

SCRATCH



Introduzione a SCRATCH

Stefano Lacaprara
INFN Padova

Logistica corso Scratch

· Docente: Stefano Lacaprarà:

- stefano.lacaprarà@pd.infn.it 0499677100

· Segreteria: Maristella Zuin

- Maristella.zuin@pd.infn.it Tel 0499677126

· Web page:

- <http://www.pd.infn.it/~lacaprar/ProgettoScuola/Scratch/>

- Contiene links alla pagina web di scratch,

- Documentazione varia,

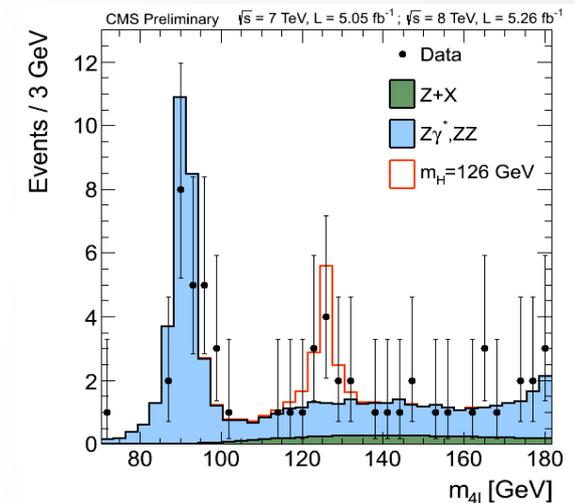
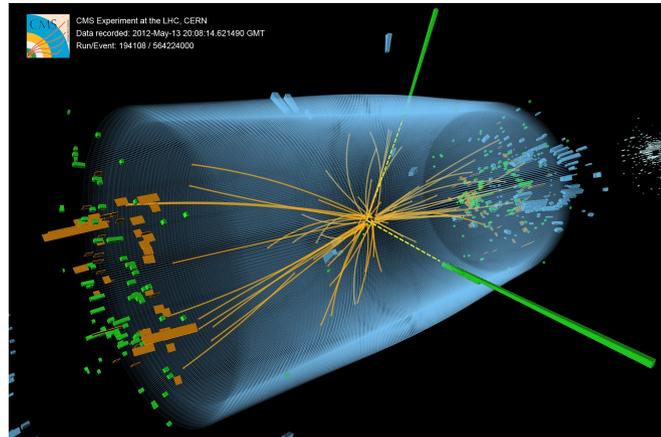
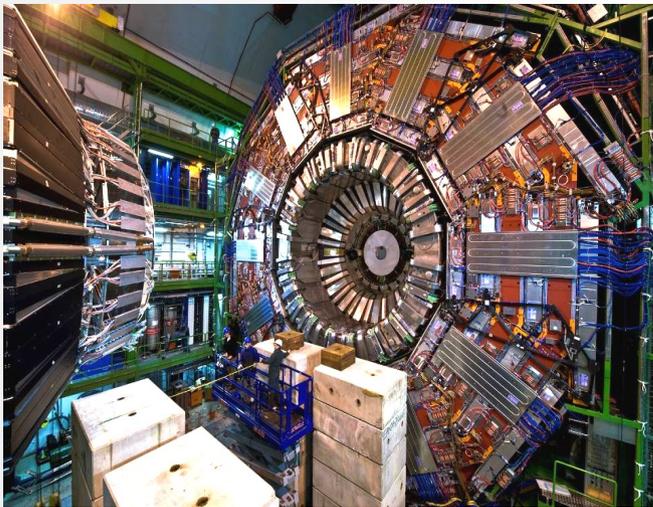
- Le slide e relativi progetti scratch saranno caricati dopo ogni incontro

- Bibliografia

-

Mi presento

- Sono un ricercatore dell'Istituto Nazionale di Fisica Nucleare (INFN)
- Lavoro nel campo delle particelle elementari, nella collaborazione CMS, al CERN di Ginevra



- Oltre al lavoro di fisico, e strumentale a quello, ho sviluppato per molti anni software a vari livelli sia direttamente per la ricostruzione della fisica degli apparati sia per il calcolo distribuito
- Ho insegnato elettromagnetismo e programmazione (C++) all'universita' di Padova e Milano Bicocca
- Tengo questi incontri per la prima volta.

Perche' questo corso

- Viviamo in un mondo scientifico/tecnologico, ma l'interesse per la scienza/tecnologia non e' diffuso come potrebbe e dovrebbe.
 - Questo e' un problema per la scienza e quindi per il mio campo specifico,
 - Ma e' un problema anche per il nostro futuro
- Perche' succede?
 - Ci sono tante cause, una di queste e' la mancanza di stimoli adeguati nell'eta' dell'apprendimento
- La presenza di scienza e tecnologia a scuola e' purtroppo limitata, non ultimo per questioni di risorse
 - Spesso e' presentata con materiali poveri e in ambiti non immediatamente collegati alla vita tecnologica che i bambini vivono.

- Esempi di esperimenti di scienza con bicchieri di carta, carrellini, fagioli, etc, quando i bambini usano tutti i giorni computer, smartphone, telecomandi, ...
 - Si crea una distanza tra quello che un bambino percepisce come scienza e quello che la scienza è veramente.
 - Da questa distanza, l'iniziale interesse e curiosità che hanno tutti i bambini verso tutto, e quindi anche verso scienza e tecnologia, può calare.
 - Per diventare indifferenza o, peggio, avversione.

Prenderli da piccoli

- Non e' solo un cliché', ma una vera necessita'.
- Diversi studi (bibliografia e' disponibile sul sito web, gran parte in inglese) mostrano che un bambino “decide” molto presto se ha interesse per scienza e tecnologia (e matematica e tutto quello che gli sta attorno)
- L'attitudine verso la scienza e' un atteggiamento che il bambino acquisisce (o non acquisisce) presto, tipicamente tra i 8 e gli 11 anni, e tipicamente non lo cambiera' piu'.
 - Se decide che non gli interessa, decidera' anche inconsciamente che la scienza, matematica, informatica, etc e' troppo difficile, e non la imparera' in futuro.
 - C'e' inoltre una grande differenza di genere, tra maschi e femmine, che e' probabilmente la singola variabile piu' importante in relazione all'attitudine verso la scienza
 - Oltre elementi culturali, sociali, familiari etc, etc...

Obiettivi

- **Primo obiettivo:** vorremmo portare nelle scuole degli esempi di scienza e tecnologia che siano attuali, moderni.
- **Secondo obiettivo:** vorremmo rendere consapevoli i bambini di come funziona alcuni pezzi della scienza e tecnologia che li circonda, in modo da suscitare loro l'interesse per diventare un giorno attori attivi nel creare tecnologia, non solo utilizzarla/subirla.
- **Infine:** vorremmo cercare di valutare l'impatto della nostra attivita', in modo quantitativo, per capire se e'

Il nostro progetto

- La nostra idea e' di cercare di portare nelle scuole, elementari e medie, qualche esempio di scienza e tecnologia moderna, cercando di:
 - Stimolare l'interesse per queste materie
 - Mostrare che dietro la quotidiana dose di tecnologia che i bambini vedono c'e' un mondo interessante e divertente di quelli che costruiscono quella tecnologia.
 - L'idea e' di avere laboratori trasportabili (*valigette*) che un tutor possa portare presso le classi interessate per svolgere attivita' di laboratorio.
 - Elettronica, meccanica, informatica, fisica, ...
 - In parallelo, coinvolgere ricercatori di diverse materie come **science ambassadors** per raccontare ai bambini cosa e' la scienza di frontiera, la ricerca.
 - Infine, cercare di valutare l'impatto di queste attivita' in modo scientifico.
- Per sviluppare il progetto stiamo cercando fondi, ma vorremmo iniziare con quello che abbiamo per verificare se la cosa e' possibile e attuabile

Informatica

- Se c'è un singolo campo onnipresente nella nostra vita oggi è l'informatica.
 - Nel bene (e nel male!) il software è una parte grandissima della nostra vita
 - È una parte fondamentale della vita di qualunque scienziato
- Programmare è una attività creativa!
 - C'è uno stereotipo imperante che associa un programmatore ad un nerd, o disadattato sociale o comunque ad una figura piuttosto negativa.
- Programmare vuol dire analizzare e risolvere problemi:
 - Per risolverli, occorre pensare, decidere, e valutare i nostri sforzi.

