

Promuovere la creatività con attività di programmazione

La prima parte di quanto segue è derivata dal materiale preparato da Martina Kabátová (kabatova@fmph.uniba.sk) e Katarína Mikolajová (mikolajova@fmph.uniba.sk) - Dept of Informatics Education, Comenius University, Bratislava – per lo Scratch WORKSHOP “Fostering creativity through programming” durante ISSEP 2011 del 29 ottobre 2011. Anche nella seconda parte alcune delle attività hanno la stessa origine. **Grazie dunque a Martina e Katarína che ne hanno concesso l'uso.**

Informazioni utili:

- **Sito web di Scratch:** <http://scratch.mit.edu>
- **dove è il download:** l'ambiente di programmazione Scratch è disponibile in molte lingue e può essere scaricato gratuitamente <http://scratch.mit.edu/download>
- **Sito web per gli insegnanti:** <http://scratched.media.mit.edu>

Cosa è Scratch?

Scratch è un ambiente di sviluppo programmi con un proprio linguaggio di programmazione usando il quale ciascuno può creare facilmente storie interattive, animazioni, giochi, musica e arte. Importante componente del progetto è la comunità Scratch sul web dove si è invitati a condividere le creazioni realizzate in modo che altri possano vederle, guardare come sono fatte e magari modificarle mettendo in pratica l'approccio guardaCapisciModifica con cui gli autori hanno concepito il progetto.

Gli strumenti programmabili che consentono agli utenti di creare animazioni, giochi interattivi, storie, simulazioni ecc dovrebbero consentire agli utenti di personalizzare i loro progetti in modo semplice - cambiare sfondi, oggetti, scegliere immagini da librerie di immagini messe a disposizione o fare figure proprie, scene o le immagini soddisfacenti lo stile e le preferenze personali.

Creando e condividendo attività Scratch i ragazzi possono imparare concetti matematici e computazionali importanti, e insieme imparano a pensare in modo creativo, a ragionare in modo sistematico e a lavorare in modo collaborativo.



Sommario. Obiettivo del laboratorio è esplorare attività e approcci che consentano agli studenti di esprimere le loro idee tramite la programmazione e offrano modo di sviluppare la loro creatività durante questo processo. Ci si concentrerà su più progetti multimediali e giochi che forniscono agli studenti un'opportunità di usare la loro immaginazione e contemporaneamente danno loro una certa libertà di esprimere se stessi in modo insolito. Cercheremo di introdurre attività selezionate sul campo di tipo diverso e con risultati aperti. Saranno illustrati problemi risolti con vari approcci.



1. Introduzione a Scratch

Vediamo per prima cosa un esempio e apriamo l'attività FarfallaBlu realizzata in una classe di scuola secondaria di primo grado. Guardiamo la schermata iniziale nella figura qui sotto.

Nella finestra compaiono: in orizzontale in alto una *barra-comandi*, di cui diremo via via, e tre colonne verticali a loro volta divise in due o più sezioni (o sottofinestre):

- **Colonna destra in alto – Stage o Scena:** come abbiamo visto in FarfallaBlu e come dice il suo nome è lo spazio in cui le creazioni prendono vita, cioè dove sono “recitate” le storie.
- **Colonna destra in basso - Lista degli Sprite o attori:** Quest'area contiene l'elenco degli attori che compaiono nelle storie. Selezionate gli attori con un clic.
- **Colonna centrale in basso - Area degli script o parti:** ogni attore recita cioè si comporta come specificato dalla sua parte o Script. Uno Script è un programma eseguibile associato a un attore ed è composto di comandi che vengono scelti tra quelli della colonna sinistra di cui si dice tra poco
- **Colonna centrale in alto - I tab Script, Costumi, Suoni:** pigiando questi tab script, costumi, o suoni associati agli attori si definiscono le loro parti cioè i loro script, i costumi cioè le immagini con cui appariranno durante la “recitazione” e i suoni (frasi registrate o canzoni o altra musica) che ciascun attore emette in diversi momenti. Provare a pigiare su ciascun tab.
- **Colonna sinistra, in alto - Paletta dei comandi (o block palette):** le istruzioni con cui possiamo realizzare le attività sono molte ed il loro elenco occuperebbe una colonna molto lunga, scomoda da consultare. Per questo le istruzioni sono divise in sei insiemi di comandi corrispondenti ai sei bottoni che si vedono in alto a sinistra ciascuno con un proprio nome e colore. Torneremo sui criteri della suddivisione dell'insieme dei comandi in sei sottoinsiemi. Quando si seleziona un sottoinsieme i comandi relativi sono disponibili nella medesima colonna (a sinistra) ma in basso.
- **Colonna sinistra in basso - comandi (o istruzioni):** contiene i comandi del sottoinsieme selezionato in Paletta. Per comporre gli script i comandi vanno trascinati nella colonna centrale. Le tessere-comando si incastrano insieme come pezzi di un puzzle. Come già detto ci sono diversi tipi di comandi e sono di colori diversi.

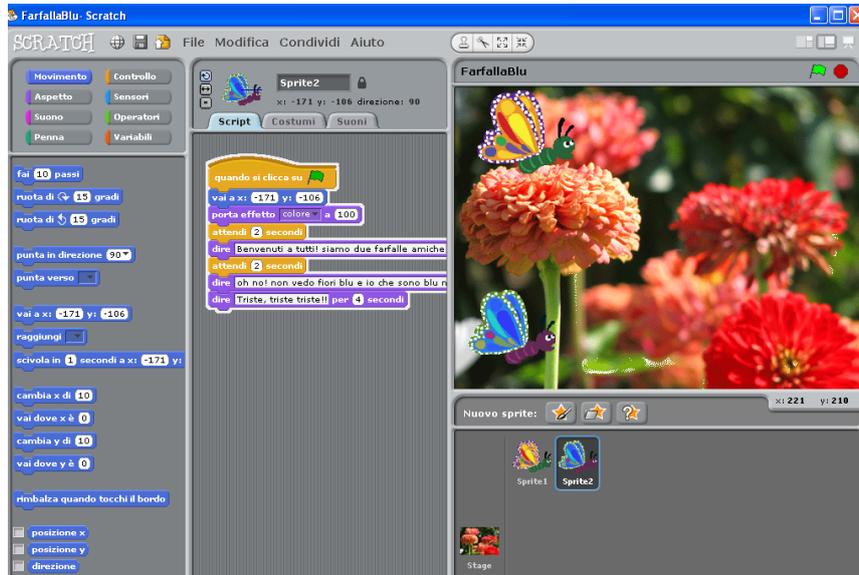


Figura 1. Una schermata per l'attività FarfallaBlu



2. Mani in pasta --- Attività pratica:

Progetto 1

Aprire Scratch e poi il file FarfallaBLu.sb :

1. cambiare le frasi che dicono le farfalle
2. cambiare lo sfondo
3. altro?

Nella figura 2 vediamo invece la schermata iniziale di Scratch senza apertura di nessun file con attività già pronta

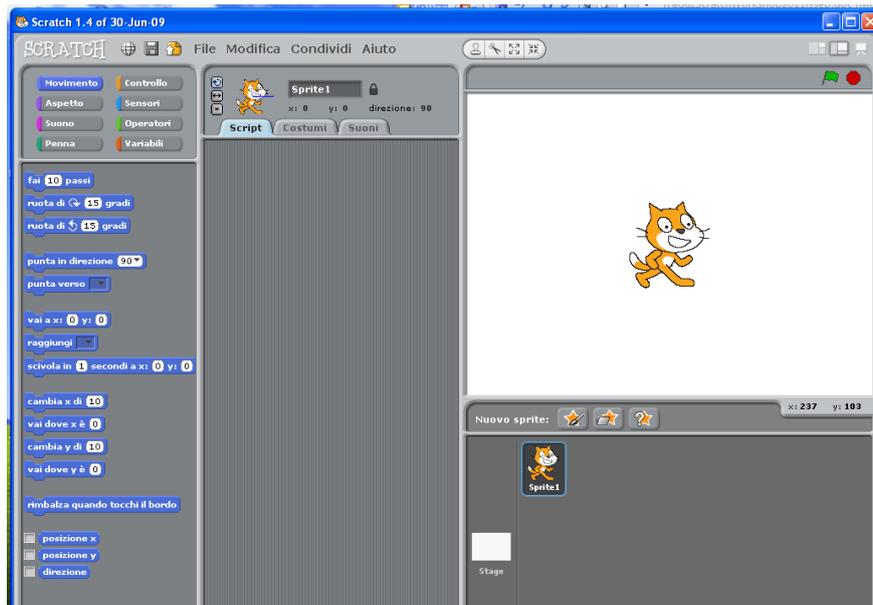
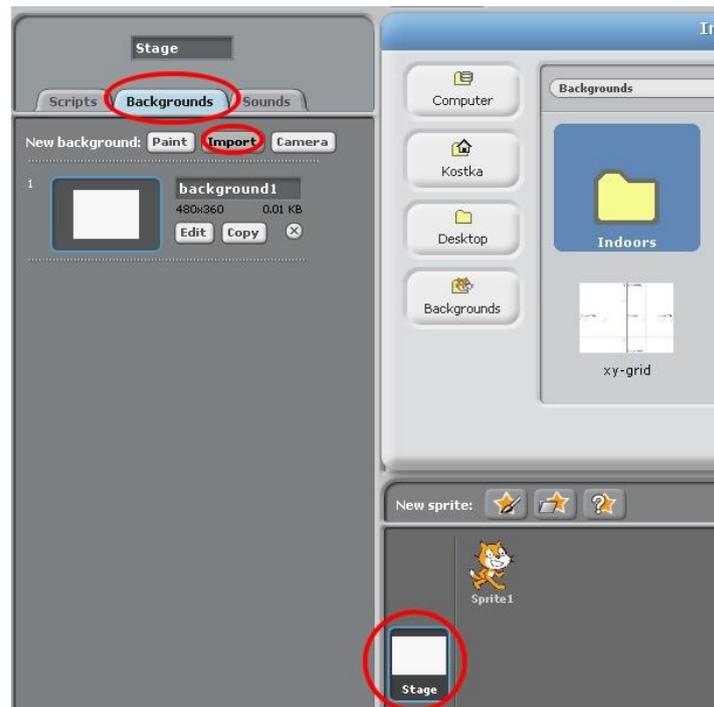


