



هدفهای رفتاری

- در پایان این فصل از هنرجو انتظار می‌رود:
- اصول ساخت انواع انیمیشن را شرح دهد.
- انیمیشن میانمایی (Tweening) را تولید کرده، تنظیمات آن را انجام دهد.
- انیمیشن فریم به فریم را شرح داده، نحوه ایجاد آنها را در نرم‌افزار به‌طور عملی انجام دهد.
- حلقه فیلم را به همراه کاربرد آن تشریح کرده، بتواند چند انیمیشن را به حلقه فیلم تبدیل کند و تنظیمات آن را تغییر دهد.
- اصول ساخت انیمیشن به روش ضبط زنده را بداند و آن را در عمل پیاده‌سازی کند.

نرم افزار دایرکتور علاوه بر قابلیت ساخت چند رسانه‌های مختلف امکان ساخت بعضی از این رسانه‌ها را نیز در محیط نرم‌افزاری خود فراهم کرده است. انیمیشن یکی از رسانه‌هایی است که امکان تولید آن در دایرکتور به روش‌های مختلفی وجود داشته، به طوری که کاربر می‌تواند پس از خلق انیمیشن مورد نظر، آن را در چند رسانه‌ای مربوطه مورد استفاده قرار دهد. علاوه بر این به دلیل وجود امکانات لازم برای ساخت انیمیشن، به طور اختصاصی نیز می‌توان از این نرم‌افزار برای ساخت انیمیشن استفاده کرد.

در حالت کلی انیمیشن تغییرات ظاهری یک اسپریت در واحد زمان است؛ تغییراتی مانند مکان، اندازه، چرخش یا محو شدن (Fade). ما در ادامه هر چه بیشتر شما را با روش‌های مختلف ساخت انیمیشن در نرم‌افزار دایرکتور آشنا خواهیم کرد. ساخت انیمیشن در نرم‌افزار دایرکتور دارای روش‌های مختلفی است که سه روش بسیار مهم و پرکاربرد آن عبارتند از:

۱- میانمایی (Tweening)

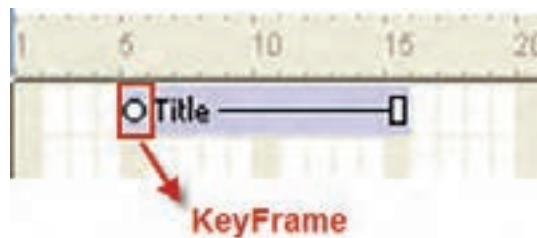
۲- انیمیشن فریم به فریم (Frame By Frame)

۳- ضبط زنده (Real Time Recording)

برای این که به طور اختصاصی با هر یک از این روش‌ها آشنا شوید. ما تا انتهای فصل همراهی کنید.

۶-۱- خلق انیمیشن به کمک Tweening:

اگر به یک اسپریت عادی که در آن تغییراتی ایجاد نشده، نگاه کنید؛ خواهید دید که شکل اولین فریم آن دایره است ولی بقیه فریم‌ها به شکل مستطیل ساده هستند. به این مفهوم که اولین فریم هر اسپریت در پنجره Score به صورت پیش فرض یک فریم کلیدی (KeyFrame) است. فریم کلیدی در واقع تعیین می‌کند وضعیت شیء تا پایان اسپریت چگونه باشد. (شکل ۶-۱)



شکل ۶-۱

به عبارت دیگر زمانی که شما یک اسپریت را انتخاب می‌کنید و در آن تغییراتی می‌دهید (به عنوان مثال آن را در صحنه جابجا می‌کنید) این تغییر در فریم کلیدی اول ضبط شده و تا پایان اسپریت حفظ می‌شود؛ زیرا تا پایان اسپریت هیچ فریم کلیدی (Key Frame) دیگری وجود ندارد. پس اگر شما در یکی از فریم‌های دیگر اسپریت یک Key Frame اضافه کنید و در آن

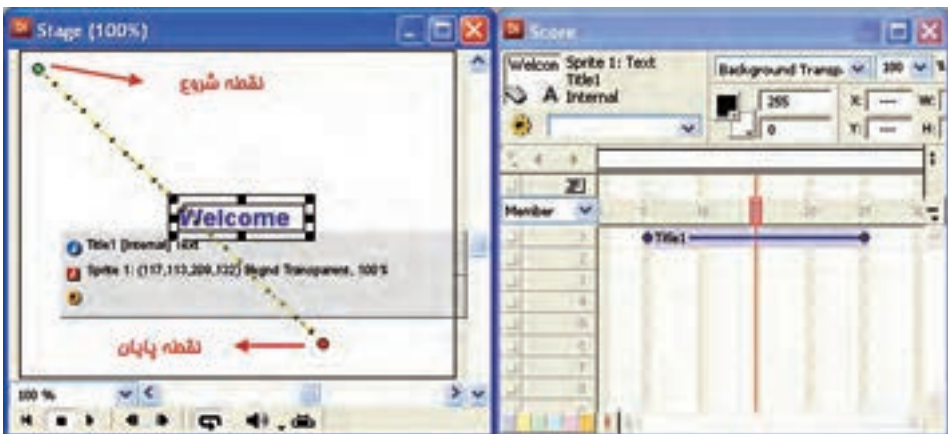
Key Frame، محل شیء را تغییر دهید، یک انیمیشن ساده ایجاد کرده‌اید. در این روش که به میانمایی یا Tweening معروف می‌باشد، دایرکتور بقیه حالت‌های بین دو فریم کلیدی را به صورت خودکار می‌سازد و شما فقط با دو Keyframe، ابتدا و انتهای مسیر مورد نظر را تعیین می‌کنید.

توجه داشته باشید که علاوه بر مسیر می‌توان خصوصیات مانند اندازه (Size)، چرخش (Rotate)، مایل کردن (Skew)، شفافیت (Blend)، رنگ پیش زمینه (Foreground Color) و رنگ پس زمینه (Background Color) را نیز در انیمیشن Tweening تغییر دهید. برای این که بیشتر با این روش و کاربرد آن آشنا شوید به مثال زیر توجه کنید:

مثال ۱: با استفاده از روش انیمیشن میانمایی یک حرکت ساده از گوشه سمت چپ بالای صفحه به گوشه سمت راست پایین صفحه طراحی کنید.

مراحل انجام کار:

- ۱- نمایش جدیدی ایجاد کنید.
- ۲- در گوشه سمت چپ بالا صحنه یک اسپریت دلخواه را قرار دهید.
- ۳- ابتدا با کلیک ماوس روی فریم انتهایی اسپریت آن را انتخاب کنید.
- ۴- به یکی از دو روش زیر یک Key Frame در این فریم درج کنید.
الف) از طریق منو Insert/ Key Frame (Ctrl+Alt+K)
- ب) کلیک راست روی فریم آخر و اجرای فرمان Insert Keyframe
- ۵- به کمک ماوس اسپریت را جابجا کرده، آن را به گوشه پایین سمت راست منتقل کنید. همانطور که در شکل ۲-۶ ملاحظه می‌کنید، دایرکتور مسیر حرکت را برای شما ترسیم می‌کند که یک خط راست بین دو فریم کلیدی ابتدا و انتهای اسپریت می‌باشد.

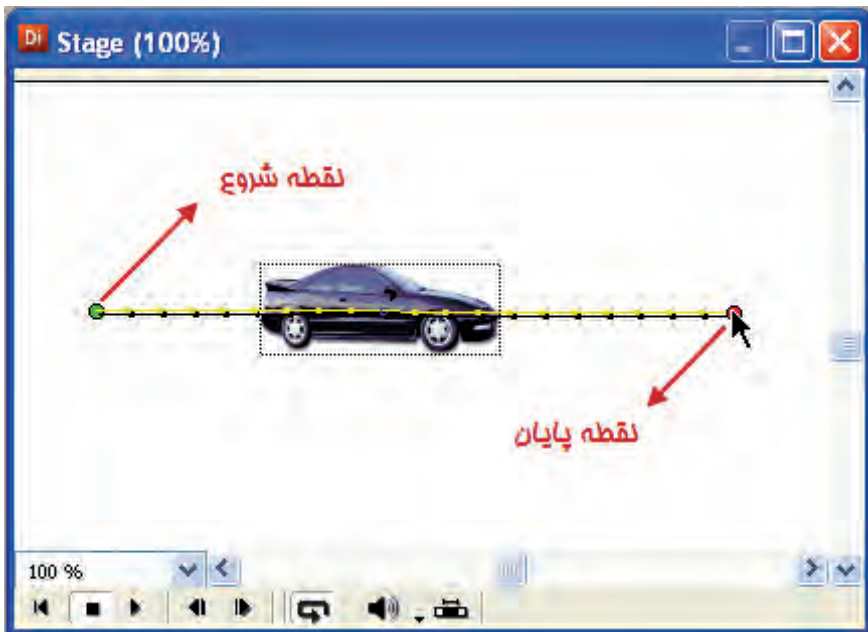


شکل ۲-۶ ایجاد انیمیشن میانمایی به کمک اضافه کردن Keyframe

۶- اکنون نمایش را با انتقال هد به ابتدای Score اجرا کنید، خواهید دید که اسپریت از بالا به پایین با سرعت ثابت (Tempo) حرکت می‌کند.

نکته: در انیمیشن حرکتی، نیازی به اضافه کردن فریم‌های کلیدی به صورت دستی نیست زیرا دایرکتور می‌تواند به صورت خودکار تغییرات را بصورت KeyFrame ثبت کند. بنابراین کافی است مراحل ساده زیر را طی کنید:

۱. اسپریت را در صحنه ایجاد کنید.
۲. در صحنه، نقطه ثبت اسپریت را به کمک ماوس به انتهای مسیر درگ کنید تا فریم کلیدی در آخرین فریم اسپریت ثبت شده، مسیر حرکت ساخته شود. (شکل ۳-۶)



شکل ۳-۶ ایجاد انیمیشن حرکتی به روش درگ کردن اسپریت

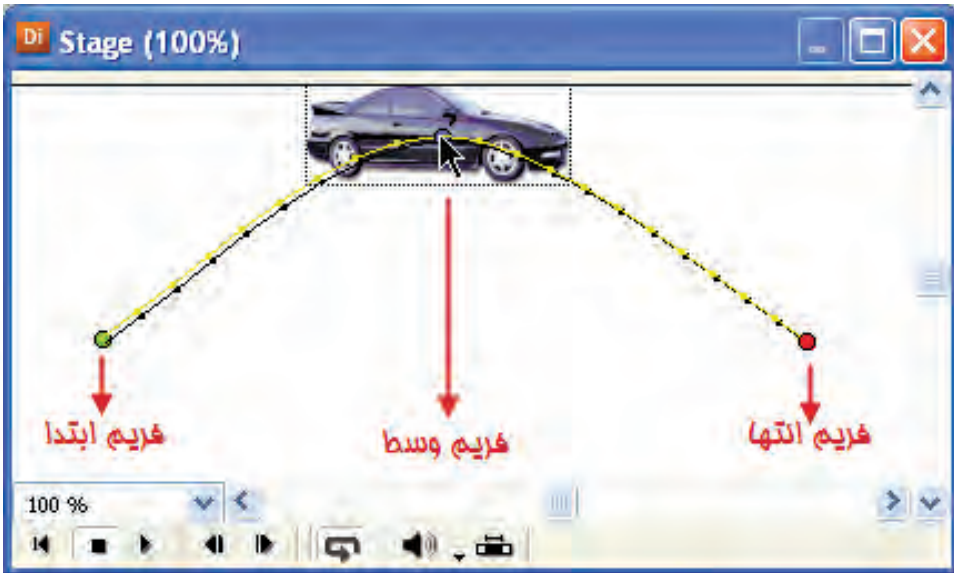
نکته: برای ایجاد یک انیمیشن حداقل نیاز به دو فریم کلیدی است که اولی به طور پیش فرض در هر اسپریتی وجود دارد و دومی را کاربر در فریم دلخواهی از اسپریت اضافه می‌کند و وضعیت شیء را در آن تعیین می‌کند.

مثال ۲:

با استفاده از انیمیشن Tweening، اسپریت مثال ۱ را که بر روی مسیر مستقیم حرکت می‌کند، بر روی مسیر منحنی حرکت دهید.

مراحل انجام کار:

هد را به فریم وسط اسپیرایت انتقال داده، دوباره با ماوس نقطه ثبت اسپیرایت را که اکنون در وسط مسیر قرار دارد، به طور مستقیم به سمت بالا و وسط stage درگ کنید تا مسیر منحنی شکل ایجاد شود. (شکل ۴-۶) سپس نمایش را اجرا کنید.

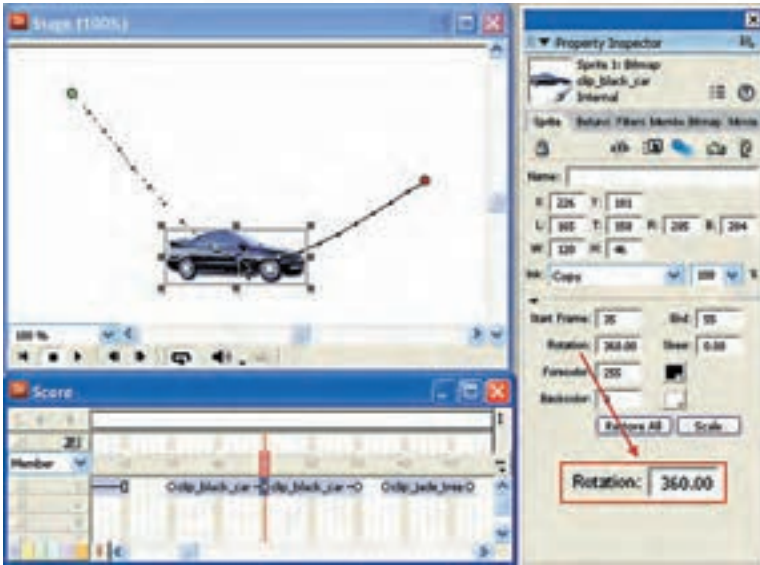


شکل ۴-۶ تبدیل مسیر مستقیم به منحنی

۲-۶-۱- ویرایش فریم‌های کلیدی

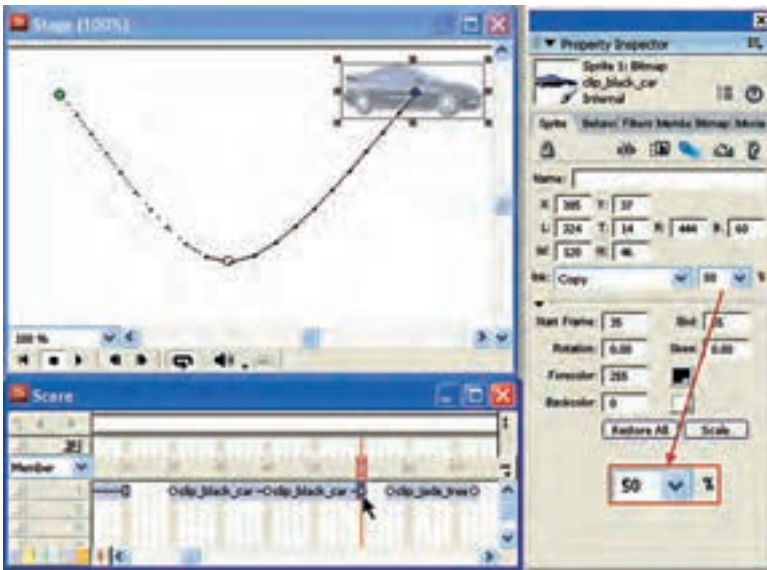
در این مرحله می‌خواهیم انیمیشن نمایش مثال قبل را ویرایش کنیم. روش کار بسیار آسان است و کافی است فریم کلیدی مورد نظر را انتخاب و شیء مورد نظر را جابجا یا تغییراتی مانند چرخش و آمیختگی (Blend) را به آن اعمال کنید. برای این منظور مراحل زیر را انجام دهید:

- ۱- فریم کلیدی اول اسپیرایت را انتخاب کرده، شیء را در صحنه به سمت بالا جابجا کنید (به کمک دکمه Shift می‌توانید عمل جابجایی را در خط مستقیم انجام دهید).
- ۲- فریم کلیدی دوم را انتخاب کرده، شیء را به پایین صحنه جابجا کنید. سپس در کادر Rotate نوار ابزار اسپیرایت یا پنجره Property Inspector، عدد ۳۶۰ را جهت چرخش شیء به اندازه یک دور کامل در جهت عقربه‌های ساعت وارد کنید. (شکل ۵-۶)



شکل ۵-۶ جابجا کردن اسپریت به سمت پایین و تنظیم چرخش

۳- فریم کلیدی آخر را انتخاب کرده، شیء را در بالای صحنه قرار دهید. سپس میزان شفافیت اسپریت را برابر ۵۰ تنظیم کنید. (شکل ۶-۶)



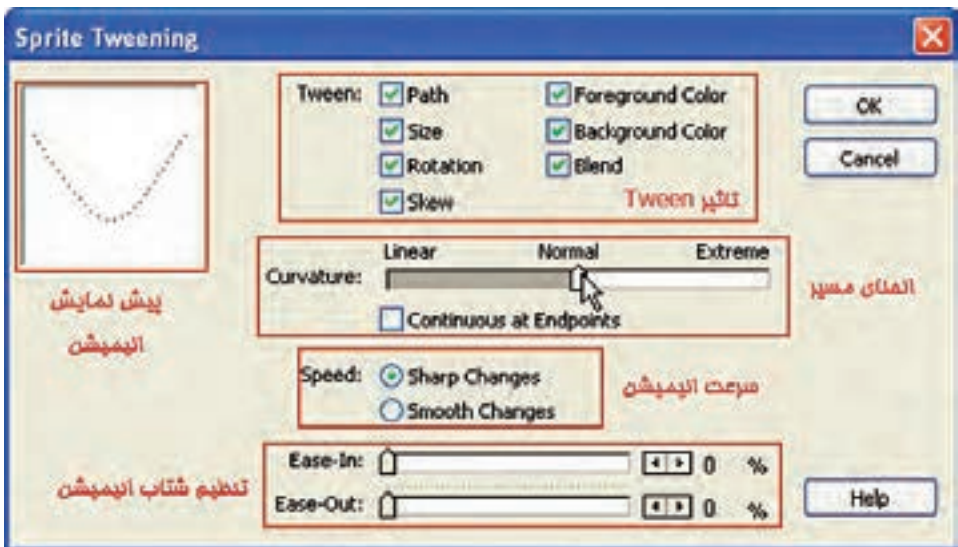
شکل ۶-۶ جابجا کردن اسپریت به سمت بالا و تنظیم شفافیت

۴- نمایش را ذخیره کرده، اجرا کنید. همانطور که ملاحظه می‌کنید از فریم کلیدی اول تا دوم دو نوع حرکت ایجاد شده است. حرکت اول جابجایی و دیگری چرخش به اندازه یک دور کامل (۳۶۰ درجه) می‌باشد. از فریم دوم تا سوم، اسپریت ضمن حرکت به سمت بالا، به صورت معکوس نیز می‌چرخد. دلیل این امر آن است که زاویه آن از ۳۶۰ درجه به سمت صفر تغییر خواهد کرد. ضمن این که در حین حرکت و چرخش، شفافیت آن کمتر یا به میزان نصف کاهش می‌یابد.

۳-۱-۶- تنظیمات Tweening

دایرکتور دارای تنظیماتی جهت کنترل انیمیشن Tweening است، به طوری که به کمک آن، کاربر می‌تواند حرکات مورد نظر را به صورت واقعی‌تری طراحی و ایجاد نماید. از مهمترین این تنظیمات حرکتی می‌توان به تغییر شتاب حرکت، سرعت انیمیشن و تنظیمات مربوط به مسیر حرکتی اشاره کرد. ما برای آشنایی هرچه بیشتر شما با این تنظیمات به بررسی تعدادی از این پارامترها و نحوه کنترل آنها می‌پردازیم. مراحل انجام کار:

- ۱- اسپریتی که بر روی آن انیمیشن میانمایی را تعریف کرده‌اید انتخاب کنید.
- ۲- از منو Modify/Sprite/Tweening را انتخاب کنید تا کادر محاوره‌ای آن مطابق شکل ۶-۷ باز شود.



شکل ۶-۷ کادر محاوره‌ای تنظیمات Tweening

۳- در بخش Tween هر خصوصیتی را که نمی خواهید عمل Tween بر روی آن اعمال شود، از حالت انتخاب خارج کنید. به طور پیش فرض همه گزینه‌ها در حالت انتخاب قرار دارند.

۴- از طریق اسلایدر موجود در بخش Curvature چگونگی انحنای مسیر میانایی را تعیین کنید. به این ترتیب که هرچه اسلایدر به سمت راست (Extreme) حرکت داده شود، انحنای مسیر بیشتر و برعکس هرچه اسلایدر به سمت چپ (Linear) حرکت کند، انحنای مسیر کمتر و در نهایت خطی می‌شود.

نکته: به طور پیش فرض اسلایدر در وسط یا حالت Normal قرار دارد که باعث حرکت اسپریت در همان مسیری می‌شود که برای آن تعیین کرده‌اید.

نکته: برای مشاهده مسیر حرکت از بخش پیش‌نمایش این پنجره که در بخش بالا و سمت چپ پنجره قرار دارد، استفاده کنید.

۵- اگر می‌خواهید ابتدا و انتهای مسیرهای بسته، مانند مسیر دایره‌ای را به صورت نرم به یکدیگر وصل کنید، گزینه Continuous at Endpoints را فعال کنید.

۶- بخش Speed (سرعت) دارای دو گزینه با عملکردهای متفاوت است:

- Sharp Changes: حرکت اسپریت بین دو فریم کلیدی به صورت عادی و سریع صورت می‌گیرد.
- Smooth Changes: حرکت اسپریت بین دو فریم کلیدی به صورت نرم و هموار صورت می‌گیرد.

۴-۱-۶- اصول پخش فریمهای انتخاب شده:

نرم‌افزار دایرکتور این امکان را به کاربران خود می‌دهد که فقط محدوده معینی از فریمهای یک اسپریت را پخش کند. در مثال قبل همانطور که مشاهده کردید انیمیشن ایجاد شده، از ابتدا به انتها پخش شد. تصور کنید بخواهیم در این مثال فقط انیمیشن موجود در فریمهای کلیدی اول تا دوم را پخش کنید برای این منظور کافی است:

۱- فریم کلیدی اول را کلیک کرده، دکمه Ctrl را پایین نگه داشته و روی فریم کلیدی دوم نیز کلیک کنید تا این محدوده انتخاب شود. (یادآوری: می‌توانید یک یا چند اسپریت را که در یک کانال قرار دارند نیز انتخاب کنید).

۲- بر روی دکمه Selected Frames Only موجود در پنجره Stage کلیک کنید تا محدوده فریم‌های مورد نظر انتخاب شوند، در این حالت خط سبز رنگی در کانال فریم‌ها ظاهر می‌شود.

شکل ۶-۸



شکل ۶-۸ فریم‌های منتخب جهت پخش

۳- حال دکمه Loop را فعال و نمایش را پخش کنید.

۶-۲- انیمیشن فریم به فریم (Frame by Frame):

این سبک انیمیشن که به آن در اصطلاح انیمیشن سنتی نیز گفته می‌شود، هر یک از فریم‌های آن با یک کادر تصویر پرشده، سپس از پخش پشت سرهم کادرها، یک حرکت یا انیمیشن ایجاد خواهد شد. به کمک این روش می‌توانید انیمیشن‌های کامل و پیشرفته‌ای را جهت خلق آثاری همچون کارتن یا تبلیغات حرفه‌ای بسازید. برای ساخت کادرهای تصویری مورد نیاز این روش نیز، می‌توان از نرم‌افزارهای مختلفی چون Photoshop و Coreldraw و... استفاده کنید.

۶-۲-۱- مراحل طراحی یک انیمیشن فریم به فریم:

برای ساخت یک انیمیشن فریم به فریم در نرم‌افزار دایرکتور، ابتدا لازم است کادرهای تصویری مورد نظر را در سایر نرم‌افزارها ایجاد کرده، سپس به محیط نرم‌افزار Import کرد. بر این اساس می‌توانید مراحل زیر را جهت ساخت یک انیمیشن به روش "فریم به فریم" دنبال کنید.

۱- طراحی فریم‌ها یا کادرهای مورد نیاز انیمیشن

۲- وارد کردن فریم‌ها پشت سر هم در Cast

۳- انتخاب فریمی از کانال‌های Score که می‌خواهید اسپریت در آنجا ساخته شود.

۴- انتخاب فریم‌های انیمیشن در Cast (اولین Cast member را انتخاب کرده، دکمه Shift را نگه داشته و روی آخرین Cast member کلیک کنید).

۵- اجرای فرمان Modify / Cast to time

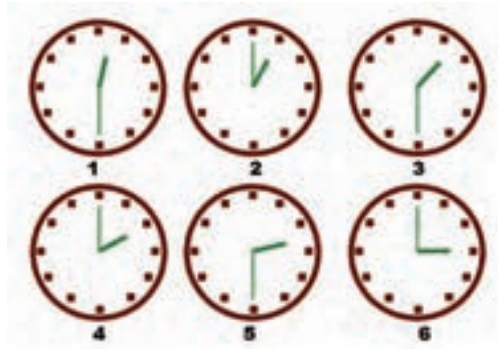
برای اینکه ساخت یک انیمیشن فریم به فریم را در عمل تجربه کنید؛ به مثال ۳ توجه کنید.

مثال ۳:

حرکت سریع زمان را توسط ۶ حالت مختلف یک ساعت نمایش دهید.

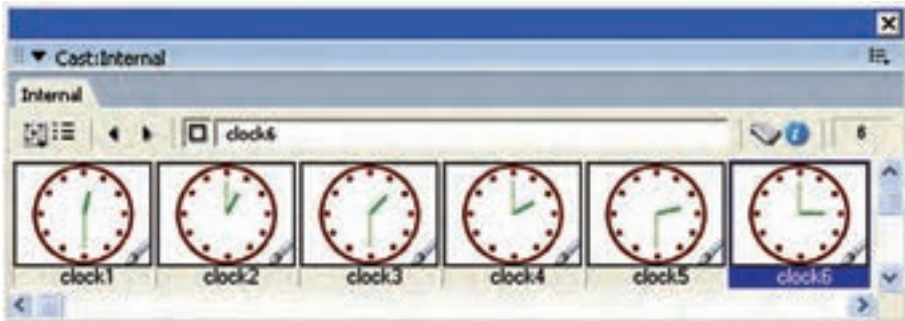
مراحل انجام کار:

۱- در Photoshop حالت‌های مختلف ساعت را مطابق شکل ۹-۶ طراحی کنید (راهنمایی: هر حالت را در یک فایل ذخیره کنید).



شکل ۹-۶

۲- فایل‌های ذخیره شده را وارد دایرکتور کنید. به طوری که خانه‌های ۱ تا ۶ پنجره Cast را مطابق شکل ۱۰-۶ در بر بگیرد.



شکل ۱۰-۶

۳- کلیه خانه‌ها را در پنجره Cast انتخاب کنید. (شماره ۱ تا ۶)

۴- فریم اول از کانال شماره یک اسپریت را انتخاب کنید.

۵- از طریق منوها فرمان $\text{Cast to time} \rightarrow \text{Modify}$ را اجرا کنید تا اسپریت ساخته شود. همانطور که مشاهده می‌کنید طول پیش فرض این اسپریت ۶ فریم است؛ زیرا دایرکتور برای هر حالت یک فریم در نظر می‌گیرد. اکنون انیمیشن آماده است و می‌توانید آن را

پخش کنید؛ اما بهتر است برای درک بهتر عملیات انجام شده، آن را به صورت دستی تست کنید. یعنی اگر فریم شماره ۲ کلیک کنید، در صحنه حالت دوم ساعت را مشاهده خواهید کرد و به همین ترتیب حالت‌های مختلف ساعت تا فریم ۶ ادامه می‌یابد.

۶- انیمیشن را پخش کنید. اگر سرعت خیلی بالاست می‌توانید به یکی از روش‌های زیر آن را کم کنید:

- چند برابر کردن طول اسپرایت (به‌عنوان نمونه اگر طول اسپرایت را ۱۲ فریم کنید، دایرکتور دو فریم برای هر حالت در نظر می‌گیرد.)

- کاهش سرعت هد (Tempo)

۷- نمایش را با نام Clock ذخیره کنید.

۳-۶- ایجاد حلقه فیلم (loop Film)

توسط این ویژگی شما می‌توانید یک انیمیشن را پس از طراحی، به صورت یک عضو مجزا در کتابخانه خود نگهداری کنید و هر جا که لازم شد از آن استفاده کنید (مانند Movie clip در Flash).

مراحل انجام کار:

۳- انیمیشن خود را طراحی کنید.

۴- کل اسپرایت را انتخاب و از آن کپی بگیرید. (copy)

۵- در یکی از خانه‌های خالی Cast آن را بچسبانید. (paste)

۶- در کادر محاوره‌ای Create Film Loop نام دلخواهی را وارد کرده، آن را تایید کنید. (نام پیش‌فرض آن Film Loop می‌باشد)

نکته: روش آسان ساخت Film Loop: پس از طراحی انیمیشن، اسپرایت آن را در Score به کمک ماوس گرفته، روی یکی از خانه‌های خالی Cast درگ کنید تا کادر محاوره‌ای Create Film Loop ظاهر شود و آن را با نام دلخواه تایید کنید.

مثال ۴:

می‌خواهیم انیمیشن Clock را که در مرحله قبلی ساخته‌ایم، به صورت یک Film loop (چرخه فیلم) ذخیره کنیم.

مراحل انجام کار:

اسپرایت را انتخاب و جلوه Ink آن را به Background Transparent تغییر دهید.

اسپرایت انیمیشن را به سمت یکی از خانه‌های خالی پنجره Cast درگ کنید.

در کادر محاوره‌ای ظاهر شده، نام Film Clock را وارد کرده، آن را تایید کنید. به این

ترتیب یک حلقه فیلم با آیکنی ویژه مطابق شکل ۱۱-۶ در Cast ساخته می‌شود که حاوی انیمیشن طراحی شده است.



شکل ۱۱-۶ چرخه فیلم ایجاد شده

۴- اکنون می‌توانید از این انیمیشن در هر جای نمایش خود استفاده کنید به‌عنوان مثال اسپریت موجود در صحنه را حذف کرده، به‌جای آن حلقه فیلم ایجاد شده را وارد صحنه کنید و نمایش را پخش کنید.

● توجه: شما نباید فریم‌های اصلی انیمیشن (تصاویر مربوط به ۶ حالت ساعت) را که در سلولهای ۱ تا ۶ پنجره Cast ذخیره کرده‌اید، حذف کنید، زیرا چرخه فیلم بر پایه این عناصر ساخته شده است.

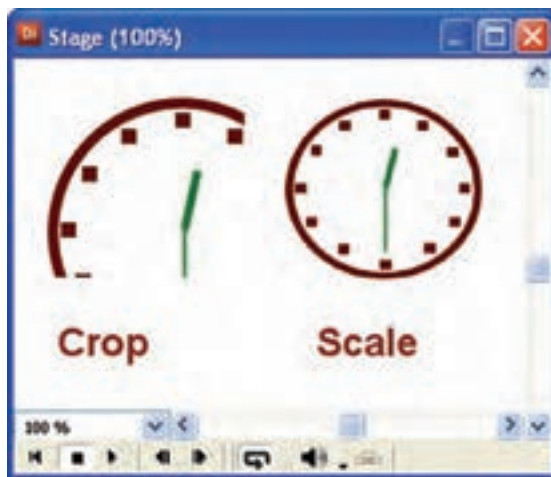
۱-۳-۶- تنظیمات حلقه فیلم:

پس از ایجاد حلقه فیلم می‌توانید برخی از تنظیمات آن را تغییر دهید، برای اینکار لازم است حلقه فیلم را انتخاب و از زبانه Film Loop موجود در Property Inspector کمک بگیرید. شکل ۱۲-۶ این زبانه را نشان می‌دهد.



شکل ۱۲-۶ تنظیمات Film Loop

گزینه	کاربرد
Framing	(قاب‌بندی): دارای دو گزینه Crop و Scale می‌باشد که بطور پیش فرض گزینه Scale آن فعال است و به شما اجازه می‌دهد اندازه حلقه فیلم را در صحنه بزرگ و یا کوچک کنید، چنانچه Crop را برگزینید در صورت کوچک کردن قاب (کادر دور حلقه فیلم)، بخشی از تصویر اصلی حلقه فیلم برش داده خواهد شد. در شکل ۱۳-۶ تاثیر این دوگزینه را در اثر کوچک کردن قاب مشاهده می‌کنید.
Center	این گزینه بطور پیش فرض غیر فعال می‌باشد و زمانی که Crop را انتخاب کنید فعال خواهد شد و هنگام تغییر اندازه قاب نقطه ثبت حلقه فیلم را در مرکز قاب قرار می‌دهد.
Audio	شما قادرید با قرار دادن یک صدا در یکی از کانالهای صدای پنجره Score و انتخاب آن به هنگام ساخت حلقه فیلم صدا را نیز به همراه انیمیشن ضبط کنید (اگر میخواهید صدا در حلقه فیلم پشت سرهم تکرار شود لازمست صدا را در Cast انتخاب و از طریق زبانه Sound گزینه Loop را فعال کنید) حال با غیر فعال کردن گزینه Audio می‌توانید مانع پخش صدای موجود در Film Loop شوید.
Loop	چنانچه این گزینه فعال باشد حلقه فیلم در صحنه بطور دائم پخش می‌شود، در غیر این صورت فقط یکبار پخش می‌شود.