Nativi digitali

- Spesso si parla dei bambini di oggi come nativi digitali, perche' sanno usare iCosi o computers prima e a volte meglio di noi.
 - E' una definizione assolutamente infelice, perche' da' l'impressione che i bambini controllino il mondo digitale. Non e' cosi'.
- Certo, sanno usare videogiochi o smartphone gia' a 3 anni: ma si chiedono cosa c'e' dentro un computer?
 - Si rendono conto che un computer (o cellulare o telecomando o ...) e' stato costruito e programmato da qualcuno?
 - Hanno mai provato a programmare un computer?
 - O si limitano ad usare passivamente cose preparate da altri?

Obiettivi progetto scratch

- L'obiettivo primario di questo progetto è dare ai bambini la consapevolezza che è possibile interagire con gli strumenti tecnico informatici che li circondano non solo in maniera passiva, ovvero semplicemente usandoli, ma anche in maniera attiva, cioè programmandoli direttamente.
- L'attività vuole essere presentata come un gioco, da fare singolarmente e collettivamente, produrre un risultato tangibile in tempi brevi e allo stesso tempo suscitare la curiosità del bambino ai temi dell'informatica, della programmazione, e, più in generale, ad interessarsi a cosa c'è dentro, in questo caso a livello software, i vari strumenti tecnologici che lo circondano.
- L'analogo hardware può essere coperto da altri progetti orientati all'elettronica, meccanica.

Cosa e' SCRATCH

- Impara a programmare mentre crei storie interattive, giochi, progetti multimediali.
- *Start from **scratch*** vuol dire *partire da zero*
 - E' una delle idee di base di scratch: non serve saper programmare
- La programmazione e' completamente grafica: non occorre sapere nulla di programmazione formale, ma si deve solo costruire strutture semplici (o piu' complesse) mettendo insieme mattoncini colorati.
 - E' pensata come un gioco, ma richiede che il bambino pensi a cosa vuole fare e sia in grado di dividere azioni complesse in pezzi elementari.
 - E' fortemente multimediale

Cosa si puo' fare

- Qualche esempio di progetti con SCRATCH
 - Personaggi che si muovono e interagiscono su un palcoscenico,
 - I personaggi interagiscono con l'utente:
 - Sei tu che comandi il computer
 - Interazioni tra personaggi
 - Slideshow interattivo,
 - Storie interattive
 - Simulazioni
 - Videogames,
 - ...
- Ci sono infinite applicazioni che fanno queste cose, ma il punto fondamentale e' che sara' il bambino a programmare il computer, a comandare il computer per fargli fare quello che vuole.

Cosa serve

- **Un PC: non serve particolarmente nuovo**
 - monitor: 800 x 480 maggiori (tutti)
 - Sistema operativo: Windows 2000 o superiori, Mac OS X 10.4 o superiori, Ubuntu Linux 9.04 o superiori
 - Altre versioni di linux sono supportate
 - La versione per linux e' sperimentale: funziona bene, ma ogni tanto puo' crashare.
 - Disco: 120 Mb liberi
 - CPU e RAM: se riesci a far funzionare il S.O., allora tipicamente il PC e' sufficiente. Computer molto vecchi (e lenti) potrebbero eseguire SCRATCH lentamente.
 - Suono/Video: altoparlanti o cuffie e microfono (per registrare), webcam per video/foto. Anche via USB

Pagine web

- Pagina principale:
<http://scratch.mit.edu/>
 - Scaricare il programma,
 - documentazione,
 - Forum degli utenti (c'e' anche sezione in italiano, non frequentatissima)
 - migliaia di progetti caricati dagli utenti
 - I progetti si possono eseguire sul web (richiede flash)
 - O scaricare ed eseguire e **modificare** localmente



The screenshot shows the Scratch website homepage. At the top, there's a navigation bar with links for 'home', 'projects', 'galleries', 'support', 'forums', and 'about'. Below this is the Scratch logo and a search bar. The main content area features a large banner with the text 'Create and share your own interactive stories, games, music, and art' and a 'Download Scratch' button. Below the banner, there are sections for 'Featured Projects' and 'Projects Selected by joletole'. The 'Featured Projects' section includes 'Root Words', 'Learning About Ancient Egypt', and 'Attack of the Frogs!'. The 'Projects Selected by joletole' section includes 'WorldCraft v3.4', 'Fallin' Lazaa' 2', and 'Fruit nina demo!'. On the right side, there are sections for 'Scratch 2.0 Beta', 'ScratchEd', and 'Video Tutorials'.

Pagine web

- <http://scratched.media.mit.edu/>
 - Pagina dedicata agli educatori. Gran parte in inglese
 - Storie, news
 - Risorse (documentazione, tutorial, corsi, ...)
 - Discussioni (forum)
 - Eventi (Workshop, Meetups, Webinars, ScratchDay: [in passato ne e' stato organizzato uno a Verona])



The screenshot shows the ScratchEd website homepage. The header features the ScratchEd logo with the tagline "learn | share | connect" and navigation links for "Join", "Sign In", "About", "Help", and "Contact". A search bar is located on the right. Below the header is a navigation menu with tabs for "Stories", "Resources", "Discussions", "Members", and "Events". The main content area includes a "What is Scratch?" section with a video player, a "What is ScratchEd?" section, and a "Get Started with Scratch" section. The bottom of the page is divided into three columns: "New Stories", "New Resources", and "New Discussions", each containing a list of recent posts with thumbnails and brief descriptions.

Installazione

- La versione attuale di SCRATCH e' la 1.4
 - http://info.scratch.mit.edu/it/Scratch_1.4_Download
 - Per windows:
 - Scaricare <http://download.scratch.mit.edu/ScratchInstaller1.4.exe> (33MB)
 - Doppio clic sul .exe e seguire le istruzioni:
 - Windows si lamentera' che l'eseguibile non e' formato e potrebbe essere pericoloso. Cliccare su “ok – installare lo stesso – mi fido di questa sorgente”
 - Dobbiamo poter scrivere nella cartella dove stiamo estraendo scratch: se ci sono errori (error opening file for writing), provare ad estrarre scratch in **desktop**
 - L'eseguibile si trova dentro la cartella **scratch** (e nel menu start)
 - Per Mac/OS
 - <http://download.scratch.mit.edu/MacScratch1.4.dmg>
 - Spostare scratch dentro cartella applicazioni
 - Per Ubuntu
 - Da Software Center: cerca SCRATCH e installa (servono diritti di amministratore)
 - Oppure si scarica pacchetto .deb (e relative dipendenze)

