

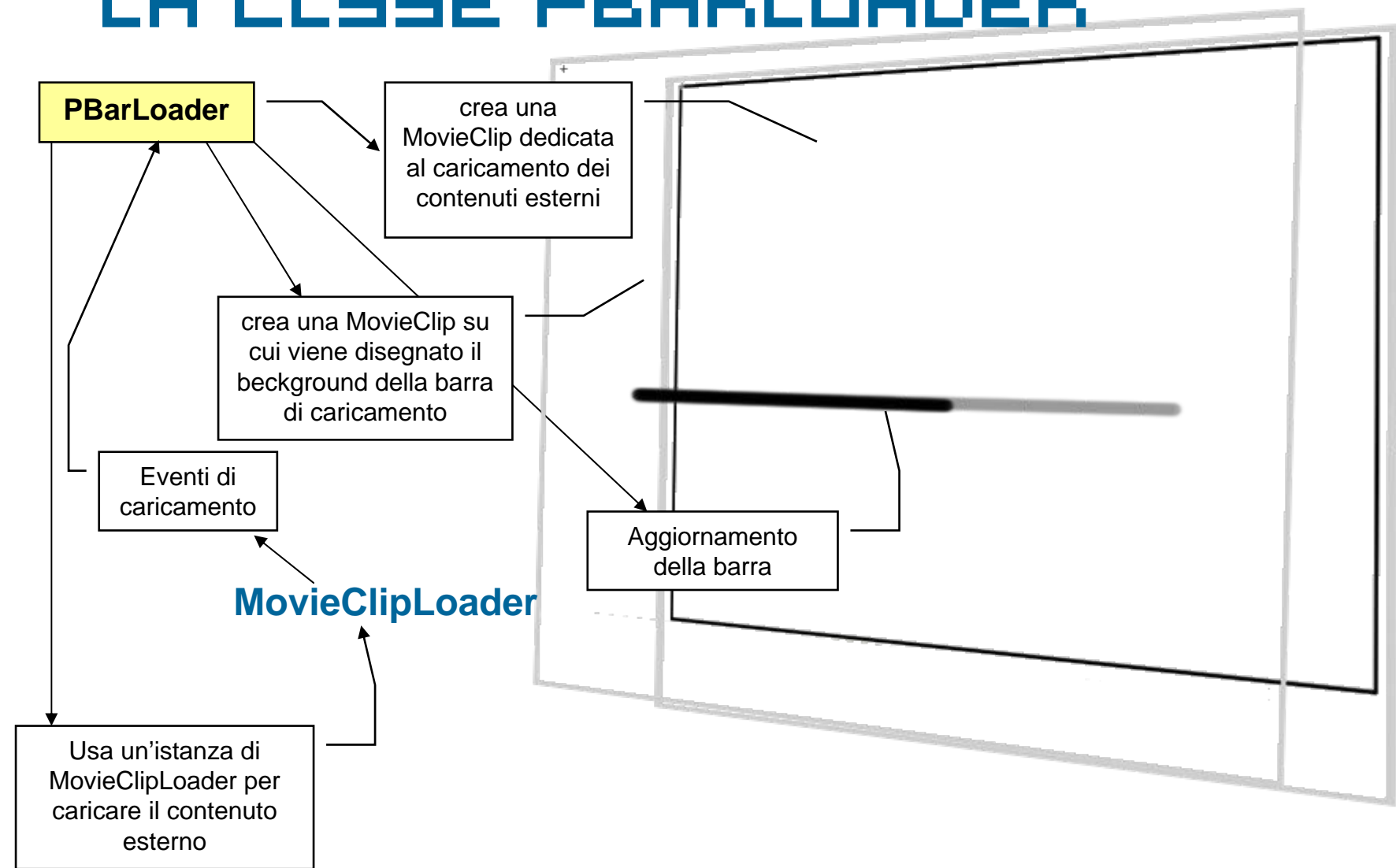
7 APRILE 2008

# DUE TIPI DI CLASSI UTENTE

## CLASSI "STAND ALONE"

- Non hanno bisogno di componenti grafiche per funzionare.
- Se hanno una componente grafica creano una o più **MovieClip** con il metodo **MovieClip.createEmptyMovieClip** e gli elementi grafici runtime.
- Se estendono altre classi queste classi a loro **non** devono estendere la classe **MovieClip**.
- Queste classi si utilizzano creandole con l'operatore **new**.