شکل ۱۳-۶ اثر گزینه‌های Framing

۴-۶- معکوس کردن یک انیمیشن: Reverse sequence

با اجرای این فرمان روی اسپرایتی که حاوی انیمیشن است، کل انیمیشن به صورت معکوس پخش می‌شود. یکی از کاربردهای مهم آن کمک به طراحی انیمیشن‌های متقارن است. به عنوان مثال حرکت به راست صحنه و بلافاصله به صورت معکوس در همان مسیر به چپ یا در انیمیشن حرکت ساعت (مثال ۳)، اگر از انیمیشن ساخته شده یک کپی گرفته، پشت سر اسپرایت اول بچسبانیم؛ با اعمال فرمان Reverse Sequence از منوی Modify می‌توانیم حرکت ساعت را به صورت معکوس داشته باشیم.

مراحل انجام کار:

- ۱- انیمیشن خود را برای یک اسپرایت دلخواه طراحی کنید.
- ۲- کل اسپرایت را انتخاب کنید.
- ۳- از منو دستور Reverse sequence → Modify را برگزینید.

۵-۶- انیمیشن با تکنیک ضبط زنده: (Recording time Real)

یکی از روشهای جالب دایرکتور برای ایجاد یک حرکت، ضبط زنده می‌باشد، به این ترتیب که هد را روی فریمی از اسپرایت مورد نظر که می‌خواهید حرکت از آنجا آغاز شود، قرار دهید. به عنوان مثال می‌توانید روی فریم اول اسپرایت کلیک کرده، سپس از منوی Control روی گزینه Real-time Recording کلیک کنید تا فعال شود و علامت ضبط که یک دایره قرمز رنگ می‌باشد در کانال اسپرایت مورد نظر دیده می‌شود (شکل ۱۴-۶). اکنون با ماوس اسپرایت را گرفته، آزادانه به هر طرف که می‌خواهید جابجا کنید. هد شروع به حرکت کرده و در هر فریم اسپرایت، یک Keyframe ایجاد می‌کند و این کار تا زمانی که ماوس را رها نکرده، ادامه می‌یابد و به این ترتیب انیمیشن شما ضبط می‌شود.



شکل ۱۴-۶ فعال شدن حالت ضبط زنده

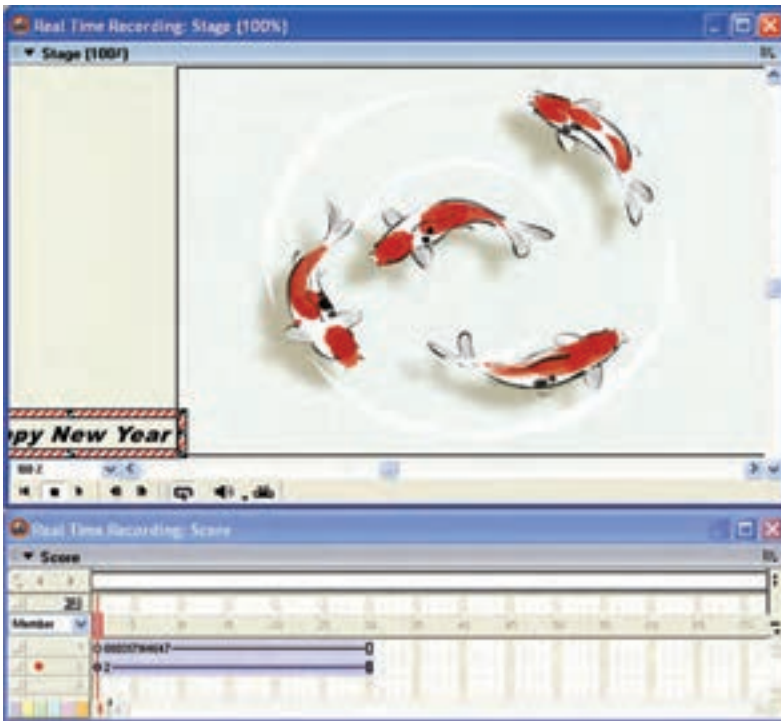
نکته: در این روش سرعت حرکت هد Tempo از اهمیت خاصی برخوردار است؛ زیرا اگر Tempo زیاد باشد شما فرصت لازم جهت جابجا کردن شیء را برای تعداد فریم‌هایی که در نظر گرفته‌اید، نخواهید داشت. ممکن است طول فریم‌های اسپرایت شما زیاد شود؛ پس بهتر است قبل از شروع به کار، سرعت را کم کنید تا کنترل بیشتری روی کار داشته باشید.

مثال:

مراحل ایجاد نمایش زیر را اجرا کنید.

- ۱- نمایش جدیدی ایجاد کنید.

- ۲- یک تصویر را به عنوان زمینه نمایش Import کنید.
- ۳- متن Happy New Year را ایجاد کنید.
- ۴- تصویر را در کانال شماره ۱ پنجره Score قرار دهید و اندازه آن را برابر Stage تنظیم کنید.
- ۵- متن را روی کانال شماره ۲ قرار دهید و مکان اولیه آن را مطابق شکل ۱۷-۶ در خارج از Stage قرار دهید؛ زیرا قرار است متن از خارج Stage وارد آن شود.
- ۶- جلوه جوهر متن را روی Background Transparent تنظیم کنید.
- ۷- سرعت (Tempo) را برابر 6fps در نظر بگیرید.
- ۸- فریم اول اسپریت متن را انتخاب و از منوی Control گزینه Real time Recording را فعال کنید تا علامت دایره قرمز رنگ در کنار کانال شماره ۲ ظاهر شود و دایرکتور را جهت ضبط زنده اعلام کند. (شکل ۱۵-۶)



شکل ۱۵-۶

- ۱۰- نوبت به تعیین مسیر می‌رسد، شما ۵ ثانیه فرصت دارید تا مسیر انیمیشن خود را تعیین کنید. البته این در صورتی است که بخواهید حرکت شما در طول ۳۰ فریم مربوط به

خلاصه مطالب

انیمیشن تغییرات ظاهری یک اسپرایت در زمان است مانند جایجایی و اندازه. دایرکتور می‌تواند به کمک سه تکنیک زیر انیمیشن را تولید کند:

۱- به کمک فریم‌های میانی (Tweening)

۲- انیمیشن فریم به فریم

۳- ضبط زنده (RealTime Recording)

در تکنیک Tweening به کمک تغییر خصوصیات اسپرایت در دو فریم کلیدی یک انیمیشن ساخته می‌شود.

به کمک فرمان Tweening → Sprite → Modify می‌توان تنظیمات Tweening را انجام داد. به وسیله دکمه Selected Frames Only موجود در پنجره Stage، می‌توان فقط فریم‌های انتخاب شده را پخش نمود.

با فرمان Cast to time → Modify می‌توان از تعدادی تصویر، یک انیمیشن فریم به فریم ایجاد نمود. در این حالت هر تصویر (قاب) یک فریم از اسپرایت را اشغال می‌کند.

به کمک فرمان Control → Real-time Recording می‌توان انیمیشن ضبط زنده را ایجاد نمود. این فرمان سریعترین روش ایجاد انیمیشن بوده و با ماوس حرکت می‌کند.

حلقه فیلم امکان ذخیره یک انیمیشن از قبل طراحی شده را به صورت یک عضو مستقل Cast در اختیار کاربر قرار می‌دهد، که باعث کاهش تعداد اسپرایت‌ها خواهد شد. برای ایجاد Film Loop کافی است اسپرایت‌های مورد نظر را انتخاب و آنها را به درون یک سلول خالی از Cast درگ کنید.

با اعمال فرمان Reverse Sequence از منوی Modify می‌توان یک انیمیشن را معکوس کرد. ردپای انیمیشن (Trails) به شما این امکان را می‌دهد که اثر اسپرایت را در حین حرکت اسپرایت روی صحنه ثبت کنید.

Learn In English

Using film loops

A film loop is an animated sequence that you can use like a single cast member. For example, to create an animation of a bird flying across the Stage, you can create a film loop of the sequence of cast members that shows the bird flapping its wings. Instead of using the frame-by-frame technique, create a sprite containing only the film loop and then animate it across as many frames as you need. When you run the animation, the bird flaps its wings and at the same time moves across the Stage.

You can also use film loops to consolidate Score data. Film loops are especially helpful when you want to reduce the number of sprite channels you are using. You can combine several Score channels into a film loop in a single channel.



واژه نامه تخصصی

Continuous	پیوسته
Curvature	انحناء
Extreme	خیلی زیاد
Fade	محو شدن
Real	واقعی
Sequence	رشته
Sharp	تند
Transparent	شفاف
Tween	بینابین

خودآزمایی:

- ۱- روشهای کلی ایجاد انیمیشن را نام ببرید.
- ۲- مفهوم Tweening را توضیح دهید.
- ۳- چگونه می‌توان فریم‌های خاصی از نمایش را پخش کرد؟
- ۴- اصول ساخت انیمیشن فریم به فریم را شرح دهید.
- ۵- کاربرد حلقه فیلم را با مثالی شرح دهید.
- ۶- مفهوم ردپای انیمیشن را توضیح دهید.

پرسش‌های چهار گزینه‌ای:

- ۱- انیمیشن میان‌گذاری برای کدامیک از خصوصیات اسپریت قابل اجرا نمی‌باشد؟
الف) Skew
ب) Blend
ج) Foreground Color
د) Ink
- ۲- کلیدهای فوری درج یک فریم کلیدی کدامند؟
الف) Ctrl+Alt+F
ب) Shift +K
ج) Ctrl+K
د) Ctrl+Alt+K
- ۳- هر اسپریت به‌طور پیش فرض دارای..... فریم کلیدی است.
الف) ۱
ب) ۲
ج) ۳
د) صفر
- ۴- برای ایجاد یک انیمیشن در مسیر منحنی حداقل نیاز به فریم کلیدی است.
الف) ۱
ب) ۲
ج) ۳
د) ۴
- ۵- اگر بخواهیم یک اسپریت را در مسیر دایره‌ای شکل حرکت دهیم، حداقل به فریم کلیدی نیازمندیم.
الف) ۳
ب) ۴
ج) ۵
د) ۶
- ۶- کدام گزینه در کادرمحاوره‌ای Sprite Tweening جهت تعیین انحناى مسیر کاربرد دارد؟
الف) Curvature
ب) Tween
ج) Ease- In
د) Ease- Out

۷- انتخاب گزینه Smooth Changes در کادر محاوره‌ای Sprite Tweening باعث می‌شود:
 الف) حرکت اسپریت بین دو فریم کلیدی به صورت نرم و هموار صورت می‌گیرد.
 ب) حرکت اسپریت بین دو فریم کلیدی به صورت عادی یا سریع صورت می‌گیرد
 ج) ابتدا و انتهای مسیرهای بسته مانند مسیر دایره‌ای را به صورت نرم به یکدیگر وصل می‌کند.
 د) افزایش انحنا می‌سیر

۸- جهت ساخت انیمیشن فریم به فریم پس از انتخاب عناصر در Cast از فرمان..... استفاده می‌شود؟

- الف) Modify → RealTime Recording
 ب) Control → Cast to time
 ج) Modify → Cast to time
 د) Control → RealTime Recording

۹- جهت معکوس کردن یک انیمیشن از کدام فرمان استفاده می‌شود؟

- الف) Edit → Reverse Sequence
 ب) Modify → Reverse Animation
 ج) Control → Reverse Sequence
 د) Modify → Reverse Sequence

۱۰- برای ثبت اثر حرکت یک اسپریت روی صحنه لازم است:

- الف) گزینه trails را غیر فعال کرد
 ب) گزینه trails را فعال کرد
 ج) گزینه Register Animation را فعال کرد
 د) گزینه Reverse Sequence را فعال کرد.

11-The Film Loops are useful for:

- a) animated sequence that you can use like a single cast member
 b) reduce the number of sprite channels you are using
 c) high speed animation
 d) A And B

کارگاه چندرسانه‌ای:

- ۱- نمایشی ایجاد کنید که در آن یک عکس با اندازه کوچک از سمت چپ به راست حرکت کرده تا وسط صحنه چهار بار جهت عقربه‌های ساعت چرخیده، به تدریج بزرگ شود؛ سپس در وسط صحنه توقف کرده، از آنجا به بعد تا خروج از صحنه چهار بار خلاف جهت عقربه‌های ساعت چرخیده و با کوچک شدن تدریجی از صحنه خارج شود. این کار را برای حداقل ۵ عکس تکرار کنید و یک Slide Show زیبا بسازید.
- ۲- به کمک انیمیشن فریم به فریم و معکوس کردن آن انیمیشن، باز و بسته شدن یک در را شبیه‌سازی کنید.
- ۳- به کمک روش ضبط زنده و ردپای انیمیشن نام محمد را در صحنه بنویسید.
- ۴- یک برنامه آموزشی برای یکی از دروس پایه ابتدایی مانند ریاضی بسازید. برای این کار از انیمیشن‌های جذابی که می‌توانید به اجزای نمایش خود اضافه کنید، کمک بگیرید تا نمایش شما برای کودکان جذاب شود.



فصل هفتم

رفتارها در دایرکتور

هدفهای رفتاری

- در پایان این فصل از هنرجو انتظار می‌رود:
- رفتار را تعریف کرده، انواع روش‌های ساخت آن را تشریح کند.
- رفتارهایی را ایجاد کرده آنها را به اسپرایت نسبت دهد.
- ترتیب اجرای چند رفتار نسبت داده شده به اسپرایت را توضیح داده و بتواند این ترتیب را تغییر دهد.
- انواع رویدادها و دستورات موجود در پنجره Behavior Inspector را توضیح دهد.

مقدمه:

از قابلیت‌های بسیار مهم و کاربردی دایرکتور در ساخت چند رسانه‌ای‌ها، امکان تعریف رفتارها (Behaviors) برای هر یک از اجزای نمایش است تا کاربر بتواند نمایش‌های محاوره‌ای (Interactive) ایجاد کرده، آنها را ارتقاء دهد. به‌طور کلی رفتار، عکس‌العملی (Action) است که یک شیء (Sprite) در برابر یک رویداد (Event) از خود نشان می‌دهد. به‌عنوان مثال شما منویی را جهت دسترسی به قسمت‌های مختلف نمایش طراحی کرده‌اید که در حالت معمول هیچ عملی را انجام نمی‌دهد. در حالی که شما می‌توانید برای هر یک از دکمه‌های موجود در منو، رفتاری را تعریف کنید تا در صورت کلیک نیز، آن قسمت از نمایش پخش شود. در این حالت نمایش طراحی شده، قابلیت غیر خطی پیدا کرده، ضمن اینکه امکان تعامل و انتخاب کاربر در اجرای گزینه یا گزینه‌های مورد نظر فراهم گردیده است. ما در این فصل قصد داریم به بررسی امکانات تعاملی نرم‌افزار Director و نحوه استفاده از آن‌ها در ساخت چند رسانه‌ای‌ها بپردازیم.

۱-۲- روش‌های کلی ایجاد یک رفتار:

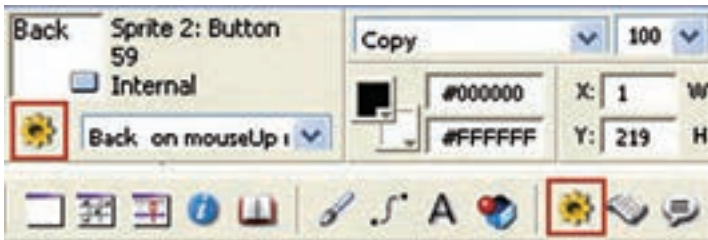
همانطور که در قسمت قبل گفتیم به عکس‌العملی که اسپریت‌ها در مقابل رویدادهای مختلف از خود نشان می‌دهند، یک رفتار گفته می‌شود، برای ایجاد رفتارها در دایرکتور معمولاً از سه روش زیر استفاده می‌شود:

- ۱- ابزار بازیبن رفتار (Behavior Inspector)
- ۲- کتابخانه رفتارهای آماده (Library Palette)
- ۳- زبان اسکریپت نویسی دایرکتور (Lingo)

۲-۲- نحوه تعریف یک رفتار با استفاده از ابزار بازیبن رفتار (Behavior Inspector)

اگر توجه کرده باشید دکمه‌ای به نام Behavior Inspector (شکل ۱-۲) در دو محل یافت می‌شود:

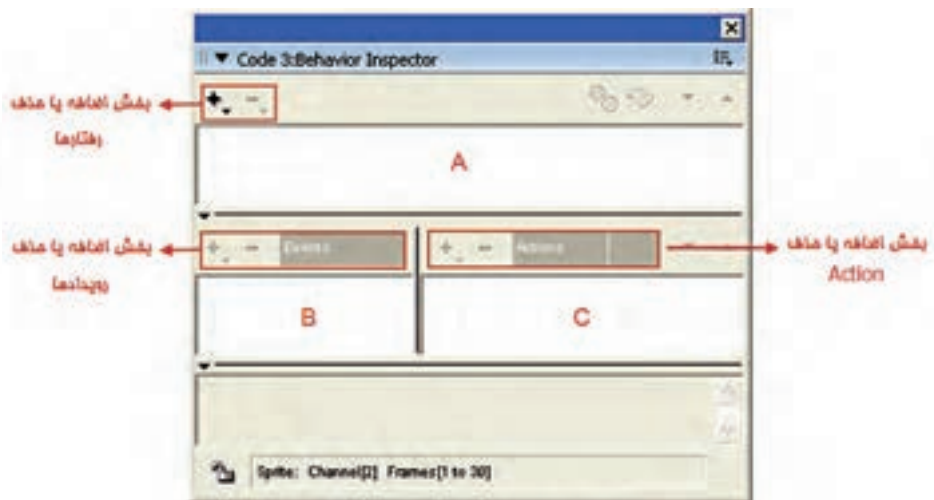
یکی در نوار ابزار دایرکتور و دیگری در نوار ابزار اسپریت موجود در Score



شکل ۱-۲ دکمه‌های باز کردن پنجره Behavior Inspector

نقطه آغاز کار، کلیک بر روی یکی از این دو دکمه است که باعث باز شدن پنجره بازبین رفتار (Behavior Inspector) می‌گردد و شما توسط این پنجره به آسانی می‌توانید رفتار خود را فقط با چند کلیک ماوس تعریف کنید.

نکته: دکمه Behavior Inspector موجود در نوار ابزار دایرکتور، یک رفتار مستقل را در پنجره Cast ایجاد می‌کند که بعد از تعریف می‌توانید آن را گرفته، به سمت اسپریت مورد نظر خود درگ کنید تا این رفتار به اسپریت نسبت داده شود؛ اما دکمه Behavior Inspector موجود در نوار ابزار اسپریت، علاوه بر اینکه نسخه اصلی رفتار را در پنجره Cast ایجاد می‌کند، آن را به اسپریت‌های انتخاب شده نیز نسبت می‌دهد بنابراین دیگر نیازی نیست به روش درگ کردن این رفتار را به یک یا چند اسپریت نسبت دهید، بلکه کافی است ابتدا اسپریت‌های خود را انتخاب و سپس روی این دکمه کلیک کنید. به هر حال با انتخاب یکی از روش‌های فوق پنجره Behavior Inspector مطابق شکل ۷-۲ باز می‌شود:



شکل ۷-۲ Behavior Inspector

همانطور که در پنجره بازبین رفتار مشاهده می‌کنید، سه بخش اصلی زیر وجود دارد:

- ۱- بخش A که به منظور اضافه یا حذف نام یک رفتار به کار می‌رود. ضمن اینکه لیست رفتارهای نسبت داده شده به اسپریت را نیز نمایش می‌دهد.
- ۲- بخش B که با آن می‌توان برای رفتار مورد نظر رویداد (Event) دلخواهی را تعریف کرد. همانطور که می‌دانید رویدادها اتفاقاتی هستند که نتیجه انجام آنها، شکل‌گیری رفتار یا رفتارهای خاصی است.

۳- بخش C که شامل یک سری رفتارهای آماده به نام Actions می‌باشد که با توجه به نوع عملکردشان دسته بندی شده اند. در ادامه بیشتر در مورد آنها صحبت خواهیم کرد.
نکته: اگر قسمتهای پایین پنجره Behavior Inspector را مشاهده نمی‌کنید، مطابق شکل ۷-۳ با کلیک روی مثلث کوچک آنها را باز کنید.



شکل ۷-۳ باز کردن بخش پایین پنجره Behavior

۷-۳- روش کلی تعریف رفتارها:

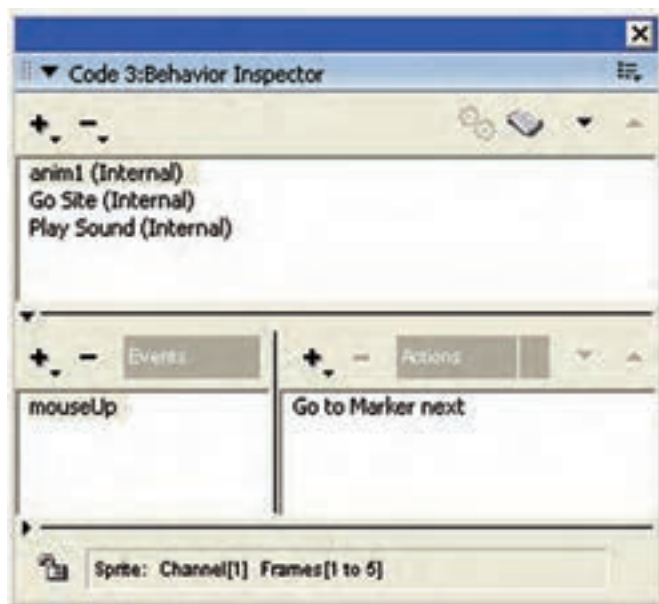
۱- منوی اصلی تعریف رفتار جدید را باز کرده، روی گزینه New Behavior... کلیک کنید و نامی را به آن اختصاص دهید.
 این نامی است که رفتار با آن شناخته شده، در Cast ظاهر می‌شود. (شکل ۷-۴)



شکل ۷-۴ ایجاد یک رفتار جدید

۲- از بخش رویدادها (Events)، رویداد یا رویدادهای مورد نظر خود را با کلیک روی دکمه + و منوی آن برگزینید. (یک رفتار می‌تواند دارای چندین رویداد باشد).

- ۳- اگر چند رویداد تعریف کرده‌اید، ابتدا یکی از آنها را انتخاب کرده، از بخش Action دستور یا دستوراتی را که می‌خواهید در صورت اتفاق افتادن آن رویداد اجرا شوند، انتخاب کنید. (منوی Action با کلیک بر روی دکمه + باز می‌شود)
- ۴- مراحل ۱ تا ۳ را برای سایر رفتارها تکرار کنید. همانطور که در شکل ۵-۷ مشاهده می‌کنید، پنجره Behavior Inspector سه رفتار ایجاد شده برای یک اسپریت را در لیست خود نمایش داده است.

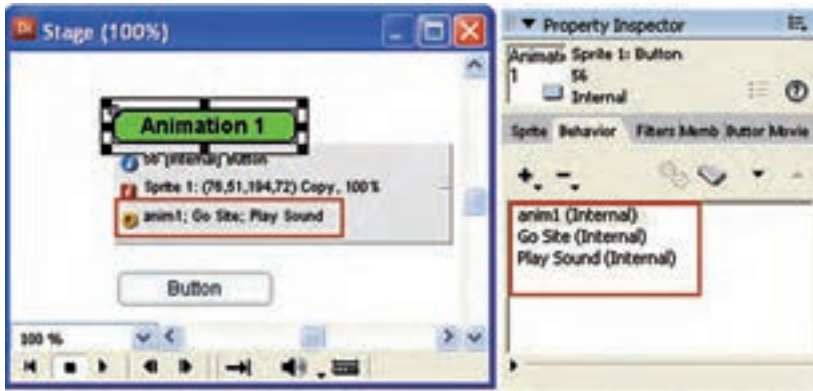


شکل ۵-۷ لیست رفتارهای ایجاد شده

۴-۷- اصول نسبت دادن رفتارها به اسپریتها:

ابتدا بهتر است مشخص کنیم، آیا به یک اسپریت، رفتارهایی نسبت داده شده است یا خیر؟ برای پاسخ به این سوال لازم است اسپریت مورد نظر خود را انتخاب و یکی از روش‌های زیر را برای مشاهده رفتارها روی آن انجام دهید:

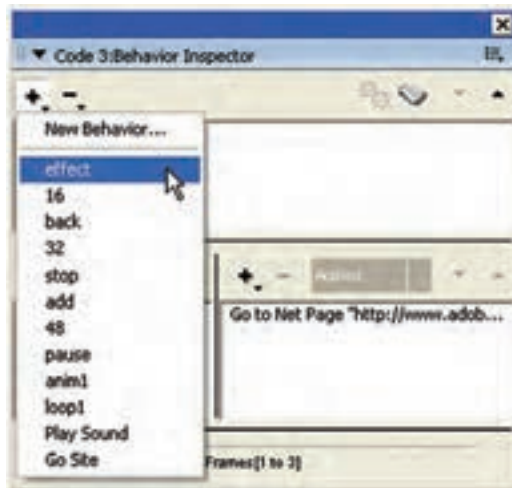
- روش اول: مشاهده لیست رفتارهای موجود در Behavior Inspector (شکل ۷-۵)
- روش دوم: مشاهده لیست رفتارهای موجود در زبانه Behavior از پنجره Property Inspector (شکل ۶-۷ سمت راست)
- روش سوم: بررسی لیست رفتارهای موجود در Sprite Overlay (شکل ۶-۷ سمت چپ)



شکل ۶-۷ مشاهده رفتارهای نسبت داده شده به اسپریت

بعد از اینکه مطمئن شدید رفتاری به اسپریت نسبت داده شده است یا خیر، می‌توانید به یکی از روشهای زیر رفتارهای دلخواه خود را به یک یا چند اسپریت انتخاب شده اضافه کنید:

روش اول: منوی اصلی اضافه کردن رفتار موجود در Behavior Inspector را باز کرده، رفتار خود را از میان کلیه رفتارهای ایجاد شده در نمایش انتخاب کنید تا به لیست اضافه شود. (شکل ۷-۷)



شکل ۷-۷ اضافه کردن یک رفتار آماده از طریق پنجره Behavior

روش دوم: باز کردن منوی اصلی اضافه کردن رفتار و انتخاب رفتار مورد نظر از طریق زبانه Behavior از پنجره Property Inspector (شکل ۸-۷)



شکل ۸-۷ اضافه کردن یک رفتار آماده از طریق زبانه Behavior

روش سوم: درگ کردن رفتار از داخل پنجره Cast بر روی اسپریت (شکل ۹-۷)

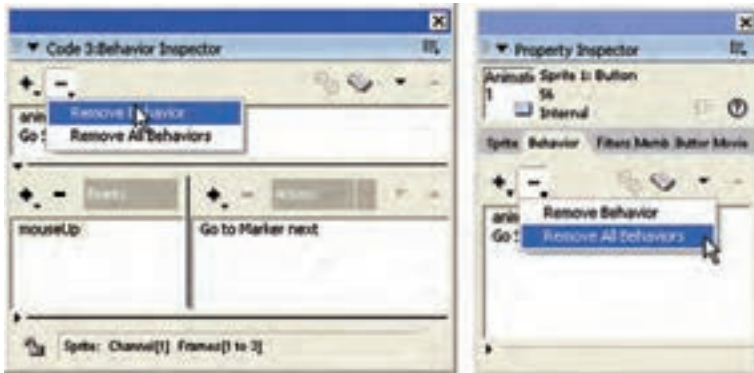


شکل ۹-۷ اضافه کردن رفتار به روش درگ کردن

۷-۵- اصول حذف یک رفتار از اسپریت

به یکی از دو روش زیر می‌توانید رفتار نسبت داده شده به اسپریت را از آن جدا کنید روش اول: با استفاده از پنجره Behavior Inspector کافی است رفتار مورد نظر را انتخاب، سپس روی دکمه Clear Behavior کلیک کنید تا منوی آن باز شود، یکی از دو فرمان Remove Behavior برای حذف رفتار انتخاب کنید و گزینه Remove All Behaviors را جهت حذف کلیه رفتارها برگزینید. (شکل ۱۰-۷ سمت چپ)

روش دوم: با استفاده از زبانه Behavior موجود در Property Inspector، که در این قسمت نیز روش حذف رفتار، مشابه روش اول می‌باشد. (شکل ۷-۱۰ سمت راست)



شکل ۷-۱۰ روشهای حذف رفتار نسبت داده شده

۷-۶- اصول ترتیب رفتارهای نسبت داده شده به اسپریت:

دایرکتور رفتارها را به همان ترتیبی که به اسپریت نسبت داده شده‌اند در Behavior Inspector و زبانه Behavior از Property Inspector لیست می‌کند؛ اما در صورت نیاز می‌توانید به آسانی این ترتیب را تغییر دهید:

- ۱- اسپریت مورد نظر را انتخاب کنید.
- ۲- پنجره Behavior Inspector یا Property Inspector را باز کنید (زبانه Behavior).
- ۳- رفتاری را که می‌خواهید آن را جابجا کنید، انتخاب کنید.
- ۴- از دکمه‌های جهت‌دار موجود در نوار ابزار این پنجره‌ها، جهت جابجایی رفتار به سمت بالا یا پایین کمک بگیرید. (شکل ۷-۱۱)



شکل ۷-۱۱ دکمه‌های تغییر ترتیب اجرای رفتارها

۷-۷- انواع رویدادهای موجود در بخش Events

همانطور که گفتیم رویدادها به اتفاقات موجود در یک پروژه گفته می‌شود که در نتیجه انجام آنها یک یا چند رفتار اجرا می‌گردند. بر این اساس، با توجه به اهمیتی که رویدادها در اجرای فرایند یک برنامه دارند، در این قسمت به بررسی اختصاصی ۵ گروه از رویدادهای موجود در دایرکتور می‌پردازیم:

- ۱- رویدادهای مربوط به دکمه سمت چپ و اشاره گر ماوس:
Mouse Up: رویداد مربوط به رها شدن دکمه ماوس از حالت فشرده (کلیک کامل)
Mouse Down: رویداد مربوط به فشرده شدن دکمه ماوس
Mouse Enter: زمانی که اشاره گر ماوس روی یک شیء قرار گیرد، این رویداد فقط یکبار اتفاق می‌افتد.
Mouse Within: زمانی که اشاره گر ماوس در محدوده یک شیء قرار گیرد، این رویداد اتفاق می‌افتد. نکته قابل توجه این است که این رویداد تا زمانی که ماوس در محدوده شیء قرار دارد، دائماً اتفاق می‌افتد.
Mouse Leave: هنگامی که اشاره گر ماوس محدوده شیء را ترک کند، این رویداد اتفاق می‌افتد.
- ۲- رویدادهای مربوط به دکمه سمت راست ماوس:
Right MouseUp: این رویداد زمانی اتفاق می‌افتد که دکمه سمت راست ماوس از حالت فشرده رها شود
Right Mouse Down: این رویداد زمان فشرده شدن دکمه سمت راست ماوس اتفاق می‌افتد
- ۳- رویدادهای صفحه کلید:
KeyUp: هریک از کلیدهای صفحه کلید به جز کلیدهای کنترلی مانند Shift وقتی از حالت فشرده رها شوند، این رویداد اتفاق می‌افتد.
KeyDown: هریک از کلیدهای صفحه کلید به جز کلیدهای کنترلی مانند Shift وقتی فشرده شوند، این رویداد اتفاق می‌افتد.
- ۴- رویدادهای مربوط به فریم:
PrepareFrame: این رویداد زمانی اتفاق می‌افتد که هد در یک فریم دلخواه قرار گیرد.
ExitFrame: این رویداد زمانی اتفاق می‌افتد که هد از یک فریم دلخواه عبور کند.
- ۵- رویدادهای مربوط به اسپریت:
BeginSprite: این رویداد زمانی اتفاق می‌افتد که هد در اولین فریم یک اسپریت قرار گیرد.
EndSprite: این رویداد زمانی اتفاق می‌افتد که هد در آخرین فریم یک اسپریت قرار گیرد.

۸-۷- انواع Action های موجود در پنجره Inspector Behavior:

همانطور که گفتیم Action به مجموعه رفتارهای آماده‌ای گفته می‌شود که در نتیجه انجام یک رویداد اجرا می‌شوند، بر این اساس می‌توان در پنجره Behavior Inspector از ۶ گروه اصلی آنها استفاده کرد:

۱- Navigation: یکی از پرکاربردترین Action های دایرکتور بوده، که از این گروه برای هدایت

کردن نمایش و هد استفاده می‌شود و شامل Action های زیر می‌باشند:

Go to Frame: هد را به فریم مشخص شده‌ای هدایت می‌کند.