Partiamo

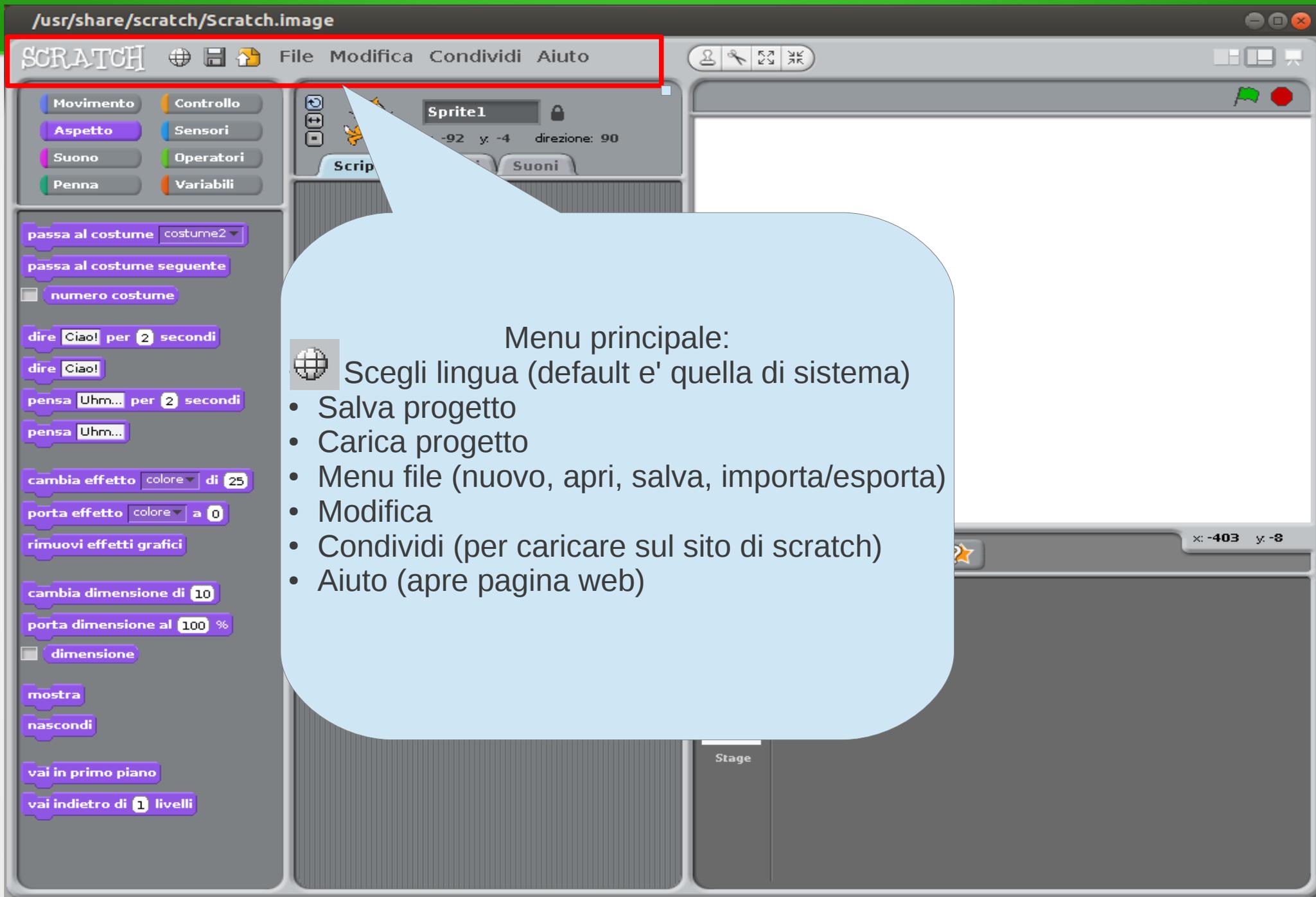


Getting Started with
SCRATCH
version 1.4



<http://scratch.mit.edu>

La schermata di SCRATCH



The image shows the Scratch application window. The title bar reads "/usr/share/scratch/Scratch.image". The menu bar includes "SCRATCH", a globe icon, a save icon, a load icon, and the menu items "File", "Modifica", "Condividi", and "Aiuto". A red box highlights the menu bar. A light blue callout box points to the globe icon in the menu bar, containing the following text:

Menu principale:

- Scegli lingua (default e' quella di sistema)
- Salva progetto
- Carica progetto
- Menu file (nuovo, apri, salva, importa/esporta)
- Modifica
- Condividi (per caricare sul sito di scratch)
- Aiuto (apre pagina web)

The background shows the Scratch workspace with a sprite named "Sprite1" and various code blocks in the left sidebar.

La schermata di SCRATCH

Palcoscenico (stage)

- Dove andremo a vedere i risultati dei nostri programmi
- Gli attori sono gli “sprite” (figurine?)
 - Per esempio questo gatto
- I tasti verde e rosso in alto a destra serviranno per far partire e fermare la scena
-

La schermata di SCRATCH

Gli sprite

- E' l'elenco di tutti gli sprite che ho creato
 - In questo caso, solo uno
- In piu' c'e' lo stage, che non e' uno sprite, ma puo' essere comandato: le scene