Figura 2. Schermata iniziale di Scratch

Come cambiare il fondale della scena: Clicca su "stage" nella lista degli Sprite. Scegliere "sfondo" tra i Tabs. Fare clic su Importa e selezionare l'immagine che si desidera utilizzare come sfondo per la scena.

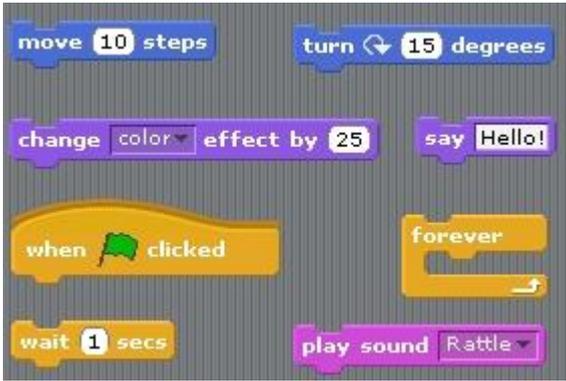


Come aggiungere un nuovo sprite alla storia: Fare clic sul pulsante "Nuovo sprite da file" nella lista Sprite. Scegliere l'immagine per il nuovo sprite.



3. Cominciamo con l' usare otto soli comandi

Attività pratica: Crea una breve storia con un fondale, uno sprite (cioè un attore) ed uno script per quell'attore utilizzando soltanto 8 comandi Scratch.

Progetto 1	<p>Aprire Scratch e poi il file FarfalluBLu.sb :</p> <ol style="list-style-type: none"> 4. cambiare le frasi che dicono le farfalle 5. cambiare lo sfondo 6. altro?
Progetto 2	<p>Ricordiamo che i comandi appartengono a insiemi differenti indicati dal colore della tessera su cui il comando è scritto. Crea una attività usando soltanto gli otto in figura:</p> 
Progetto 3	<p>Migliora il tuo progetto utilizzando altri quattro comandi:</p> 
Fai vedere e illustra	<p>Mostra ai tuoi compagni di classe l'attività che hai realizzata nel progetto 3:</p> <ul style="list-style-type: none"> • Che cosa fa? • Quali sono le prime impressioni nell'usare Scratch come un gioco?



4. La comunità Scratch e altre attività

Dimostrazione: Scratch ha una comunità molto viva di utenti entusiasti. Sul sito ufficiale di Scratch chiunque può registrare e caricare un proprio progetto da condividere con gli altri. I progetti spaziano da strumenti interattivi di musica , giochi , applicazioni didattiche , labirinti , simulazioni , storie animate a tutto il possibile . Alcuni esempi :

- **Pianoforte:** <http://scratch.mit.edu/Attivitàs/Animecat33/1209527>
- **Twig's Adventure- Maisie's Maze:** <http://scratch.mit.edu/Attivitàs/7scratch7/1207684>
- **Fantasy RPG:** <http://scratch.mit.edu/Attivitàs/wedsnesday/1155407>



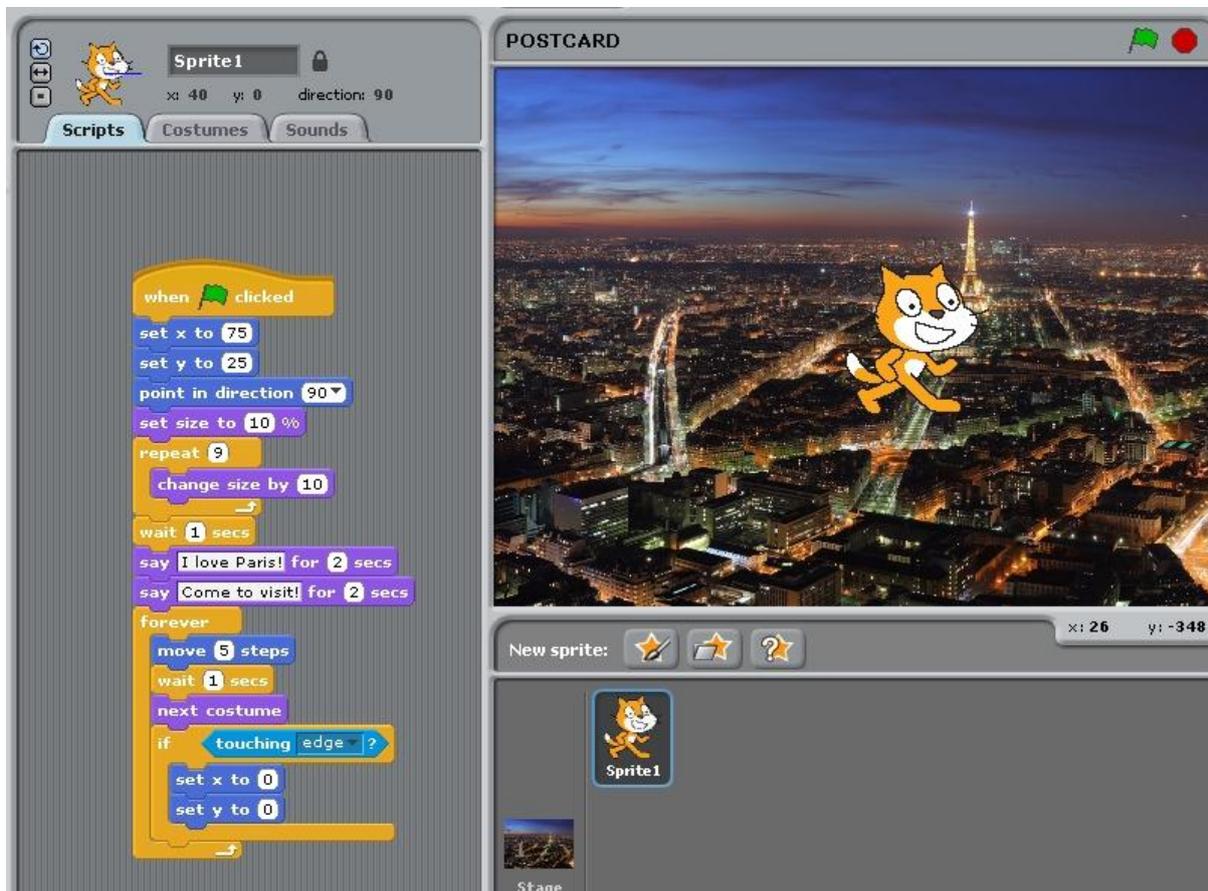
Nel web

Vai al sito web dei progetti <http://scratch.mit.edu/channel/featured> e cerca qualcosa che ti piace. Esplora pochi progetti, cerca di descrivere in quale categoria avrebbero misura (ad esempio storia animata, gioco, labirinto, quiz, ...).



5. Cartoline e interazione

Ora vogliamo creare una “cartolina interattiva” cioè una attività Scratch con reazione di uno degli attori ad una azione proveniente dall'esterno (cioè eseguita da chi ha in mano la tastiera o il mouse per esempio) come accade nell' esempio di cartolina molto semplici già creata su una passeggiata in spiaggia oppure quella cui si riferisce la figura che segue su una visita a Parigi che non è nei materiali.