Go to Marker: هد را به Marker مشخص شده‌ای هدایت می‌کند. Marker ها در حقیقت

نشانگرهایی هستند که از آنها برای انتقال هد به بخش خاصی از یک پروژه استفاده

می‌شود. در مورد Marker ها و کاربرد آنها در ادامه بیشتر صحبت خواهیم کرد.

Go to Movie: هد را به نمایش (فیلم) مشخص شده‌ای برای پخش آن هدایت می‌کند،

این فیلم باید یک نمایش دایرکتور با پسوند Dir باشد که در مسیر مشخصی ذخیره شده

است در این حالت نیازی به وارد کردن یا Import این فیلم نمی‌باشد.

Go to Net Page: صفحه وب مشخص شده‌ای را در مرورگر ویندوز باز می‌کند.

Exit: موجب خاتمه پخش نمایش جاری می‌گردد.

۲- Wait: این دسته از Action ها مربوط به توقف نمایش بوده، شامل گزینه‌های

زیر می‌باشند:

On Current Frame: هد را در فریم جاری نگه می‌دارد.

Until Click or KeyPress: هد را آنقدر نگه می‌دارد تا کاربر کلیک کرده یا کلیدی را از

صفحه کلید فشار دهد.

For Time Duration: به اندازه زمان تعیین شده که واحد آن ثانیه می‌باشد، هد را نگه

می‌دارد. مثلاً شما می‌توانید در فریم مشخصی هد را مدت زمان مشخصی متوقف کنید.

پس از اتمام این زمان هد به حرکت عادی خود ادامه می‌دهد. البته توجه داشته باشید که

حداکثر زمان توقف هد ۶۰ ثانیه معادل یک دقیقه می‌باشد.

۳- Sound: این گروه شامل دستوراتی جهت کنترل و اجرای اصوات موجود در یک پروژه

می‌باشد که شامل موارد زیر است:

Play Cast Member: با انتخاب این گزینه لیست فایل‌های صوتی که به داخل پنجره cast

وارد (Import) کرده‌اید، ظاهر شده و شما می‌توانید فایل خود را جهت پخش انتخاب

کنید. مثلاً می‌توانید یک رفتار طراحی کنید، به طوری که در صورت قرار گرفتن ماوس

روی گزینه‌های منو، صدای کوتاهی نیز پخش شود.

Play External File: انتخاب این گزینه باعث نمایش کادری می‌شود که شما می‌توانید

مسیر و نام فایل صوتی را که در خارج از محیط دایرکتور قرار دارد مشخص کنید تا

پخش شود.

Beep: موجب پخش صدای بیپ می‌شود.

Set Volume: جهت تنظیم حجم صدا استفاده می‌شود.

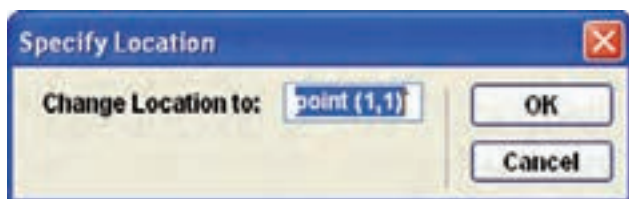
۴- Frame: شامل دستوراتی جهت کنترل رفتار در یک فریم می‌باشد که به شرح زیر است:
Change Tempo: به شما این گزینه می‌توان سرعت هد را در یک فریم خاص تغییر داد.
به‌عنوان نمونه می‌توان یک رفتار را به صورتی طراحی کرد که در صورت کلیک روی دکمه‌ای خاص سرعت نمایش کم یا زیاد شود.

Perform Transition: به‌وسیله این گزینه می‌توان یک جلوه انتقال را برای یک رویداد مشخص تعیین نمود به‌عنوان نمونه می‌توان یک رفتار را به صورتی طراحی کرد که در صورت رسیدن هد به اولین فریم، یک اسپریت (رویداد Begin Sprite)، جلوه انتقالی خاصی اجرا شود.

Change Palette: با این گزینه می‌توان سیستم رنگهای پیش فرض ویندوز را تغییر داد.
(سیستم پیش فرض رنگها System Win می‌باشد).

۵- Sprite: این گروه دارای دستوراتی جهت کنترل اسپریت‌ها بوده، شامل گزینه‌های زیر می‌باشد:

Change Location: محل قرارگیری یک اسپریت را در Stage تغییر می‌دهد به این ترتیب که پس از کلیک روی این گزینه، کادر محاوره‌ای Specify Location مطابق شکل ۱۲-۷ ظاهر می‌شود. در این کادر مقابل گزینه Change Location to (گزینه Point) را مشاهده می‌کنید که نشان می‌دهد به‌طور پیش فرض مختصات X (طول) و Y (عرض) روی ۱ تنظیم شده است.



شکل ۱۲-۷ کادر محاوره‌ای Specify Location

Change Cast member: این گزینه جالب این امکان را در اختیار شما قرار می‌دهد که بتوانید یک Cast Member موجود در صحنه را با یک Cast Member دیگر که در پنجره Cast قرار دارد تعویض کنید. کاربرد این گزینه در طراحی دکمه‌های چندحالتی است که در مثالهای بعد از آن استفاده خواهیم کرد.

Change Ink: این گزینه می‌تواند جلوه جوهری (Ink) یک اسپریت را تغییر دهد.
Cursor: شامل دو گزینه جهت تغییر شکل ماوس و برگرداندن آن به حالت عادی می‌باشد:

Change Cursor: این گزینه به منظور تغییر شکل ماوس استفاده می‌شود.
Restore Cursor: این گزینه شکل ماوس را به حالت عادی خود بر می‌گرداند.
حال که با انواع رفتارهای آماده در پنجره Behavior Inspector آشنا شدید برای این که به صورت کاربردی با هریک از آنها آشنا شوید در ادامه به ذکر چند مثال می‌پردازیم:
مثال ۱: در این مثال می‌خواهیم یک دکمه دو حالت را ایجاد کنیم. به طوری که در حالت Up و Over شکل آن تغییر کند.

۱- دو حالت دکمه خود را به دلخواه و به هر شکلی که مایلید طراحی کنید. برای اینکار می‌توانید از انواع نرم‌افزارهای گرافیکی مانند Photoshop استفاده کنید. با توجه به اینکه نرم‌افزار دایرکتور فرمت PSD و زمینه شفاف آنها را شناسایی می‌کند به راحتی وارد دایرکتور شده، می‌توانید دکمه‌ها یا تصاویری داشته باشید که زمینه آنها شفاف (Tran-parent) باشد، همانطور که می‌دانید در دکمه‌هایی که شکلی غیر چهارضلعی دارند، در صورتی که نرم‌افزار مقصد قادر به حذف زمینه آن نباشد، زمینه سفیدی در اطراف دکمه ظاهر می‌شود.

۲- اکنون دو حالت دکمه خود را وارد دایرکتور کنید (Import). در این مثال دو حالت دکمه در فتوشاپ ایجاد شده، به نامهای BTN1 و BTN2 ذخیره شده اند که اولی حالت برجسته دکمه و متن آن سفید رنگ است و دومی حالت فرو رفته آن و رنگ متن آن زرد است و هر دو دارای زمینه شفاف و هم اندازه‌اند.

۳- اکنون حالت اول دکمه خود را وارد صحنه کنید که در این مثال BTN1 می‌باشد (شکل ۱۳-۷).

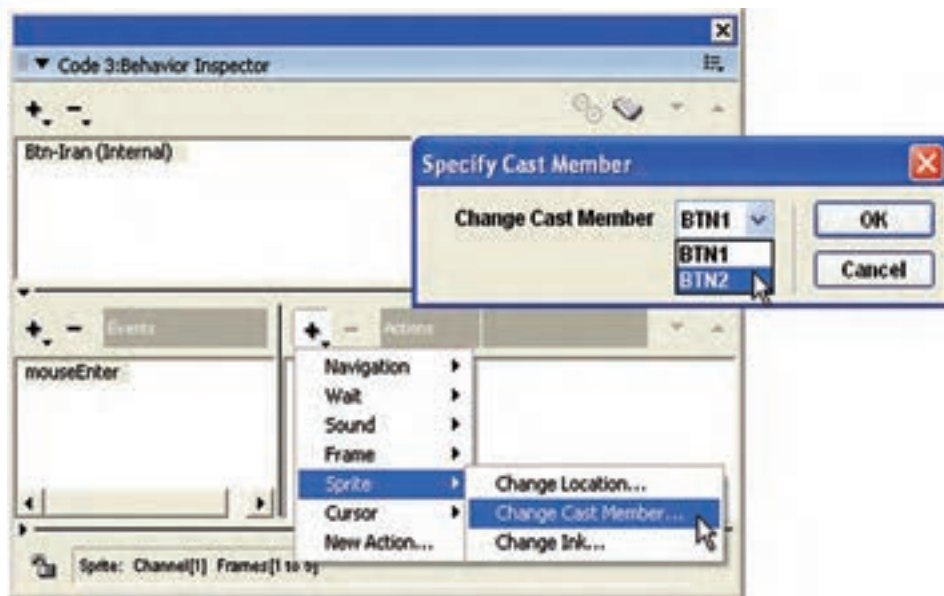


شکل ۱۳-۷

اکنون نوبت به طراحی رفتار مورد نظر برای این دکمه دو حالت می‌رسد که از رویدادهای MouseEnter و MouseLeave استفاده می‌شود، زمانیکه کاربر ماوس را در محدوده اسپریت روی دکمه قرار می‌دهد. رویداد اولی اتفاق افتاده و لازم است شکل دکمه از BTN1 به BTN2 عوض شود، ماوس از محدوده اسپریت خارج می‌شود، رویداد دوم اتفاق افتاده و لازم است دکمه از BTN2 به BTN1 وقتی که به حالت اولیه‌اش برگردد پس ابتدا اسپریت BTN1 را انتخاب کرده، سپس دکمه Behavior Inspector موجود در نوار ابزار اسپریت پنجره Score را کلیک کنید تا بازبین رفتار ظاهر شود.

یادآوری:

- همانطور که قبلاً گفته شد تفاوت این دکمه با دکمه مشابه در نوار ابزار اصلی دایرکتور آنست که این دکمه رفتار تعریف شده را به اسپریت‌های انتخاب شده نسبت می‌دهد.
- ۶- رفتار جدیدی با نام دلخواه مانند Btn-Iran ایجاد کنید.
 - ۷- از قسمت Events رویداد Mouse Enter را انتخاب کنید.
 - ۸- برای رویداد فوق از قسمت Actions از زیر مجموعه Sprite گزینه Change Cast ... Member را برگزینید تا کادر محاوره‌ای Specify Cast Member جهت انتخاب حالت دوم دکمه مطابق شکل باز شود، سپس از لیست آن گزینه BTN2 را برگزینید. (شکل ۱۴-۷)



شکل ۱۴-۷

- ۹- تا اینجا اگر نمایش را اجرا کرده، ماوس را روی دکمه ببرید، شکل آن عوض می‌شود، اما با کنار رفتن ماوس به حالت اولیه خود برنمی‌گردد، پس لازم رویداد دوم و Action آن را نیز تعریف کنیم .
- ۱۰- مراحل ۷ و ۸ را دوباره تکرار کنید. با این تفاوت که در مرحله ۷ رویداد Mouse Leave را برگزینید و در مرحله ۸ از کادر Specify Cast Member گزینه BTN1 را برگزینید.
- ۱۱- اکنون رفتار شما کامل است و می‌توانید آن را تست کنید و در صورت دلخواه نمایش خود را ذخیره کنید. (توصیه میشود قبل از تست کامل، فرمان توقف را در فریم آخر نمایش درج کنید تا هد در آنجا توقف کند).

مثال ۲:

در این مثال می‌خواهیم یک Slide Show با یک منوی ساده سه گزینه‌ای طراحی کنیم. گزینه‌های این منو سه عبارت متنی Nature و Flowers و Exit می‌باشند؛ اما شما می‌توانید به جای آنها از سه دکمه دلخواه که نحوه طراحی آنها در مثال قبلی ذکر شد، استفاده کنید تا نمایش زیباتر و جذاب‌تر شود.

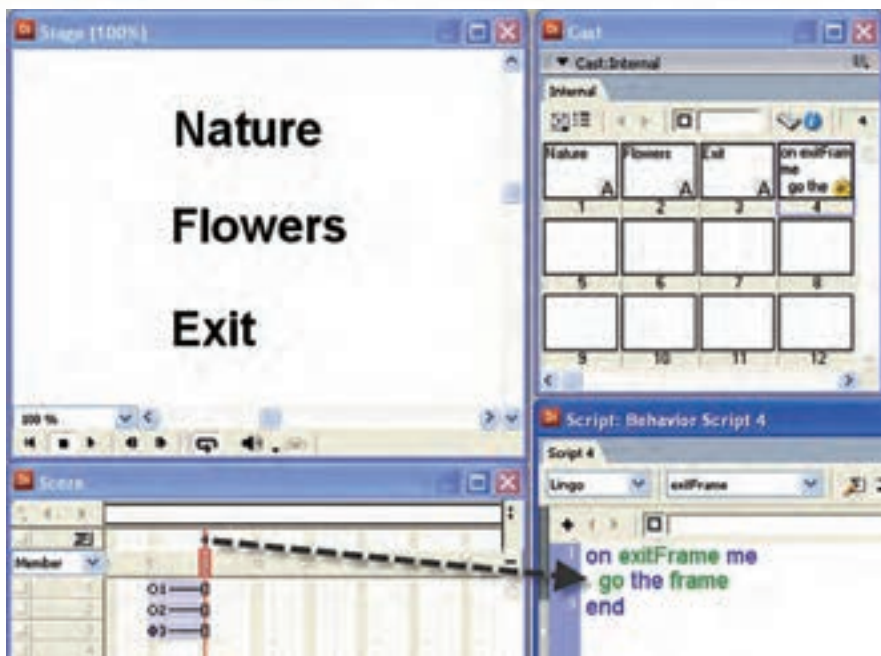
هنگامی که کاربر روی Nature کلیک می‌کند (رویداد MouseUp)، هد با فرمان Go to frame به فریمی منتقل می‌شود که اولین اسپرایت مربوط به تصاویر طبیعت از آنجا آغاز می‌شود. پس از پخش، این سه، تصویر، توقف می‌کند تا کاربر روی آخرین تصویر موجود در صحنه کلیک کند تا به منو برگردد.

هنگامیکه کاربر روی Flowers کلیک می‌کند (رویداد MouseUp) هد توسط فرمان Go to frame به فریمی منتقل می‌شود که اولین اسپرایت مربوط به تصاویر گلها از آنجا آغاز می‌شود. این سه تصویر دائم پخش می‌شوند تا کاربر روی یکی از آنها کلیک کند تا به منو برگردد.

و بالاخره اگر روی گزینه Exit کلیک کند، نمایش خاتمه می‌یابد.

مراحل انجام کار:

- ۱- سه متن Nature , Flowers و Exit را ایجاد کنید.
- ۲- تصاویر مربوط به طبیعت و تصاویر مربوط به گلها را وارد پنجره Cast کنید. (حداقل ۳ تصویر برای هر کدام)
- ۳- سه متن ایجاد شده را وارد صحنه کرده، آنها را مطابق شکل در فریم‌های ۵ تا ۱۰ کانالهای اول و دوم و سوم پنجره Score قرار دهید.
- ۴- اکنون دستور توقف را مطابق شکل ۱۵-۷ برای فریم ۱۰ تایپ کنید.



شکل ۷-۱۵ آماده کردن رفتار توقف

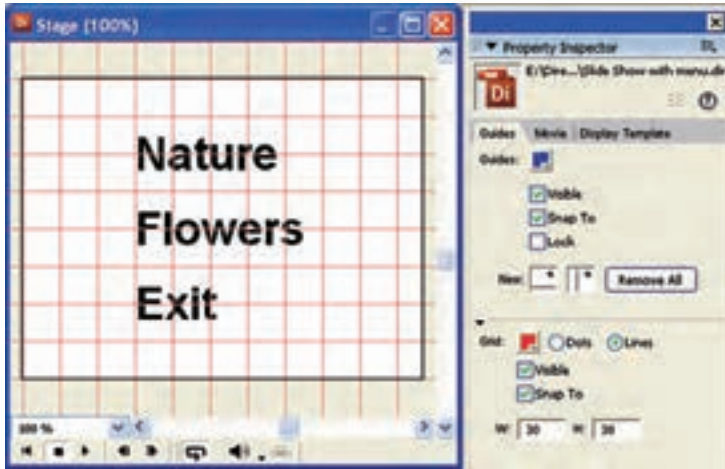
یادآوری: در فریم ۱۰ از کانال رفتار (محل مشخص شده با ماوس) دوبار کلیک کنید. سپس در رویداد مربوطه دستور مورد نظر را تایپ کنید. توجه داشته باشید که برای قرار دادن گزینه‌های منو به صورت دقیق، طوری که از سمت چپ تراز و فاصله بین آنها یکسان باشد، می‌توان از روش‌های زیر استفاده کرد:

روش اول:

استفاده از خطوط شبکه بندی (راهنما) (Grid) و خاصیت Snap to Grid که برای این منظور کافی است:

الف - در قسمت خالی از Stage کلیک راست کرده و گزینه Movie Properties را برگزینید تا کادر محاوره‌ای Property Inspector باز شود.

ب- زبانه Guides and Grids را فعال کرده، در بخش Grid (راهنما) هر دو گزینه Visible و Snap to Grid را علامت دار کنید و مقدار ۳۰ را در هر دو کادر مربوط به فواصل افقی و عمودی خطوط وارد کنید. سپس متون را با توجه به این خطوط ترازبندی کنید. (شکل ۷-۱۶)



شکل ۱۶-۷ ترازبندی به کمک Grid

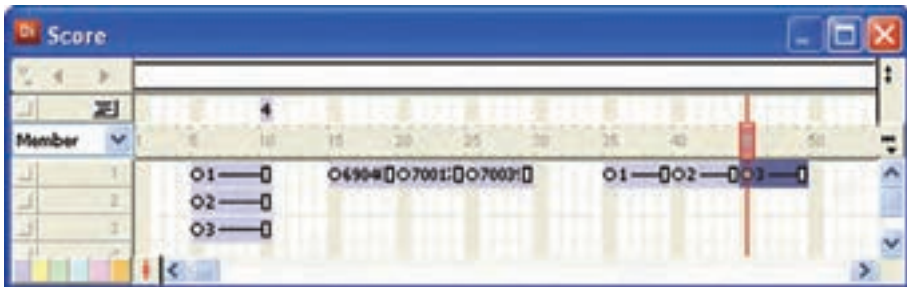
Snap to Grid خاصیت جالبی است که باعث می‌شود در صورت نزدیک شدن اسپرایت یا متن‌ها به نقاط تلاقی خطوط شبکه، اسپرایت به این نقطه بچسبد و به این ترتیب از سمت چپ یا بالا به راحتی قابل ترازبندی است.

روش دوم:

در این روش می‌توانید با مختصات X و Y موجود در نوار ابزار اسپرایت، آنها را ترازبندی و یا فاصله آنها را میزان کنید فقط کافی است هر سه متن را (به کمک Shift) انتخاب و مقدار دلخواه را در کادر X، جهت تراز شدن آنها از سمت چپ وارد کنید.

همچنین با مختصات Y امکان تنظیم فاصله عمودی آنها از یکدیگر وجود دارد.

۵- حال نوبت به وارد کردن تصاویر به داخل صحنه می‌باشد. پس تصاویر طبیعت را پشت سر هم از فریم ۱۵ تا ۲۹ و تصاویر گلها را از فریم ۳۵ تا ۴۹ مطابق شکل ۱۶-۷ در کانال اول Score بچینید (برای هر تصویر ۵ فریم در نظر بگیرید).



شکل ۱۷-۷ تعیین مقصد برای منو

۶- قصد داریم رفتاری به نمایش اضافه کنیم، به طوری که تصاویر مربوط به طبیعت، دائم پخش شوند. بنابراین باید دستوری بنویسید که هد، پس از رسیدن به فریم ۴۹، دوباره به فریم ۳۵ برگردد تا تصاویر دوباره پخش شوند. این دستور بسیار ساده `go frame 15` می باشد که باید پس از دو بار کلیک در فریم ۴۹ کانال رفتار نوشته شود. نام این رفتار را Loop Nature بگذارید. (شکل ۱۸-۷)



شکل ۱۸-۷ طراحی حلقه پخش

- ۷- به طور مشابه رفتاری برای تصاویر گلها بنویسید که آنها نیز دائم پخش شوند.
- ۸- حال می خواهیم برای اولین گزینه منوی خود اولین رفتار را طراحی کنیم. برای اینکار ابتدا متن Nature را انتخاب کرده، روی دکمه Behavior Inspector موجود در نوار ابزار اسپریت کلیک کنید تا Behavior Inspector باز شود.
- ۹- رفتار جدیدی با نام Play Nature ایجاد کنید.
- ۱۰- رویداد این رفتار را Mouse Up در نظر گرفته، از زیر منوی Navigation موجود در منوی Actions، روی گزینه Go to Frame کلیک کنید تا کادر دریافت شماره فریم باز شده، سپس در این کادر عدد ۱۵ را وارد کنید.
- ۱۱- به طریق مشابه رفتاری به نام Play Flowers را برای گزینه دوم منو تعریف کنید، با این تفاوت که شماره فریم را ۳۵ در نظر بگیرید.
- ۱۳- تا این مرحله اگر نمایش را اجرا کنید، خواهید دید که با کلیک روی گزینه Nature، هد به فریم ۱۵ منتقل شده، پخش تصاویر به صورت یک حلقه تکرار می شود، اما هنوز رفتاری برای برگشت به منو طراحی نکرده ایم. پس هد را به قسمت منو برده، گزینه دوم منو (Flowers) را امتحان کنید. با کلیک روی این گزینه هد به فریم ۳۵ منتقل شده، تصاویر گلها را دائم پخش می کند که برای برگشت از این قسمت نیز لازم است یک رفتار طراحی کرد.
- ۱۴- جهت برگشت به منو از تصاویر کمک گرفته، رفتاری برای آنها آماده کنید که در صورت کلیک بر روی هر یک از تصاویر، هد به منو (فریم ۵) بازگشت کند. (یعنی تصاویر مانند دکمه عمل می کنند). بنابراین هر شش اسپریت مربوط به تصاویر را در Score انتخاب کرده، رفتار Back را جهت برگشت به منو تعریف کنید و نمایش را تست کنید.
- ۱۵- در پایان رفتار، خروج از نمایش را برای گزینه Exit منو تعریف کنید. (راهنمایی: از بخش Navigation و از فرمان Exit استفاده کنید).

۷-۹- مارکر و مزایای آن (Marker)

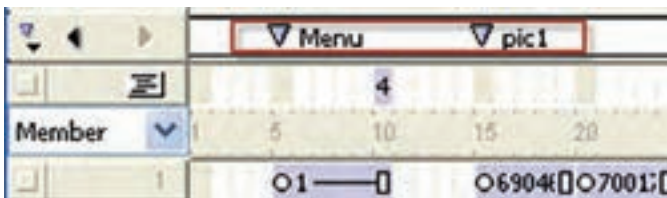
مارکر در واقع برچسبی برای مشخص نمودن یک فریم خاص است و می‌تواند در انتقال هد به آن فریم، کمک شایانی کند. به این ترتیب که به‌جای استفاده از دستور Go to Frame که نیاز به شماره فریم دارد، از دستور Go to Marker استفاده می‌کنیم و نام مارکر مورد نظر را برای آن مشخص می‌کنیم، که منجر به طراحی آسان تر یک رفتار خواهد شد.

از مزایای مارکرها می‌توان به موارد زیر اشاره کرد:

- ۱- مشخص شدن یک فریم خاص به کمک یک نام
- ۲- امکان جابجا کردن یک مارکر روی فریم‌ها در زمان طراحی نمایش
- ۳- بالا رفتن سرعت طراحی رفتارها و خواناتر شدن آنها

۷-۹-۱- چگونگی تعریف یک مارکر:

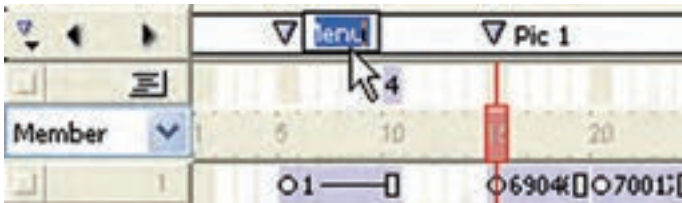
دایرکتور در پنجره Score یک کانال مخصوص استفاده از مارکرها پیش بینی کرده است که کار با آن بسیار ساده است. این کانال در بالای کانالهای افکت قرار دارد. جهت تعریف یک مارکر کافی است در فریم خاصی از این کانال کلیک کنید تا یک مارکر با نام پیش فرض New Marker ایجاد شود، سپس می‌توانید نام آن را به دلخواه تغییر دهید. در شکل ۷-۱۹ دو مارکر به نام‌های Menu و Pic1 در فریمهای ۵ و ۱۵ درج شده است.



شکل ۷-۱۹ دو مارکر درج شده در کانال مارکر

۷-۹-۲- روش تغییر نام یک مارکر:

برای این منظور کافی است روی آن در کانال Marker کلیک کنید تا وارد حالت ویرایش شده، نام جدید را تایپ و در انتها Enter کنید. (شکل ۷-۲۰)



شکل ۷-۲۰ تغییر نام یک مارکر

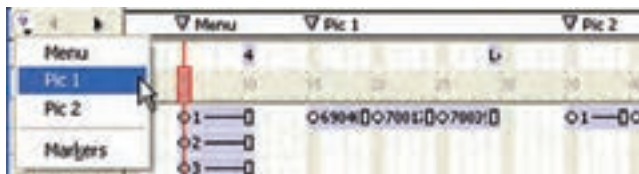
۷-۹-۳- روش حذف یک مارکر:

این کار نیز ساده است، کافی است ماوس را روی علامت مثلث شکل مارکر ببرید تا شکل درست شود، سپس آن را به داخل پنجره Score و به سمت پایین درگ کنید.

۷-۹-۴- انتقال سریع هد بر روی مارکرهای موجود:

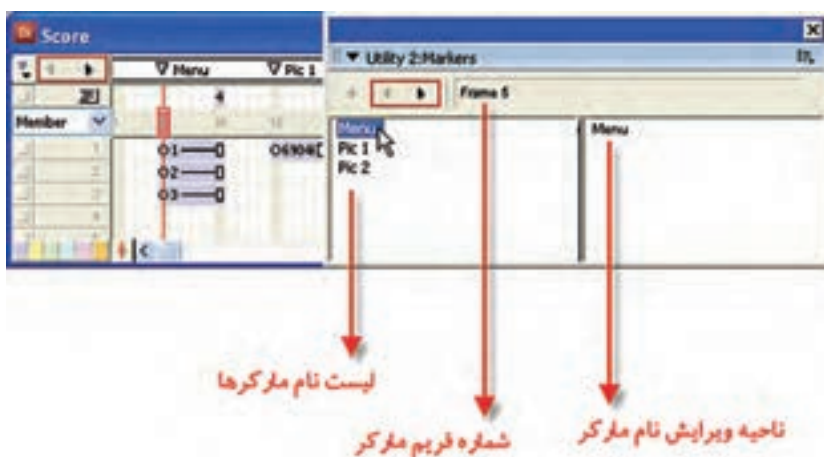
در سمت چپ کانال مارکر سه دکمه وجود دارد که امکان حرکت سریع هد روی مارکرها را فراهم می‌آورد که به شرح آنها می‌پردازیم:

دکمه Markers Menu: باعث باز شدن منویی می‌شود که تمامی مارکرهای موجود در آن به ترتیب لیست شده است، با کلیک روی هر کدام، هد به آنجا منتقل می‌شود. شکل (۷-۲۱)



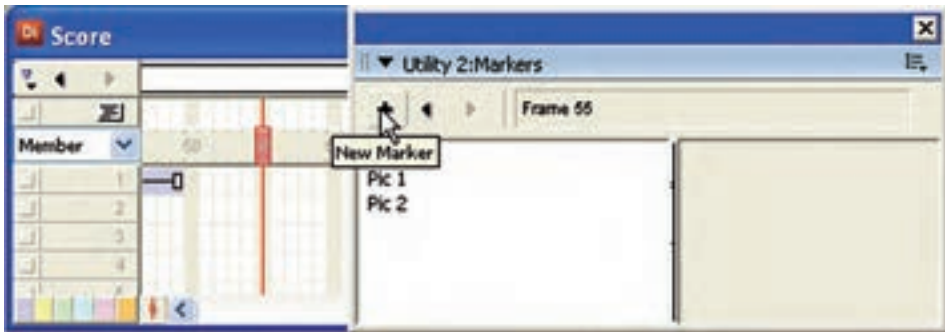
شکل ۷-۲۱

آخرین گزینه این منو Markers می‌باشد که موجب باز شدن پنجره Markers می‌گردد. در این پنجره نیز می‌توانید مارکرها را ویرایش کنید (شکل ۷-۲۲) توسط دکمه Previous Marker می‌توان هد را به مارکر قبلی و به کمک دکمه Next Marker می‌توان هد را به مارکر بعدی منتقل کرد. این دو دکمه در پنجره Markers نیز وجود دارند. (شکل ۷-۲۲)



شکل ۷-۲۲ ویرایش مارکرها

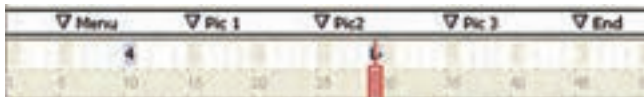
نکته: در پنجره Markers نیز می‌توانید یک Marker جدید ایجاد کنید. کافی است هد را به فریم مورد نظر نباید دارای Marker باشد در Score انتقال داده سپس روی دکمه New Marker کلیک کنید تا Marker جدیدی بنام New Marker به لیست Markerها اضافه شود، در انتها نام آن را تغییر دهید. (شکل ۲۳-۷)



شکل ۲۳-۷ اضافه کردن مارکر توسط پنجره Markers

۵-۹-۷- Markerهای نسبی:

در دایرکتور می‌توان از Markerها به شکل نسبی نیز استفاده کرد که در بسیاری از موارد موجب ساده تر شدن طراحی رفتارها می‌گردد. برای درک مفهوم Markerهای نسبی از شکل ۲۴-۷ کمک می‌گیریم:



شکل ۲۴-۷ وضعیت قرارگیری هد نسبت به مارکرها

به مارکری که هد به تازگی از آن عبور کرده، Marker جاری (Pic2) می‌گویند که برای برگشت به این Marker از فرمان go loop یا تابع go loop() می‌توان استفاده کرد. Marker بعدی Pic3 می‌باشد که برای انتقال هد به آن، از فرمان Go Next() و یا تابع Go Next() و سرانجام برای انتقال هد به Marker قبلی (Pic1) از فرمان Go Previous() یا تابع Go Previous() استفاده می‌شود.

نکته: جهت حرکت نسبی از طریق پنجره Behavior Inspector کافیست پس از انتخاب فرمان Go Marker از مجموعه Navigation و باز شدن لیست Markerها، یکی از گزینه‌های زیر را جهت حرکت نسبی انتخاب کنید:
 Loop: حرکت به Marker جاری

Previous: حرکت به Marker قبلی

Next: حرکت به Marker بعدی

نکته: در لینگو، زبان اسکریپت‌نویسی دایرکتور، فرمانی به شکل کلی (n) go marker وجود دارد که به شما این امکان را می‌دهد به تعداد دلخواه (n)، به سمت جلو یا عقب روی Markerها حرکت کنید. چنانچه به n مقدار صفر بدهید، انتقال به Marker جاری، چنانچه $n > 0$ باشد انتقال به n Marker بعدی و بالاخره اگر $n < 0$ باشد انتقال به n قبلی صورت می‌گیرد. به مثالهای زیر توجه کنید:

مثال ۱: Go marker (0) موجب انتقال به Marker جاری می‌شود.

مثال ۲: Go marker (3) موجب انتقال به Marker 3 بعدی می‌شود.

مثال ۳: Go marker (-2) موجب انتقال به Marker 2 قبلی می‌شود.

