

# UNIVERSITÀ DELLA CALABRIA



*Dipartimento di MATEMATICA e INFORMATICA*

*Tesi finale del corso Percorsi Abilitanti Speciali (PAS)*



**SCRATCH**

*per iniziare a programmare*

*Relatore PROF.<sup>ssa</sup> SIMONA PERRI*

*Candidato PROF. SERGIO PORCO*

*Anno Accademico 2013/2014*

## **Indice**

|                           |   |
|---------------------------|---|
| <b>Introduzione</b> ..... | 1 |
|---------------------------|---|

### **Contesto e progettazione**

|                              |   |
|------------------------------|---|
| <i>Identificazione</i> ..... | 2 |
| <i>Prerequisiti</i> .....    | 2 |
| <i>Contenuti</i> .....       | 2 |
| <i>Obiettivi</i> .....       | 2 |
| <i>Metodologia</i> .....     | 3 |
| <i>Strumenti</i> .....       | 3 |
| <i>Tempi</i> .....           | 3 |
| <i>Verifica</i> .....        | 3 |

### **Organizzazione dell'unità Didattica.**

|   |    |
|---|----|
| 1 <i>Introduzione a Scratch</i> .....               | 4  |
| 2 <i>L'interfaccia di Scratch</i> .....             | 5  |
| 3 <i>Conosciamo la schermata di Scratch</i> .....   | 7  |
| 4 <i>Conosciamo i blocchi per programmare</i> ..... | 9  |
| 5 <i>Iniziamo a programmare</i> .....               | 11 |
| 6 <i>Esercitazioni e verifiche</i> .....            | 20 |
| 7 <i>Sitografia</i> .....                           | 21 |
| 8 <i>Conclusioni</i> .....                          | 21 |

## **Introduzione**

*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 chiunque. Spesso si parla dei bambini di oggi come nativi digitali, perché sanno usare smartphone o computers prima e a volte meglio di noi. È una definizione assolutamente infelice, perché dà l'impressione che i bambini controllino il mondo digitale. Non è così, certo sanno usare videogiochi o smartphone già a 3 anni: ma si chiedono cosa c'è dentro un computer? si rendono conto che un computer (o cellulare o telecomando o ...) è stato costruito e programmato da qualcuno? hanno mai provato a programmare un computer? O si limitano ad usare passivamente cose preparate da altri?*

*In questa tesi viene affrontato il tema dell'insegnamento dell'informatica in particolare della programmazione nella Scuola secondaria. In particolare si ritiene che, per un corretto approccio con il computer o altri strumenti tecnologici, sia importante che gli alunni comprendano che quest'ultimi sono soltanto delle macchine che eseguono i comandi che gli vengono impartiti. Ho potuto constatare che generalmente gli alunni pensano che il computer sia intelligente. Per spiegare la programmazione è necessario prima far comprendere che il computer esegue ordini che sono stati dati dai programmatori e non ha un'intenzionalità ed una intelligenza propria. L'insegnamento della programmazione può essere finalizzato ad aiutare gli alunni ad instaurare un corretto approccio con il computer o altri strumenti tecnologici, in maniera che interagiscano con gli strumenti tecnico informatici che li circondano non solo in maniera passiva, ovvero semplicemente usandoli, ma anche in maniera attiva, cioè programmandoli direttamente.*

*Cit.: Il computer non è una macchina intelligente che aiuta le persone stupide, anzi è una macchina stupida che funziona solo nelle mani delle persone intelligenti.*

**Umberto Eco**

*Programmare è una attività creativa! vuol dire analizzare e risolvere problemi. Per risolverli, occorre pensare, decidere, e valutare i nostri sforzi.*

*L'attività vuole essere presentata come un gioco, da fare singolarmente e collettivamente, allo scopo di produrre un risultato tangibile in tempi brevi e allo stesso tempo suscitare la curiosità degli alunni sui temi della programmazione, e più in generale, su cosa c'è dentro, in questo caso a livello software, nei vari strumenti tecnologici che lo circondano.*

*Per ottenere tutto ciò uso Scratch, un nuovo linguaggio di programmazione che rende semplice la creazione di storie interattive, di giochi e di animazioni nonché la condivisione, con altri, sul web delle proprie creazioni. Scratch nasce per aiutare i ragazzi nello sviluppo delle competenze d'apprendimento con il quale creano e condividono progetti e, contestualmente, imparano concetti matematici e computazionali fondamentali inoltre, li predispone a pensare più creativamente, a ragionare sistematicamente e a lavorare collaborando.*

*La tesi è sviluppata partendo dalla descrizione del contesto e della progettazione, per poi definire l'organizzazione dell'unità didattica, che comprende l'introduzione a Scratch la sua interfaccia, la schermata e i blocchi, per poi passare alla programmazione con alcuni esempi da fare insieme, e delle esercitazioni e/o verifiche da proporre durante e a fine percorso.*

## **Identificazione**

- Scuola: Istituto Tecnico Industriale
- Analisi del contesto: E' importante che il docente, la prima volta che entra in contatto con la classe, delinea il tipo di intervento da attuare, calibrandolo sulla base dei livelli cognitivi di riferimento, del contesto socio-culturale, delle aspirazioni e delle aspettative degli studenti, e dei diversi livelli di preparazione. Le lezioni saranno organizzate con riferimenti attuali e utilizzando il più possibile attività laboratoriali, in modo da attirare l'attenzione degli allievi.
- Destinatari: L'unità didattica è strutturata per ragazzi frequentanti la prima classe. Sicuramente sarà una classe numerosa intorno ai 30 allievi.
- Periodo: secondo quadrimestre

## **Prerequisiti**

- Conoscenza di base del computer e della sua architettura.
- Conoscenza del sistema operativo e della gestione dei file.
- Conoscenza degli algoritmi e rappresentazione tramite flow chart e pseudo codifica.
- Il concetto di variabile.
- Saper distinguere tra problema, algoritmo e programma.
- Il problem solving.

## **Contenuti**

- L'interfaccia di Scratch: Uso, componenti e funzionamento.

## **Obiettivi**

- Conoscere il concetto di linguaggio.
- Comprendere la differenza tra linguaggi naturali e formali.
- Conoscenza dell'interfaccia di Scratch.
- Autonomia nel realizzare programmi per risolvere problemi semplici tramite Scratch

## **Metodologia**

- *Lezione frontale e interattiva in laboratorio.*
- *Uso del PC e di strumenti multimediali.*
- *Attività individuale e di gruppo in laboratorio guidata dall'insegnante.*

## **Strumenti**

- *Laboratorio di informatica.*
- *Videoproiettore e/o LIM (se presente).*
- *Libro di testo e dispense preparate dal docente.*

## **Tempi**

- *Almeno 20 ore di lezione frontale ed interattiva in laboratorio.*

## **Verifica**

- *Saranno proposte almeno due verifiche in laboratorio, intermedie dopo lo svolgimento di alcuni argomenti e una verifica sommativa finale.*
- *Gli allievi dovranno realizzare dei semplici programmi e qualche test con domande a risposta multipla e a risposta aperta.*

## **Introduzione a Scratch**

*Scratch è sia un linguaggio di programmazione che una comunità online (<http://scratch.mit.edu>), dove i ragazzi possono programmare e condividere con altri provenienti da qualunque parte del mondo i loro oggetti multimediali interattivi come ad esempio storie, giochi e animazioni. Su internet sono disponibili risorse di ogni tipo su Scratch, e il sito ufficiale prevede anche una sezione per educatori e una per genitori.*

*Quando i ragazzi creano con Scratch imparano allo stesso tempo a pensare in modo creativo, a lavorare in collaborazione e a ragionare in modo sistematico.*

*Anche se è stato specificamente progettato per ragazzi dagli 8 ai 16 anni, Scratch è usato in realtà da persone di ogni età, inclusi bambini più piccoli supportati dai loro genitori. Start from scratch vuol dire partire da zero: è una delle idee di base di Scratch, non serve saper programmare ma si impara a programmare mentre si creano storie interattive, giochi, progetti multimediali.*