6. Create la vostra cartolina

Attività pratica: Create la vostra cartolina.

Attività 3

Crea la tua cartolina:

- Scegli una destinazione,
- trova sul web una bella immagine che possa comparire come sfondo della cartolina
- aggiungi degli attori (cioè degli sprite), falli muovere, falli emettere un suono
- aggiungi qualche frase tipo "Benvenuti a XXX"

Cosa abbiamo imparato

Discutiamo insieme:

- Guardate insieme a qualcun altro le vostre cartoline proprie e condividete osservazioni su Scratch venute fuori durante la creazione delle cartoline.
- Scrivete una lista di suggerimenti per chi dovesse o volesse trovarsi anche lui a creare una cartolina
- Quali comandi Scratch hai trovato più utili o anche obbligatori da usare in un progetto interattivo?



7. Invento e costruisco una equazione con una variabile poi la risolvo

Anche con conoscenza di informatica limitate si possono costruire attività interdisciplinari come l'indovinello *IndovinoNumeroPensato.sb* e l'indovinello simile dove si deve capire *PerchéRimaneStessoNumero.sb*. Interesseranno i docenti di matematica.

=== Mettere qui i materiali dei lucidi relativi ===



8. La comunità ScratchED

Demonstration: "A wide variety of educators have been supporting Scratch creators, in both formal and informal learning environments: a teacher who wants to share stories about Scratch and cross-curricular integration; a researcher who wants feedback on materials developed for exploring Scratch as participatory literacy; a parent who wants advice on how to introduce Scratch at a local all-girls high school; a museum program director who wants to connect with other museums who have introduced Scratch. In response to this growing community of educators working with Scratch, we developed ScratchEd. Lanciata nel luglio 2009, ScratchEd è una nuova comunità online in cui gli educatori che usano Scratch condividono storie, **scambiano** risorse, fanno domande e trovano altri con cui condividere interesse per qualche attività

Vai all'indirizzo <http://scratched.media.mit.edu/> e cerca risorse che possano aiutarti su Scratch con la tua classe.

- **Esperienze:** What are your stories about working with Scratch? With stories, you can share your experiences, challenges, and reflections.
- **Risorse:** What helps you to support people learning with Scratch? With resources, you can share all types of content, across different curricular areas and ages.
- **Discussione:** How do I do that? What do you think? With discussions, you can share your questions, suggestions, and insights with others.
- **Compagni di comunità:** Whether close by or far away, there are numerous Scratch educators around the world for you to connect with. Explore the map or browse the members list to find people with shared interests and experiences.

The screenshot shows the SCRATCHED website interface. At the top, there is a navigation bar with the logo 'SCRATCHED learn | share | connect' and links for 'Join', 'Sign In', 'About', 'Help', and 'Contact'. A search bar is also present. Below the navigation bar are four main sections: 'Stories', 'Resources', 'Discussions', and 'Members'. Each section has a corresponding icon and a brief description of its purpose. The 'Learn' section includes links for 'Read Stories' and 'Access Resources'. The 'Share' section includes links for 'Create an Account' and 'Sign In'. The 'Connect' section includes links for 'Join a Discussion' and 'Meet Members'.



Riflessioni sul ruolo dell'insegnante nello spingere i suoi alunni ad esprimere la propria creatività

Feldhusen & Treffinger (1980, in Fasko, 2000) e QCA (1999) hanno scritto consigli per creare un'atmosfera che stimoli la creatività degli allievi e danno suggerimenti agli insegnanti per promuovere la creatività nelle classi:

- teacher should **appreciate and foster unusual ideas**, solutions and responses of students. It is important **to perceive failures or mistakes positively** as a part of the creative process,
- teacher should provide the pupils with **various resources, prompts and support**, enable them to reflect and concentrate,
- teacher should **be open to pupils' ideas or interests and adapt them into lesson plan** if possible. **Unexpected events** can motivate and stimulate pupils to work creatively, it's allowed to put aside lesson plan and 'go with the moment',
- time is very important in the creative endeavor, pupils need **sufficient time to think about creative ideas and explore them**, that might not occur immediately or spontaneously,
- teacher should encourage pupils to **solve divergent problems** which **support lateral thinking**, encourage pupils to ask questions, make connections, work imaginatively, **ask open-ended questions such as "What if...?"** to explore various perspectives,
- teacher should **establish a confidence building approach** where pupils can take risk and freely present, communicate and share their ideas within the classroom. Atmosphere in which **pupils feel safe** is key factor for stimulating creativity,

- teacher should **have fun and laugh with pupils**, respect them, **value what pupils do and say**, **reward originality and imagination**,
- teacher should **participate in creative activities**, act and think creatively as a role model. Demonstrating that he or she is still learning can help to establish an opened atmosphere,
- teacher should **create conditions for teamwork**, where pupils can cooperate with classmates or even with **pupils from a different age group**.

Cosa ne pensate?

Questo ruolo chiede che l'insegnanti abbia piú capacit  di quanto richieda l'instructive teaching?



9. Libera la tua creativit  – idee di attivit 

Attivit  pratica: crea piccole attivit  basandoti sulle idee che seguono. Poi sviluppane una in un progettino piú ampio degli altria tuo piacimento .

Attivit  4

Balliamo insieme

Use several figures to create dance theatre - each figure will dance in a different way. Find suitable background and dance music. You can add some extra effects - e.g. small bird will suddenly fly over the stage. Use color effects and changing of costumes. You may add music and sounds.



Attivit  5

Acquario

Create interactive fish tank. Put inside various fish. They move around like fish do and when clicked they do something - e.g. change the direction, or do flip-flop.

You can add can with fish food - invent some way to feed the fish.

Attivit  6

Dizionario di disegni interattivi

Crea un dizionario di disegni interattivi su un tema particolare, per esempio una fattoria o una biblioteca - find suitable background (farm) and pictures for sprites (animals). When the sprite-animal is clicked it will pop a bubble with English and Slovak name of the animal.



Il castello spaventoso

Create a scene with monsters and spooks that move and make sounds.

L'orchestra degli extraterrestri

Creare una attività in cui ci siano vari extraterrestri. When clicked each of them will play a different instrument. How to make them play all at once?

If you don't have speakers make them jump.



Riflessioni sugli strumenti che favoriscono la creatività

Digital technologies (DT) have the potential in supporting creativity and creative process. Loveless (2002) describes four key features that digital technologies provide for creative process:

- **provisionality** – they provide easy way to make changes or make step back during the performance, to try alternatives, to make drafts or to trace of the thinking process and development of ideas,
- **visualization and interactivity** – DT can provide dynamic, reasonable and immediate feedback on decisions or actions currently made, they can simulate or visualize processes or realities and enable manipulation of variables or changing conditions, so users can better understand and make connections and relations,
- **capacity and range** in which DT provide access to information from all over the world in any time, it's also wide range of tools that digital technologies enable us to use in thinking and problem solving process,
- **speed and automatic functions** which enable users to store, transform, transmit and especially analyze, synthesize, interpret, share and communicate information and ideas more effectively or more intelligibly.

Greene (2002) and Schneiderman (2002) analyze properties and characteristics of digital tools that support creativity. They define some requirements that DT should fulfill to facilitate creativity:

- DT should support "***pain-free explorations and experimentations***",
- enable users **to move back** (return to previous steps) or forward, work continuously and step-by-step,
- there should be **not big penalties for errors or mistakes** and **success should be rewarded** with meaningful response,
- there should be "***immediate and useful feedback for one's action***",
- DT should support **meaningful visualization of data and processes** for better understanding and exploring relationships and connections,
- enable **explore various alternatives and solutions** through 'what-if' scenarios and simulations,
- provide **access to large databases of resources and digital libraries** to find inspiration and gather knowledge,

- **let users to disseminate their outcomes**, products or artifacts to gain reputation and broaden databases of accessible resources.

Discussione

Which of these features are the most important in your opinion?

Would you add some other features?

Do you have some examples of tools that do not support creativity?



