Girls Who Code At Home

Attività coraggiosa, ma non perfetta
Debug su Scratch

Panoramica dell’attività

In questa attività approfondiremo maggiormente ciò che significa **attività coraggiosa, ma non perfetta** quando si parla di coding! Se un programma non funziona come ci aspettavamo, ciò non significa che si tratti di una sconfitta. Sono molte le competenze che possiamo apprendere e implementare per risolvere gli errori e per migliorare i nostri prodotti o i nostri progetti.

Il **Debug** è una strategia che gli informatici utilizzano per cercare e per risolvere i problemi o i **bug** nel loro programma. Durante questa attività ti aiuteremo ad applicare soluzioni di debug o a risolvere tre programmi su Scratch! Prima di approfondire questa attività, ti consigliamo di dare un’occhiata alla Woman in Tech del momento, Ayanna Howard. Il primo progetto di Ayanna con la NASA è stato quello di creare un robot che imparasse a conoscere l’ambiente su Marte. Scopri di più su come Ayanna si serva del suo coraggio e della sua sicurezza per realizzare i suoi progetti innovativi.

Materiali

- Scratch online o Scratch offline
- Sfida di debug 1
- Sfida di debug 2
- Sfida di debug 3
- Facoltativo: Volantino Sfida di debug
- Soluzioni volantino Sfida di debug

Women in Tech Spotlight: Ayanna Howard

Fonte dell'immagine: Black Sci-Fi

I personaggi esemplari da cui ispirarsi possono avere qualsiasi forma e dimensione. Infatti, il primo personaggio esemplare al quale si è ispirata la Dott.ssa Ayanna Howard, non era nemmeno un essere umano! La Donna bionica, una super eroina robotica, è stata fonte di ispirazione per la Dott.ssa Howard non solo per la costruzione di robot, ma per coltivare la sua passione per l'ingegneria, sin dalla sua più giovane età. Ha proseguito gli studi di elettrotecnica e informatica all'università e alla scuola di specializzazione e infine, si è laureata in economia.

In linea con il suo amore per l'apprendimento, la robotica e l'elettrotecnica, la Dott.ssa Howard è attualmente docente di bioingegneria e co-fondatrice e Chief Technology Officer della società di robotica educativa, Zyrobotics. Inoltre, ha sviluppato, insieme alla NASA, dei robot che stanno imparando a vivere su Marte.

Guarda questo video per saperne di più sul primo incarico della Dott.ssa Howard alla NASA. Inoltre, nel video la Dott.ssa Howard spiega il perché dell'importanza della diversità sul posto di lavoro e come sia stata in grado di reagire di fronte a situazioni spiacevoli sul lavoro.

Riflessione

Essere un informatico significa molto di più che essere bravo a fare coding. Prenditi del tempo per riflettere su come Ayanna e il suo lavoro hanno a che fare con i cavalli di battaglia che i grandi informatici puntano a costruire: coraggio, resilienza, creatività e motivazione.

 **CORAGGIO**	La Dott.ssa Howard ha mostrato coraggio quando si è ritrovata a dover parlare in occasione di una situazione spiacevole al lavoro. Che cosa diresti a te stessa in quel momento?

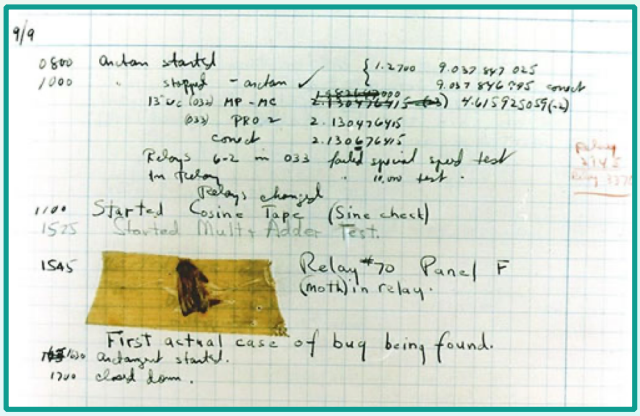
Condividi le tue risposte con un membro della tua famiglia, con un'amica o con un amico. Incoraggia gli altri a ottenere maggiori informazioni su Ayanna per partecipare alla discussione!