# LA CLASSE PBarLoader

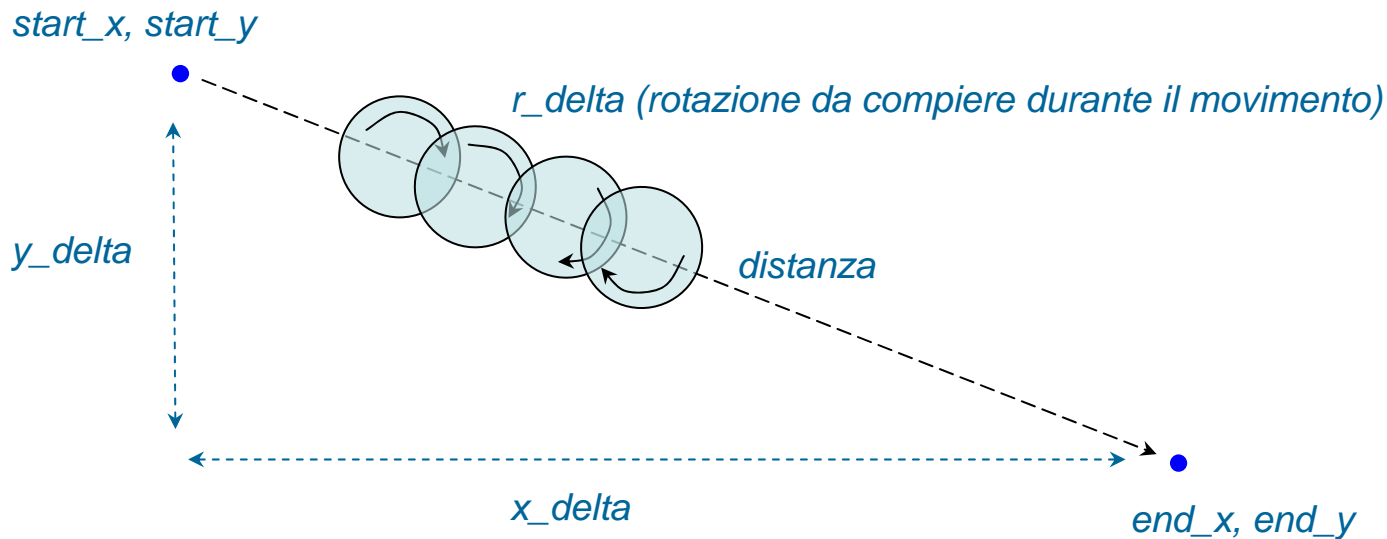


# CLASSI CHE ESTENDONO LA CLASSE MOVIECLIP

- Sono sempre collegate a una MovieClip realizzata design-time
- Di solito operano su le componenti grafiche e/o sulle animazioni di cui la MovieClip è composta.
- Ogni elemento che viene utilizzato deve avere un nome e questo nome deve essere dichiarato come proprietà della classe.
- La stessa classe può essere collegata a MovieClip graficamente diverse.