Gardner's Multiple Intelligences and Scratch

Scratch can enhance learning and teaching by tapping into the multiple intelligences in the classroom:

	<ul style="list-style-type: none"> ✓ Problem solving ✓ Working with numbers ✓ Using xy coordinates ✓ Performing mathematical calculations <p>Click HERE for Scratch Example</p>		<ul style="list-style-type: none"> ✓ Composing music ✓ Singing (recording voice or uploading audio file) <p>Click HERE for Scratch Example</p>
	<ul style="list-style-type: none"> ✓ Story telling ✓ Speaking and explaining (recording voice) ✓ Creating how-to tutorials for different aspects of Scratch <p>Click HERE for example of student-created video tutorials</p>		<ul style="list-style-type: none"> ✓ Simulation of remote locations; students may not be able to visit far away locations but can imagine and create worlds within scratch. ✓ Developing of natural experiment models; students can simulate activities involving the natural world like rainfall, gravity and verify physical experiments.
	<ul style="list-style-type: none"> ✓ Use of hand & eye coordination (in games etc) ✓ Use of Scratch Sensor Boards to control Sprites from the real world <p>Click HERE for example of Scratch Sensor Boards being used in the classroom</p>		<ul style="list-style-type: none"> ✓ Group work ✓ Collaboration and co-teaching ✓ Leaving comments on other peoples' Projects
	<ul style="list-style-type: none"> ✓ Puzzle building ✓ Manipulating images ✓ Drawing and sketching 		<ul style="list-style-type: none"> ✓ Personal reflection of experiences

Programming robots - L'Alligatore affamato

Explore the new blocks for programming robots and try to make alligator to open and close its mouth.

Try using the motion sensor inside.



10. "Tell me a story"

In second part of this workshop we will create some more complicated Attivit s such as games or stories.

Demonstration: Story creation with Scratch provides opportunities to explore a variety of computational concepts and skills. Here are some blocks that are frequently useful in stories.

WAIT

Insert a pause



VISIBILITY

Make a sprite appear or disappear



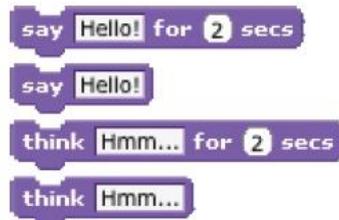
STRINGS

Test, access, and change words and sentences



SAY/THINK

Have a speech or thought bubble appear over a sprite



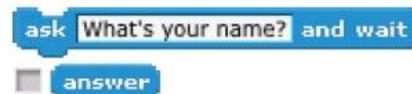
COSTUMES

Change the appearance of your sprite



ASK

Get input to use in a project



SOUNDS

Play recorded audio



COORDINATE

Synchronize actions between and within sprites



Una presentazione

Create your own slideshow – a collection of background images accompanied by audio narration or text titles.

You can put there one character that would comment on the background.

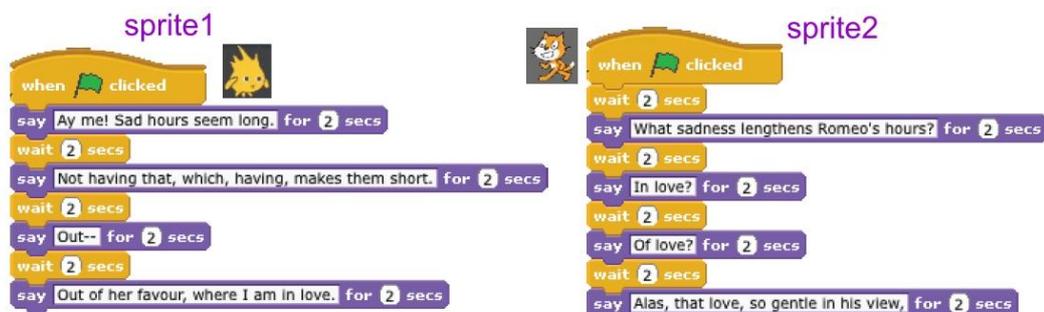


10.1 Crea tu una storia

Attività pratica: Create your own story with characters, various backgrounds, text and sound.

Conversation

Get two characters talking to each other. Use the say and wait blocks to coordinate the conversation. You can look up some dramatic play on the internet to give you inspiration or you can invent your own.



Attività 2

Don't forget to set some appropriate background and find some fitting sprites for the characters.



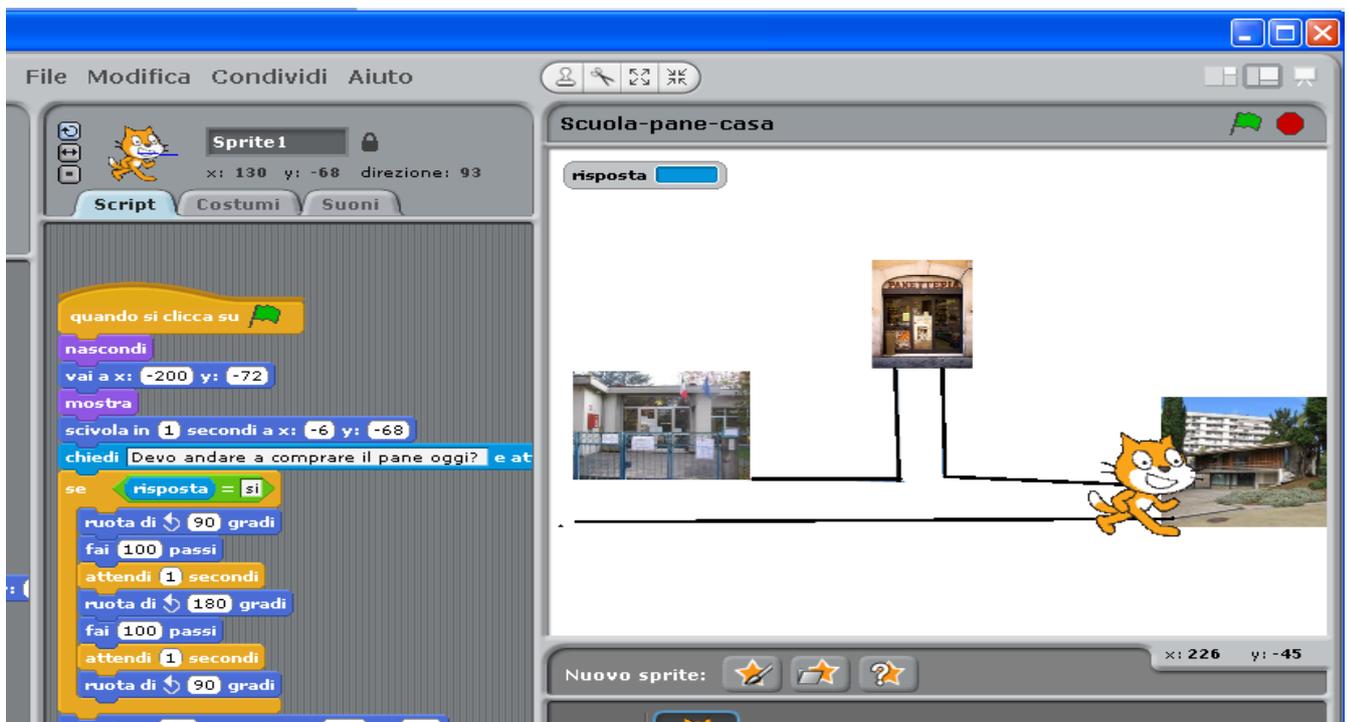


10.2 Racconta una storia tua

Vediamo insieme l'attività ScuolaPaneCasa, poi leggiamo lo script e in particolare ci soffermiamo sul comando di selezione

“se condizione sequenza-comandi altrimenti sequenza-comandi”

Il comando di selezione è stato introdotto con questa storia per sottolineare quanto esprima situazioni quotidiane dove già facciamo delle scelte.



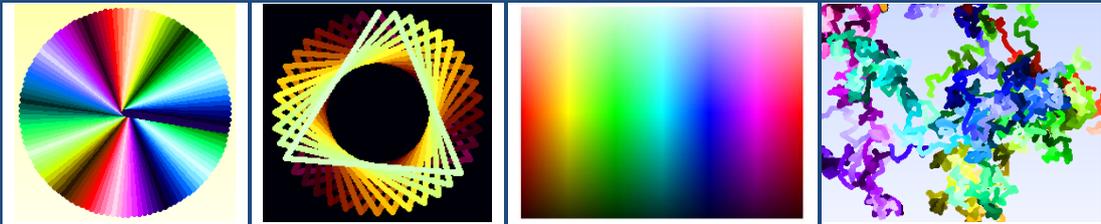
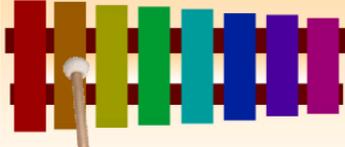
Attività pratica: raccontare in forma di storia Scratch qualcosa anche molto semplice della propria vita meglio se con attività differenti che vengono eseguite a seconda che succeda una cosa o un'altra (ad esempio scelgo di vedere un film romantico oppure un western a seconda che sia in compagnia di una certa amica o di un certo amico).

11. Three steps to foster creativity through given task:

- **use multimedia** to enable pupil's express their ideas in the original way (to raise pupils' motivation, attract them to programming),
- pose the problem where **result is given** but they can **explore and generate their own strategies** to get the result (problem should offer more than one solution to be solved),
- ask pupils to **come up with their own problems to solve** (teacher should be very flexible to adjust the problem so that pupils could solve them based upon their knowledge and skills).