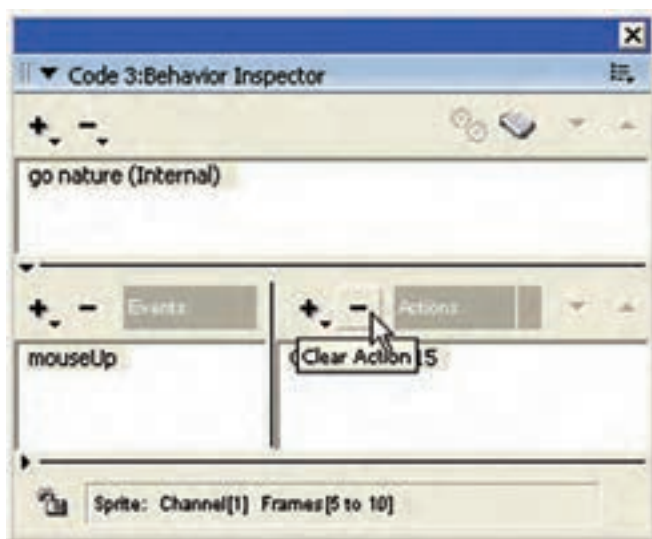
مثال ۴: در این مثال می‌خواهیم یکی از رفتارهای مثال قبلی (مثال ۳) را ویرایش کنیم و در آن از مارکرها استفاده کنیم.
مراحل انجام کار:

۱- نمایش مثال ۳ را باز کنید، سپس برای فریم ۱۵ آن یک مارکر به نام Pic1 اضافه کنید.

۲- اولین گزینه منوی خود را انتخاب کنید (Nature)

۳- پنجره Behavior Inspector را باز کنید.

۴- از بخش اکشن‌ها دستور Go to frame 15 را انتخاب و با کلیک روی دکمه Clear Action آنرا حذف کنید. (شکل ۲۵-۷)



شکل ۲۵-۷

۵- اکنون منوی اکشنها را باز کرده یا از زیر منوی Navigation، گزینه Go to Marker را انتخاب کنید تا لیست مارکرها باز شود، از لیست مارکر Pic1 را انتخاب و آن را تایید کنید.

۶- نمایش را ذخیره کرده، آن را اجرا کنید.

۷- حال برای بخش منومارکری به نام Menu و برای تصاویر گلها مارکری به نام Pic2 درج کرده، بقیه رفتارهای حرکت را روی فریمها (Navigation) به روش مشابه ویرایش کنید.

خلاصه مطالب

رفتار عکس‌العملی (Action) است که یک شیء (Sprite) در برابر یک رویداد (Event) از خود نشان می‌دهد.

در دایرکتور به سه روش می‌توان رفتارها را ایجاد نمود که عبارتند از:

- به کمک ابزار بازبین رفتار (Behavior Inspector)
 - به کمک کتابخانه رفتارهای آماده (Library Palette)
 - به کمک لینگو (Lingo)، زبان اسکریپت نویسی دایرکتور
- یک رفتار می‌تواند دارای چندین رویداد باشد و به ازای هر رویداد می‌توان چندین عمل (Action) را تعریف نمود.

جهت مشاهده رفتارهای نسبت داده شده به اسپریت، کافی است پس از انتخاب اسپریت یکی از روشهای زیر را به کار برید:

- روش اول: مشاهده لیست رفتارهای موجود در Behavior Inspector
- روش دوم: مشاهده لیست رفتارهای موجود در زبانه Behavior از پنجره Property Inspector (شکل ۶-۷ سمت راست)
- روش سوم: بررسی لیست رفتارهای موجود در Sprite Overlay (شکل ۶-۷ سمت چپ) روشهای اضافه کردن رفتارها به یک یا چند اسپریت:
- روش اول: منوی اصلی اضافه کردن رفتار موجود در Behavior Inspector را باز کرده، رفتار خود را از میان کلید رفتارهای ایجاد شده در نمایش انتخاب کنید تا به لیست اضافه شود.
- روش دوم: باز کردن منوی اضافه کردن رفتار و انتخاب رفتار مورد نظر از طریق زبانه Behavior از پنجره Property Inspector.

- روش سوم: درگ کردن رفتار از داخل پنجره Cast روی اسپریت حذف یک رفتار از اسپریت به دو روش امکان پذیر است:
 - ۱- با استفاده از پنجره Behavior Inspector
 - ۲- با استفاده از زبانه Behavior موجود در Property Inspector
- بطور کلی ۵ دسته رویداد در پنجره Behavior Inspector به شرح زیر موجود می باشد:
 - رویدادهای مربوط به دکمه اصلی ماوس و حرکت آن مانند Mouse Up
 - رویدادهای مربوط به دکمه سمت راست مانند Right Mouse Up
 - رویدادهای صفحه کلید مانند Key Up
 - رویدادهای مربوط به فریم مانند Exit Frame
 - رویدادهای مربوط به اسپریت مانند Begin Sprite
- Action های مربوط به پنجره Behavior Inspector نیز به ۶ گروه اصلی زیر طبقه بندی شده اند که عبارتند از:
 - ۱- Navigation: جهت هدایت کردن نمایش و هد مانند Go to Marker
 - ۲- Wait: مربوط به توقف نمایش مانند On Current Frame
 - ۳- Sound: جهت پخش اصوات مانند Play Cast Member
 - ۴- Frame: شامل فرامینی جهت کنترل رفتار در یک فریم می باشد مانند Change Tempo
 - ۵- Sprite: شامل دستوراتی جهت کنترل اسپریتها مانند Change Location
 - ۶- Cursor: شامل دو گزینه جهت تغییر شکل ماوس و برگرداندن آن به حالت عادی
- مارکر در واقع برچسبی برای مشخص نمودن یک فریم خاص است که مزایای زیر را در اختیار نمایش قرار می دهد:
 - ۱- مشخص شدن یک فریم خاص به کمک یک نام
 - ۲- امکان جابجا کردن یک مارکر روی فریمها در زمان طراحی نمایش
 - ۳- بالا رفتن سرعت طراحی رفتارها و خواناتر شدن آنها (رفتارها قابل فهم تر می گردند).
- جهت تعریف یک مارکر کافی است در فریم خاصی از کانال Marker کلیک کنید تا یک مارکر با نام پیش فرض New Marker ایجاد شود. سپس می توانید نام آن را به دلخواه تغییر دهید، کافی است روی آن در کانال Marker کلیک کنید تا وارد حالت ویرایش شده، نام جدید را تایپ و در انتها Enter کنید.
- جهت حذف یک مارکر کافیسست ماوس را روی علامت مثلث شکل مارکر ببرید تا شکل دست شود، سپس آن را به داخل پنجره Score (به سمت پایین) درگ کنید.
- در بخش سمت چپ کانال مارکر دو دکمه و یک منوی بازشو جهت حرکت سریع روی مارکرها وجود دارد.
- در دایرکتور می توان از مارکرها به شکل نسبی نیز استفاده نمود که موجب ساده تر شدن

طراحی رفتارها می‌گردد.
جهت حرکت نسبی از طریق پنجره Behavior Inspector کافی است پس از انتخاب فرمان Go Marker از مجموعه Navigation و باز شدن لیست مارکرها یکی از گزینه‌های زیر را جهت حرکت نسبی انتخاب کنید:
Loop: حرکت به مارکر جاری
Previous: حرکت به مارکر قبلی
Next: حرکت به مارکر بعدی

Learn In English

Some Events in the Behavior inspector:

BeginSprite contains statements that run when the playback head moves to a frame that contains a sprite that was not previously encountered.

EndSprite contains the statements that run when the playback head leaves a sprite and goes to a frame in which the sprite does not exist.

RightMouseUp indicates that the right mouse button was released. (On the Mac, Director treats a Control-click the same as a right mouse click on a Windows system.)

RightMouseDown indicates that the right mouse button was clicked.

MouseEnter indicates that the pointer entered a sprite's region.

MouseLeave indicates that the pointer exited a sprite's region.

MouseWithin indicates that the pointer is within the sprite's region.

KeyUp indicates that a key was released in a text or field sprite.

KeyDown indicates that a key was pressed in a text or field sprite.

واژه نامه تخصصی	
Action	عمل
Event	رویداد
Grid	شبکه
Leave	رها کردن
Nature	طبیعت
Navigation	کشتیرانی
Perform	انجام دادن
Prepare	آماده کردن
Snap	چسبیدن به
Specify	تعیین کردن
Within	در مدت

خودآزمایی:

- ۱- رفتار را با ذکر یک مثال شرح دهید؟
- ۲- روشهای متعدد ایجاد یک رفتار را نام ببرید.
- ۳- به چند روش می‌توان یک رفتار را به یک اسپرایت الحاق نمود؟
- ۴- رویدادهای مربوط به دکمه سمت راست ماوس را نام ببرید؟
- ۵- Marker را به همراه کاربرد آن تشریح کنید؟

پرسش‌های چهار گزینه‌ای

- ۱- به چند روش می‌توان یک رفتار را آماده نمود؟
الف) ۴ ب) ۳ ج) ۲ د) ۱
- ۲- کدامیک از گزینه‌های زیر علاوه بر ایجاد یک رفتار، آن را به اسپرایت نیز نسبت می‌دهد؟
الف) دکمه Behavior Inspector موجود در نوار ابزار اسپرایت
ب) دکمه Behavior Inspector موجود در نوار ابزار اصلی دایرکتور
ج) دکمه Script موجود در نوار ابزار اصلی دایرکتور
د) دکمه Behavior Inspector موجود در Property Inspector
- ۳- کدام جمله صحیح نیست؟
الف) هر اسپرایت می‌تواند بیش از یک رفتار داشته باشد.
ب) هر رفتار می‌تواند بیش از یک رویداد داشته باشد.
ج) برای هر رویداد نمی‌توان بیش از یک عمل تعریف نمود.
د) برای هر رفتار می‌توان بیش از یک عمل تعریف نمود.
- ۴- کدام گزینه صحیح است؟
الف) Behavior Inspector لیست رفتارهای نسبت داده شده به اسپرایت را نشان می‌دهد.
ب) Behavior Inspector موجود در Property Inspector لیست رفتارهای نسبت داده شده به اسپرایت را نشان می‌دهد.
ج) Sprite Overlay لیست رفتارهای نسبت داده شده به اسپرایت را نشان می‌دهد.
د) همه موارد
- ۵- کدام رویداد تا زمانی که ماوس در محدوده شیء قرار دارد به‌طور دائم اتفاق می‌افتد؟
الف) Right Mouse ب) Mouse Enter
ج) Mouse Within د) Mouse Down

۶- رویداد مربوط به رسیدن هد به اولین فریم یک اسپریت کدام است؟

الف) Prapare Frame

ب) End Sprite

ج) Begin Sprite

د) Start Sprite

۷- کدام Action به اندازه زمان تعیین شده برحسب ثانیه، هد را نگه می‌دارد؟

الف) On Current Frame

ب) Until Click or KeyPress

ج) For Time Duration

د) موارد الف و ج

۸- کدامیک از فرامین زیر جهت تعویض Member یک اسپریت به کار می‌رود.

الف) Change Ink

ب) Change Location

ج) Change Cursor

د) Change Cast member

۹- جهت حذف یک مارکر؟

الف) مارکر را به کمک ماوس به داخل پنجره Score درگ کنید.

ب) روی مارکر کلیک راست کرده، گزینه Remove Marker را اجرا کنید.

ج) پس از انتخاب مارکر در Score دکمه Del صفحه کلید را بزنید.

د) هر سه مورد

۱۰- کار فرمان Go loop چیست؟

الف) انتقال به مارکر بعدی

ب) برگشت به مارکر قبلی

ج) برگشت به اولین مارکر

د) انتقال به مارکری که نام آن loop می‌باشد.

11-Which Event Occure when the playback head leaves a sprite and go to next frame?

a) BeginSprite

b) MouseLeave

c) EndSprite

d) KeyDown

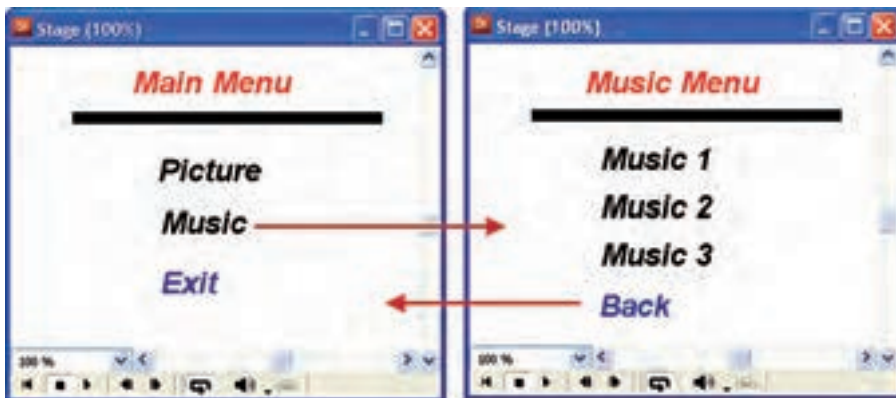
کارگاه چند رسانه‌ای

۱- پروژه‌ای جهت نمایش تصاویر و پخش موزیک به فرم زیر طراحی کنید:



شکل ۷-۲۶ منوی اصلی و ارتباط آن با زیر منوی Picture

- چنانچه کاربر بر روی گزینه Picture کلیک کرد، تعداد ۱۰ تصویر توسط دکمه‌های کنترلی با جلوه‌های انتقال (Transition) پخش شده و قابلیت برگشت توسط دکمه Home را دارا باشد. (شکل ۷-۲۶)
- با کلیک بر روی گزینه Music به زیر منوی آن مطابق شکل ۷-۲۷ رفته و قادر به پخش سه موزیک به دلخواه بوده و توسط گزینه Back به منو اصلی بازگشت کند.



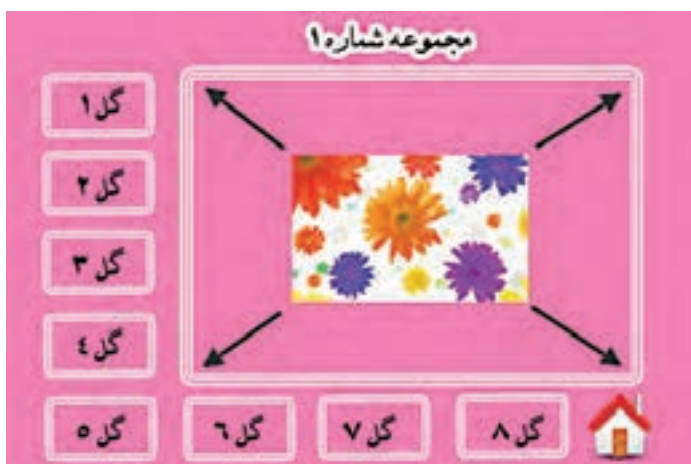
شکل ۷-۲۷ منوی اصلی و ارتباط آن با زیر منوی Music

۲- یک آلبوم عکس جهت نمایش تصاویر گلها طراحی کنید بطوریکه منوی اصلی آن بصورت شکل ۷-۲۸ باشد:



شکل ۷-۲۸ منوی اصلی آلبوم تصاویر

با کلیک بر روی هر مجموعه زیر منویی مطابق شکل ۷-۲۹ ظاهر شود (به عنوان نمونه با کلیک بر روی دکمه "مجموعه ۱" زیر منو با عنوان مجموعه شماره ۱ و تصاویر منحصر به فرد این مجموعه باشد) و با کلیک بر روی هر دکمه، تصویر مربوطه از وسط کادر پخش تصویر با اندازه کوچک و به حالت چرخشی بتدریج بزرگ شده و در محل مناسب قرار گیرد. در صورت کلیک بر روی دکمه Home به منوی اصلی بازگشت کند.



شکل ۷-۲۹ زیر منوی آلبوم تصاویر گلها

۳- یک برنامه آموزشی ساده ایجاد کنید که چهار عمل اصلی ریاضیات را آموزش دهد.
(راهنمایی: برای انتخاب نوع عمل یک منو در نظر گرفته و از تصاویر کوچک فانتزی به
همراه جلوه‌های حرکتی زیبا جهت جذابیت کار استفاده کنید.)





فصل هشتم

رفتارها به کمک Lingo

هدفهای رفتاری

- در پایان این فصل از هنرجو انتظار می‌رود:
- با لینگو، گرامر و اصطلاحات آن آشنا شود.
- نحوه تعریف و به‌کارگیری انواع متغیرهای محلی و سراسری را فراگیرد.
- اصول نوشتن Event Handler و Custom Handler و فراخوانی آنها را فراگیرد.
- اصول تعریف لیستها و به‌کارگیری آنها را در نمایش بدانند.
- اصول کار با حلقه‌های تکرار و فرمان شرطی را فرا گرفته، از آنها استفاده کنند.
- اسکریپت‌ها را به‌کمک امکانات پنجره Script بنویسد.
- اصول خطیابی و رفع خطا با Debugger Window را فراگیرد.
- با انواع پیامها آشنا شده، قادر باشد پیامی را به یک یا تمامی اسپرایت‌ها ارسال کند.

مقدمه:

Lingo زبان برنامه نویسی (اسکرپت نویسی) دایرکتور است که به شما امکان می دهد نمایش خود را بسیار قدرتمند و انعطاف پذیر کنید. این زبان در ابتدا شامل یک سری دستورات ساده جهت کنترل انیمیشن بوده، اما اکنون با داشتن قابلیت های شیء گرای، دارای امکانات فراوانی برای برنامه نویسی چند رسانه ای شده است. به کمک لینگو، دایرکتور می تواند بسیار فراتر از یک ابزار ایجاد انیمیشن یا نمایش باشد. به عنوان مثال شما می توانید برنامه های کاربردی کامل مانند بازیها، نرم افزارهای آموزشی و برنامه های کاربردی تجاری بسازید و از کار با دایرکتور لذت ببرید.

۱-۸- اسکرپت (script) چیست؟

به مجموعه ای از دستورات لینگو که درون یک رویداد (اداره کننده) نوشته شده و عمل خاصی را انجام می دهند، اسکرپت گفته می شود، اسکرپت ها در کنار سایر اجزاء نگهداری می شوند و می توان از آنها در قسمت های مختلف نمایش استفاده کرد. به عنوان مثال شما می توانید یک اسکرپت کلی برای حرکت به قسمت خاصی از نمایش تهیه کنید و آن را به هر دکمه ای که می خواهید نسبت دهید.

نکته: در دایرکتور می توانید از هر دو زبان لینگو (Lingo) یا جاوا اسکرپت JavaScript برای نوشتن اسکرپت ها استفاده کنید. ما در این مبحث آموزشی به بررسی زبان لینگو خواهیم پرداخت.

۲-۸- آشنایی با اصطلاحات لینگو

لینگو نیز مانند هر زبان دیگری دارای اصطلاحاتی است که لازم است قبل از اسکرپت نویسی، فرا گرفته شود. این اصطلاحات عبارتند از:

ثابتهای (Constant)

مقادیری هستند که در طول اجرای اسکرپت ثابت بوده، کاربر نمی تواند آنها را تغییر دهد. لینگو دارای ثابتهای از پیش تعریف شده ای مانند True, False, Space و Enter می باشد که می توان از آنها در اسکرپت ها استفاده کرد، اما اجازه تعریف ثابتهای شخصی را نمی دهد.

رویدادها (Events)

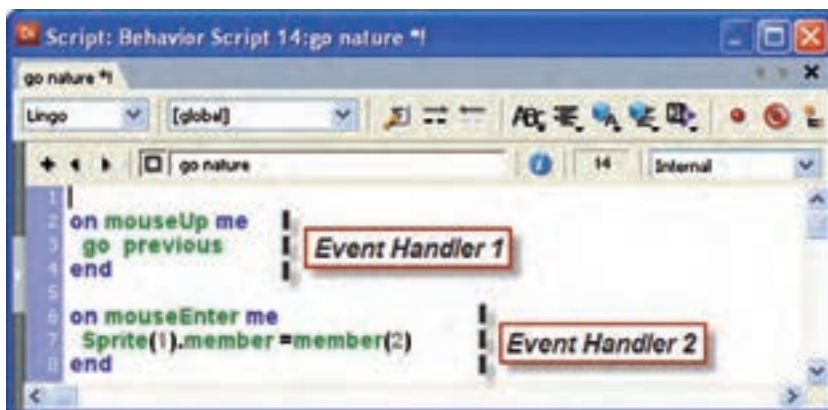
رویدادها در حین شرایط مختلفی از اجرای پروژه اتفاق می افتند، به عنوان مثال زمانیکه نمایش شروع می شود و یا زمانیکه هد می خواهد از یک فریم خارج شود.

عبارات (Expressions)

بخشی از یک دستور است که مقداری را تولید می کند، مانند عبارت 3×4 که نتیجه آن برابر ۱۲ است.

مدیر رویداد (Event handler) و پیام‌ها (Messages)

Event Handler که به‌طور مختصر به آن Handler می‌گویند، مجموعه‌ای از دستورالعمل‌های درون یک اسکریپت است که در قبال یک رویداد خاص مانند MouseUp اجرا می‌شود. در واقع هنگامی که یک رویداد در حین اجرای یک نمایش اتفاق می‌افتد، دایرکتور یک پیام خاص را تولید و به کلیه اسکریپت‌های متصل به اسپریت ارسال می‌کند. در صورتی که آن اسکریپت دارای مدیر رویدادی برای آن پیام (رویداد) باشد، دستورات نوشته شده در آن اجرا می‌شود. در غیر این صورت آن پیام نادیده گرفته خواهد شد. برای درک بهتر مطلب به شکل ۸-۱ و توضیحات آن توجه کنید:



شکل ۸-۱ دو مدیر رویداد موجود در Script

شکل ۸-۱ اسکریپتی را با دو مدیر رویداد (Event Handler) نشان می‌دهد که اولی برای مدیریت رویداد mouse Up و دومی برای مدیریت رویداد mouse Enter نوشته شده است. هنگامیکه رویداد mouse Enter روی اسپریت مربوط به این اسکریپت اتفاق می‌افتد، یعنی کاربر ماوس را به درون محدوده مستطیل شکل اسپریت انتقال می‌دهد، دایرکتور پیام mouse Enter را تولید کرده، آن را به اسکریپت ارسال می‌کند. چنانچه مدیر رویدادی برای این پیام وجود داشته باشد که در این مثال Event Handler 2 می‌باشد، دستورات نوشته شده در آن اجرا می‌گردد و به‌طور مشابه چنانچه پیام mouse Up به اسکریپت ارسال شود دستورات Event Handler 1 اجرا می‌شود و کلیه پیام‌های دیگر نادیده گرفته خواهند شد.

کلمات کلیدی (Keywords):

کلمات رزرو شده‌ای هستند که دارای معنی مشخصی هستند. به‌عنوان مثال در لینگو کلمه کلیدی end برای پایان دادن به یک مدیر رویداد استفاده می‌شود. همچنین کلمه کلیدی If شروع یک دستور شرطی می‌باشد.

عملگرها (Operators)

علامتهای خاصی هستند که عمل مشخصی را روی یک یا چند مقدار انجام می‌دهند، مانند عملگر جمع (+) که دو مقدار را به هم اضافه می‌کند. به عنوان مثال: عبارت $4+3$ که حاصل آن مقدار ۷ می‌باشد.

توابع (Functions)

اسکرپت‌های آماده‌ای هستند که عمل خاصی را روی داده‌های ورودی (پارامترها) انجام می‌دهند تابع می‌تواند دارای پارامتر ورودی نباشد و یک خروجی را به عنوان نتیجه پردازش برگرداند. مانند تابع $\text{Sqrt}(x)$ که جذر x را برمیگرداند. توابع جهت فراخوانی نیاز به هیچ شیئی ندارند و به صورت مستقل قابل فراخوانی هستند.

متدها (Methods)

متد بسیار شبیه تابع می‌باشد. بنابراین، پارامترهایی را دریافت و عملی را بر اساس آنها انجام می‌دهد. (ممکن است متد پارامتر ورودی نداشته باشد) اما تفاوت آن با تابع این است که جهت فراخوانی لازم است از طریق شیئی که متد درون آن تعریف شده، اقدام کرد. به عنوان مثال متد $\text{insertFrame}()$ یکی از متدهای شیئی `Movie` می‌باشد و جهت فراخوانی آن لازم است از فرمان `movie.insertFrame()` استفاده کنید.

پارامترها (Parameters)

به کمک پارامترها می‌توان انواع متفاوتی از مقادیر را به توابع، متدها و اسکرپت‌ها ارسال کرد. به عنوان مثال هنگام به کارگیری تابع $\text{Sqrt}(x)$ که برای جذرگیری به کار می‌رود، لازم است مقداری را به پارامتر x ارسال کرد تا تابع بتواند بر اساس مقدار این پارامتر، حاصل جذر x را محاسبه و برگشت دهد. در توابع و متدها، پارامترها درون پرانتز نوشته می‌شوند، اما اگر بخواهید برای اسکرپت شخصی خود پارامترهایی را در نظر بگیرید، نیازی به استفاده از پرانتز ندارید.

خصوصیات (Properties)

کلمات رزرو شده‌ای هستند که بیانگر ویژگی‌های متفاوت یک شیء مانند یک اسپریت یا یک عضو متنی می‌باشند، مانند `Width` که بیانگر عرض یک اسپریت، `Blend` که برای دسترسی به میزان شفافیت اسپریت یا `Text` که بیانگر متن تنظیم شده برای یک عضو متنی است. برای دسترسی به خصوصیات یک شیء از فرم کلی `Object.Property` استفاده می‌کنیم که `Object` در واقع شیء مورد نظر و `Property` یک ویژگی آن می‌باشد به عنوان مثال دستور `Member(1).Text = "OK"` موجب تغییر متن موجود در عضو متنی شماره یک ذخیره شده در `Cast` می‌گردد.

دستورات (Statements)

فرمانهایی به شکل صحیح هستند که دایرکتور می‌تواند آنها را اجرا کند. به بیان دیگر هر اسکریپت از تعدادی دستور العمل ساخته شده است. به‌عنوان مثال Go the frame یک دستورالعمل صحیح و اجرایی است که باعث متوقف شدن هد در فریم جاری می‌گردد.

متغیرها (Variables)

دایرکتور نیز مانند سایر زبانهای برنامه نویسی از متغیرها جهت نگهداری و به‌روز کردن مقادیر استفاده می‌کند. هر متغیر با یک نام در اسکریپت تعریف و سپس شناسایی می‌شود. جهت تعیین مقدار یک متغیر از عملگر = استفاده می‌کنیم، مانند $sum = 0$

لیست‌ها (Lists)

لیست در واقع تعدادی مقدار هم نوع می‌باشد که به‌وسیله یک متغیر قابل نگهداری و استفاده است (در سایر زبانها مانند JavaScript به آن آرایه می‌گویند) مانند لیست اسامی چند نفر، یا لیست قیمت چند کالا و... به‌عنوان مثال $data = [1,3,5]$ یک لیست ساده عددی است.

۳-۸- آشنایی با گرامر لینگو

در این بخش به بررسی شکل کلی نوشتن (Syntax)، انواع دستورات در لینگو می‌پردازیم:

استفاده از حروف کوچک و بزرگ (Case- Sensitive):

لینگو به حروف بزرگ یا کوچک حساس نیست بنابراین مؤلف می‌تواند دستورات را به‌صورت بزرگ یا کوچک یا ترکیبی از آنها تایپ کند. به‌عنوان مثال هر دو عبارت (1) SPRITE و (1) Sprite یک مفهوم را شامل می‌شوند.

۱-۳-۸- نوشتن توضیحات (Comments):

برای نوشتن توضیحات در مورد خطوط اسکریپت، قبل از نوشتن توضیحات، از دو علامت خط اتصال (کاراکتر منها) چسبیده به هم استفاده کنید. مثال زیر چگونگی این کار را نشان می‌دهد.

on mouseUp me

go "pic 1" -- goto Marker Named "Pic 1"

end

۲-۳-۸- کاربرد پرانتز (Parentheses):

پرانتز دارای دو کاربرد متفاوت است: ۱- در توابع و متدها ۲- در عبارات ریاضی

پرانتز در توابع و متدها:

در انتهای هر تابع یا متدی باید از پرانتز استفاده کنید. حتی اگر آن متدی تابع دارای

پارامتر ورودی نباشد.

به‌عنوان مثال برای استفاده ازمتد beep که فاقد پارامتر بوده، صدای بوق کامپیوتر را در می‌آورد؛ باید از دستور Sound.Beep() _ استفاده کرد. توجه کنید که استفاده از پرانتز ضروری است وگرنه موجب ایجاد خطا خواهد شد.
پرانتز در عبارات ریاضی:

همانطور که می‌دانید از پرانتز جهت دسته بندی و تعیین اولویت عملگرها در عبارات ریاضی استفاده می‌شود به‌عنوان مثال برای محاسبه میانگین سه مقدار x, y, z از رابطه زیر استفاده می‌شود:

$$\text{Ave} = (x + y + z) / 3$$

۳-۸- نحوه نوشتن Handler و اضافه کردن پارامترهای آن:

نوشتن یک هندلر (Handler) در لینگو بسیار آسان است به این ترتیب که ابتدا از کلمه کلیدی on استفاده کرده، سپس نام هندلر را نوشته و در صورت نیاز، در انتها پارامترها را به ترتیب ذکر می‌کنیم (استفاده از پرانتز برای پارامترها اختیاری است) و پس از نوشتن دستورات هندلر، آن را با کلمه کلیدی end خاتمه می‌دهیم (در Event Handler نام هندلر نام رویداد می‌باشد).

شکل کلی نوشتن یک هندلر:

On HandlerName (Parameter1,Parameter2,...)

Statements

End

به مثال‌های زیر توجه کنید:

مثال ۱:

On MouseDown me

Go frame 5

End

مثال فوق یک Event Handler می‌باشد که نام آن Mouse Down بوده دارای یک پارامتر به‌نام me می‌باشد (تمامی هندلرهای رفتار که برای اسپرایت‌ها تولید می‌شوند، دارای یک پارامتر پیش فرض به‌نام me هستند که از طریق آن می‌توان به شیء اسپرایت دسترسی داشت) که با کلیک روی اسپرایت مربوطه اجرا شده، دستور انتقال هد به فریم شماره ۵ را انجام می‌دهد.

مثال ۲:

On Sum (x,y)

Z= x + y

End

در این مثال نام هندلر Sum و پارامترهای آن X و Y می‌باشد و عمل جمع کردن مقادیر پارامترها را انجام می‌دهد.

نوشتن دستورات طولانی در چند سطر:

برای نوشتن یک دستور طولانی می‌توان از علامت \ استفاده و آن را به دو یا چند سطر تقسیم کرد. مثال زیر چگونگی انجام این عمل را نشان می‌دهد:

```
On MouseUp me
T= Sprite(2).Member\
.Paragraph[1].word[3] \
.char[1]
End
```

۴-۳-۸- دسترسی به خصوصیات و متدهای اشیاء:

برای دسترسی به خاصیت یا متد مشخصی از یک شیء، از گرامر نقطه‌ای استفاده می‌شود. در این گرامر ابتدا نام شیء، سپس یک نقطه جهت جداکردن، و در انتها نام یک خاصیت (property) یا متد (Method) را ذکر می‌کنیم. به مثالهای زیر توجه کنید:

مثال ۱:

Member ("Title").text: دسترسی به خاصیت متن موجود در عضو متنی که به نام Title در کتابخانه Cast ذخیره شده است.

مثال ۲:

Sound(1).Stop(): متد Stop() موجب توقف کامل صدای در حال پخش موجود در کانال شماره یک صدا می‌گردد.

عناصر متنی دارای سلسله مراتب بالا به پایین هستند که عبارتند از: پاراگراف، کلمه و کاراکتر. بنابراین شما می‌توانید به هر قطعه دلخواه از متن دسترسی داشته باشید، به شرط آنکه سلسله مراتب را به‌طور صحیح با نقطه از یکدیگر جدا کنید. مثالهای زیر چگونگی دسترسی به قطعات متن را نشان می‌دهد:

مثال ۱:

Member ("Main").Paragraph [1]: دسترسی به پاراگراف اول عضو متنی Main

مثال ۲:

Member ("Main").Paragraph[1].Word[3]: دسترسی به کلمه سوم در پاراگراف اول عضو متنی Main

مثال ۳:

Member ("Main").Paragraph[1].Word[3].char[4]: دسترسی به کاراکتر چهارم از کلمه سوم موجود در پاراگراف اول عضو متنی Main

۵-۳-۸- دسترسی به عضو کتابخانه Cast از طریق اسپریت آن:

همانطور که می‌دانید هر اسپریت به یکی از اعضای موجود در Cast متصل است و آن عضو را در صحنه نمایش می‌دهد. در لینگو می‌توان با استفاده از خاصیت Member اسپریت، به عضو و کتابخانه‌ای که عضو در آن ذخیره شده، دسترسی داشت. برای این منظور از شکل کلی Member.(Index OR Name) Sprite استفاده کنید. در این فرم کلی منظور از Index شماره کانال اسپریت است که به جای آن می‌توان از نام اسپریت نیز استفاده کرد (در این صورت نام را درون علامت‌های نقل قول دوتایی محصور کنید).