Fase 1: in che cosa consiste il Debug? (5 min)

Pensa a un momento in cui hai cercato di risolvere un problema: per esempio, quando ti sei ritrovata a dover risolvere un problema difficile per la scuola o quando hai cercato di ritrovare un oggetto che avevi perduto. Sei stata in grado di risolvere il problema nell'immediato? Molto spesso, ci ritroviamo a risolvere un problema, ma a volte ci accade che prima di riuscirci dobbiamo affrontare una serie di sconfitte. I programmatori passano molti giorni a cercare di trovare e risolvere i problemi del loro codice! L'insuccesso è parte integrante del mondo dell'informatica 😄.

Un errore in un programma o in un hardware viene chiamato **bug.** Il processo di identificazione e di rimozione degli errori o dei bug dall'hardware o dal software di un computer viene chiamato **debug**. Le origini di queste parole risalgono a Grace Hopper, una delle primissime pioniere nel campo dell'informatica. Mentre stava lavorando su uno dei primi computer, Grace Hopper e il suo team trovarono all'interno del computer una falena che stava creando un errore, si trattava di un vero e proprio insetto (bug in inglese). Il termine utilizzato da Grace per identificare la falena incollata con del nastro adesivo sul suo diario è stato registrato come il primo bug informatico della storia. Oggi, è possibile ammirare il diario di Grace presso il Smithsonian Museum of American History a Washington D.C.

Il primo bug informatico della storia



Fonte dell'immagine: Atlas Obscura

Prenditi qualche minuto per pensare a cosa significhi per te la frase **Attività coraggiosa, ma non perfetta**. Perché è importante essere **coraggiosi** quando si prova a fare qualcosa di nuovo, anziché cercare di essere **perfetti**? Molto spesso, ci ritroveremo a dover affrontare delle sfide quando stiamo cercando di imparare qualcosa di nuovo. È importante essere **resilienti** e cercare di guardare ai nostri errori e alle nostre sfide come a delle opportunità di apprendimento. L'attività di debug rappresenta un'opportunità per imparare dai nostri errori. Queste strategie spesso ti aiutano ad affrontare delle sfide più complicate, che potrebbero presentarsi in futuro.

Fase 2: accedi a Scratch ed esplora l'interfaccia (5-10 min)

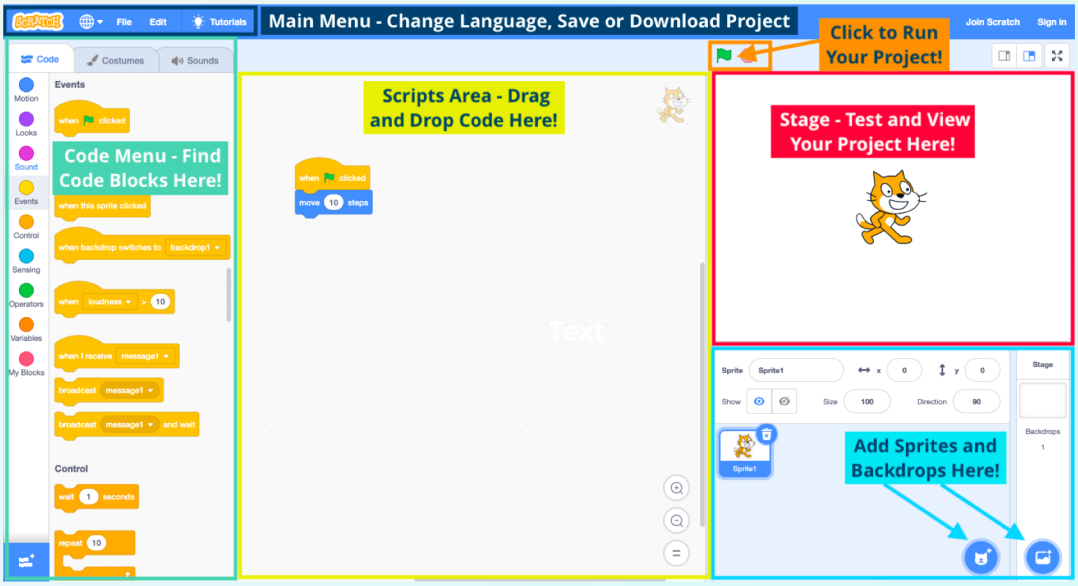
Scratch è una piattaforma di programmazione gratuita; utilizza un linguaggio di programmazione basato su blocchi ed è sviluppata da MIT, che permette di programmare storie interattive, giochi e animazioni.

