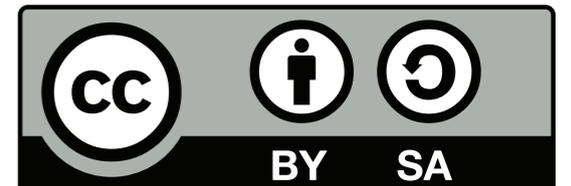


Scratch “avanzato”

Alberto Montresor

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



All the high-school students will be taught the fundamentals of computer technology, will become proficient in binary arithmetic and will be trained to perfection in the use of the computer languages that will have developed out of those like the contemporary Fortran.

– Isaac Asimov (1964)
Previsioni sul 2014.

Presentiamoci

- Qualche parola su di voi...
- Avete seguito uno dei corsi Scratch organizzati qui?
- Avete già utilizzato Scratch in classe?
 - Se sì, come? Quali sono le difficoltà che avete incontrato?
 - Se no, perché? Quali difficoltà vi aspettate?
- Cosa vi aspettate da oggi?
- Qual è il vostro obiettivo nel portare il coding in classe?

Meta-livello, e meta-meta-livello

- Io insegno a voi come aiutare gli studenti a pensare in maniera computazionale
- Metto in pratica con voi quello che insegno
- Voi riflettete su quello che vi dico
 - Vi può piacere o non piacere
 - Posso aver sbagliato il livello (troppo facile, troppo difficile)
 - Il mio obiettivo è comunque indurvi a pensare su come pensano gli studenti
- Ne discutiamo assieme

Papert, Seymour (2005).

You can't think about thinking without thinking about thinking about something.

Contemporary Issues in Technology and Teacher Education, 5(3/4), 366-367.

Programmare è difficile

- Si ragiona a più livelli di astrazione
 - Basso livello - singoli comandi
 - Alto livello - visione d'insieme
- Misconcezioni
- Problem solving
- Pensiero logico, pensiero algoritmico
- Sindrome della schermata vuota

Insegnare a programmare è difficile

- Una materia che l'insegnante medio non ha seguito a scuola
- Una materia che solo di recente è stata portata nelle scuole primarie/secondarie
- Letteratura sull'argomento sparsa, difficile reperimento
- Non è chiaro qual è l'obiettivo

Approcci all'insegnamento della programmazione

Copiare codice

Descrizione "algoritmo"

Sfida

Progetti tematici

Tinkering, progetti personali



+

Scaffolding

-

+

Struttura

-

-

Creatività

+

Curriculum

- Alcuni “curriculum”
 - Creative computing curriculum:
 - <http://creativecomputing.gse.harvard.edu/guide/index.html>
 - <https://www.weturtle.org/dettaglio-ebook/11/creative-computing-guida-completa.html> (in Italiano, registrazione email)
 - Codeclub:
 - <https://projects.raspberrypi.org/it-IT/codeclub>
 - Coderdojo:
 - Es: <http://www.coderdojovr.it/risorse/>