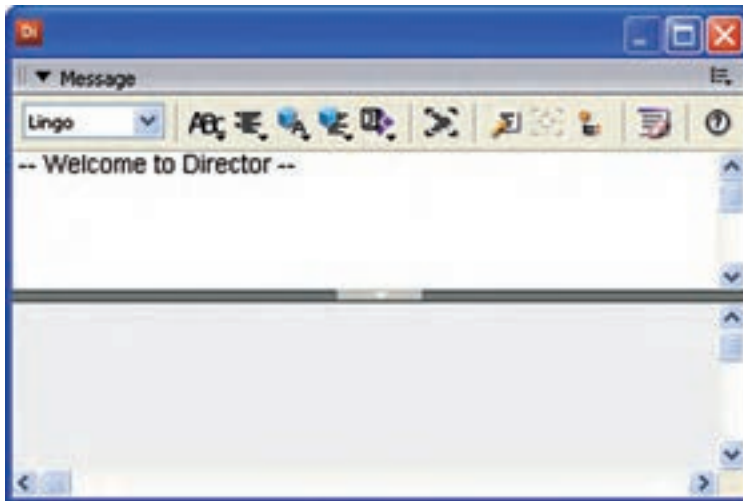
مثال: جهت دسترسی به عضوی از Cast که اسپریت شماره دو به آن متصل است، از فرمان زیر استفاده کنید:

Put Sprite(2).Member

نکته: نتیجه این فرمان در پنجره Message نمایش داده می‌شود.

۴-۸- شروع کار با لینگو:

پنجره Message نقطه شروع مناسبی برای تایپ دستورات لینگو و مشاهده نتایج آنها می‌باشد اگر این پنجره را مشاهده نمی‌کنید، می‌توانید روی دکمه Message window از نوار ابزار اصلی دایرکتور کلیک کنید یا از منوی window گزینه Message را برگزینید تا این پنجره مطابق شکل ۲-۸ ظاهر شود.



شکل ۲-۸ پنجره Message

این پنجره دارای دو بخش است، در بخش بالا می‌توان دستورات را خط به خط تایپ و نتایج آنها را در قسمت پایین پنجره مشاهده کرد.

اکنون با ذکر چند مثال به نحوه استفاده از این پنجره می‌پردازیم.

مثال ۱:

در این مثال ساده چند عبارت ساده تایپ و نتایج آنها را مشاهده خواهیم کرد، برای اینکه مقادیر را در Message ببینید باید از دستور ساده put استفاده کنید. اکنون دستورات زیر را تایپ و نتایج آنها را مشاهده کنید.

put "Hello"

Put 5*2

Hello : نتایج :

10

مثال ۲:

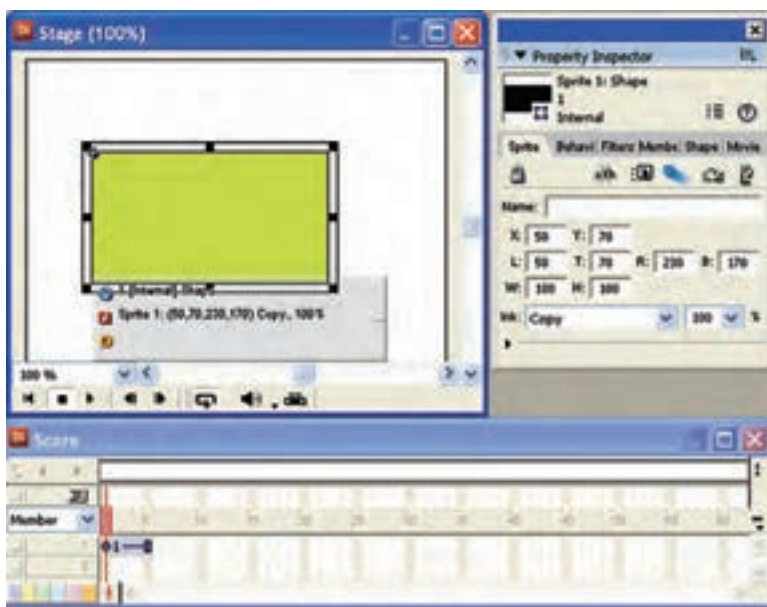
در این مثال یک مستطیل را در صفحه قرار داده، به کمک put، اندازه، ارتفاع و عرض آن، به همراه محل قرار گیری در stage را نمایش می‌دهیم.

مراحل کار:

یک نمایش خالی ایجاد کنید.

در جعبه ابزار، ابزار مستطیل تو پر را فعال کرده، به کمک عمل درگ یک مستطیل در صفحه ترسیم کنید. دقت کنید که اسپریت در کانال ۱ پنجره score قرار داشته باشد.

(شکل ۳-۸)



شکل ۳-۸ ترسیم یک مستطیل در صحنه و تنظیم مکان و اندازه

به کمک نوار ابزار اسپریت یا زبانه sprite از property Inspector مقادیر زیر را تنظیم کنید.

X=50 Y=70 W=180 H=100

اکنون دستورات زیر را در پنجره Message وارد کرده، نتایج آنها را بررسی کنید:

put sprite (1). locH مختصات افقی اسپریت

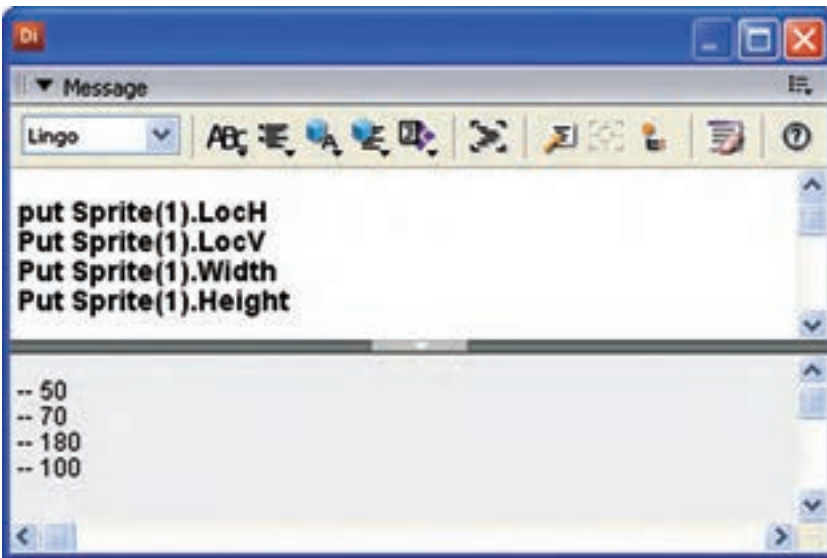
put sprite (1). locV مختصات عمودی اسپریت

put sprite (1). width عرض اسپریت

put sprite (1). Heigh ارتفاع اسپریت

نکته: منظور از sprite (1) اسپریتی است که در کانال شماره ۱ پنجره score قرار گرفته است و هم اکنون هد بر روی آن قرار دارد.

شکل ۴-۸ دستورات و نتایج آنها را نشان می‌دهد:



شکل ۴-۸ تایپ دستورات و مشاهده نتایج

در این مثال با چهار خاصیت اصلی اسپریت‌ها آشنا شدید که عبارتند از:
 LocH: مخفف Location Horizontal و به مفهوم مختصات افقی اسپریت است (X)
 LocV: مخفف Location Vertical و به مفهوم مختصات عمودی اسپریت است (Y)
 Width: اندازه عرض اسپریت به پیکسل
 Height: اندازه ارتفاع اسپریت به پیکسل

۸-۵- آشنایی با انواع اسکریپت‌ها:

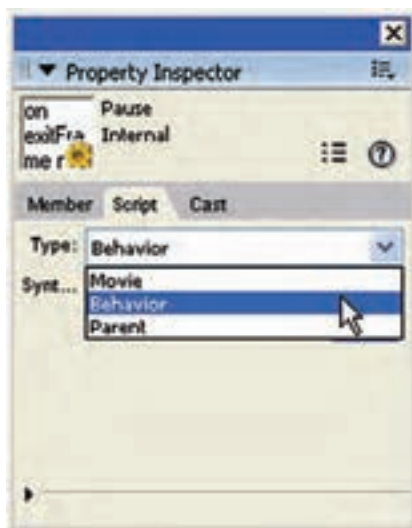
در دایرکتور چهار نوع اسکریپت وجود دارد که عبارتند از:
 ۱- Behavior (رفتار) ۲- Movie Script (اسکریپت فیلم) ۳- Cast Member Script (اسکریپت عضو)
 ۴- Parent Script (اسکریپت والد)
 هر چهار نوع اسکریپت در Cast ذخیره شده، با آیکن‌های ویژه‌ای قابل شناسایی و مشاهده هستند. (شکل ۸-۵)



شکل ۸-۵ آیکن شناسایی انواع Script

۸-۶- تعیین نوع اسکریپت:

از طریق زبان Script موجود در Property Inspector می‌توان نوع اسکریپت نوشته شده را تعیین کرد. این گزینه‌ها عبارتند از:
 Behavior – Movie Script و Parent Script
 در ضمن گزینه‌ای نیز جهت انتخاب زبان اسکریپت نویسی به نام Syntax نیز موجود است. (شکل ۸-۶)

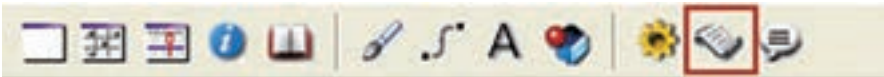


شکل ۸-۶ تعیین نوع اسکریپت به کمک زبان Script

۸-۷- اصول استفاده از پنجره Script

پنجره Script ابزاری است که در آن می‌توانید اسکریپت‌های مورد نیاز خود را بنویسید یا آنها را ویرایش کنید. اگر این پنجره را مشاهده نمی‌کنید، می‌توانید به یکی از روشهای زیر، آن را باز کنید:

۱- دکمه Script Window موجود در نوار ابزار اصلی دایرکتور (شکل ۸-۷)

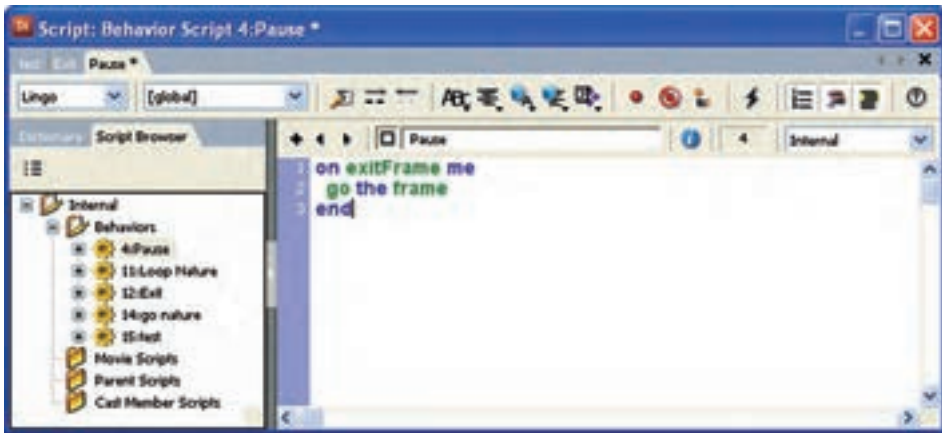


شکل ۸-۷ دکمه باز کردن پنجره Script

۲- از طریق منوها Window/Script (Ctrl+0)

۳- دوبار کلیک روی یک اسکریپت در پنجره Cast

اجرای هر یک از روشهای فوق موجب باز شدن پنجره Script مطابق شکل ۸-۸ می‌گردد.

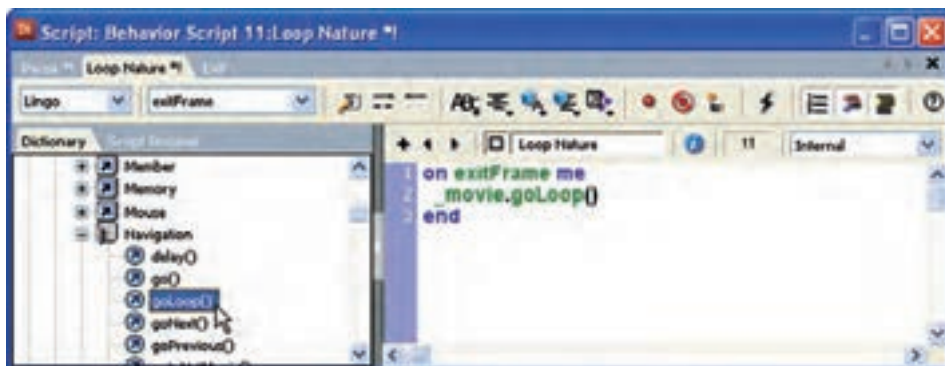


شکل ۸-۸ پنجره Script - نمای Script Browser

این پنجره دارای دو بخش چپ و راست و یک نوار ابزار اصلی است: بخش سمت راست محل تایپ و ویرایش اسکریپت‌هاست و اصلی‌ترین بخش این پنجره می‌باشد.

بخش سمت چپ یک مرورگر کاربردی است که به نسخه ۱۱ دایرکتور اضافه شده و دارای دو زبانه اصلی به نام‌های Dictionary View و Script Browser است:

توسط Dictionary View (نمای کتابخانه) می‌توانید به کلیه خصوصیات، متدها و توابع داخلی هر یک از زبانهای لینگو و جاوااسکریپت دسترسی پیدا کرده، از آنها در نوشتن اسکریپت‌ها استفاده کنید (شکل ۹-۸)



شکل ۹-۸ پنجره Script - نمای Dictionary

۸-۲-۱- جستجوی توابع و استفاده از آنها جهت ایجاد اسکریپت‌های جدید به کمک نمای Dictionary View:

پنجره Script را باز کنید.

روی زبانه Dictionary کلیک کنید.

از منوی کرکره‌ای موجود در نوار ابزار پنجره Script، یکی از زبانهای Lingo و یا Java Script را انتخاب کنید (زبان پیش فرض Lingo می‌باشد) تا توابع کتابخانه بر اساس آن به‌روز شود.

در این کتابخانه، توابع بر اساس نوع عملکردشان به گروه‌های مشخصی طبقه بندی شده‌اند، برای باز کردن هر گروه روی علامت + کنار آن کلیک کنید، اگر می‌خواهید لیست توابع مرتب شده بر اساس حروف الفبا را مشاهده و به کار ببرید، گروه Index را باز کنید.

برای اضافه کردن یک تابع به ویرایشگر روی آن دوبار کلیک کنید. اکنون نمایش را ذخیره کنید تا تغییرات ذخیره شوند. در پایان برای اطمینان از تغییرات اعمال شده در اسکریپت‌ها، روی دکمه Recompile All Modified Scripts در نوار ابزار پنجره Script کلیک کنید تا دستورات کامپایل شده و دایرکتور در صورت وجود خطا، آنها را اعلام کند. (شکل ۱۰-۸)



شکل ۱۰-۸ دکمه Recompile All Modified Scripts

۲-۸-۲- کاربردهای Script Browser:

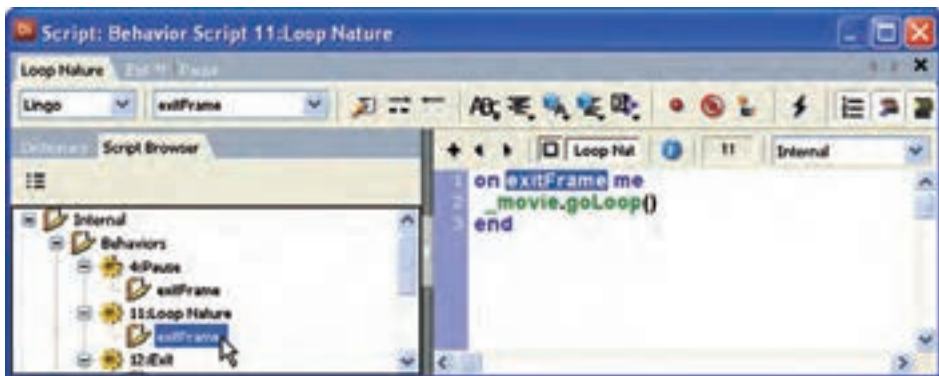
به طور کلی از این بخش پنجره Script در دایرکتور برای جستجوی اسکریپتها و هندلرها، بر اساس ساختار درختی یا لیست مرتب‌سازی اسکریپتها بر اساس نام، نام کتابخانه (Cast)، شماره ردیف عضو (Cast Number) یا نوع اسکریپت در یک لیست و باز کردن یک handler در ویرایشگر پنجره Script استفاده می‌شود.

۳-۲-۸- جستجو و ویرایش اسکریپتها به کمک نمای Script Browser:

برای این که بیشتر با نمای Script Browser و نحوه کار با آن آشنا شوید، ما به صورت مرحله‌ای نحوه کار با آن آموزش می‌دهیم. بنابراین از شما می‌خواهیم مراحل زیر را دنبال کنید:
پنجره Script را باز کنید.

روی زبانه Script Browser کلیک کنید تا اسکریپتها بر اساس ساختار درختی به نمایش درآید. در این نما اسکریپتها بر اساس نوع، طبقه‌بندی شده‌اند (Behavior Script-Movie Script - Parent Script و Cast Member Script).

جهت باز کردن یک اسکریپت در ویرایشگر دوبار روی آن کلیک کنید. Handlerها نیز به صورت ساختار درختی در زیر نام هر اسکریپتی نمایش داده می‌شوند. (شکل ۱۱-۸ و Handler به نامهای mouse Up و sum موجود در اسکریپت Exit را نشان می‌دهد)



شکل ۱۱-۸ باز کردن یک Script

۱- برای نمایش اسکریپتها به صورت لیست، از دکمه Script Explorer View کمک بگیرید. (شکل ۱۲-۸)

Script Name	#	Cast	Type	Syntax
Pause	4	Internal	Behavior	Lingo
Loop Nature	11	Internal	Behavior	Lingo
Exit	12	Internal	Behavior	Lingo

شکل ۸-۱۲ نمایش اسکریپت‌ها به صورت List

۴-۷-۸- آشنایی بیشتر با نوار ابزار پنجره Script:

این نوار ابزار دارای دکمه‌های مفیدی جهت تایپ سریع دستورات می‌باشد که امکان استفاده از دستورات لینگو را بدون نیاز به ابزار Dictionary View برطرف می‌کند. همچنین دارای ابزارهایی جهت قالب بندی متن اسکریپت‌هاست و بالاخره به مؤلف این امکان را می‌دهد که اسکریپت، خود را به صورت پیشرفته اجرا و کنترل کند. (شکل ۸-۱۳)



شکل ۸-۱۳

۸-۸- متغیرها (Variables)

مانند سایر زبانهای برنامه نویسی، لینگو نیز امکان استفاده از متغیرها را فراهم ساخته است. نوع متغیر با توجه به مقداری که درون آن ذخیره می‌گردد، مشخص می‌شود و نیازی به تعریف نوع ندارد. اکنون پنجره Message را باز کرده و دستورات زیر را در آن تایپ کنید.

X = 70 (متغیر عددی) صحیح
 Y = 12.75 (متغیر عددی) اعشاری
 Z = 1.0 (متغیر عددی) اعشاری
 Name = " Director MX" (متغیر رشته‌ای)
 F=True متغیر منطقی
 Put x
 Put y

Put z
Put Name
Put F

۱-۸-۸-۱ اصول کار با متغیرهای محلی و سراسری

اگر یک متغیر درون یک هندلر تعریف شود، آن متغیر محلی بوده، محدوده اعتبار آن فقط در همان هندلر خواهد بود. به عنوان مثال به هندلر زیر دقت کنید.

On MouseDown me

N=1

Put N

End

اما متغیرهای سراسری در اولین خط پنجره script و در ناحیه Global تعریف می‌شوند و در هر روال رویداد موجود در همان اسکریپت، قابل دسترس می‌باشند؛ اما اگر یک روال رویداد در یک اسکریپت جداگانه تعریف شده باشد و بخواهد از یک متغیر سراسری که قبلاً تعریف شده، استفاده کند. کافی است در اولین سطر رفتار در Global تعریف شود. نحوه کلی تعریف این متغیرها به فرم < نام متغیر > Global است. مثال زیر بیانگر این مطلب است:

Global Sum

On MouseUp me

Sum=0

End

۹-۸- عملگرها (operators)

لینگو نیز مانند سایر زبانهای اسکریپت نویسی شامل مجموعه کاملی از عملگرهاست که عبارتند از:

- عملگرهای ریاضی
- عملگرهای مقایسه‌ای
- عملگرهای منطقی
- عملگرهای رشته‌ای

۱-۹-۸- عملگرهای ریاضی موجود در لینگو عبارتند از:

- ۱- عملگر پرانتز برای تعیین اولویت‌ها ()
- ۲- عملگر جمع با علامت +
- ۳- عملگر تفریق با علامت -
- ۴- عملگر ضرب با علامت *

۵- عملگر تقسیم با علامت /

۶- عملگر باقیمانده تقسیم در حالت صحیح Mod

اکنون برای تست این عملگرها می‌توانید در پنجره Message دستورات زیر را تست کنید:

$$X = 12$$

$$Y = 5$$

$$\text{Put } x + y \rightarrow 17$$

$$\text{Put } x - y \rightarrow 7$$

$$\text{Put } x \times y \rightarrow 60$$

$$\text{Put } x / y \rightarrow 2$$

$$\text{Put } x \text{ mod } y \rightarrow 2$$

$$\text{Put } (x+y)*2 \rightarrow 34$$

همانطور که ملاحظه می‌کنید حاصل تقسیم ۱۲ بر ۵ مقدار صحیح ۲ می‌باشد، اما حاصل تقسیم اعشاری آن مقدار ۲/۴ است، زیرا عملگر تقسیم در لینگو هم می‌تواند نتیجه صحیح برگرداند و هم نتیجه اعشاری و این بستگی به نوع عملوندهایش دارد، اگر دو عملوند صحیح باشد، مانند مثال فوق، نتیجه خارج قسمت صحیح تقسیم عدد اول بر عدد دوم خواهد بود، اما اگر حداقل یکی از عملوندها اعشاری باشد. نتیجه تقسیم اعشاری برگردانده می‌شود(اعداد اعشاری با چهار رقم اعشار نمایش داده می‌شوند). اگر می‌خواهید نتیجه تقسیم x بر y (در مثال فوق) را به صورت کامل ببینید باید از توابع تبدیل نوع، به شرح زیر استفاده کنید.

تابع float

این تابع مقدار صحیحی را گرفته و تبدیل به اعشاری می‌کند.

مثال ۱:

$$X = 12$$

$$\text{Put float } (x) \rightarrow 12.0000$$

$$Y = 5$$

$$\text{Put float } (x) / Y \rightarrow 2.4000$$

مثال ۲:

$$X=12.0$$

$$Y=5$$

$$\text{Put } X/Y \rightarrow 2.4000$$

نکته: در مثال ۲ به دلیل اعشاری بودن مقدار X نیازی به استفاده از تابع float نیست.

تابع Integer

این تابع مقداری اعشاری را گرفته و آن را گرد می‌کند.

مثالهای زیر را تست کنید.

Put Integer (5) → 5
 Put Integer (5.25) → 5
 Put Integer (5.5) → 6

۲-۹-۸- اولویت عملگرهای ریاضی:

عملگرهای زبان لینگو نیز مانند سایر زبانهای برنامه نویسی اولویت اجرا دارند، به این ترتیب که ابتدا عملیات ضرب و تقسیم و mod از چپ به راست و سپس عملیات جمع و تفریق اجرا می گردند.

۳-۹-۸- استفاده از پرانتز:

به کمک پرانتز می توان اولویت عملگرها را تغییر داد، مثلاً کاری کرد که ابتدا عملیات جمع و تفریق انجام شود. مثالهای زیر نمونه ای از کاربرد پرانتز در محاسبه میانگین سه عدد است. این دستورات را در Message آزمایش کنید:

Ave = float (12 + 15 + 20)/3
 Put Ave → 15.6667

نکته: اگر در یک عبارت عملگرهایی با اولویت یکسان وجود داشته باشد از چپ به راست اجرا می شوند.

۴-۹-۸- عملگرهای منطقی موجود در لینگو عبارتند از:

عملگر نقیض not
 عملگر "و" منطقی and
 عملگر "یا" منطقی or
 اولویت اجرای این عملگرها عبارتند: از الف) not ب) and و or: و در حالت کلی، هم ردیف عملگرهای ریاضی می باشند.

۵-۹-۸- عملگرهای مقایسه ای موجود در لینگو عبارتند از:

کوچکتر با علامت <
 کوچکتر یا مساوی با علامت <=
 بزرگتر با علامت >
 بزرگتر یا مساوی با علامت >=
 مساوی با علامت =
 نامساوی با علامت <>

نکته: اولویت عملگرهای مقایسه‌ای یکسان و پس از عملگرهای ریاضی و منطقی اجرا می‌شوند.

برای آشنایی بیشتر با عملگرهای منطقی و مقایسه‌ای دستورات زیر را اجرا کنید:

Put (7>2) AND (12<>12) → 0

Put (7>2) OR (12<>12) → 1

Put NOT(17=17) → 0

نکته مهم:

در لینگو همانند زبان C مقدار منطقی False با مقدار عددی صفر و مقدار منطقی True با مقدار عددی یک تفسیر می‌شود.

۶-۹-۸- عملگرهای رشته‌ای موجود در لینگو عبارتند از:

عملگر الحاق دو رشته: &

عملگر الحاق دو رشته و اضافه کردن یک فاصله (Space) بین آنها: &&

دو عملگر رشته‌ای فوق دارای اولویت یکسان می‌باشند و در حالت کلی همانند عملگرهای مقایسه‌ای در انتها اجرا می‌شوند.

برای آشنایی بیشتر با عملگرهای رشته‌ای، دستورات زیر را اجرا کنید:

Put "Director"&"11" → " Director11"

Put "Director"&&"11" → " Director 11"

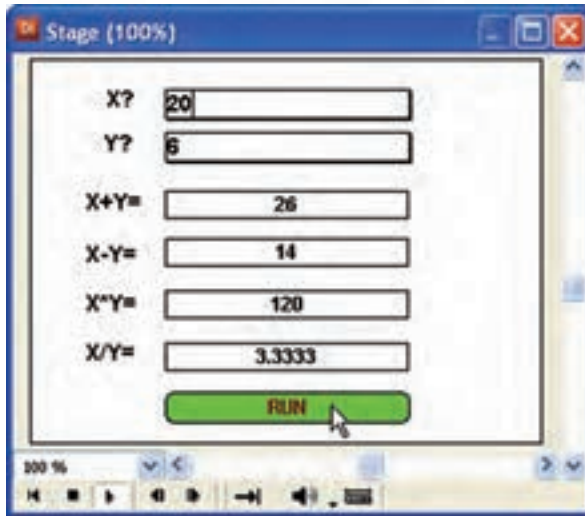
۱۰-۸- یک مثال کاربردی جهت استفاده از عملگرها و متغیرها

می‌خواهیم یک پروژه محاسباتی ایجاد کنیم، به طوری که کاربر بتواند دو مقدار را وارد کرده، با زدن دکمه RUN حاصل جمع، تفریق، ضرب و تقسیم آنها را مشاهده کند.

مراحل انجام کار:

۱- یک نمایش جدید ایجاد کنید

۲- جعبه ابزار را در حالت نمایش classic قرار داده، ۶ عدد فیلد متنی به کمک ابزار Field و ۶ عدد متن به کمک ابزار Text به عنوان برچسب روی صحنه قرار دهید و در انتها یک دکمه به کمک ابزار Push Button در پایین صفحه به شکل زیر قرار دهید.



شکل ۱۴-۱ طراحی صحنه برای پروژه چهار عمل اصلی

توجه: هر بار که یکی از ابزارهای فوق را انتخاب و در صفحه کلیک می‌کنید، می‌توانید متن یا عنوان مورد نظر را وارد کنید، اما اگر می‌خواهید پس از نصب کنترل‌ها مقادیر مورد نظر خود را وارد کنید، باید روی کنترل (مثلاً Field) دابل کلیک کنید تا به حالت ویرایش برود و سپس عنوان خود را وارد کنید.

۳- فیلدهای مقابل عناوین x ? و y ? را قابل ویرایش کنید (Editable) تا کاربر بتواند هنگام اجرا مقادیر خود را درون آنها وارد کند، برای این منظور هر دوی آنها را انتخاب و در پنجره Property Inspector وارد زبانه Field شده گزینه Editable را تیک بزنید.

نکته: از قسمت پایین زبانه Field برای کنترل ظاهر کادر فیلد متنی استفاده کنید، گزینه Border ضخامت حاشیه و گزینه Box Shadow سایه آن را تنظیم می‌کند.

۴- بهتر است رنگ زمینه دکمه خود را تغییر دهید تا در صفحه بهتر قابل رویت باشد. برای این منظور روی دکمه دو بار کلیک کرده تا وارد حالت ویرایش شود، سپس رنگ زمینه را به کمک جعبه ابزار به رنگ زرد تغییر دهید.

۵- برای دسترسی راحت‌تر به اسپریت‌ها، به جای استفاده از شماره کانال آنها، در score می‌توان از یک نام دلخواه که به اسپریت داده می‌شود، کمک گرفت بنابراین برای اسپریت مقابل عنوان x ? (فیلد متنی اول) نام txt_x ؛ اسپریت مقابل عنوان y ? (فیلد متنی دوم) نام txt_y ؛ اسپریت مقابل عنوان $x+y =$ (فیلد متنی سوم) نام txt_Sum ؛ برای اسپریت مقابل عنوان $x-y =$ (فیلد متنی چهارم) نام txt_Min ؛ برای اسپریت مقابل عنوان $x/y =$ (فیلد متنی پنجم) نام txt_Mul و بالاخره برای اسپریت مقابل عنوان $x/y =$ (فیلد متنی ششم) نام

txt_Div را در نظر بگیرید. (جهت انتساب یک نام به یک اسپریت ابتدا آن را انتخاب و سپس وارد زبانه sprite از پنجره Property Inspector شده و در کادر Name نام آن را وارد کنید).

۶- اکنون باید اسکرپتی به صورت یک رفتار (Behavior) برای دکمه بنویسید که بتواند چهار عمل اصلی را روی ورودیهای x و y انجام دهد، بنابراین پنجره script را به کمک دکمه آن در نوار ابزار اصلی دایرکتور باز کنید. سپس در کادر cast member Name نام دلخواهی مانند Beh-Run وارد کرده، با زدن Enter وارد قسمت اصلی این پنجره شده، کد زیر را در آن تایپ کنید:

On mouseup me

```
x=integer(member("txt_x").text)
y=integer(member("txt_y").text)
sum=x+y
mine=x-y
mul=x*y
div=float(x)/float(y)
member("txt_Sum").text=string(sum)
member("txt_Min").text=string(mine)
member("txt_Mul").text=string(mul)
member("txt_Div").text=string(div)
End
```

۷- اکنون پروژه خود را تست کنید.

نکته ۱:

در مثال فوق از طریق اسپریت برای دسترسی به متن استفاده شده است که فرم کلی آن به صورت زیر است:

```
Sprite( Index OR Name).Member.Text
```

در فرم کلی داده شده، Index شماره کانال اسپریت می باشد که به جای آن می توان از Name یا نام اسپریت استفاده کرد و منظور از Member یک عضو متنی است که محتویات اسپریت را تشکیل می دهد.

نکته ۲:

لازم است برای نمایش متغیرهای عددی با فیلدهای متنی، آنها را به متن تبدیل کنید. برای تبدیل عدد به کاراکتر از تابع String استفاده کنید.

۱۱-۸- اصول ایجاد مدیر رویداد (Event Handler) و پیامها

همانطور که در ابتدای فصل نیز اشاره شد، برای نوشتن یک مدیر رویداد (Event Handler) باید از فرم کلی زیر استفاده کنید:

```
On Event me  
Statements  
End
```

در این فرم کلی On و End کلمات کلیدی هستند که به ترتیب سطر آغاز و پایان یک مدیر رویداد را مشخص می‌کند و منظور از Event یکی از رویدادهای تعریف شده در دایرکتور است که در فصل هفتم با انواع رویدادهای مهم دایرکتور آشنا شدید. me که به‌عنوان یک پارامتر برای هندلر عمل می‌کند، در واقع اشاره‌گری است به اسپریتی که رویداد روی آن اتفاق می‌افتد و از طریق آن می‌توان به برخی از خواص اسپریت دسترسی پیدا کرد، مانند me.SpriteNum که شماره کانال اسپریت را برمی‌گرداند. استفاده از me اختیاری است. و بالاخره Statements دستوراتی است که قرار است در صورت اتفاق افتادن رویداد اجرا شوند. مثال زیر نحوه نوشتن مدیر رویداد و مثالی از کاربرد me را نشان می‌دهد.

مراحل انجام کار:

- ۱- یک نمایش جدید ایجاد کنید.
- ۲- سه اسپریت دلخواه در صحنه ایجاد کنید به‌طوری که کانالهای شماره ۱ تا ۳ را اشغال کنند.
- ۳- پنجره اسکریپت را باز کرده، رفتار زیر را در آن تایپ کنید و آن را به نام Test1 ذخیره کنید.

```
On MouseUp me  
put me.SpriteNum  
End
```

- ۴- رفتار Test1 را به هر سه اسپریت بچسبانید.
- ۵- یک رفتار توقف در انتهای نمایش قرار دهید.
- ۶- اکنون پنجره Message را باز کرده، نمایش را اجرا کنید، با کلیک روی هر یک از اسپریت‌ها، شماره کانالی که اسپریت در آن قرار گرفته در پنجره Message نمایش داده می‌شود.

