

REALIZZARE UN LABIRINTO

In questo progetto impareremo a

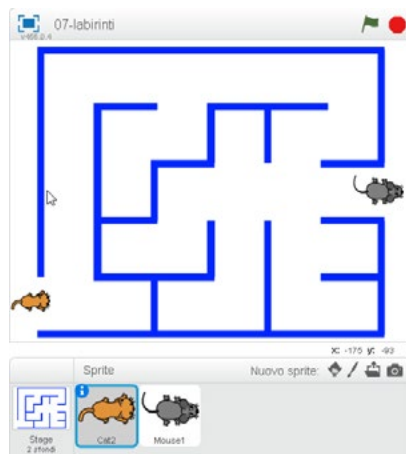
- muovere uno sprite con la tastiera
- utilizzare la condizione logica (se... allora)
- inserire il codice ad oggetti diversi
- utilizzare i messaggi per permettere agli sprite di comunicare tra loro

Anche in questo progetto faremo uso di procedure.

Il materiale è presente nella cartella *07-Labirinti*.

Avviare un nuovo progetto. Inserire come sfondo l'immagine di un labirinto, disegnandolo o caricandone uno dalla cartella del materiale.

Inserire due sprite dalla libreria di *Scratch*: *cat2* e *mouse1* (il gattino e il topolino visti dall'alto). Posizionare il gattino all'ingresso del labirinto e il topolino all'uscita.



MUOVERE UNO SPRITE CON LA TASTIERA

Per muovere il gattino utilizzeremo le frecce cursore. Le frecce destra e sinistra muoveranno il gattino sull'asse orizzontale (asse X), quelle verso l'alto e il basso sull'asse verticale (asse Y).

Selezionare il gattino ed inserire il codice. Dovrebbe essere ormai semplice reperire i blocchi controllando il loro colore.



Provare a testare il codice. Il codice risulta funzionante, ma ogni volta che si vuole ricominciare il gattino va posizionato manualmente in posizione iniziale.

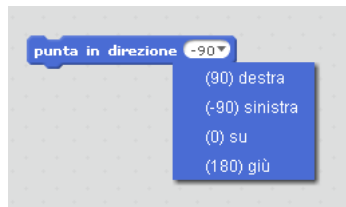
Inseriamo il codice che lo riporti in posizione iniziale alla pressione della bandierina:



Inserire la seconda istruzione di movimento (la freccia sinistra). Abbiamo già sperimentato nel capitolo precedente l'istruzione **CAMBIA x DI ...** e la funzione dei numeri positivi e negativi.



Testare il programma. Il gattino si muove, ma “va a marcia indietro”! È necessario che il gattino guardi nella direzione verso la quale sta camminando. Per questo utilizzeremo l'istruzione **PUNTA IN DIREZIONE ...**



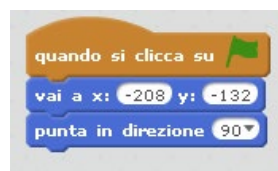
Il codice di spostamento inserito diventa quindi:



Con la stessa procedura inserire anche il codice per lo spostamento verso l'alto e verso il basso.



Il blocco va inserito anche nel codice che riporta in posizione iniziale il gattino (altrimenti potrebbe trovarsi a guardare nella direzione errata).



LA CONDIZIONE LOGICA

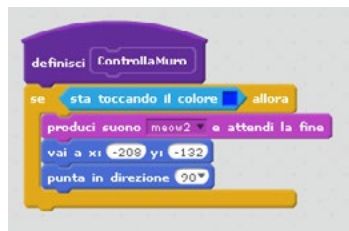
Il gattino si muove ora correttamente, torna in posizione iniziale, ma non rispetta le regole di un labirinto: attraversa i muri senza rilevarli. Ogni volta che il gattino si muove deve controllare di non toccare un muro: questo controllo va fatto per ogni spostamento, qualsiasi sia la freccia che si preme. Il codice andrebbe quindi scritto quattro volte. Per ovviare a questo utilizzeremo una procedura (creeremo un nuovo blocco). Questo rende più rapida la scrittura del codice ma, soprattutto, permette di modificarlo con più rapidità (se serve modificare qualcosa si interviene in un unico blocco di codice, non in quattro).

Definire un blocco (che nell'esempio ho chiamato ControllaMuro) ed inserire il codice **SE STA TOCCANDO IL COLORE ALLORA...**



Le istruzioni che inseriremo nel blocco verranno eseguite se il gattino tocca il colore delle pareti del labirinto (per impostare il colore cliccare sulla casella colorata dell'istruzione e poi su una parete del labirinto).