1. **Registrati o Accedi a Scratch.**
   Per salvare il tuo lavoro sulla piattaforma online di Scratch, dovrai creare un account, se non ne possiedi già uno. Per creare il tuo account, segui le istruzioni sul modulo di iscrizione. Se hai meno di 13 anni avrai bisogno dell'indirizzo e-mail di un tuo genitore, per registrarti. Se non desideri creare un account puoi anche decidere di scaricare e utilizzare la versione offline di Scratch 3.0.

2. **Esplora l'interfaccia di Scratch.**
   Nel caso in cui stessi effettuando il tuo primo accesso a Scratch, prenditi qualche minuto per **Creare** 

   un nuovo progetto, così potrai esplorare l'interfaccia di Scratch. Puoi anche guardare questo tutorial su come Cominciare su Scratch!

   Nell'immagine qui sotto appaiono una serie di tasti, sulla piattaforma di Scratch, con i quali dovrai interagire.



Fase 3: esamina le strategie di Debug su Scratch (5 min)

Prima di cominciare ad applicare soluzioni di debug, prenditi qualche minuto per esaminare alcune strategie che potresti usare quando cercherai di trovare e risolvere un bug sul tuo programma.

1. **Descrivere l'errore (o bug).** Come ti sei resa conto della presenza di un bug nel programma? Pensa a quello che ti aspettavi che sarebbe successo e a quello che è successo realmente.
2. **Rileggi il tuo codice e pensa a quali possano essere gli errori.** Leggi tutte le stringhe del tuo codice, una dopo l'altra. Se preferisci, puoi anche leggerle a voce alta! Mentre starai rileggendo il tuo codice, pensa a quale/quali stringa/stringhe di codice (o blocchi di Scratch) possano essere all'origine dell'errore o possano avere portato a dei risultati inaspettati.
3. **Evidenzia alcuni marcatori di codici nel tuo codice in modo che possano essere utilizzati come punti di riferimento durante la tua verifica.** Nel caso in cui il tuo codice fosse troppo lungo, suddividerlo in più sezioni potrebbe essere una buona idea. Su Scratch puoi utilizzare un *blocco **dire***. Questo ti permetterà di sapere esattamente dove si trova il tuo programma rispetto alle stringhe del codice. Questa strategia ti può aiutare a restringere le possibili posizioni del bug.