۱۲-۸- پیامها و انواع آن:

- در حالت کلی دو نوع پیام در دایرکتور وجود دارد که عبارتند از:
- ۱- پیامهای سیستم (System Messages)
 - ۲- پیامهای شخصی (Custom Messages)
- در این بخش به بررسی پیامهای سیستم پرداخته و پیامهای شخصی را به بخش بعدی (اصول ایجاد Custom Handler) موکول می‌کنیم.

همانطور که در ابتدای فصل نیز ذکر شد، منظور از System Messages همان رویدادهای استاندارد و از پیش تعیین شده دایرکتور است که هنگام نوشتن یک مدیر رویداد (Event Handler) از آنها استفاده می‌شود مانند Mouse Down که مربوط به ماوس است یا KeyUp که مختص صفحه کلید می‌باشد. زمانی که رویدادی برای یک اسپریت اتفاق می‌افتد، دایرکتور یک پیام را که هم نام با رویداد است (مانند KeyUp) را به کلیه اسکریپت‌های نوشته شده برای اسپریت ارسال می‌کند و در صورت وجود مدیر رویدادی طبق آن پیام، دستورات موجود در هندلر اجرا می‌شود، در غیر این صورت آن پیام نادیده گرفته خواهد شد.

۱۳-۸- اصول ایجاد و فراخوانی یک Custom Handler:

Custom Handler در واقع یک هندلر شخصی است که اسکریپت نویس برای انجام عملیات مشخصی بدون در نظر گرفتن رویدادهای سیستم می‌نویسد و سپس می‌تواند از درون هندلرهای دیگر آن را فراخوانی کند. فرم کلی نوشتن یک هندلر شخصی به فرم زیر است:

On HandlerName [Parameter1,Parameter2,...]

Statements

Return [Value]

End

در فرم کلی فوق، منظور از HandlerName نام دلخواه هندلر و منظور از Parameter پارامترهای هندلر می‌باشد که اختیاری هستند (بهتر است پارامترها را درون پرانتز قید کنید) و منظور از Statements دستوراتی است که هنگام فراخوانی هندلر اجرا می‌شوند و بالاخره اگر قصد دارید هندلر، مقداری را همانند یک تابع برگشت دهد از فرمان اختیاری Return در آخرین سطر دستورات استفاده کنید.

۱۳-۸- روش فراخوانی یک هندلر شخصی:

جهت فراخوانی یک Custom Handler با توجه به اینکه آیا دارای فرمان Return می‌باشد یا خیر، دو روش وجود دارد:

- روش اول:

اگر هندلر شخصی دارای Return نباشد، کافی است نام آن را تایپ و سپس در صورت وجود پارامترها، مقادیر آنها را در ادامه به ترتیب ذکر کنید. به عنوان مثال هندلر زیر را در نظر بگیرید:

Global T

On Sum (x,y)

T=x+y

End

ساده ترین شکل فراخوانی هندلر فوق استفاده از فرمان زیر است:

Sum (2,3)

دستور فوق موجب فراخوانی هندلر با مقادیر 2 برای پارامتر x و 3 برای پارامتر y می‌گردد و حاصل جمع آنها در متغیر سراسری T ذخیره می‌گردد که عدد ۵ می‌باشد و کار هندلر به پایان می‌رسد.

• روش دوم:

اگر هندلر شخصی دارای Return است، یعنی مقداری را بر میگرداند برای فراخوانی آن لازم است نام هندلر را به همراه مقدار پارامترهای آن در یک عبارت ذکر کنید. به‌عنوان مثال هندلر زیر را در نظر بگیرید:

On Sum(x,y)

T=x+y

Return T

End

اکنون جهت فراخوانی هندلر فوق، کافی است از عبارت ساده زیر کمک بگیرید:

Z=Sum(2,3)

در این روش فراخوانی نیز مقدار پارامترها همان مقادیر قبلی هستند یعنی $x=2$ و $y=3$ و درون هندلر، حاصل جمع پارامترها در متغیر محلی به نام T ذخیره شده، این نتیجه به وسیله Return برگشت داده می‌شود و درون متغیر Z قرار می‌گیرد.

مثال عملی:

مثال زیر نمونه‌ای از هندلر شخصی و فراخوانی آن را نشان می‌دهد. در این مثال ساده یک هندلر شخصی به نام Sum وجود دارد که دارای دو پارامتر به نامهای x و y بوده، عمل جمع را بر روی این دو پارامتر انجام می‌دهد و به کمک یک دکمه فراخوانی شده، جمع دو عدد ۲ و ۳ را نمایش می‌دهد.

مراحل انجام کار:

یک پروژه جدید ایجاد کنید.

جعبه ابزار را در حالت Classic قرار داده، یک دکمه در صحنه قرار دهید.

یک عنصر متنی به کمک یکی از ابزارهای Text یا Field در صحنه قرار دهید و در پنجره

Cast نام این عنصر را "Disp" تنظیم کنید.

اکنون اسکرپت رفتار زیر را تایپ کرده، آن را به دکمه بچسبانید

On Sum x,y.

T=x+y

End

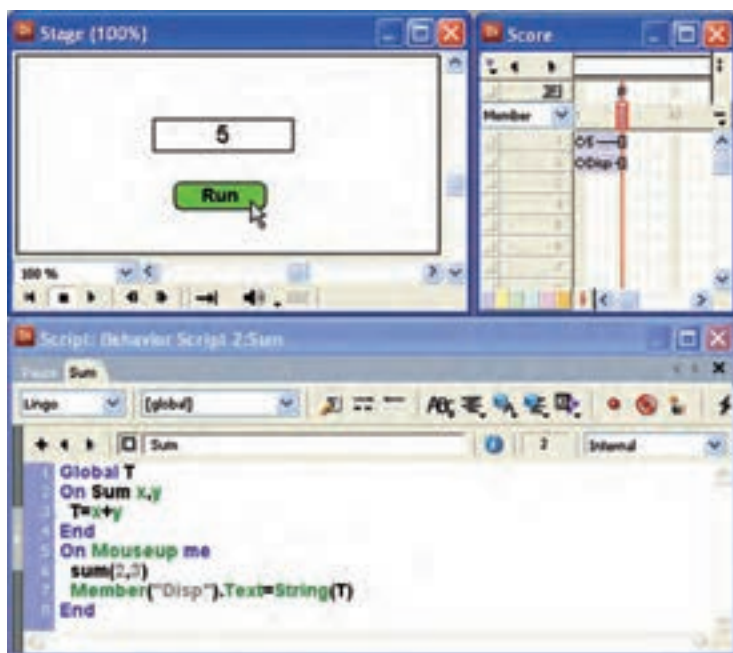
On Mouseup me

sum(2,3)

Member("Disp").Text=String(T)

End

یک دستور توقف در آخرین فریم درج کرده، آن را اجرا کنید. همانطور که ملاحظه می‌کنید با کلیک روی دکمه فرمان $\text{Sum}(2,3)$ اجرا شده، موجب فراخوانی هندلر شخصی Sum می‌گردد. به این ترتیب که مقدار 2 به پارامتر x و مقدار 3 به پارامتر y ارسال شده، حاصل جمع آنها در متغیر سراسری T قرار می‌گیرد که برابر با 5 است. آنگاه کنترل اجرا به ادامه هندلر Mouse up برگشته و مقدار عددی T را ابتدا به متن تبدیل کرده، سپس در عنصر متنی Disp نمایش می‌دهد. شکل (۸-۱۵)



شکل ۸-۱۵ نمایش مجموع دو عدد توسط یک هندلر شخصی

۸-۱۴- اصول ایجاد Movie Script

Movie Script یا اسکریپت فیلم به منظور نوشتن هندلرهایی در سطح فیلم دایرکتور به کار می‌رود. به بیان دیگر محل نوشتن هندلرهای سراسری و همچنین هندلرهایی است که رویدادهای فیلم دایرکتور را شامل می‌شوند که مهمترین آنها عبارتند از:

Start Movie: این رویداد در زمان اجرای فیلم دایرکتور و درست قبل از اینکه هد شروع به پخش اولین فریم کند، اتفاق می‌افتد و مناسب برای اجرای دستوراتی است که می‌خواهید در ابتدای شروع نمایش، آنها را اجرا کنید.

Stop Movie: این رویداد در زمانی که دایرکتور اجرای نمایش را خاتمه می‌دهد، اتفاق می‌افتد. بنابراین اگر می‌خواهید دستوراتی درست قبل از خاتمه نمایش، اجرا شوند، از این

رویداد کمک بگیرید.

جهت ایجاد یک Movie Script یک سلول خالی را در Cast انتخاب، پنجره اسکریپت را فعال کرده و از زبانه Script موجود در Property Inspector، نوع آن را Movie برگزینید. اکنون مثال شکل ۱۵-۸ را با افزودن یک Movie Script کامل می‌کنیم تا با این نوع اسکریپت بیشتر آشنا شوید:

مراحل انجام کار:

۱- یک اسکریپت جدید از نوع Movie ایجاد کرده، هندلرهای زیر را در آن تایپ کنید:

```
On startMovie  
member("Disp").text = " Start "  
End
```

```
On stopMovie  
member("Disp").text = " End "  
End
```

۲- اکنون نمایش را تست کنید. با کلیک روی دکمه اجرای نمایش عبارت Start و به محض قطع نمایش عبارت End در عنصر متنی ظاهر می‌شود.

۳- در این مرحله قصد داریم با انتقال هندلر Sum به درون اسکریپت فیلم و استفاده از فرمان Return، نمایش را بهینه‌سازی کنیم بنابراین این هندلر را از درون اسکریپت دکمه به درون اسکریپت فیلم انتقال دهید. در این حالت محتویات این دو اسکریپت باید به شکل زیر باشد:

محتویات اسکریپت فیلم:

```
On Sum (x,y)  
T=x+y  
Return T  
End  
On startMovie  
member("Disp").text="Start"  
End
```

```
On stopMovie  
member("Disp").text="End"  
End
```

محتویات اسکریپت رفتار برای دکمه RUN:

```
On Mouseup me
```

```
Z=sum(2,3)
Member("Disp").Text=String(Z)
End
```

نکته:

یکی از اصول اسکریپت نویسی در دایرکتور این است که تمامی هندلرهای شخصی را درون اسکریپت فیلم نوشته، آنها را از درون اسکریپت‌های دیگر فراخوانی کنید. اکنون دوباره نمایش را اجرا و آنرا تست کنید تا به صحت عملکرد آن پی ببرید.

۱۵-۸- لیست و کاربرد آن:

همانطور که در ابتدای فصل اشاره شد، لیست در واقع تعدادی مقدار هم نوع می‌باشد که به وسیله یک متغیر قابل نگهداری و استفاده است و جهت تعریف و نگهداری داده‌های هم نوع، مانند اسامی، مسیرها، مختصات و... به کار می‌رود. فرم کلی تعریف لیست:

```
ListName=[data1,data2,...]
```

x = [0,2,4,6,8] لیست عددی مثال ۱:

Name=["Book", "Pen", "Eraser"] لیست رشته‌ای مثال ۲:

۱۵-۸- دسترسی به داده‌های لیست:

برای دسترسی به هر یک از داده‌های موجود در لیست، کافی است نام لیست را به همراه شماره اندیس آن درون کروشه ذکر کنیم. اندیس داده‌ها از شماره یک آغاز می‌شود. دستورات زیر را درون پنجره Script امتحان کنید:

```
Mylist= [ "a", "b", "c" ]
```

```
Put Mylist[1] → "a"
```

```
Put Mylist[2] → "b"
```

```
Put Mylist[3] → "c"
```

۱۵-۸- پاک کردن مقادیر درون لیست:

جهت پاک کردن لیست My list از فرمان ساده زیر استفاده کنید:

```
Mylist = [ ]
```

۱۵-۸- مرتب‌سازی لیست‌ها:

در دایرکتور می‌توان لیست‌ها را با متد Sort() مرتب کرد. این متد، داده‌های لیست را به صورت صعودی مرتب می‌کند و فرم کلی استفاده از آن به شکل زیر است:

```
List Name.Sort()
```

مثال زیر را در پنجره Message امتحان کنید.

```
Data=[ 22,4,10,1]
```

```
Data.Sort()
```

```
Put Data
```

۴-۱۵-۸- اضافه کردن مقدار جدید به انتهای لیست:

برای اضافه کردن مقدار به داده‌های لیست، از تابع Add به شکل زیر استفاده می‌شود.

```
List Name.Add(Data)
```

مثال زیر را در پنجره Message امتحان کنید.

```
Computer =[ " Main borad","CPU","RAM"]
```

```
Computer.Add("Graphic")
```

```
Put Computer
```

۵-۱۵-۸- تغییر داده مشخصی در لیست:

```
setAt list, orderNumber, value
```

در فرم کلی فوق، list نام لیست موجود و Order Number شماره اندیسی است که می‌خواهید داده آن را با مقدار جدید value عوض کنید.

مثال زیر را در پنجره Message امتحان کنید.

```
Number =[ 10,15,17,25]
```

```
SetAt Number,3,15
```

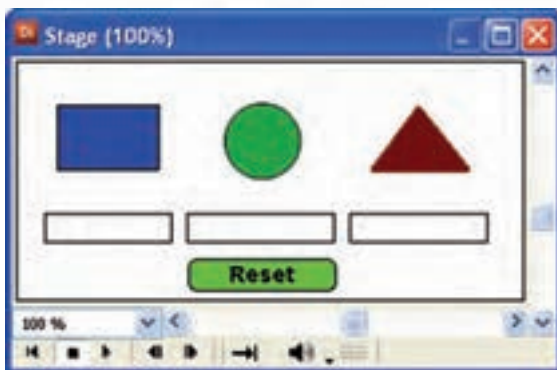
```
Put Number
```

۶-۱۵-۸- مثال عملی:

مثال زیر را اجرا تا نحوه استفاده از لیستها را در عمل تجربه کنید. در این مثال نام اشکال در یک لیست ذخیره شده، با کلیک روی هر شکل، نام آن از طریق لیست، درون فیلد متن نمایش داده خواهد شد، در ضمن کاربر با کلیک روی دکمه Reset قادر به خالی کردن فیلدها خواهد بود.

مراحل انجام کار:

- ۱- نمایش جدیدی ایجاد کنید.
- ۲- یک مستطیل، دایره و مثلث طراحی کرده، در صحنه بچینید.
- ۳- در زیر هر شکل یک کادر متن خالی قرار دهید (از ابزار Field کمک بگیرید).
- ۴- یک دکمه با عنوان Reset به صحنه اضافه کنید. شکل ۱۶-۸ صحنه طراحی شده را نشان می‌دهد.



شکل ۱۶-۸

۵- رفتار توقف هد را در آخرین فریم نمایش درج کنید(طول اسپرایت‌ها می‌تواند ۵ فریم و حتی کمتر باشد)

```
on exitframe me  
go the frame  
End
```

۶- اسکریپت‌های زیر را جهت کامل شدن نمایش تایپ کنید:
اسکریپت فیلم جهت تعریف لیست به صورت سراسری

```
Global ShapeList  
on StartMovie  
ShapeList =["Rectangle","Circle","Trianglae"]  
End
```

رفتار شکل مستطیل:

```
global shapelist  
On mouseup me  
member("name1").Text= shapelist[1]  
End
```

رفتار شکل دایره:

```
global shapelist  
on mouseup me  
member("name2").Text= shapelist[2]  
End
```

رفتار شکل مثلث:

```
global shapelist  
on mouseup me  
member("name3").Text = shapelist[3]  
End
```

رفتار دکمه Reset:

```
on mouseUp me  
member("Name1").Text = ""  
member("Name2").Text = ""  
member("Name3").Text = ""  
End
```

۷- اکنون پروژه را اجرا و آن را تست کنید.

۱۶-۸- عبارات شرطی:

همانطور که می‌دانید دستورات شرطی به دستوراتی گفته می‌شود که روند اجرای برنامه را مشروط به اجرای شرطی می‌نماید. مهمترین دستور ساخت عبارات شرطی If ... then می‌باشد که می‌توان هر نوع عبارت شرطی را با آن ایجاد کرد و به چهار فرم زیر قابل استفاده است.

فرم ۱: IF شرط then دستور

فرم ۲: IF شرط then

دستورات

End If

فرم ۳: IF شرط then

دستورات

Else

دستورات

End If

عملکرد If به این شکل است که اگر شرط برقرار باشد، دستور یا دستورات بعد از آن اجرا می‌شود. چنانچه شرط برقرار نباشد، هیچ دستوری اجرا نمی‌شود مگر آنکه شرط شامل else باشد که در این صورت دستورات بعد از else اجرا خواهد شد.

به مثالهای ساده زیر دقت کنید:

مثال ۱:

```
If n=12 then count = count + 1
```

مثال ۲:

```
If n>=12 then
```

```
Count = count + 1
```

```
Else
```

```
Sprite("result").Member.text = "Invalid Number"
```

```
End If
```

۱-۱۶-۸- یک مثال کاربردی از عملکرد دستور if:

می‌خواهیم یک پروژه ایجاد کنیم، به طوری که یک توپ بعد از سه بار حرکت متناوب (بالا و پایین) متوقف شود.
 مراحل انجام کار:
 یک توپ را طراحی و در cast قرار دهید.
 توپ را وارد صفحه کرده، در کانال ۱ از فریم ۱ تا ۱۵ قرار دهید.
 حرکت را از بالای stage به سمت پایین طراحی کنید.
 یک کپی از اسپریت گرفته، در فریم ۱۶ تا ۳۰ قرار دهید، سپس آن را معکوس کنید (Reverse) تا این مرحله یک حرکت رفت و برگشتی را ایجاد کند.
 اکنون باید یک متغیر سراسری داشته باشیم که در هر بار اجرای نمایش، مقدار آن یک واحد افزوده شود. بنابراین Movie Script ایجاد کرده نام start را برای آن تایپ کرده، کد زیر را در آن بنویسید.

```
Global n
On startMovie
n = 0
end
```

همانطور که می‌دانید رویداد start Movie در ابتدای اجرای نمایش اتفاق می‌افتد و محل مناسبی برای تعریف متغیرهای سراسری (Global) و مقدار دهی اولیه به متغیرها و یا سایر اجزای نمایش است.

اکنون در کانال script، پنجره score، روی فریم ۳۰ دو بار کلیک کنید تا پنجره Script ظاهر شود، سپس نام Script را وارد کرده، کد زیر را در آن تایپ کنید:

```
Global n
on exitFrame me
n=n+1
if n<3 then
go frame 1
else
go the frame
end if
End
```

۷- اکنون می‌توانید پروژه نهایی را تست کرده و به حرکت هد دقت کنید.

۱۷-۸- دستورات تکرار (حلقه‌ها):

کاربرد حلقه‌ها انجام عملیات تکراری موجود در یک رفتار است به عنوان مثال محاسبه مجموع اعدادی که در محدوده خاصی قرار دارند یا جا به جا کردن تعدادی اسپریت پشت سر هم به اندازه معین و... نمونه‌هایی از کاربردهای دستورات حلقه می‌باشند.

انواع حلقه‌ها در Lingo:

۱- حلقه تکرار شمارشی Repeat with

۲- حلقه شرطی Repeat while

اکنون یکایک حلقه‌های فوق را بررسی می‌کنیم:

۱-۸- حلقه تکرار شمارشی (Repeat With)

کاربرد این حلقه‌ها زمانی است که تعداد تکرار عملیات مشخص باشد. مثلاً حلقه‌ای که فقط ۱۰ بار اجرا شود. شکل کلی این حلقه به صورت زیر است:

repeat with counter = start to finish

statement(s)

end repeat

در فرم کلی فوق Counter یک متغیر شمارنده مانند I است که به ترتیب مقادیر Start تا finish در آن قرار می‌گیرد، start مقدار اولیه برای شروع حلقه و finish مقدار پایانی آن می‌باشد.

فرم فوق مربوط به حلقه افزایشی است که در آن باید مقدار اولیه حلقه کوچکتر یا مساوی مقدار پایانی آن باشد تا حلقه حداقل یک بار تکرار شود.

نکته: اگر می‌خواهید حلقه کاهشی ایجاد کنید باید به جای کلمه to از down to استفاده کنید. (بین دو کلمه down و to قرار دادن حداقل یک فاصله ضروری است).

برای بررسی این حلقه یک دکمه روی stage قرار داده، یک رفتار برای آن ایجاد کرده، دستورات زیر را تایپ کنید.

on Mouseup me

Repeat with i=1 down to 5

put i

end repeat

end

اکنون نمایش را اجرا کنید، روی دکمه کلیک کنید و نتیجه را در Message مشاهده کنید.

اکنون حلقه را به صورت زیر تغییر دهید.

on Mouseup me

Repeat with i=1 to 5

put i

end repeat

end

نتیجه را بررسی کنید.

۲-۱۷-۸- مثال کاربردی حلقه تکرار شمارشی:

در این مثال می‌خواهیم امکان شفاف کردن چندین تصویر را به‌طور هم‌زمان در اختیار کاربر قرار دهیم. برای این کار از خاصیت Blend اسپرایت‌ها کمک بگیرید و مراحل زیر را انجام دهید:

- ۱- تعداد ۴ عدد تصویر دلخواه را به ترتیب در کانالهای ۱ تا ۴ پنجره Score بچینید.
- ۲- دو عدد دکمه با برچسبهای Blend و Reset در کانالهای ۵ و ۶ قرار دهید. (راهنمایی: طول اسپرایت‌ها را ۵ فریم تنظیم کنید)
- ۳- رفتار توقف هد را در آخرین فریم نمایش درج کنید. شکل ۱۷-۸ صحنه نمایش را نشان می‌دهد.



شکل ۱۷-۸

۴- اکنون رفتارهای زیر را برای دکمه‌ها بنویسید:

رفتار دکمه Blend

```
On MouseUp me
Repeat With i=1 to 4
Sprite(i).Blend=50
End Repeat
End
```

رفتار دکمه Reset

```
On MouseUp me
```

```
Repeat With i=1 to 4  
Sprite(i).Blend=100  
End Repeat  
End
```

۵- در پایان نمایش را ذخیره و عملکرد دکمه‌ها را تست کنید.

۳-۱۷-۸- حلقه تکرار شرطی Repeat while:

نوع دیگر حلقه در لینگو، حلقه شرطی است که تعداد تکرار آن بستگی به وضعیت شرط آن دارد. چنانچه شرط برقرار باشد، دستورات تکرار می‌شوند و این تکرار تا زمانی ادامه می‌یابد که شرط نقض شود فرم کلی استفاده از دستورات شرطی به صورت مقابل است.

```
Repeat while شرط  
بلوک دستورات حلقه  
End Repeat
```

مثال‌های زیر را می‌توانید با یک دکمه امتحان کنید تا به طرز کار این حلقه بهتر پی

ببرید:

(نتیجه اجرا نمایش اعداد ۱ تا ۵ است که در پنجره Message نمایان می‌شود)

```
On Mouseup me  
i=1  
Repeat while i<=5  
Put i  
i=i+1  
End Repeat  
End
```

نکته:

همانطور که می‌دانید در حلقه Repeat With، میزان گام حلقه برابر ۱ یا ۱- است و چنانچه بخواهید مقدار تغییرات شمارنده حلقه، عددی غیر از ۱ و ۱- باشد، باید از حلقه شرطی Repeat while استفاده کنید. مثال زیر این مطلب را به خوبی نشان می‌دهد.

```
On MouseUp me  
i=50  
Repeat while I >=30  
Put i  
i=i- 5  
End Repeat  
End
```

۸-۱۸ خطایابی و رفع آن به کمک Debugger Window

به طور معمول اسکریپت نویسی با هر زبانی خالی از اشکال نبوده، ممکن است با خطاهایی مواجه شده یا عملی که مورد انتظار شماست به درستی اجرا نشود. به عنوان مثال نحوه نوشتن یک دستور (Syntax) اشتباه بوده، باعث ایجاد خطا میگردد. در این حالات لازم است زمان کافی جهت رفع خطاها (Debugging) را داشته باشید و با ابزارهای کمکی که مهمترین آنها Debugger Window می باشد، خطاها را برطرف کنید.

این پنجره دارای چندین ابزار مفید برای پیدا کردن علت خطاها در اسکریپت است و به کمک آن می توانید سرعت محل خطا را بیابید. این ابزار امکان اجرای خط به خط اسکریپت را دارا بوده، همچنین می تواند از اجرای هندلرهایی که اکنون مورد نیاز نیستند جلوگیری کند (Skip) و مقدار متغیرها را در هر لحظه نمایش داده، تغییرات سایر اشیاء مانند اسپریتها را نشان دهد.

هنگامی که خطایی در زمان کامپایل کردن یک اسکریپت رخ دهد، دایرکتور محل خطا را در پنجره Debugger Window نشان داده و پیغامی مبنی بر علت بروز خطا ظاهر می کند که با توجه به آن می توانید به رفع خطا اقدام کنید. شکل ۸-۱۸ زیر یک مثال پیغام خطا را که هنگام اجرای اسکریپت به وجود آمده، نشان می دهد.



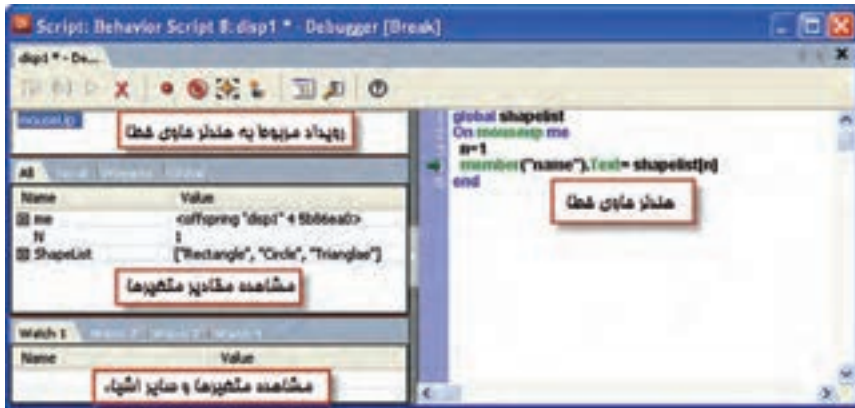
شکل ۸-۱۸ پیغام خطای دایرکتور

در سطر اول علت بروز خطا و در سطر دوم خطا را نشان می دهد و سه گزینه را در اختیار کاربر قرار می دهد که به شرح زیر می باشند:

- ۱- با کلیک روی دکمه Debug... وارد محیط Debugger Window شده، می توانید با امکانات آن اقدام به تست و خطایابی نمایید.
- ۲- با کلیک روی دکمه Script، پنجره اسکریپت فعال شده، دایرکتور سطر حاوی خطا را مشخص می کند.
- ۳- با کلیک روی دکمه Cancel عملیات خطایابی لغو می شود.

۱۹-۸- رفع خطا به کمک امکانات Debugger Window:

شکل ۱۹-۸ پنجره Debugger را در وضعیت بروز خطا نشان می‌دهد:



شکل ۱۹-۸ پنجره Debugger

در عنوان این پنجره نام و نوع اسکریپت به همراه اندیس آن نشان داده می‌شود که در شکل فوق این عنوان به صورت زیر می‌باشد:

Script:Behavior Script 8:disp1

مفهوم عنوان فوق این است که اسکریپت ۸ که در سلول شماره ۸ پنجره Score ذخیره شده، نام آن disp1 می‌باشد و دارای اشکال است.

- در اولین بخش سمت چپ، زیر نوار ابزار رویداد اسکریپت حاوی خطا را نشان می‌دهد.

- در بخش دوم می‌توانید مقدار متغیرها را چک کنید که شامل ۴ زبانه به شرح زیر است:

All: لیست تمامی متغیرهای وابسته به هندلر، که شامل متغیر مرجع me، متغیرهای محلی، سراسری و غیره است.

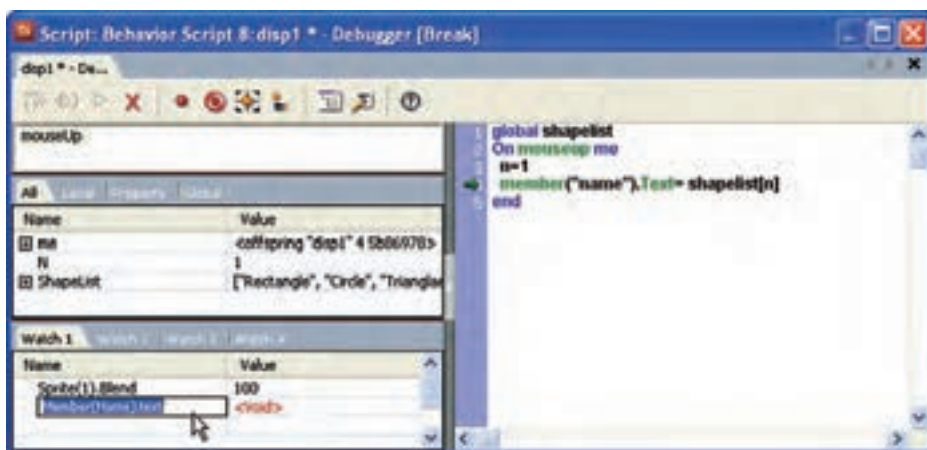
Local: لیست متغیرهای محلی وابسته به هندلر به همراه مقدار آنها

Property: لیست متغیرهای Property (مشخصه) به همراه مقدار آنها (نوع خاصی از متغیرهای دایرکتور هستند که کاربر می‌تواند آنها را تعریف و مورد استفاده قرار دهد)

Global: لیست کلیه متغیرهای سراسری وابسته به هندلر

- در بخش پایین که با زبانه‌های Watch 1 تا Watch 4 مشخص شده، امکان وارد کردن نام متغیرها یا اشیاء دیگر مانند Sprite(1) یا Member(5).text وجود دارد که امکان

طبقه بندی آنها را نیز به وسیله چهار زبانه فراهم کرده است. جهت وارد کردن نام یک متغیر کافی است در بخش سمت چپ هر ردیف دوبار کلیک کنید تا مکان نما ظاهر شده، سپس مقدار مورد نظر را تایپ کنید. شکل ۸-۲۰



شکل ۸-۲۰ دسترسی به خواص توسط زبانه‌های Watch

۸-۲۰- ارسال یک پیام به یک یا تمامی اسکریپت‌ها:

فرض کنید اسکریپت زیر به اسپرایت شماره یک (اسپرایت موجود در کانال ۱ پنجره Score) متصل شده است:

On mouseUp me

member("Name1").Text=""

member("Name2").Text=""

member("Name3").Text=""

end

On SetText

member("Title").text="Start"

End

همانطور که ملاحظه می‌کنید این اسکریپت دارای یک Event Handler، برای رویداد mouseup و یک Custom Handler به نام SetText می‌باشد که اولی جهت پاک کردن محتویات سه عنصر متنی و دومی جهت مقدار دهی به یک عضو متنی نوشته شده است. در حالت عادی جهت اجرای مدیر رویداد mouse up لازم است کاربر روی اسپرایت شماره یک کلیک کند تا عمل پاک شدن متنها اجرا شود به عبارت دیگر باید پیغام mouse up با کلیک روی اسپرایت به اسکریپت ارسال شود تا عملیات مورد نظر اجرا شوند، اما دایرکتور روش

جالب دیگری را برای اجرای این هندلر داراست که با آن می توان از طریق اسکریپت های دیگر اقدام به ارسال پیغام mouse up به این اسکریپت کرد.
برای این منظور از متدهای SendSprite() و SendAllSprites() استفاده کنید که اولی برای ارسال یک پیغام به یک اسپرایت و دومی جهت ارسال یک پیغام به کلیه اسپرایت ها کاربرد دارد.
فرم کلی استفاده:

SendSprite(spriteNameOrNum, event {, args})

SendAllSprites(spriteNameOrNum, event {, args})

آشنایی با پارامترهای این دو متد:

Sprite Name ORNum: نام اسپرایت یا شماره کانال اسپرایت

Event: نام رویداد که قبل از آن لازم است از علامت # استفاده شود.

Args: پارامترهای هندلر که اختیاری بوده، ولی در صورت وجود پارامتر در هندلر باید

مقادیر آنها به ترتیب ذکر شود.

بنابراین با توجه به توضیحات فوق جهت اجرای مدیر رویداد mouseup ذکر شده در بالا

کافی است فرمان زیر را در یک هندلر یا پنجره message بنویسید.

SendSprite (1,#mouseup)

و بالاخره جهت اجرای هندلر شخصی Set Text کافی است فرمان زیر اجرا شود:

SendSprite (1,#SetText)



زبان برنامه نویسی (اسکرپت نویسی) دایرکتور به نام Lingo شناخته می‌شود. به مجموعه‌ای از دستورات لینگو که درون یک رویداد (اداره کننده) نوشته شده و عمل خاصی را انجام می‌دهند، اسکرپت گفته می‌شود.