*La programmazione è completamente grafica: non occorre conoscere un linguaggio di programmazione, ma mettendo insieme mattoncini colorati, si arriverà alla realizzazione di programmi anche piuttosto complessi.*

*E' pensata come un gioco, ma richiede che l'utente pensi a cosa vuole fare e sia in grado di dividere azioni complesse in pezzi elementari. E' fortemente multimediale.*

*Scratch si presenta come un ambiente di apprendimento grafico. Racchiude lo spirito costruzionista di Logo ed Etoys e il suo carattere di tipo visuale non ostacola l'utente con sintassi rigide, diminuendo così la possibilità di errori e dando una sensazione di immediatezza alla stesura del codice. La sua prima versione fu sviluppata nel 2006 dal gruppo Lifelong Kindergarten capeggiato da Mitchel Resnik presso il Mit Media Lab a Boston.*

*Venne creato per un segmento di pubblico giovane con lo scopo di favorire l'apprendimento grazie all'utilizzo di soluzioni intuitive e di un'interfaccia decisamente user friendly. Scratch consente infatti la costruzione di programmi e una loro successiva ed efficace verifica rendendo la programmazione facilmente assimilabile. Il tutto è reso possibile grazie alla costruzione di codice tramite blocchi prefabbricati, abbinati ad un intelligente approccio del tipo drag and drop (clicca, trascina e rilascia). Ogni blocco agisce direttamente o indirettamente su attori detti Sprite, rappresentati da piccoli personaggi la cui funzione è quella di eseguire le azioni previste dal codice all'interno del progetto. L'introduzione di questi personaggi interattivi, gli Sprite, può essere interpretato come il tentativo di un avvicinamento da parte di Scratch al paradigma di programmazione orientato agli oggetti (object oriented), tuttavia non esistono i concetti di classe o ereditarietà, prerogative essenziali per essere considerato tale. Ogni comando opera solamente sullo Sprite nel quale appare; in altri termini, il ricettore implicito di ogni comando è sempre lo Sprite che lo invoca. Questo fa sì che ogni Sprite abbia un insieme di script personali, che ne facilità la comprensione, infatti gli script presenti all'interno di uno Sprite descrivono completamente il suo comportamento, inoltre i cambiamenti sul codice sono localizzati così da non avere ripercussioni su altri Sprite.*

*Inoltre Scratch supporta un meccanismo, che è in grado di trasmettere in broadcast un messaggio; ciò significa che tale messaggio verrà ricevuto da ogni altro Sprite, il quale gestirà l'evento a seconda di come è stato programmato, il messaggio lanciato da uno Sprite può essere ricevuto da più Sprite ed attivare quindi più script (uno-a-molti), ed è infine asincrono in quanto il comando di broadcast non attende che gli script che lo hanno ricevuto e gestito vengano terminati.*

## L'interfaccia grafica

L'interfaccia utente (fig 1) vede lo schermo diviso in diversi riquadri: la tavolozza dei blocchi, le informazioni relative allo Sprite corrente e la zona script, lo stage (ovvero il background) e la lista degli Sprite.

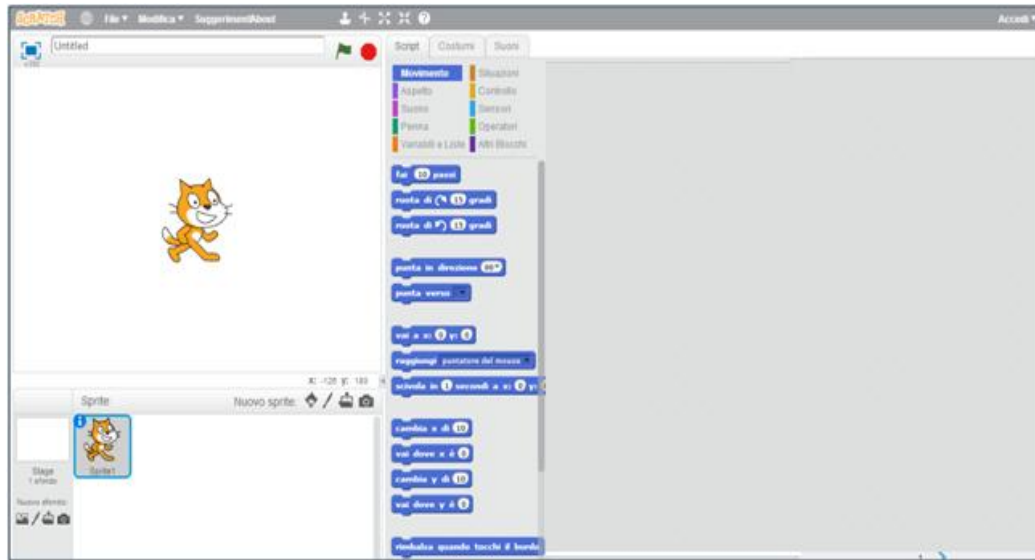


fig. 1

L'obiettivo è quello di lasciare a disposizione un numero minimo di blocchi non compromettendo la possibilità di trattare una vasta gamma di progetti. Una strategia applicata è stata quella di raggruppare operazioni simili in singoli blocchi dotati di menù a tendina rendendo visibili le singole operazioni solo su richiesta. Con queste premesse, la prospettiva di un ambiente dotato di un'interfaccia a finestra singola, un set di comandi contenuto e una programmazione grafica a blocchi rendono questo linguaggio una scelta più appetibile ad un segmento giovane di utenti. Sempre più spesso infatti, Scratch viene usato dagli studenti dei college per corsi di introduzione all'informatica come, ad esempio, succede tutt'oggi ad Harvard.

Il motto della comunità online di Scratch recita: *Imagine, Program, Share*. Il suo significato evidenzia la condivisione e la creatività come parti importanti della filosofia che muove gli utenti di Scratch. Per sollecitare il suo utilizzo e la condivisione dei progetti è stata creata una comunità online di Scratch, distribuita in molti paesi del mondo che contava, al dicembre 2011, più di 950000 utenti registrati.

La palette dei blocchi, o tavolozza dei blocchi, contiene pittogrammi, detti appunto blocchi. La copia di un blocco può essere trascinata sull'area script, giustapponendola ad altri per creare programmi. Per far sì che la tavolozza sia di dimensioni contenute, essa viene organizzata in dieci insiemi, che raggruppano i blocchi a seconda delle funzioni che eseguono: motion (movimento), looks (sguardo), sound (suono), pen (penna), data (variabili e liste), events (situazioni), control (controllo), sensing (sensori), operators (operatori) e more blocks (altri blocchi). Colori e forme diversi contraddistinguono differenti tipi di blocchi. Nelle versioni 1.3.1 e inferiori, gli operatori si chiamavano numbers. Una caratteristica fondamentale di Scratch è il codice multi-thread con scambio di messaggi, ma la versione corrente non tratta le procedure come strutture di prima classe e non ha alcuna opzione di I/O per i file e supporta solo gli array monodimensionali a lunghezza arbitraria, noti come liste. Scalari in virgola mobile e stringhe sono supportati a partire dalla versione 1.4, ma con limitata capacità di manipolazione delle stringhe. Vi è un forte contrasto

*tra le potenti funzioni multimediali abbinato allo stile di programmazione multi-thread e la visibilità (scope) piuttosto limitata del linguaggio di programmazione Scratch.*