Use-modify-create

- Un buon modello per l'ambiente Scratch

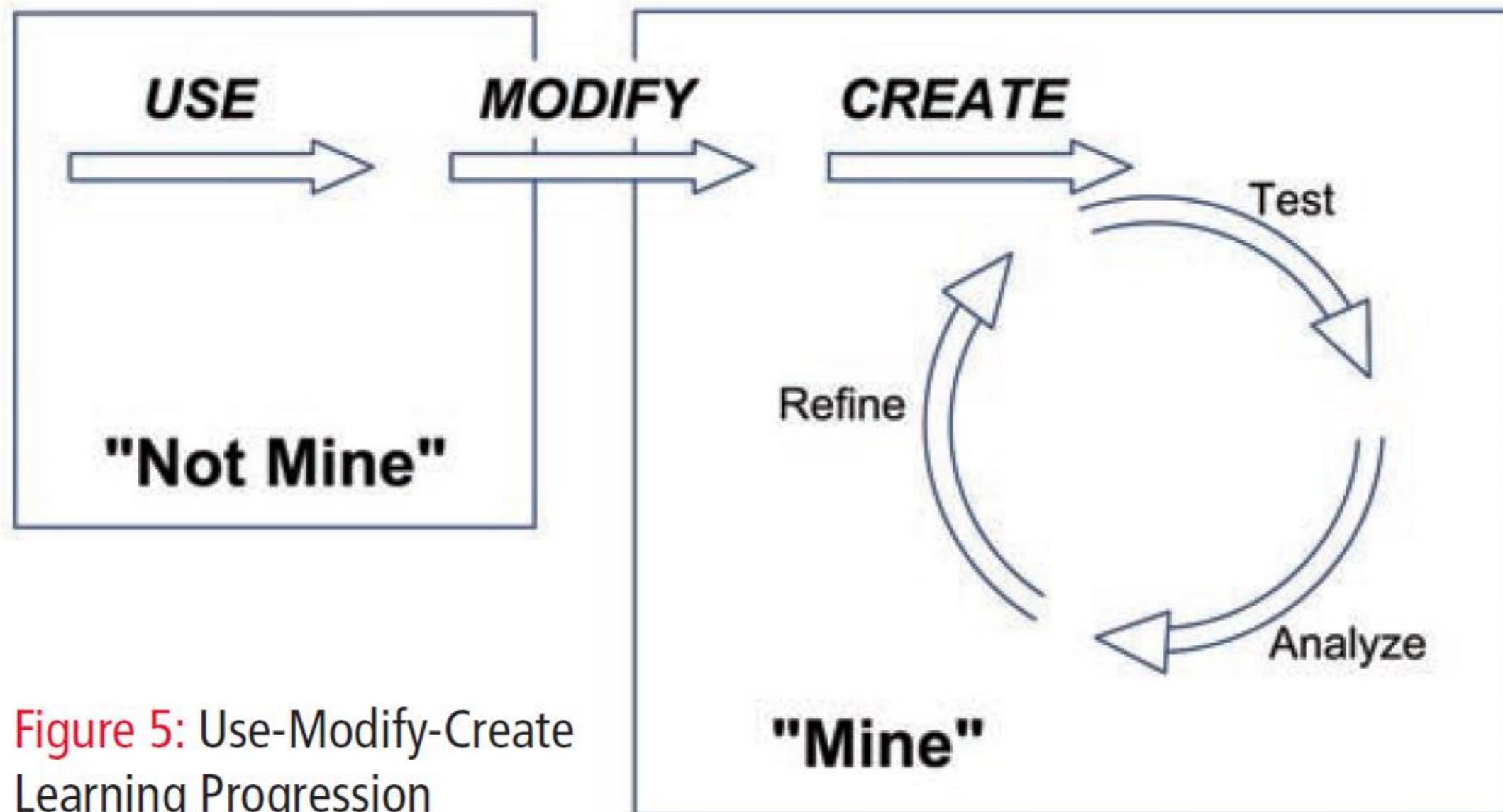


Figure 5: Use-Modify-Create Learning Progression

PRIMM

- **Predict**

Viene mostrato ad uno studente un programma; lo studente deve descrivere il suo comportamento

- **Run**

Lo studente verifica la propria predizione eseguendo il programma

- **Investigate**

Lo studente osserva il comportamento del programma in funzione, osservando ad esempio il contenuto delle variabili