اصطلاحات لینگو عبارتند از:

ثابتها مانند True، رویدادها مانند KeyUp؛ عبارات که تولید کننده یک مقدار هستند، مانند x-1، مدیر رویداد که Event Handler که به‌طور مختصر به آن Handler می‌گویند و مجموعه‌ای از دستورات عملهای درون یک اسکرپت است که در قبال یک رویداد خاص مانند MouseUp اجرا می‌شود، کلمات کلیدی (Keywords) کلمات رزرو شده‌ای هستند که دارای معنی معینی هستند مانند دستور end - عملگرها (Operators) علامتهای خاصی هستند که عمل مشخصی را بر روی یک یا چند مقدار انجام می‌دهند مانند عملگر جمع (+)؛ توابع (Functions) اسکرپت‌های آماده‌ای هستند که عمل خاصی را روی داده‌های ورودی (پارامترها) انجام می‌دهند، مانند تابع $\text{sqrt}(x)$ که جذر x را برمیگرداند؛ متد که بسیار شبیه تابع می‌باشد با این تفاوت که جهت فراخوانی لازم است از طریق شیئی که متد درون آن تعریف شده، اقدام کرد؛ به کمک پارامترها می‌توان انواع متفاوتی از مقادیر را به توابع، متدها و اسکرپت‌ها ارسال نمود؛ خصوصیات (Properties) کلمات رزرو شده‌ای هستند که بیانگر ویژگی‌های متفاوت یک شیء مانند یک اسپریت یا یک عضو متنی می‌باشند، مانند Width؛ دستورات (Statements) فرمانهایی به شکل صحیح هستند که دایرکتور می‌تواند آنها را اجرا کند مانند "menu" go؛ از متغیرها جهت نگهداری و به‌روز کردن مقادیر استفاده می‌شود، مانند $f=1$ - لیست در واقع تعدادی مقدار هم نوع می‌باشد که در یک متغیر قابل نگهداری و استفاده است، مانند $n=[25,34,77]$

آشنایی با گرامر لینگو:

لینگو به حروف بزرگ یا کوچک حساس نیست، هر دو عبارت (1)SPRITE و (1)Sprite یک مفهوم را شامل می‌شوند. برای نوشتن توضیحات از دو علامت خط اتصال (کاراکتر منحنی) چسبیده به هم استفاده کنید.

پرانتهز دارای دو کاربرد متفاوت است:

- ۱- در توابع و متدها جهت تعیین پارامتر
- ۲- در عبارات ریاضی جهت تعیین اولویت

برای دسترسی به خاصیت یا متد مشخصی از یک شیء، از گرامر نقطه‌ای استفاده می‌شود. در این گرامر ابتدا نام شیء سپس یک نقطه جهت جدا کردن، و در انتها نام یک خاصیت

(property) یا متد (Method) را ذکر می‌کنیم. مانند member(1).text پنجره Message ابزار مناسبی جهت تایپ دستورات لینگو و مشاهده نتایج آنها می‌باشد. در دایرکتور چهار نوع اسکریپت وجود دارد که عبارتند از:

۱- Behavior (رفتار)

۲- Movie Script (اسکریپت فیلم)

۳- Cast Member Script (اسکریپت عضو)

۴- Parent Script (اسکریپت والد)

به کمک پنجره Script می‌توانید اسکریپت‌های مورد نیاز خود را بنویسید یا آنها را ویرایش کنید.

اگر یک متغیر درون یک هندلر تعریف شود آن متغیر محلی بوده و فقط در همان هندلر قابل دسترس خواهد بود، اما متغیرهای سراسری در اولین خط پنجره script و در ناحیه Global تعریف می‌شوند و در سایر روالها نیز قابل دسترس می‌باشند.

به‌طور کلی عملگرهای موجود در لینگو عبارتند از:

۱- عملگرهای ریاضی

۲- عملگرهای مقایسه‌ای

۳- عملگرهای منطقی

۴- عملگرهای رشته‌ای

On و End کلمات کلیدی هستند که به ترتیب سطر آغاز و پایان یک مدیر رویداد را مشخص می‌کنند.

در حالت کلی دو نوع پیام در دایرکتور وجود دارد که عبارتند از:

۱- پیامهای سیستم (System Messages)

۲- پیامهای شخصی (Custom Messages)

Movie Script یا اسکریپت فیلم به‌منظور نوشتن هندلرهایی در سطح فیلم دایرکتور به کار می‌رود، مانند StartMovie

در فرمان شرطی If اگر شرط برقرار باشد، دستور یا دستورات بعد از آن اجرا می‌شود؛ چنانچه شرط برقرار نباشد، هیچ دستوری اجرا نمی‌شود، مگر آنکه شرط شامل else باشد که در این صورت دستورات بعد از else اجرا خواهد شد.

انواع حلقه‌ها در Lingo عبارتند از ۱- حلقه تکرار شمارشی Repeat with ۲- حلقه شرطی Repeat while

پنجره Debugger Window دارای ابزارهایی برای پیدا کردن علت خطاها و اجرای خط به خط اسکریپت‌هاست. همچنین می‌تواند از اجرای هندلرهایی که اکنون مورد نیاز نیستند، جلوگیری کند (Skip) و مقدار متغیرها را در هر لحظه نمایش داده، تغییرات سایر اشیاء مانند

اسپرایت‌ها را نشان دهد.
به‌وسیلهٔ متدهای `SendSprite()` و `SendAllSprites()` می‌توان یک پیغام را به یک یا
تمامی اسپرایت‌ها ارسال نمود.

Learn in English

Global variables in Lingo

In Lingo, variables are considered local by default, and you do not need to precede the variable name with any keyword. To declare a global variable, you must precede the variable with the keyword `global`.

If you declare a global variable at the top of a script and before any handlers, the variable is available to all handlers in that specific script. If you declare a global variable within a handler, the variable is available only to that handler; however, if you declare a global variable with the same name within two separate handlers, an update to the variable's value in one handler will also be reflected in the variable in the other handler.



واژه نامه تخصصی	
Border	حاشیه
Comments	توضیحات
Constant	ثابت
Counter	شمارنده
Custom	دلخواه
Debugger	اشکال یابی
Explorer	جستجوگر
Expression	عبارت
Function	تابع
Global	سراسری
Local	محلی
Method	روش
Operator	عملگر
Parent	والد
Push	نشاندن
Put	گذاشتن
Recompile	کامپایل مجدد
Repeat	تکرار
Return	برگشتن
Send	فرستادن
Sensitive	حساس
Shadow	سایه
Sort	مرتب کردن
Statement	دستور
Syntax	ترکیب
Variable	متغیر

خودآزمایی:

- ۱- مفهوم اسکریپت را توضیح داده، زبانهای مورد استفاده جهت اسکریپت نویسی دایرکتور را نام ببرید.
- ۲- مدیر رویداد چیست، آنرا با مثالی شرح دهید.
- ۳- روش دسترسی به خصوصیات و یا متدهای یک شیء در لینگو چگونه است؟ با چند مثال شرح دهید.
- ۴- کاربردهای پنجره Message را ذکر کنید.
- ۵- نحوه تعریف متغیرهای محلی و سراسری را با چند مثال ذکر کنید و بیان کنید نوع متغیر در لینگو چگونه مشخص می‌شود؟
- ۶- اصول ایجاد و فراخوانی یک Custom Handler را با مثالی شرح دهید.
- ۷- Movie Script چیست؟ کاربردهای آنرا ذکر کنید.
- ۸- لیست، نحوه تعریف آن و دسترسی به اجزای آنرا با مثالی توضیح دهید.
- ۹- انواع حلقه تکرار را به همراه کاربردهایشان بیان کنید.
- ۱۰- کاربرد Debugging Window را تشریح کنید.

پرسش‌های چهار گزینه‌ای

۱- کدام گزینه روش صحیح دسترسی به خواص اسپرایت را نشان می‌دهد؟

الف) Sprite(1).Height

ب) Sprite(1)-Height

ج) Sprite(1):Height

د) Sprite(1)_Height

۲- برای نوشتن توضیحات از کدام علامت استفاده می‌شود؟

الف) -

ب) --

ج) "

د) '

۳- اگر بخواهیم یک دستور طولانی را در دوسطر بنویسیم، لازم است در انتهای سطر اول از علامت..... استفاده شود.

الف) \

ب) //

ج) -

د) \

- ۴- جهت دسترسی به پاراگراف دوم عضو متنی که در سلول شماره یک کتابخانه Cast ذخیره شده است، کدام فرمان صحیح است؟
الف) Member[1].Paragraph[2]
ب) Member(1).Paragraph[2]
ج) Member(1).Text.Paragraph[2]
د) Member(1).Paragraph(2)
- ۵- کار فرمان Put Sprite(1).Loch چیست؟
الف) نمایش مختصات عمودی اسپرایت
ب) نمایش اندازه عرض اسپرایت
ج) نمایش اندازه ارتفاع اسپرایت
د) نمایش مختصات افقی اسپرایت
- ۶- کلیدهای فوری جهت باز کردن پنجره Script کدام است؟
الف) Ctrl+0
ب) Ctrl+1
ج) Shift+0+Ctrl
د) Ctrl+T
- ۷- کدامیک از نماهای موجود در پنجره Script امکان دسترسی سریع به خصوصیات و متدها را فراهم آورده، در تایپ سریع دستورات کمک می‌کند؟
الف) Script Browser
ب) Script View
ج) Dictionary View
د) Dictionary Browser
- ۸- کدام متغیر عددی اعشاری است؟
الف) $a = 1$
ب) $a = 1.0$
ج) $a = 1.2000$
د) موارد ب و ج
- ۹- جهت تبدیل مقادیر عددی به رشته از کدام تابع باید استفاده نمود؟
الف) float()
ب) string()
ج) Integer()
د) Real()
- ۱۰- کدامیک از رویدادهای زیر قبل از توقف فیلم دایرکتور (Movie) اتفاق می‌افتد؟
الف) StopMovie
ب) EndMovie
ج) StartMovie
د) PauseMovie
- ۱۱- حاصل عبارت $30/4$ کدامست؟
الف) 7
ب) 7.5
ج) 7.5000
د) موارد ب و ج

۱۲- با توجه به هندلر داده شده کدام گزینه فراخوانی صحیح آنرا نشان می‌دهد؟

On Mul (a,b)

Return a*b

End

Mul (3,4) = C (الف)

Mul 3,4 (ب)

C = Mul (3,4) (ج)

C = Mul (3*4) (د)

۱۳- با توجه به تعریف $L = ["25", "39", "72"]$ حاصل عبارت $L[0]$ کدام گزینه است؟

الف (25)

ب) "25"

ج) " 72 "

د)پیغام خطا

۱۴- پس از اجرای اسکریپت مقابل مقدار Str کدام است؟

Data = ["Iran", "-Tehran", "-Enghelab"]

Str = ""

Repeat With i =1 to 3

Str = str&Data[i]

End Repeat

الف (Iran-Tehran-Enghelab)

ب) Iran-Tehran-Enghelab

ج) -Iran-Tehran-Enghelab

د) -Iran-Tehran-Enghelab

۱۵- اسکریپت مقابل چه عملی را انجام می‌دهد (با فرض اینکه اسپرایت‌های 1 تا 10 از نوع

متنی هستند)؟

Repeat With i=10 down to 1

Sprite(i).LocH=50

End Repeat

الف) کلیه اسپرایت‌ها از سمت چپ ترازبندی می‌شوند

ب) کلیه اسپرایت‌ها از سمت راست ترازبندی می‌شوند

ج) عرض کلیه اسپرایت‌ها برابر 50 می‌شود

د) ارتفاع کلیه اسپرایت‌ها برابر 50 می‌شود

- ۱۶- جهت ارسال پیام Clear به اسپرایت شماره ۳ کدام گزینه صحیح است؟
الف) SendSprite (Sprite 3,Clear)
ب) SendAllSprite (3,#Clear)
ج) SendSprite (3,#Clear)
د) SendSrite (3,"Clear")

- 17-Which Keyword is used for declare Variables that are available to Several Handlers?
a) Var
b) Top
c) Global
d) Local

کارگاه چند رسانه‌ای

- ۱- به کمک یک Custom Handler نمایشی ترتیب دهید که کاربر بتواند وضعیت زوج یا فرد بودن مقدار صحیح ورودی را چک کند.
- ۲- با توجه به لیست داده شده، اسکریپتی بنویسید که مجموع اعداد لیست را در یک کادر متنی نمایش دهد؟
Data = [10,20,30,40,50]
- ۳- نمایشی آماده کنید که دارای یک متن با حداقل سه پاراگراف بوده، امکانات زیر را در اختیار کاربر قرار دهد:
الف) شماره ردیف پاراگراف را دریافت کرده، آن را نمایش دهد.
ب) شماره ردیف پاراگراف را به همراه شماره ردیف یک کلمه دریافت کرده، آن کلمه را نمایش دهد.
- ۴- در صحنه ۴ تصویر قرار داده و مقدار Blend آنها را برابر 30 تنظیم کنید، سپس رفتارهایی بنویسید که با قرار گرفتن ماوس بر روی هر تصویر بتدریج مقدار Blend برابر 100 شود و با خارج شدن ماوس از روی تصویر مقدار Blend به حالت اولیه برگردد. (راهنمایی: از حلقه شمارشی کمک بگیرید)



فصل نهم

توانایی توزیع پروژه به فرمت‌های مختلف

هدفهای رفتاری:

- در پایان این فصل، از هنرجو انتظار می‌رود:
- انتشار و مفهوم آن‌را توضیح دهد.
- فرمت‌های مختلف انتشار در دایرکتور را بشناسد و تنظیمات آن‌را انجام دهد.
- کاربرد فایل محافظت شده را توضیح داده، نحوه ساخت یک فایل محافظت شده را عملاً انجام دهد.
- روش استفاده از فایل محافظت شده را شرح دهد.

مقدمه:

مفهوم انتشار، گرفتن خروجی از یک نمایش دایرکتور، برای استفاده در محیط‌های مختلف مانند نمایش اجرایی برای ویندوز یا مکینتاش، نمایش به شکل HTML برای محیط وب یا کاربردهای دیگر است. البته دایرکتور امکان گرفتن خروجی به صورت ترتیبی از تصاویر یا فرمت ویدئویی را نیز در اختیار کاربر قرار داده است که برای این منظور لازم است از فرمان File/Export استفاده کنید.

جهت گرفتن خروجی با تنظیمات پیش فرض نرم‌افزار که برای گرفتن خروجی به صورت فایل اجرایی (exe) در مسیر فایل اصلی دایرکتور تنظیم شده است کافی است فرمان File/ublish را اجرا کنید.

نکته: برای اجرای دستور Publish می‌توان از دکمه مربوطه در نوار ابزار اصلی دایرکتور نیز استفاده کرد. (شکل ۹-۱)



شکل ۹-۱ دکمه Publish

مثال ۱: هدف: باز کردن یک پروژه آماده موجود در نرم‌افزار و ایجاد یک خروجی بصورت اجرایی با تنظیمات پیش فرض
مراحل کار

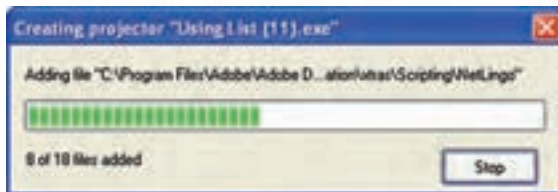
۱- از مسیر نصب دایرکتور فایل Imaging را باز کنید. چنانچه دایرکتور در C: نصب شده باشد مسیر کامل آن بصورت زیر است:

C:\program files\Adobe\Adobe Director11\ Configuration\Lingo\Imaging.dir

این نمونه آموزشی که توسط شرکت سازنده در دایرکتور تعبیه شده است، یک محیط نقاشی ساده جهت آموزش زبان Lingo می‌باشد که توسط آن می‌توان اشکال ساده را بر روی صحنه ترسیم نمود، به عنوان نمونه جهت ترسیم یک خط کافیسست در نقطه اول کلیک کرده، و در پایان بر روی نقطه دوم نیز کلیک کنید تا یک خط بین دو نقطه ایجاد شود.

۲- بر روی دکمه Publish نوار ابزار کلیک کنید (شکل ۹-۱) تا فایل اجرایی با تنظیمات پیش فرض ایجاد شده و بلافاصله اجرا شود، اکنون آنرا تست کنید.

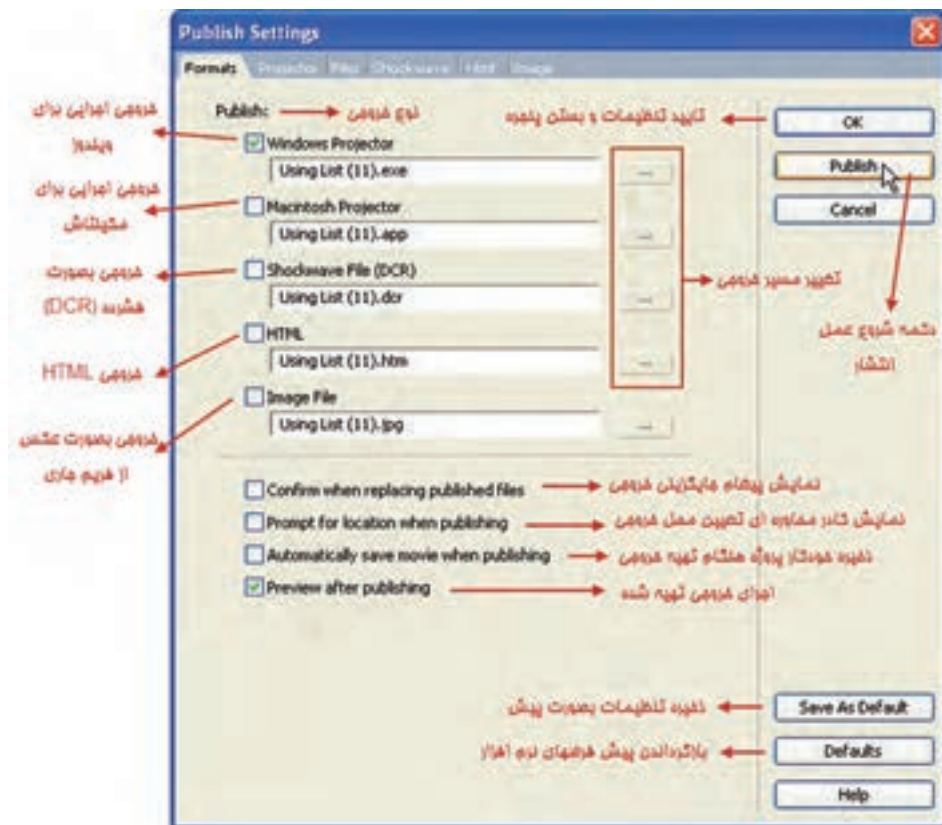
نکته: چنانچه تنظیمات پیش فرض موجود در (Publish Setting) را دستکاری نکرده باشید، فایل اجرایی توسط دکمه Close پنجره و یا دکمه Esc صفحه کلید بسته می‌شود. با اجرای فرمان Publish پردازش جهت تولید فایل اجرایی آغاز شده (شکل ۹-۲) و دایرکتور پس از ایجاد فایل مربوطه، آنرا اجرا می‌کند.



شکل ۲-۹ پردازش جهت تولید فایل اجرایی

۹-۱- تنظیمات انتشار Publish Setting:

جهت دسترسی به تنظیمات انتشار، لازم است از کادر محاوره‌ای Publish Setting که از طریق منوی File قابل دسترسی است، استفاده کنید. این کادر محاوره‌ای به شما امکان می‌دهد نمایش خود را با فرمت‌های مختلف انتشار دهید. پس از انتخاب این گزینه پنجره آن مطابق شکل ۳-۹ نمایان می‌شود:



شکل ۳-۹ پنجره تنظیمات Publish

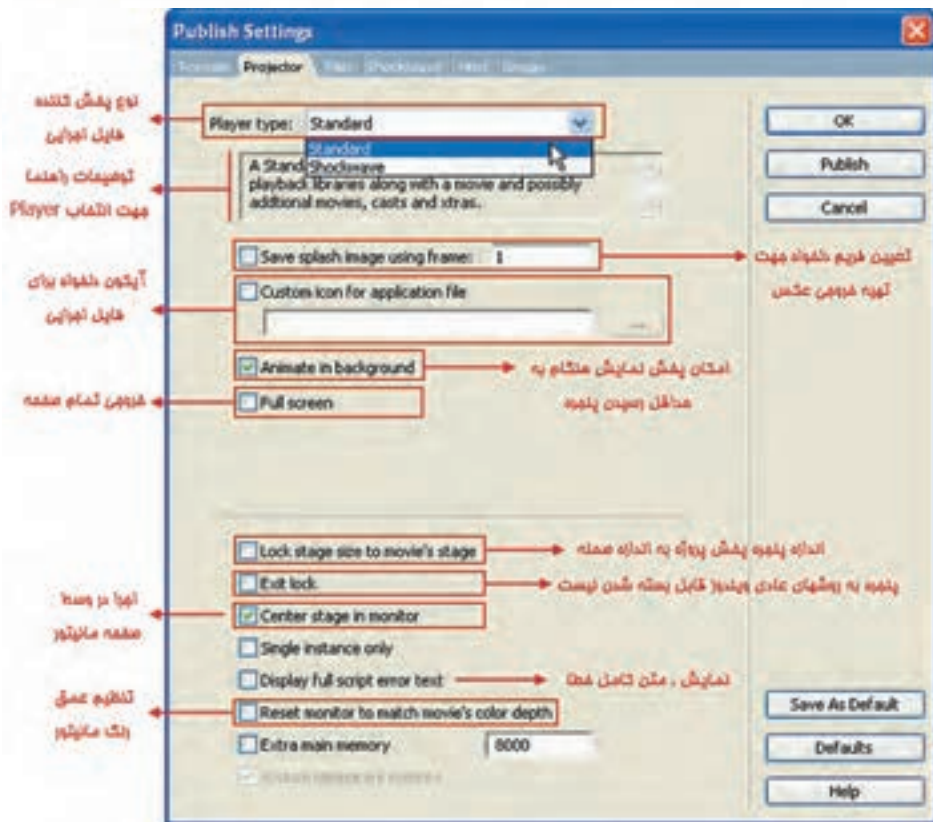
۹-۱-۱- آشنایی با زبانه Formats

کاربرد	گزینه‌های مهم زبانه Formats
فایل اجرایی با پسوند Exe برای سیستم عامل ویندوز	Windows Projector
خروجی اجرایی برای کامپیوتر Macintosh	Macintosh Projector
فایل DCR یک فایل فشرده شده و کم حجم برای پخش در مرورگر اینترنت است که توسط یک فایل HTML بارگزاری شده و به این ترتیب امکان دسترسی به نمایش در اینترنت فراهم می‌گردد. البته در صورت داشتن ShockWave Player می‌توان آنرا به طور مجزا پخش کرد.	ShockWave File (DCR)
گرفتن خروجی برای وب	HTML
توسط این گزینه می‌توان یک فریم خاص از نمایش را به صورت فایل تصویری با فرمت Jpg ذخیره کرد.	Image File
چنانچه خروجی از قبل وجود داشته باشد، پیغامی مبنی بر جایگزینی آن نمایش می‌دهد.	Confirm when replacing published files
نمایش کادر محاوره‌ای جهت تعیین محل ذخیره‌سازی فایل خروجی	Prompt for location when publishing
ذخیره خودکار پروژه هنگام تهیه خروجی	Automatically save movie when publishing
پس از تهیه خروجی آنرا اجرا می‌کند.	Preview after publishig

جدول ۹-۱- گزینه‌های مهم زبانه Formats

۹-۱-۲- تنظیمات Projector

جهت دسترسی به تنظیمات فایل اجرایی وارد زبانه Projector شوید که گزینه‌های زیر را در اختیار شما قرار می‌دهد. (شکل ۹-۴):



شکل ۹-۴ تنظیمات زبانہ Projector

کاربرد	گزینه‌های مهم زبانہ Projector
نوع پخش کننده فایل اجرایی که در حالت عادی بر روی استاندارد (Standard) قرار دارد ولی گزینه ShockWave نیز در لیست آن مشاهده می‌شود که نوعی پخش کننده مخصوص فایل‌های DCR (کم حجم برای اینترنت) می‌باشد و باید بر روی سیستم نصب شده باشد.	Player type
توسط این گزینه می‌توان یک آیکون دلخواه برای فایل اجرایی انتخاب نمود.	Custom icon for application file

کاربرد	گزینه‌های مهم زبانه Projector
فعال بودن این گزینه امکان پخش نمایش را در هنگام به حداقل رساندن پنجره آن می‌دهد، در غیر این صورت پخش نمایش با Minimize شدن آن به طور موقت متوقف خواهد شد	Animated in Background
از این گزینه زمانی استفاده کنید که اندازه نمایش از اندازه مانیتور کوچکتر بوده و قصد دارید خروجی بصورت تمام صفحه نمایش داده شود. در این حالت یک حاشیه به رنگ زمینه Stage دور تا دور نمایش را می‌پوشاند.	Full Screen
باعث می‌شود که اندازه پنجره پخش فایل به اندازه اصلی نمایش تنظیم شده و قابل تغییر نباشد.	Lock stage size to movie's stage
در حالت عادی پنجره پخش فایل را می‌توان با دکمه Close موجود در نوار عنوان و یا دکمه Esc صفحه کلید بست، با علامتدار کردن این گزینه این قابلیت از نمایش گرفته شده و فقط توسط رفتارها می‌توان آن را بست.	Exit Lock
بطور پیش فرض فعال بوده و باعث باز شدن پنجره پخش پروژکتور در مرکز صفحه نمایش می‌گردد.	Center stage in monitor
در صورت بروز خطا در اسکریپت‌های نمایش، متن کامل خطا را نمایش می‌دهد.	Display Full Script Error Text
تنظیم عمق رنگ مانیتور را با تنظیم عمق رنگ نمایش در حال اجرا یکسان می‌سازد.	Reset Monitor To Match Movie's Color Depth

جدول ۹-۲ گزینه‌های مهم زبانه Projector

۹-۱-۳- تنظیمات ShockWave

برای دسترسی به تنظیمات فایل ShockWave به زبانه آن در پنجره publish setting مراجعه کنید تا تنظیمات زیر را در اختیار شما قرار دهد (شکل ۵-۹):

مثال ۲: هدف: باز کردن یک پروژه آماده موجود در نرم‌افزار و ایجاد یک خروجی بصورت اجرایی با تنظیمات زیر:

- در حالت عادی کاربر قادر به بستن پروژه نهایی نباشد.
- فایل اجرایی بصورت تمام صفحه اجرا شود.
- در صورت Minimize شدن به کار خود ادامه دهد.

مراحل کار

۱- از مسیر نصب دایرکتور فایل Sound Control را باز کنید. چنانچه دایرکتور در C: نصب شده باشد مسیر کامل آن بصورت زیر است :

C:\program files\Adobe\Adobe Director11\ Configuration\Learning\Lingo\ Sound Control.dir

این نمونه آموزشی که توسط شرکت سازنده در دایرکتور تعبیه شده است یک کنترل کننده صدای چند کاناله می‌باشد که دارای امکانات و ظاهر بسیار جذابی است و می‌توانید آنرا تست کنید.

- ۲- پروژه را در محل دیگری مجدداً ذخیره کنید تا نسخه اصلی آن بدون دستکاری بماند.
- ۳- یک دکمه خروج دلخواه بر روی صحنه اضافه کنید تا کاربر بتواند از پروژه خارج شود. (شکل ۵-۹)



شکل ۵-۹ پروژه Sound Control به همراه دکمه خروج

۴- کادر محاوره‌ای تنظیمات Projector را باز کرده و گزینه‌های Full Screen - Animated in Background و Exit Lock را علامت‌دار کنید و در نهایت با

زدن دکمه Publish و تایید آن خروجی اجرایی تهیه کرده و آنرا تست کنید. همانطور که مشاهده می کنید پنجره پروژه در وسط مانیتور باز می شود. پخش صدا را آغاز کرده و پنجره را Minimize کنید، ملاحظه خواهید کرد که پروژه به کار خود ادامه می دهد. حال سعی کنید پنجره را توسط دکمه Close آن ببندید، این کار عملی نمی باشد و تنها راه بستن آن استفاده از دکمه خروج تدارک دیده شده است.



شکل ۶-۹ تنظیمات زبانه Shockwave

کاربرد	گزینه‌های مهم زبانه ShockWave
به منظور فشرده‌سازی تصاویر به صورت Standard دایرکتور و یا JPEG که در این حالت توسط اسلایدر می‌توان میزان فشرده‌سازی را تعیین نمود.	Image Compression

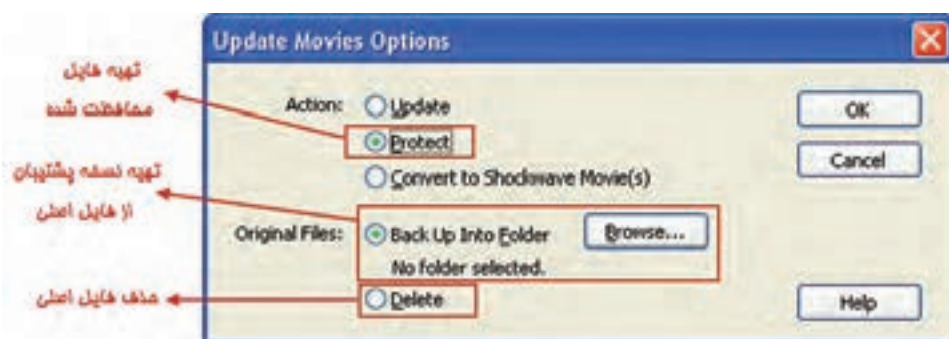
کاربرد	گزینه‌های مهم زبانه ShockWave
<p>جهت فشرده‌سازی اصوات: برای این منظور باید گزینه Compression Enabled را تیک زده و از لیست پایین آن نرخ پخش صوت را بر حسب کیلو بیت بر ثانیه تعیین کرد. در صورتی که می‌خواهید صدای استریو را به صدای مونو تبدیل کنید باید گزینه Convert Stereo to Mono را علامتدار کنید.</p>	<p>Audio Compression</p>

۹-۲- ایجاد فایل محافظت شده (DXR):

دایرکتور می‌تواند از فایل اصلی شما یک فایل محافظت شده با پسوند DXR بسازد (پسوند DIR به DXR و پسوند CST به CXT تبدیل می‌شود)، به این مفهوم که این فایل قابل ویرایش نخواهد بود. به‌عنوان مثال نمایش اصلی به یک نمایش فرعی متصل است و شما که می‌خواهید یک پروژکتور از فایل اصلی ایجاد کنید، بهتر است فایل فرعی را محافظت کنید (Protect) تا بقیه افراد نتوانند به محتویات نمایش دسترسی داشته باشند و قابلیت‌های موجود در آن را تغییر دهند.

روش کار:

۱- از منوی Xtras گزینه Update Movie را انتخاب کنید تا کادر محاوره‌ای آن مطابق شکل ۹-۷ باز شود.



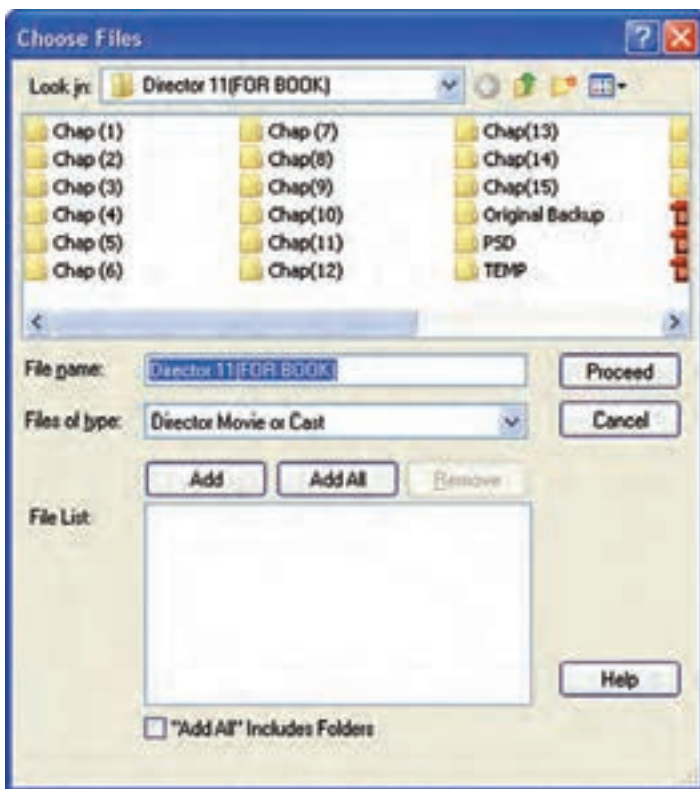
شکل ۹-۷ تهیه فایل محافظت شده

۲- از بخش Action دکمه رادیویی Protect را برگزینید.

۳- در بخش Original Files گزینه Back Up Into Folder و سپس دکمه ...Browse را

جهت انتخاب یک پوشه، به منظور گرفتن پشتیبان از فایل‌های اصلی انتخاب کنید. (در کادر محاوره‌ای Select Folder for original files پس از انتخاب مسیر مورد نظر روی دکمه Select Folder کلیک کنید).