# GESTIONE DEL MOVIMENTO

```
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;
```



# ANCORA SULLA GESTIONE DEGLI EVENTI

# IL MODELLO GESTORE DI EVENTI

```
stella_mc.onRelease = function(){  
    this._rotation++;  
}
```

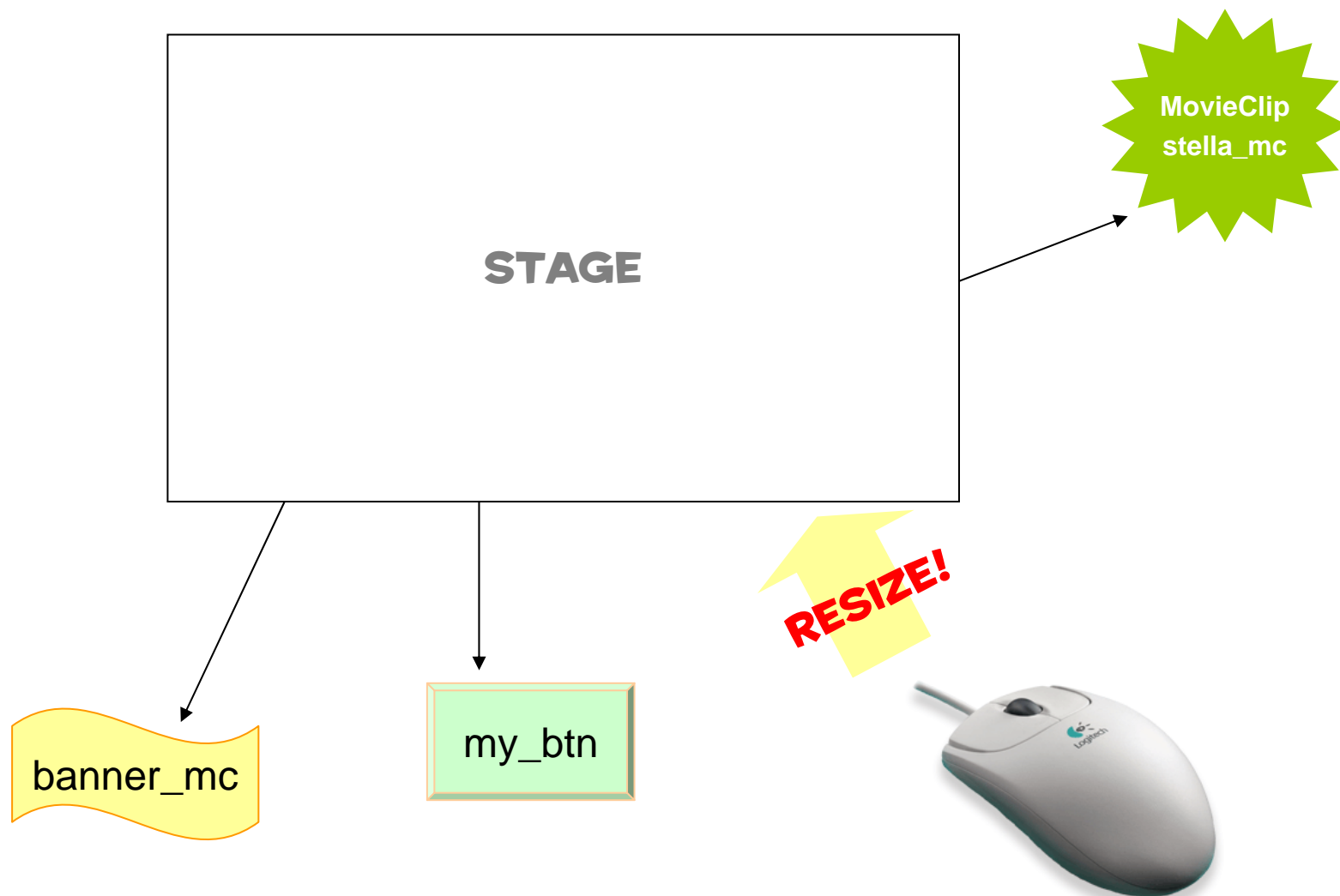
MovieClip  
stella\_mc

CLICK!





# IL MODELLO BROADCASTER-LISTENER



# CARICAMENTO ED ESECUZIONE DI UN FILMATO FLASH (SWF)

## IL PLAYER: CARICAMENTO

- Il filmato verrà caricato in una **MovieClip** utilizzando il metodo **MovieClip.loadMovie()**
- Verrà controllato lo stato del caricamento, usando i metodi **MovieClip.getBytesLoaded()** e **MovieClip.getBytesTotal()**.

## IL PLAYER: ESECUZIONE

- Verrà controllato lo stato dell'esecuzione attraverso le proprietà **MovieClip.\_currentframe** e **MovieClip.\_totalframes**.
- Una proprietà **autoplay** stabilirà se l'esecuzione del filmato sarà automatica o partirà a richiesta dell'utente
- Una proprietà **loop** stabilirà se l'esecuzione del filmato verrà ripetuta indefinitamente e si fermerà all'ultimo frame.

## IL PLAYER: LIMITI

- Il filmato verrà eseguito al frame rate del filmato che lo carica e non a quello per cui è stato creato.
- Per semplicità la dimensione del filmato riprodotto verrà assunta come fissa e conosciuta.
- Si assumerà che sulla linea temporale del filmato non ci siano comandi che ne modifichino il flusso d'esecuzione.

