T2C - Training to Code

A. Ricci, L. Tarsitano, A. Croatti
CRIAD / UNIBO

[modulo-02]
ELEMENTI DI PROGETTAZIONE E CODING

AGENDA

- Premessa metodologica
  - metafore, chiavi narrative
- Elementi di progettazione dei programmi e coding
  - “come progetto un copione?”
  - “come scrivo un copione?”
    - elementi base per costruire algoritmi

AGENDA

- Premessa metodologica
  - metafore, chiavi narrative
- Elementi di progettazione dei programmi e coding
  - “come progetto un copione?”
  - “come scrivo un copione?”
    - elementi base per costruire algoritmi

RICHIAMI:
RUOLO DEL COMPUTER NELLA VISIONE COSTRUZIONISTA

da strumento di *calcolo*

a strumento di *simulazione*, ***modellazione***

- “computer come creta, pennello”
- linguaggio di programmazione
  => linguaggio di *modellazione*

RICHIAMI: MICROMONDI

- Programmi come *micromondi*
  - ambienti/mondi “computazionali”/digitali/virtuali progettati per favorire l’esplorazione e apprendimento (co-costruito) di concetti e competenze inter-disciplinari

METAFORE E NARRAZIONE

- Introduzione di metafore e “narrazioni” utili a guidare l’ideazione e progettazioni di programmi / microfoni
  - “invarianti" rispetto allo specifico linguaggio usato

METAFORE E NARRAZIONE

- Introduzione di metafore e “narrazioni” utili a guidare l’ideazione e progettazioni di programmi
  - “invarianti" rispetto allo specifico linguaggio usato
- Esempi
  - **Personificazione**
  - **Teatralizzazione**

PERSONIFICAZIONE

- Identificazione di un oggetto, un *personaggio* con cui pensare e interpretare il micromondo, con cui *identificarsi,* “mettersi nei suoi panni” per capire cosa vuol dire eseguire i comandi
  - tartaruga
- Programmare = comunicazione con l’oggetto identificato
  - insegnare alla tartaruga come risolvere un problema o come svolgere un certo compito, in modo *operazionale*

TEATRALIZZAZIONE

- Micromondo come spettacolo teatrale da mettere in scena
  - **personaggi**, **palcoscenico**
- Il programmatore/i programmatori sono i **registi** dello spettacolo
  - coding => scrivere il copione (**script**) dei personaggi
    - sceneggiatura = programma del micromondo nel suo complesso
  - Linguaggio con cui scrivo lo script => linguaggio per comunicare (con gli attori) e modellare (rappresentare lo spettacolo)
  - Testing e debugging => fare le prove, studiare lo spettacolo vedendolo svolgere, ragionare su come migliorarlo
- Gli utenti che mandano in esecuzione il programma sono il **pubblico** (che può interagire)

METAFORE

- Elementi del programma: **attori, scena**
- In Scratch/Snap!:
  - gli attori si chiamano “*sprite*” e la scena “stage”
  - il copione di ogni attore/sprite si chiama **script**
- Spazio e tempo
  - alla scena definisce anche uno spazio (palcoscenico), dove si possono muovere gli attori (in modo che siano visibili dal pubblico)

SCENEGGIATURA E SCENOGRAFIA

- Visione del singolo attore vs. visione d'insieme, di *sistema*
  - "sceneggiatura"
    - programmare la visione d'insieme del micromondo nel suo complesso
  - "scenografia"
    - design della scena dove si muovono gli attori

MICROMONDI E LINGUAGGI

- Importanza cruciale *del linguaggio* nella realizzazione del micromondo
  - definisce il livello di astrazione
    - insieme dei concetti con cui rappresentiamo il micromondo e le sue dinamiche
  - può essere costituito da *simboli* di tipo diverso
    - parole, figure, …
  - ruolo chiave nel governare la “complessità” del micromondo

ESEMPI

- Esempi dal progetto COGITO
  - micromondo tartarughe, insalate e buche
  - micromondo disegno
  - micromondo matematica
  - micromondo parole

AGENDA

- Premessa metodologica
  - metafore, chiavi narrative
- Elementi di progettazione dei programmi e coding
  - “come progetto un copione?”
  - “come scrivo un copione?”
    - elementi base per costruire algoritmi

COME PROGETTARE IL COPIONE?