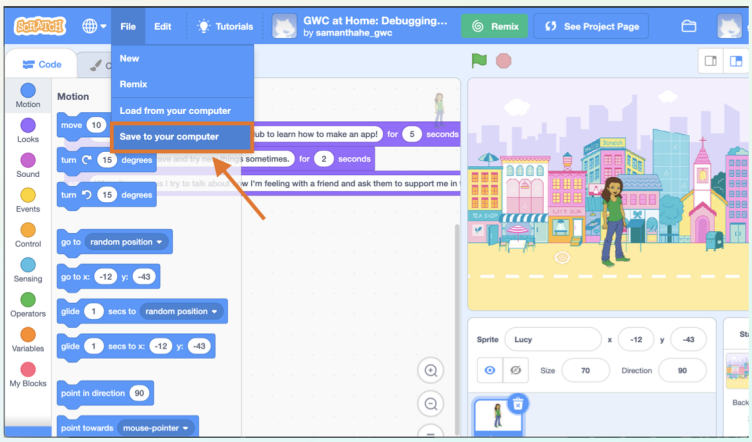
   

   Per esempio, se sai che quando premi sulla prima parte di codice evidenziata (ossia quando lo sprite ripete il testo nel *blocco dire* che stai utilizzando come codice evidenziato) e che il programma funziona correttamente, ciò significa che il bug, molto probabilmente, non esiste prima della prima parte evidenziata. Se noti il bug nel tuo programma, prima di premere sulla seconda parte di codice evidenziata, allora saprai che il tuo bug molto probabilmente si trova nel codice che sta tra la prima e la seconda parte evidenziata.
4. **Quando avrai trovato il bug, cerca di capire il motivo per cui il codice è un bug.** Pensa al motivo per cui la stringa di codice o il blocco di codice possa causare un errore sul tuo programma. Alcuni errori comuni potrebbero essere dovuti a errori di spelling, di posizionamento dei blocchi di codice in un ordine errato o perché stai utilizzando un blocco sbagliato. Alcuni editor di codice includono messaggi di errore che descrivono il motivo per cui una particolare stringa di codice sia un bug. Scratch non prevede questo genere di messaggi di errore, quindi dovrai rifletterci un po' su!
5. **Risolvere e testare il tuo codice.** Prova a risolvere la tua stringa di codice. In seguito, prova a eseguire nuovamente il tuo programma. Il bug è ancora esistente? In questo caso, prova un'altra soluzione e testalo di nuovo! Il bug è presente in un'altra area? È possibile che tu abbia avuto un secondo bug nel programma. Dovrai quindi rivedere passo-passo i vari passaggi e cercare dove si trova il nuovo bug. Nel caso in cui non ci dovessero essere dei bug nel tuo programma, ciò significa che il hai risolto il tuo bug!

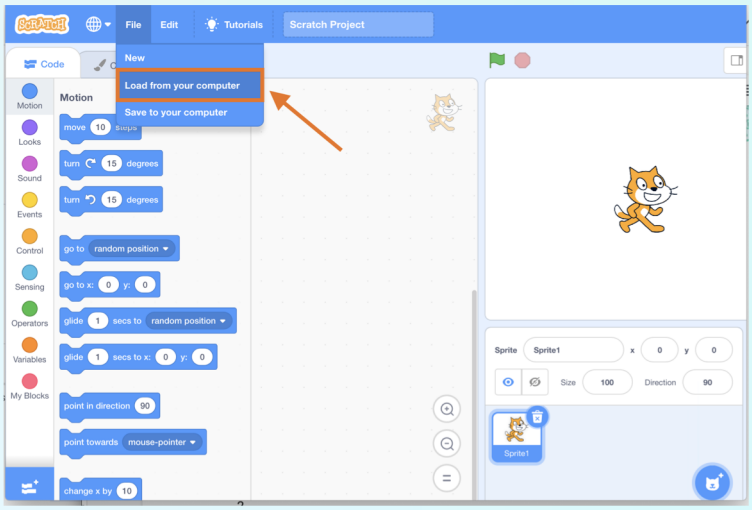
Fase 4: affronta la Sfida di debug 1 (5-10 min)

Conoscenze pregresse: *per potere risolvere questa sfida dovresti avere già imparato il concetto di blocchi di Evento e Look. Prova a dare un'occhiata a questo tutorial Cominciare, per imparare di più su questi blocchi.*

1. **Remixare il codice di avvio.** Fai clic sul pulsante Remix in alto a destra per fare una copia del progetto.
    - Se stai utilizzando la versione di Scratch Editor offline, allora dovrai salvare una copia del progetto sul tuo computer. Fai clic sul pulsante See inside per vedere il codice del progetto.
    - Fai clic su **File**, sulla barra di navigazione, e scegli l'opzione **Salvare sul mio computer** nel menu a tendina.



    - Apri l'Editor Scracth Offline sul tuo computer. Fai clic su **File** sulla barra di navigazione e scegli l'opzione **Caricare sul mio computer**, nel menu a tendina. Trova il file del progetto salvato sul tuo computer e fai clic su **Apri.**



Fase 4: affronta la Sfida di debug 1 (Continua)

2. **Testa il programma e identifica il bug.** In questo progetto, Lucy, il nostro sprite (od oggetto) principale si presenta all'utente. Tuttavia, quando la bandiera verde è premuta non succede nulla! Esegui il programma premendo la bandiera verde  . Esamina il codice e verifica se puoi identificare il bug. Utilizza il **volantino sfida**, alle pagine 12-13, per aiutarti a risolvere e per riflettere sul bug.

3. **Risolvi il bug e verifica la soluzione qui.** Dai un'occhiata alla pagina 14 per vedere più nel dettaglio quello che è stato detto su questo bug!