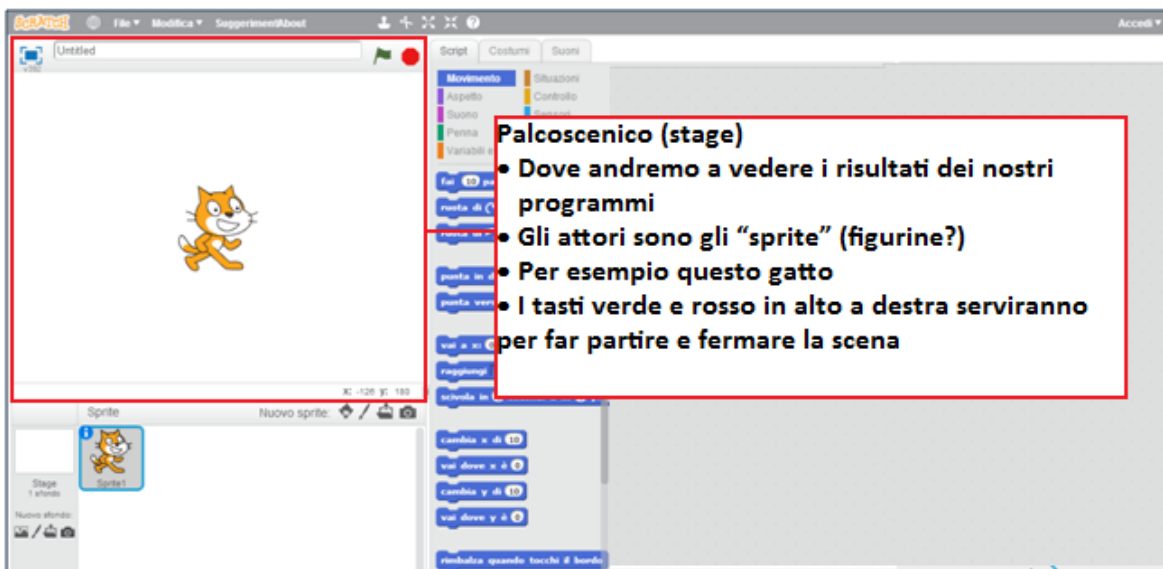
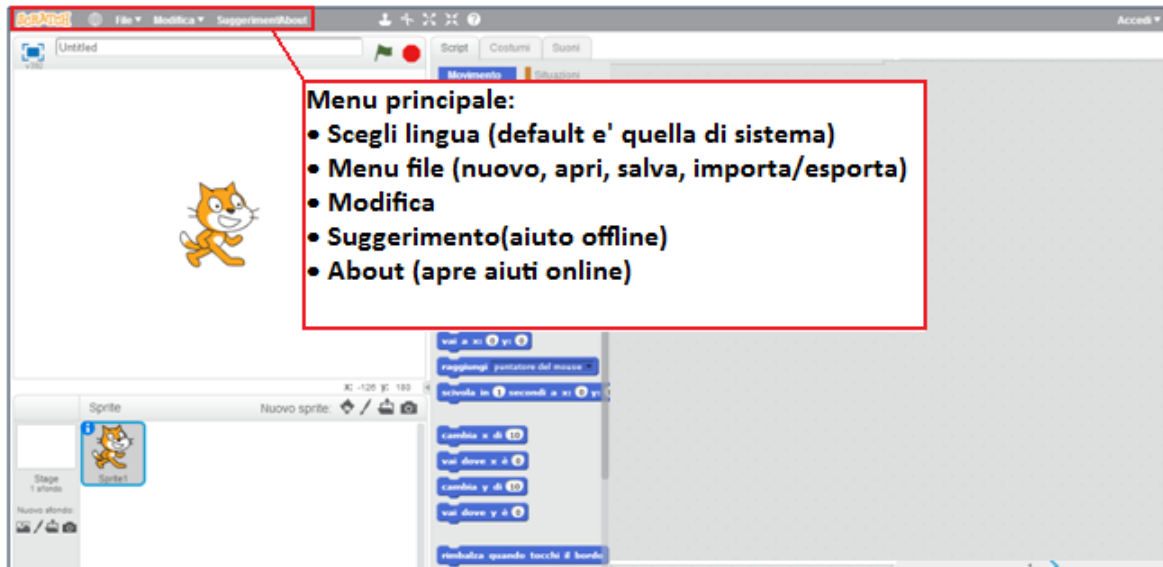
*Altra cosa importante è che in Scratch non c'è distinzione tra compilazione, prova o stesura di un programma; esiste un unico stato all'interno del quale l'utente può scrivere codice trascinando e componendo blocchi che possono poi essere testati, cliccandovi sopra e ricevendo un riscontro grafico immediato dell'azione eseguita. Procedendo in questo modo vengono eliminate tutte le transizioni, forzando chi crea un programma ad essere in ogni momento attento e impegnato a scrivere codice e a testarlo per poi eventualmente modificarlo e correggerlo. Così facendo si stimola un approccio alla sviluppo di tipo bottom-up, organizzato in passaggi successivi, per raggiungere l'obiettivo finale passando per traguardi intermedi. A livello pratico il tutto si traduce nella composizione di frammenti di script testati singolarmente, successivamente combinati e assemblati tra loro in unità più grandi fino a raggiungere la soluzione del problema. Questo metodo nasconde risvolti positivi come il fatto che l'utente scopre con facilità le funzioni di nuovi blocchi.*

*Altra forza di Scratch è che non ha messaggi di errore, gli errori di sintassi sono stati eliminati in quanto i blocchi sono componibili tra loro solo in modi che possano avere senso. La loro forma infatti, suggerisce il corretto aggancio con altri blocchi non lasciando spazio a quelli che in un linguaggio di programmazione testuale sarebbero errori di sintassi. I creatori di Scratch si sono ingegnati anche per eliminare gli errori in fase di esecuzione (run time) rendendo tutti i blocchi sensibili ai valori passati in ingresso ma, nonostante lo sforzo, non è stato possibile eliminare del tutto ogni errore e deve essere il programmatore a premurarsi di scrivere script che eseguano ciò che lui vuole. L'aggiunta di piccoli ma non indifferenti dettagli nel design (flashing dei blocchi quando testati, degli oggetti quando acceduti, ...) fanno una grossa differenza per la comprensione del codice scritto, riducendo al minimo al possibilità di errori.*

*Gli script in Scratch vengono costruiti componendo più blocchi. La forma di quest'ultimi suggerisce come debbano essere agganciati e il sistema drag and drop rifiuta automaticamente connessioni prive di significato. In Scratch la grammatica visiva dei blocchi gioca il ruolo delle regole di sintassi, più comunemente usate negli altri linguaggi di programmazione.*



