



## Ciclo

### Panoramica:

<b>Spiegazione del concetto</b>	<b>Per sempre</b> - Ripeti più e più volte  <b>Ripeti () volte</b> - Ripeti i il numero specificato di volte  Esempio: camminare, respirare, lavarsi i denti, alba, tramonto, ecc.
<b>Obiettivi formativi</b>	<ol style="list-style-type: none"><li>1. Comprendere il concetto di Ciclo;</li><li>2. Distinguere tra Ciclo di conteggio e Ciclo infinito;</li><li>3. Progetta animazioni usando cicli di conteggio e Cicli infiniti.</li></ol>
<b>Preparazione dell'insegnamento</b>	<ol style="list-style-type: none"><li>1. Una lavagna e un pennarello per lavagna (oppure puoi usare una lavagna e gessetti);</li><li>2. Un Codey e un dongle Bluetooth (o il cavo USB) per ogni studente, ma va bene se 2 o 3 studenti condividono un set;</li><li>3. Un computer con mBlock 5 installato per studente, ma va bene se 2 o 3 studenti condividono un computer;</li><li>4. Distribuisci un rapporto di auto-revisione e un rapporto sul progetto per ogni studente.</li></ol>
<b>Durata</b>	60-90min



## Procedura di insegnamento:

### Fase 1: Ripasso - Sequenza

Ripasso:

- Cos'è il sequenziamento?
- Gli studenti possono pensare a casi di sequenziamento nella vita quotidiana?
- Cosa sono Bug e Debug?
- Come trovare gli errori?

La **sequenza** fa riferimento a una serie di passaggi per eseguire un'attività, come ad esempio, per mettere l'anguria in frigorifero, lavare i denti, lavare i capelli, ecc.

I bug sono gli errori che causano errori nell'esecuzione di un programma rispetto a come era pensato. Il debug è un processo per trovare bug e li corregge. Per trovare bug, dobbiamo controllare i programmi dall'alto verso il basso.

### Fase 2 : Specificare nuove conoscenze - Ciclo

Per abbellire il nostro parco, i giardinieri stanno piantando alberi lungo il marciapiede. I passi per piantare un albero sono: scavare un buco, posare un alberello, coprirlo con terra e avanzare di 5 metri per piantare un altro albero. Se la strada è lunga 20 metri, i giardinieri dovranno ripetere la stessa sequenza di passaggi quattro volte.

Nei programmi per computer, a volte utilizziamo i cicli di conteggio per ripetere il pezzo di codice un numero specifico di volte. In altre occasioni, il ciclo potrebbe essere per sempre, o in altre parole, senza condizioni di chiusura. Nella vita di tutti i giorni, ci sono infiniti cicli attorno a noi, come il tramonto e l'alba. Allo stesso modo, quando scriviamo programmi, usiamo i blocchi **per sempre** per eseguire il pezzo di codice all'infinito.

**Ciclo** e **Ripetizione** hanno lo stesso significato. Se ripeti il fare qualcosa, stai eseguendo un ciclo. Chiedi agli studenti se riescono a pensare a comportamenti ripetitivi e quante volte ripetono. Gli insegnanti possono dare spunti qui: lavarsi i denti, due volte al giorno; respira, per sempre.

### Fase 3: Gioco Iniziale - Tocca a ritmo

**Regola del gioco:**

1. La lettera A rappresenta toccando il tavolo con la mano sinistra; B rappresenta toccando il tavolo con la mano destra; C rappresenta toccando il tavolo con entrambe le mani. Tocca il tavolo seguendo i seguenti ritmi:

ABABABC ABABABC



Il caos potrebbe verificarsi quando gli studenti sono invitati a toccare il tavolo a una velocità elevata. In questo caso, suggeriamo agli insegnanti di chiedere agli studenti se è necessario semplificare il processo semplificando i battiti. Semplifica i battiti per essere

ABABABC Ripeti due volte

2. E se introducessimo un altro beat? Aggiungi un'azione, D per battere le mani:

ABABABC ABABABC ABCABC ABABABC  
ABABABD ABABABD ABDABD ABABABD  
ABABABC ABABABC ABCABC ABABABC  
ABABABD ABABABD ABDABD ABABABD

Semplifica i battiti

ABABABC Ripeti due volte

ABC Ripeti due volte

ABABABC

Ripeti due volte

ABABABD Ripeti due volte

ABD Ripeti due volte

ABABABD

### Suggerimenti :

1. Gli insegnanti possono scrivere i battiti alla lavagna e possono contrassegnare i battiti mentre lavorano con gli studenti per capire le regole;
2. Se vuoi progettare i ritmi da solo assicurati che i battiti siano semplici. Inoltre, non rendere il gioco troppo lungo, altrimenti gli studenti potrebbero perdere presto l'interesse.