4. **Rifletti sul processo di debug che hai messo in pratica.** Pensa alla natura del bug e al modo in cui ha influenzato il programma. Quando si impara a fare coding, molti programmatori tengono vicino un foglietto dove sono riportati tutti gli errori più comuni. Questo foglietto gli permette di imparare dai loro errori quando in futuro dovranno lavorare su nuovi progetti.

Fase 5: affronta la Sfida del debug 2 (5-10 min)

Conoscenze pregresse: *Per poter risolvere questa sfida dovresti avere già imparato i concetti di blocchi di Evento, Look, Movimento, Rilevamento e Loop. Prova a dare un'occhiata a questo tutorial Codificare un cartone animato per imparare di più su questi blocchi.*

1. **Remixare il codice di avvio.** Fai clic sul pulsante Remix in alto a destra, per fare una copia del progetto.
   - Se stai utilizzando la versione di Scratch Editor offline, allora dovrai salvare una copia del progetto sul tuo computer. Fai clic sul pulsante See inside per vedere il codice del progetto.
   - Fai clic su **File** sulla barra di navigazione e scegli l'opzione **Salvare sul mio computer** nel menu a tendina.
   - Apri l'Editor Scracth Offline sul tuo computer. Fai clic su **File** sulla barra di navigazione e scegli l'opzione **Caricare sul mio computer** nel menu a tendina. Trova il file del progetto salvato sul tuo computer e fai clic su **Apri.**

2. **Testa il programma e identifica il bug.** In questo programma Avery dovrebbe cominciare nel centro della pagina e dovrebbe presentarsi. Poi, dovrebbe camminare verso destra fino a quando non raggiunge il bordo che le dà accesso al club di coding. La prima volta che premiamo sulla bandiera verde funziona ma quando premiamo nuovamente sulla bandiera verde, Avery è ancora sulla destra anziché cominciare al centro dello schermo. Esegui il programma premendo la bandiera verde. Esamina il codice e verifica se puoi identificare il bug. Utilizza il **volantino sfida**, alle pagine 12-13, per aiutarti a risolvere e per riflettere sul bug.

3. **Risolvi il bug e verifica la soluzione qui.** Attenzione, è possibile che tu abbia codificato Avery in modo tale che cominci in una posizione diversa rispetto alle posizioni che sono state elencate nelle soluzioni. Dai un'occhiata alla pagina 15 per vedere più nel dettaglio quello che è stato detto su questo bug!

4. **Rifletti sul processo di debug che hai messo in pratica.** Pensa alla natura del bug e al modo in cui ha influenzato il programma. Quando si impara a fare coding, molti programmatori tengono vicino un foglietto dove sono riportati tutti gli errori più comuni. Questo foglietto gli permette di imparare dai loro errori quando in futuro dovranno lavorare su nuovi progetti.

Fase 6: affronta la Sfida di debug 3 (5-15 min)

Conoscenze pregresse: *Per potere risolvere questa sfida dovresti avere già imparato i concetti di blocchi di Evento, Look, Suoni, Aspetta e Loop. Prova a dare un'occhiata a questo tutorial Codificare un cartone animato per imparare di più su questi blocchi.*

1. **Remixare il codice di avvio.** Fai clic sul pulsante Remix in alto a destra per fare una copia del progetto.
   - Se stai utilizzando la versione di Scratch Editor offline, allora dovrai salvare una copia del progetto sul tuo computer. Fai clic sul pulsante See inside per vedere il codice del progetto.
   - Fai clic su **File** sulla barra di navigazione e scegli l'opzione **Salvare sul mio computer** nel menu a tendina.
   - Apri l'Editor Scratch Offline sul tuo computer. Fai clic su **File** sulla barra di navigazione e scegli l'opzione **Caricare sul mio computer** nel menu a tendina. Trova il file del progetto salvato sul tuo computer e fai clic su **Apri.**

2. **Testa il programma e identifica il bug.** Quando premi sulla bandiera verde , Ada dovrebbe parlarti di lei. Poi, dovrebbe ballare al ritmo di musica. Quando Ada comincia a ballare, la musica non funziona come dovrebbe! La musica ricomincia in loop ogni volta che Ada si muove. Come possiamo risolvere questo codice, in modo che la musica suoni fino alla fine, mentre Ada balla? Esegui il programma premendo la bandiera verde. Esamina il codice e verifica se puoi identificare il bug. Utilizza il **volantino sfida**, alle pagine 12-13, per aiutarti a risolvere e per riflettere sul bug.

