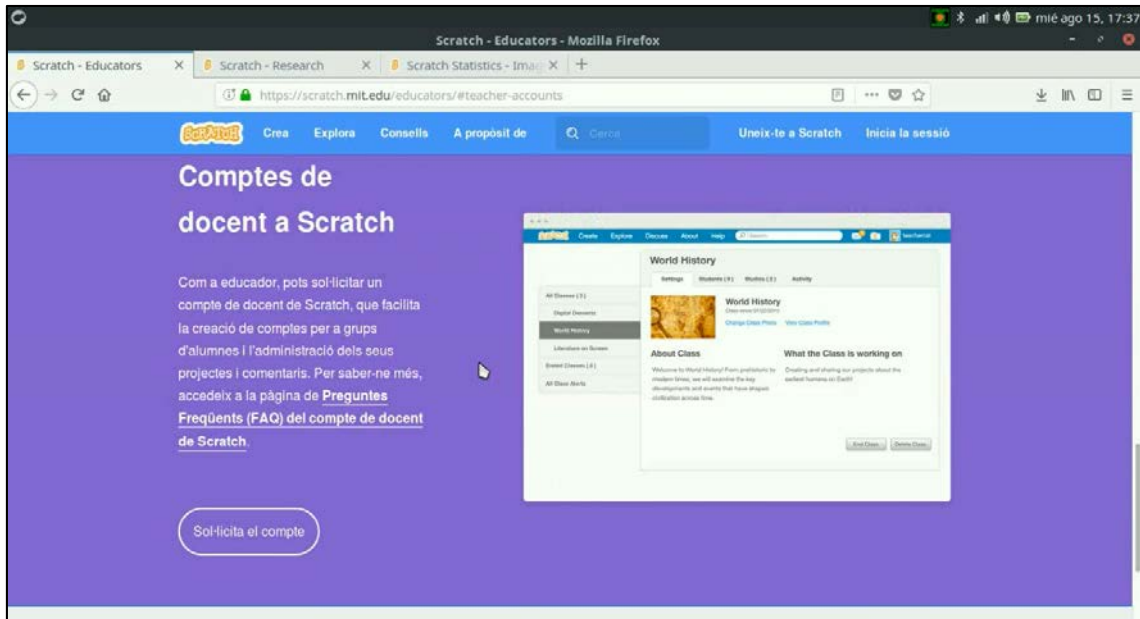
Informació general

A banda, a la pàgina general d'a propòsit de, trobarem informació sobre la quantitat de llengües a les quals està traduït Scratch, l'ús de Scratch arreu del món, estadístiques de Scratch,...

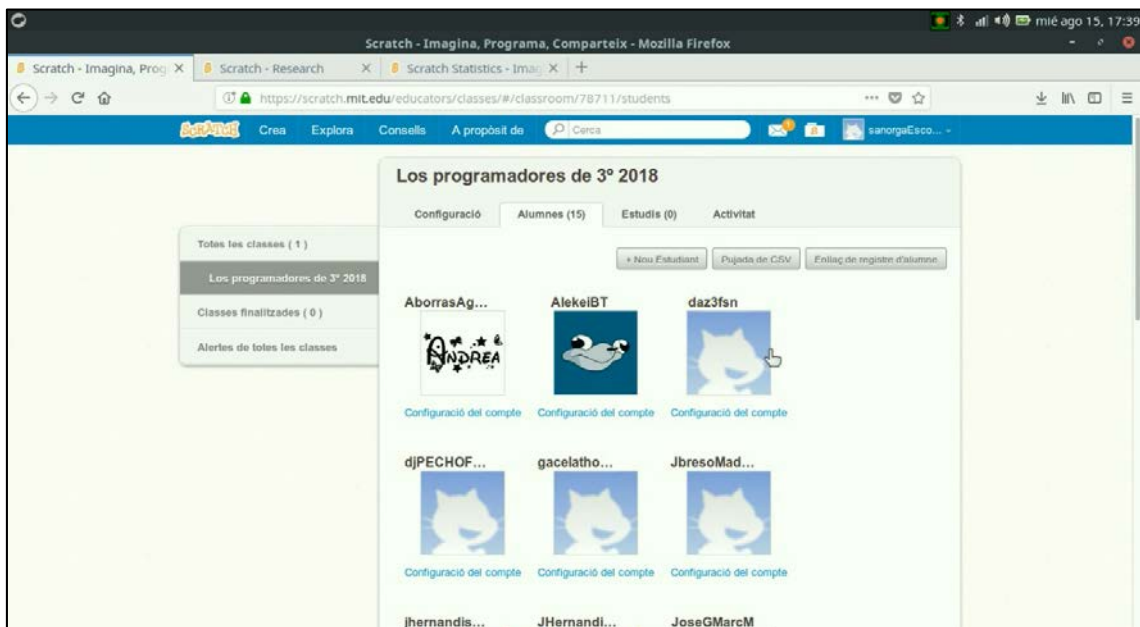


Informació per a docents

El més ressenyable de la pàgina de informació per a docents, ho trobem a l'hora de crear-nos un compte per a docent.



Aquest recurs, ens dona la possibilitat crear un usuari completament nou, el qual ens permet gestionar una classe amb els nostres alumnes per tal de veure, qualificar i avaluar, els diferents projectes que vagen creant els nostres alumnes.



A aquesta pàgina, per a afegir un nou alumne tenim 3 opcions:

1. Afegir directament nosaltres el nom d'usuari i contrasenya de l'alumne.

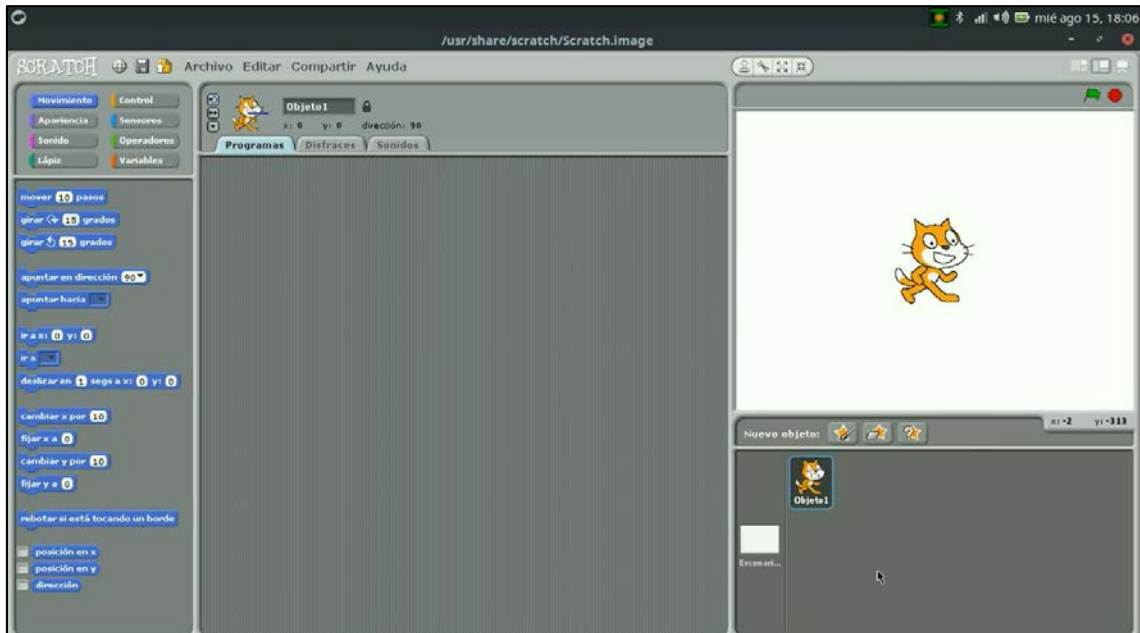
2. Crear un full de càlcul, en format .csv, a la qual hi haja un llistat amb els noms d'usuaris i les seues respectives contrasenyes.
3. Generar un enllaç, a través del qual els alumnes pugen donar-se de alta com a estudiants, escollint ells mateixos tant el nom d'usuari com la seua contrasenya.

Scratch offline

Per tal de descarregar alguna de les versions offline de Scratch hem de fer clic en:

Consells —> Editor fora de línia (al final de la pàgina) —> Scratch 1.4.

L'editor ens permetrà crear els projectes que desitgem en un entorn diferent al de Scratch 2.0 de la web. Però amb la mateixa funcionalitat de blocs.

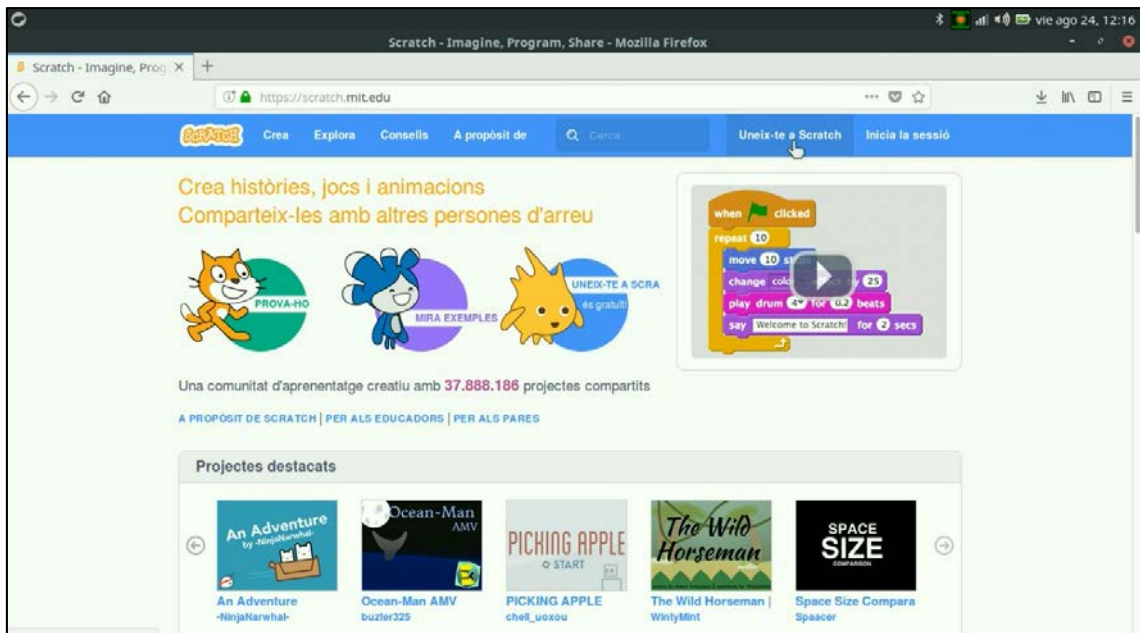


A Scratch 1.4 no tindrem accés a les seues funcionalitats en línia com: compartir un projecte, tutorials online, etc... Però sí podrem guardar un projecte i obrir-ho posteriorment a el Scratch online.

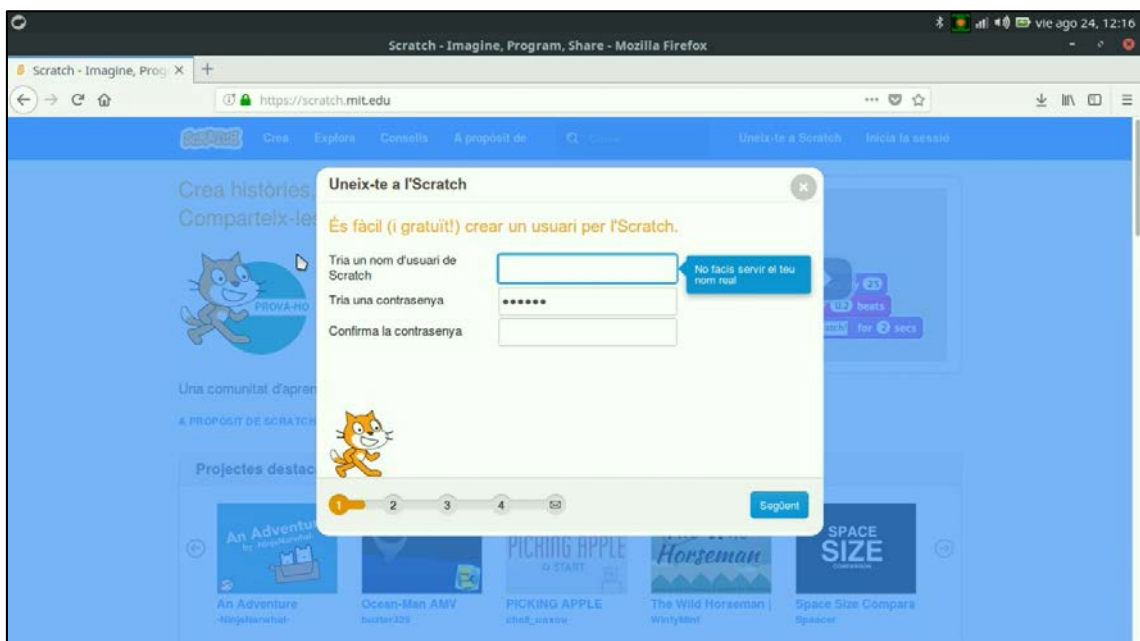
Creació d'usuari

Per tal de crear un usuari nou a Scratch hem de:

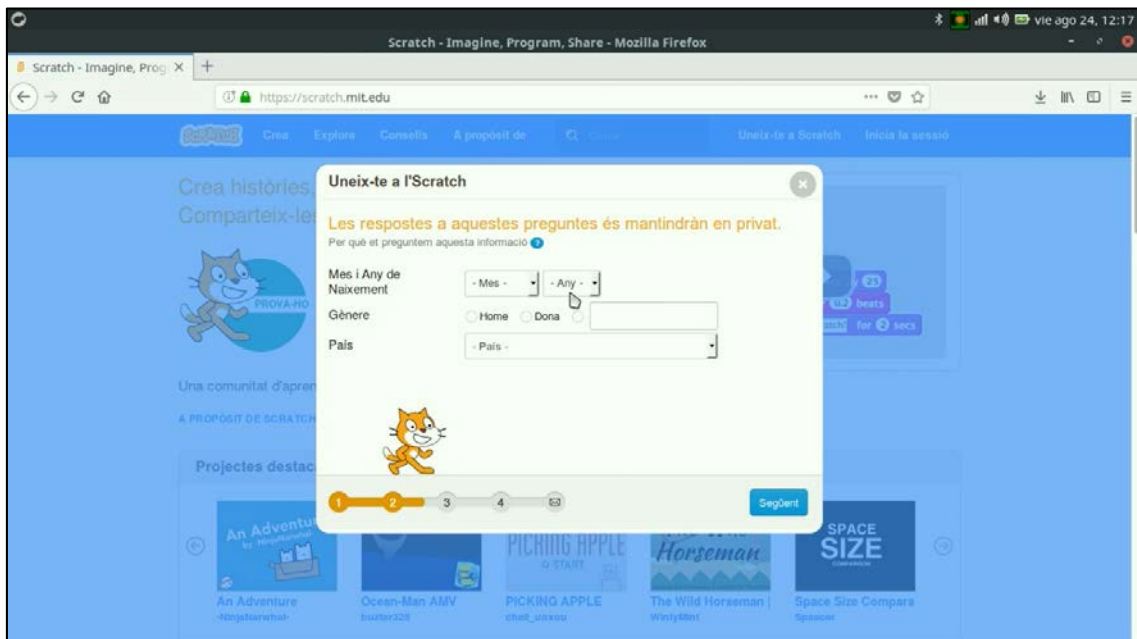
1. Fer clic al botó de Uneix-te a Scratch.



2. Triar-ne un nom d'usuari que no estiga utilitzat, i escollir-ne una contrasenya.



3. Establir-ne la data de naixement, el gènere i el país de residència.

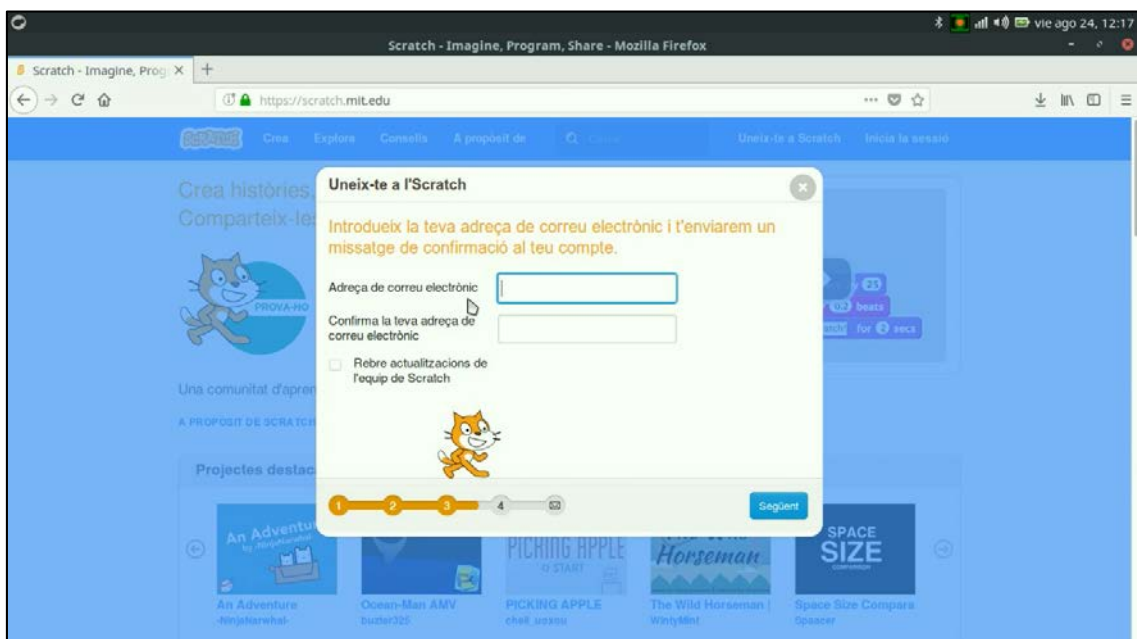


The screenshot shows the Scratch registration process in a Mozilla Firefox browser. A modal window titled "Uneix-te a l'Scratch" is displayed over the main site. The modal contains the following fields and options:

- Mes i Any de Naixement:** Two dropdown menus, one for the month ("Mes") and one for the year ("Any").
- Gènere:** Radio buttons for "Home" and "Dona", followed by an empty text input field.
- País:** A dropdown menu labeled "País".

Below the fields is a progress indicator with four steps, where the second step is highlighted. A "Següent" button is located at the bottom right of the modal. The background shows the Scratch homepage with various project thumbnails.

4. Facilitar-ne una adreça de correu electrònic vàlida.



The screenshot shows the next step of the Scratch registration process. The modal window "Uneix-te a l'Scratch" now displays the following fields and options:

- Adreça de correu electrònic:** A text input field.
- Confirma la teva adreça de correu electrònic:** A second text input field for email confirmation.
- Rebre actualitzacions de l'equip de Scratch:** A checkbox for opting into updates.

