

CLASSI DERIVATE DALLA CLASSE MOVIECLIP

LA CLASSE MOVIECLIP

- La classe MovieClip è l'oggetto software che ci consente di gestire gli elementi visuali in Flash da ActionScript.
- Attraverso la classe Movie Clip posso gestire:
 - MovieClip creati runtime utilizzando ActionScript.
 - Simboli di tipo MovieClip memorizzati in libreria e trascinati durante lo sviluppo sullo stage.
 - Simboli di tipo MovieClip memorizzati in libreria e caricati sullo stage in fase runtime.
 - Filmati memorizzati su disco e caricati runtime all'interno del filmato principale.
 - Immagini bitmap (formato jpeg) memorizzate su disco.

LA CLASSE MOVIECLIP

- La classe MovieClip normalmente NON si crea utilizzando il comando **new**.
- Quando trascino un Simbolo di tipo MovieClip memorizzato in libreria sullo stage durante lo sviluppo automaticamente creo un oggetto MovieClip che è istanza del simbolo in libreria.
- Posso creare MovieClip runtime all'interno di altre MovieClip applicando il metodo **CreateEmptyMovieClip()**.
- Posso inserire istanze di simboli di tipo MovieClip memorizzati in libreria in altre MovieClip applicando il metodo **AttachMovieClip()**.
- Posso caricare filmati o immagini bitmap memorizzati su disco sostituendo il contenuto di MovieClip esistenti o creandone di nuove.

MOVIECLIP CREATA DESIGN TIME

- Quando trascino un simbolo MovieClip sullo stage creo una nuova istanza di quel simbolo.
- Se a quella istanza design time do un nome, attraverso quel nome posso modificare le proprietà di quella istanza e applicare ad essa i metodi di MovieClip.
- Intermini di Programmazione ad Oggetti quando creo un simbolo MovieClip creo un modello di filmato che condivide con la classe MovieClip tutte le proprietà e tutti i metodi, e che aggiunge alla MovieClip astratta i contenuti visuali presenti.
- L'istanza creata appartiene alla MovieClip (film principale o secondario) entro alla quale è stata inserita. Programmaticamente la si può raggiungere utilizzando indifferentemente la sintassi degli Oggetti **contenitore.nomeistanza** che la sintassi degli array associativi **contenitore["nomeistanza"]**

MODIFICA DELLA POSIZIONE E DELL'ASPETTO DI UN CLIP FILMATO

- Per modificare le proprietà di un clip filmato durante l'esecuzione, creare un'istruzione che assegna un valore a una proprietà.
- Le proprietà `_x`, `_y`, `_rotation`, `_xscale`, `_yscale`, `_height`, `_width`, `_alpha` e `_visible` determinano l'aspetto geometrico del clip filmato di tutti gli elementi in esso contenuto.
- Le seguenti proprietà sono a sola lettura: `_currentframe`, `_droptarget`, `_framesloaded`, `_parent`, `_target`, `_totalframes`, `_url`, `_xmouse` e `_ymouse`.

CREIAMO UNA CLASSE

FASI DELLA CREAZIONE DI UNA CLASSE

- Le fasi di realizzazione di una classe possono essere così riassunte:
 - **Progettazione**: comprende il lavoro da svolgere prima della scrittura del codice. Si tratta di definire il funzionamento del nuovo oggetto che la classe mette disposizione, e quindi stabilire quali proprietà e metodi saranno necessari.
 - **Scrittura del codice**. Si passerà quindi alla fase di scrittura del codice. Dovremo realizzare un file di classe secondo le specifiche che abbiamo visto nel capitolo precedente, preoccupandoci soprattutto della corretta sintassi di quanto scriviamo.
 - **Test e correzione degli errori**. Dovremo poi creare un file FLA di prova in cui testare il corretto funzionamento della classe e correggere gli eventuali errori
 - **Inserimento in una o più applicazioni**. Una volta testato il funzionamento della classe passeremo al suo utilizzo in alcune applicazioni flash.