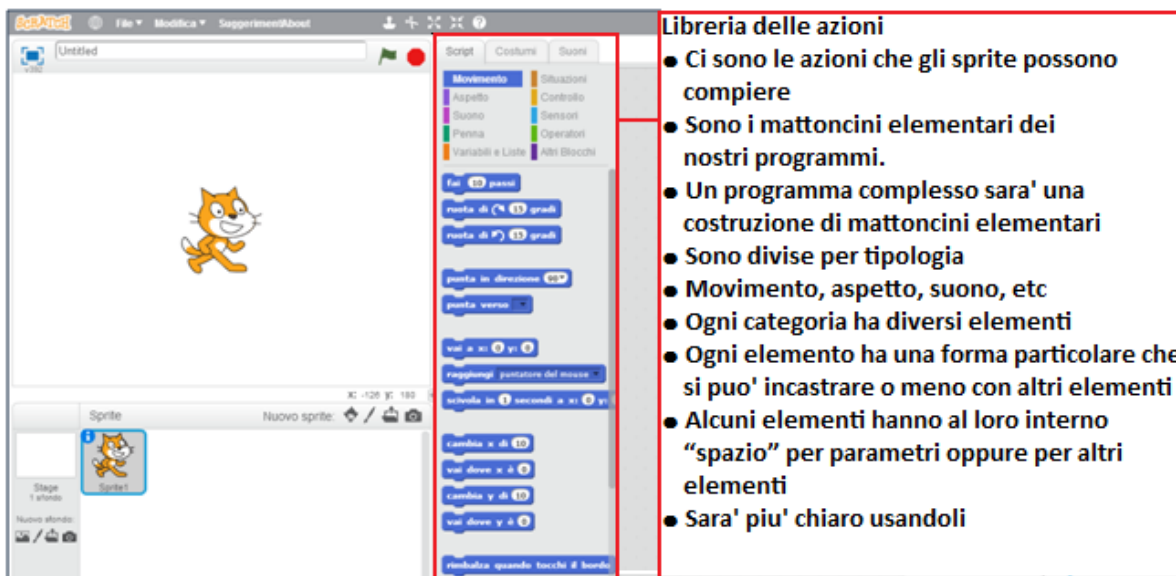
## Conosciamo la schermata di Scratch





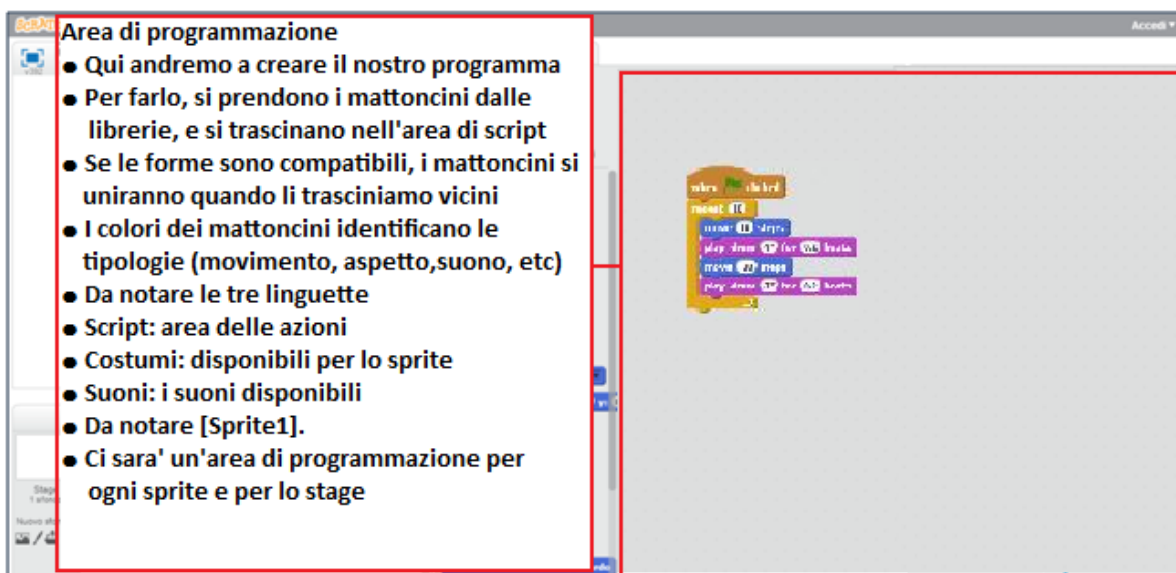
**Gli sprite**

- 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



**Libreria delle azioni**

- Ci sono le azioni che gli sprite possono compiere
- Sono i mattoncini elementari dei nostri programmi.
- Un programma complesso sara' 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 puo' incastrare o meno con altri elementi
- Alcuni elementi hanno al loro interno "spazio" per parametri oppure per altri elementi
- Sara' piu' chiaro usandoli



**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 trasciniamo 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 sara' un'area di programmazione per ogni sprite e per lo stage

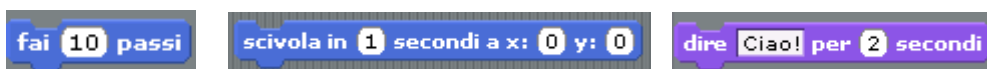


## Conosciamo i blocchi per programmare

Per programmare dobbiamo prendere i blocchi/mattoncini e trascinarli nell'area di script. Esistono 4 tipi di blocchi in Scratch:

1. blocchi comando (command blocks): presentano un incavo nella parte superiore e una protuberanza simmetrica nella parte inferiore. Rappresentano l'equivalente dell'espressioni, condizionali e non, dei linguaggi testuali classici. Composti tra loro, utilizzando la forma ad incastro tramite il drag and drop, formano sequenze di comandi detti stack (pile).

Esempio di blocco comando



2. blocchi funzione (function blocks): la loro forma è tondeggiate ai lati e non presentano incastrici a differenza degli altri blocchi. Sono da considerare come degli operatori. Questo tipo di blocchi non sono adoperati in sequenze lineari di codice, in cui blocchi successivi vengono eseguiti sequenzialmente, come i blocchi comando, bensì vengono usati come argomenti da passare ai blocchi comando oppure vengono annidati tra loro per costruire espressioni. I blocchi funzione ritornano un valore.

Esempio di blocco funzione



3. blocchi per la gestione di eventi (trigger blocks): presentano una caratteristica forma arrotondata nella parte superiore. Collegano gli eventi, quali la pressione di un tasto o il clic del mouse, con le operazioni che li gestiscono. Alcuni di questi blocchi presentano uno spazio per inserire parametri incorporato che può accogliere numeri, stringhe, valori booleani e in alcuni frangenti anche blocchi funzione. L'esecuzione dei blocchi ad essi sottostanti comincia nel caso l'evento descritto abbia luogo. Un caso particolare è quello in cui uno stack di codice inizia con un blocco di questo tipo avente come argomento una bandiera verde, esso verrà eseguito quando l'utente cliccherà il pulsante start.

Esempio di blocco per la gestione di eventi



4. blocchi per le strutture di controllo (control structure blocks): presentano delle aperture atte ad accogliere sequenze nidificate di comandi. Sono una specie di blocchi comando con la differenza che possono contenere una o più sequenze di comandi nidificate. La forma e l'uso intuitivi di questi blocchi li rende l'ideale per l'introduzione dell'informatica e dei suoi concetti base. Rappresentano molti dei costrutti condizionali tipici dei linguaggi di programmazione testuali ( if, while, for, ...).  
Esempio di blocco per le strutture di controllo

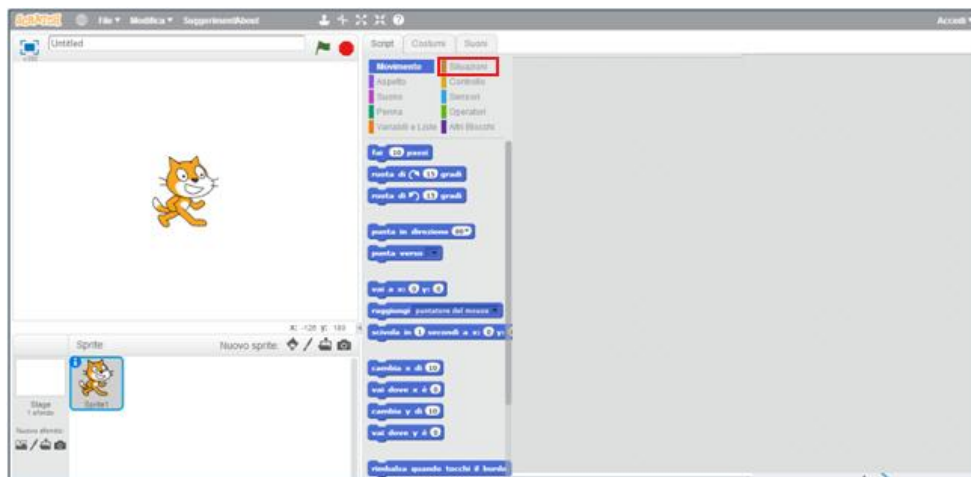


Scratch ha 3 tipi di dato di prima classe: booleani, numerici e stringhe. Sono quindi gli unici 3 tipi di dato che possono essere usati in un'espressione, salvati in una variabile o restituiti da un blocco reporter di base. In quanto linguaggio visuale, Scratch suggerisce il tipo di parametro che si aspetta di ricevere con la forma dello slot dove andrà a incastrarsi l'eventuale blocco parametro e il tipo di dato che restituirà un blocco reporter con la forma del blocco reporter stesso. Nel primo caso Scratch mette a disposizione tre forme differenti per i tre tipi di dato differenti ovvero booleani, numerici e stringhe. Nel secondo caso invece, un blocco funzione può essere di sole due forme. Questa è la conseguenza della non tipizzazione di Scratch, che raggruppa assieme stringhe e valori numerici accettandoli indistintamente come se fossero di un unico tipo.