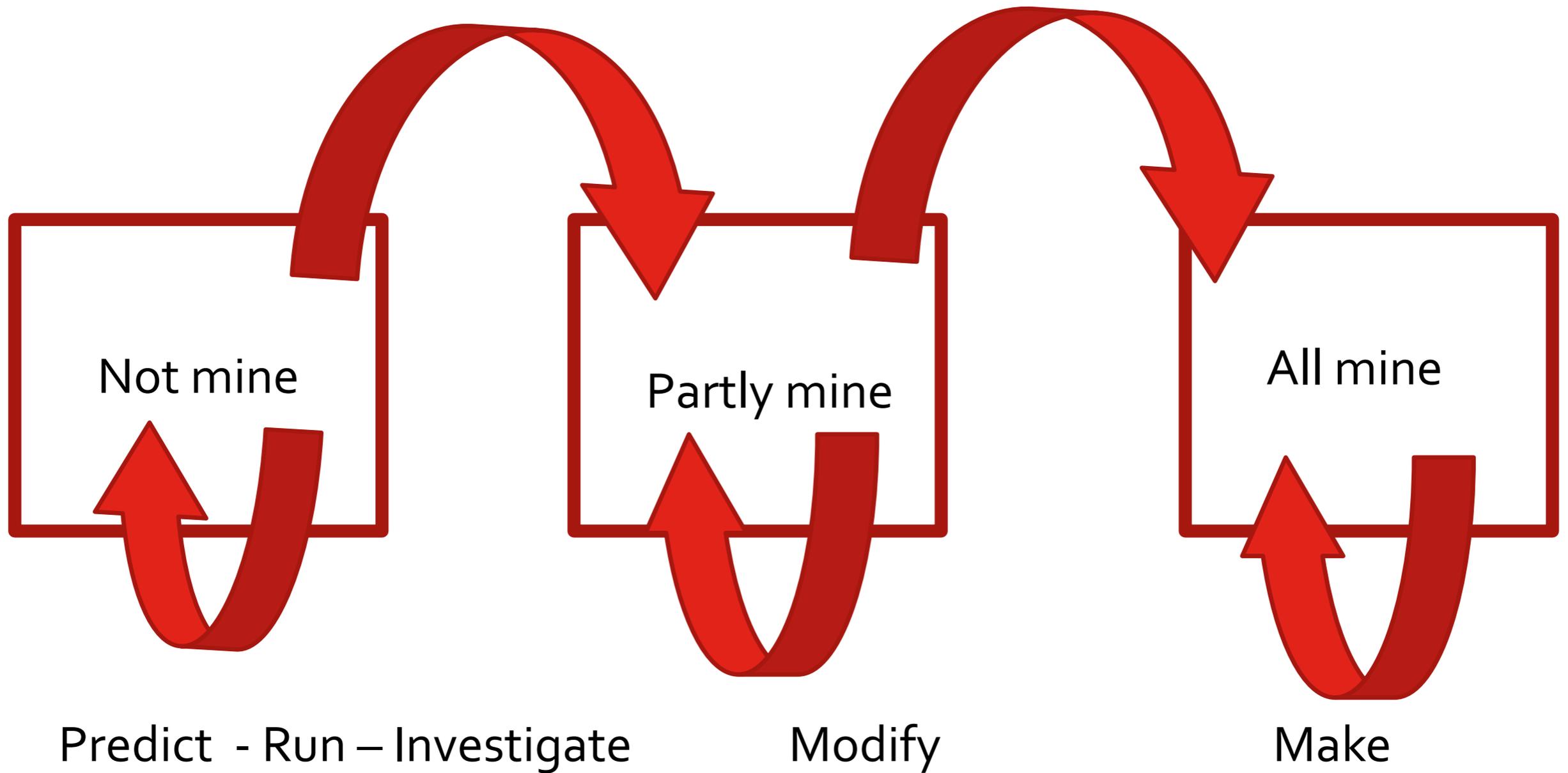
- **Modify**

Lo studente modifica il programma, cercando di ottenere risultati diversi

- **Make**

Si chiede allo studente di scrivere nuovi programmi che utilizzano i principi appena appresi

PRIMM + Use-Modify-Create



Predict



- Cosa fa questo programma?
- Quali concetti devo conoscere per capirlo?

