

preparare gli innovatori di domani



CREATIVE COMPUTING

Creative Computing significa imparare ad apprendere come:

pensare in modo creativo

ragionare in maniera sistemica (computational thinking)

lavorare in collaborazione.

Queste sono le **competenze essenziali** per vivere nella nostra era.

La scuola può e dev'essere il luogo dove si **preparano gli innovatori di domani**; e può farlo inserendo il **Creative Computing** nelle attività didattiche, facendo così il miglior uso possibile delle aule informatiche. I nostri giovani non hanno bisogno di imparare ad usare un pacchetto di office automation, tipo Word od Excel, ma hanno bisogno di sviluppare, comprendere ed applicare i principi della programmazione informatica e della *computer science*.

Anzi, per dirla ancor meglio: ***non imparare a programmare ma programmare per imparare***

Indice

IL CREATIVE COMPUTING.....	2
IL LINGUAGGIO SCRATCH.....	4
PERCHÈ INSEGNARE IL CREATIVE COMPUTING.....	6
IL PIANO FORMATIVO.....	7
IL PROGRAMMA.....	9
COLLABORAZIONI.....	14

IL CREATIVE COMPUTING

creative - per lungo tempo la *computer science* ossia l'informatica e la programmazione dei computer è stata ritenuta un'attività non creativa, non artistica, percepita come “*roba da nerd*” e distante dagli interessi dei più giovani. Niente di più sbagliato. Il *Creative Computing* intende proprio abbattere questo stereotipo ma soprattutto agevolare lo sviluppo della **creatività** proprio attraverso la programmazione dei computer. *Perché?*

La conoscenza delle cose e l'intelligenza nel saperle leggere non bastano; serve l'enzima della Creatività per risolvere problemi, per inventare soluzioni innovative a problemi inaspettati. Ne sono convinti ormai tutti, compresi migliaia tra i principali manager di grandi aziende mondiali che ritengono la creatività competenza fondamentale.

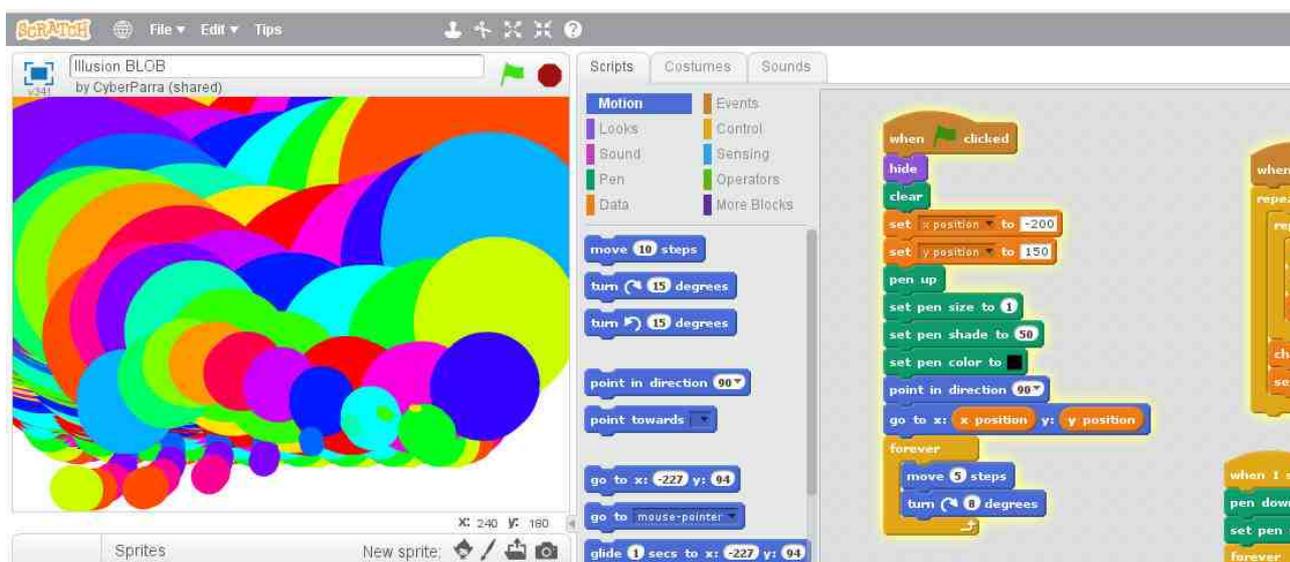
computing - oggi la maggior parte dei giovani sono dei *consumatori* digitali ma questo non ne fa degli esperti; tutt'altro. Obiettivo del Creative Computing è farli diventare **creatori**



digitali fornendogli la possibilità di sviluppare *concetti, pratiche e prospettive* per diventare consapevoli delle tecnologie e capaci di utilizzarle per creare.