3. **Risolvi il bug e verifica la soluzione qui.** Attenzione, è possibile che tu abbia codificato Ada in modo tale che cominci in una posizione diversa rispetto alle posizioni che sono state elencate nelle soluzioni. Dai un'occhiata alla pagina 16 per vedere più nel dettaglio quello che è stato detto su questo bug!

4. **Rifletti sul processo di debug che hai messo in pratica.** Pensa alla natura del bug e al modo in cui ha influenzato il programma. Quando si impara a fare coding, molti programmatori tengono vicino un foglietto dove sono riportati tutti gli errori più comuni. Questo foglietto gli permette di imparare dai loro errori quando in futuro dovranno lavorare su nuovi progetti.

Fase 7: condividi la tua creazione (5 min)

1. **Condividi il tuo progetto su Scratch.**
   Quando avrai risolto una sfida, premi sul pulsante Condividi su Scratch. Aggiungi sulla sezione Istruzioni, come hai risolto il bug!

2. **Condividi come stai affrontando le sfide con Girls Who Code at Home!**
   Non dimenticarti di condividere i tuoi progetti sui social media. Tagga @girlswhocode e utilizza l'hashtag #codefromhome. Potremmo anche presentarti sul nostro account!

Volantino Sfida di debug

Istruzioni: mentre affronti ciascuna sfida, pensa alle seguenti domande.

- In che cosa consiste il bug? Che cosa sarebbe dovuto accadere? Ci sono diversi bug?
- In che modo il bug ha avuto un impatto sul programma e perché?
- In che modo hai risolto il bug?

Sfida di debug 1 (5-10 min)

Codice di avvio: https://scratch.mit.edu/projects/387548698/

Bug:

In che modo il bug ha avuto un impatto sul programma e perché?

In che modo hai risolto il/i bug?

Volantino Sfida di debug

Sfida di debug 2 (5-10 min)

Codice di avvio: https://scratch.mit.edu/projects/387549618/

Bug:

In che modo il bug ha avuto un impatto sul programma e perché?

In che modo hai risolto il/i bug?

Sfida di debug 3 (5-15 min)

Codice di avvio: https://scratch.mit.edu/projects/387851932/

Bug:

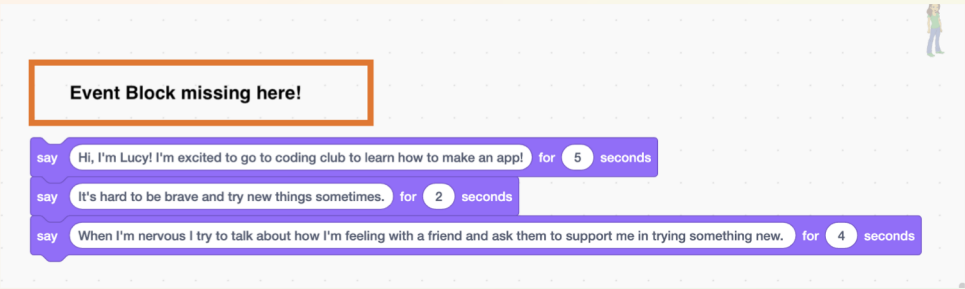
In che modo il bug ha avuto un impatto sul programma e perché?

In che modo hai risolto il/i bug?