## Iniziamo a programmare

Avviamo Scratch e iniziamo .....

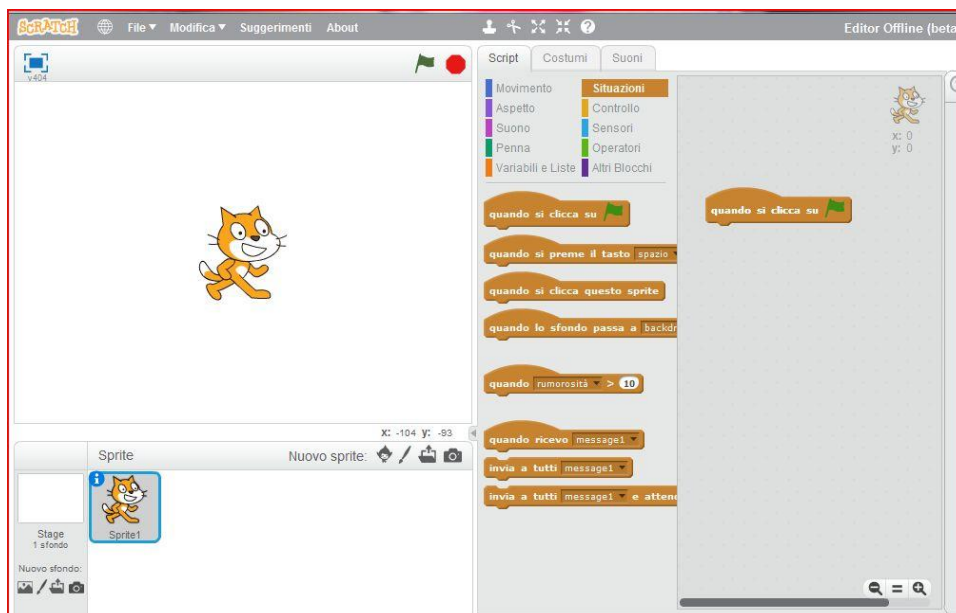
Facciamo dire ciao al nostro gatto .....



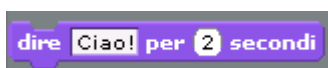
Clicchiamo sullo script **Situazioni**, prendiamo il primo mattoncino



e trasciniamolo nell'area di programmazione

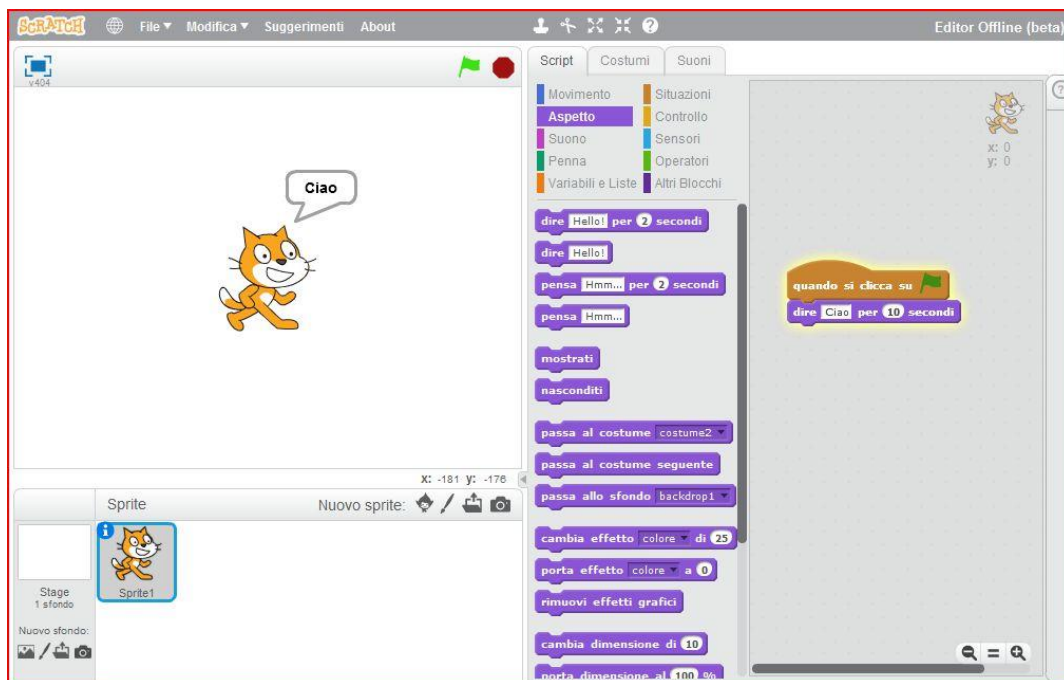


Adesso clicchiamo sullo Script **Aspetto**, e prendiamo il mattoncino



trasciniamolo e uniamolo al precedente.

Ecco fatto, clicchiamo sui blocchi o sulla bandierina verde ed il nostro gattino ci saluterà.

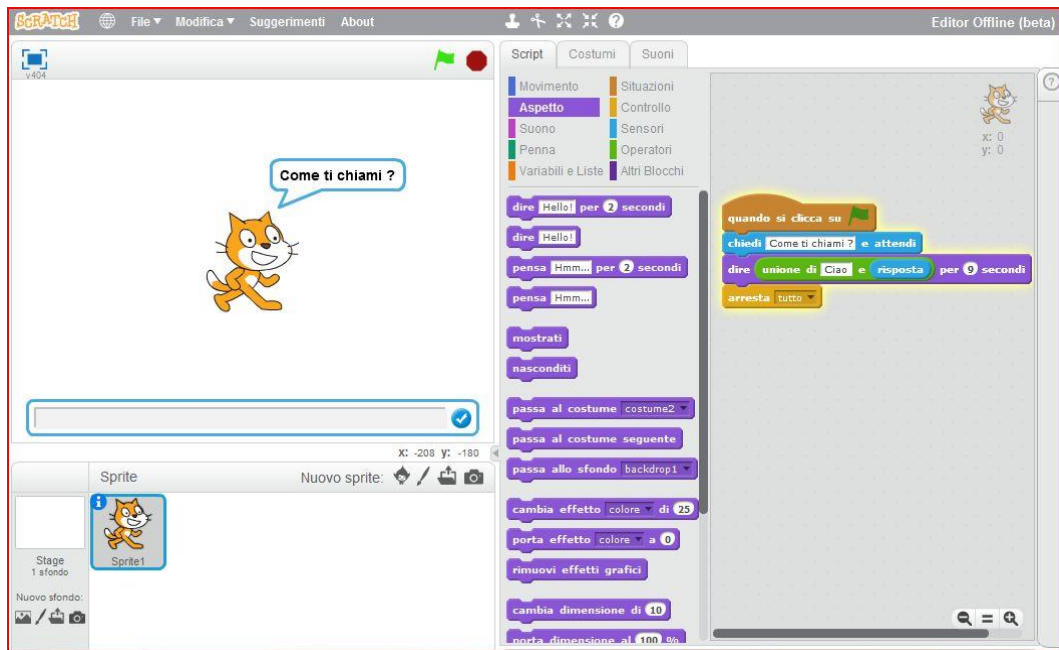


Abbiamo realizzato il nostro primo programma con Scratch.