Questo si ottiene non imparando ad utilizzare un pacchetto di office automation od un qualsiasi altro programma ma sviluppando i **concetti** alla base della programmazione e della computer science: la *sequenza*, le *iterazioni*, il *parallelismo*, le *condizioni*, gli *eventi*, e poi l'uso dei *dati* e degli *operatori*

Creare giochi, animazioni, prodotti interattivi applicando questi concetti non solo sviluppa le competenze digitali ma anche il pensiero che quegli stessi concetti sono applicabili nei vari contesti della nostra vita (*computational thinking*)



We SCRATCH

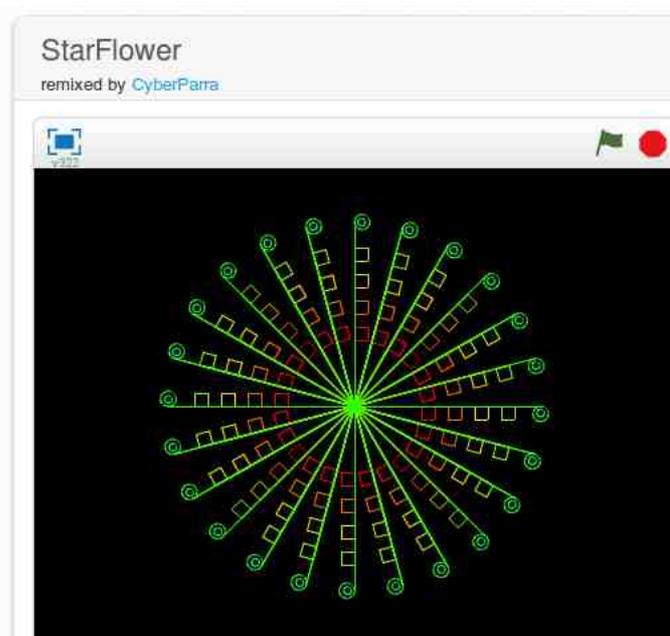
creare - programmare - imparare

IL LINGUAGGIO SCRATCH



Scratch è un linguaggio di programmazione visuale semplice ed intuitivo, basato su elementi grafici significativi (assimilabili ai mattoncini Lego) con i quali creare: **storie animate, videogiochi, questionari interattivi, arte multimediale, simulazioni scientifiche, musica e fumetti.**

Chiunque può creare i propri prodotti multimediali interattivi semplicemente impilando i **mattoncini** Scratch così come si fa con i Lego. Le proprie creazioni potranno essere poi condivise con la grande **comunità online** di Scratch che offre una piattaforma **sociale** dove condividere i propri progetti, ricevere e dare commenti, creare delle gallerie, scaricare altri progetti per modificarli (**remix**), creandone così dei nuovi prodotti partendo da quelli di altri.



Le attività che si possono fare, ma anche la filosofia di fondo e l'ambiente social della piattaforma, fanno di Scratch lo strumento fondamentale per il **Creative Computing**

Il nome Scratch deriva da *scratching* la tecnica usata dai disk jockey hip-hop che trascinano avanti avanti e indietro i dischi in vinile per fondere insieme brani musicali. Scratch ti permette di fare qualcosa di simile; miscelare in modo creativo diversi tipi di media: testi, foto, suoni, grafica, musica; con l'obiettivo di realizzare creazioni multimediali.



Progettato e sviluppato dal **Lifelong Kindergarten group del Massachusetts Institute of Technology (MIT) Media Lab** Scratch è utilizzato da milioni di ragazze/i e persone di tutte le età nel mondo.

Ecco qualche numero della comunità Scratch :

circa **1.400.000** membri registrati - oltre **400 mila** utenti che hanno caricato progetti - oltre **3 milioni** di progetti caricati - oltre **54 milioni** di scripts

Scratch è utilizzato in **tantissimi ambienti e strutture educative**, formali ed informali, tra cui: scuole, università, centri doposcuola, campi estivi, librerie, musei, hackerspace, librerie.

PERCHÈ INSEGNARE IL CREATIVE COMPUTING

Già nel presente ma ancor di più nei prossimi anni a venire non ci sarà attività professionale, qualsiasi essa sia, dove si potrà fare a meno di conoscenze e competenze in ambito scientifico-tecnologico integrate con quelle umanistiche.