The progress indicator now highlights the third step, and the "Següent" button remains at the bottom right. The background of the Scratch homepage is still visible.

Una volta em acabat aquestes passes, ja podem fer clic a "Inicia la sessió", escriure el nostre usuari i contrasenya, i començar a gaudir de l'ús de Scratch!

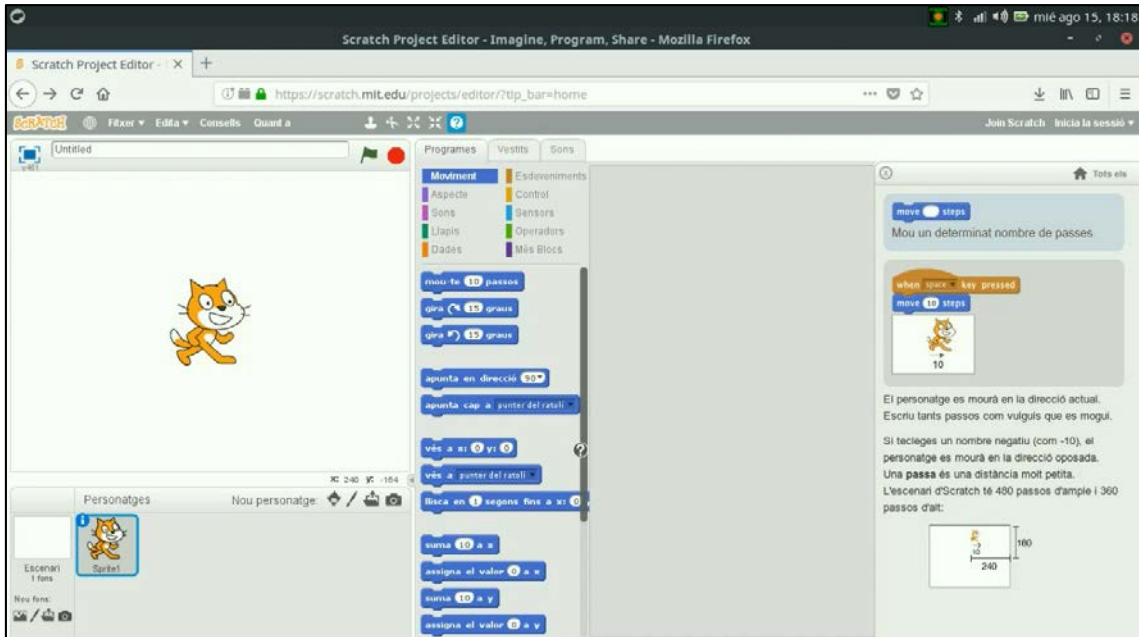
La barra de tasques

Si entrem dins de la pestanya de “Crea”, podem trobar una barra de tasques que ens ajudarà alhora de crear els nostres projectes, exportar-los, copiar personatges,...

Aquesta barra està situada a la capçalera de la pàgina.



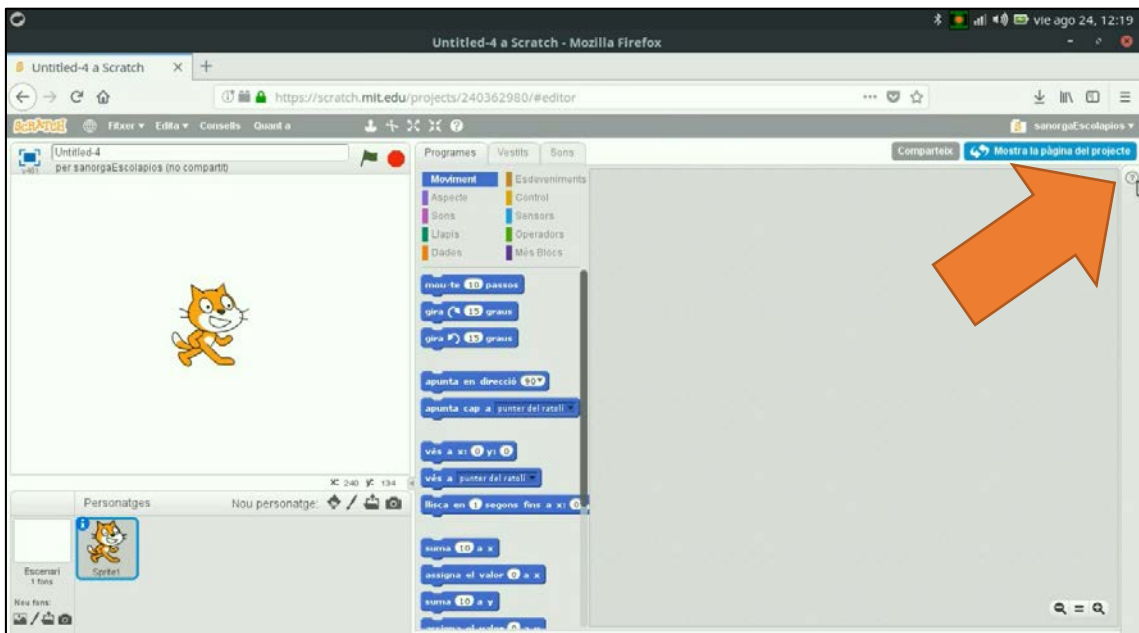
1. **Botó de Scratch:** torna a la pàgina principal de Scratch.
2. **Bola del món:** ens permet triar l'idioma en que se'ns presenta Scratch.
3. **Desplegable Fitxer:** ens permet:
 - a. Crear un nou projecte.
 - b. Carregar un projecte des de el nostre ordinador.
 - c. Descarregar el nostre projecte a l'ordinador.
 - d. Invertir tot el que hem fet en aquest projecte.
4. **Desplegable Edita:** ens permet:
 - a. Desfer la última acció.
 - b. Veure l'escenari més xicotet.
 - c. Posar el mode Turbo (farà que les accions es desenvolupen de manera molt més ràpida).
5. **Consells:** ens porta a la pàgina de “Consells”.
6. **Quant a:** ens porta a la pàgina “A propòsit de”.
7. **Duplica:** ens permet duplicar un objecte.
8. **Tisores:** elimina un objecte.
9. **Creix:** ens permet fer més gran un personatge.
10. **Encongeix:** fa els objectes o personatges més menuts.
11. **Ajuda del bloc:** ens facilitarà una explicació sobre com funciona un bloc fent clic sobre ell. A banda, apareixerà algun exemple d'ús.



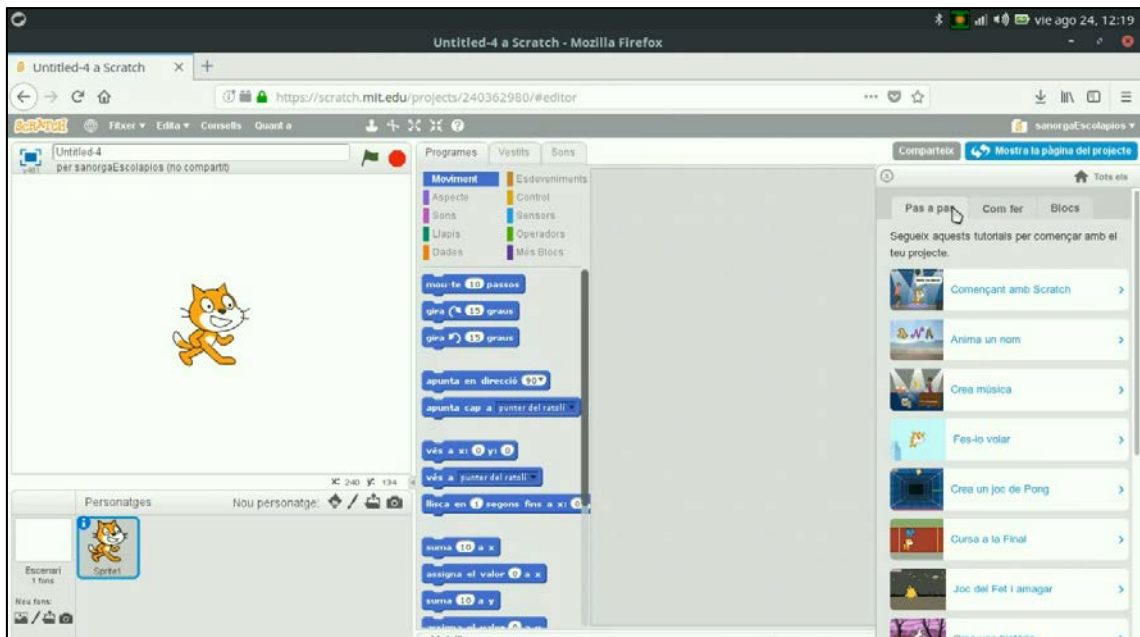
Tutorials de Scratch

Els tutorials de Scratch ens donaran una guia sobre com utilitzar la pestanya de Crea. A través d'ells coneixerem els diferents blocs de Scratch per tal de poder gastar-los en projectes més avançats.

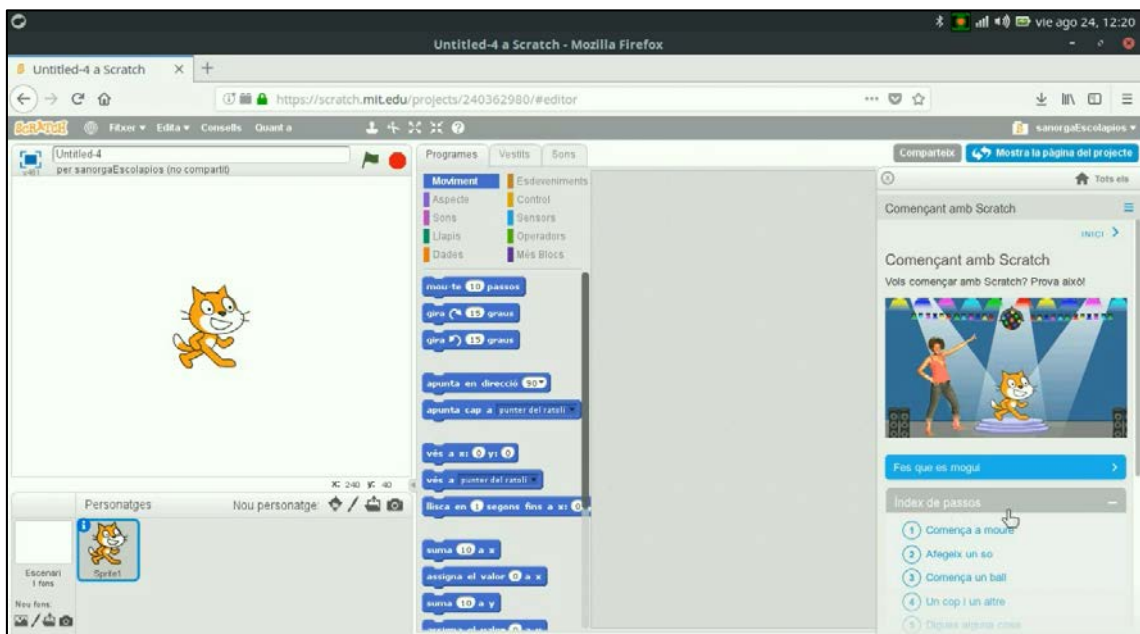
Podem trobar els tutorials fent clic a la icona de l'interrogant situada a la part dreta de la pantalla.



Una volta fem clic se'ns obrirà un desplegable on podrem triar el tutorial que volem fer.



A banda dels tutorials, també podem accedir des de aquí, tant a guies sobre com fer alguns moviments concrets, com a l'ajuda del bloc que tenim fent clic a la barra de tasques, i que hem explicat anteriorment.



Una volta triem un tutorial, se'ns obrirà un menú al qual podrem triar si començar des de el principi, o si per el contrari volem anar a un pas concret dins d'un Índex de passos

A partir d'ací l'únic que hem de fer es seguir les instruccions que se'ns presenten per tal de crear el nostre primer joc.

Bloc 3: La pestanya de crea. Conceptes bàsics.

Aquest document forma part del curs [Pensament computacional a l'aula amb Scratch](#) del [CEFIRE CTEM](#).

Aquesta obra està subjecta a la llicència Reconeixement-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Per veure una còpia d'aquesta llicència, visiteu

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.

Autoria: Santiago Ortega



CONTINUACIÓ DE LA PESTANYA DE "CREA"

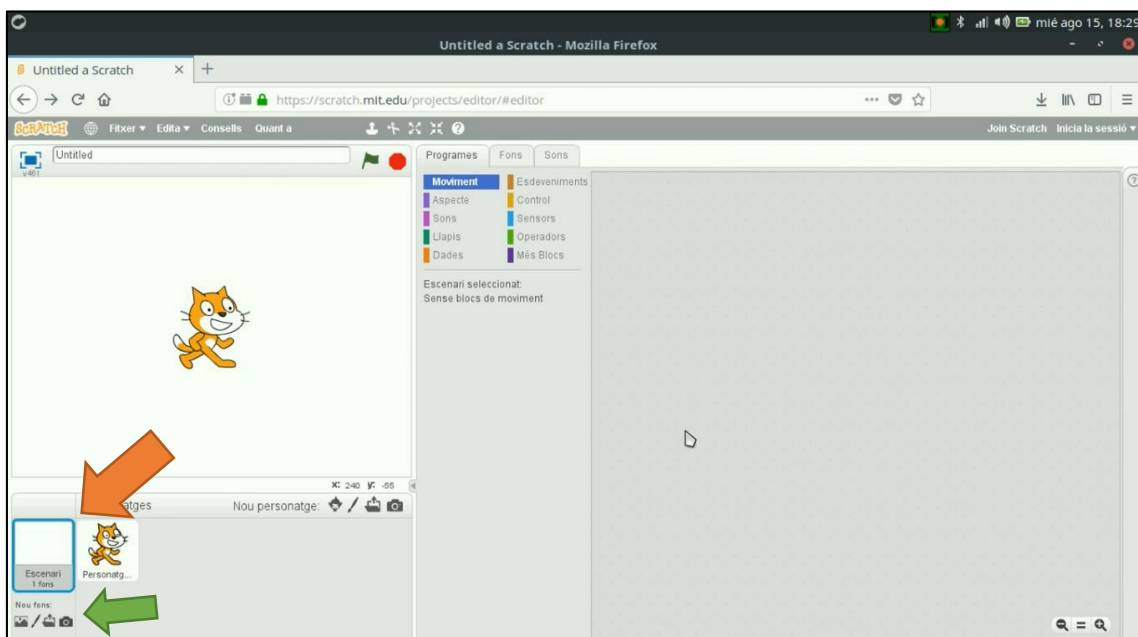
Durant aquesta setmana ampliarem els continguts ja iniciats la setmana passada sobre la pestanya de Crea. Al finalitzar aquesta setmana haurem de ser capaços ja de poder crear alguns jocs bàsics amb Scratch, lo qual serà el temari de la setmana vinent.

Escenaris

L'escenari o fons, és la imatge o conjunt d'imatges que s'utilitzen per tal de donar suport visual al joc o programa.

En general, l'escenari no interactuarà de manera directa en el joc, però sí de manera indirecta com veurem en posteriors vídeos.

L'escenari té els seus propis blocs, de fet, alguns conjunts de blocs, com els de moviment no estaran presents quan estem a l'apartat d'escenari.

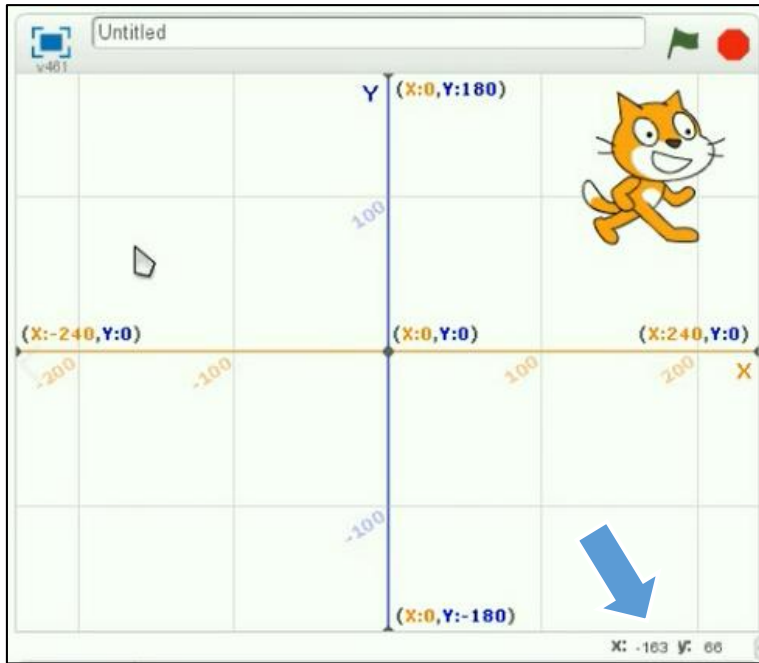


Per **seleccionar l'apartat d'escenari**, haurem de fer clic al racó inferior esquerre. És en aquest racó, on apareixen totes les opcions per tal de **crear un nou escenari**:

1. Carregar un de la galeria d'imatges.
2. Dibuixar un utilitzar l'editor gràfic de Scratch.
3. Pujar una imatge des de el nostre ordinador.
4. Fer una fotografia utilitzant la càmera del ordinador (en cas de disposar una)

PENSAMENT COMPUTACIONAL A L'AULA AMB SCRATCH

PROGRAMACIÓ VISUAL



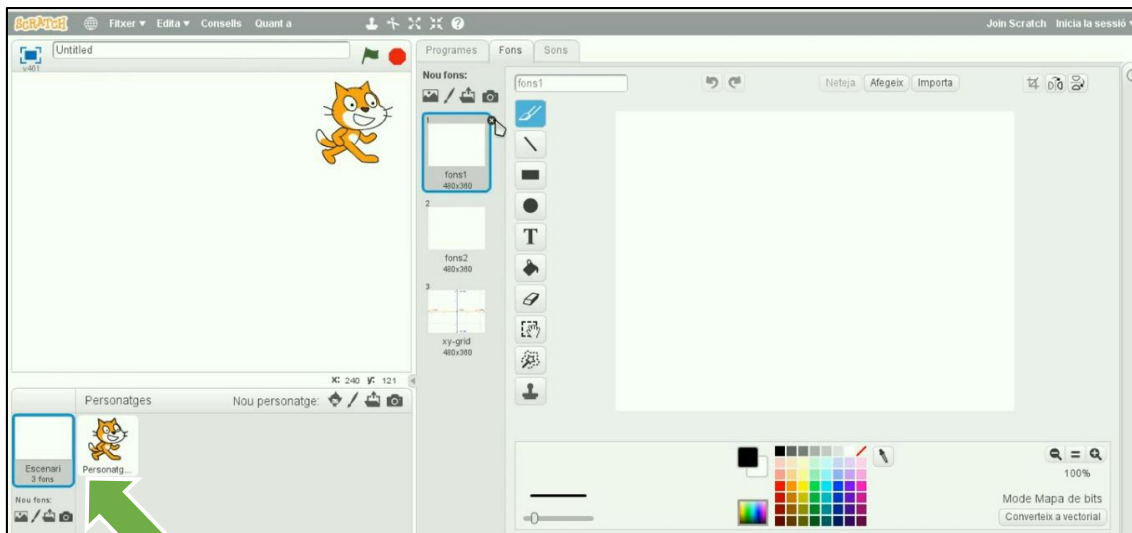
La pantalla de Scratch té un grandària total de 480x360 px.