La schermata di SCRATCH

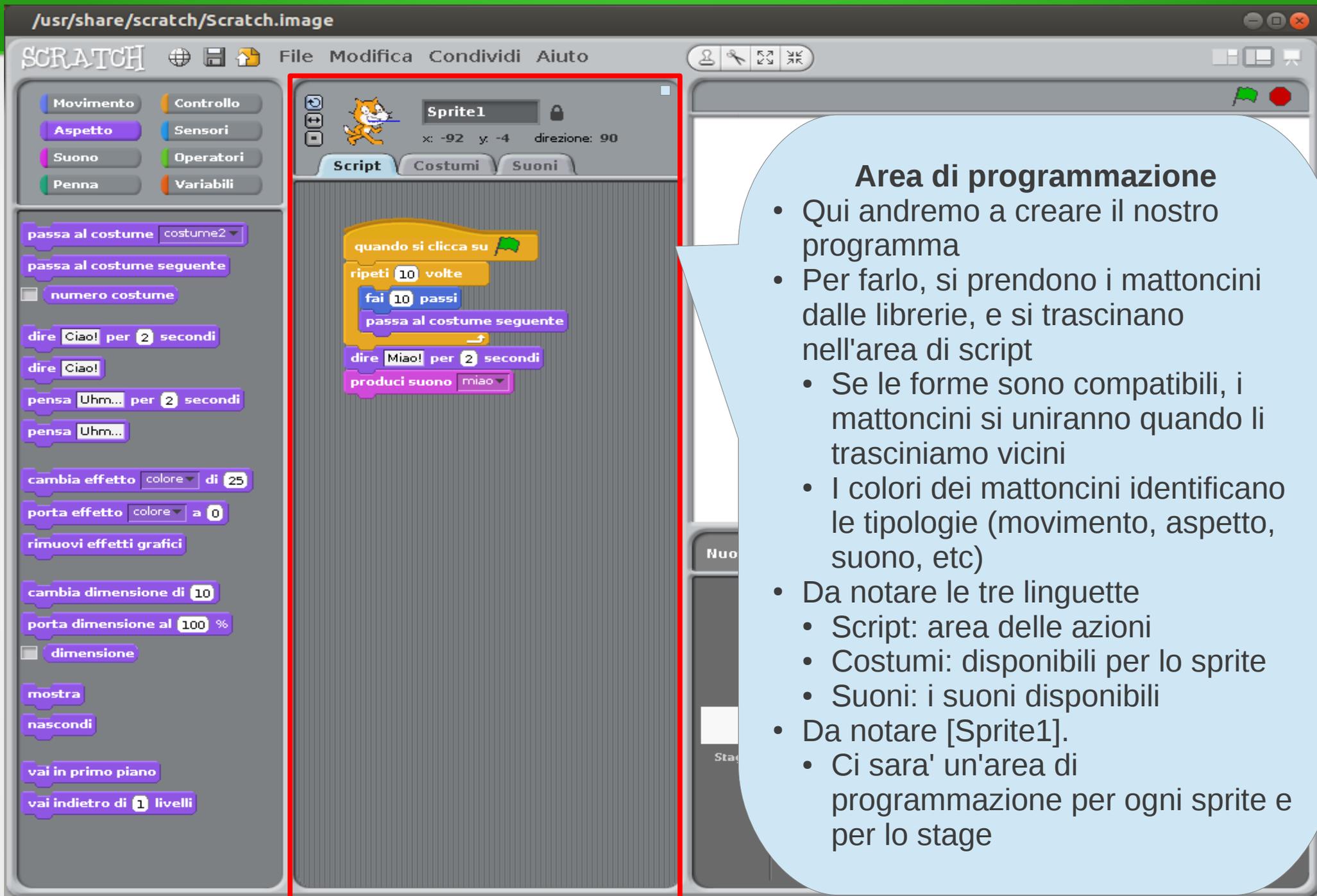


The screenshot shows the Scratch interface with the 'Libreria delle azioni' (Action Library) highlighted in red. The library is organized into categories: Movimento, Controllo, Aspetto, Sensori, Suono, Operatori, Penna, and Variabili. The 'Libreria delle azioni' is the first category, containing various action blocks such as 'passa al costume', 'dire', 'cambia effetto', 'cambia dimensione', 'mostra', 'nascondi', 'vai in primo piano', and 'vai indietro di livelli'.

Libreria delle azioni

- Ci sono le azioni che gli sprite possono compiere
 - Sono i mattoncini elementari dei nostri programmi.
 - Un programma complesso sarà una costruzione di mattoncini elementari
- Sono divise per tipologia
 - Movimento, aspetto, suono, etc
 - Ogni categoria ha diversi elementi
 - Ogni elemento ha una forma particolare che si può incastrare o meno con altri elementi
 - Alcuni elementi hanno al loro interno "spazio" per parametri oppure per altri elementi
 - Sarà più chiaro usarli

La schermata di SCRATCH



The screenshot shows the Scratch interface with a red box highlighting the script area for a sprite named 'Sprite1'. The script area contains the following code:

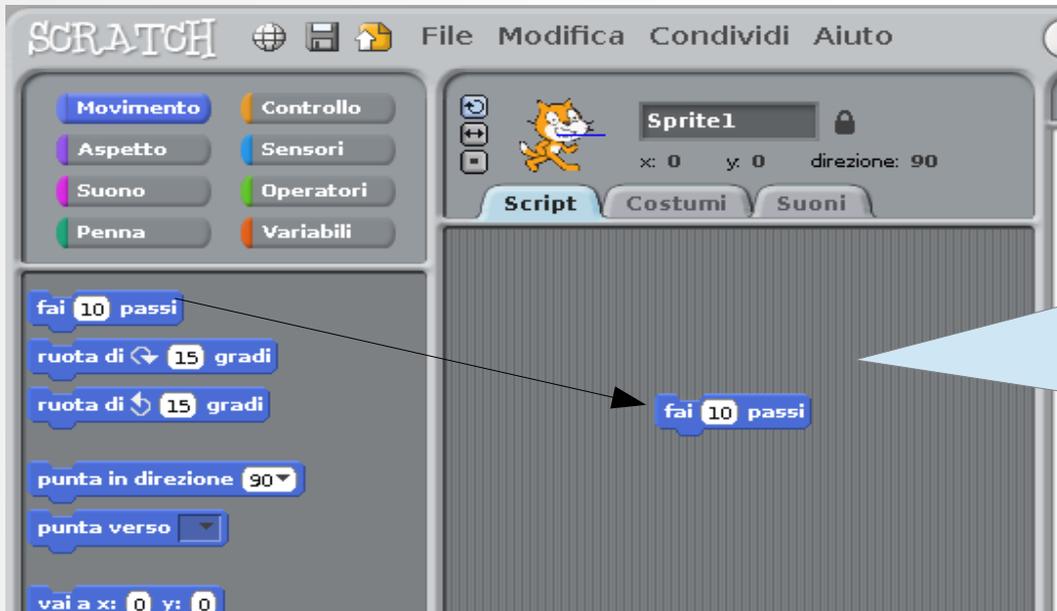
```
quando si clicca su [bandierina verde]
  ripeti 10 volte
    fai 10 passi
    passa al costume seguente
  dire "Miao!" per 2 secondi
  produci suono [miao]
```

The interface also shows the left sidebar with various categories: Movimento, Controllo, Aspetto, Sensori, Suono, Operatori, Penna, and Variabili. The script area has three tabs: Script, Costumi, and Suoni.

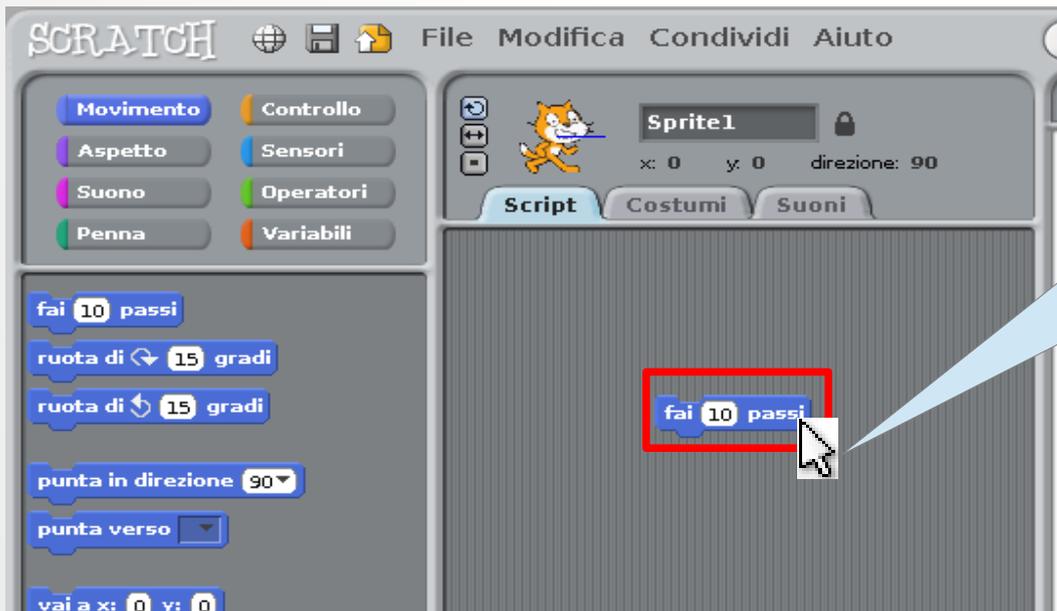
Area di programmazione

- Qui andremo a creare il nostro programma
- Per farlo, si prendono i mattoncini dalle librerie, e si trascinano nell'area di script
 - Se le forme sono compatibili, i mattoncini si uniranno quando li trascini vicini
 - I colori dei mattoncini identificano le tipologie (movimento, aspetto, suono, etc)
- Da notare le tre linguette
 - Script: area delle azioni
 - Costumi: disponibili per lo sprite
 - Suoni: i suoni disponibili
- Da notare [Sprite1].
 - Ci sarà un'area di programmazione per ogni sprite e per lo stage