- Identificazione di principi / metodi / capacità di base per “progettare buoni copioni”
  - ovvero, criteri / metodi / capacità *a supporto del ragionamento per risolvere un problema utilizzando il pensiero computazionale*
- Tre principi/competenze di base
  - **decomposizione**
  - **astrazione**
  - **generalizzazione**

DECOMPOSIZIONE

- *Capacità di scomporre un problema in sotto-problemi più semplici, risolvere questi separatamente e quindi trovare la soluzione complessiva componendo i risultati parziali*
  - cruciale per affrontare la complessità dei problemi
    - supportare l'indeterminatezza gestita
- Aspetto metodologico
  - procedere **incrementalmente** verso la risoluzione di un problema o la creazione di un sistema
  - sia top-down (pianificazione), sia bottom-up (bricolage)

ESEMPI

- Esempi dal progetto COGITO
  - micromondo tartarughe, insalate e buche
  - micromondo disegno

ASTRAZIONE

- *Capacità di ridurre la complessità nascondendo dettagli irrilevanti e focalizzando sugli elementi principali per il problema che si deve risolvere o sistema da costruire*
  - stretta relazione con il linguaggio con cui specifichiamo il micromondo
- Aspetto metodologico
  - supporto al procedere incrementale mediante la definizione *di più livelli di astrazione*, **gerarchici**
  - muoversi in verticale per livelli: zoom in/out, raffinamento/astrazione

ESEMPI

- Esempi dal progetto COGITO
  - micromondo tartarughe, insalate e buche
  - micromondo disegno

DEFINIRSI IL PROPRIO LINGUAGGIO

- Dato un micromondo da progettare
  - capire quali sono i concetti chiave, il livello di astrazione
  - definire l’insieme delle azioni azioni corrispondenti
- Supporto dalla piattaforma Snap!
  - definizione di nuovi blocchi
  - possibilità di nascondere i blocchi

GENERALIZZAZIONE

- *Capacità di adattare la soluzione identificata per un problema specifico per risolvere tutti i problemi relativi ad una classe/insieme/categoria*
- Aspetto metodologico
  - supportare nello sviluppo incrementale momenti di *fattorizzazione* fino all'identificazione di nuovi livelli di astrazione

ESEMPI

- Esempi dal progetto COGITO
  - micromondo tartarughe, insalate e buche
  - micromondo disegno

AGENDA

- Premessa metodologica
  - metafore, chiavi narrative
- Elementi di progettazione dei programmi e coding
  - “come progetto un copione?”
  - “come scrivo un copione?”
    - elementi base per costruire algoritmi

METODI PER SCRIVERE IL COPIONE?

- Approccio “incrementale” nella costruzione dei programmi
  - imparare a ragionare in modo ordinato, per passi, *facendo tuttavia prove intermedie*, per aiutarsi a capire come procedere
    - mandare in esecuzione il programma prima che sia completato
- “Filosofia del debugging” e importanza degli errori
  - imparare a osservare e capire il comportamento

COSTRUZIONE E ESPLORAZIONE

- Bilanciamento di 2 aspetti
  - Costruzione pianificata (volta ad un obiettivo)
    - top-down - scomponendo, a partire dall’obiettivo
    - bottom-up - componendo e adattando esistente (bricolage)
  - Tinkering (“spaciugamento” - cit. Vaccari)
    - esplorazione libera

CODING - QUADRO CONCETTI

- Elementi di base
  - **istruzioni e sequenze di istruzioni**
  - **eventi e comunicazione (prossimo modulo)**

ISTRUZIONI E SEQUENZE

- Script come **sequenza** ordinata di istruzioni
  - importanza dell'ordine con cui vengono eseguite
  - ogni istruzione può essere un comando o azione che ha effetti sull'attore che la esegue o sull'ambiente
- Nel linguaggi visuali (Scratch, Snap!)
  - istruzione come blocchi
  - sequenze => catene di blocchi