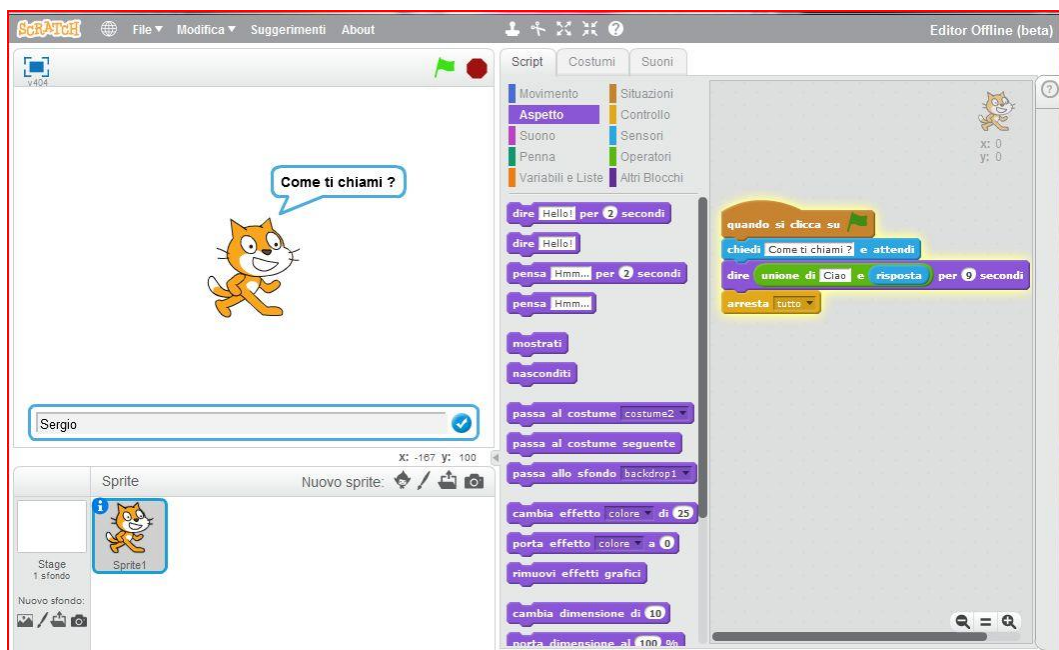
Adesso realizziamo un programma, che interagisce con l'utente, chiedendogli il nome in input e personalizzando così il saluto (output), per fare questo utilizzeremo 6 semplici blocchi.

|   |  |
|---|--|
| A close-up view of the Scratch script editor showing four nested blocks. The first is an orange "quando si clicca su" (when green flag clicked) block. The second is a blue "chiedi Come ti chiami? e attendi" (ask "Come ti chiami?" and wait) block. The third is a purple "dire unione di Ciao e risposta per 2 secondi" (say "unione di Ciao e risposta" for 2 seconds) block. The fourth is a yellow "arresta tutto" (stop all) block. | <p>In questo programma utilizziamo degli Script annidati, ovvero lo Script <b>Sensori</b> (dire...) con lo Script <b>Operatori</b> (unione ...) e lo Script <b>Sensori</b> (risposta), così annidati, il nostro gattino dopo averci chiesto di digitare il nome, ci saluterà unendo a ciao il nome inserito da tastiera.</p> |
|---|--|

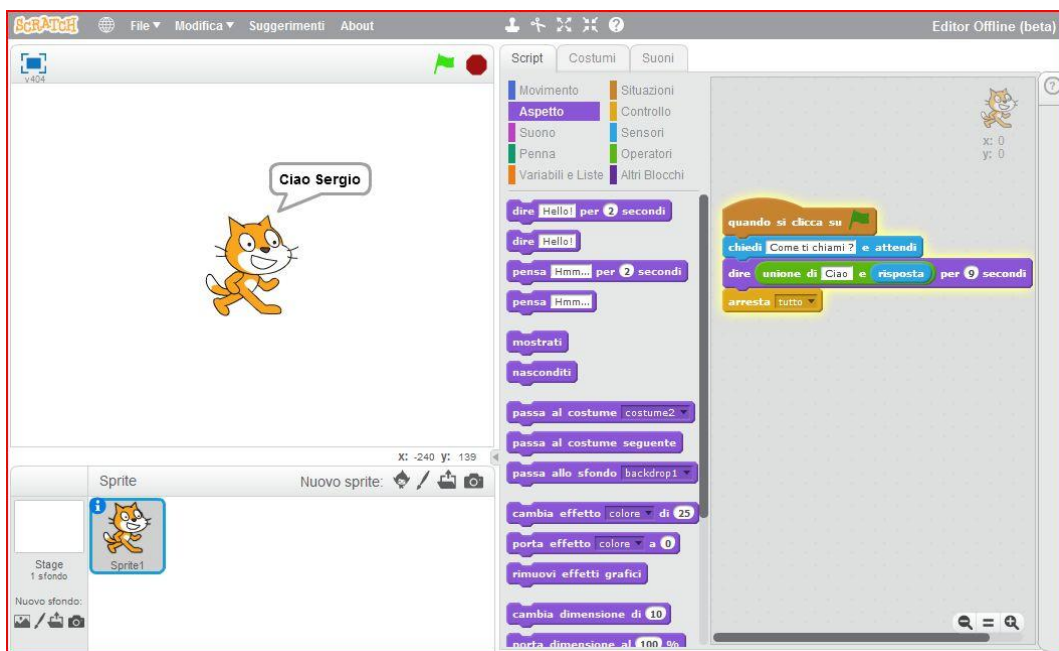
Eseguiamo



Dopo il messaggio lo Sprite, attende la risposta, si apre una finestra nella parte bassa dello stage



*E dopo aver inserito il nome .....*



*Voilà abbiamo realizzato il nostro secondo programma con Scratch.*

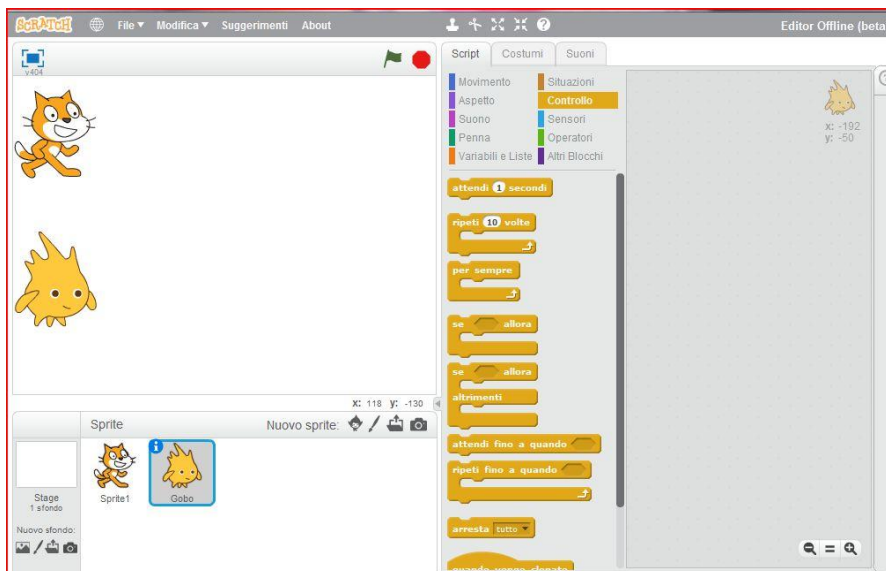


Ora realizziamo un programma dove impareremo ad :

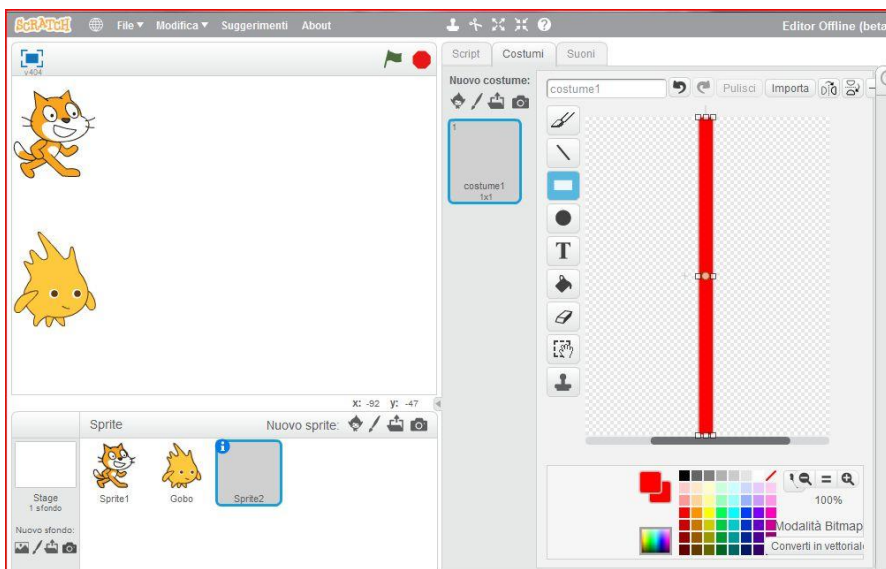
- Ad aggiungere Sprite.
- A modificare e creare i costumi.
- Ad utilizzare Script di **Controllo**, il blocco "Ripeti fino a quando" ovvero un CICLO.