Non che sia necessario divenire matematici od ingegneri, ne super-tecnici di computer, ma sarà necessario essere **Consapevoli** del mondo digitale, delle sue funzionalità, dei suoi meccanismi e dei suoi modelli; insomma sarà necessario essere **Fluenti Digitali**. E sarà (è) necessario sviluppare un *pensare sistemico (computational thinking)* assieme ad un *pensare creativo*. Pena il divario digitale che separerà quelli che “sanno” da quelli che “non sanno”, e quindi quelli che troveranno un lavoro, che svilupperanno idee, progetti ed impresa, da quelli che non lo potranno fare.



Aiutare i giovani ha sviluppare **Competenze** e conoscenze nei settori definiti STEAM (scienza, tecnologia, ingegneria, arte, matematica), anche se poi faranno i giornalisti, gli archeologi o i poeti, li metterà nelle condizioni di vivere in armonia e con maggior sicurezza e capacità il rapporto con il “mondo”.

IL PIANO FORMATIVO

Gli obiettivi delle attività formative sono:

- apprendere in modo creativo e divertente i principi, i concetti ed i fondamenti della **programmazione**
- divenire **consapevoli** e competenti sul mondo digitale: da nativi digitali a fluenti digitali; da consumatori a **creatori**
- acquisire capacità e competenze per esprimersi in maniera **creativa**, e collegare le idee con la loro realizzazione.
- acquisire capacità e competenze per creare prodotti digitali multimediali
- sviluppare il pensiero strutturato (**computational thinking**) assieme al pensiero creativo
- **DIVERTIRSI**



Con SCRATCH si possono creare attività formative in grado di fornire un'esperienza motivante, significativa e divertente: corsi/laboratori; workshop, eventi, contest

Il **modello** proposto per i Corsi/Laboratori di **Creative**

Computing&Scratch si basa su tre elementi: *Tem*, *Concetti*, *Attività*

1 esplorare Tem diversi: *arte, storie&animazioni, giochi, simulazioni, musica*

2 sviluppare i Concetti base della programmazione e della computer science: *la sequenza, le iterazioni, il parallelismo, le condizioni, gli eventi, e poi l'uso dei dati e degli operatori*

3 realizzare Attività

- **creare progetti** ossia realizzare dei moduli multimediali ed interattivi, animazioni, giochi, simulazioni, arte, basati sui propri interessi.
- **debugging** ossia investigare su progetti Scratch non funzionanti e trovare la soluzione
- **remix** ossia partire da progetti di altri per crearne dei propri

ed altro.

Sarà inoltre possibile progettare percorsi che includano esperienze con

interfacce e dispositivi esterni quali **Leg**
o Robot, **Arduino** o la scheda **Makey Makey** che permette, ad esempio, di poter giocare con Scratch utilizzando un joystick fatto in casa



oppure di suonare uno strumento autocostruito come il **MaMa Clarinet**

IL PROGRAMMA

PROGRAMMA GENERALE

INTRODUZIONE

introduzione al Creative Computing

Scratch: il programma, la comunità, il cloud

le basi del movimento, le basi del suono, le azioni ripetitive

Modificare i Colori, I personaggi (creazione, modifica, cancellazione)

i Dialoghi, gli Effetti grafici

Suoni, voci e musica

Creare la prima animazione

IL MOVIMENTO

i passi, la posizione xy, le direzioni / bussola, la rotazione per gradi,

scivolare/settare posizione xy,

L'ASPETTO

creazione modifica eliminazione di personaggi ed oggetti, i costumi associati ai personaggi/oggetti, i vari effetti grafici, le dimensioni, i livelli, il dialogo e il fumetto

I SUONI

creazione, importazione editing di suoni, creazione brani musicali con note,

tempi e strumenti, gestione parametri battute, tempi, pause e volume

gestione degli strumenti

LA GRAFICA

il disegno, la penna, i colori, la dimensione della penna, e la luminosità

I SENSORI

controllo del Mouse e tastiera, controllo contatto, distanza, posizione;

controllo colori; il timer, il rumore di fondo, domande/input da tastiera

VARIABILI E LISTE

creazione lettura scrittura e cancellazione di variabili, variabili locali, globali e di rete

creazione lettura scrittura e cancellazione di liste (database)

GLI EVENTI

gestione sincronizzazioni e coordinamento set istruzioni e dialoghi;
broadcasting, invio messaggi, controllo eventi

CONTROLLI / CONDIZIONI

gestione controlli, esecuzione istruzioni su verifica condizioni (se, se/allora, altrimenti, attendi fino a, ripeti fino a), iterazioni e cicli

GLI OPERATORI

operazioni matematiche (somme, prodotti, radici, funzioni trigonometriche, ecc); numeri casuali scelti tra due estremi di un intervallo, operazione logiche booleane (e, o, non) comparazioni (maggiore, uguale, minore) operazioni su stringhe (unione, lunghezza, lettera)