Run

- <https://scratch.mit.edu/projects/326484753/embed>

Investigate

- <https://scratch.mit.edu/projects/326484753>
 - Visualizzate la variabile **VelocitàY**
 - Aumentate il tempo di attesa a 0.2 secondi
 - Si può scrivere in maniera più compatta?
- Suggerimento: provate a lavorare in coppia (~~se vi va~~)

Modify

- <https://scratch.mit.edu/projects/326484753>
 - (a) Oltre a saltare in alto, fate in modo che lo sprite si muova orizzontalmente (salto in avanti)
 - (b) Lasciate una traccia del percorso fatto, utilizzando la penna
 - (c) Opzionale: rendete il percorso più “fluido” possibile
- Suggerimento: provate a lavorare in coppia (~~se vi va~~)

Make

- Cosa vorreste fare con la tecnica che avete appena imparato?

“Read before you write”

- Avete imparato a leggere prima di scrivere
- Imparare a programmare richiede uno sforzo linguistico

“Se volete fare gli scrittori, ci sono due esercizi fondamentali: leggere molto e scrivere molto. Non conosco stratagemmi per aggirare questa realtà, non conosco scorciatoie”.

Stephen King
On Writing

“An algorithm must be seen to be believed, and the best way to learn what an algorithm is all about is to try it”

Donald Knuth
The Art of Computer Programming

Spiegare ad alta voce / Farsi spiegare

- Leggiamo ad alta voce questo progetto
- <https://scratch.mit.edu/projects/26320014/editor/>

- Avete un progetto che volete spiegarci?

Tracing

```
quando si clicca su [bandierina verde]
  porta var1 a 9
  porta var2 a var1
  cambia var1 di 1
  dire var1 + var2 per 2 secondi
  porta var1 a 5
  dire var1 + var2 per 2 secondi
```

