

Pensiero computazionale

Lezione 7 - Strategie didattiche

Alberto Montresor

Università di Trento

2017/05/05

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



Framework per il pensiero computazionale

<http://scratched.gse.harvard.edu/ct/>

Il laboratorio del MIT Media Lab ha sviluppato un framework operativo per l'insegnamento del pensiero computazionale

- **Definizione**

- Concetti computazionali (computational concepts)
- Prassi computazionali (computational practices)
- Riflessioni computazionali (computational perspectives)

- **Valutazione**

- Interviste basate su artefatti (progetti)
- Scenari di progettazione
- Documentazione di progetto

- **Curriculum per il pensiero computazionale**

- Bilancio fra rigidità di percorso e libertà di esplorazione

Concetti computazionali

- **Sequenza:** identificare una serie di passi per compiere un'operazione
- **Cicli:** eseguire la stessa sequenza più volte
- **Parallelismo:** eseguire più cose contemporaneamente
- **Eventi:** una cosa causa l'altra
- **Condizioni:** prendere decisioni in base a condizioni
- **Operatori:** supporto per le operazioni matematica
- **Dati:** memorizzare, leggere, modificare dati
- **Moduli:** definire nuovi elementi del linguaggio

Prassi computazionali

- **Sperimentare ed iterare:** sviluppare un poco, poi provare, poi sviluppare ancora un po'
- **Testare e fare debug:** assicurarsi che le cose funzionino, identificare gli errori
 - Papert: gestione dell'errore
- **Riutilizzare e remixare:** creare qualcosa partendo da progetti esistenti o idee
 - Verso il mondo open source
- **Astrarre e modularizzare:** esplorare connessioni fra il tutto e le parti
 - Gestire la complessità

Riflessioni computazionali

- **Espressività**

- Comprendere che la computazione è un mezzo di espressione
- "Posso creare"

- **Connessione**

- Riconoscere il potere di creare **con** e **per** gli altri
- "Posso fare di più se ho accesso agli altri"

- **Porsi domande**

- sentirsi in grado ("empowered") di porsi domande sul mondo
- "posso (utilizzare la computazione per) pormi domande per capire come funziona il mondo (in senso computazionale)"

Valutazione

- "Interviste" basate su artefatti (progetti)
 - Protocollo di valutazione:
 - Griglie di valutazione
http://scratched.gse.harvard.edu/ct/files/Student_Assessment_Rubric.pdf
- Scenari di progettazione
 - Computational Thinking Design Scenarios (Studio)
- Documentazione di progetto
 - Scrittura di un diario
 - Commenti nel progetto

Protocollo di valutazione

Defining Scratch

Ask the learner to define Scratch, and explain its functionality:

- *If a friend wasn't here today and asked you what Scratch is, and what you can do with it, what would you tell them?*

http://scratched.gse.harvard.edu/ct/files/Student_Interview_Protocol.pdf

Protocollo di valutazione

Project Feedback

Share two Scratch projects with the learner and ask them to provide feedback to the project creator for one of the projects.

- *Is there anything you would want to ask the creator before giving them feedback?*
- *What suggestions would you give the creator for improving the project? How do you think they could make it more interesting or fun to play with?*
- *Any ideas for how you would do this?*

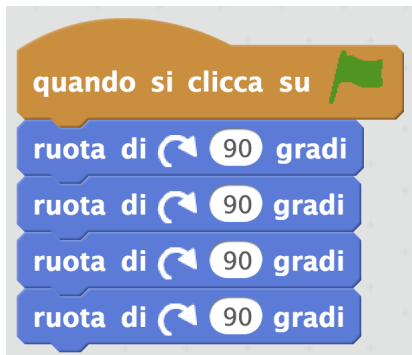
Debug It!

Present the learner with a Debug It! challenge from Scratch Curriculum Guide

- *What's going on?*
- *How would you fix it?*
- *Want to give it try and do what you told me?*
- *Did it work like you expected it to work?*
- *Can you tell me what you think is going on after your changes?*
- *(If the learner is not able to debug the project) Before we move on, where would you go for help if you wanted to fix this?*

Debug it!

Jessica (senz'acca) vuole scrivere uno script per far fare una capriola al proprio sprite. Lo script mostrato qui sotto, tuttavia, non fa nulla!



http://scratched.gse.harvard.edu/sites/default/files/debug_it.pdf

Debug it!