Infatti realizzeremo una gara tra due Sprite. Chi vince deve dire "HO VINTOOOOOOOOO".

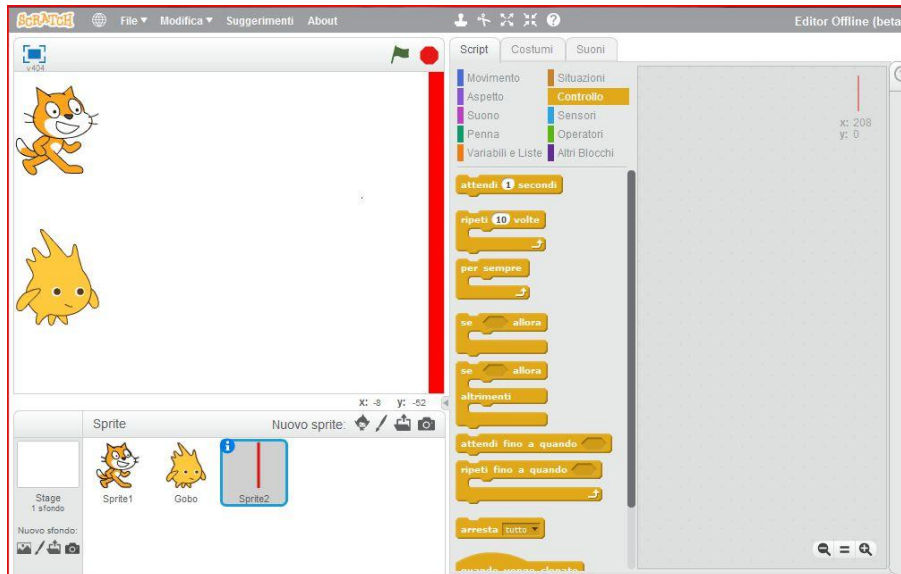
1. Scegliamo un nuovo Sprite che gareggerà con il nostro gattino di Scratch cliccando sul bottone "Scegli nuovo sprite dalla libreria" sulla destra del testo "Nuovo Sprite". Sceglilo a piacimento.
2. Posizionare con il mouse i due Sprite sulla parte sinistra dello schermo. Dovreste trovarvi in una situazione simile a questa:



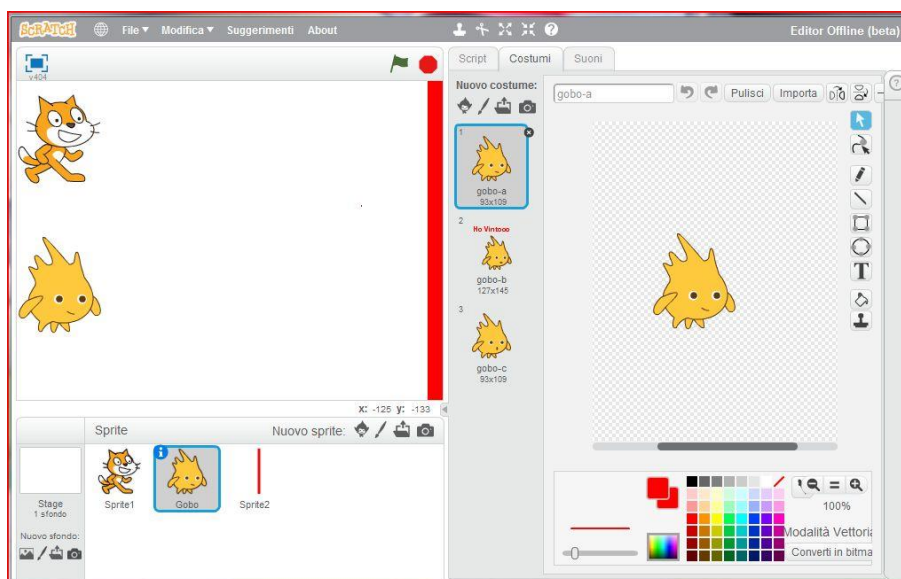
3. Ora è necessario disegnare il traguardo. Sarà anch'esso uno Sprite che questa volta dovremo disegnare da soli cliccando su "Disegna un nuovo Sprite" posizionato sulla destra del bottone precedente. Disegnare un rettangolo rosso come visualizzato in figura facendo attenzione che il livello di zoom in basso a destra sia impostato su 100%.



4. Una volta terminato cliccare su Script (barra in alto) e posizionare il traguardo sulla destra dell'area (Stage) con i protagonisti del nostro gioco. Dovreste trovarvi in questa situazione:



5. Ora dobbiamo creare i costumi. I costumi sono delle versioni grafiche differenti degli Sprite. Selezioniamo entrambi gli Sprite e aggiungiamo la scritta "HO VINTOOOO" sul secondo costume. Se lo Sprite che avete scelto non ha un secondo costume dovreste cliccare con il tasto destro del mouse e cliccare su "DUPLICA". Dovreste trovarvi nella situazione seguente:



6. A questo punto scriviamo il programma. **IMPORTANTE:** ad ogni Sprite deve essere assegnato il proprio codice affinché riesca a compiere le operazioni che vogliamo. Cliccare su entrambi gli Sprite che gareggeranno (non il traguardo!) e scrivere, trascinando i blocchi, il seguente codice di programmazione:



Per realizzare il codice corretto scegliete i blocchi a seconda del colore che vedete in figura. Per selezionare correttamente il colore nel blocco azzurro cliccare sul quadratino e poi sul colore rosso del traguardo.

7. A questo punto è possibile cliccare sulla bandierina verde sopra lo stage e vedere cosa accade. Se tutto ha funzionato correttamente dovrete trovarvi in questa situazione:



**IL GIOCO E' TERMINATO.**

*Primo esercizio*

Provate a modificare/migliorare questo esercizio, ecco alcune idee:




- Mettendo uno sfondo. Cliccare su "Scegli uno sfondo dalla libreria" dal menu in basso a sinistra e scegliere uno sfondo a piacere.
- Sostituendo delle macchine, dei treni o dei camion agli sprite attuali. Puoi anche inserire tue foto o immagini prese dal web.
- Aggiungendo suoni mentre la gara è in corso o quando uno dei due Sprite vince la gara.
- A modificare i costumi degli sprite mentre la gara è in corso.
- ecc. ecc.







Provateci da soli! Senza timore di sbagliare!

Adesso progettiamo e realizziamo un programma in cui sullo sfondo scarichiamo una immagine da internet di una porta di campo da calcio e posizioneremo quattro sprite: un arbitro, il gatto di Scratch che rappresenta il calciatore rigorista, una scimmietta che rappresenta il portiere ed un pallone. La catena di azioni da generare è quella di far fischiare l'arbitro, (anche il suono del fischietto va scaricato da internet) in seguito al fischio il calciatore calcerà il rigore quindi il pallone terminerà la sua corsa in porta mentre il portiere si muoverà per cercare di parare, se il pallone entra in porta l'arbitro convalida il goal altrimenti dirà parato.