Giochiamo coi suoni, i colori, le forme, gli effetti

Attività 3	<p>Play with colors and shapes</p> <p>Create some nice colorful image. Play with lengths, colors and angles.</p> 
Attività 4	<p>Xylophone</p> <p>Create a xylophone and make it sounded.</p> 
Attività 5	<p>Play with image effects</p> <p>Replace costume of the sprite by your photo. Try to apply some effects when moving mouse over the photo to make "strange things" happen with it.</p>  <p><small>Click the Green Flag. Move mouse over photos. Click the Green Flag again to restart.</small></p>



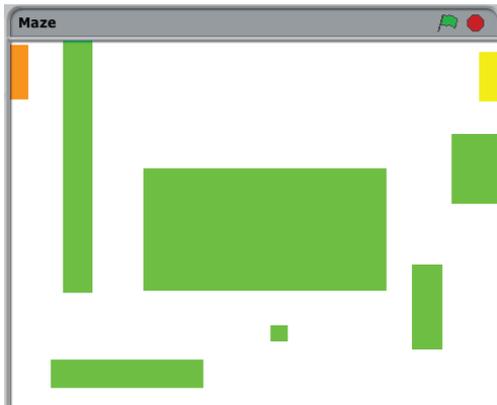
I giochi

I giochi offrono molte opportunità di esplorazione di concetti dell'informatica che finora non abbiamo visto.

Attività 6

Il labirinto

- GOAL: Get from the start of the maze to the end
- RULES: Don't touch the green walls
- OUTCOME: Win when the yellow marker is reached



There will be 7 scripts for the sprite in shape of orange box. Green and yellow shapes are background. The scripts for orange box should do:

- direct the orange box in 4 directions and move it,
- bounce the orange box away from green walls,
- announce "You win!" in case the orange box have reached the yellow box,
- restart the position of orange box in case green flag is pressed.

Parte 2 - Verso attività con più informatica

Nella prima parte di questo lavoro abbiamo introdotto Scratch come strumento per realizzare attività quali storie e indovinelli, giochi semplici. In questa seconda parte introduciamo concetti che ci permettono di creare in Scratch attività con più programmazione.



1. Le variabili.

Introduciamo qui quanto ci serve per realizzare attività più complesse a cominciare dal poter manipolare dei dati per esempio poter calcolare il punteggio di un giocatore impegnato in un gioco.

Attività-1 senza calcolatore. Organizzo un quiz di 5 domande con “premio” finale se chi risponde ha totalizzato un certo punteggio. Decido che una risposta corretta aumenta il punteggio di tre mentre una risposta sbagliata causa una diminuzione di due. Ognuno provi a specificare come procedere per calcolare correttamente il punteggio finale avendo a disposizione un foglio di carta su cui scrivere.

Passo 1. Se si fa una prova interrogando i presenti che tengono ciascuno nota del punteggio in genere tutti calcolano correttamente il risultato. Qui di seguito il procedimento descritto da uno dei presenti:

1. Faccio la prima domanda
2. Sento la risposta
3. Se la risposta è corretta scrivo sulla carta 3 altrimenti scrivo -2
4. Sento la seconda domanda e la risposta
5. Se la risposta è corretta aggiungo 3 al numero scritto sulla carta altrimenti gli tolgo 2
6. E così via ripeto le operazioni 4 e 5 per le domande successive fino alla quinta domanda e relativa risposta
7. Il numero scritto sulla carta è il punteggio finale.

Domanda: come faccio ad essere sicura che sono arrivata alla quinta domanda?

Risposta: conto anche le domande cioè scrivo anche il numero della domanda sulla carta.

Passo 2-a. Qui di seguito riscrivo il procedimento scritto sopra.

1. Faccio la prima domanda
2. Sento la risposta
3. Se la risposta è corretta scrivo sulla carta 3 altrimenti scrivo -2
4. Scrivo sulla carta 1 per tenere conto di quante domande ho posto
5. Sento la seconda domanda e la risposta

6. Se la risposta è corretta aggiungo 3 al numero scritto sulla carta altrimenti gli tolgo 2
7. Aggiungo 1 al numero che conta le domande (per evitare confusione col punteggio gli scrivo accanto un nome per esempio numDomande e do un nome anche al numero del punteggio (lo chiamo Punteggio)
8. E così via ripeto le operazioni 5 e 6 per le domande successive fino alla quinta domanda e relativa risposta. Arrivare alla quinta domanda vuol dire arrivare ad avere numDomande=5
9. Punteggio è il punteggio finale.

Passo 2-b. Altro modo per scrivere il procedimento sopra.

1. Faccio la prima domanda
2. Sento la risposta
3. Se la risposta è corretta scrivo sulla carta Punteggio = 3 altrimenti scrivo Punteggio = -2
4. Scrivo sulla carta 1 per tenere conto di quante domande ho posto
5. Sento la seconda domanda e la risposta
6. Se la risposta è corretta aggiungo 3 al numero scritto sulla carta altrimenti gli tolgo 2
7. Aggiungo 1 a numDomande
8. Se numDomande<5 torno al punto 5 altrimenti proseguo col punto successivo.
9. Il numero in Punteggio è il punteggio finale.

Domanda: E se di domande ne voglio porre sei o sette o un numero n ?

Risposta: semplice: devo arrivare ad avere numDomande=6 oppure 7 oppure n.

Passo 3. Allora per essere più precisi decidiamo quanto segue: dico per prima cosa quante domande farò e poi seguiranno le domande. Qui di seguito riscrivo il procedimento tenendo conto che abbiamo deciso di avere sulla carta tre numon opportune etichette tre etichette per non confonderli tra loro : NumDomande, Punteggio, QualeDomanda.

1. Sento il primo numero e lo scrivo (per non dimenticarlo) accanto al nome NumDomande
2. Faccio la prima domanda
3. Sento la risposta
4. Se la risposta è corretta scrivo sulla carta Punteggio = 3 altrimenti scrivo Punteggio = -2
5. Scrivo sulla carta 1 e gli do nome QualeDomanda
6. Se QualeDomanda<NumDomande vuol dire che non ho ancora finito le domande quindi devo eseguire i punti dal 7 al 9
7. Sento la domanda successiva e la risposta
8. Se la risposta è corretta aggiungo 3 a Punteggio altrimenti gli tolgo 2
9. Aggiungo 1 a numDomande
10. Torno al punto 6 altrimenti proseguo col punto successivo.
11. Il numero in Punteggio è il punteggio finale.

Il quiz può essere giocato con una attività Scratch dove ci sono due attori: quello che fa le domande e quello che risponde. Chi fa le domande deve anche dire se la risposta data dall'altro attore è corretta. Si veda uno scheletro possibile per questa attività in SchelQuiz dove i due attori sono aiutati ciascuno da un gruppo di amici rispettivamente nel porre le domande e nel dare le risposte. Si suggerisce di iniziare provando le istruzioni dello scheletro suggerito, poi cominciare a completarlo, dopodiché modificarlo, alla fine trovandosi con degli script completi (e quindi una storia di contorno alle domande) che possono esser del tutto differenti dalla bozza iniziale.



Creiamo un quiz con punteggio finale

Impariamo ad usare le variabili progettando e realizzando una storia dove vengono poste domande al termine delle quali si da un punteggio.

Progetto 1

Usiamo le variabili

Progettare un quiz con domande su qualche argomento trattato a scuola

Per ogni risposta esatta il punteggio aumenta di 1.

Progetto 2

Come il Progetto 1 ma calcolando diversamente il punteggio, ad esempio togliendo 1 per ogni risposta sbagliata e aggiungendo 2 per ogni risposta esatta.