Tomas (senz'acca) vuole muovere il suo sprite con i tasti destro e sinistro, e vorrebbe che lo sprite dicesse "destra" quando si trova nella metà destra dello schermo, "sinistra" altrimenti. Ma le cose non vanno come previsto...

The image shows a Scratch script with four main blocks:

- quando si preme il tasto** freccia destra ▾
 - cambia x di 10
- quando si preme il tasto** freccia sinistra ▾
 - cambia x di -10
- quando si clicca su** [bandierina] (green flag)
 - se** **posizione x** > 0 **allora**
 - dire** Destra!
 - altrimenti**
 - dire** Sinistra!

http://scratched.gse.harvard.edu/sites/default/files/debug_it.pdf

Intermezzo – Cultura dell'errore

Life has meaning only in the struggle. Triumph or defeat is in the hands of the Gods. So let us celebrate the struggle! (Stevie Wonder)

<http://scratched.gse.harvard.edu/resources/debugging-scratch-resources-and-strategies>

Intermezzo – Cultura dell'errore

- Bug walls
- Strategie di debug
 - Leggi il codice ad alta voce
 - Pensa come un computer
 - Rimuovi blocchi di codice per vedere cosa fanno
 - Per tentativi, ma informati
- Le domande giuste

Overheard in the Classroom

Getting stuck is a natural and necessary part of learning, but when students are spinning their wheels they can lose motivation. In these moments, a teacher's words and actions can have a big impact. Here are some things that Ingrid Gustafson, Instructional Technology Specialist in Cambridge, MA, says to help middle school students get unstuck:

"Let's break this down. What should we do first?"



"If I'm the player, how am I playing this game?"

"It seems like you know what you want to do but you don't know how to do it. Can we talk it through?"

"Go to the [Looks] blocks. Is there something in there that's going to help you?"

"If you look at the code like you're reading a sentence, what do you notice? Is something missing?"

"What is it you are trying to do? What do you think isn't working? What have you tried already?"

Intermezzo – Condivisione dell'errore

- Debugging tramite collaborazione
 - Proiezione del codice sulla LIM per discussione di gruppo
 - Pair programming
 - Code review
- Condivisione online
 - <http://scratched.gse.harvard.edu/discussions/discussion-about-scratch/sub-discussion>
 - <https://scratch.mit.edu/discuss/21/>

Protocollo di valutazione

Project Process

Now that the class has been introduced to Scratch, ask the learner about their planning/development process for future Scratch projects.

- (If the learner has had the opportunity to start building up a larger project) *I saw that you were making X. What do you hope this will eventually look like or do? What do you think you'll need to do to make this?*
- *I know you've only just started with Scratch, but after seeing the kinds of things you can do with it, what kinds of projects could you imagine wanting to make?*
- *What made you think of that?*
- *What might you need to do to create the project you just described?*
- *Where would you go to get help for doing different things in Scratch?*

Griglie di valutazione

EXPERIMENTING AND ITERATING	LOW	MEDIUM	HIGH
Describe how you built your project	Student provides a basic description of building a project, but no details about a specific project.	Student gives a general example of building a specific project.	Student provides details about the different components of a specific project and how they were developed.
Describe different things you tried out as you were working on your project	Student does not provide specific examples of what they tried.	Student gives a general example of trying something in the project.	Student provides specific examples of different things they tried in a project.
Describe revisions you made to your project and why you made them	Student says they made no revisions, or only states they made revisions but gives no examples.	Student describes one specific revision they made to the project.	Student describes the specific things they revised in the project and why.
Describe a time when you tried to do something new	Student provides no examples trying to do something new.	Student provides a general example of trying to do something new in the project.	Student describes specific new things they tried in a project.

Griglie di valutazione

TESTING AND DEBUGGING	LOW	MEDIUM	HIGH
Describe a time when your project didn't run as you wanted	Student does not describe a situation that involves a problem with a project.	Student describes what went wrong in the project, but not what they wanted it to do.	Student gives a specific example of what happened and what they wanted to have happen when they ran the project.
Describe how you investigated the cause of the problem	Student does not describe a problem.	Student describes reading through the scripts, but does not provide a specific example of finding a problem in the code.	Student describes reading through the scripts and provides a specific example of finding a problem in the code.
Describe how you fixed the problem	Student does not describe what problems they experienced, or the solution.	Student provides a general example of making a change and testing it out to see if it worked.	Student provides a specific example of making a change and testing it to see if it worked.
Describe how other ways to solve the problem	Student does not provide an example of trying to solve a problem.	Student provides a general example of another solution to the problem.	Student provides specific examples of other solutions to the problem.