Iniziamo a programmare

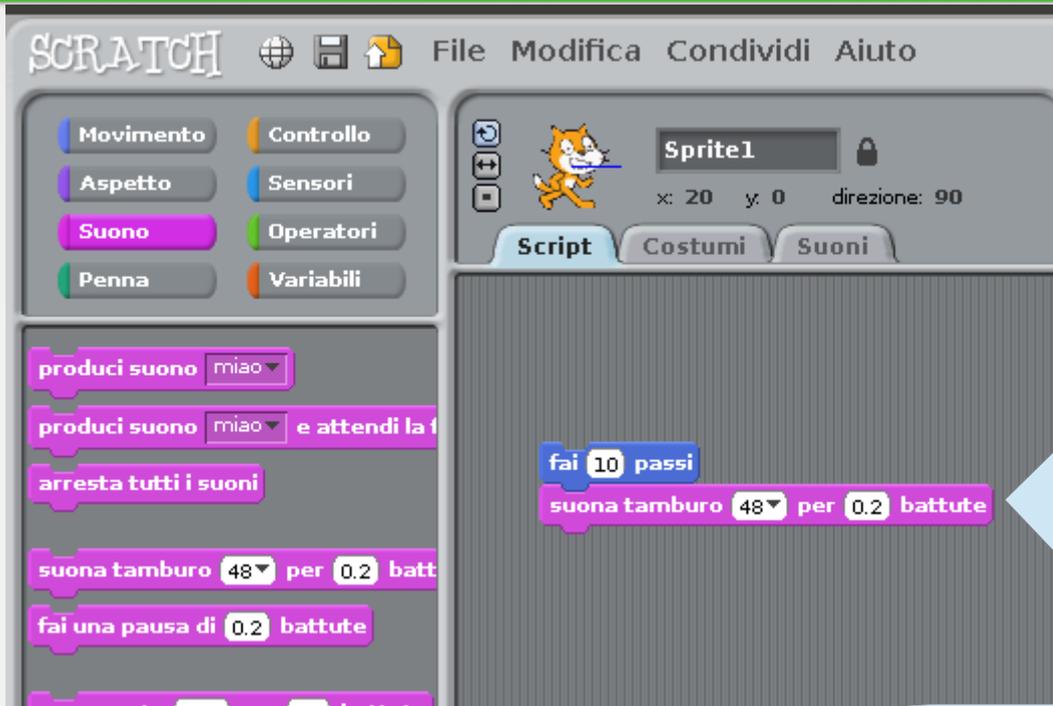


Trascina un mattoncino
fai 10 passi
nell'area di programmazione

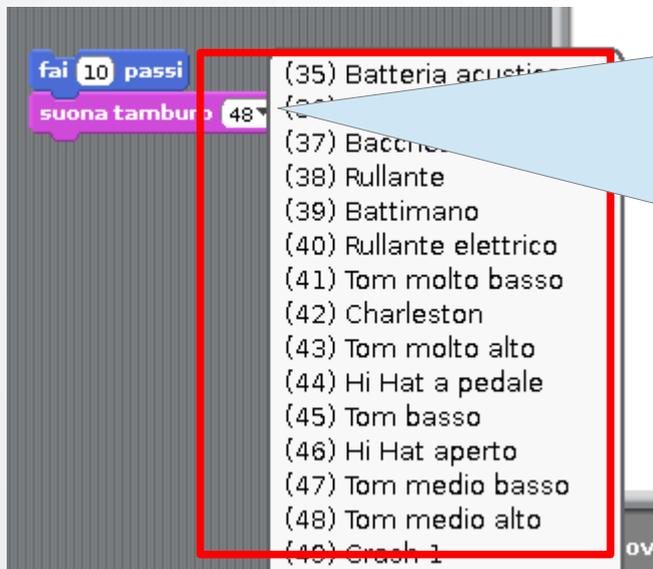


Click su mattoncino per far
muovere il gatto di 10 passi

Aggiungiamo un suono

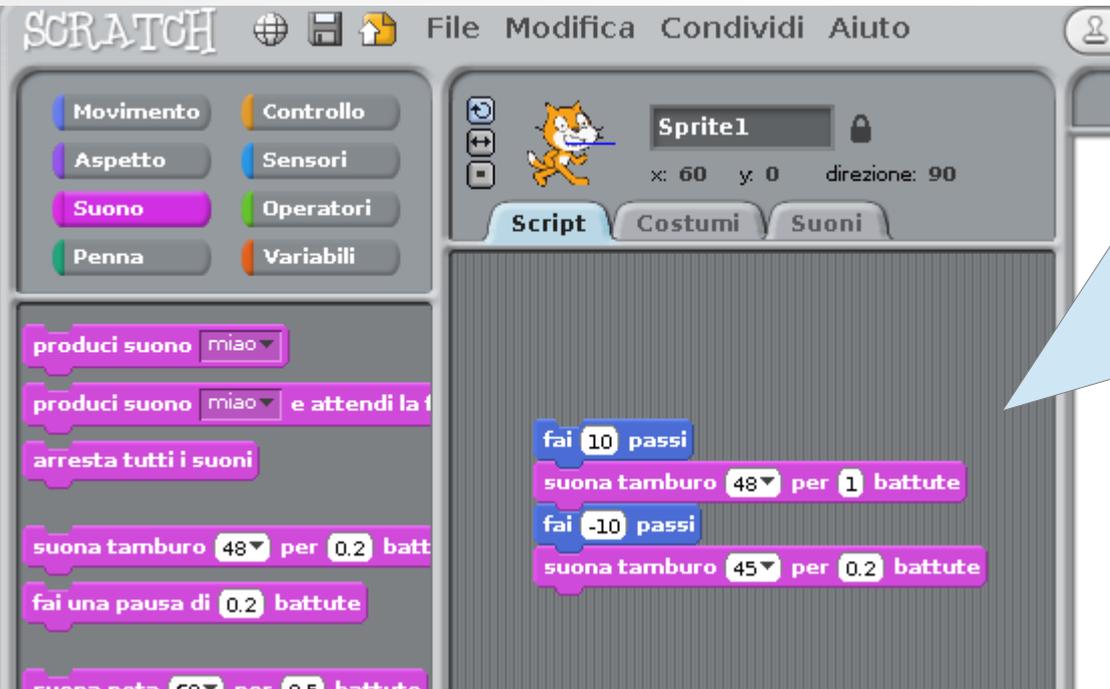


- Dalla libreria **Suono** (viola) trascina
- **Suona tamburo 48 per 0.2 battiti**
- nell'area di programmazione appena sotto l'altro mattoncino.
- I mattoncini si incastrano perche' sono compatibili.
- Ho creato un programma con due azioni:
 - Fai 10 passi
 - Suona il tamburo
- **Click** su mattoncino per sentire il tamburo



- Alcune azioni (mattoncini) sono configurabili, cioè posso cambiare alcuni parametri dell'azione stessa.
- Area bianca dentro il mattoncino:
 - Fai **N** passi: default e' 10, ma posso cambiarlo
 - Suona tamburo **X**. Se clicco su X(default e' 48), mi compare una lista di tutti gli strumenti che posso suonare.
 - Per **Y** battute: per quanto tempo suonarlo.

Primi passo di danza



- Ho aggiunto:
 - **Fai [-10] passi**
 - -10 vuol dire tornare indietro
 - **Suona altro tamburo**
- Adesso il mio gatto fa un passo avanti, suona un tamburo, fa un passo indietro e suona un tamburo diverso.
- Click su qualunque mattoncino per far partire il programma.
- **Pero' lo fa una volta sola!**
 - Potrei ripetere i 4 mattoncini ma non e' cosi' che si programma.

