

Pensiero Computazionale e Coding

Bootcamp - Alternanza Scuola Lavoro

[modulo-04]

**ELEMENTI DI PROGETTAZIONE
E CODING /3**

AGENDA

- Da Algoritmi a Sistemi e Micromondi
 - gestione del tempo
 - eventi
 - programmi con più attori (sprite)
 - comunicazione fra attori

DA ALGORITMI A “SISTEMI”

- Dagli aspetti “algoritmici” ad aspetti “sistemistici” => programmi come *sistemi di elementi che interagiscono*
- Micromondi come sistemi
 - comportamento che si sviluppa nel tempo, gestendo eventi *asincroni*, coinvolgendo uno più attori (sprite) che comunicano fra loro
- Esempi rilevanti
 - giochi
 - simulazione

RICHIAMO CHIAVE NARRATIVE

- Personificazione e teatralizzazione
 - micromondo come spettacolo da mettere in scena
- Elementi del programma: **attori, scena**
 - Il programmatore/i programmatori sono i **registi** dello spettacolo
 - Gli utenti che mandano in esecuzione il programma sono il **pubblico** (che può interagire)
 - In Scratch/Snap!: attori => “sprite”, scena => “stage”
- Spazio e tempo
 - alla scena definisce anche uno spazio (palcoscenico), dove si possono muovere gli attori (in modo che siano visibili dal pubblico)

COME SCRIVERE IL COPIONE?

- Identificazione di alcuni elementi invarianti “paradigmatici” che prescindono lo specifico linguaggio con cui descriviamo il micromondo
 - “meta-modello”
- Tre elementi
 - **istruzioni / comandi / azioni**
 - **eventi**
 - **comunicazione**

PROGRAMMI CHE “NON TERMINANO”

- Definizione di comportamenti che possono “non terminare”
 - dal disegno del quadrato, all’alante che si muove
 - blocco “per sempre”



TEMPORIZZAZIONI

- Azioni per gestire il tempo e realizzare temporizzazioni
 - blocco “attendi”  `attendi 1 secondi`

CASUALITA' E VARIABILITA'

- Introduzione di elementi di casualità

- operatore “numero a caso”

numero a caso tra 1 e 10

- Variabilità

IMMAGINI E TEMPO - ANIMAZIONI

- Combinare immagini e tempo
 - sfruttare la possibilità di cambiare “costume” agli sprite
 - blocchi “passa al costume” A purple Scratch block with the text 'passa al costume' and a small downward-pointing arrow on the right side.
- Immagini
 - sia disegni dei bimbi (importate)
 - sia foto (es: fatte con il tablet), sia digitali...

SUONI E TEMPO

- Combinare suoni e tempo
 - possibilità di produrre suoni
 - sia suoni precedentemente registrati
 - sia note

produci suono ▼

suona nota 60 per 0.5 battute

EVENTI

- Nella progettazione e scrittura dei copioni poter specificare *quando* eseguire determinate sequenze di istruzioni/azioni
 - blocchi “Quando” 
- Utilità
 - gestione interazioni “**asincrone**”
 - definire comportamenti “**reattivi**”

EVENTI

- Esempi
 - ▶ “Quando si clicca su bandierina verde”
 - ▶ “Quando sono cliccato”
 - ▶ “Quando si preme il tasto XXX”
 - ▶ “Quando <condizione>”
- Esempio “Pioggia nel Pineto”

quando si clicca su 

quando si preme il tasto spazio ▼

quando sono cliccato ▼

quando 

quando ricevo ▼

PROGRAMMI CON PIU' ATTORI

- Possibilità di creare programmi composti da più attori, ognuno con il proprio comportamento
 - capacità espressiva cruciale per riuscire a costruire micromondi di una certa complessità
- Richiamo alla chiave narrativa
 - *nello spettacolo ci sono più ruoli, ogni attore ha il proprio ruolo e copione*

DIVISIONE DEL LAVORO

- A livello di progettazione: principio “**divisione del lavoro**”
 - si scompone il compito/comportamento complessivo del programma in sotto compiti, che si assegnano ad attori diversi
- Ogni attore/sprite ha il proprio compito, che svolge in modo autonomo dagli altri
 - a seconda del compito da svolgere gli attori/sprite possono essere di tipo diverso (ad esempio visibili/non visibili...)

COMUNICAZIONE FRA ATTORI

- Un meccanismo fondamentale in micromondi composti da più attori è la **comunicazione** basata su scambio di messaggi
 - azione invia un messaggio 
 - evento quando ricevi un messaggio 
- Utile per forme di *coordinamento*, *sincronizzazione*

ESEMPI

- Dialogo e sincronizzazione fra attori in *storytelling*
- Componenti interattivi (GUI) in giochi e applicazioni

CLONING

- Possibilità di creare un nuovo attore/script dal programma di un altro attore
 - azione crea/elimina un clone  
 - evento “quando sono clonato” 