MODELLO DI ESECUZIONE

- Modello di esecuzione
  - eseguire uno script (da parte di un attore) significa interpretare ed eseguire meccanicamente una dopo l’altra le istruzioni che fan parte del copione
    - importanza della non ambiguità

ISTRUZIONI: MACRO-CATEGORIE

- Tre macro categorie
  - **Controllo**
  - **Output**
  - **Input**

CONTROLLO

- Sono istruzioni che hanno effetto sul flusso di esecuzione del programma
- Esempi importanti
  - **“ripeti”** => permette di ripetere una o più istruzioni
    - cicli, iterazioni
  - **“se”** => per permette di eseguire istruzioni solo se valgono certe condizioni
    - “istruzioni condizionali”

OUTPUT

- Producono un qualche effetto osservabile “in uscita”, ovvero sul mondo/ambiente
  - es: scrivere un messaggio, disegnare una linea, produrre un suono, inviare una mail, accendere un led collegato al computer o un motorino,…

INPUT

- Solo il duale dell'output, permettono di acquisire in ingresso informazioni/dati dall'ambiente
  - esempi: leggere in input un valore immesso con la tastiera, leggere la posizione del mouse, recuperare il contenuto di una pagina Internet, leggere lo stato di un sensore collegato al computer,…

ESEMPI

- Esempi dal progetto COGITO
  - micromondo tartarughe, insalate e buche
  - micromondo disegno

ISTRUZIONI CON CON PARAMETRI

- I parametri sono informazioni che permettono di caratterizzare/specificare/specializzare effetto di un comando/azione
  - esempio: azione/blocco “dire”
- Applicazione del principio di generalizzazione

ESEMPI

- Esempi dal progetto COGITO
  - micromondo tartarughe, insalate e buche
  - micromondo disegno

EFFICACIA ED EFFICIENZA

- Efficacia dello script
  - quanto efficacemente riesce a descrivere il modo per giungere ad un determinato obiettivo o scopo
    - quanto è sintetico, elegante, ..
- Efficienza dello script
  - quanti passi impiega per per giungere ad un determinato obiettivo o scopo

COMPUTAZIONE

- Capacità di rappresentare e manipolare informazioni, dati, simboli
  - capacità di *rappresentazione* -> descrivere in modo rigoroso
  - capacità di *manipolazione* -> computare, elaborare
  - …funzionali allo sviluppo competenze relative alla capacità di rappresentare e manipolare informazioni
- Concetti
  - Valori, tipi, espressioni
  - Il concetto di variabile
  - la definizione di nuovi blocchi

VALORI, ESPRESSIONI

- Valori e tipi
  - tipi di dati di base, “concreti”
    - numerici, alfanumerici (“stringhe”)
  - tipi di dati “astratti”
    - creiamo nuovi tipi di dati
- Espressioni e operatori
  - combinazione di valori secondo operatori vari

IL CONCETTO DI VARIABILE

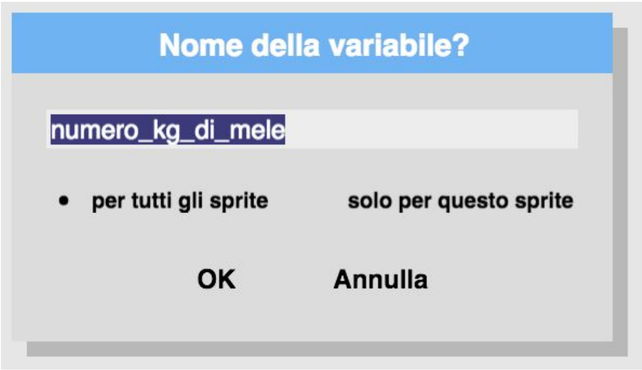
- Competenza di base funzionale alla creazione dei micromondi
  - *saper ragionare usando simboli (chiamate **variabili**) che rappresentano o denotano valori, astraendo da quello che è il valore specifico*
  - esempi
    - numero_di_mele_comprate, temperatura_stanza, paese_da_visitare…

FORME DI CONCRETIZZAZIONE

- Variabile come scatolina o cassetto (mentale) funzionale a tenere traccia di informazioni, etichettate con un nome
  - ha un’etichetta (nome) e un contenuto (valore) che può cambiare nel tempo
- Variabile come *nota* mentale, che serve per tener traccia di valori associati a nomi/simboli
  - come una riga di una tabella a 2 colonne (nome,valore)