BLOCCHI E CLONI

creazione e richiamo di procedure; creazione gestione di cloni dei personaggi/oggetti

LE ESTENSIONI

cenni e demo sull'uso di schede esterne che dialogano con Scratch; possibili creazioni di progetti attraverso interfacce esterne

CONCETTI DI PROGRAMMAZIONE SVILUPPATI

sequenza - istruzioni/blocchi uno dopo l'altro secondo un ordine preciso

iterazione/ciclo - ripetizione di istruzioni (ripeti n volte, per sempre)

eventi - pianificare azioni quando accade l'evento x (tasto premuto, ricezione messaggio, ecc)

parallelismo - esecuzione di due o più programmi indipendenti (script) nello stesso momento

condizioni - esecuzione di istruzioni al verificarsi di precise condizioni (se, se/allora altrimenti, fino a quando)

variabili - memorizzazione di numeri e testi in una variabile, lettura e aggiornamento della variabile con nuovo dato

liste - creazione di liste di dati

sincronismo/coordinamento - esecuzione istruzioni in base ad eventi temporali per sincronizzare ad esempio i dialoghi (attendi, quando ricevo, invia a tutti)

operatori - uso di operazioni matematiche trigonometriche e logiche (maggiore, minore, e, o non, ecc)

numeri casuali - selezioni di numeri interi tra un intervallo

controllo - di eventi, di condizioni e dei sensori (tasto premuto, distanza da, sta toccando, ecc)

inserimento dati dinamico - inserimento dati durante esecuzione del programma (chiedi e attendi, registra risposta)

TEMATICHE

Il complesso delle attività formative e di apprendimento è organizzato su sei macro tematiche in modo che gli studenti possano esplorare generi diversi per esprimere la propria creatività mentre sviluppano le competenze di programmazione e computational thinking.

1) Introduzione - introduzione al Creative Computing / Scratch con demo ed esempi

2) Arte - studenti esplorano e sperimentano la creazione di progetti che includono: il disegno, la pittura, la musica e la danza

3) Storie - studenti creano progetti narrativi che includano personaggi, scene e dialoghi.

4) Giochi - studenti creano progetti di giochi che includano regole ed

obiettivi

5) Simulazioni - studenti creano progetti che riproducano un a simulazioni

6) Progetti Indipendenti - Studenti creano un progetto indipendente basato sui propri interessi, definendo la tipologia e la pianificazione, collaborando con gli altri per migliorarlo e presentandolo al compeltamento

APPROCCIO DIDATTICO

incrementale - si parte sviluppando piccoli progetti da testare e sperimentare, poi via via si passa a progetti più grandi e complessi

debugging - la risoluzione dei problemi è costantemente presente nell'attività didattica formativa. Oltre che risolvere problemi dei propri progetti, ci saranno progetti non funzionanti ad hoc di cui trovare errori e soluzioni

riuso - (remixing) creare nuovi progetti partendo da quelli di altri o dai propri precedentemente sviluppati. Riutilizzando in tutto o in parte pezzi di codice

modulare - creare progetti mettendo assieme progetti più piccoli

CAPACITÀ GENERALI SVILUPPATE

ragionamento logico e sistemico

pensiero creativo

ricerca errori e problem solving

lavoro in collaborazione e condivisione

concepire l'errore e l'insuccesso come una opportunità e non un fallimento di cui vergognarsi

creazione progetti multimediali ed interattivi

creazione programmi con linguaggio di programmazione visuale

COLLABORAZIONI

Scratch è un'applicazione che permette lo sviluppo di competenze trasversali (*soft skill*) nell'area **digitale**, nell'area cosiddetta **STEAM** (Science, Technology, Engineering, Art, Mathematics), nell'area della **comunicazione** e nelle modalità **sociali** di relazionarsi e di produrre in modo **collaborativo** remixando prodotti e codici di altri. Non a caso in tutto il mondo, Scratch è utilizzato in svariati ambienti e strutture educative, formali ed informali, tra cui: scuole, università, centri doposcuola, campi estivi, librerie, musei, hackerspace. Per questa ragione gli Scratch Lab possono fare parte di qualsiasi programma formativo, essere corsi a se stanti od inseriti in corsi di formazione, di istruzione e master di ogni tipo.

L'uso migliore che possano fare le scuole delle aule informatiche è quello di introdurre il Creative Computing / Scratch nelle attività didattiche, attraverso ad esempio, lezioni settimanali di un ora per classe



Adriano Parracciani

Scratch Educator

info@adrianoparracciani.it

www.adrianoparracciani.it



[o-o] WeBot

> Robotica Educativa > Creative Computing > Tinkering