Introduco il concetto di **Iterazione (loop)**.

Ho un insieme di azioni che voglio ripetere N volte:

Raggruppo le azioni

Le inserisco dentro un modulo (mattoncino speciale) che ha come compito di ripetere una azione (o un insieme di azioni) N volte.

Indipendentemente dal tipo di azione

Ripetere i passi



- Dalla libreria **Controllo** prendo
- **Ripeti [10] volte**
- Attenzione alla forma del mattoncino:
 - Ha una tacchetta in alto, e una in basso, come i mattoncini di prima:
 - Si puo' incastrare prima o dopo gli altri mattoncini.
 - Ha spazio tra le due *ganasce* e le tacchette mostrano che puo' accogliere uno o piu' mattoncini.
 - Mentre lo trascino nell'area di programmazione, la forma cambia per poter accogliere i 4 mattoncini che ho gia' nell'area.



Ripetere i passi all'infinito

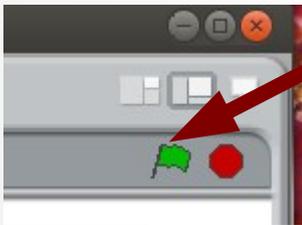
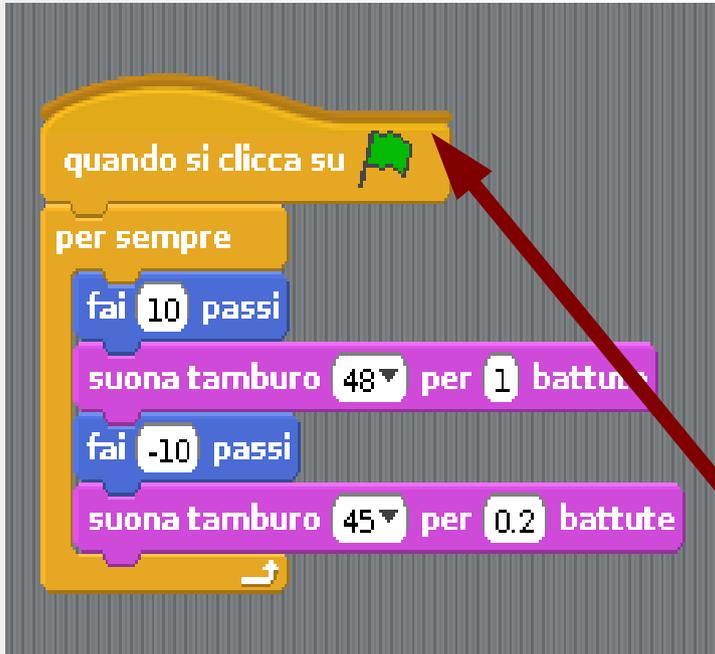


- Dalla libreria **Controllo** prendo
 - **Per sempre**
 - E' un iteratore (ripete le azioni dentro le ganasce)
 - E le ripete per sempre (non per N volte)
- Attenzione alla forma del mattoncino:
 - E' come **ripeti [10] volte** ma non ha la tacchetta in fondo
 - Visto che e' per sempre, non e' possibile fare un'altra azione dopo!
- Per cambiare [ripeti 10 volte] con [per sempre]:
 - Trascino fuori i mattoncini dalle ganasce
 - Poi li metto dentro l'altro loop
- Per fermare [per sempre] click su bottone rosso in altri a destra



Far partire il programma

Bandiera Verde



- Per partire un programma in **[controllo]**, c'e' il mattoncino
- **Quando si clicca su [bandiera verde]**
- Attenzione alla forma del mattoncino:
 - Si possono attaccare mattoncini dopo, ma non prima.
- Ho un programma completo:
 - C'e' un bottone di partenza (e uno di arresto)
- **Proviamo a introdurre piu' interazione**

Interazione



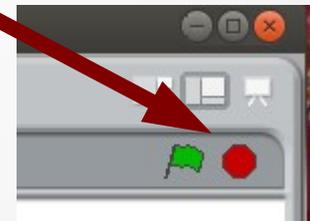
- Introduco un secondo programma
 - Indipendente dal primo (non c'e' collegamento tra i mattoncini)
- Viene attivato da un mattoncino controllo
 - **Quando si preme il tasto [spazio]**
 - Posso decidere di cambiare tasto
 - Azione e' cambiare **colore** di [25]
 - Ci sono diversi effetti grafici che si possono applicare su uno sprite
- Che cosa ci aspettiamo che succeda?

- Quando premo il tasto **verde**, il **programma 1 parte**, e lo sprite inizia a ballare.
- Se poi, in qualsiasi momento premo il **tasto spazio**, allora parte anche il **programma 2** (in parallelo) e lo sprite cambia colore.
- **Ho un programma che reagisce a stimoli esterni (tastiera): e' un programma interattivo!**

Ripetere i passi all'infinito



- Dalla libreria **Controllo** prendo
- **Per sempre**
- E' un iteratore (ripete le azioni dentro le ganasce)
- E le ripete per sempre (non per N volte)
- Attenzione alla forma del mattoncino:
 - E' come **ripeti [10] volte** ma non ha la tacchetta in fondo
 - Visto che e' per sempre, non e' possibile fare un'altra azione dopo!
- Per cambiare [ripeti 10 volte] con [per sempre]:
 - Trascino fuori i mattoncini dalle ganasce
 - Poi li metto dento l'altro loop
- Per fermare [per sempre] click su bottone rosso in altri a destra



Piu' sprite!

- ▶ Fino ad ora un solo sprite:
 - ▶ Posso usarne piu' di uno.
- ▶ Per aggiungerli, tre possibilita':



Disegna un nuovo sprite

- ▶ Si apre una finestra dove posso disegnarlo (simile a paint)



Seleziona nuovo sprite da file

- ▶ Scratch ne ha molti disponibili, divisi per categoria (animali, fantasia, lettere, ...)