## IL PLAYER: COMANDI

- Comando **Play**. Verrà realizzato utilizzando il metodo **MovieClip.Play()**.
- Comando **Pause**. Verrà realizzato utilizzando il metodo **MovieClip.Stop()**.
- Comando **Stop**. Verrà realizzato utilizzando il metodo **MovieClip.GotoAndStop( 1)**.
- L'esecuzione potrà saltare ad un punto arbitrario utilizzando il metodo **MovieClip.GotoAndPlay(...)**.

# IL PLAYER: INTERFACCIA

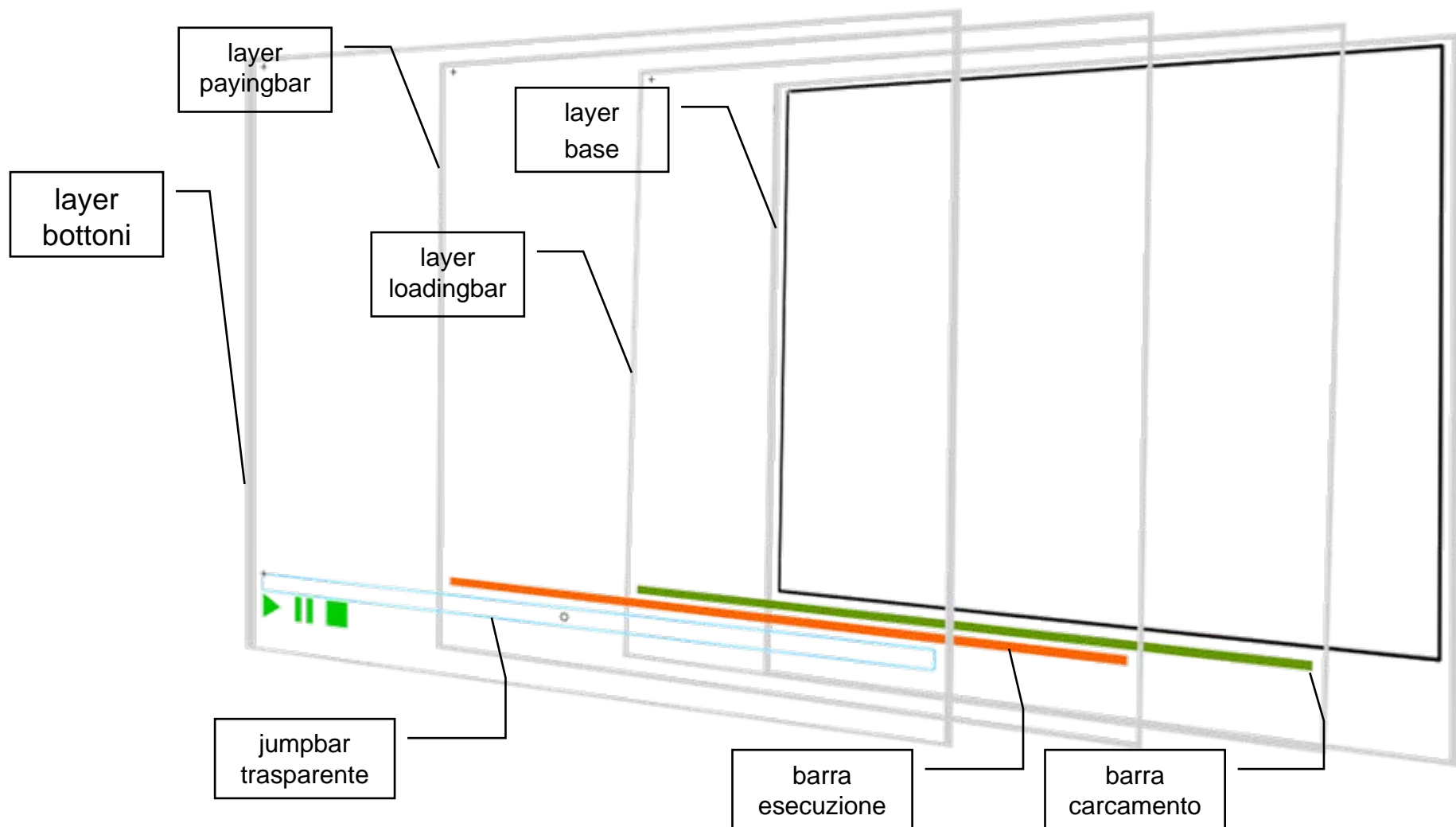
- Un rettangolo indicherà l'area in cui verrà caricato il filmato.
- Una barra indicherà lo stato del caricamento
- Una seconda barra indicherà lo stato di avanzamento dell'esecuzione
- Tre bottoni saranno associati ai comandi **Play**, **Pause** e **Stop**.
- Cliccando sulla barra si potrà saltare ad un punto arbitrario del filmato.