Hi ha un fons específic a la galeria d'imatges que ens permetrà fer-nos una idea aproximada de com funciona aquesta pantalla.

És important recordar que l'origen de coordenades està

situat al centre de la pantalla, per lo tant, hi hauran valors tant positius com negatius.

Recordeu també que en el racó inferior dret d'aquesta pantalla apareixeran en tot moment les **coordenades** de la posició actual del nostre ratolí.



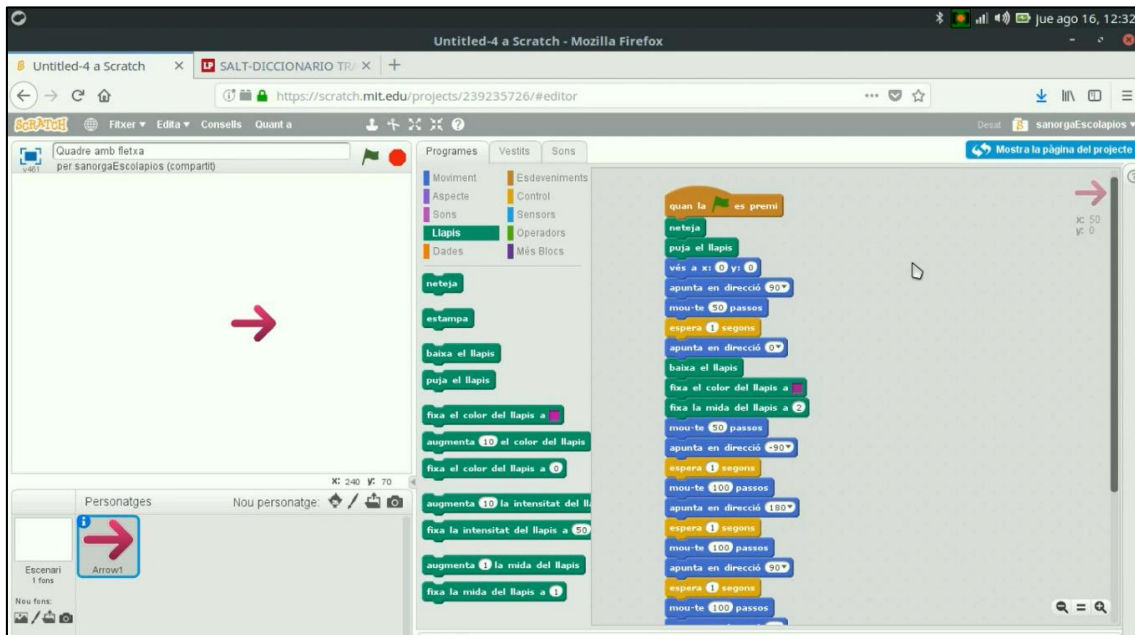
Per tal d'esborrar, ho modificar un fons ja carregat, haurem de fer clic sobre la pestanya de fons.

Des d'ací podrem: canviar-li el nom al fons, pintar-lo, escriure text,... A més a més, podrem treballar la imatge tant de manera vectorial, com en mapa de bits. El qual ens permetrà tindre una major versatilitat alhora de modificar els fons ja creats.

Llapis

El llapis de Scratch és herència d'un antic programa, del qual vàrem parlar la primera setmana que deia Logo. Logo funcionava donant una sèrie d'ordres amb les quals, hi havia una tortugueta, que en manejar-se creava un dibuix com si estiguera pintant amb un llapis.

La funció del llapis de Scratch funcionarà de manera molt semblant, amb l'excepció de que les ordres les donarem utilitzant blocs.



Els blocs específics de la funció del llapis de Scratch són els següents:



Neteja: Borra tot el dibuixat per el llapis.

Estampa: Estampa la imatge del personatge a l'escenari.

Baixa el llapis: Comença a dibuixar.

Puja el llapis: Deixa de dibuixar.

Fixa el color del llapis: Assigna un color de la paleta de colors de manera manual.

Augment __ el color del llapis: Canvia un color de manera numèrica.

Fixa el color del llapis a __: Assigna un color de manera numèrica.

Augmenta__ la intensitat/fixa la intensitat: Si la intensitat és 0, el color del llapis serà negre. Si la intensitat és 100, el color del llapis serà blanc.

Augmenta/fixa la mida del llapis: Canvia el grossor.

Preguntes

Per tal de fer preguntes a l'usuari en Scratch i poder crear una conversa, hem de conèixer dos tipus de blocs diferents: els de digues i pensa per un costat, i els de pregunta per l'altre.

Els blocs de digues i pensa estan situats dins del grup d'“Aspecte”.



La diferencia entre digues i pensa és exclusivament, que en el primer el núvol de còmic que apareixerà tindrà la forma de que el personatge està parlant, i en el segon, pareixerà que està pensant el text que haguérem escrit prèviament.

Els blocs relatius a preguntes en Scratch es troben dins del conjunt de “Sensors”.



Alhora de plantejar una pregunta en Scratch, el més interessant es que després podem utilitzar aquesta resposta per tal de crear una condició, acció o succés posterior. Podent inclusivament, concatenar la resposta amb altres paraules que diga el nostre personatge.

És important tindre en compte, que Scratch utilitza la última resposta donada per el usuari. Per lo que si volem fer varies preguntes, i utilitzar de manera no seqüencial les respostes, haurem d'utilitzar també una llista, element que veurem més endavant en el curs.

Moviment de personatges

Els personatges a Scratch es poden moure de diverses maneres. Tots, els moviments dels nostres personatges els trobarem dins del apartat moviment. Anem a veure algunes d'aquestes maneres de moure, segons el bloc que utilitzen:



El personatge es moure un nombre de passos en la direcció cap a la que està apuntant.





El personatge girarà un nombre de graus determinat, ja siga en el sentit de les manetes del rellotge o a la inversa.





La direcció del personatge apuntarà en l'angle indicat, o variarà apuntant cap al punter del ratolí o un altre personatge.





El personatge anirà immediatament a una posició determinada per l'eix "x" i "y". O anirà directament a la posició que ocupe el ratolí.



El personatge es desplaçarà en un temps determinat, a una posició determinada per l'eix "x" i "y".





Li sumarà o restarà (si el valor es negatiu), un nombre determinat a la última posició de l'objecte. Cada coordenada té el seu propi bloc.





Assigna un valor concret a la coordenada corresponent. Cada coordenada té el seu propi bloc.



Com el seu propi nom indica, el personatge, en cas de tocar una vorera canvia de direcció en un angle reflexiu.



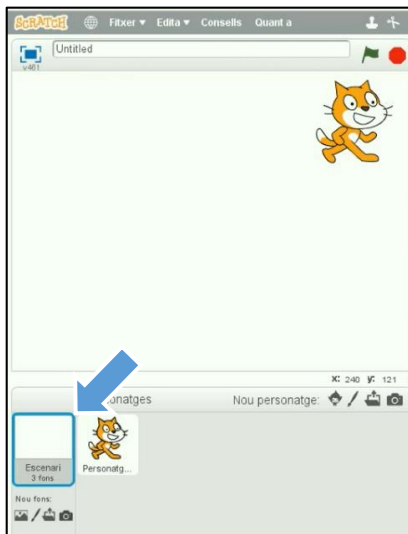






Permet fixar la rotació del personatge en cas de fer un canvi de sentit. Aquesta acció també la podem fer manualment (no programar-la) des de el botó de "i" del personatge.

Canvis de fons i de vestits de personatges



Canvi de fons

Encara que hi ha diverses maneres de fer un canvi d'escenari. Nosaltres anem a explicar per ara la més senzilla.

Per fer un canvi de fons o escenari, el primer que hem de fer es fer clic al quadret inferior esquerre on posa escenari. Posteriorment, haurem de clicar dins del apartat de "Aspecte". És ací on trobarem els blocs relatius a canvis d'escenari.



El bloc canvia el fons, canvia el fons a l'indicat i passa a la següent acció immediatament.

El d'espera, fa el mateix, però espera a que acaben TOTES les seues accions abans de passar a la següent ordre.

Per altre costat, el de següent fons de pantalla, passa al següent fons, segons el ordre en el que estiguen ordenats a la pestanya de fons.



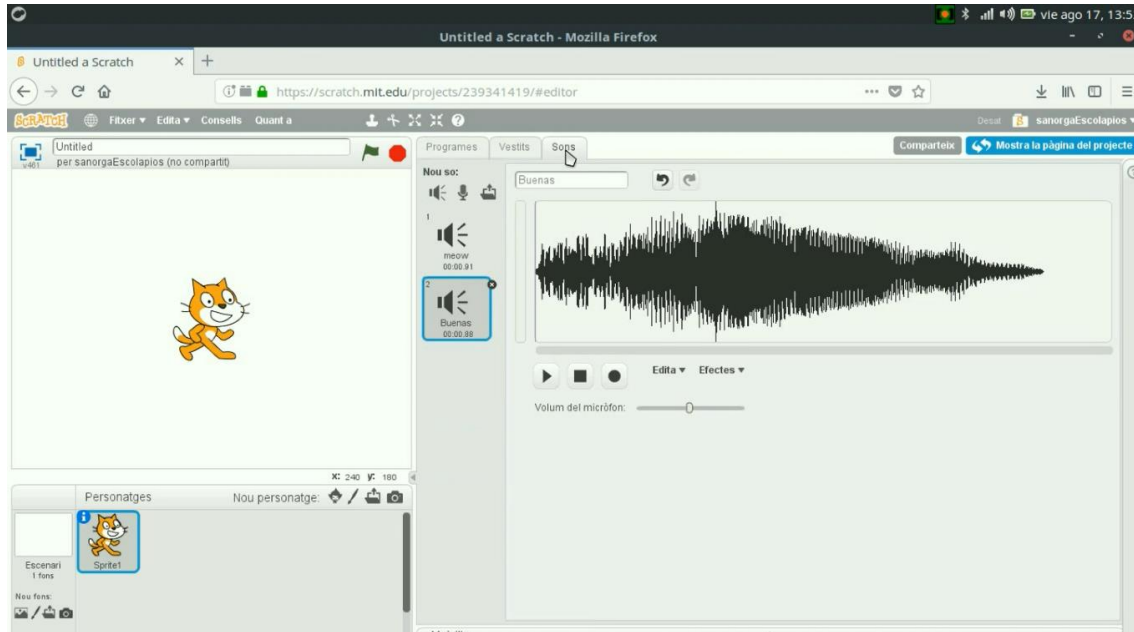
La resta de blocs, eliminen qualsevol efecte que hagem pogut crear prèviament, modifiquen el color de l'escenari, o el fixen en un determinat.

Nota important: Recordeu, en el cas de canvi d'escenari. Posar-hi una ordre al començament del joc per a que retorni al escenari inicial.

Sons

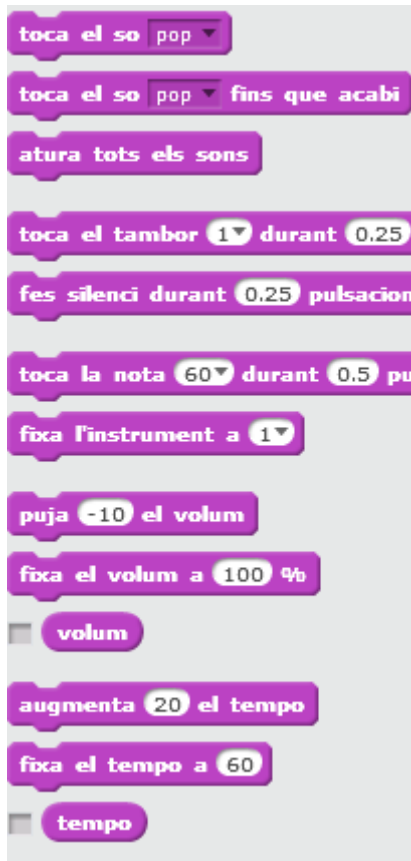
El sons a Scratch es poden crear de 3 maneres diferents:

1. Agafant un de la galeria de sons de Scratch.
2. Gravant un nosaltres mateixa amb un micròfon.
3. Carregar un des de un fitxer del nostre ordinador.



Si triem fer una gravació, sens obri una sèrie d'opcions després de gravar-lo, dins del menú d'editar, apareixeran accions comuns a un editor de sons com: desfer i tornar; retallar, copiar i enganxar; esborrar i seleccionar-ho tot.

A més, el editor de sons de Scratch ens permet, crear alguns efectes, com son: augmentar progressivament, disminuir progressivament, augmentar o disminuir el volum del so, fer un silenci, o invertir el so.



La diferència entre els blocs tocar el so, i tocar el so fins que acabi, es que el segon, s'espera a que acabe el so per a passar al bloc següent.

L'opció del tambor ens permet crear una percussió amb diferents instruments, o crear silencis durant un temps determinat.

Podem tocar notes a un menú desplegable del bloc, així com escriure numèricament la que vulguem. A més, podem triar un instrument de entre 21 diferents.

Pujar, el volum i el tempo ho fan de manera numèrica a partir d'un valor previ.

Fixar, el volum i el tempo, estableix un valor concret per a cadascun d'ells.

Modularitat i extensions

La modularitat és, en programació modular i més específicament en programació orientada a objectes, la propietat que permet subdividir una aplicació en parts més xicotetes (crides mòduls) , cada una de les quals ha de ser tan independent com siga possible de l'aplicació en si i de les restants parts.

[https://es.wikipedia.org/wiki/Modularidad_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Modularidad_(inform%C3%A1tica))

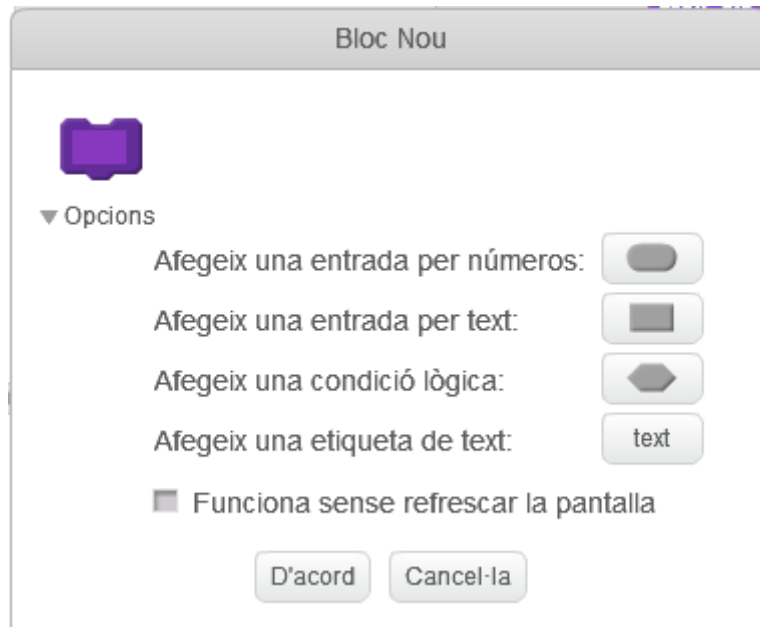
A Scratch, podem trobar aquesta propietat de dues maneres:

1. Afegint una extensió del tipus:
 - a. Picoboard.
 - b. Lego WeDo 1.0
 - c. Lego WeDo 2.0
2. Creant un bloc propi.

PicoBoard i Lego WeDO son plaques de programació independents de Scratch. No obstant això Scratch ens permet programar-les per tal de que funcionen adequadament. Com que no és el motiu d'aquest curs, no anirem més enllà.

Creant un bloc propi

Si dins de l'apartat "Més Blocs", fem clic en "Crea un Bloc", se'ns obrirà una finestra a la qual haurem de donar-li un nom, i decidir si volem afegir alguna entrada numèrica o de text, així com alguna condició lògica o etiqueta.

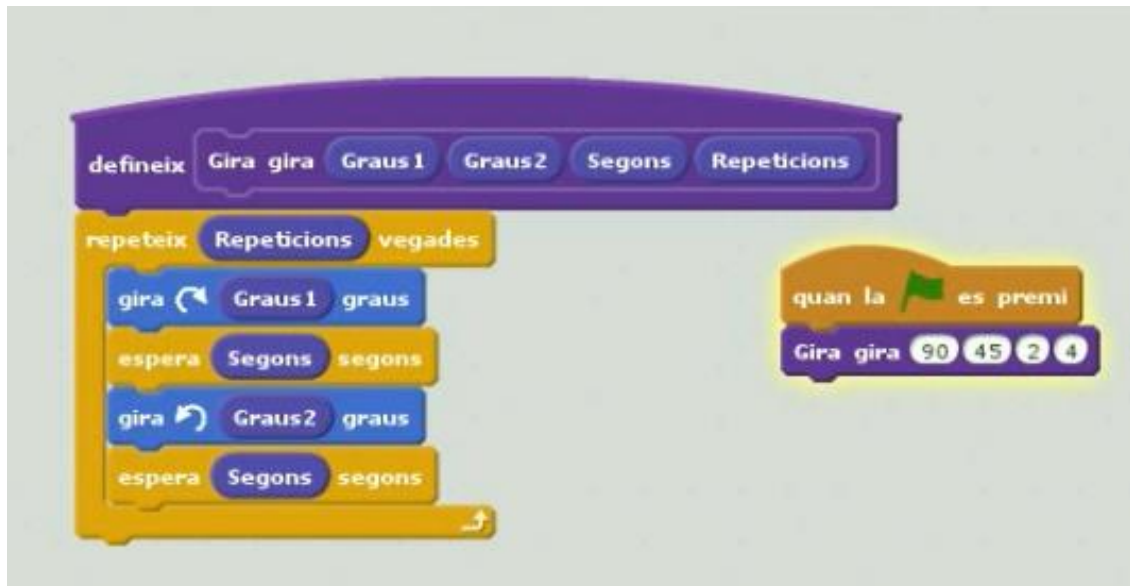


És important recordar, que el que anem a crear és únicament un bloc nou. Per tal de que aquest funcione, haurem de indicar-ho al nostre programa amb un esdeveniment inicial o similar.