Griglie di valutazione

REUSING AND REMIXING	LOW	MEDIUM	HIGH
Describe how you found inspiration by trying other projects and reading their scripts	Student does not describe how they found ideas or inspiration from other projects.	Student provides a general description of a project that inspired them.	Student provides a specific example of project that inspired them and how.
Describe a time you used a part of another project as a part of your project	Student does not describe how they adapted scripts, ideas, or resources from other projects.	Student provides a general description of scripts, ideas, or resources they adapted from other projects.	Student provides specific examples of scripts, ideas, or resources they adapted from other projects and how.
Describe a time you modified an existing project (either someone else's or your own) to improve or enhance it	Student does not describe modifying another project.	Student provides a general description of modifications they made to another project.	Student provides specific examples of modifications they made to other projects and why.
Describe how you give credit to others' work that you built on or were inspired by	Student does not give credit to others.	Student names people or projects that inspired them.	Student describes documentation in project and/or on the Scratch website of the people and projects that inspired them.

Griglie di valutazione

ABSTRACTING AND MODULARIZING	LOW	MEDIUM	HIGH
Describe how you decided what sprites were needed for your project, and where they should go	Student provides no description of how they selected sprites.	Student provides a general description of deciding to choose certain sprites.	Student provides a specific description of how they made decisions about sprites based on goals for the project.
Describe how you decided what scripts were needed for your project, and what they should do	Student provides no description of how they created scripts.	Student provides a general description of deciding to create certain scripts.	Student provides a specific description of how they made decisions about scripts based on goals for the project.
Describe how you organized the scripts	Student does not describe how they organized scripts.	Student provides a general description of how they organized scripts.	Student provides specific examples of how they organized scripts and why.

Griglie di valutazione

EXPERIMENTING AND ITERATING	LOW	MEDIUM	HIGH
Describe how you built your project	Student provides a basic description of building a project, but no details about a specific project.	Student gives a general example of building a specific project.	Student provides details about the different components of a specific project and how they were developed.
Describe different things you tried out as you were working on your project	Student does not provide specific examples of what they tried.	Student gives a general example of trying something in the project.	Student provides specific examples of different things they tried in a project.
Describe revisions you made to your project and why you made them	Student says they made no revisions, or only states they made revisions but gives no examples.	Student describes one specific revision they made to the project.	Student describes the specific things they revised in the project and why.
Describe a time when you tried to do something new	Student provides no examples trying to do something new.	Student provides a general example of trying to do something new in the project.	Student describes specific new things they tried in a project.

Altri approcci alla valutazione

- Portfolio analysis
- <http://happyanalyzing.com/>

Scenari di progettazione

- Spiegare cosa fa un certo progetto
- Descrivere come potrebbe essere esteso
- Sistemare eventuali errori
- Aggiungere feature

Esempio

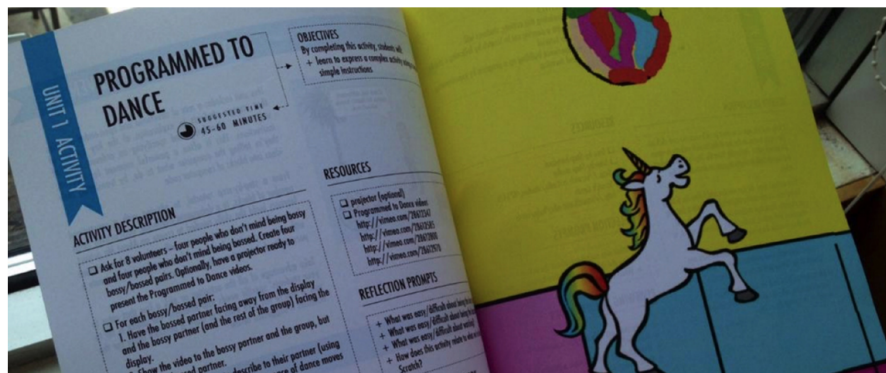
<https://scratch.mit.edu/studios/573426/>