## IL PLAYER: GLI ELEMENTI

- Una MovieClip per caricare il filmato. Verrà creata dinamicamente: ***filmato\_mc***
- Una barra di caricamento (tipo MovieClip): ***loadingbar\_mc***
- Una barra indicatore di esecuzione (tipo MovieClip): ***playingbar\_mc***
- Tre bottoni associati ai comandi **Play**, **Pause** e **Stop**. (tipo Button): ***play\_btn***, ***pause\_btn***, ***stop\_btn***.
- Una barra invisibile per saltare ad un punto arbitrario del filmato: ***iump\_mc*** (tipo MovieClip).



# COSTRUZIONE INTERFACCIA



# IL PLAYER: LE PROPRIETÀ

```
//proprietà private
private var filmato_mc:MovieClip;

private var loadingbar_mc:MovieClip;

private var playingbar_mc:MovieClip;

private var bar_width:Number;

//bottoni
private var play_btn:Button;
private var stop_btn:Button;
private var pause_btn:Button;
private var jump_mc:MovieClip;

private var init_movie:Boolean;

//proprietà pubbliche
public var loop:Boolean = false;
public var autostart:Boolean = true;

//clip in cui viene caricato il filmato
//ogni contenuto verrà sostituito
//viene segnalato lo stato del caricamento
//modificando la sua larghezza
//viene segnalato lo stato dell'esecuzione
//modificando la sua larghezza

//larghezza che ha la barra all'inizio
//corrisponde alla larghezza che avranno le due
//barre quando segnano il 100%

//esegue l'azione "play"
//esegue l'azione "stop"
//esegue l'azione "pause"
//esegue l'azione "jumpTo"
//((salta ad un punto qualsiasi del filmato))

//variabile che serve per far sì che l'azione
//di inizializzazione del filmato
//durante il ciclo ripetuto ?onEnterFrame'
//sia compiuta una volta sola

//queste due proprietà mi consentono di
//personalizzare il comportamento del player:
//autostart: partenza automatica non
//appena il primo frame del filmato
//sono pronti o attesa del play dell'utente
//loop: riproduzione infinita del filmato
//o stop alla fine
//i valori assegnati qui sono quelli di default.
```

# IL PLAYER: I METODI

```
//Questa funzione viene chiamata dalla funzione pubblica
//SWF_player() che è il constructor della classe
//questi comandi vengono quindi eseguiti quando per la prima volta
//il compilatore trova l'istanza della classe sullo stage
//e servono quindi ad inizializzare il nostro player
private function init () {
    // tutti i bottoni vengono abilitati (enabled = true)
    jump_mc.enabled = true;
    play_btn.enabled = true;
    stop_btn.enabled = true;
    pause_btn.enabled = true;

    //A ogni bottone viene assegnata un'azione (viene chiamato un
    //metodo del player). Il player è rappresentato dalla proprietà _parent di
    //ogni bottone che rappresenta la movieClip in cui il bottone è
    //contenuto

    play_btn.onRelease = function () {
        this._parent.playClip();
    }

    stop_btn.onRelease = function () {
        this._parent.stopClip();
    }

    pause_btn.onRelease = function () {
        this._parent.pauseClip();
    }

    jump_mc.onPress = function () {
        var my_percent:Number = this._xmouse/this._parent.bar_width;
        this._parent.jumpTo(my_percent);
    }
    //infine assumo come 100% delle barre la larghezza attuale della barra
    //di esecuzione
    bar_width = playingbar_mc._width;
}
```