Com es pot veure a la imatge inferior, el conjunt de blocs esquerre defineix el que serà el nou bloc de nom "Gira gira", i el conjunt de blocs dret defineix quan ha de posar-se en funcionament aquest bloc.



Si afegim entrades numèriques o de altre tipus al nostre bloc, podem utilitzar-les per tal de referenciar diferents aspectes o valors que vulguem donar de manera externa i/o repetitiva a les nostres accions. D'aquesta manera, s'estalviaren per exemple, tindre que modificar un valor una gran quantitat de vegades.



A l'exemple superior s'ha utilitzat una entrada numèrica per tal de indicar els graus de gir d'un personatge concret. Una altra entrada per definir els segons d'espera entre cada gir. I una última per tal de definir la quantitat de repeticions que han d'haver-hi.

Bloc 4: Jocs bàsics a Scratch.

Aquest document forma part del curs [Pensament computacional a l'aula amb Scratch](#) del [CEFIRE CTEM](#).

Aquesta obra està subjecta a la llicència Reconeixement-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Per veure una còpia d'aquesta llicència, visiteu

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.

Autoria: Santiago Ortega

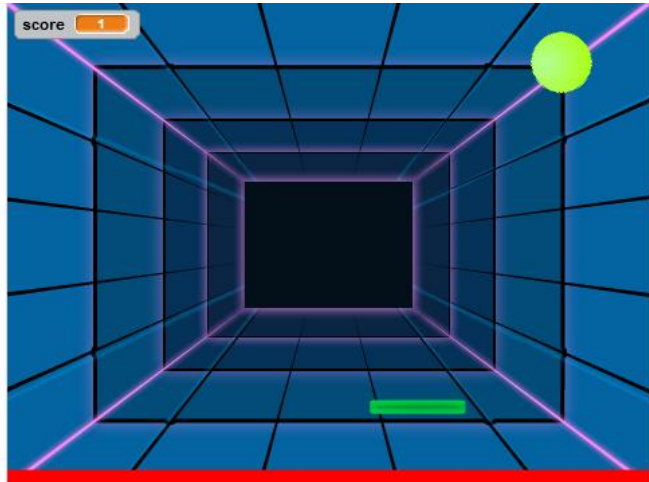


JOCS BÀSICS A SCRATCH

El pong

El pong, breakout, o arkanoid, son diferents noms i jocs d'un mateix concepte. La idea essencial d'aquest joc és golpejar una pilota amb una paleta evitant que aquesta caiga per darrere de la mateixa.

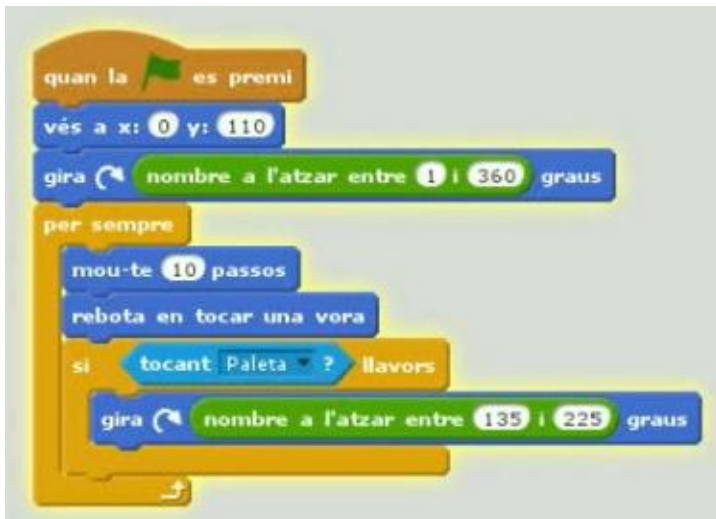
S'hi poden afegir a aquest concepte, blocs que trencar, modificadors de la llargària de la paleta, etc. Nosaltres començarem per el més bàsic.



El primer que hem de fer és crear o carregar dos personatges, un que siga la pilota i l'altre que siga la paleta. A banda d'això haurem de carregar els fons que considerem necessaris, per els diferents canvis que creem.

Una volta ho tenim hem de començar a crear el **codi de la pilota**. Per fer-ho, hem de pensar que la pilota haurà de començar cada volta d'una

manera diferent, ha de rebotar en tocar una vora, i també ho ha de fer en tocar la nostra paleta. Depenent de l'angle de rebot, el joc serà més o menys impredecible.

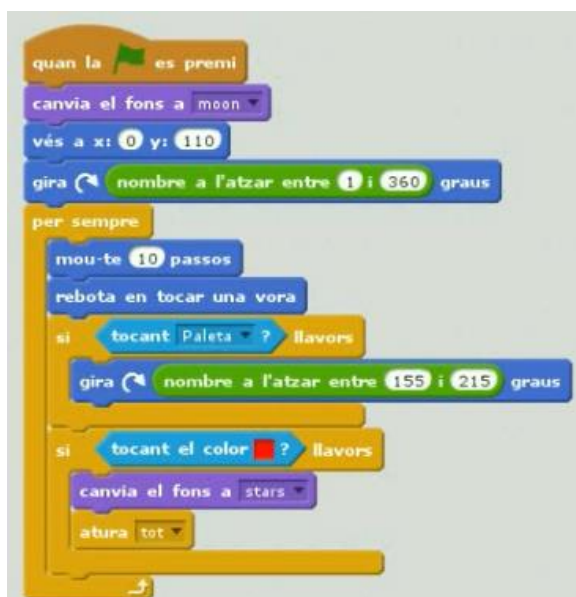




Per crear el codi de la paleta hem de tindre en compte: la seua posició inicial, el seu desplaçament cap a la dreta i a la esquerra i la velocitat a la que ho farà.

Si desitjarem crear un joc per a dos jugadors, seria tan senzill com crear una altra paleta, a l'altre extrem de la pantalla, copiar el codi i assignar-hi tecles diferents als moviments dreta i esquerra.

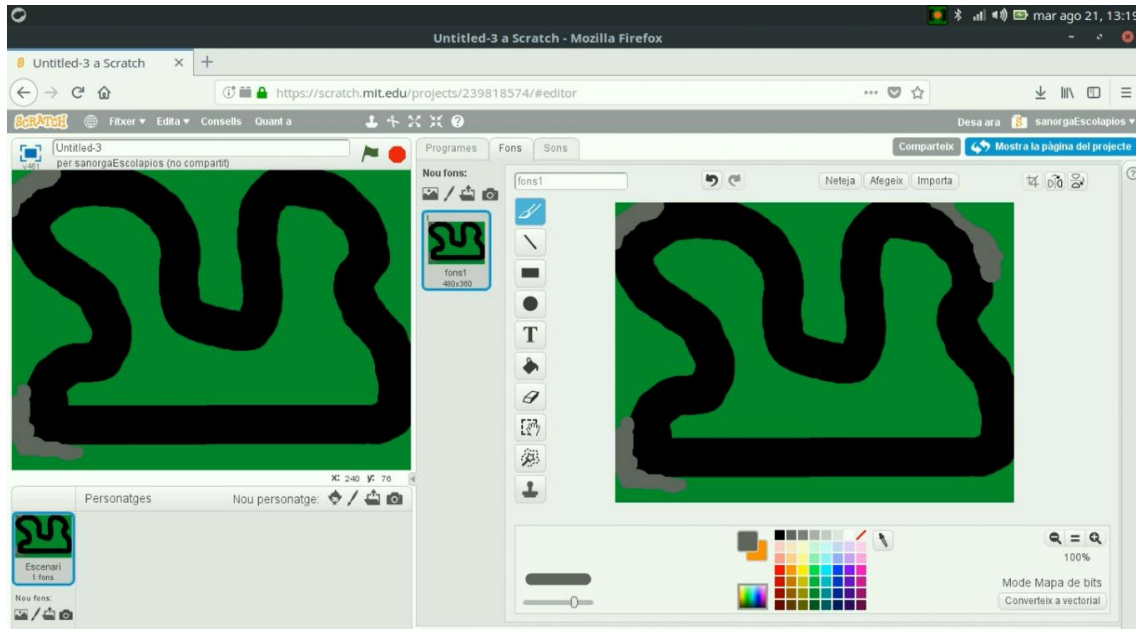
Per crear ara un final de joc, el primer que hem de fer es modificar l'escenari i crear una línia roja baix de la nostra paleta. Recordeu que podem fer-ho des de l'editor gràfic de Scratch ja siga en mapa de bits o en dibuix vectorial.



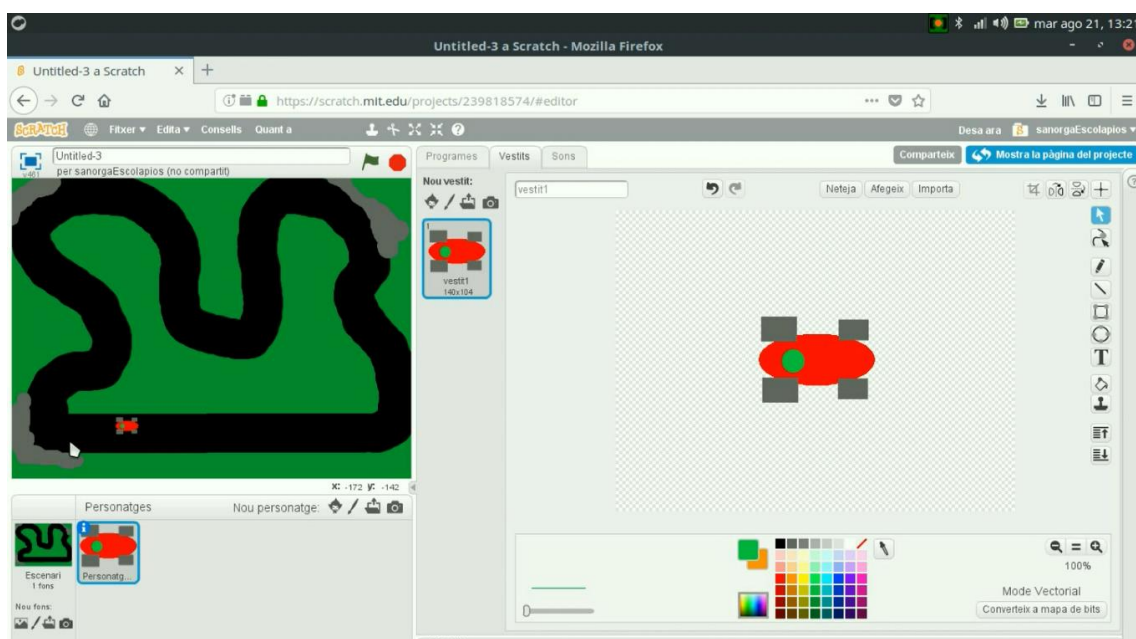
Una volta ho tenim, modificarem el codi de la pilota per tal de que si toca el color roig que hem dibuixat al nostre fons, canvie al fons següent i afegirem el bloc "atura tot", que detindrà tot aquest programa, fent que la pilota es detinga.

Circuit de carreres

El circuit de carreres és un joc que es pot dissenyar per un, dos o més jugadors. La idea principal és crear un circuit, dibuixar un cotxe i duplicar-lo les voltes que siga necessari segons el nombre de jugadors, i fer que en el cas de que algun dels cotxes s'isca, torne al punt inicial.



El primer que hem de fer es crear el nostre circuit amb un mínim de dos colors, un per a la pista, i l'altre per als exteriors de la mateixa. Podem a més, afegir escapatòries a les corbes per donar-li més realisme i fer-lo més senzill. Recordem que podem utilitzar tant el dibuix en mapa de bits com el vectorial.

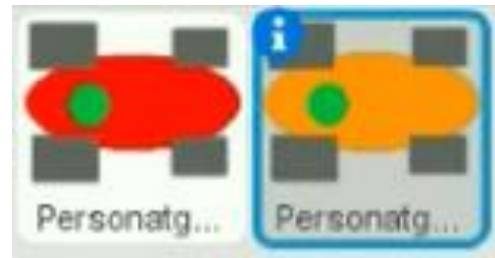


PENSAMENT COMPUTACIONAL A L'AULA AMB SCRATCH

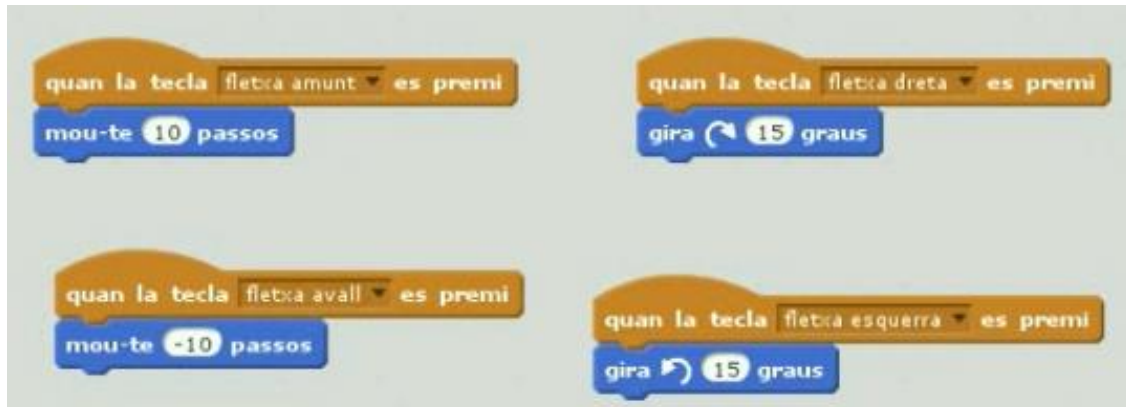
PROGRAMACIÓ VISUAL



El següent que farem serà crear el nostre prototip de cotxe, si hem de crear més d'un, l'únic que hem de fer és duplicar el objecte i després modificar el seu color de farcit.



És important recordar, que no es pot modificar objectes creats a mapa de bits des de vectorial, però sí al contrari.



Ara és moment doncs, de crear el codi que mourà els nostres cotxes, per fer-ho, hem de crear ordres de moviment per a avançar, retrocedir, i girar a dreta i esquerra. Depenent de la velocitat que vulguem imprimir-li, modificarem el valor numèric tant de passos com de graus.

Recordeu que per copiar les ordres d'un objecte a un altre, només hem de fer clic i arrastrar les ordres fins a l'altre objecte. Això sí, es copiaran íntegres, per el que haurem de modificar la tecla que accione cada moviment.



Ara, el que hem de fer és establir la posició direcció inicial dels nostres cotxes, així com la condició de que si se n'ix del circuit retorne al punt inicial. Per fer açò últim, hem decidit utilitzar el bloc de lliscar, per a que es veja com el cotxe retorna paulatinament.

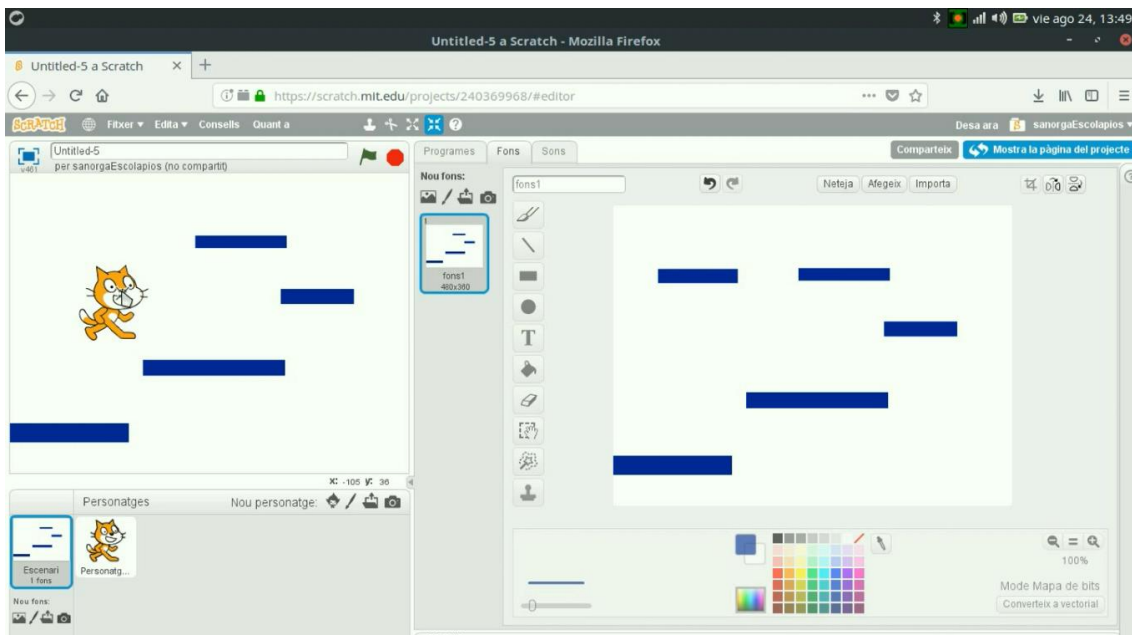
És important que la posició inicial i de tornada dels cotxes no siga la mateixa.



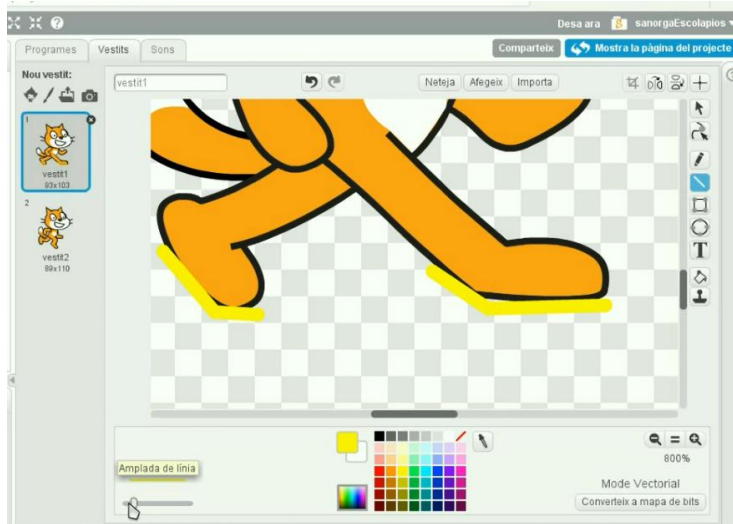
Joc de plataformes

La idea del joc de plataformes és crear un personatge que vaja movent-se per damunt de unes plataformes sense caure. A partir d'aquest concepte, es pot ampliar creant objectius, desplaçaments de l'escenari en scroll, etc...

Per crear el més bàsic del nostre joc, haurem primer de tot de triar un personatge i dibuixar una sèrie de plataformes al nostre escenari. En el cas de que vulguem que les plataformes es moguen haurem de crear-les com a personatges independents del fons o escenari.



Per tant, el primer que farem serà crear a la pestanya d'escenari unes plataformes del color que nosaltres triem. Després reduïrem la grandària del nostre gat per tal de que quadre amb l'escala de les plataformes.