VARIABILI: DEFINIZIONE E USO

- Le variabili vanno definite/create
  - specificando il nome



- operazione di *assegnamento* per associare ad una variabile un determinato valore
  - esempio: numero_kg_di_mele := 3

- le variabili vengono *valutate* quando sono usate in usate in espressioni
  - esempio: numero_kg_di_mele * costo_al_kg

UTILITA’ VARIABILI

- Ruolo nel ragionamento
- Tener traccia di dati che cambiano nel tempo
- Progettazione e implementazione di algoritmi
- Gestione dell’input
- Concetto di parametro nella definizione di blocchi

RUOLO NEL RAGIONAMENTO

- Le variabili come *elemento del linguaggio* corrispondente al micromondo
  - importanza della **scelta dei nomi** delle variabili
  - esempio micromondo matematica
- Supporto a forme di *ragionamento astratto e strutturato* (astrazione e decomposizione)
  - esempio micromondo matematica
  - punto molto importante

VARIABILI E TEMPO

- Le variabili come elemento necessario per tenere traccia di valori che cambiano nel tempo
  - esempio: *punteggio* nel gioco la “La Pioggia del Pineto”
    - punteggio := punteggio + 1

VARIABILI E ALGORITMI

- Le variabili sono cruciali nella progettazione e stesura di algoritmi
  - non solo in ambito informatico
- Esempi:
  - fare la somma dei primi N numeri (evitando la formula di Gauss)
  - scrivere una parola al contrario
  - contare le vocali di una parola
    - combinazione con il blocco **se**

VARIABILI E ALGORITMI

- Con tipi di dati strutturati: *liste di elementi*
  - trovare il valore minimo di una lista di numeri
  - ordinare una lista di numeri

VARIABILI E INPUT

- Le variabili come elemento necessario per gestire input
  - esempio micromondo con “Come ti chiami?” / variabile “risposta”

VARIABILI E DEFINIZIONE DI BLOCCHI

- Applicazione del principio di generalizzazione nella definizione di nuovi blocchi
  - da “disegnaQuadrato” di lato prefissato a “disegnaQuadrato di lato X”, dove X è il parametro (variabile) che indica la lunghezza del lato

DEFINIZIONE DI NUOVI BLOCCHI

- Capacità espressiva cruciale: poter definire nuovi blocchi, come composizione dei blocchi esistenti
  - es: disegnaQuadrato
- In ambito informatico: programmazione *procedurale* o *funzionale*
  - nuovi blocchi come procedure o funzioni che, una volta definite, posso usare nel programma richiamandone semplicemente il nome

DEFINIZIONE DI NUOVI BLOCCHI

- Meccanismo cruciale per supportare la progettazione e definizione dei linguaggi associati ai micromondi
  - ovvero il *livello di astrazione*
    - es: quadrato vs. singole linee

UTILITA' PER IL RAGIONAMENTO

- **Astrazione**
  - un blocco (procedura, funzione) => in fase di uso, con un nome identifichiamo una funzionalità/azione/procedura astraendo da come sia implementata, da come funziona
- **Generalizzazione**
  - blocchi con parametri => possiamo risolvere problemi diversi (disegnare quadrati di lato diverso) usando il medesimo procedimento, cambiando solo parametri
- **Supporto a forme di ragionamento induttivo e costruzione "ricorsive"**
  - esempi: calcolo del fattoriale, curva/fiocco di Von Kock

UTILITA' PER LA PROGETTAZIONE

- **Modularità**
  - I blocchi / procedure / funzioni ci permette di creare programmi, "copioni" modulari
    - non solo programmi, ma *ragionamenti modulari*
- **Identificazione e risoluzione errori / debugging**
  - modularità => aiuta a trovare errori - impatto su "debugging"
- "**Riusabilità"** e **"Condivisione"**
  - la possibilità di definire blocchi e suddividere programmi in blocchi, facilità il riuso di porzioni di programmi in progetti diversi e lo scambio fra gruppo diversi