PROGETTAZIONE

- Le fasi di progettazione è probabilmente la più importante riduce possibilità di errore, necessità di aggiustamenti. In altri termini ci fa risparmiare tempo.
- Il fase progettuale dobbiamo:
 - Definire l'obiettivo del nuovo tipo di oggetto che andiamo a realizzare
 - Definire le funzionalità che ci permettono di conseguire il nostro obiettivo
 - Definire le proprietà che ci servono per modificare il comportamento dell'oggetto
 - Definire i metodi che consentiranno al nostro oggetto di svolgere le suo funzioni

DEFINIZIONE DELL'OBIETTIVO

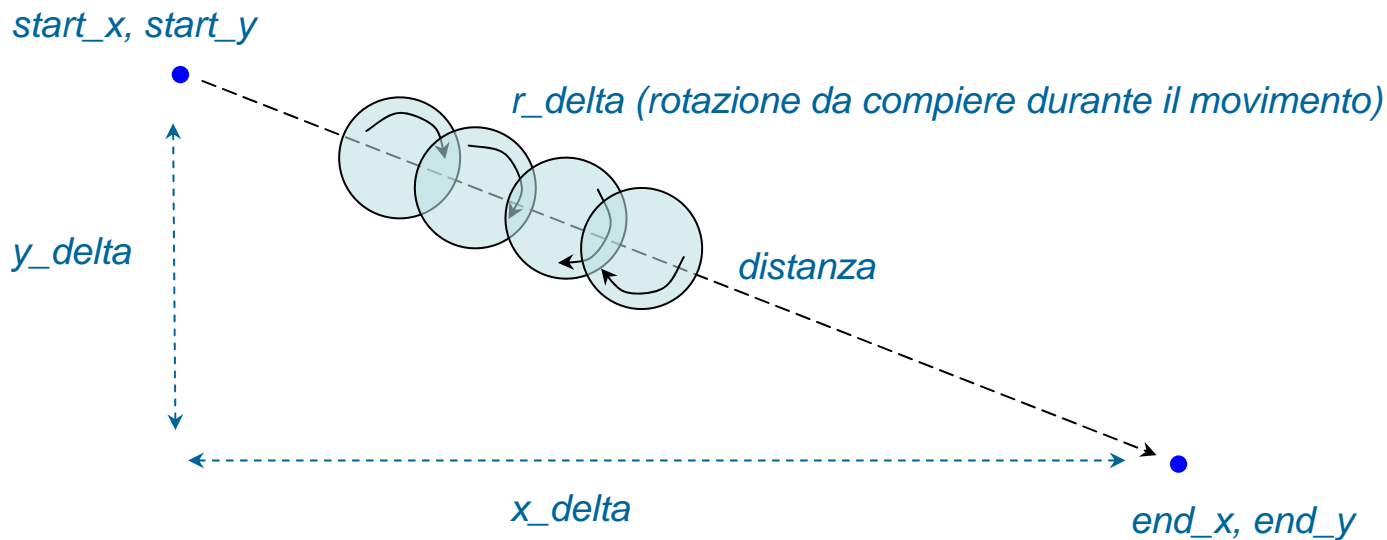
- Il nostro obiettivo è aggiungere a una MovieClip una funzionalità che consenta di spostarla (animazione) da un punto all'altro dello schermo secondo modalità determinate programmaticamente.
- Chiamerò la classe AnimClip

FUNZIONALITÀ

- Per la nostra AnimClip definiamo le funzionalità base:
 - Il movimento comincerà dalla posizione corrente. Dovrò poter definire in che **coordinate** terminerà l'animazione.
 - Dovrò poter stabilire la **velocità** del movimento.
 - Dovrò poter definire se la mia clip **rotolerà** o **scivolerà** verso la destinazione.

FUNZIONALITÀ

```
x_delta = end_x - start_x;  
y_delta = end_y - start_y;  
distanza = Math.sqrt((x_delta * x_delta) + (y_delta * y_delta))  
totalTime = distanza/speed * 1000  
r_delta = distanza/circonferenza * 360;
```



ALGORITMO

1. Memorizzo il punto di partenza (**start_x** e **start_y, start_rotation**)
2. Memorizzo il tempo di partenza
3. Calcolo il tempo totale
4. Creo un evento **onEnterFrame** che:
 1. Controlla quanto tempo è trascorso
 2. In proporzione al tempo trascorso modifica le proprietà **_x** e **_y** della MovieClip.
 3. In proporzione al tempo trascorso modifica la proprietà **_rotation**.
 4. Quando il tempo totale è trascorso interrompe l'evento **onEnterFrame**

PROPRIETÀ

- Le proprietà sono variabili che definisco all'interno della classe e che contengono le informazioni e i riferimenti agli oggetti utili al funzionamento della classe.
- Le proprietà possono essere pubbliche o private:
 - Le proprietà pubbliche sono quelle utilizzando le quali personalizzo l'istanza della classe
 - Le proprietà private sono quelle che uso esclusivamente nel codice scritto nella definizione di classe.
- Inoltre per ogni proprietà dovrò stabilire il tipo e definire un valore di default.