Sfida di debug 1 Soluzione

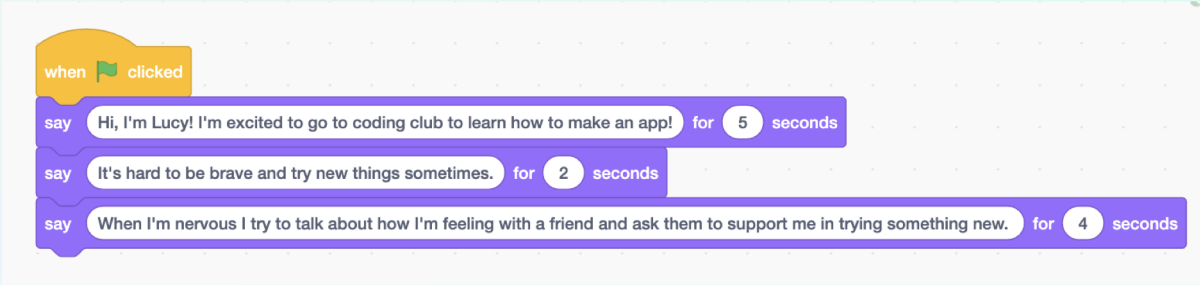
Codice di avvio con bug: https://scratch.mit.edu/projects/387548698/
Esempio di codice di soluzione: https://scratch.mit.edu/projects/388218006/

Bug: Quando si clicca sulla bandiera verde Lucy dovrebbe presentarsi. Invece, non succede niente! Questo perché non c'era nessun codice blocco nel codice.



In che modo il bug ha avuto un impatto sul programma e perché? Senza un *blocco evento*, il computer non sa quando deve eseguire il codice che abbiamo scritto. Quindi, quando la bandiera verde è stata premuta, non è successo niente perché non abbiamo detto al computer di fare qualcosa.

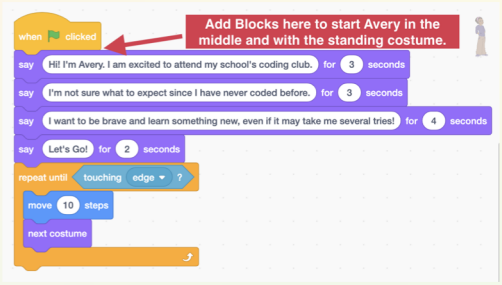
In che modo hai risolto il/i bug? Dobbiamo dire al computer che **Quando si clicca su bandiera verde** che deve svolgere il codice che abbiamo scritto. Abbiamo usato il *blocco* when clicked e l'abbiamo posizionato di fronte ad altri codici di blocco.



Sfida di debug 2 Soluzione

Codice di avvio con bug: https://scratch.mit.edu/projects/387549618/
Esempio di soluzione codice: https://scratch.mit.edu/projects/388333942/

Bug: questo programma contiene **due** bug. Dopo aver cliccato sulla bandiera verde per la seconda volta, Avery non comincerà più dallo stesso punto ma comincerà dal centro dello schermo. In realtà, comincia a spostarsi da dove l'avevamo lasciata l'ultima volta, sulla destra dello schermo con un costume da passeggiata. Il primo bug corrisponde alla sua posizione, mentre il secondo corrisponde al suo costume (o look).

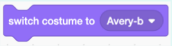


In che modo il bug ha avuto un impatto sul programma e perché? In questo programma Avery dovrebbe cominciare nel centro della pagina e dovrebbe presentarsi. Poi, dovrebbe camminare verso destra fino a quando non raggiunge il bordo che le dà accesso al club di coding. Poiché non abbiamo codificato in modo che Avery cominci sempre nella stessa posizione, ogni volta che la bandiera verde viene premuta, Avery inizia la sua sequenza di codice dove si era fermata l'ultima volta. Nel nostro caso, sul bordo dello schermo a destra.

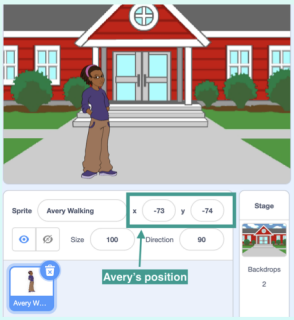
In che modo hai risolto il/i bug? Per far sì che Avery cominci sempre dalla stessa posizione, dobbiamo usare 