3. Progetta il tuo gioco

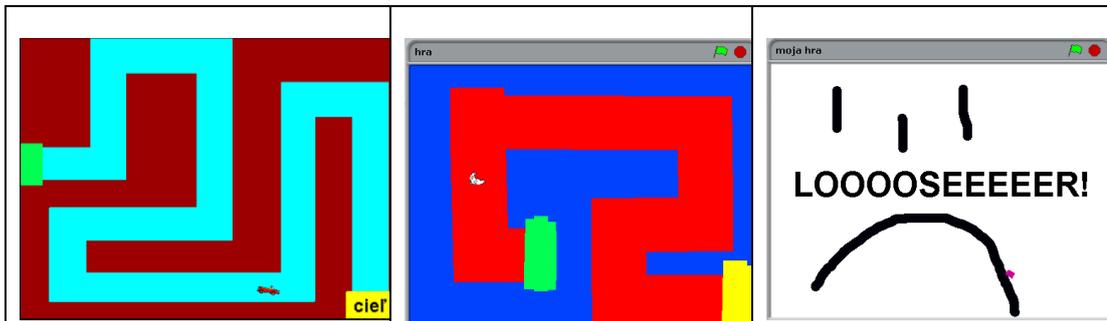
Attività:

Migliora giochi della prima parte

parte

Modification

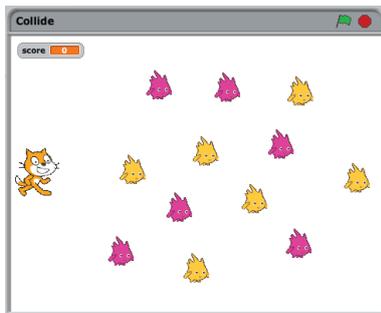
- Make the player moving constantly in direction of mouse. In case the player have reached "the end" box, change the level to the more difficult one. You can add "losing screen" in case the player have reached "the forbidden" area.



Collide

- GOAL: Help the cat navigate a gobo minefield
- RULES: Collect yellow gobos to earn points, avoid pink gobos to avoid losing points
- OUTCOME: Maximize your score

Attività 7



There will be no script for the stage. Each gobo and the cat are separate sprites.

Cat's scripts should:

- reset the cat's position and the score,
- have the cat follow the mouse cursor.

Yellow gobo's script:

- when the cat collides with a yellow gobo, the gobo disappears and the score increases by 10.

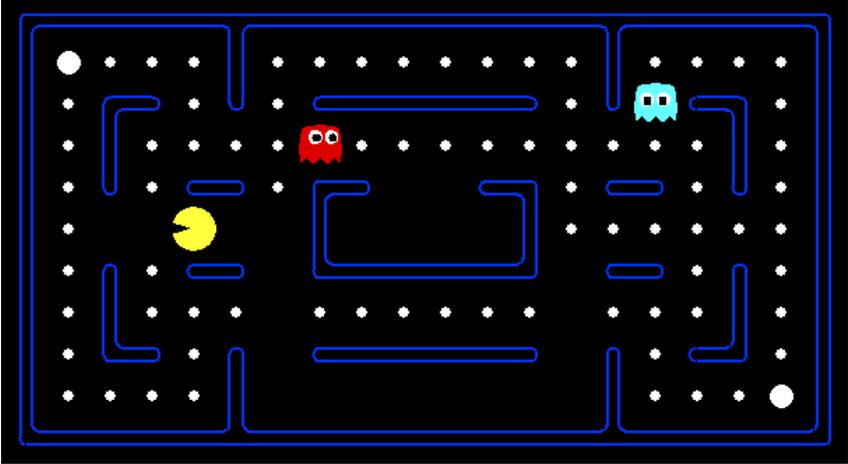
Pink gobo's script:

- when the cat collides with a pink gobo, the gobo disappears and the score decreases by 10.



Idee per attività varie di gioco e non

Hands on activity: Choose one of following prompts or make up your own Attività.

Attività 8	<p>Ezop's fable</p> <ul style="list-style-type: none"> • Find some Ezop's fable on the internet. • Create a short animated story according the fable. • Find appropriate characters and backgrounds, make them talk. • Create a multi scene story with various backgrounds
Attività 9	<p>Pacman</p> <p>Create a pacman game - a character collects dots in a maze and has to avoid monsters.</p> 
Attività 10	<p>Cars</p> <p>Create a game for two players (each one uses different keys). Each of them drives one car on racing track and their goal is to reach the end of track first. If they bump into sidewalls they get penalty seconds.</p>

3. Ancora più variabili

Consideriamo ora un secondo problema da risolvere senza calcolatore cioè in modo unplugged.

Attività 2 senza calcolatore. Uno dei presenti dice una sequenza di numeri (almeno due) e gli altri devono alla fine dire quale è il maggiore di tali numeri. Normalmente tutti trovano lo stesso risultato, vediamo di specificare come.

PROBLEMA: Trovare il massimo di n numeri detti ad alta voce senza usare fogli (su cui prendere appunti o simile). Abbiamo almeno due numeri.

Passo 1. Diciamo ad alta voce una sequenza qualunque di almeno due numeri

Passo 2. Normalmente tutti i partecipanti trovano il massimo dei numeri detti. Chiediamo di pensare al procedimento attraverso il quale ciascuno è arrivato al risultato

Specifica del procedimento. Ciascuno descriva in linguaggio naturale il modo in cui ha proceduto, facendo attenzione a che il procedimento possa essere compreso, per esempio, da un bimbo che sa confrontare due numeri (magari limitandoci a numeri tra 0 e 30 o simile) ma non molto di più

Esempio di descrizione del procedimento in linguaggio naturale (cioè linguaggio usato quotidianamente in contrapposizione a linguaggio formale di cui diciamo più oltre):

1. Ascolto il primo numero
2. Tengo in mente il primo numero
3. Ascolto il numero successivo
4. Confronto il numero successivo con il numero che tengo in mente, se è maggiore tengo in mente il numero successivo, altrimenti continuo a tenere in mente il numero che avevo in mente prima del confronto
5. Chiedo se ci sono altri numeri, se no il numero che ho in mente è il maggiore dei numeri sentiti, FINE attività altrimenti:
6. Ascolto il numero successivo
7. Confronto il numero successivo con il numero che tengo in mente e così via fino a quando non ci sono più numeri.

Riscrivo il punto 7 più precisamente:

- 7'. Confronto il numero successivo con il numero che tengo in mente e vado a ripetere l'esecuzione delle operazioni 3, 4 e 5 fino a quando non ci sono più numeri

Ma allora posso scrivere il procedimento come segue:

1. Ascolto il primo numero
2. Tengo in mente il primo numero
3. Ascolto il numero successivo
4. Confronto il numero successivo con il numero che tengo in mente: se è maggiore tengo in mente il numero successivo altrimenti continuo a tenere in mente il numero che avevo in mente prima del confronto
5. Chiedo se ci sono altri numeri, se ci sono torno al punto 3 altrimenti : il numero che ho in mente è il maggiore dei numeri sentiti

Let's see and discuss some descriptions: are they precise? i.e. someone having no idea of how to perform the given task can get to the solution?

osservazioni: a) Scrivere le varie azioni in disordine: non ha senso è sbagliato

b) Per scegliere tra due diverse sequenze di azioni: come facciamo? Bisogna esprimere: b1) la causa per cui si sceglie l'una o l'altra sequenza poi b2) esprimere le due sequenze differenti.

Stessa procedura espressa in un linguaggio un poco più strutturato – ma struttura definita con regole che decido col “gruppo con cui lavoro”; lo chiamiamo pseudo-linguaggio:

1. Ascolto il primo numero e da ora lo chiamo PrimoNumero
2. NumeroInMente \leftarrow PrimoNumero (stabilisco voglia dire: a NumeroInMente do il valore di quello che chiamo PrimoNumero)
3. Ascolto il numero successivo che chiamo NumSucc
4. Se NumSucc > NumeroInMente allora NumeroInMente \leftarrow NumSucc
5. Se ci sono altri numeri allora vado al punto 3 altrimenti : NumeroInMente è il maggiore dei numeri sentiti.
6. Fine procedimento

Consideriamo ora un terzo problema da risolvere in modo unplugged.

Attività 3 senza calcolatore. Ogni giorno mi viene detta la temperatura media di quel giorno. A fine mese devo comunicare a qualcuno quale è stata la temperatura massima, quale la minima e in quali date si sono verificate. Non posso scrivere su carta ma ho telefono non smart usabile.

Da inserire seguendo i lucidi:

1. usa il cellulare per trovare max. min e media delle temperature di un mese e le date in cui si sono verificate temperature minima e massima
2. memoria cellulare \leftrightarrow memorie calcolatore (persistenti/non)
3. figura con architettura di von Neumann minima e esempi di device di I/O
4. Le variabili di un programma Scratch sono simili alla rubrica del cellulare: <nome, valore> , ma non persistenti



4. Algoritmi e programmi

Diagrammi di flusso o a blocchi o flowchart.