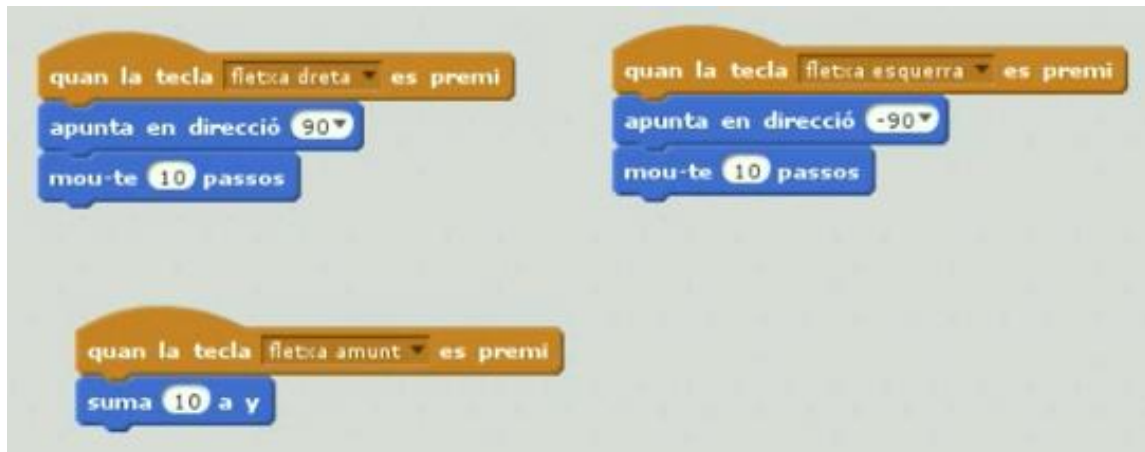
Seguidament, situarem unes línies de color a la part de baix dels peus del nostre personatge, per tal de crear un color diferent als que té i evitar així que es quede enganxat quan creem el codi per evitar que caiga en tocar una plataforma.

Una volta ja hem fet el pas anterior, és hora de crear el codi que farà que el nostre personatge caiga, però es detinga en tocar una plataforma.

Per fer-ho, crearem un esdeveniment inicial, i un condicional de per sempre, si no, el color dels peus del personatge (en aquest cas el groc), està tocant el color de les plataformes (en aquest cas el blau), llavors canvie la posició "y" del nostre



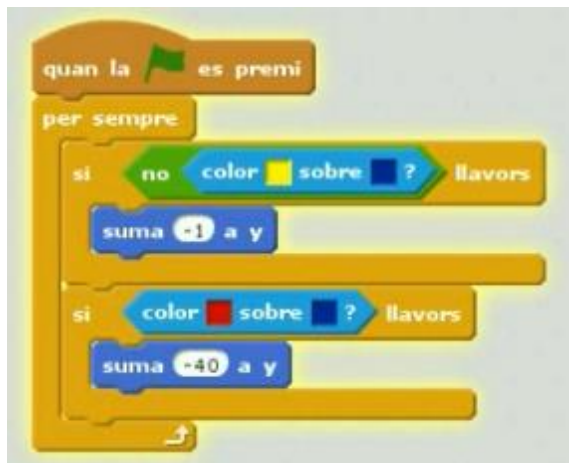
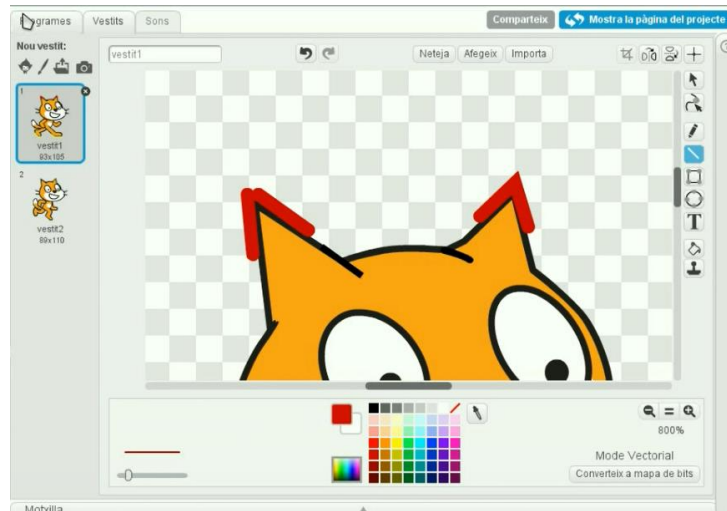
personatge per -1 (en el cas de que vullguem que caiga més ràpid haurem de modificar aquest valor fent-lo més negatiu).



Ens queda ara crear el moviment del nostre personatge. Triarem doncs, un conjunt de tecles per als seus desplaçaments horitzontals a esquerra i dreta, fent que apunte en direccions oposades (sentits si som matemàticament estrictes) i que es moga una quantitat de passos determinada.

Després haurem d'escollir una tecla per al moviment vertical, fent que aquest siga superior sempre, al de l'efecte de caiguda creat anteriorment.

Ara, per tal d'evitar que el nostre personatge pugui traspasar les plataformes, fent doncs, trampa. Crearem unes línies de un altre color a la banda de dalt d'aquest. Torna a ser important, que les línies no siguin massa fines, però lo suficientment discretes per a no embrutar el dibuix.



Ara el que restarà, serà afegir un condicional al codi del nostre personatge, per tal de que sempre, si el color de les línies del cap, està tocant el color de les plataformes, l'envie a una posició inferior immediatament. Creant així un efecte de rebot. Resulta essencial, que aquest valor negatiu siga superior al del desplaçament vertical.

Bloc 5: Variables, llistes, clons... Conceptes avançats.

Aquest document forma part del curs [Pensament computacional a l'aula amb Scratch](#) del [CEFIRE CTEM](#).

Aquesta obra està subjecta a la llicència Reconeixement-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Per veure una còpia d'aquesta llicència, visiteu

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.

Autoria: Santiago Ortega.



VARIABLES, LLISTES, CLONS... CONCEPTES AVANÇATS

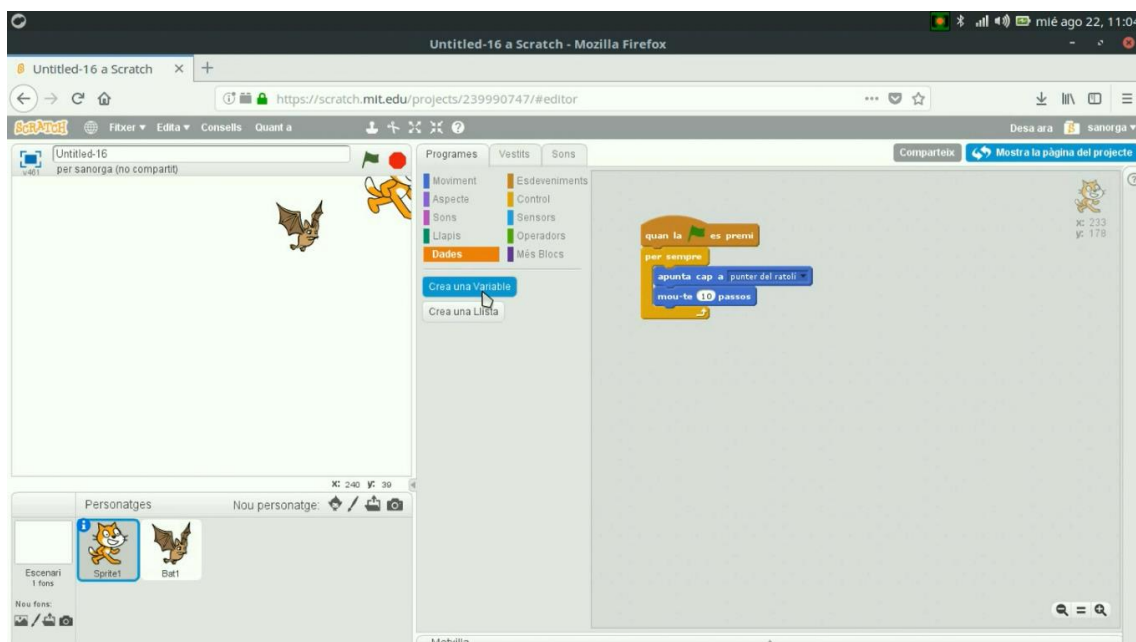
Variables

Una variable a Scratch ens permet emmagatzemar un valor numèric canviant amb la etiqueta de un nom concret. D'aquesta manera podem aprofitar la variació d'aquesta variable per crear condicionals, operadors,...

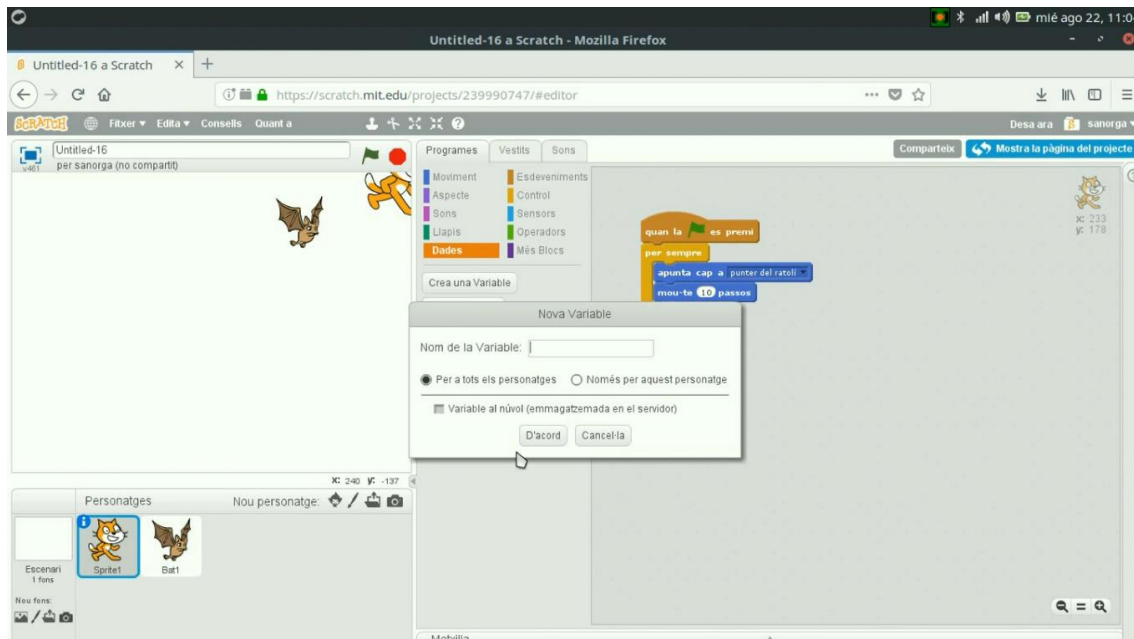
Exemples d'ús de variables a Scratch:

- Vides.
- Punts.
- Repeticions d'una iteració.
- Velocitat.
- ...

Les variables estan ubicades dins del apartat de dades. Al fer clic, ens apareixeran dos opcions, un botó amb l'opció "Crea una variable", i un altre botó amb l'opció "Crea una llista".



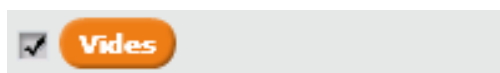
Si fem clic en “Crear una variable”, se’ns obrirà una finestra que ens demanarà que li posem un nom.



A banda del nom, ens apareixen tres opcions alhora de crear-la:

- **Per a tots els personatges.** (Podrem utilitzar-la a qualsevol personatge).
- **Només per a aquest personatge.** (No apareixerà a la resta de personatges per a utilitzar-la com a bloc).
- **Variable al núvol.** (Molt útil si volem emmagatzemar dades per a diferents usuaris com ara puntuacions, rècords, etc...)

Els **blocs** que apareixeran una volta creem la variable i li donem un nom, com per exemple: Vides, son els següents:



Bloc de Referència: el podem utilitzar per referència al valor de la variable a un altre bloc.



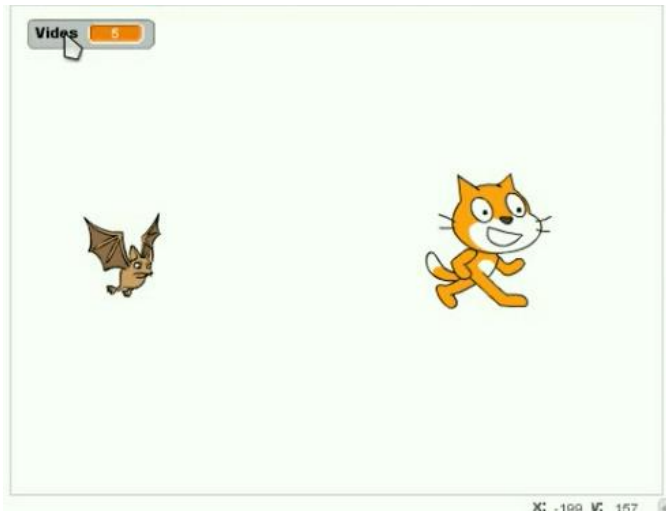
Assigna un valor inicial a la variable.



Augmenta o disminueix (valor negatiu) el valor de la variable.



Mostra o amaga la variable a l'escenari. Però continuarà funcionant.



A més, si fem clic i arrastrem el quadre de la variable, podrem modificar la seua posició inicial (racó superior dret) a qualsevol lloc de la pantalla.

També tenim la possibilitat de canviar la forma en que es presenta aquest quadre, de forma que si fem doble clic sobre ell, es presentarà de tres formes diferents; la primera indicarà el nom i el nombre actual de la variable; la segona indicarà únicament el nombre actual; i la tercera indicarà nom i nombre actual i a més permetrà modificar el valor de la variable utilitzant el ratolí amb una rodeta.



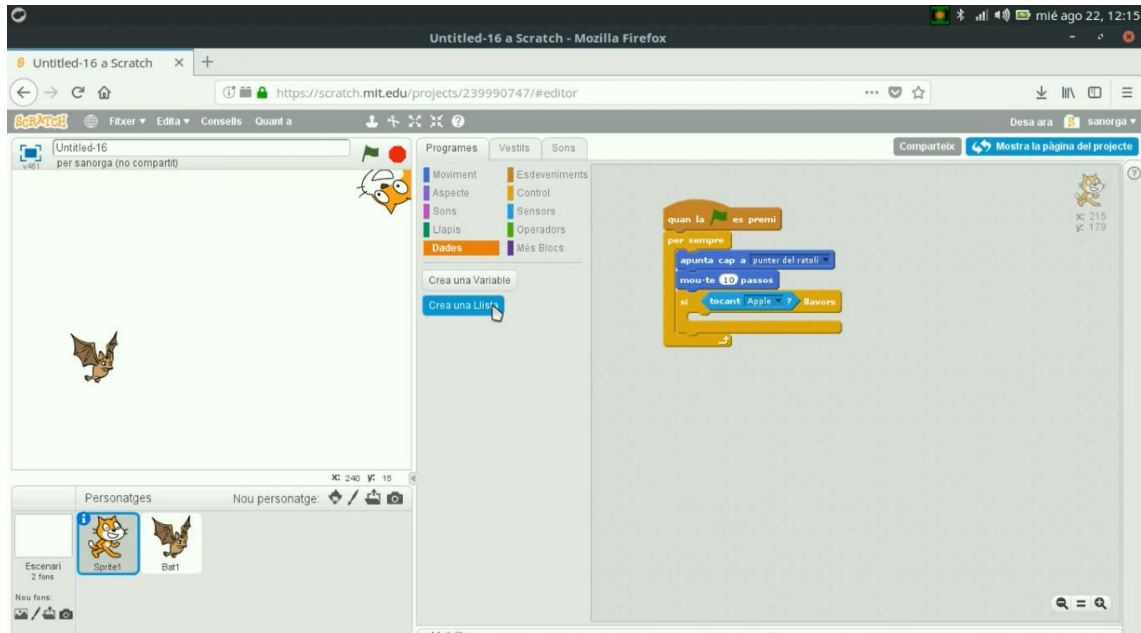
Llistes

Les llistes a Scratch ens permeten caràcters alfabètics i numèrics, que poden tindre un semàntic o no. A banda d'aquesta clara diferència amb les variables, les llistes a Scratch permeten emmagatzemar més d'un valor, mentre que la variables tan sols pot emmagatzemar un.

Exemples d'ús de les llistes a Scratch:

- Recol·lecció d'objectes.
- Respostes a preguntes.
- Total de punts d'un o diferents jugadors.
- Respostes valides a certes preguntes.
- ...

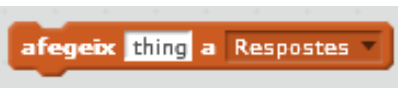
Les llistes estan ubicades dins del apartat de dades. Al fer clic, ens apareixeran dos opcions, un botó amb l'opció "Crea una variable", i un altre botó amb l'opció "Crea una llista".



Els **blocs** que apareixeran una volta creem la llista i li donem un nom, com per exemple: Respostes, son els següents:



Bloc de Referència: el podem utilitzar per referència a la llista a un altre bloc.



Afegeix un element a la primera posició de la llista.



Esborra un element de la llista en una posició concreta.



Afegeix un element a una posició concreta de la llista



Canvia un element concret de la llista per un altre.



Podem fer referència tant a un element concret de la llista o a la llargada de la mateixa.



Preguntem a la llista si conté un element concret.



Mostra o amaga la llista respectivament a la pantalla del joc.