Documentazione

Tre possibilità di lavoro riflessivo:

- Tenere traccia delle proprie attività in forma di diario
 - Esempio
- Utilizzando i meccanismi di Scratch per commentare il codice
- Utilizzando meccanismi di screenshot per produrre tutorial su come realizzare il progetto

Curriculum per il pensiero computazionale



CREATIVE COMPUTING CURRICULUM GUIDE

To support computational thinking in the classroom, we developed the Creative Computing curriculum guide. The guide is a collection of ideas, strategies, and activities for an introductory creative computing experience using the Scratch programming language. The activities are designed to support familiarity and fluency with computational creativity and

Curriculum per il pensiero computazionale

- <http://scratched.gse.harvard.edu/guide/files/CreativeComputing20141015.pdf>
- http://scratched.gse.harvard.edu/guide/files/CreativeComputing20140820_LearnerWorkbook.pdf

Intermezzo - Parson's Programming Puzzles

Definizione

Un meccanismo interattivo per presentare problemi di programmazione limitando le possibilità di errore

Parsons Problems

Check your understanding

sc-1-5: Construct a block of code that correctly implements the accumulator pattern.

Drag from here

```
x = x + 1
```

```
for i in range(10)
```

```
x = 0
```

Drop blocks here

Check Me

Reset

Your program has too few code fragments.

Intermezzo - Parson's Programming Puzzles

A Scratch script with the following blocks:

- porta **Totale** a **0**
- cambia **Conta** di **1**
- cambia **Totale** di **Conta**
- porta **Conta** a **0**
- quando si clicca su 
- ripeti **100** volte
- dire **Hello!** per **2** secondi

<https://scratch.mit.edu/projects/158999635>

Intermezzo - Parson's Programming Puzzles

10 BLOCKS

WHAT CAN YOU CREATE WITH ONLY 10 SCRATCH BLOCKS?

Create a project using only these 10 blocks. Use them once, twice, or multiple times, but use each block at least once.

START HERE

- Test ideas by experimenting with each block.
- Mix and match blocks in various ways.
- Repeat!

FEELING
STUCK?

THAT'S OKAY! TRY THESE THINGS...

go to x: 0 y: 0

glide 1 secs to x: 0 y: 0

say Hello! for 2 secs

show

hide

set size to 100 %

play sound meow until done

wait 1 secs

when this sprite clicked

repeat 10



FINISHED?

Riflessioni sul ruolo del (bravo) insegnante

- **Insegnamento basato sull'apprendistato**
 - L'insegnante è un professionista nel campo dell'insegnamento
 - Il suo ruolo è mostrare tale professionalità agli studenti
 - Gli studenti assumono il ruolo di apprendisti
- **Insegnamento basato sul trasferimento di conoscenze**
 - L'insegnante conosce perfettamente il contenuto del corso
 - Il suo ruolo è trasmettere conoscenza
 - Il ruolo degli studenti è di raggiungere gli obiettivi del corso
- **Insegnamento basato sullo sviluppo cognitivo**
 - L'insegnante parte da quello che il singolo studente conosce
 - Il suo ruolo è quello di individuare i punti deboli degli studenti, e correggere eventuali problemi cognitivi
 - Gli studenti sono coinvolti nel processo cognitivo tramite ad esempio la **peer instruction**

Peer instruction

- Tutoring (coderdojo-style) fra diverse classi di età
- Sviluppo di progetti mirati per l'apprendimento delle classi precedenti
 - Esempio: Studenti di 5a elementare che producono strumenti educativi sulle frazioni per gli studenti di 4a elementare

Quale curriculum?

- Quali sono i modi effettivi per imparare (insegnare) il pensiero computazionale?
- Quali sono i concetti analoghi a:
 - Numeri (Materne)
 - Aritmetica (Primaria)
 - Algebra (Secondaria I grado)
 - Analisi (Secondaria II grado)
- Come integrare questo curriculum nei piani didattici?
 - E' necessario separarla dalle altre materie?
 - "La coperta è troppo corta" (cit. Pavani)

Alcuni link di partenza

ScratchED: <http://scratched.gse.harvard.edu/>

Computing Education Blog: <https://computinged.wordpress.com/>

Alcuni esempi:

- <http://scratched.gse.harvard.edu/resources/favole-al-computer-fairy-tales-computer>