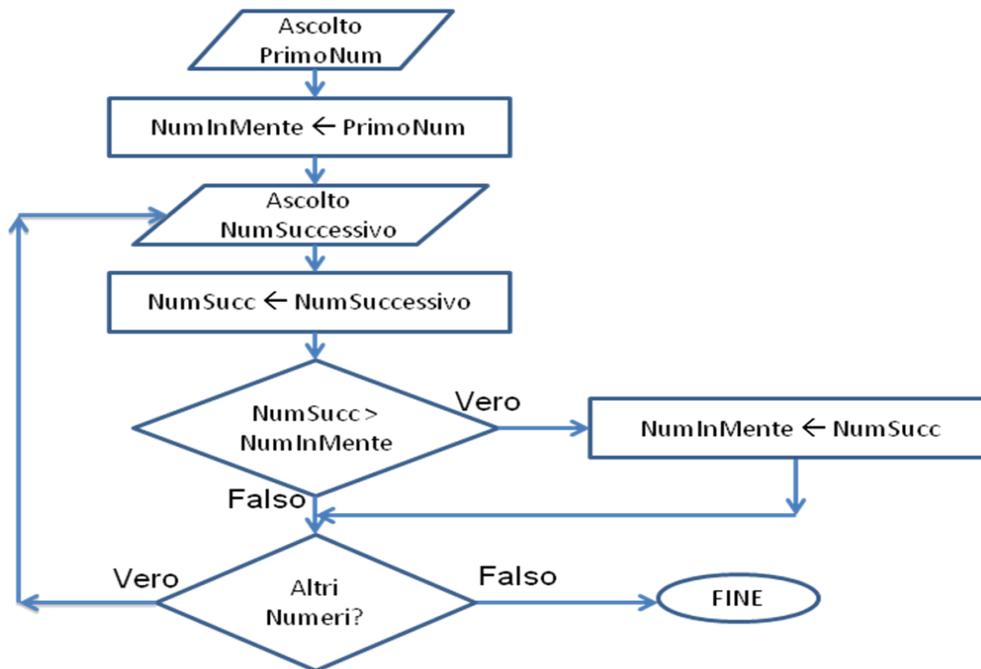
Esistono varie notazioni per la rappresentazione con diagrammi di flusso di algoritmi. Tutte le notazioni sottendono a un meta-modello molto semplice, caratterizzato da una lettura sequenziale:

1. si parte dal blocco iniziale, graficamente rappresentato con una ellissi al cui interno è scritto INIZIO o START,
2. si segue la freccia in uscita,
3. si giunge al blocco successivo e si effettua l'operazione descritta nel blocco
4. si procede iterando i passi 2 e 3 fino a giungere al blocco finale, graficamente rappresentato con una ellissi al cui interno è scritto FINE o END.

Tra le operazioni si distinguono:

- azione, comportano una attività o un'elaborazione, graficamente rappresentata con un rettangolo,
- test, che indicano due o più direzioni di continuazione di un a seconda del verificarsi o meno di una condizione. E' rappresentato con un rombo a due uscite (vero/falso) all'interno del quale è scritta la condizione
- ingresso/uscita, che comportano l'immissione di informazioni dall'esterno oppure l'invio di informazioni verso l'esterno, rappresentati con un parallelogramma.

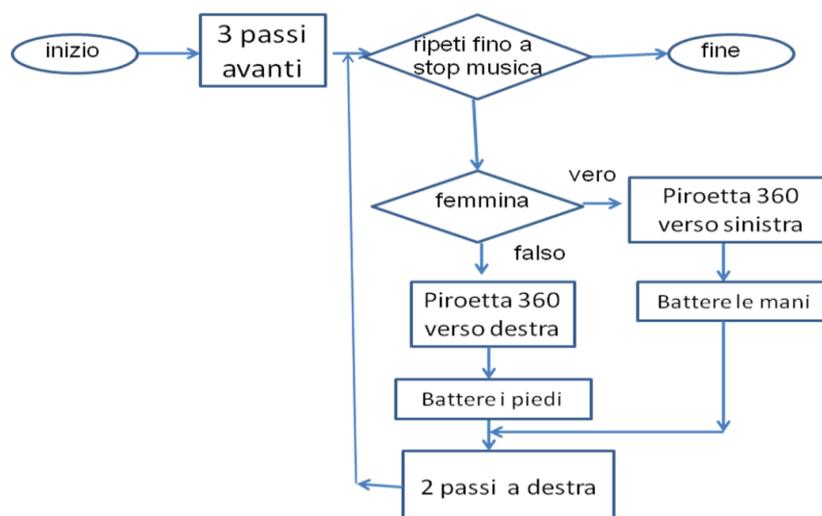
Il procedimento per trovare il massimo in una sequenza di (almeno due) numeri vista in precedenza potrebbe essere specificato con un diagramma di flusso come il seguente:



Discussione. Nella vita quotidiana abbiamo tanti esempi di processi più o meno complessi che ci troviamo a dover comunicare in modo preciso. Indichiamone qualcuno.

One of my favorite and easiest example is the table: “What to do in case the alarm sounds” present in every school. The order of the actions is important.

Other examples are dances: they can be very easy sequences of instructions. Specify a dance in a flow diagram



9. Strumenti che favoriscono la creatività

In questo percorso nel primo capitolo si è voluto avviare il lettore a usare Scratch come ambiente per creare attività mentre nel secondo si sono introdotti argomenti che consentono di progettare e realizzare attività con più componenti di programmazione.

Crediamo infatti che primo obiettivo della presenza di strumenti di sviluppo software nella scuola primaria sia la promozione della creatività. Nella scuola secondaria di primo grado all'obiettivo creatività si associa quello dell'acquisizione dei concetti fondamentali della programmazione per poter creare attività multidisciplinari. In ogni caso per entrambi i livelli di istruzione e quindi per il primo ciclo in genere stimolare la creatività è obiettivo ineludibile.

10.1 Funzioni caratteristiche

Per questo torniamo a leggere sia Shneiderman 2002 sia Romeike 2007: il primo perché mette in evidenza le proprietà che devono soddisfare (quali funzionalità dobbiamo trovare nei) gli ambienti di sviluppo programmi per promuovere la creatività, il secondo perché ha ripreso le osservazioni del primo calandole nella realtà degli ambienti di sviluppo più noti. Per la verità entrambi parlano di IT e non di ambienti di sviluppo ma se per il primo possiamo dire strumenti digitali per il secondo, dagli esempi e dalle affermazioni fatte, si evince che pensa soprattutto agli strumenti di sviluppo sw.

Shneiderman scrive che per favorire la creatività gli strumenti digitali devono offrire funzionalità attraverso le quali l'utente:

- sia incoraggiato a esplorare e sperimentare e naturalmente possa farlo senza problemi (pain-free)
- possa avere feedback immediato e utile per quanto sta facendo
- possa sbagliare senza grandi penalizzazioni e essere gratificato quando fa qualcosa di buono
- sia facilitato nel fare e disfare
- abbia modo di visualizzare quello che sta facendo anche in corso d'opera
- sia incoraggiato a cercare di estendere quello che sa e cercare ispirazione per fare cose nuove
- possa comporre qualcosa di nuovo per quanto più possibile passo-passo
- sia facilitato nel diffondere quello che ha realizzato per avere valorizzato il suo lavoro.

Romeike riprende Shneiderman e porta esempi di ambienti di sviluppo sw che offrono agli utenti le funzionalità da Shneiderman indicate come proprietà caratteristiche di uno strumento digitale che favorisce la creatività. Per ognuna delle proprietà caratteristiche viene citato anche Scratch, oggetto di questo nostro documento. Col permesso dell'autore riporto qui per intero il paragrafo "5.1 Support for creative practice" in "Three drivers for creativity in Computer Science Education" (in attesa di tradurlo).

"Computer based experimentation and exploration can inspire new ideas and should involve design, simulation, surprise and creation (Mitchell et al., 2003). Programmers utilize the possibility of experimentation allowed by the programming environments by testing out ideas and possible paths to solutions. Using selective trial and error is an essential part of software development (Glass, 2006). Those "what if" scenarios are explored quickly and can easily be changed. This is only possible due to the support of the environment. Scratch, as other programming environments for beginners, makes