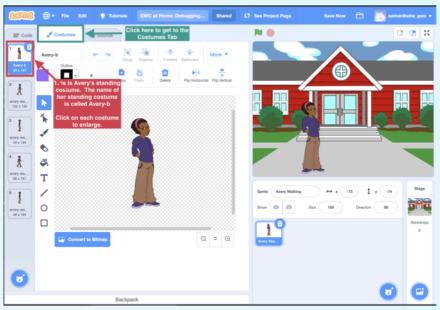
. Il modo più semplice per selezionare la posizione di partenza di Avery è quello di utilizzare il tuo mouse e di trascinare Avery dove vuoi che cominci. Vedrai che i numeri che si trovano dopo x: e y: cambieranno a seconda della posizione di Avery. Dovresti vederlo anche dalla descrizione di sprite, che si trova sotto lo stage di Scratch. Utilizziamo il *Vai al blocco* come il primissimo blocco dopo il nostro *blocco Evento* 

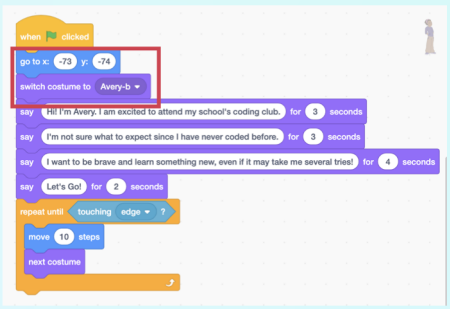
.

Poi, vogliamo che Avery cominci con il costume che la mostra in piedi. Per questo dobbiamo usare 

esattamente dopo che abbiamo impostato la sua posizione di partenza, per assicurarci che cominci sempre con il costume che la mostra in piedi. Se non sei sicura a che cosa corrisponda il suo costume che la mostra in piedi, guarda la scheda **Costumi** e registra il nome del costume con il quale desideri che cominci. Codifichiamo questi due blocchi di codici in primis poiché vogliamo che il nostro sprite parta sempre dalla stessa posizione, esattamente quando la bandiera verde è premuta, prima che Avery cominci a parlare e a camminare!



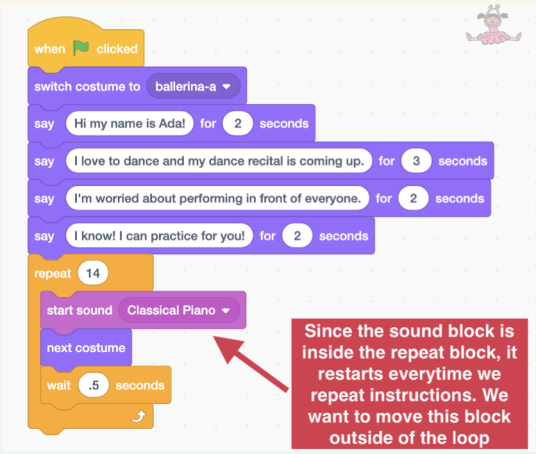


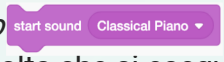


Sfida di debug 3 Soluzione

Codice di avvio con bug: https://scratch.mit.edu/projects/387851932/
Esempio di soluzione codice: https://scratch.mit.edu/projects/388346112/

Bug: quando premi sulla bandiera verde  , Ada dovrebbe presentarsi. Poi, dovrebbe ballare al ritmo di musica. Quando Ada comincia a ballare, la musica non funziona come dovrebbe! La musica ricomincia in loop ogni volta che Ada si muove. Il bug ha a che vedere con la sequenza o con l'ordine con cui sono stati codificati i blocchi.



In che modo il bug ha avuto un impatto sul programma e perché? Attenzione, il nostro *blocco del suono* 

è stato codificato all'interno del loop di ripetizione. Questo significa che ogni volta che si esegue il loop del codice all'interno, il Pianoforte Classico si riavvia di nuovo!

In che modo hai risolto il/i bug? Dobbiamo estrarre il nostro *blocco del suono* al di fuori del loop, ma dove possiamo estrarlo? Dobbiamo ripensare a quello che vogliamo ottenere. Cerca di fare delle prove e di spostare il *blocco del suono* prima e dopo il *blocco di ripetizione*. Se spostiamo il *blocco del suono* dopo il loop di ripetizione, la musica partirà dopo che Ada avrà ballato, e questo non è quello che vogliamo. Se spostiamo il *blocco del suono* prima del loop di ripetizione, la musica suonerà in primis e poi Ada comincerà a ballare. Questo è quello che vogliamo! Puoi decidere se vuoi che il suono inizi quando balla o se invece preferisci che inizi quando si presenta.