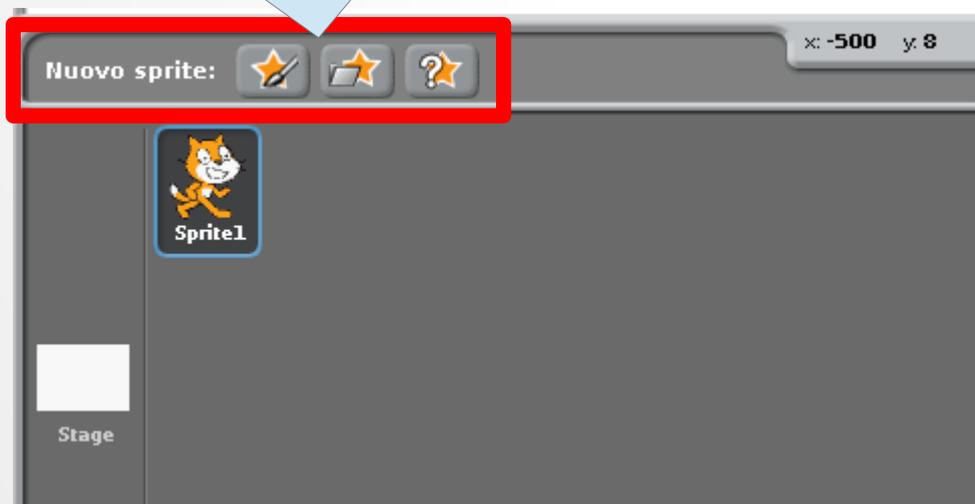


Uno sprite a sorpresa

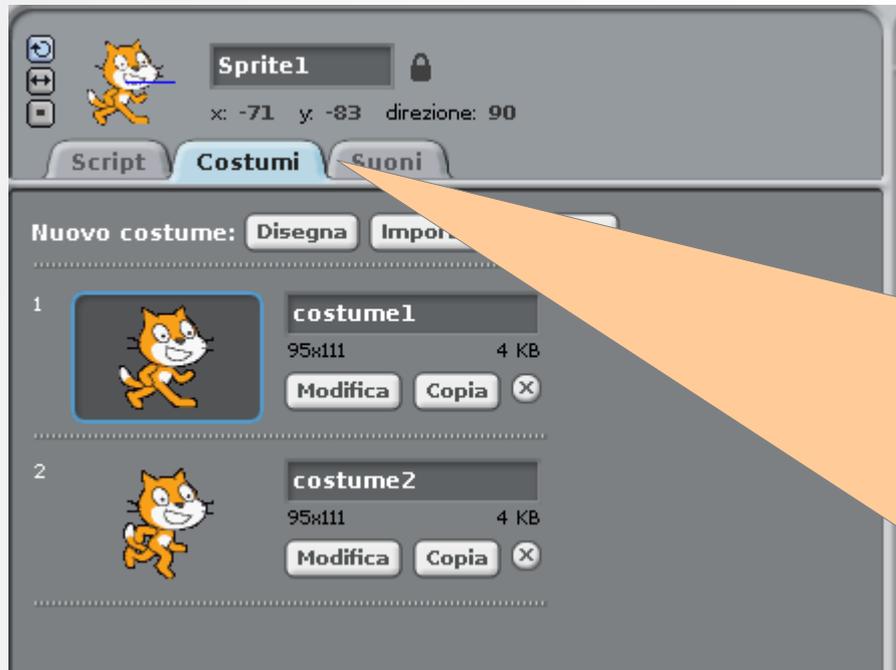
- ▶ Lo pesca tra quelli disponibili



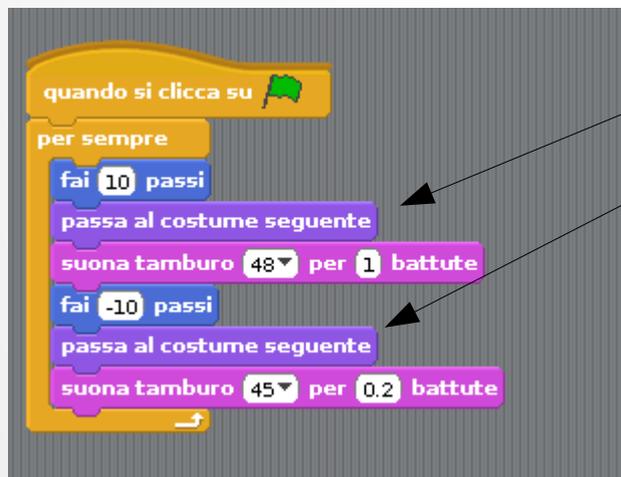
- **Adesso ho due sprite**
 - Il secondo l'ho disegnato io
- **Ogni sprite ha la sua area di programmazione:**
 - Il programma che abbiamo scritto prima appare solo se selezione lo sprite gattino (sprite1)
 - Il secondo sprite non ha ancora programmi associati.



Animazione



- Per animare uno sprite, si possono usare i **costumi**
- Ogni sprite puo' avere diversi costumi, che possono essere variazioni dello sprite originale
 - Per esempio figure vestite in modo diverso
 - Oppure figure in posizioni diverse
- Il gattino ha due costumi: alternandoli, possiamo animare lo sprite per farlo camminare.
- Possiamo aggiungere altri costumi a piacere, disegnandoli da zero, o copiando uno gia' esistente e modificandolo.



Lo sfondo (stage)



- Lo stage e' gestito come uno sprite speciale
 - Si possono definire script (programmi) per lo sfondo
 - Solo un insieme limitato di azioni sono possibili, per esempio, nessuna di movimento
 - Puo' avere diversi sfondi (analogo di costumi)
 - Si possono associare diversi suoni



Costumi e sfondi

- Costumi o sfondi si possono:
 - Disegnare
 - Importare: una immagine già esistente (es foto)
 - Acquisire con webcam collegata al pc (se presente)



- Per creare uno sprite da webcam, occorre:
 - Prima disegnarne uno nuovo “finto”
 - Poi si seleziona lo sprite nuovo, e si sceglie [costumi]
 - Qui si crea un nuovo costume selezionando [web cam]
 - Volendo si modifica l'immagine acquisita
 - A questo punto si puo' cancellare il primo costume “finto”

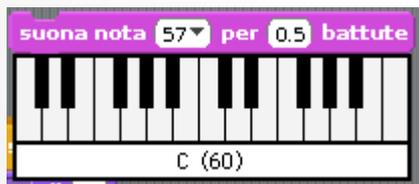


Suoni

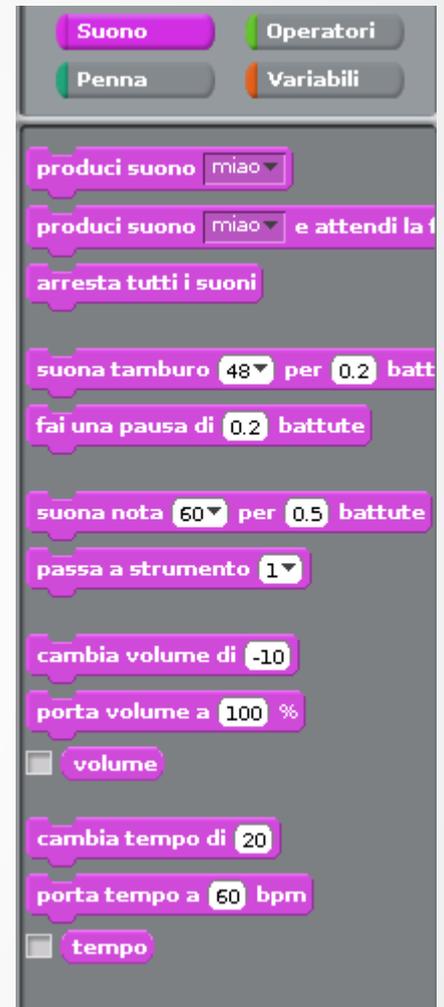
- Ci sono diversi modi di far suonare scratch

- Uno l'abbiamo già visto: la libreria **[suono]**

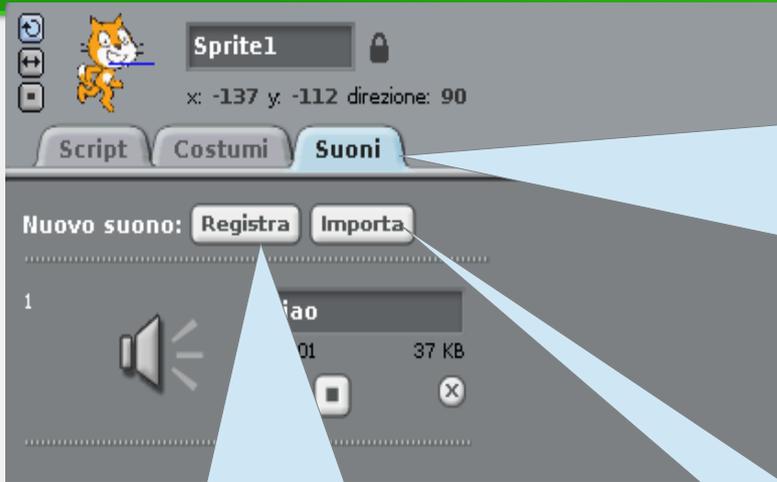
- **Suona tamburo** si possono scegliere diversi strumenti
 - **Suona nota** c'è una tastiera con due ottave e nomenclatura anglosassone (A=La, B=Si, C=Do, ...) compresi # e b