exploration even more accessible; operations can easily be executed on the objects and the results can be observed in real-time. Thus also learning to program involves the exploration of the programming environment and language (“What are my possibilities?”) and the experimentation with the objects and constructs they offer (“What happens if I apply them like this?”). The immediate and useful feedback for one’s actions given by the compiler or interpreter supports the experimentation and is quite unique to software development – in other fields the tools to work with either do not provide qualitative feedback automatically or do not point out problems. It can be observed with many students that the action-related feedback promotes a sense of control and with it self-regulated learning. As Shneiderman requires no big penalties from creativity supporting software, the biggest penalty of a SDE is the error-feedback itself. As students in other fields have the teachers or class-mates as only source of feedback, when learning to program this is done by the machine in a not intimidating way. The functioning program will be rewarding for success, e.g. in Scratch this will be a nice animation or game. Supporting for experimentation and consideration of feedback is the certitude that nothing can “break” easily. Scratch offers an easy way to undo and redo the last changes. While the aforementioned points of creative work in other fields only are possible in simulations, in programming the experimentation is done at the product itself. Environments for learning how to program also meet others of Shneiderman’s criteria, e.g. visualizing data and processes, which helps to organize ideas and facts and to discover relationships. The trend to visual programming reflects that creativity-relevant aspect. In Scratch programming constructs (such as loops, operations, broadcasts) are represented as colorful building-blocks, separating the kinds of constructs in different colors. In design oriented approaches data structures and processes are modeled and viewed as diagrams with their relationships and may be automatically transformed into code and vice versa (e.g. with Fujaba). The trend to such a visual, more abstract, level of programming suggests that it will become even more emphasized in the future and thus support creativity better. It is important for a creativity supporting environment to make knowledge easy accessible and provide inspiration. Programming environments allow searching in a help knowledge database and provide a documentation of comments and examples. Scratch additionally offers inspiring example-programs made by other children and provides cards that explain constructs to be used and initiates experimentation with those constructs. Composing a work step-by-step allows to slowly approach a problem by leaving plenty of possibilities open as not all important decisions are made at the beginning as with the topdown design. While top-down design is more emphasized in professional software development, the process of learning to program may be different (Kaasbøll, 1998) and involves a bottom-up step by step approach. It allows to review the work done and to evaluate it in the light of what shall be done. This is important to creative practice, since often the outcome is not set clearly from the beginning. In Scratch this matter is implemented by making it possible to run and review a program at any point of the process. Disseminating the results to gain recognition is important for peer-recognition and for motivation of creative work. As the classroom only offers limited possibilities to present one’s achievements, the internet allows to present them to a wide audience. Scratch has a built-in function to upload a work to an internet server where the programs can be shared and evaluated by other learners and also serve for inspiration to others. Hence Scratch, as other environments for learning to program do, fulfills Shneiderman’s tasks for creativity supporting software“

Finiamo questa lunga citazione di Romeike e del suo lavoro con una ulteriore ma ultima citazione anch’ essa molto importante, dal paragrafo 5.2 “I concetti di computer science sono la base per una pratica creativa con l’ uso della IT” nel lavoro citato: “Se opportuni sistemi sw sono necessari per favorire la creatività in computer science, conoscere i concetti basilari della computer science è indispensabile per concepire usi creativi della IT in altri ambienti. ... tra i concetti basilari è incluso il

pensiero algoritmico ... Conoscere concetti base della computer science permette agli studenti di essere creativi in altre discipline e ambienti”.

10.2 Ribadiamo che l'ambiente di sviluppo Scratch è tra questi strumenti

Come abbiamo già scritto Scratch compare per ciascuna funzionalità negli esempi che Romeike fa su quali sw offrano le funzionalità che Shneidermann considera indispensabili per favorire la creatività.

Questo significa che la conoscenza di Scratch, o se vogliamo anche soltanto l'averlo usato qualche volta, fornisce elementi di confronto nell'analisi di come altri sw si comportino riguardo ciascuna caratteristica. Per questo pensiamo che l'insegnante debba puntare ed insistere su questi aspetti più che su quelli di avvio alla programmazione in sé quando propone Scratch ai suoi allievi ma soprattutto quando lo propone ai colleghi durante una iniziativa di aggiornamento. In questo senso Scratch è modello di ambiente di sviluppo che favorisce la creatività degli utenti.

Qui di seguito riprendiamo ciascuna caratteristica puntualizzando dove è stata oggetto di osservazione in queste note magari aggiungendo un esempio o una nota dove necessario ribadire l'attenzione.

L'utente va incoraggiato a esplorare e sperimentare e naturalmente deve poterlo fare senza problemi (pain-free). La funzione di “aiuto” o help disponibile pigiando col tasto destro del mouse è un esempio e così la eseguibilità di un unico comando. Lo stesso ambiente favorisce la sperimentazione essendo molto ricco di strumenti interni (registrazione di suoni, file dove sono già disponibili immagini, suoni per creare le prime esperienze)

L'utente deve poter avere feedback immediato e utile per quanto sta facendo La funzione di esecuzione di uno script nella forma in cui si trova ad un certo momento va in questo senso. La possibilità di estrarre parte di uno script ed eseguirlo

L'utente deve poter sbagliare senza grandi penalizzazioni e essere gratificato quando fa qualcosa di buono La gratificazione dell'utente Scratch è la storia che si svolge sulla scena durante l'esecuzione del programma.

L'utente deve essere facilitato nel fare e disfare. Scratch non permette errori sintattici rendendo disponibili o indisponibili insieme di istruzioni diverse a seconda che si siano create variabili e quali oppure no ad esempio.

L'utente deve aver modo di visualizzare quello che sta facendo anche in corso d'opera Contrariamente ad altri ambienti non bisogna arrivare ad avere un certo pattern di codifica per eseguire un programma, ad ogni nuova istruzione si può decidere di eseguire da un punto ad un altro, basta estrarre se il caso la porzione di istruzioni da provare. Si può decidere con una spunta di vedere o non il contenuto di una variabile o renderlo non visibile, e così via.

L'utente deve essere incoraggiato a cercare di estendere quello che sa e cercare ispirazione per fare cose nuove La metodologia tipica degli ambienti Scratch è: cerca, prova, modifica, crea. L'impostazione globale dell'ambiente Scratch incoraggia ad estendere un comportamento guardingo: ad esempio un utente alle prime esperienze sceglie per i suoi script immagini dai file scaricati al momento del download ma dopo un po' vedendo che può scegliere tra alternative possibili gli viene spontaneo esplorare quelle.

L'utente deve poter comporre qualcosa di nuovo per quanto più possibile passo-passo.

L'utente deve essere facilitato nel diffondere quello che ha realizzato per avere valorizzato il suo lavoro. Le comunità utenti e docenti sono caratteristiche di Scratch e molto frequentate.

11.CONCLUSIONI

Quanto si è osservato finora ci porta a concludere che Scratch ha un valore non tanto come linguaggio (su cui in realtà potremmo avanzare anche qualche critica) ma come ambiente che favorisce gli utenti nell'avvicinarsi a capire come progettare oggetti sw e ad effettivamente crearli. Lo stesso ambiente diventa poco pratico quando i programmi diventano lunghi, quando si vorrebbe avere disponibili strutture dati oltre le variabili e le liste, ecc. Insomma diventa insoddisfacente quando l'utente non è più da avviare al progetto e realizzazione di programmi: per quanto riguarda la programmazione l'avvio è l'obiettivo dichiarato degli autori di Scratch (e nostro nel proporre l'uso). Arrivati a questo punto gli utenti che vogliono continuare ad approfondire algoritmi, programmazione e conoscenze degli ambienti digitali in genere passeranno ad altri strumenti.

A termine di questo lavoro citiamo ancora una volta Ralf Romeike dal paragrafo 5.2 dal lavoro Romeike 2007a "I concetti di computer science sono la base per una pratica creativa con l'uso della IT" del lavoro già ampiamente citato: *"Se opportuni sistemi sw sono necessari per favorire la creatività in computer science, conoscere i concetti basilari della computer science è indispensabile per concepire usi creativi della IT in altri ambienti. ... tra i concetti basilari è incluso il pensiero algoritmico ... Conoscere concetti base della computer science permette agli studenti di essere creativi in altre discipline e ambienti"*.

Bibliografia

Fasko, D. (2000) *Education and creativity*. Creativity Research Journal, 13(3-4), p. 317-327. Routledge, Philadelphia.

Greene, S. L. (2002) *Characteristics of applications that support creativity*. Communications of the ACM, Vol. 45, No. 10 (Oct. 2002), Pages 100-104.

Loveless, A., M. (2002) *Literature review in creativity, new technologies and learning*. Futurelab series, Report 4, School of education, University of Brighton. ISBN 0-9544695-4-2. On-line: <http://archive.futurelab.org.uk/resources/documents/lit_reviews/Creativity_Review.pdf>

Qualifications and Curriculum Authority (QCA, 1999): *How can teachers promote creativity?* On-line: <<http://webarchive.nationalarchives.gov.uk/20100823130703/http://curriculum.qcda.gov.uk/key-stages-1-and-2/learning-across-the-curriculum/creativity/howcanteacherspromotecreativity/index.aspx>>

Romeike, R. (2007a) - A Creative Introduction to Programming (About flying Elephants, Dogs, Cats and Ideas!) <http://www.cs.uni-potsdam.de/~romeike/UEWettbewerb/index-english.htm>

Romeike, R. (2007b): Three drivers for creativity in computer science education. In: Benzie, D.; Iding, M. (eds.): Proceedings of IFIP-Conference on "Informatics, Mathematics and ICT: A golden triangle", June 27-29, 2007, Boston, USA.

Schneiderman, B. (2002) *Creativity support tools*. Communications of the ACM, Vol. 45, Issue 10 (Oct. 2002), p. 116-120.