Variables Table

Input

Expression evaluator

Output

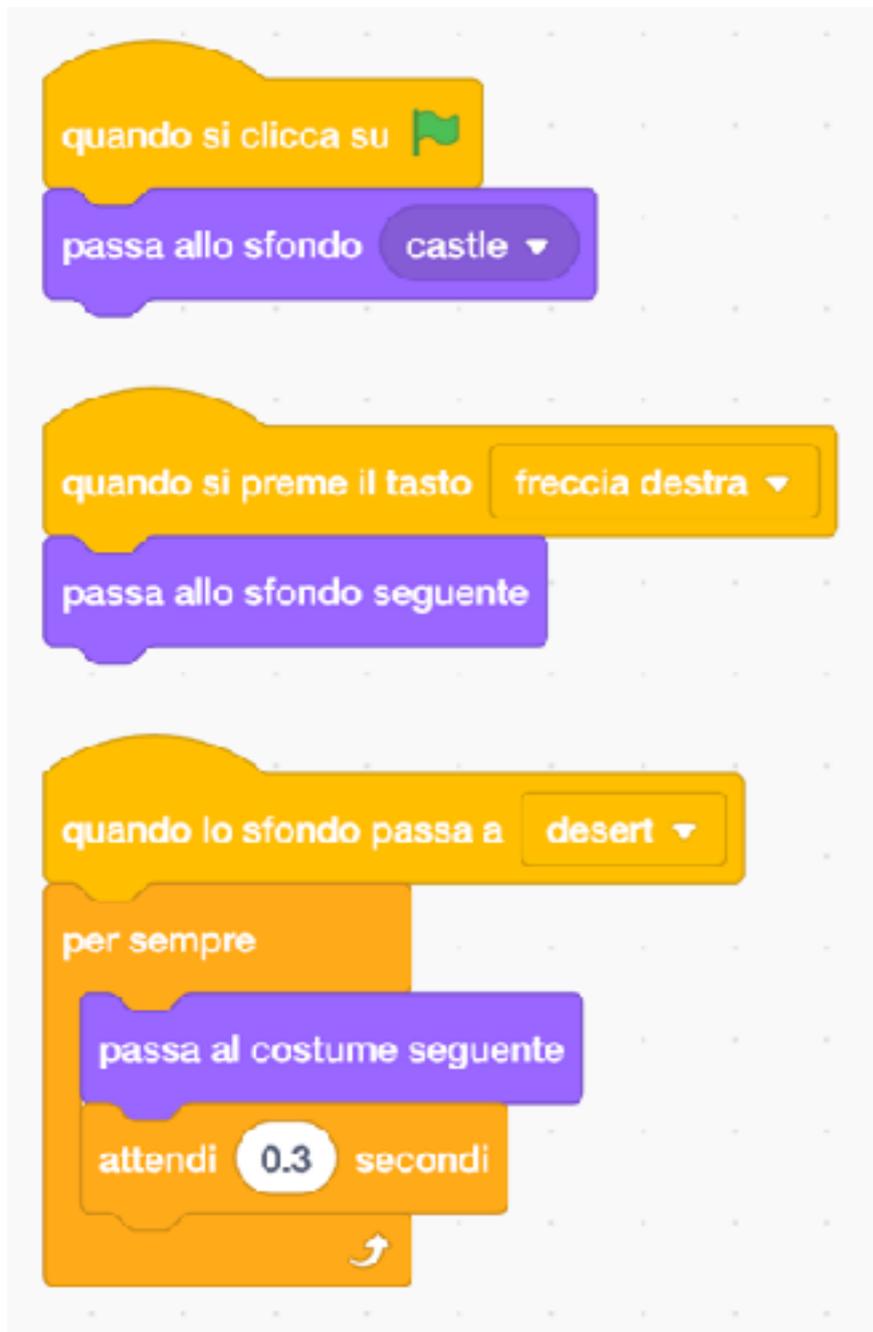
Tracing - Esercizio

- Esercizio:
 - Lavorate in coppia
 - Scrivete un pezzo di codice da proporre come attività di tracing
 - Meta-scaffolding:
 - Deve contenere un “Chiedi () e attendi”
 - Deve contenere un “Se-allora-altrimenti”
- Proponetelo a tutti!

Debugging

<https://scratch.mit.edu/projects/24269131/editor/>

(Da creative computing curriculum)



In questo progetto, la scena cambia quando si preme il tasto freccia destra. Il dinosauro - dovrebbe essere nascosto in ogni scena, tranne quando la scena passa allo sfondo dell'auditorium.

Nell'auditorium, il dinosauro dovrebbe apparire e fare una danza.

Ma il dinosauro è sempre presente e non balla al momento giusto.

Come sistemiamo il programma?

Debugging



- Cosa fa questo programma?
- C'è qualcosa di sbagliato?

Debugging



- Cosa fa questo programma?
- C'è qualcosa di sbagliato?

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?"