Se il gattino tocca le pareti verrà riprodotto un file audio e il gattino tornerà alla posizione iniziale. Il file audio *meow2* è già presente nella scheda *Suoni* del gattino.



La procedura va richiamata ogni volta che il gattino si muove. Il codice diventa quindi:



Testare il programma.

GESTIRE I MESSAGGI

Inseriremo ora una finestra di messaggio con i complimenti, da visualizzare quando viene completato il labirinto.

Creare un nuovo sfondo caricando dalla libreria lo sfondo *party* e scrivendo all'interno "Compli-

menti”.



Questa videata andrà mostrata quando il labirinto è terminato. Verrà anche riprodotto un file sonoro. Aprite la scheda **Suoni** del gattino e caricate dalla libreria il suono *Triumph*.

Definire un nuovo blocco (nel codice del gattino) e inserire il codice:



Richiamare questa procedura ad ogni movimento del gattino.



In questo modo a labirinto completato verrà riprodotto il suono e verrà nascosto il gattino. È necessario nascondere anche il topolino e cambiare lo sfondo. Il gattino non controlla però gli altri oggetti: deve inviare un **MESSAGGIO** che venga raccolto dagli oggetti, in modo che questi possano comportarsi di conseguenza.

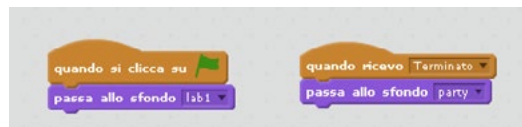
Aprire la scheda **SITUAZIONI** e aggiungere un blocco **INVI A TUTTI MESSAGGIO**. Creare un nuovo messaggio. È possibile scegliere liberamente il nome. Io lo chiamo **TERMINATO**



Il codice diventa quindi



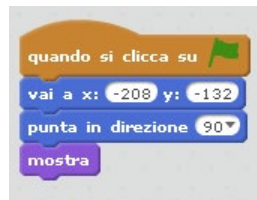
Quando lo sfondo riceverà il messaggio, passerà allo sfondo *party*. Va inserito anche un codice che riporti allo sfondo con il labirinto, altrimenti non sarà possibile ricominciare il gioco. Allo sfondo inseriremo quindi il codice



Al topolino inseriremo il codice che lo fa nascondere quando è terminato il labirinto e lo visualizza nuovamente quando si ricomincia.



Va aggiunta anche l'istruzione che renda nuovamente visibile il gattino quando si riavvia il gioco (altrimenti rimarrà invisibile).



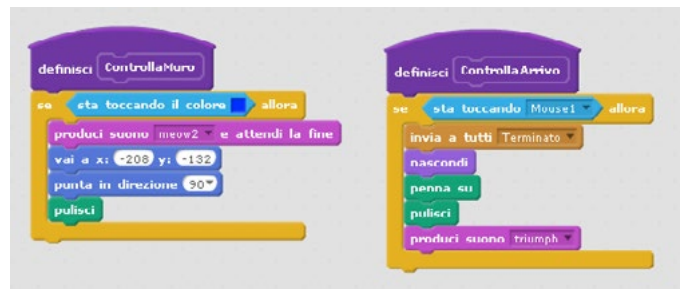
DISEGNARE LA TRACCIA DEL PERCORSO

Abbiamo già sperimentato l'utilizzo della penna nel progetto precedente. L'utilizzo nel labirinto non presenta differenze. Va però prestata attenzione a quando la pagina va "pulita" dalla traccia: quando il gattino tocca il muro e torna in posizione iniziale, quando viene mostrata la schermata finale e quando si vuole iniziare nuovamente il gioco.

All'inizio del gioco si pulisce la videata e si imposta la penna:



L'istruzione **PULISCI** verrà richiamata anche in altre situazioni



Nella procedura **CONTROLLAARRIVO**, se il labirinto è stato completato, dopo aver nascosto il gattino è necessario alzare la penna: il gattino, anche se nascosto, potrebbe infatti continuare a scrivere nella videata con i complimenti se vengono premuti i tasti freccia.

IN AULA

Provare a “giocare” con il codice spostando il blocco **PULISCI** nei vari frammenti di codice. Cosa succede?

Provare a realizzare un programma simile, senza utilizzare la guida passo-passo, nel quale si deve muovere un'automobilina senza uscire di strada (nella cartella del materiale sono presenti sia gli sfondi che l'immagine dell'automobilina).