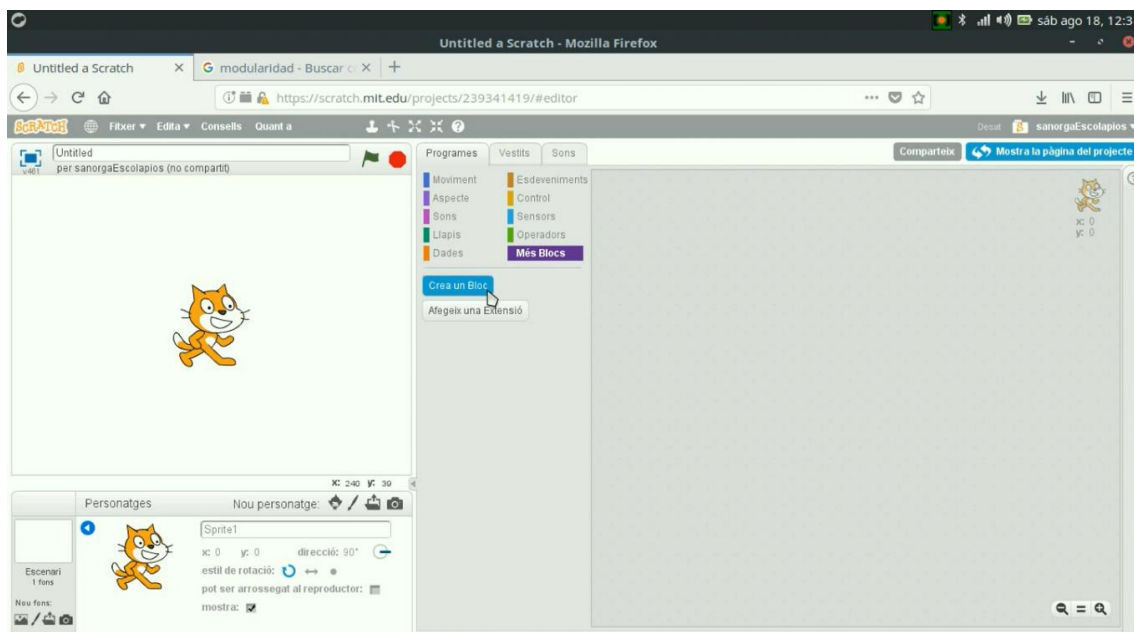
Modularitat i extensions

Extensions

Si volem connectar a Scratch i programar una placa del tipus:

- Picoboard
- LegoWeDo 1.0
- LegoWeDo 2.0

Hem d'utilitzar les extensions que estan ubicades dins del apartat de més blocs. Si fem clic a "Afegeix una extensió", s'obrirà una finestra on podrem triar el tipus de placa que connectarem.



Posteriorment ens apareixeran una sèries de Blocs propis de cada placa, que ens permetran configurar i programar aspectes com: servos, motors, etc...

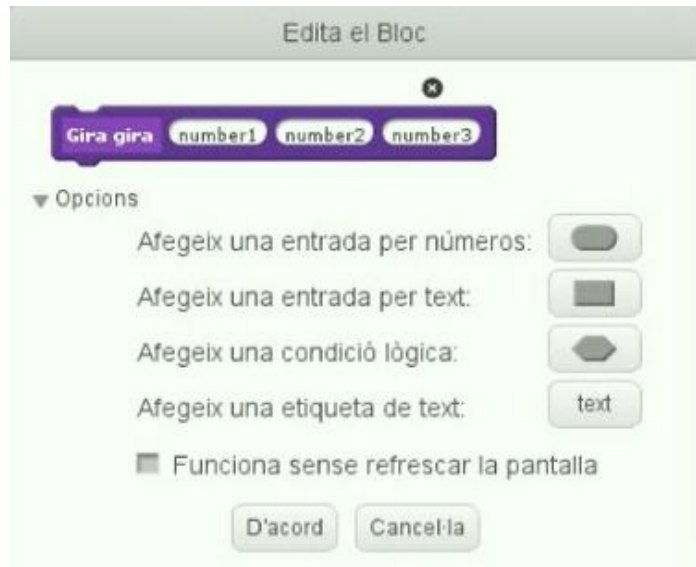


Modularitat

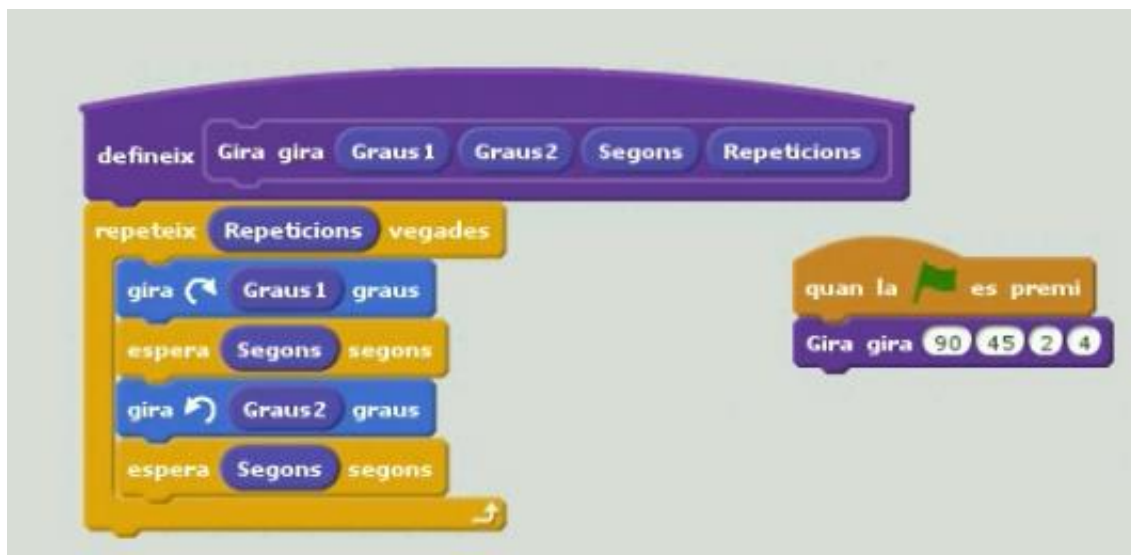
La modularitat ens permetrà crear blocs nous a Scratch completament personalitzats per nosaltres que incloguen altres blocs de qualsevol de les opcions de Scratch (excepte d'un esdeveniment inicial).

Per crear un nou bloc haurem d'entrar dins del apartat de "Més Blocs" i fer clic en "Crea un Bloc". Llavors se'ns obrirà una finestra a la qual podrem triar el nom del bloc i afegir-ne:

- Entrades numèriques.
- Entrades de text.
- Entrades lògiques.
- Etiquetes.



Una volta hagem finalitzat el procés de creació del bloc, haurem de definir quines seran les accions que haurà de realitzar i posteriorment, haurem d'utilitzar-lo dins d'un esdeveniment inicial.

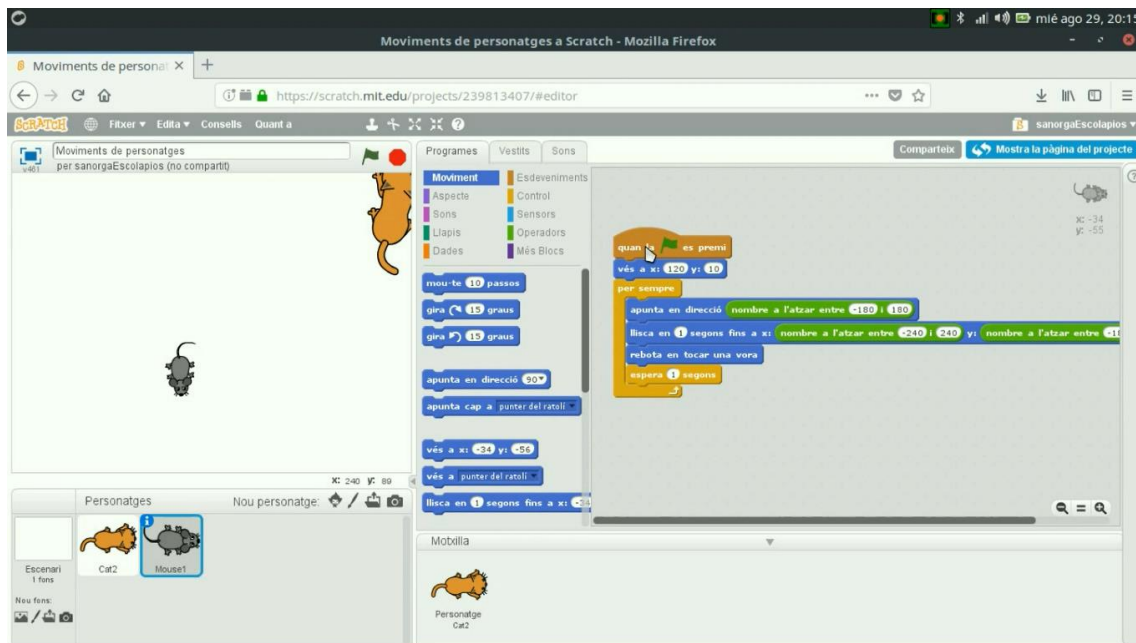


Al exemple superior hem creat un bloc anomenat: "Gira gira", el qual presenta quatre entrades numèriques, dos per a els graus de gir, una per als segons d'espera entre cada gir, i una última per a la quantitat de repeticions de totes les accions. D'aquesta manera modificant únicament

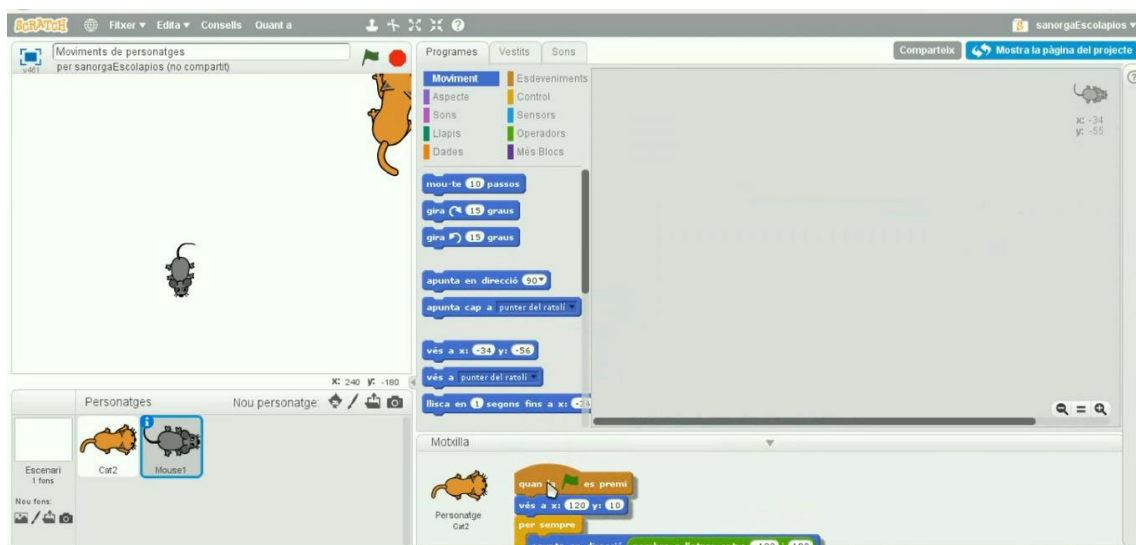
els valor inicials del nou bloc podem modificar les diferents accions programades.

Motxilla

La motxilla és un recurs de la IDE de Scratch que ens permetrà importar objectes i programes d'un projecte a un altre sense necessitat de crear còpies del projecte o fer-ho de manera manual.



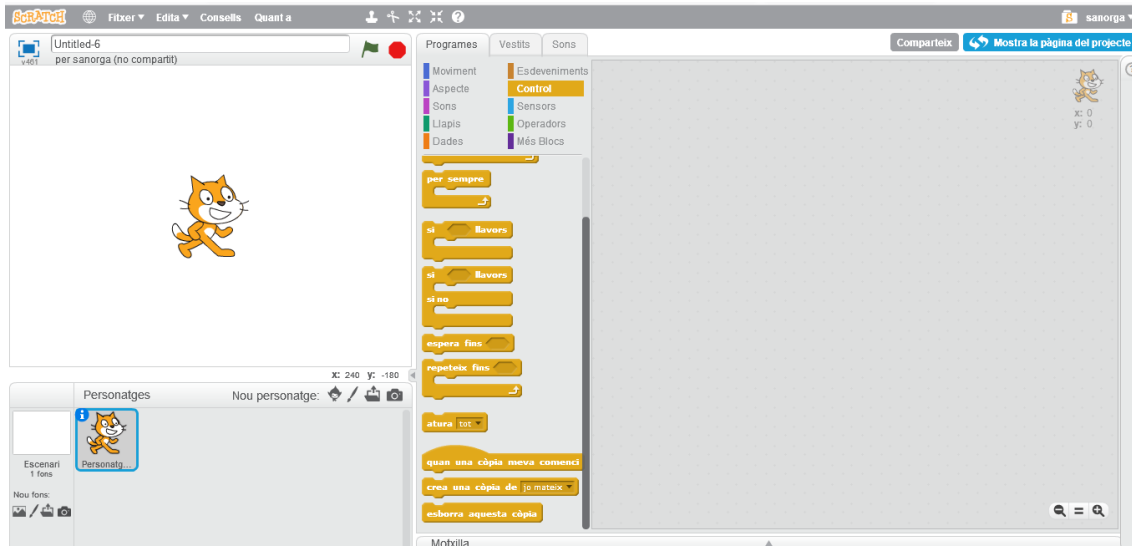
Per tal d'utilitzar la motxilla a Scratch, l'únic que haurem de fer és arrossegar a la pestanya inferior on indica "Motxilla", l'objecte o programa que volem importar. Després obrirem la motxilla a un altre projecte i arrossegarem novament el objecte o programa, però en aquest cas, en sentit contrari.



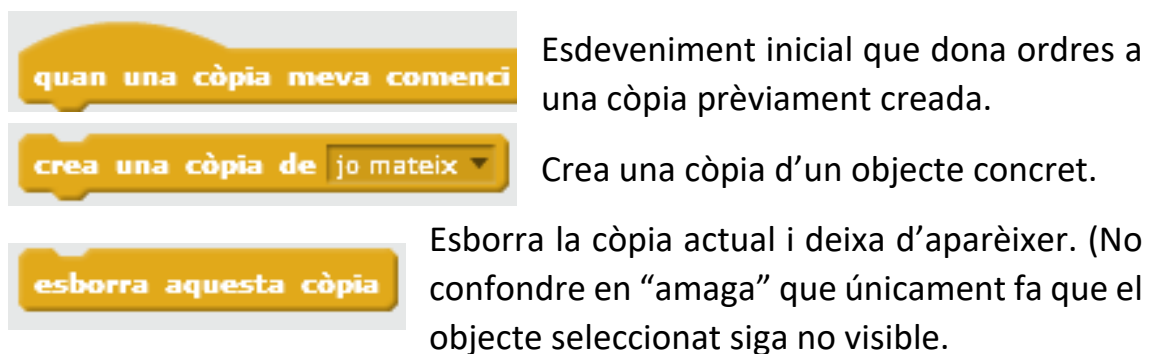
Còpies o clons

Les còpies o clons a Scratch permeten crear altres objectes iguals al copiat que duraran mentre no es reinicie el joc.

Trobarem els blocs pertinents dins del apartat de “Control” a la banda de baix de tots els blocs.



Els blocs que trobarem relacionats amb la creació de còpies són els següents:



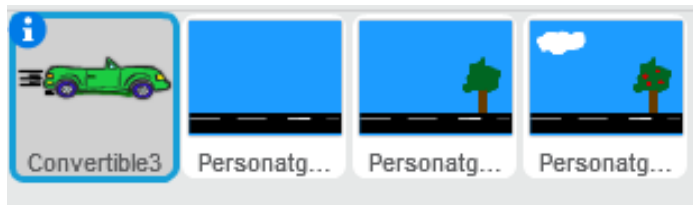
Per crear una còpia a Scratch i que aquesta realitze alguna acció haurem sempre de fer dos coses:

1. Ordenar al personatge original (o a una còpia existent) que cree una còpia de si mateix.
2. Programar les ordres del clon o còpia.

Scrolls

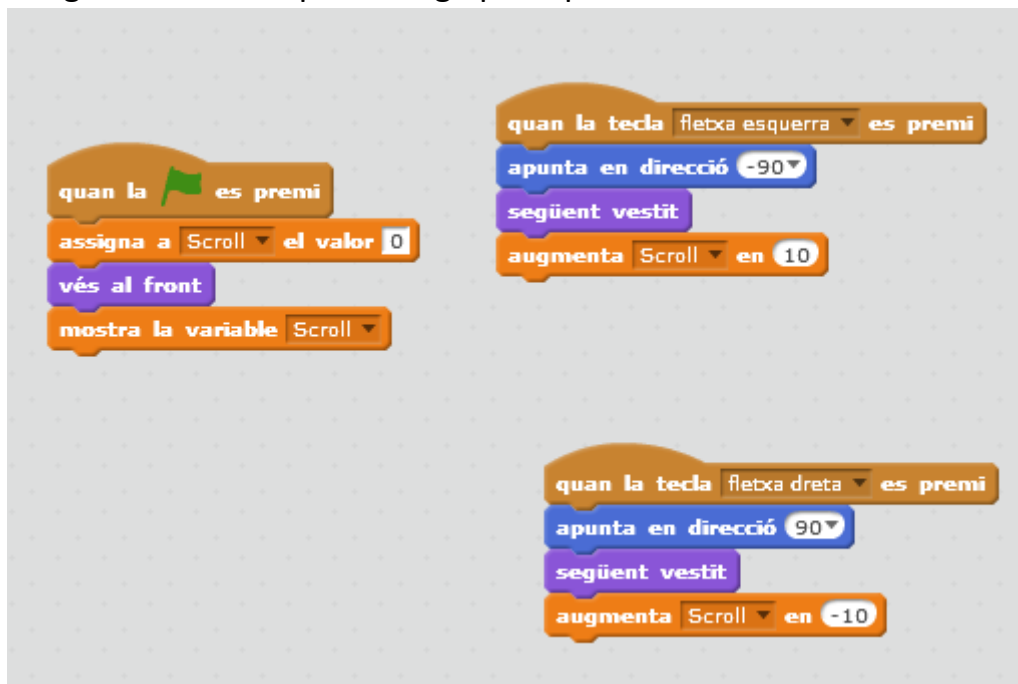
Scroll és el desplaçament de l'escenari a un joc, que dona la impressió al observador de que el personatge principal està en moviment respecte del fons.

Per realitzar aquest efecte a Scratch, hem de crear diversos personatges que actuen com a fons. De manera, que el moviment d'aquests

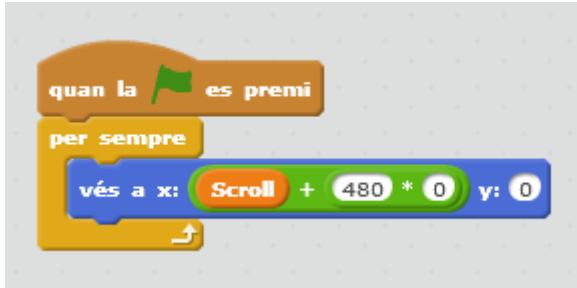


personatges-fons, quede condicionat al "moviment" del nostre personatge.

Una volta creats aquests personatges, utilitzarem una variable (anomenada "Scroll" per ser coherents), la qual condicionarà posició relativa de cada un dels personatges-fons. Per fer-ho inclourem un augment de la variable cada volta que polsem la tecla de "moviment" que assignem al nostre personatge principal.



Si a banda li afegim, al nostre personatge, un canvi de vestuari, podrem crear la ilusió de que aquest estarà en moviment, cada volta que polsem una tecla.