# IL PLAYER: I METODI

```
//azione "play"
//oltre a far partire il fimato si rendono
//attivi solo i bottoni necessari
//i bottoni inattivi sono resi semitrasparenti
private function playClip () {
    play_btn.enabled = false;
    stop_btn.enabled = true;
    pause_btn.enabled = true;
    play_btn._alpha = 30;
    stop_btn._alpha = 100;
    pause_btn._alpha = 100;
    jump_mc.enabled = true;
    filmato_mc.play();
    trace(play_btn.enabled);
}

//azione "stop"
//il fimato viene fermato e riportato all'inizio
//sono resi attivi solo i bottoni necessari
//i bottoni inattivi sono resi semitrasparenti
private function stopClip () {
    play_btn.enabled = true;
    stop_btn.enabled = false;
    pause_btn.enabled = false;
    play_btn._alpha = 100;
    stop_btn._alpha = 30;
    pause_btn._alpha = 30;
    jump_mc.enabled = true;
    filmato_mc.gotoAndStop(1);
    trace(play_btn.enabled);
}
```

# IL PLAYER: I METODI

```
//azione "pause"
//il fimato viene fermato alla posizione collente
//sono resi attivi solo i bottoni necessari
//i bottoni inattivi sono resi semitrasparenti
private function pauseClip() {
    play_btn.enabled = true;
    stop_btn.enabled = false;
    pause_btn.enabled = false;
    play_btn._alpha = 100;
    stop_btn._alpha = 30;
    pause_btn._alpha = 30;
    jump_mc.enabled = true;
    filmato_mc.stop();
}

//azione "jump to"
//il fimato viene fatto saltare ad una posizione qualsiasi
//sono resi attivi solo i bottoni necessari
//i bottoni inattivi sono resi semitrasparenti
private function jumpTo (f:Number) {
    play_btn.enabled = false;
    stop_btn.enabled = true;
    pause_btn.enabled = true;
    play_btn._alpha = 30;
    stop_btn._alpha = 100;
    pause_btn._alpha = 100;
    jump_mc.enabled = true;
    filmato_mc.gotoAndPlay(Math.round(f*filmato_mc._totalframes));
}

//queto è il constructor della istanza della classe
//chiama il metodo init
public function SWF_player () {
    init();
}
```

# IL PLAYER: I METODI

```

/* Questo metodo carica e manda in esecuzione il filmato.
Viene impostata la gestione dell'evento onEnterFrame
che ripete indefinitamente una serie di comandi e quindi viene
lancato il caricamento del filmato.
Il nostro gestore onEnterFrame fa tre cose:
1 - Controlla che venga caricato almeno il primo frame del filmato
   e quindi se autostart è falso blocca il filmato sul primo frame.
   NB: la variabile init_movie garantisce che questa operazione venga fatta
   una sola volta
2 - Aggiorna le due barre secondo lo stato del caricamento e lo stato della
   esecuzione.
3 - Controlla quando viene raggiunta la fine del filmato. Quando è raggiunta
   se loop è falso il gestore dell'evento onEnterFrame viene distrutto
   altrimenti l'esecuzione del filmato riprende dall'inizio.
*/

public function loadClip (url:String) {
    //la partenza del filmato deve essere
    //ancora controllata
    init_movie = true;
    this.onEnterFrame = function () {
        // 1 - controllo autostart
        if (init_movie) {           //se devo controllare ancora lo stato del filmato
            if (filmato_mc._framesloaded > 1) { // se è stato caricato
                                                // al meno un frame
                init_movie = false; //il filmato è stato controllato
                                                //non ripeterò più questa azione

                if (autostart) {
                    playClip();           //se autostart è vero eseguo il filmato
                } else {
                    stopClip(); //altrimenti lo metto in stop
                }
            }
        } else {
            //2 - barra caricamento
            loadingbar_mc._width = (filmato_mc.getBytesLoaded()/filmato_mc.getBytesTotal())* bar_width;
            //3 - barra di esecuzione
            playingbar_mc._width = (filmato_mc._currentframe /filmato_mc._totalframes) * bar_width;
            //4 - controllo fine filmato
            if (filmato_mc._currentframe == filmato_mc._totalframes) {
                if (loop) {
                    filmato_mc.gotoAndPlay(1);
                } else {
                    stopClip();
                }
            }
        }
        // impostato il gestore onEnterFrame lancio il caricamento
        // del filmato
        filmato_mc.loadMovie(url);
    }
}

```