- **Produci suono** ogni sprite ha un insieme di suoni che si possono scegliere o estendere



Registrazione suoni



- Nella linguetta **Suoni** nell'area di programmazione di uno sprite (o dello stage) c'è l'elenco dei suoni disponibili.
- Per esempio il gattino ha disponibile un **miao**, che posso ascoltare cliccando su 

- Se clicco su **[registra]** si apre il programma per registrare suoni del mio PC, e posso registrare.
 - Serve ovviamente un microfono (integrato o esterno)
 - **Attenzione:** su linux (ubuntu) a volte questo fa crashare scratch. Meglio salvare il progetto prima!

- Si possono anche importare suoni **[importa]**
 - Scratch ha una libreria di suoni già pronta, divisa per tipi
 - Oppure posso usare altri file audio (wav, mp3)

Riassumendo: abbiamo imparato a

- Creare programmi per uno sprite
 - Movimento
 - Costumi
 - Suoni
 - Interazione
 - cicli
- Gestire piu' di uno sprite
- Gestire sfondi

Proviamo a mettere tutto insieme

- **Un nuovo progetto:**

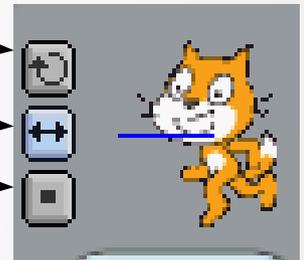
1. Proviamo a creare un gattino che cammina avanti e indietro per lo schermo, girandosi e miagolando quando arriva al bordo.
2. Usiamo uno sfondo appropriato: anzi, piu' d'uno. E vogliamo poter scegliere tra 2 diversi sfondi usando le frecce destra e sinistra della tastiera
 1. Magari mettiamo anche una musica di sottofondo adeguata e diversa per i due sfondi.
3. Infine, mettiamo un ballerino che danza sullo sfondo, che cambia colore ogni volta che il gattino tocca il bordo dello schermo

Analisi del programma

1. Dovreste già saperlo fare.

1. Vi manca il fatto di girarsi quando si arriva al bordo, ma c'è un mattoncino apposta **rimbalza quando tocchi il bordo**
2. Attenzione che “rimbalzare” vuol dire non solo destra-sinistra, ma anche sopra-sotto, quindi al primo rimbalzo il gattino si troverà capovolto. Per evitarlo usano i bottoni nell'area di programmazione, vicino allo sprite

1. Può ruotare
2. **Voltati solo destra e sinistra**
3. Non ruotare



2. Facile:

1. Unica cosa è comandare il cambio sfondo con i tasti freccia destra e sinistra
 2. Possiamo far suonare la musica agli sfondi.
- ## 3. È un po' più complicato: mancano alcuni pezzi
1. segue

Azioni condizionali e coordinazione

- Dovete far compiere ad uno sprite una azione quando qualcosa succede ad un altro sprite:
- scomponiamo il problema in due parti:
 - Uno sprite deve “parlare” all'altro sprite
 - (“Hey, io ho toccato il bordo!”),
 - l'altro deve ascoltare e
 - quando sente il messaggio giusto fare quello che deve fare.
 - **Messaggistica**
- Il primo sprite deve parlare solo quando qualcosa accade:
 - **azione condizionale**

Inviare messaggi

- Se ci sono due persone che devono coordinare una qualche attività, bisogna che si parlino.
 - Per esempio, quando il primo ha finito la sua parte, dice al secondo “ho finito” e allora il secondo può iniziare a fare la sua parte.
- Anche in programmazione è così.
 - In Scratch ogni sprite può inviare a tutti un messaggio usando  e inserendo un nome per il messaggio (qui “bordo”)
 - Gli altri sprite ascoltano tramite 
 - Ovviamente devono “ascoltare” la stessa “parola” che viene inviata dall'altro sprite (bordo)
 - I mattoncini che seguono questo controllo, vengono eseguiti dopo che è arrivato il messaggio “bordo”

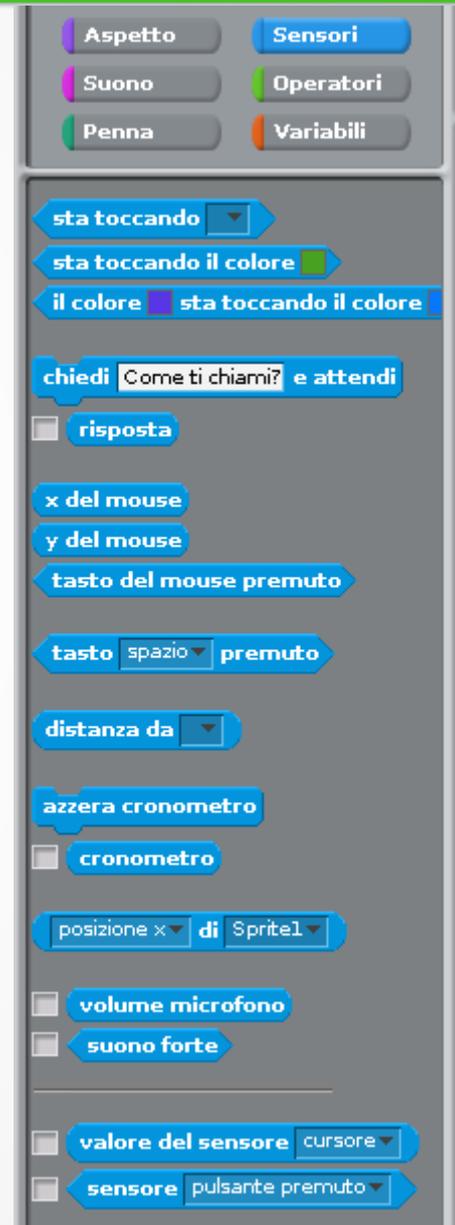
Azione condizionale

- E' uno degli elementi base di qualunque programma. Devo fare qualcosa quando una certa condizione e' vera.
- Ci sono molti modi diversi per farlo.
 - Usiamo
 - **Per sempre quando <succede qualcosa>**
 - Da notare sempre la forma del mattoncino:
 - Puo' seguire o precedere qualsiasi altro mattoncino,
 - Tra le ganasce puo' accogliere uno o piu' mattoncini (come il ciclo **per sempre**)
 - C'e' un “buco” dopo il se con una forma particolare. Qui dobbiamo mettere la **condizione** che si dovra' verificare perche' le azioni tra le ganasce vengano eseguite.



Condizioni

- Si trovano nella libreria **[sensori]**
- Hanno una forma particolare, con gli estremi spigolosi
 - **<sta toccando []>**
 - La forma e' quella adatta per inserirsi nel buco di **[per sempre quando < >]**
 - Le opzioni di **<sta toccando>** sono [bordo, mouse, altri sprite]. Quello che ci serve.



Le prossime volte

- Nuovi progetti: cartoline multimediali, presentazioni interattive, giochi...
- Introduzione agli operatori
- Introduzione alle variabili
- Piu' interazione con utente: il mouse
- Numeri casuali
- ...