PROPRIETÀ PRIVATE

- Memorizzerò in alcune proprietà private le informazioni che devo consultare durante l'animazione.

Nome	Descrizione	Visibilità	Tipo	Default
start_x e start_y	Memorizzano le coordinate da cui è partito il movimento	private	Number	
start_rotation	Memorizza la rotazione della MovieClip al momento in cui è partito il movimento	private	Number	
x_delta e y_delta	Di quanto deve spostarsi la MovieClip sugli assi x e y per arrivare a destinazione.	private	Number	
r_delta	Di quanto deve ruotare la MovieClip durante il movimento.	private	Number	
startTime	Tempo segnato dal timer all'inizio del movimento	private	Number	

PROPRIETÀ PUBBLICHE

- Avrò poi due proprietà pubbliche che mi consentono di personalizzare il comportamento della MovieClip: il livello della velocità con cui la MovieClip si sposta e se la MovieClip rotola o scivola.

Nome	Descrizione	Visibilità	Tipo	Default
speed	Determina la velocità in pixel per secondo con cui si muove la MovieClip.	public	Number	100
rolling	Se il contenuto è vero rotola, altrimenti scivola.	public	Boolean	true

I METODI

- I metodi sono funzioni che definisco all'interno della classe e che svolgono le operazioni necessarie al funzionamento della classe.
- I metodi, come le proprietà, possono essere pubblici o privati:
 - I metodi pubblici sono quelli che utilizzo per far svolgere agli oggetti istanze della classe la loro funzione
 - I metodi privati sono quelli usati esclusivamente da altri metodi della classe.

I METODI

- In AnimClip definirò un solo metodo, che ho deciso di chiamare **animMoveTo** e che ha il compito di far partire il movimento della MovieClip.
 - **animMoveTo**(**end_x**:Number, **end_y**:Number)

I METODI: CONSTRUCTOR

- In **constructor** è sempre un metodo pubblico che ha lo stesso nome della classe.
- Nel caso di AnimClip non definirò il constructor e la classe lo erediterà dalla classe da cui deriva: MovieClip.
- La mia nuova classe AnimClip sarà pertanto per tutto identica ad un'altra MovieClip con la sola eccezione delle proprietà e del metodo che ho aggiunto.

TRASCINAMENTO DI CLIP FILMATO

- Applicando ad un Movie Clip il metodo **startDrag()** si rende il clip filmato trascinabile.
- In questo modo è possibile il trascinare oggetti in giochi, barre di scorrimento e cursori.
- Un clip rimane trascinabile fino a quando il trascinamento non viene interrotto esplicitamente mediante il metodo **stopDrag()** o fino a quando il trascinamento non viene applicato ad un altro clip filmato mediante **startDrag()**.
- È possibile trascinare in un solo clip filmato alla volta.

CREAZIONE DI CLIP FILMATO IN FASE DI RUNTIME

- Le istanze di clip filmato possono essere realizzate non solo nell'ambiente di creazione di Flash, ma anche in fase di runtime.
- Ogni istanza di clip filmato creata in fase di runtime deve avere:
 - un nome di istanza unico nel MovieClip in cui è contenuto
 - un valore di profondità (z-order) ugualmente unico
- La profondità specificata determina in che ordine il nuovo clip si sovrappone agli altri clip sulla stessa linea temporale.
- Per ogni livello di profondità avremo un unico clip. Creando un clip ad un livello già occupato da un altro clip quest'ultimo viene sostituito dal nuovo clip creato.

CREAZIONE DI CLIP FILMATO IN FASE DI RUNTIME

- Posso creare un nuovo MovieClip solo in un MovieClip già esistente. Il MovieClip che contiene il MovieClip creato (runtime o design time) viene indicato dalla proprietà **_parent** del MovieClip.
- Ho tre metodi a disposizione per creare un nuovo MovieClip in un MovieClip esistente:
 - **MovieClip.createEmptyMovieClip()**
 - **MovieClip.attachMovie()**
 - **MovieClip.duplicateMovieClip()**