El codi, de cada un dels personatges-fons, serà el mostrar a la esquerra. De manera que com es pot observar, la seua posició quedarà determinada per un valor fixe, producte de la grandària de

l'escenari de Scratch i el nombre de l'orde respecte dels altres personatges-fons, i per una Variable, anomenada "Scroll" que serà la que augmentarà o disminuirà en funció de la tecla de moviment que es premi.

La posició vertical, serà sempre la mateixa en el cas de que només haja un desplaçament horitzontal. En el cas de que volguérem fer també un desplaçament vertical, hauríem de repetir tot el procés incloent també una variable a banda per a aquest desplaçament.

Bloc 6: Jocs intermedis a Scratch. Projecte final.

Aquest document forma part del curs [Pensament computacional a l'aula amb Scratch](#) del [CEFIRE CTEM](#).

Aquesta obra està subjecta a la llicència Reconeixement-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Per veure una còpia d'aquesta llicència, visiteu

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.

Autoria: Santiago Ortega.



1. PONG INTERMEDI

Per fer el nostre joc del “Pong”, un poc més complicat, el que anem a fer es crear un programa, al qual la nostra pilota cree una sèrie de clons de manera progressiva. D'aquesta manera, aconseguirem augmentar la dificultat del joc de manera creixent.

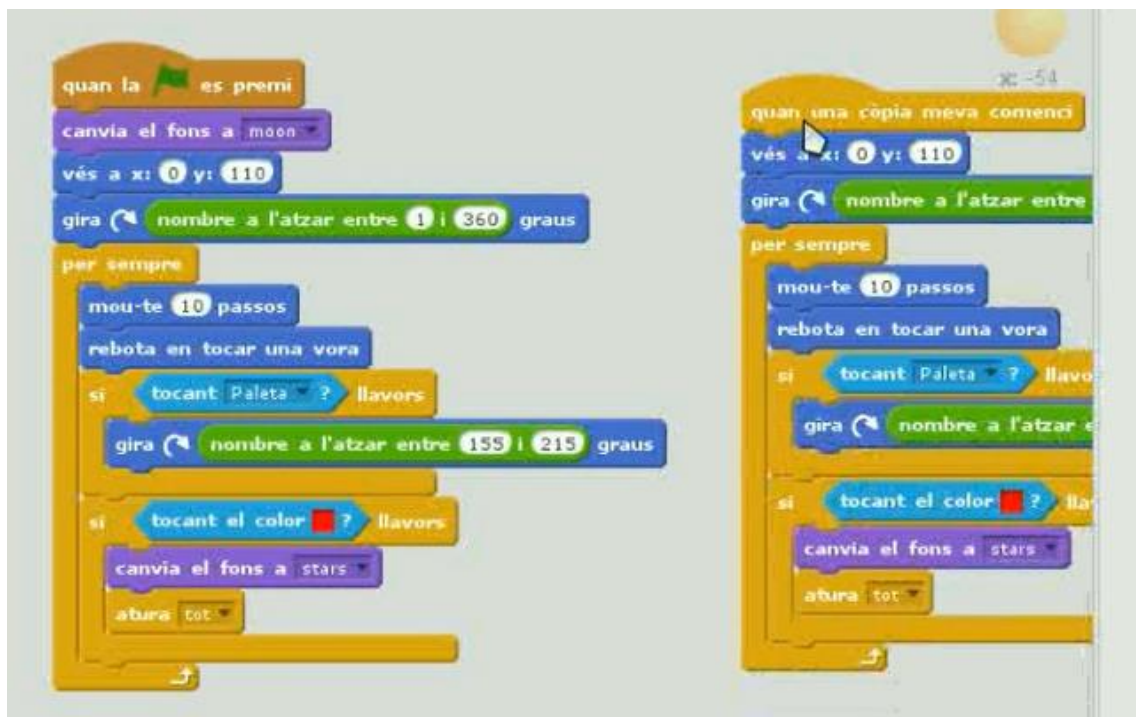
A més, crearem també la possibilitat de fer-lo per a dos jugadors, o de que el jugador haja de utilitzar les seues dos mans per a jugar.

Variable Punts

Crearem un variable anomenada “Punts”, que vaja contant cada volta que la paleta colpege a la pilota.

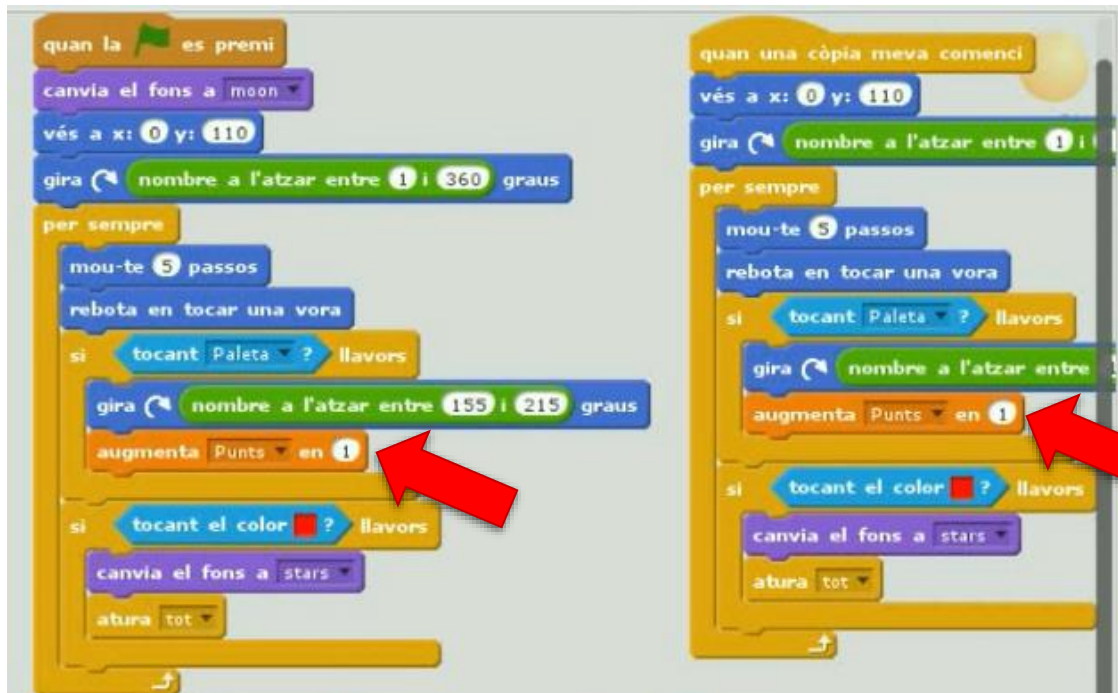


Instruccions dels clons



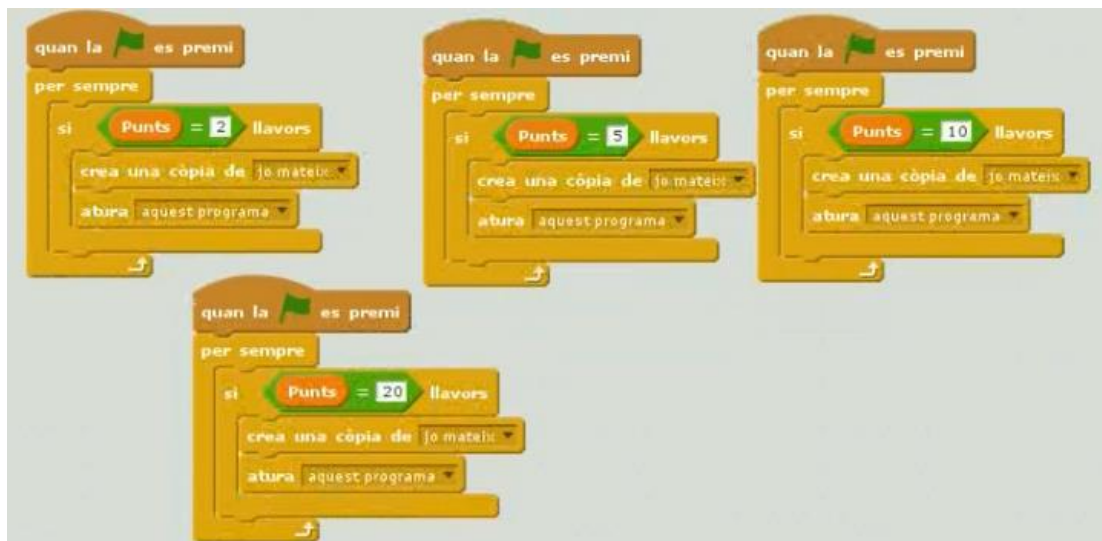
En segon lloc, crearem les instruccions dels clons, les quals, hauran de reproduir el moviment original del nostre objecte inicial (la pilota).

Recordeu que el bloc inicial “Quan una còpia meua comenci”, està situat dins del apartat de “Control” de color groc.



Seguidament, afegirem el Bloc “Augmenta Punts en 1” (fletxa roja) a cada conjunt de blocs creats, tant al original com al conjunt dels clons.

Dificultat creixent

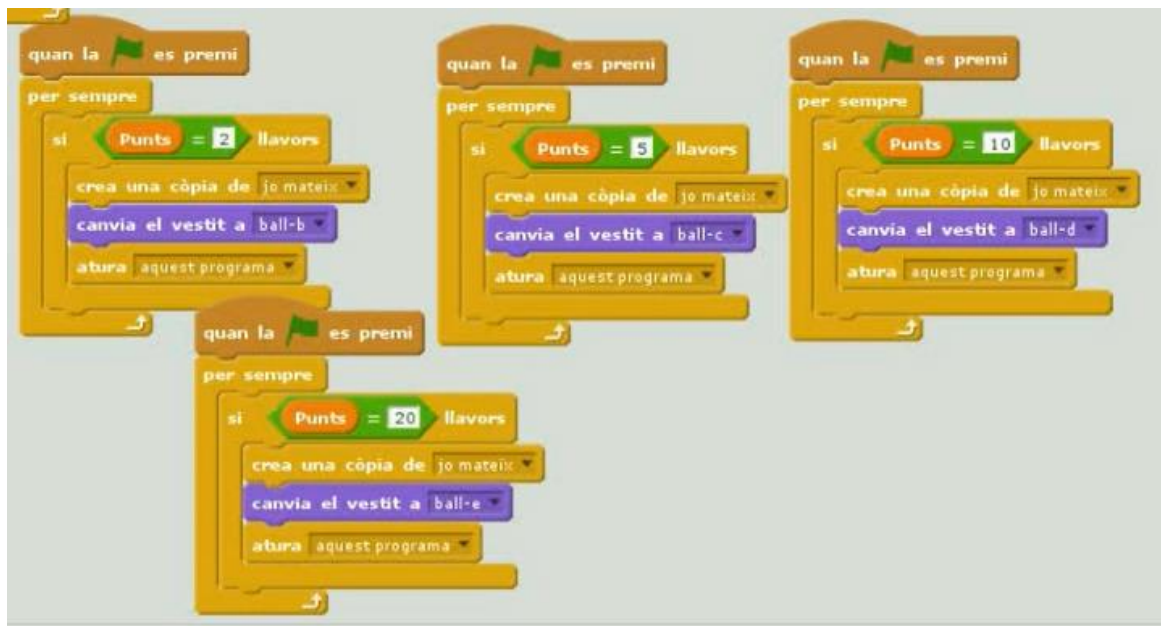


El següent pas, serà crear un conjunt de blocs de forma que, cada un d'ells cree una única còpia del objecte original quan arribe a certa quantitat de punts, i després s'ature. Aquesta última part, s'aconsegueix utilitzant el bloc “Atura”, i seleccionant del desplegable la opció “aquest programa”. De lo contrari, el programa continuarà creant blocs mentre la puntuació

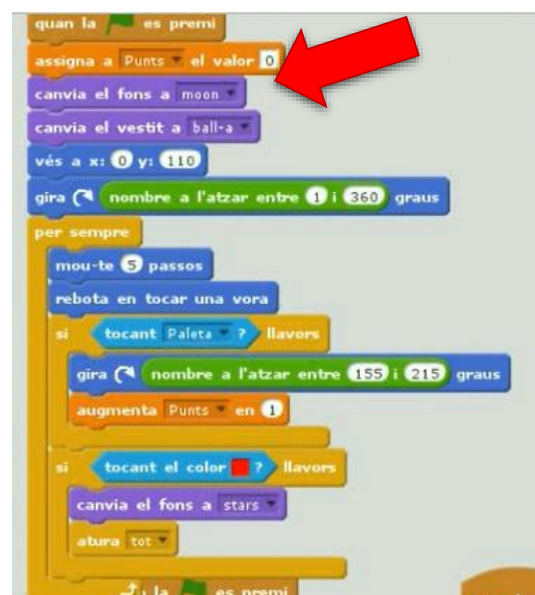
sigui la indicada dalt, i a la llarga Scratch es bloquejarà degut a la quantitat de clons creats, que tendeix al infinit.

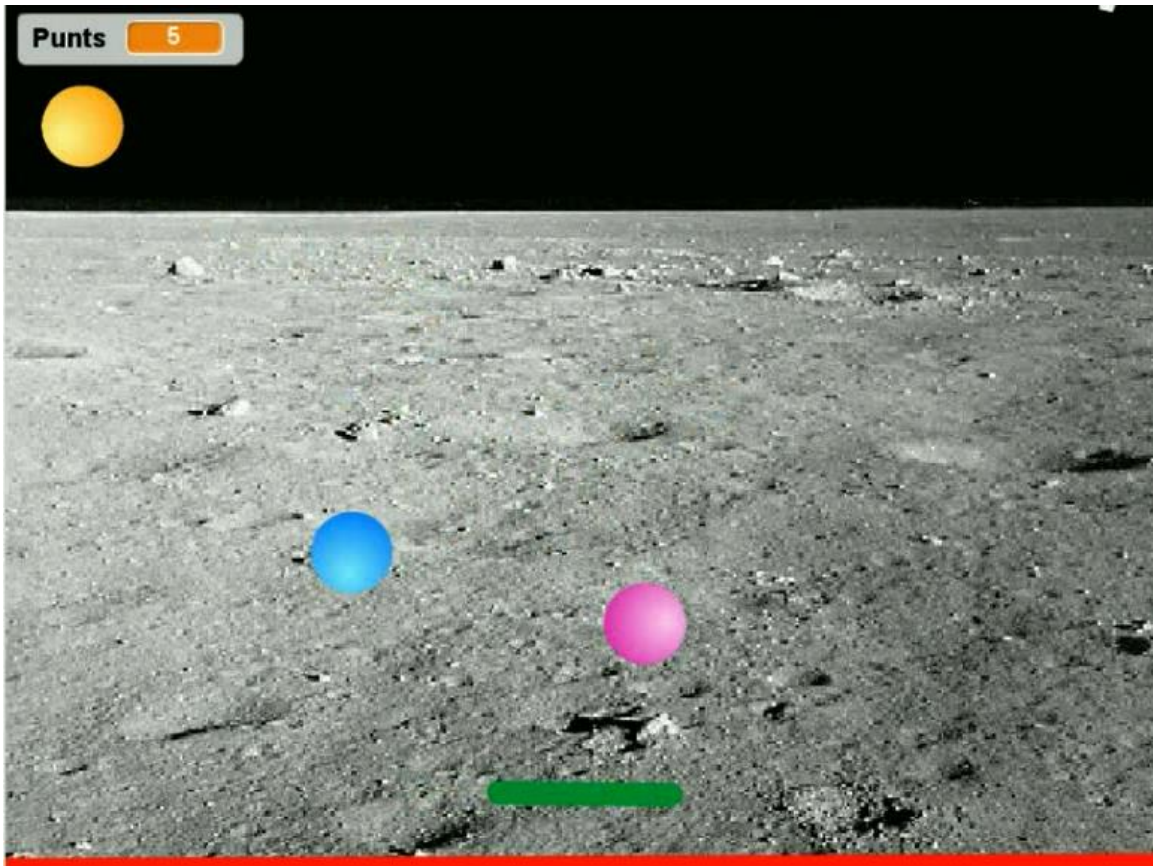
Canvis de vestit en clons

Per a aconseguir un canvi de color a la pilota cada volta que es crea un nou clon, haurem de integrar el bloc de “canvia el vestit a...” dins del conjunt de blocs que ordenaven la creació de còpies. D'aquesta manera, podem aconseguir que cada volta que es cree un clon nou, les pilotes tinguin colors diferents, fent el joc molt més vistós.



A banda, podem afegir un bloc de “Canvia el vestit” al conjunt de blocs del programa de la pilota original, per tal d'indicar-li per quin vestit ha de començar.





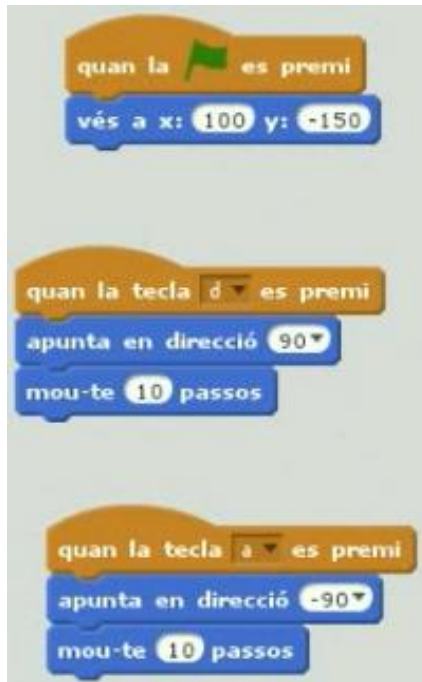
D'aquesta manera, hem aconseguit que cada pilota tinga un color diferent a mesura que es van creant nous clons.

Duplicant la paleta

Si resulta molt complicat el nostre joc, volem afegir una versió per a dos jugadors, o fer que el jugador tinga que utilitzar les dos mans per tal de jugar, el que tenim que fer és duplicar la nostra paleta.