### Fase 4: Attività di studio

Quando scriviamo programmi, usiamo spesso i blocchi **Ripeti** per rendere i nostri programmi puliti. Possono salvarci dal trascinare lo stesso blocco più e più volte. Con i blocchi di ripetizione, Codey Rocky può eseguire ripetutamente la stessa sequenza di istruzioni. Quando il blocco di ripetizione viene eseguito una volta, lo contiamo come un ciclo o una iterazione.

La ripetizione può essere per sempre o avviene in momenti specifici. Per esempio, ci laviamo i denti due o tre volte al giorno mentre l'alba e il tramonto avvengono intorno a noi all'infinito. I due blocchi di loop differenti sono i seguenti:



Fai una domanda agli studenti: qual è la differenza tra ripeti per sempre e quelli ripetuti ( ) volte. Indirizza l'attenzione degli studenti su un fatto: il blocco **per sempre** non ha alcuna sporgenza nella parte inferiore che consente di bloccare altri blocchi. Il blocco per sempre è progettato per ripetere i codici all'infinito, quindi non possiamo apporre altri blocchi su di esso.

Ripeti due volte

### Compito 1: Il Pane cotto a vapore non può saltare

Storia: il pane cotto a vapore insiste per l'imparare a saltare. Ha provato molte volte ma non riesce a farlo. Gli insegnanti mostrano l'animazione di un pane cotto a vapore che non riesce a saltare e dimostra come realizzare l'animazione. Successivamente, è tempo che gli studenti lavorino alle loro animazioni allo stesso modo degli insegnanti. Esempio di progetto: The Steamed Bread Can not Jump



#### Program Story:

Il pane cotto a vapore vuole saltare in aria. Ha provato dieci volte ma ancora non ce la fa.

### Compito 2: Il Pane cotto a vapore salta

Storia: un giorno, il pane cotto a vapore ha finalmente imparato come saltare. Continua a saltellare e può andare dove vuole. Chiedi allo studente di progettare un



programma il pane cotto a vapore salta usando blocchi per sempre. Gli insegnanti possono fare riferimento al progetto di esempio: il pane cotto a vapore salta (*Non consigliato per mostrarlo agli studenti troppo presto. Fagli fare un pò di pratica da soli prima*). Progetto di esempio: The Jumping Steamed Bread



#### Program Story:

Il pane cotto a vapore è finalmente sollevato da terra e continua a rimbalzare; Ogni rimbalzo viene fornito con un suono; Il programma verrà eseguito all'infinito.

**Sfida:** Usa un altro evento “Quando stai agitando Codey”

#### **Compito 3 : DIY Project**

Gli studenti lavorano in coppia per progettare animazioni utilizzando cicli infiniti o cicli di conteggio. Devono presentare le loro opere, ma dovrebbero prima completare il rapporto del progetto. Gli studenti dovrebbero mostrare le loro opere seguendo le domande del rapporto.

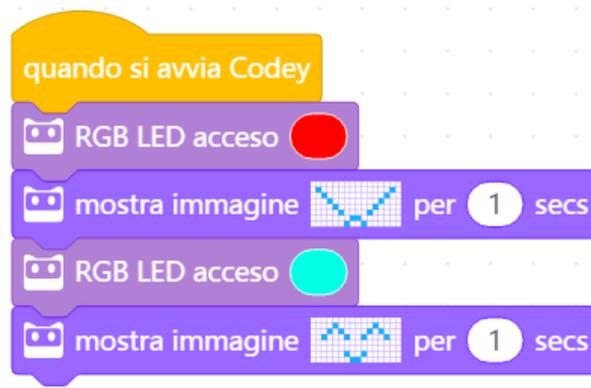
#### **Suggerimenti:**

1. Se gli studenti hanno abbastanza tempo e sono studenti rapidi, digli di progettare animazioni utilizzando più cicli di conteggio o cicli annidati.
2. Gli insegnanti possono adattare il limite di tempo in base agli scopi didattici e alla tipologia di classe. Tempo consigliato: 20 min.

#### **Estensioni:**



1. Progetta diverse animazioni: attivate da diversi eventi (pulsanti) e costituite da una combinazione di cicli di conteggio e cicli infiniti;
2. Aggiungi un mix di suoni e effetti LED RGB per rendere l'animazione emozionante, ad esempio:



### Fase 5: Conclusione

Descrivi a cosa si riferisce il concetto di Ciclo (Loop). Loop significa che la macchina ripete pezzi di codice all'infinito o un numero specifico di volte.

Gli studenti completano il rapporto di auto-revisione.



### Autovalutazione

Nome :

Età :

- Rispondi alle seguenti domande e registra il tuo risultato :

Descrivi ciò che hai imparato con una o due frasi.

Descrivi brevemente cosa ti piace di più e di meno in questa sessione di classe

Quello che mi piace di più

Quello che mi piace meno

Disegna un ciclo di conteggio e un ciclo infinito che è successo intorno a te oggi :

Ciclo di conteggio

Ciclo infinito

Raccontaci una breve storia ( Descrivi come funzionano i seguenti codici dall'alto verso il basso )



	<p><i>Quando Codey sta tremando,</i></p>
--	--

Puoi dipingere come ti senti in questa sessione di classe nell'angolo in alto a destra del rapporto di autovalutazione.