Con questo programma impareremo a:

- passare da un costume ad un altro di uno Sprite, infatti il gatto e la scimmia avranno due costumi che cambiano durante l'esecuzione del programma.
- far interagire gli Sprite, inviando messaggi.
- attivare uno Sprite al ricevimento di un messaggio.
- utilizzare Script di **Controllo**, il blocco "Se - Allora - Altrimenti" ovvero una ISTRUZIONE CONDIZIONALE (istruzione fondamentale in informatica).
- utilizzare la funzione Random che genera numeri casuali in intervallo di numeri dati.

|   |  |
|---|--|
|   | <p>Per prima cosa scarichiamo da internet una immagine simile questa, o in alternativa la si può disegnare, la carichiamo come sfondo e cancelliamo quello bianco di default.</p>  |
|  | <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>quando si clicca su </p> <p>produci suono POLWHST2 e attendi la fine</p> <p>invia a tutti fischiato</p> </div> <div style="border: 1px solid red; padding: 5px; margin-bottom: 10px;"> <p>quando ricevo parato</p> <p>produci suono POLWHST2</p> <p>dire PARATA per 2 secondi</p> <p>invia a tutti ok</p> </div> <div style="border: 1px solid red; padding: 5px;"> <p>quando ricevo goal</p> <p>produci suono POLWHST2</p> <p>dire GOL per 2 secondi</p> <p>invia a tutti ok</p> </div> <ol style="list-style-type: none"> <li>1. Scegliamo tra gli Sprite presenti in libreria l'arbitro.</li> <li>2. Costruiamo i codici di programma che lo riguardano:             <ol style="list-style-type: none"> <li>a) il 1° codice in cui è presente la bandiera verde avvia il programma, produce il suono (fischio) ed invia un messaggio (fischiato) a tutti i Sprite.</li> <li>b) <b>Il 2° o il 3°</b> iniziano con il blocco <b>quando ricevo</b> con relativo nome del messaggio (<b>parato o goal</b>) questo fa sì che lo Sprite esegua uno dei due blocchi (parato o goal), i codici sono simili l'unica differenza è nel messaggio del blocco dire, ovvero dirà PARATA o GOL.</li> </ol> </li> </ol> |

|   |   |  |
|---|---|--|
|    |    | <p>Questo Sprite a due codici di programma :</p> <ol style="list-style-type: none"> <li>1. Il 1° codice si attiva quando riceve il messaggio <b>fischiato</b>, qui è presente il blocco di <b>passa al costume</b>, per cambiare il costume dello Sprite, ed invia un messaggio.</li> <li>2. Il 2° codice si attiva quando riceve il messaggio <b>ok</b></li> </ol>  |
|    |   | <p>Anche questo Sprite ha due codici di programma</p> <ol style="list-style-type: none"> <li>1. Il 1° più complesso, infatti sono presenti sia le funzioni che generano in numero casuale, ma soprattutto è presente la <b>struttura di controllo</b> (se-allora-altrimenti) che in funzione della condizione, invia o uno o l'altro messaggio.</li> <li>2. Il 2° codice serve solo per riposizionare lo Sprite nella posizione iniziale.</li> </ol> |
|  |  | <p>Anche questo Sprite ha due codici di programma</p> <ol style="list-style-type: none"> <li>1. Il 1° in cui sono presenti i blocchi, <b>passa al costume</b>, <b>vai a</b>, e <b>scivola</b> che incorpora la funzione che genera numeri casuali</li> <li>2. Il 2° codice serve solo per passare all'altro costume e riposizionare lo Sprite nella posizione iniziale.</li> </ol>   |

Ecco fatto, anche questo gioco è finito il risultato finale è un giochino semplice e divertente l'invito è sempre quello di provare a modificarlo e migliorarlo, in modo da prendere sempre più confidenza con Scratch, usando la vostra creatività, immaginazione e la fantasia.

## **Esercitazioni e verifiche**

*Così come le lezioni sono state preparate e sviluppate seguendo la linea del gioco, realizzando programmi partendo da quello più semplice a quello più complesso, anche le esercitazioni e le verifiche, seguiranno la stessa filosofia.*

*Credo che così facendo, sarà più facile che un ragazzo si impegni e si interessi alla realizzazione di un programma che simuli un'animazione o un videogames, invece di uno che calcoli l'area di un triangolo. Sicuramente così si riesce a stimolare e incuriosire gli alunni, creando inconsciamente forse, anche una sorta di competizione tra loro a chi realizza l'animazione o il videogames migliore, avvicinandoli in modo semplice, divertente e giocoso ai temi fondamentali dell'informatica e della programmazione.*

*Questo è molto utile anche per ovviare ad un problema che ogni insegnante vive con gli alunni: ogni volta che si è in laboratorio e quindi si dà loro la possibilità di avere un computer, la prima cosa che fanno è cercare qualche videogames per passarci le ore.*

*Scratch costituisce la possibilità di sostituire il gioco creato da altri, stimolando l'alunno a realizzare in autonomia il proprio gioco. L'apprendimento avviene pertanto realizzando videogames e animazioni.*

*Questi sono alcuni esercizi da sottoporre ai ragazzi durante il corso come esercitazione o verifiche, da realizzare singolarmente o in gruppi.*

### **1° Esercizio**

*Progettare un acquario con alcuni pesci che vanno avanti ed indietro.*

### **2° Esercizio**

*Creare uno Sprite con almeno tre costumi, controllare l'orientamento dello sprite con i tasti freccia destra e sinistra, controllare il costume dello sprite con i tasti + e - .*

### **3° Esercizio**

*Progettare il gioco del pong.*

### **4° Esercizio**

*Progettare un gioco in cui un gatto ed un drago si muovono sullo stage. Il gatto si muove attraverso la pressione dei tasti freccia presenti sulla tastiera, il drago lo insegue automaticamente e cerca di bruciare il gatto con la sua fiamma.*

### **5° Esercizio**

*Realizzazione di un gioco costituito da uno squalo che mangia il pesciolino che, in maniera randomica, nuota all'interno del gioco. Lo squalo deve essere mosso dal giocatore attraverso le frecce della tastiera. Quando il pesciolino viene mangiato deve emettere un suono e visualizzare un testo sullo schermo.*

### **6° Esercizio**

*Realizzazione di un gioco costituito da uno Sprite navicella controllato nei movimenti sinistra-destra dalla tastiera e posizionata in basso nello schermo; premendo la barra spaziatrice la navicella emette un missile che procede verticalmente fino al fondo dello schermo, se il missile "tocca" lo sprite B, che si muove nello schermo di moto uniforme da destra a sinistra e viceversa, si ha una esplosione.*

## **Sitografia**

Scratch on-line: <http://scratch.mit.edu/>

Download Scratch 2.0 offline editor: <http://scratch.mit.edu/scratch2download/>

<http://learnscratch.org/>

<http://www.maffucci.it/2014/03/29/micro-corso-di-scratch-per-incominciare-ad-imparare-a-programmare/>

[http://www.cheforte.it/index.php?option=com\\_content&view=article&id=1701&catid=1031&Itemid=840](http://www.cheforte.it/index.php?option=com_content&view=article&id=1701&catid=1031&Itemid=840)

[http://it.wikipedia.org/wiki/Scratch\\_\(informatica\)](http://it.wikipedia.org/wiki/Scratch_(informatica))

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

<http://www.youtube.com/watch?v=pkhjX792yVI>

## **Conclusioni**

*Scratch risulta essere un linguaggio il cui utilizzo è indirizzato a quella fascia di utenti che cercano un approccio morbido all'informatica e in particolare alla programmazione. Questo genere di persone necessita infatti di un linguaggio modulare, intuitivo e più di ogni altra cosa sempre attivo, che restituisca dei feedback immediati ad ogni comando, così da iniziare a prendere confidenza con i costrutti, le strutture e il metodo di ragionamento. Tutte caratteristiche, queste, che descrivono integralmente Scratch.*

*Considerando quindi le sue potenzialità e gli ambiti in cui trova applicazioni concrete, viene rinnovata l'impressione che Scratch, probabilmente per scelta dei suoi progettisti, sia rivolto ad utenti alle prime armi, comunque aiutati nell'apprendimento anche dalle numerose risorse disponibili.*

*Concludo con un pensiero che riguarda tutto il percorso fatto dal mese di febbraio ad oggi, ritenendo che il condividere e il confrontarsi con i colleghi e i docenti, abbia sicuramente arricchito il mio bagaglio culturale e professionale.*