Per fer-ho, haurem de fer clic amb el botó dret sobre la icona de la paleta dins del rectangle de personatges, i al menú desplegable, tornar a fer clic a "duplica".





```
quan la bandera es premi
  vés a x: 100 y: -150

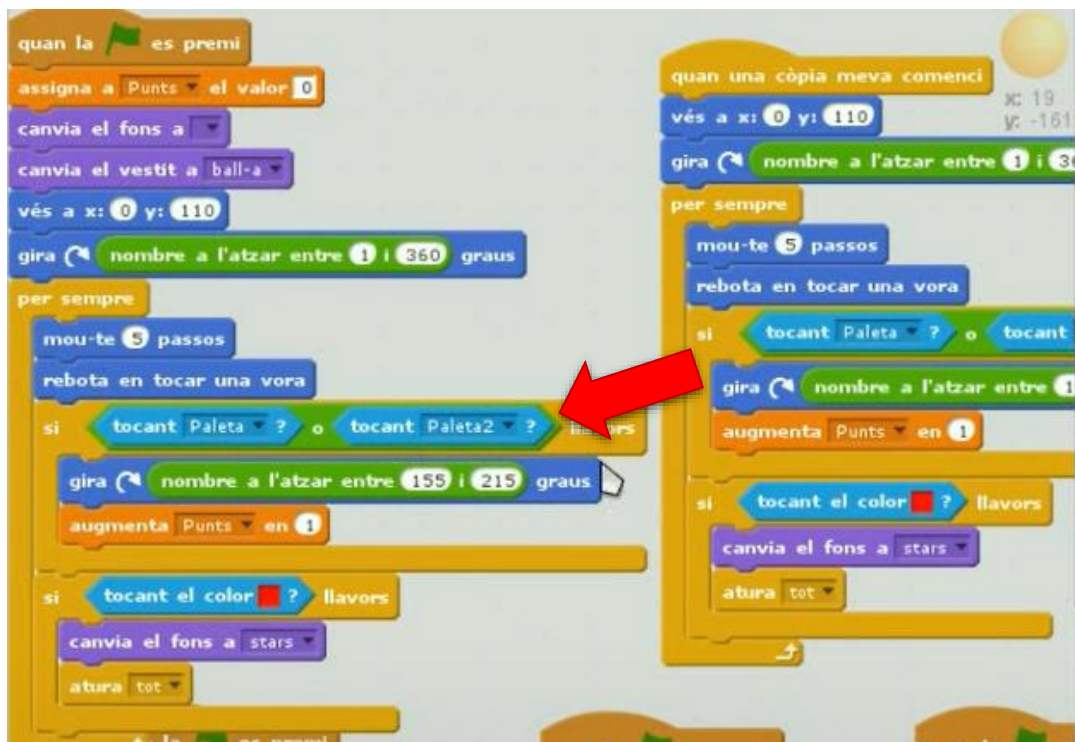
quan la tecla d es premi
  apunta en direcció 90
  mou-te 10 passos

quan la tecla a es premi
  apunta en direcció -90
  mou-te 10 passos
```

El que haurem de fer ara, es modificar les ordres que té la nova paleta, per tal de modificar la posició inicial d'aquesta, així com les tecles que cal estrènyer per accionar el seu moviment.

Conseqüentment, també haurem de modificar la posició inicial de la paleta original, per tal de que queden de la manera més simètrica possible respecte del centre de la pantalla.

A més haurem d'afegir al codi de la pilota original i de les còpies un operador que condicione el rebot a qualsevol de les dos paletes. De lo contrari, només ho farà amb la original, perquè encara que siguin còpies, són objectes diferents.



```
quan la bandera es premi
  assigna a Punts el valor 0
  canvia el fons a
  canvia el vestit a ball-a
  vés a x: 0 y: 110
  gira nombre a l'atzar entre 1 i 360 graus
  per sempre
    mou-te 5 passos
    rebota en tocar una vora
    si tocant Paleta ? o tocant Paleta2 ? llavors
      gira nombre a l'atzar entre 155 i 215 graus
      augmenta Punts en 1
    si tocant el color ? llavors
      canvia el fons a stars
      atura tot

quan una còpia meva comenci
  vés a x: 0 y: 110
  gira nombre a l'atzar entre 1 i 360
  per sempre
    mou-te 5 passos
    rebota en tocar una vora
    si tocant Paleta ? o tocant Paleta2 ? llavors
      gira nombre a l'atzar entre 1 i 360
      augmenta Punts en 1
    si tocant el color ? llavors
      canvia el fons a stars
      atura tot
```

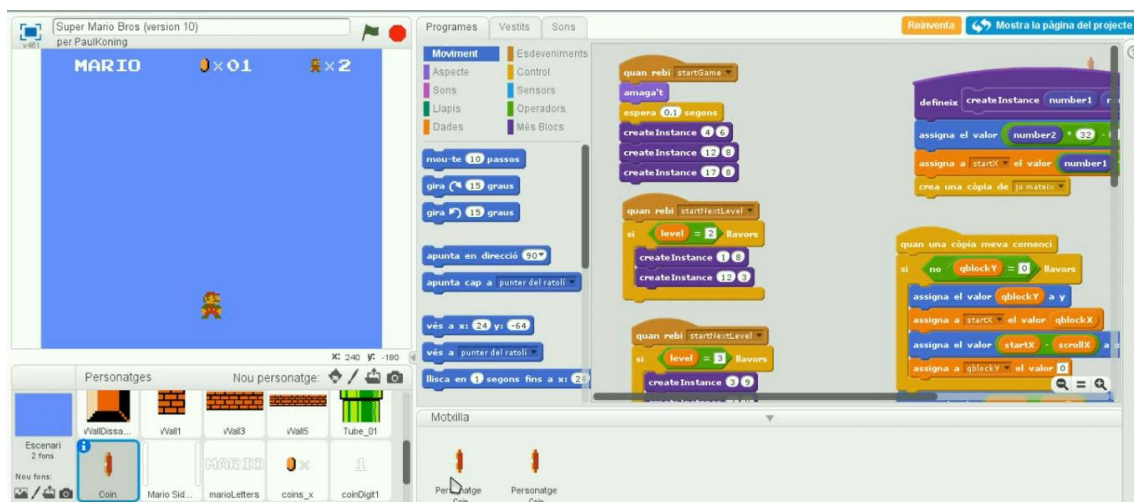
Per fer-ho afegirem el bloc  i duplicarem el sensor de tocar la paleta, per tal de que ho faça amb les dos.

2. CIRCUIT INTERMEDI

Per a evolucionar el nostre joc de carreres, anem a utilitzar la motxilla per copiar un recurs gràfic d'un joc creat per un altre usuari. Per altra banda, crearem una variable, la qual al arribar a una quantitat determinada el joc finalitzi, canviant de fons, i fent que cotxes, monedes i variables desapareguen a la vista.



Copiar les monedes

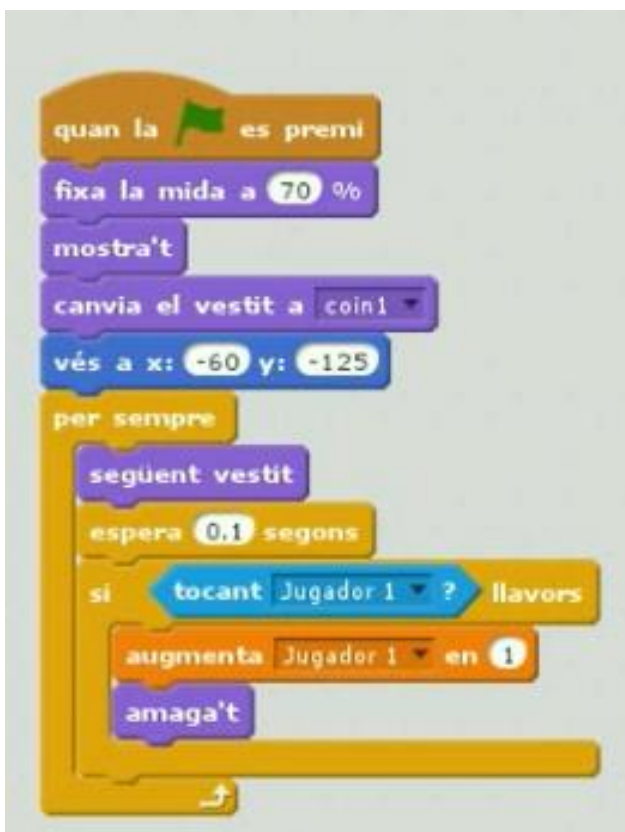


Per copiar aquest recurs, el que hem fet és entrar al joc de Super Mario Bros (version 10), creat per l'usuari PaulKoning amb enllaç <https://scratch.mit.edu/projects/49905542/>

Codi a les monedes

Seguidament, creades les còpies que considerem necessàries de les monedes per duplicat, un per cada jugador. Passarem a establir el seu codi.

El codi creat a les monedes l'hem agrupat en dos conjunts de blocs. El primer conjunt que s'inicia amb l'esdeveniment inicial "quan la bandera es premi" farà lo següent:



- **Canviar la mida** de la moneda a un percentatge determinat.
- **Mostrar-se** (perquè quan acabe el joc s'amagarà).
- **Canviar el vestit** al primer (si desitgem que totes les monedes semblen que giren al mateix temps).
- Anar a una **posició determinada** (aquesta posició serà pròpia de cada parell de monedes).

- Un **condicional "per sempre"**, el qual inclourà els **canvis de vestuari** per crear l'efecte de gir i la **desaparició i l'augment** de la **variable** de cada jugador en el cas de que hi haja contacte amb el seu cotxe. Això sí, com estem creant parells de monedes, una afectarà al jugador 1, i l'altra la jugador 2.ç

Un segon conjunt de blocs, el qual tinga com a esdeveniment inicial "quan rebí missatge1", i seguidament un "amaga't". D'aquesta manera condicionarem la seua presència a la recepció d'un missatge concret.



Aquest últim conjunt de blocs, l'inclouem també als cotxes, per a que així puguem desaparèixer al finalitzar el joc.

Finalització del joc

Per crear l'efecte de la finalització del joc, crearem un conjunt de blocs els quals:

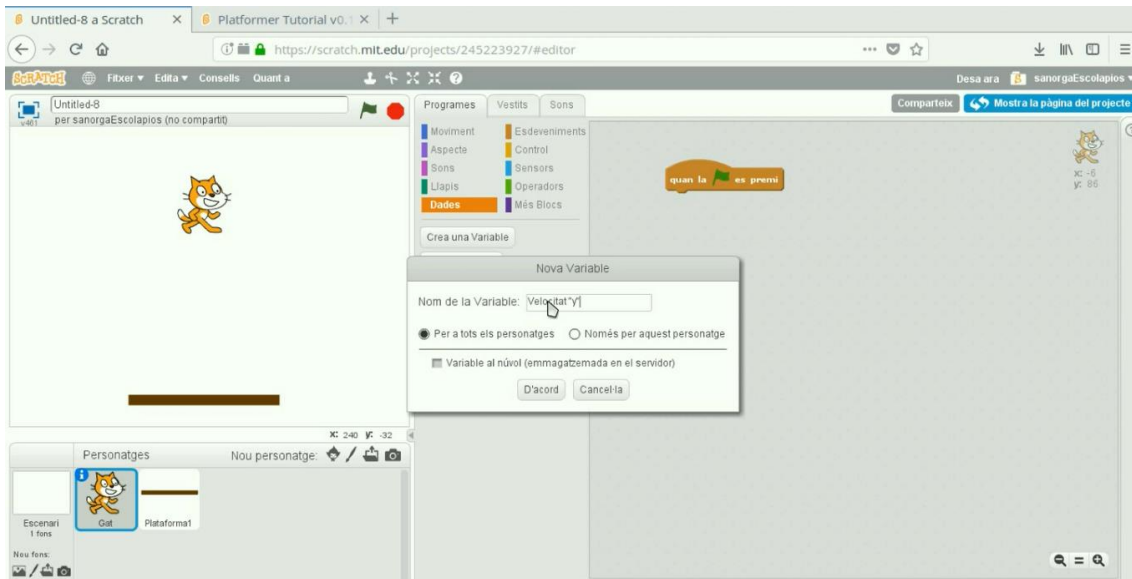
1. Mostren les variables "Jugador 1" i "Jugador 2", a mode de puntuacions.
2. Canviarem el fons al inicial al començar.
3. Crearem un condicional per sempre el qual inclourà dos condicionals "si ____ llavors", els quals condicionaran la puntuació obtinguda per el jugador 1 y el jugador 2 per separat, de manera que al arribar al màxim obtingut per les monedes, canvien a fons diferents, envien un missatge a tots els objectes (que serà el que gastarem perquè s'amaguen), i amagarem les dos variables.



3. PLATAFORMES INTERMEDI

Per crear un joc de plataformes i poder utilitzar després el que coneguem sobre els Scrolls, haurem de començar-lo de nou, i crear les diferents plataformes com objectes independents. Així, en cas de que desitgeu combinar els dos coneixements podreu fer-ho sense problemes.

a



A més, anem a crear l'efecte de la gravetat i de fregament amb les plataformes fent ús de les variables. Per això, crearem dos variables, una que anomenarem "Velocitat y", i un altra que anomenarem "Velocitat x", cadascuna encarregada de l'efecte de la gravetat i del lliscament respectivament.

Efecte gravetat

Per crear l'efecte de la gravetat crearem dos conjunts de blocs, en el primer, assignarem el valor inicial de la variable, i dins d'un condicional de "per sempre", associarem el valor de la variable "Velocitat y" al valor de la posició y del nostre personatge de manera creixent amb el bloc:



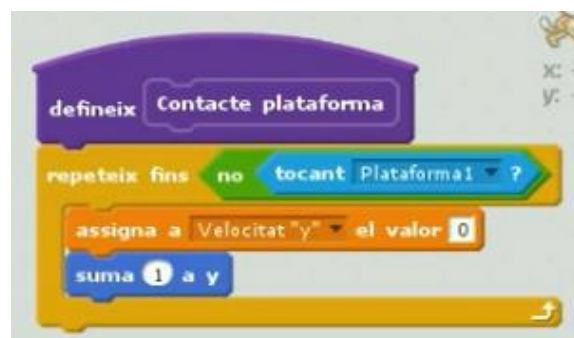
Seguidament, augmentarem el valor de la nostra variable "Velocitat y" en -1, per aconseguir així que per sempre estiga caient. Per últim crearem un bloc nou, per tal de fer que el nostre personatge es detinga i ho faça de manera immediata al tocar la plataforma.



Per tal de crear el nostre Bloc "Contacte plataforma" i crear que l'efecte de parada de la caiguda siga immediat, haurem de fer clic sobre el quadret de :
"Funciona sense refrescar la pantalla"

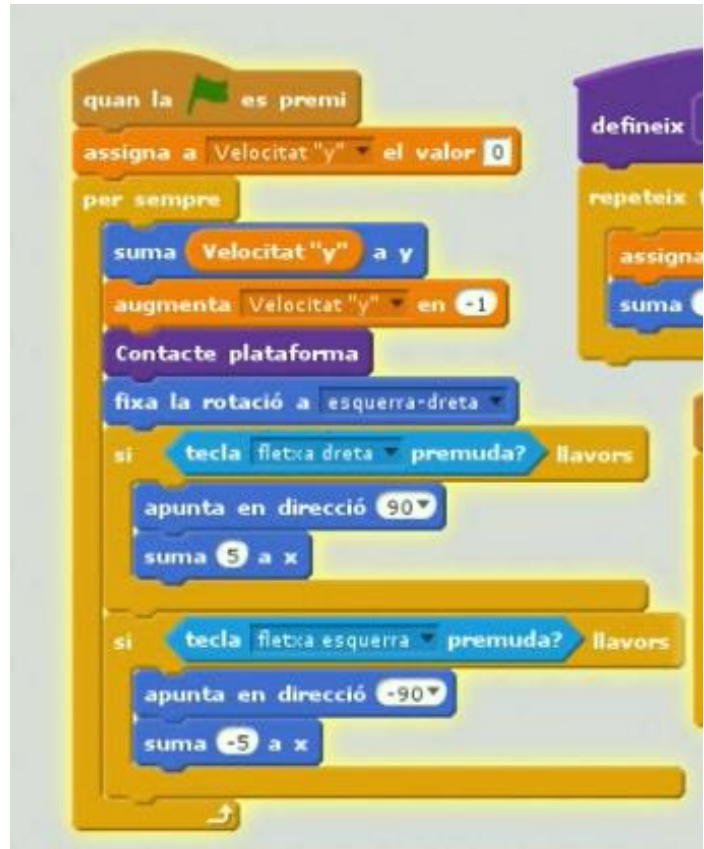
D'aquesta manera aconseguirem que l'efecte no es veja fins que no haja finalitzat.

Per últim, dins del nostre bloc "Contacte plataforma", crearem un condicional "repeteix fins" "no" "tocant Plataforma1", i dins assignarem a la variable "Velocitat y" el valor 0, i li sumarem 1 al seu valor.

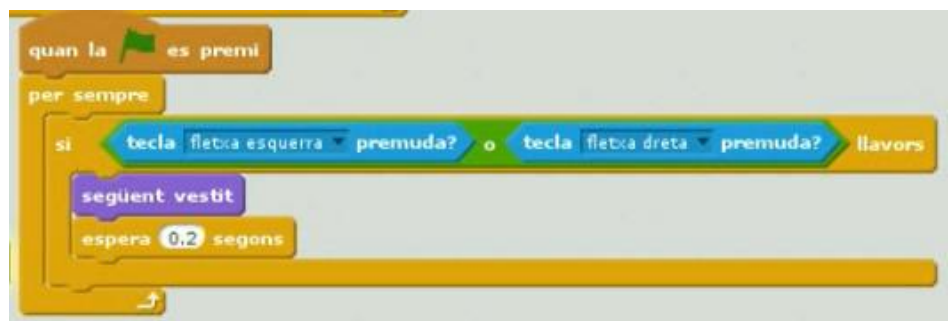


Moviment horitzontal amb tecla premuda

Per crear l'efecte d'un moviment continu quan mantenim pressionada una tecla, utilitzarem el codi de la dreta. De manera que dins del condicional "per sempre", afegirem un "fixa rotació a esquerra-dreta" i dos condicionals "si llavors", que incloguen que quan un sensor detecte que la fletxa esquerra o dreta estiguen premudes, el personatge canvie la seua direcció (90 dreta, -90 esquerra) i sume un valor concret a la x (si a la dreta es positiu, a la esquerra negatiu).



A més, si desitgem que hi haja un canvi de vestuari en el nostre personatge, que no estiga condicionat per la quantitat de moviment, sinó que siga completament independent al mateix, crearem un conjunt de blocs a banda, els quals, dins d'un "per sempre" "si", alguna de les dos tecles de moviments estan premudes, canvie al següent vestit, i s'espere una quantitat de temps determinada per aconseguir l'efecte desitjat de rapidesa entre les passes.



Moviment horitzontal amb lliscament i fregament

Ara el que crearem serà l'efecte de que existeix un lliscament amb les plataformes, de forma que, encara que deixem de polsar les tecles de moviment, el nostre personatge continuarà avançant.



Per fer-ho, crearem un altra variable a la que anomenarem "Velocitat x". Després inclourem al inici del codi del nostre personatge, un bloc que assigne a aquesta variable el valor de 0 al començament del joc. Després associarem el valor de la variable amb el bloc "Suma___ a x". I per últim condicionarem la interacció de cada tecla de moviment amb el augment o disminució de la nostra variable (positiva cap a la dreta, negativa cap a la esquerra).

Posteriorment, afegirem seguidament de les últimes instruccions i dins del condicional de "per sempre" del nostre personatge, un blocs que permeten disminuir el valor de la variable per crear l'efecte de fregament.



Per fer-ho crearem dos condicionals "si...llavors" que augmenten la variable en cas que aquesta siga menys a 0, o que la disminueixen en cas de que siga major a 0.