## Rapporto di progetto

**Nome :**

**Nome del gruppo :**

- Segui queste domande per presentare il tuo lavoro e devi dare le tue risposte durante la presentazione.

Quali compiti hai intrapreso? Descrivilo in una o due frasi.

Hai qualche idea su come risolvere i compiti?

Annota o disegna ogni singolo pezzo di ispirazione, che sia buono o cattivo. Lo stai facendo per esplorare più possibilità. Se hai bisogno, usa più fogli per registrare le tue idee.

Descrivi l'effetto finale del tuo progetto e perché scegli l'effetto in una o due frasi

Effetto

*Esempio : quando si preme il pulsante A, il pane cotto a vapore rimbalza dieci volte.*

Perché scegli questo (questi) effetto?

*Esempio : è divertente ed è facile progettare il pane al vapore.*



Hai incontrato ostacoli? Hai qualche soluzione? Descrivilo con una o due frasi. (La soluzione può essere un piano approssimativo)

Le difficoltà <i>Esempio: impossibile connettere Codey Rocky a mBlock ; Impossibile caricare i codici; Impossibile raggiungere il consenso ...</i>	soluzioni <i>Esempio: accendi Codey Rocky; Sasso carta forbici...</i>
---	--

Gli studenti possono rispondere alle seguenti domande dopo la sessione di condivisione.

Ti piace il tuo Progetto? Descrivi cosa ti piace di più e meno del progetto con una o due frasi. Ed eventuali miglioramenti in futuro?

<input type="radio"/> Lo amo <input type="radio"/> Mi piace <input type="radio"/> Così così <input type="radio"/> Non mi piace <input type="radio"/> Odio
Quello che mi piace di più <i>Esempio: l'animazione è vivida; il suono corrisponde all'animazione.</i>
Quello che mi piace meno <i>Esempio: è difficile per le persone raccontare di cosa tratta l'animazione a prima vista.</i>
Migliorare <i>Esempio: ridisegna l'animazione e rendila più semplice questa volta; aggiungi suoni e luci.</i>



Valutazione dell'insegnate:

1. Cooperazione (30%): valutare il rendimento del gruppo per quanto riguarda la divisione del lavoro, la collaborazione e il coordinamento.
2. Completezza (20%): valutare se il progetto è sufficientemente completo. Naturalmente, il progetto deve attenersi all'argomento.
3. Innovazione (20%): valuta la creatività del progetto.
4. Funzionalità (20%): valutare se il lavoro è abbastanza funzionale
5. Difficoltà (10%): valuta qual è il livello di difficoltà del lavoro

CSTA

gradi	identificativo	Standard CSTA K-12 CS interinali	Concetto di Lavoro	Concetto Pratico
K-2	1A-A-5-2	Costruisci programmi, per svolgere un'attività o come mezzo di espressione creativa, che include sequenze, eventi e loop semplici, usando un linguaggio di programmazione visiva basato su blocchi, sia in modo indipendente che collaborativo (ad es., Programmazione coppia).	Algoritmi e programmi	Creazione di artefatti computazionali
K-2	1A-A-3-7	Costruire ed eseguire algoritmi (serie di istruzioni passo-passo) che includono sequenze e loop semplici per eseguire un compito, sia in modo indipendente che collaborativo, con o senza un dispositivo di calcolo.	Algoritmi e programmi	Riconoscere e definire problemi computazionali
K-2	1A-A-6-8	Analizzare e correggere (correggere) un algoritmo che include sequenze e loop semplici, con o senza un dispositivo di calcolo.	Algoritmi e programmi	Test e raffinazione
3-5	1B-A-5-4	Costruisci programmi, per risolvere un problema o per l'espressione creativa, che includono sequenze, eventi, loop, condizionali, parallelismo e variabili, usando un linguaggio di programmazione visuale basato su blocchi o un linguaggio basato sul testo, sia in modo indipendente che collaborativo (es. coppia di programmazione).	Algoritmi e programmi	Creazione di artefatti computazionali
3-5	1B-A-3-7	Costruisci ed esegui un algoritmo (set di istruzioni passo passo) che include sequenze, loop e condizionali per eseguire un compito, sia in modo indipendente che collaborativo, con o senza un dispositivo di calcolo.	Algoritmi e programmi	Riconoscere e definire problemi computazionali
3-5	1B-A-6-8	Analizza ed esegui il debug (correzione) di un algoritmo che include sequenze, eventi, loop, condizionali, parallelismo e variabili.	Algoritmi e programmi	Test e raffinazione
6-8	2-A-5-6	Sviluppa programmi, sia in modo indipendente che collaborativo, che includono sequenze con cicli annidati e rami multipli. [Chiarimento: a questo livello, gli studenti possono utilizzare linguaggi di programmazione basati su blocchi e / o testo.]	Algoritmi e programmazioni	Creazione di artefatti computazionali