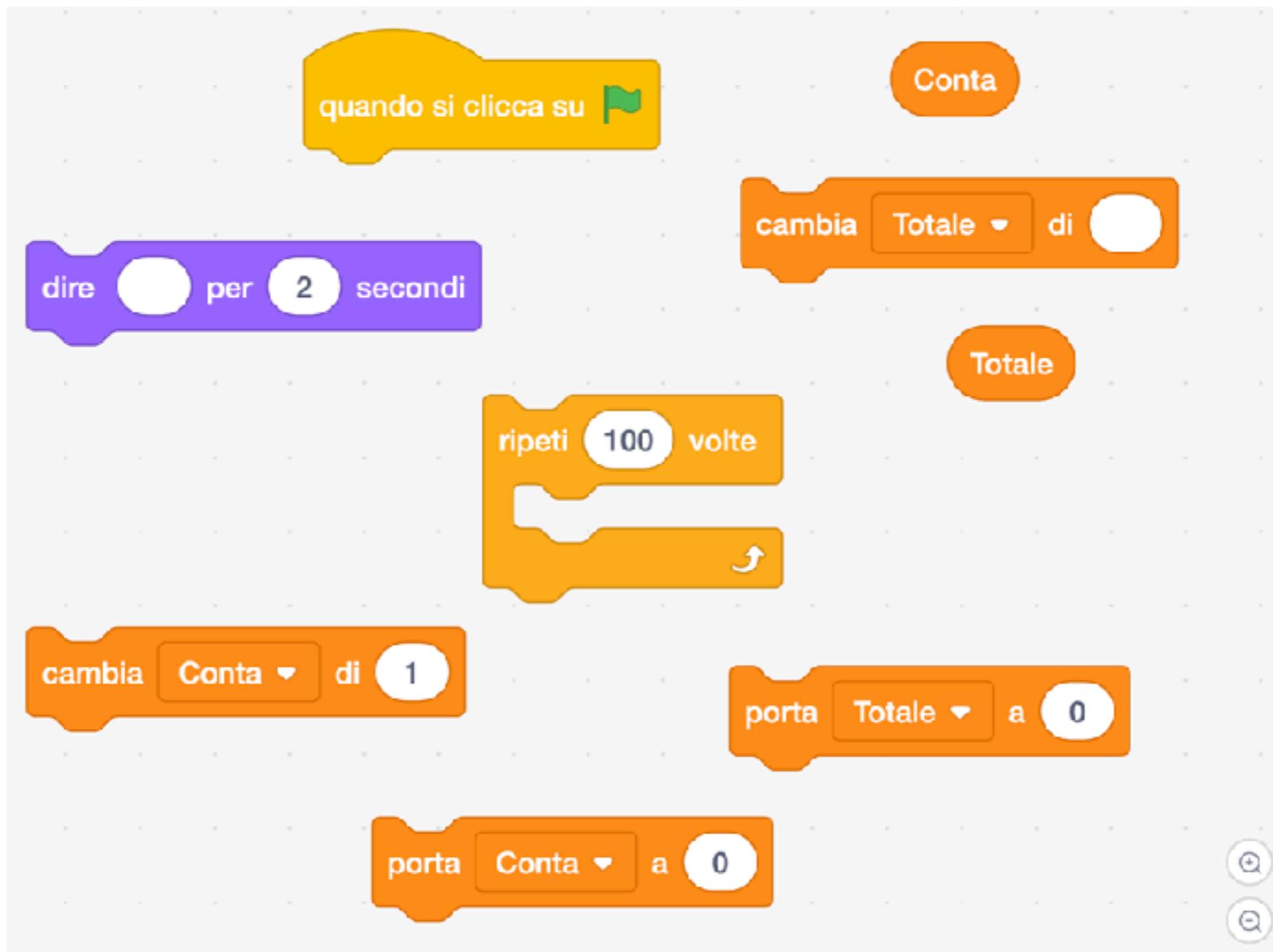
"Go to the [Looks] blocks. Is there something in there that's going to help you?"

"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?"

Parson's Puzzle

- Scrivete un programma che calcola la somma dei primi 100 numeri naturali (Storia di supporto: Gauss a scuola)
- <https://scratch.mit.edu/projects/223801093/editor/>



Micro-pattern: Accumulatore



- Un **accumulatore** è una variabile che “accumula” una serie di valori, per esempio quando si deve eseguire una serie di operazioni
- Totale è un accumulatore

(Parentesi) Approccio creativo

- Parson' puzzle on steroid
 - Provate a eseguire l'attività "10 Blocks"

scivola in 1 secondi a x: 0 y: 0

vai a x: 0 y: 0

dire Ciao! per 2 secondi

mostra nascondi

porta dimensione a 100 %

riproduci suono Miao e attendi la fine

quando si clicca questo sprite

attendi 1 secondi

ripeti 10 volte

Sfida - Unisci i puntini

- Eseguite l'attività "Unisci i puntini"
- <https://drive.google.com/drive/folders/0BzIYujM7kIXnMndvSXdWSWoxLXc>

Micro-pattern: “One-way flag”



- “Bandiera ad una via”
- All’inizio, PrimoPunto è sì (vero)
- Quando viene fatto click per la prima volta, primo punto viene portato a no
- Serve a “ricordarci” che una certa condizione si è verificata

Sfida - Senza scaffolding

- Scrivete un programma che mantiene il “record” di un punteggio
- Per poterlo impostare come esercizio
 - Prendete in input dall'utente un numero alla volta, tramite un ciclo
 - Fermatevi quando l'utente inserisce un numero negativo
 - Al termine, dite qual è il valore più grande che è stato inserito dall'utente
- Discussione
 - Che ne pensate di un approccio del genere?

Micro-pattern - Most-valued holder, sentinel



- “Valore preferito”
- All’inizio, PrimoPunto è sì (vero)
- Quando viene fatto click per la prima volta, primo punto viene portato a no
- Serve a “ricordarci” che una certa condizione si è verificata
- Il valore -1 è una sentinella - ci dice quando smettere

(Alcune) Misconcezioni

- Il computer conosce le intenzioni del programma o di una parte di codice, agisce di conseguenza.
- Il computer è capace di dedurre le intenzioni del programmatore.
- I valori sono aggiornati automaticamente secondo un contesto logico.
- Il sistema non permette operazioni irragionevoli.
- Parallelismo magico: più righe di un programma possono essere attive o conosciute.

(Alcune) Misconcezioni

- Difficoltà a comprendere la sequenzialità delle istruzioni.
- Difficoltà a comprendere i tempi di esecuzione
- L'istruzione if viene eseguita non appena la sua condizione diventa vera.
- Tutte le istruzioni di un programma vengono eseguite almeno una volta.

Misconcezione - Intenzionalità

- Il gatto dovrebbe fare una capriola completa quando si preme spazio. Ma non fa nulla! Perché?



Misconcezione - Parallelismo

- Il gatto dovrebbe fare un balletto e miagolare, ma finché miagola non si muove! Perché?



Misconcezione - Durata dei “se”

- Il gatto si muove a destra e sinistra, e va bene. Ma dovrebbe anche dire se sta a destra o a sinistra, ma dice sempre le stesse cose. Perché?

The image displays Scratch code blocks on a grid background. At the top, there are two yellow 'when key pressed' blocks. The left one is for the 'right arrow' key and contains a blue 'change x by 10' block. The right one is for the 'left arrow' key and contains a blue 'change x by -10' block. Below these, on the left side, is a yellow 'when green flag clicked' block followed by an orange 'if-then-else' block. The 'if' part is a green arrow-shaped block containing 'posizione x < 0' and is labeled 'se' on the left and 'allora' on the right. It contains a purple 'say Sono a sinistra dello schermo!' block. The 'else' part is labeled 'altrimenti' and contains a purple 'say Sono a destra dello schermo!' block.

Intermezzo - Cultura dell'errore

- Sfruttate le misconcezioni in classe
- Chiedete a tutti perché un certo script non va

Pair programming

- Un concetto nato nell'industria
- Molto più efficace in ambiente educativo
- Evidenza che funziona meglio per le ragazze
- Come funziona
 - Coppie di studenti, con livelli simili
 - Uno al computer, l'altro in supporto
 - Scambio di ruoli periodico
- Favorisce la comunicazione sui concetti espressi e quindi l'apprendimento
- Peer instruction

Tocca a voi?

- Nella slide seguente, ho dei problemi (con vari livelli di scaffolding) da suggerirvi
- Prima di “girare pagina”, tuttavia:
 - Volete proporre un problema voi?

Ora vi metto alla prova

- **Pooyan**

- Nessun materiale, studiate il codice
- <https://scratch.mit.edu/projects/29458436/editor/>

- **Sistema solare**

- Tutorial da seguire, suggerimenti per espandere
- <https://scratch.mit.edu/projects/88066172/embed>

- **Mathematical bridge**

- No scaffolding, penna
- <https://scratch.mit.edu/projects/326479349/embed>

- **Esplosione morte nera**

- Tutorial parziale, cloni
- <https://scratch.mit.edu/projects/296726204/embed>

- **Simon**

- No scaffolding, liste
- <https://scratch.mit.edu/projects/31290778/embed>

Ora vi metto alla prova

- Chroma
 - Solo suggerimenti, cloni
 - <https://scratch.mit.edu/projects/153425980/>
- Prof, ho fatto un videogioco. Come faccio a fare una classifica dei migliori punteggi”?
 - Panico!
 - Di cosa abbiamo bisogno?

Mathematical bridge, Cambridge