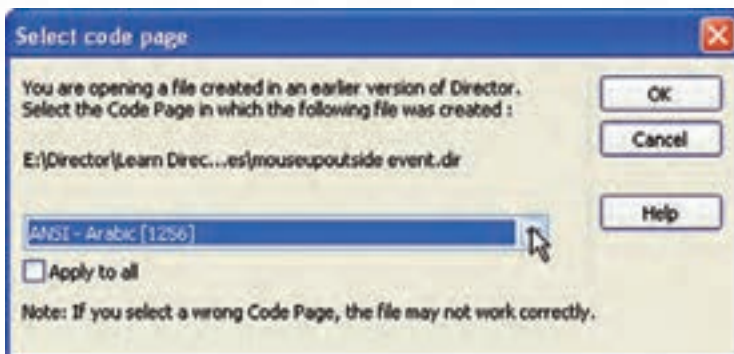
۴- با تایید کادر محاوره‌ای شکل ۷-۹ کادر محاوره‌ای جدید Choose Files مطابق شکل ۸-۹ ظاهر می‌شود. در این کادر محاوره‌ای فایل‌های اصلی را که می‌خواهید آنها را به حالت محافظت شده، درآوردید انتخاب و به File List اضافه کنید و روی دکمه Proceed کلیک کنید:



شکل ۸-۹

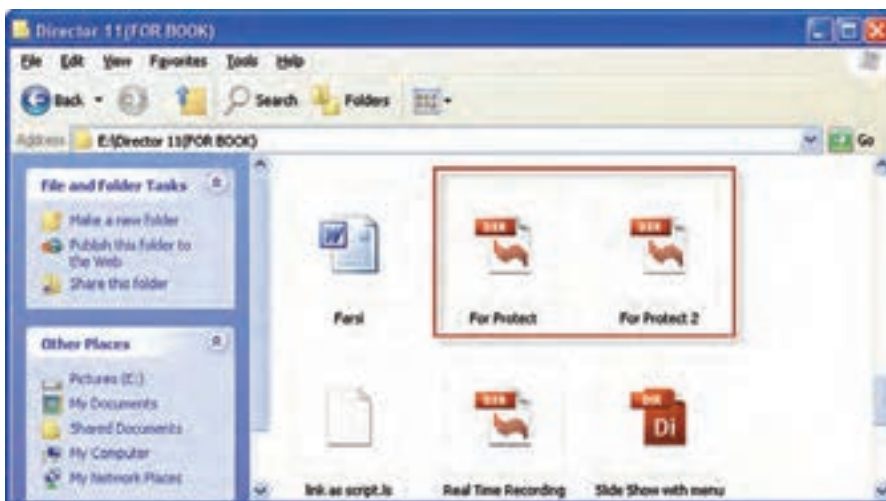
۱- پیغامی مبنی بر انصراف یا تایید ادامه عملیات ظاهر می‌شود. بنابراین روی Continue کلیک کنید.

۲- در آخرین مرحله لازم است در کادر محاوره‌ای Code Page گزینه مناسب را انتخاب و آن را تایید کنید. (شکل ۹-۹)



شکل ۹-۹

۳- اکنون اگر مسیر فایل‌های اصلی را باز کنید، مشاهده خواهید کرد که نوع فایلها و آیکن آنها تغییر کرده است. پسوند فایل‌های محافظت شده DXR می‌باشد که قابل استفاده بوده اما امکان ویرایش آنها وجود ندارد. فایل‌های اصلی در کشوی تعیین شده در مرحله سه ذخیره می‌شوند و می‌توانید به آنها دسترسی داشته باشید. (شکل ۹-۱۰ دو فایل محافظت شده را نشان می‌دهد).



شکل ۹-۱۰

۳-۹- نحوه استفاده از فایل محافظت شده:

جهت استفاده از نمایش حفاظت شده، کافی است آن را در کشوی پروژه کپی کرده، سپس از درون پروژه اصلی به کمک یک فرمان لینگو آن را فراخوانی کنید. شکل کلی فراخوانی و

اجرای یک نمایش (محافظت شده یا محافظت نشده) از درون یک نمایش دیگر به شرح زیر می‌باشد:

go to movie "Path\Filename"
مثال go to movie "project.dxr"

در مثال فوق فایل در مسیر جاری پروژه قرار دارد. بنابراین نیازی به نوشتن مسیر کامل پروژه نیست، اما اگر این فایل درون یک زیرشاخه موجود در مسیر پروژه قرار دارد باید نام زیر شاخه نیز ذکر شود مانند "Main\Project.dxr" و بالاخره اگر فایل در مسیر کاملاً متفاوتی ذخیره شده است، باید مسیر کامل آن ذکر شود.

خلاصه مطالب

یک ویدئوی دیجیتالی در واقع یک انیمیشن از تصاویری است که بطور زنده به همراه صدا ضبط شده‌اند و توسط بازپخش آن می‌توان صحنه را بطور زنده بازسازی نمود. وارد کردن فایل‌های ویدئویی همانند سایر رسانه‌هاست اما با این تفاوت مهم که Cast Member ایجاد شده همواره به فایل اصلی لینک می‌شود.

جهت وارد کردن ویدئو بدرون صحنه کفایست یک اسپرایت به طول حداقل یک فریم از آن ایجاد کنید.

دایرکتور قادر به پخش فایل‌های ویدئویی AVI و QuickTime بصورت DTS(Direct To Stage) می‌باشد که در این وضعیت درایورهای پخش و کنترل ویدئو، در بهترین حالت ممکن آغاز به پخش فایل می‌کنند.

جهت پخش و کنترل ویدئوی QuickTime، لازمست نرم افزار QuickTime بر روی سیستم نصب شود.

دسترسی به تنظیمات ویدئوی QuickTime از طریق زبانه آن در Property Inspector امکان پذیر است.

جهت پخش و کنترل فایل‌های Windows Media، دایرکتور به هیچگونه درایور اضافی نیاز ندارد زیرا درایور پیش فرض این نوع فایل، قبلاً توسط شرکت مایکروسافت در ویندوز تعبیه شده است.

با انتخاب ویدئوی WindowsMedia در پنجره Cast و یا اسپرایت ساخته شده از آن در صحنه می‌توان به خصوصیات این ویدئو از طریق زبانه WindowsMedia موجود در Property Inspector دسترسی داشت.

لینگو دارای چندین متد کاربردی آسان جهت کنترل ویدئوی WindowsMedia است که

لازمست آنها را توسط اسپریت ویدئو مورد استفاده قرار داد مانند متد Play() که باعث پخش ویدئو می‌گردد، همچنین این نوع رسانه دارای چندین خصوصیت کاربردی است که از طریق عضو ویدئویی (Member) قابل دسترسی و ویرایش می‌باشد مانند Loop که یک مقدار منطقی است و تعیین می‌کند آیا ویدئو در وضعیت تکرار باشد یا خیر.

در دایرکتور به آسانی می‌توان از فایل‌های صوتی و ویدئویی RealMedia استفاده نمود به شرط آنکه نرم افزار RealPlayer و یا نسخه‌های مشابه آنرا نصب کنید .

با انتخاب عنصر RealMedia در پنجره Cast و یا اسپریت ساخته شده از آن در صحنه می‌توان به خصوصیات این ویدئو از طریق زبانه RealMedia موجود در Property Inspector دسترسی داشت .

دایرکتور امکان استفاده از محتویات DVD را داراست و بطور پیش فرض مسیر درایو DVD سیستم را به عنوان مسیر جستجوی محتویات DVD در نظر می‌گیرد و در صورت آماده بودن دیسک حاوی DVD Video آنرا باز کرده و نمایش می‌دهد ، اما این امکان وجود دارد که مسیری پیش فرض را به یکی از کسوه‌های دیسک سخت تغییر داده و به ویدئوی کپی شده بر روی دیسک سخت نیز دسترسی پیدا کرده و آنرا نیز در صحنه نمایش داد.

- نیازمندی‌های سیستم جهت پخش صحیح ویدئوی DVD :

۱- درایو DVD

۲- نرم افزار DVD Player

۳- DVD Decoder مناسب که قابلیت DirectShow را دارا باشد مانند WinDVD-ATI

DVD و یا Nvidia DVD

دایرکتور بکمک فرمان DVD → Window می‌تواند عمل پیوند به محتویات DVD را اجرا کند.

Learn in English

Using default Publish settings

To create Shockwave content, use the File > Publish command. The default setting creates a Projector file with Preview enabled.

Select File > Publish.

Save your movie if prompted to do so.

Director creates and automatically starts a Projector version of your movie.

If your movie needs Xtra extensions that fall beyond the range of the default publish settings, (for example, Windows Media) you will be prompted to add them.

واژه نامه تخصصی	
Animated	سرزنده
Compression	فشرده‌گی
Dimensions	مختصات
Context	زمینه
Expand	گسترش دادن
Export	صادر کردن
Percentage	درصد
Preserve	محافظت کردن
Proceed	اقدام کردن
Progressive	پیشرو
Projector	پرژکتور، نورافکن
Protect	حفاظت کردن
Stretch	منبسط شدن

خودآزمایی:

- ۱- انواع فرمت‌های خروجی قابل انتشار در دایرکتور را شرح دهید.
- ۲- روش دسترسی به تنظیمات انتشار (Publish Setting) را شرح دهید.
- ۳- کدام زبانه در کادر محاوره‌ای Publish Setting مربوط به تنظیمات فایل اجرایی است، دو نمونه از این تنظیمات را نام ببرید؟
- ۴- منظور از فایل محافظت شده چیست، کاربرد آن را شرح دهید.

پرسش‌های چهار گزینه‌ای

- ۱- با کدامیک از فرمانهای زیر می‌توان عمل انتشار (Publish) را اجرا کرد؟
 الف) دکمه Export نوار ابزار
 ب) Publish File
 ج) Xtras / Publish Setting
 د) File / Publish
- ۲- کدامیک از خروجی‌های زیر را نمی‌توان در دایرکتور تهیه کرد؟
 الف) Exe
 ب) Html
 ج) avi
 د) Swf
- ۳- کدام گزینه امکان اجرای نمایش را هنگام به حداقل رسیدن پنجره آن فراهم می‌آورد؟
 الف) غیر فعال کردن Animated in Background
 ب) فعال کردن Animated in Background
 ج) فعال کردن Exit Lock
 د) فعال کردن Full Screen
- ۴- کدام گزینه باعث باز شدن پنجره پخش پروژکتور در مرکز صفحه نمایش می‌گردد؟
 الف) Exit Lock
 ب) Center Stage In Monitor
 ج) Full Screen
 د) Lock stage size to movie's stage
- ۵- پسوند فایل‌های محافظت شده کدام است؟
 الف) DXR
 ب) CXT
 ج) XDR
 د) DIR

۶- فرمان ساخت فایل محافظت شده کدام ست؟

الف) File→Update Movie

ب) Xtras→Protect Movie

ج) Modify→Movie→Update

د) xtras→Update Movie

7- File > Publish Command ,In default Create File.

a) Projector (Exe)

b) ShockWave(DCR)

c) Xtra(x32)

d) Video (Avi)

کارگاه چندرسانه‌ای

۱- یکی از نمایش‌هایی که در فصل‌های قبل ایجاد کرده اید را به حالت اجرایی در آورده، آن را اجرا کنید.

۲- یکی از نمایش‌های خود را به صورت محافظت شده در آورده، سعی کنید آن را ویرایش کنید.

۳- به کمک فرمان Export یکی از انیمیشن‌های خود را به صورت فایل ویدئویی صادر و آن را تست کنید.



فصل دهم

افزودن صدا و انواع رسانه

هدف‌های رفتاری

- در پایان این فصل از هنرجو انتظار می‌رود:
- صدا را وارد کرده، خصوصیات آن را تنظیم کند.
 - صدا را وارد Score کرده، کنترل کند.
 - اصول استفاده از کانالهای صدا را بداند و آنها را کنترل کند.
 - قابلیت پخش و کنترل فایل‌های صوتی خارجی را بتواند در دایرکتور انجام دهد.
 - اصول همزمان‌سازی صوت و فیلم را توضیح دهد.
 - اصول فشرده‌سازی صداهای داخلی را به همراه تنظیمات آن شرح دهد.
 - روش استفاده از یک فیلم دایرکتور را درون فیلم جدید بتواند عملاً انجام دهد.
 - قابلیت استفاده از محتویات Flash داشته، بتواند آن را در دایرکتور کنترل کند.
 - اصول استفاده از کنترل‌های Activex را توضیح دهد.
 - خصوصیات کنترل‌های Activex را تنظیم کرده، بتواند آنها را به صحنه اضافه کند.

مقدمه:

اصولا صدا را یکی از رسانه‌های بسیار کاربردی و تأثیرگذار در تهیه و تولید چند نرم‌افزار یا سیستم‌های چند رسانه‌ای می‌دانند، که می‌تواند نقش بسیار مهمی را در ارائه جذاب محتوای الکترونیکی ایفا کند. بدین لحاظ ما در این مبحث سعی خواهیم کرد شما را با اصول استفاده و کنترل صدا در دایرکتور آشنا کنیم، علاوه بر صدا در ادامه نیز به نحوه استفاده و کنترل فیلمهای Flash و روش استفاده از کنترل‌های Activex خواهیم پرداخت.

۱-۱-۱- وارد کردن صدا به درون Cast:

به کمک فرمان Import می‌توان صدا را نیز مانند سایر رسانه‌ها به دو شکل Standard و Link وارد Cast کرده، مورد استفاده قرار داد. در حالت Standard، کل محتویات فایل صوتی وارد Cast شده، به حجم نمایش افزوده می‌گردد، اما در حالت Link تنها یک اتصال بین نمایش و فایل صوتی برقرار شده، به این ترتیب دایرکتور می‌تواند به محتویات فایل موجود در دیسک دسترسی داشته باشد، مزیت این روش Import، آن است که حجم پروژه اصلی زیاد نشده، سریعتر قابل اجرا خواهد بود.

نکته: در یک پروژه واقعی بهتر است در مسیر پروژه، یک پوشه ویژه فایل‌های صوتی ساخته و سپس از این مسیر، آنها را به شکل Link Import نمایید.

فرمت فایل‌های صوتی که با دایرکتور ۱۱ پشتیبانی می‌شوند، عبارتند از:

AIFF, WAV, MP3, Shockwave Audio, Sun AU

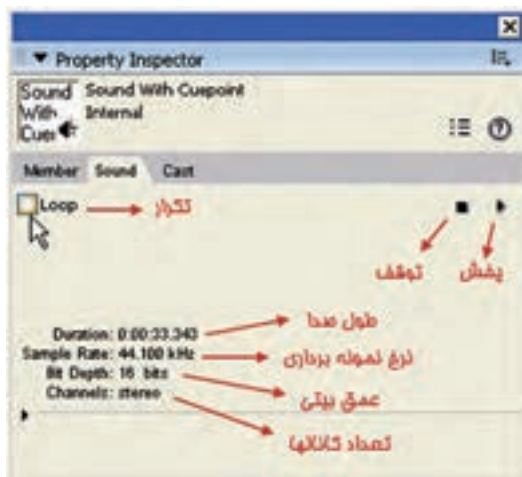
شکل ۱-۱ دو فایل صوتی وارد شده را نشان می‌دهد. اولین سلول پنجره Cast یک فایل صوتی را نشان می‌دهد که به صورت Standard وارد شده و آیکن آن یک بلندگو می‌باشد، اما در دومین سلول، یک فایل صوتی که به حالت Link وارد شده است، قابل مشاهده است که آیکن کمی متفاوت بوده و آیکن بلندگوی آن بر روی یک برگه که پوشه آن تا خورده، نمایش داده شده، ضمن این که سه تصویر موجود نیز به صورت Link وارد شده‌اند.



شکل ۱-۱-۱ دو روش وارد کردن صدا

۱۰-۲- مشاهده و تنظیم خصوصیات صدا:

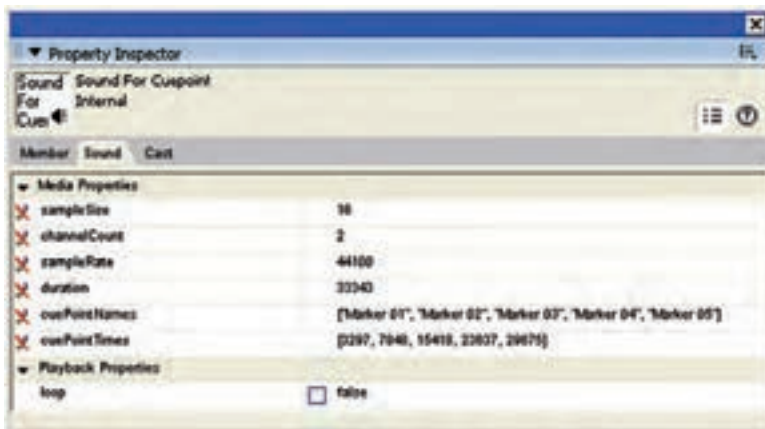
جهت نمایش و تنظیم خصوصیات صدای وارد شده، کافی است پس از انتخاب آن در Cast وارد زبانه Sound از پنجره Property Inspector شوید تا به ویژگی‌های صوت انتخاب شده، دسترسی پیدا کنید. شکل ۱۰-۲ زبانه Sound را در حالت عادی نمایش می‌دهد:



شکل ۱۰-۲ گزینه‌های زبانه Sound در حالت عادی

اگر حالت نمایش را با دکمه List View Mode عوض کنید، می‌توانید به اطلاعات بیشتری مانند نام cuePointها و زمان آنها نیز دسترسی پیدا کنید. (شکل ۱۰-۳)

نکته: در این حالت نمایشی طول فایل بر حسب میلی ثانیه داده شده است.



شکل ۱۰-۳ زبانه Sound در حالت List View

۳-۱۰- وارد کردن صدا به Score:

راحت ترین روش استفاده از اصوات بعد از وارد کردن آنها (Import) به درون Cast، قرار دادن آنها در کانالهای صوتی موجود در کانالهای افکت پنجره Score می باشد. همانطور که می دانید دو کانال صدا در بخش کانالهای افکت پنجره score وجود دارد که با قرار دادن همزمان دو صدا در آنها، این دو صدا با هم مخلوط شده و همزمان پخش می شوند (Mixer). به عنوان مثال می توان یک موزیک زمينه و یک صدای اصلی مانند دیالوگ را در این دو کانال قرار داد تا همزمان پخش شوند. اما دایرکتور محدود به همین دو کانال نبوده، قادر است ۸ کانال صدا را پشتیبانی کند که برای این کار باید مقداری کد Lingo بنویسید در مثالهای بعدی نحوه استفاده از چند کانال صوتی را به طور همزمان فرا خواهید گرفت. مثال ۱: یک Slide show طراحی کنید، به طوری که همراه آن یک صدای زمينه نیز پخش شود.

در این مثال می خواهیم چند عکس را در یک حلقه تکرار عکس قرار داده، از یک صدای Loop شده برای آنها استفاده کنیم.

مراحل انجام کار:

۱- تعداد ۵ عکس را به اندازه دلخواه در یک نمایش جدید Import کنید. (مثلاً به اندازه ۸۰۰ در ۶۰۰)

۲- اندازه stage را به اندازه عکسها تنظیم کنید.

۳- عکسهای موجود در Cast را انتخاب کرده، فرمان Modify / Cast to Time را اجرا کنید تا یک اسپریت ۵ فریمی ساخته شود.

۴- کانالهای Effects را باز کرده، در فریم اول کانال Tempo دو بار کلیک کرده، سرعت را برابر 1fps تنظیم کنید. به این ترتیب زمان پخش هر عکس ۱ ثانیه خواهد بود.

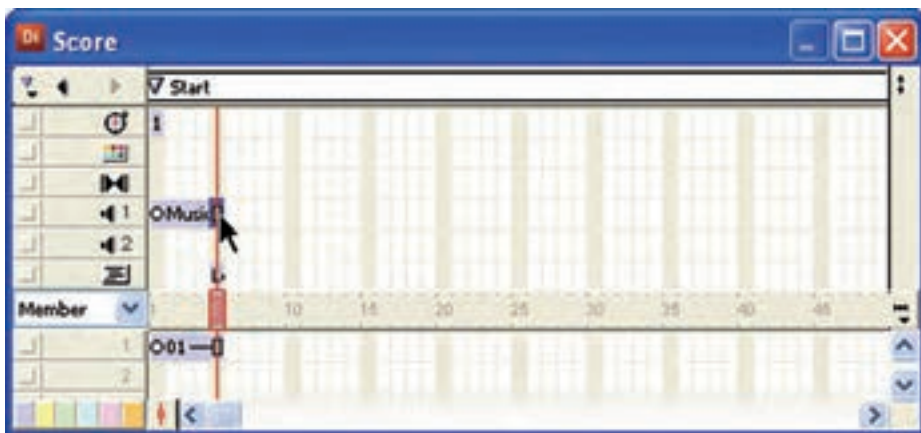
۵- اکنون می خواهیم یک حلقه تکرار برای پخش عکسها ایجاد کنیم. به طوریکه دائم در حال تکرار باشند. بنابراین یک مارکر به نام start در فریم اول قرار داده، رفتار زیر را در فریم ۵ کانال رفتار تعریف کنید و نام آن را Loop show بگذارید.
رفتار Loop show:

```
on exitFrame me  
go "Start"  
end
```

۷- اکنون نوبت وارد کردن صداست، دو صدای دلخواه را به درون Cast وارد کنید.

۸- به کمک پنجره Property Inspector می توانید صدا را تست کرده، گزینه Loop آنرا برای تکرار در نمایش فعال کنید.

۹- کانالهای افکت را باز کرده، صدا را روی یکی از کانالهای صدا قرار دهید و طول آنرا برابر طول کل نمایش تنظیم کنید. شکل ۴-۱۰



شکل ۴-۱۰ وارد کردن صدا به Score

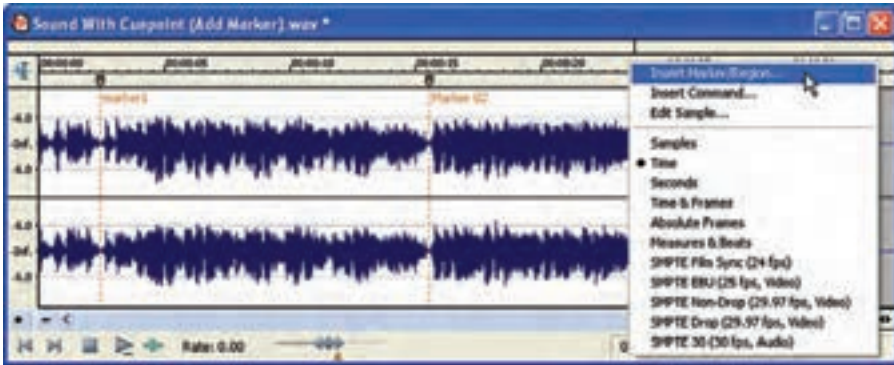
نمایش خود را ذخیره و اجرا کنید. اکنون صدای دوم را در کانال دوم صدا قرار دهید و نمایش را تست کنید. در این حالت هر دو صدا با هم پخش شده و مخلوط (Mix) می‌شوند.

۴-۱۰- همزمان‌سازی صدا و فیلم دایرکتور در Score:

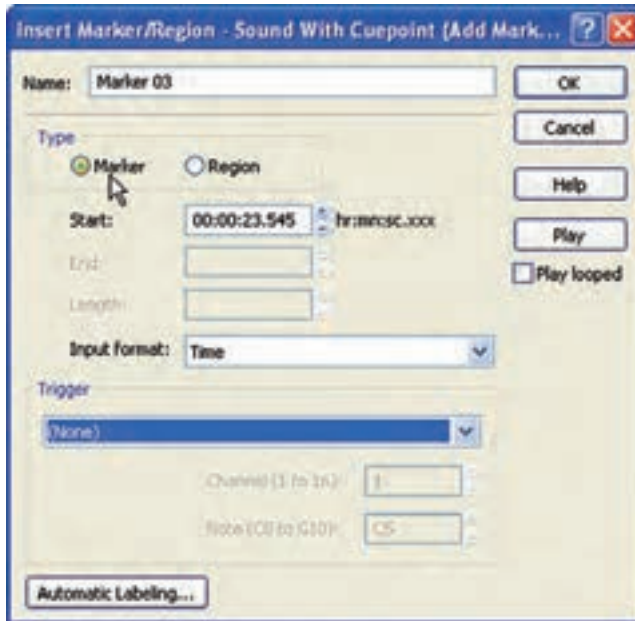
همزمان‌سازی صدا و فیلم در دایرکتور به راحتی امکان پذیر است زیرا در این نرم‌افزار قابلیت ویژه تشخیص Cue Point های موجود در فایل‌های صوتی Wav تعبیه شده است. به زبان ساده یک CuePoint (نقطه راهنما) عبارت است از یک برچسب که زمان خاصی از فایل صوتی را علامت گذاری کرده است. با نگهداشتن هد دایرکتور در مدت زمانی که هد صدا به CuePoint مشخصی می‌رسد می‌توان عمل همزمانی را به راحتی انجام داد. برای درک بهتر مطلب مثال ۲ را اجرا کنید:

مثال ۲:

- ۱- در این مثال قصد داریم متن اذان را به همراه صدای آن به طور هماهنگ نمایش دهیم. یک فایل صوتی اذان را آماده کنید.
- ۲- این فایل را درون نرم‌افزار Sound Forge باز کنید. (می‌توانید از نرم‌افزار قدرتمند Audition نیز کمک بگیرید)
- ۳- در این مرحله باید در انتهای گفتار هر بخش از اذان یک Cuepoint درج کنیم. بنابراین فایل را اجرا کرده، هد را در پایان "الله اکبر" متوقف کنید. سپس روی بخش بالایی پنجره ویرایش که خط‌کش زمان در آنجاست، کلیک راست کرده، فرمان Insert Marker/Region را اجرا کنید (شکل ۵-۱۰) تا کادر محاوره‌ای آن مطابق شکل ۶-۱۰ ظاهر شود.



شکل ۵-۱۰



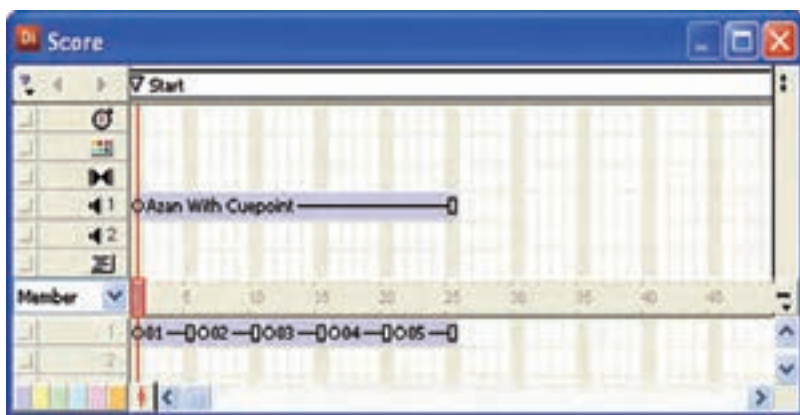
شکل ۶-۱۰ علامت‌گذاری توسط کادر محاوره‌ای Insert Marker

۴- در کادر محاوره‌ای Insert Marker/Region (شکل ۶-۱۰) گزینه Marker را از بخش Type انتخاب کرده، در صورت نیاز نام دلخواهی را در بخش Name وارد کنید و در پایان کادر محاوره‌ای را تایید کنید.

۵- در انتهای هر یک از بخشهای اذان نیز یک مارکر اضافه کنید.

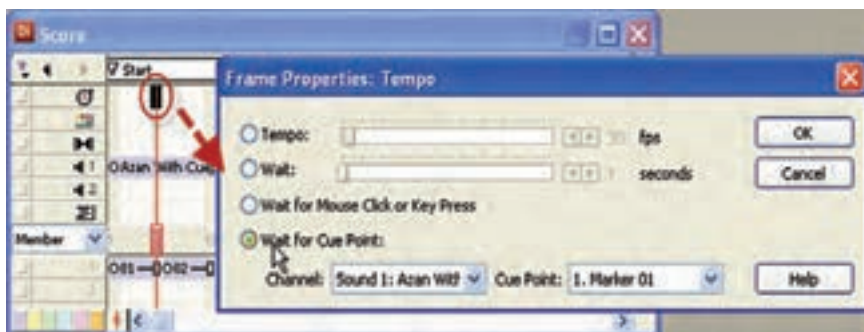
۶- اکنون فایل را به فرمت Wav ذخیره کرده، آنرا به‌داخل دایرکتور Import کنید.

- ۷- به کمک نرم‌افزار Photoshop برای هر یک از بخشهای اذان یک فایل گرافیکی، شامل متن اذان طراحی کنید و پس از ذخیره، آنها را با همان فرمت اصلی Psd وارد دایرکتور کنید. (راهنمایی: در صورت نیاز می‌توانید تصاویری با زمینه شفاف ایجاد کنید)
- ۸- تصاویر مربوط به متن اذان را به ترتیب وارد Score کرده، برای هر کدام یک اسپریت ۵ فریمی ایجاد کرده، پشت سرهم در کانال شماره یک قرار دهید. (شکل ۷-۱۰)
- ۹- صدای آماده شده را نیز وارد کانال صدای شماره ۱ پنجره Score کنید. (شکل ۷-۱۰)



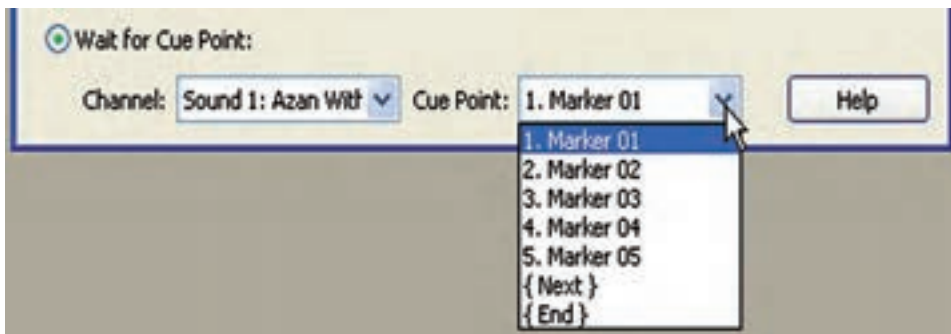
شکل ۷-۱۰ وارد کردن متن‌ها و صدای اذان بدرون Score

- ۱۰- اکنون به تنظیم مهمترین بخش نمایش می‌پردازیم که عبارت است از نگهداشتن هد در آخرین فریم مربوط به متن هر بخش از اذان (فریمهای ۵ - ۱۰ - ۱۵ - ۲۰ - ۲۵ و...). برای این منظور در فرم ۵ از کانال Tempo دوبار کلیک کنید تا کادر محاوره‌ای Frame Properties: Tempo ظاهر شود، سپس گزینه Wait for cue point را انتخاب کنید تا گزینه‌های آن مطابق شکل ۸-۱۰ فعال شوند.



شکل ۸-۱۰

همانطور که ملاحظه می‌کنید دایرکتور به‌طور خودکار صدای موجود در کانال یک را در لیست Channel تشخیص داده، کلیه Cue point های موجود در فایل صوتی را در لیست Cue point قرار می‌دهد، که به‌طور خودکار اولین علامت موجود در فایل صوتی به‌نام Marker 01 را در لیست انتخاب کرده است و شما قادرید هر یک از علامتهای راهنمای موجود در لیست را انتخاب کنید. (شکل ۹-۱۰)



شکل ۹-۱۰

نکته:

همانطور که در شکل ۹-۱۰ ملاحظه می‌شود، دو گزینه {Next} و {End} نیز، به‌صورت پیش فرض در لیست Cue point ها قرار دارند. (حتی در صورت درج نشدن علامت راهنما در فایل صوتی (این دو گزینه همواره در دسترس هستند از این دو گزینه در موارد زیر استفاده می‌شود:

{Next}: این گزینه به مفهوم Cue Point بعدی است، بنابراین، به‌صورت نسبی عمل کرده تا رسیدن صدا به علامت بعدی، هد دایرکتور را متوقف نگه می‌دارد.
{End}: در صورت انتخاب این گزینه هد دایرکتور تا پخش کامل فایل صوتی (رسیدن به انتها) متوقف می‌ماند و بهترین گزینه برای پخش کامل یک صوت است و نیازی به درج علامت Cue point در فایل صوتی نیست.

- ۱۱- با تایید گزینه Marker 01 یا انتخاب گزینه {Next} کادر محاوره‌ای را تایید کنید.
- ۱۲- هد را در فریم اول قرار داده، نمایش را تست کنید و به حرکت هد توجه کنید. هد دایرکتور تا رسیدن صدا به اولین علامت، متوقف مانده، بنابراین کاربر قادر به مشاهده متن اولین بخش اذان خواهد بود.
- ۱۳- مراحل ۱۰ تا ۱۲ را برای سایر بخشهای فایل صوتی اذان کامل کرده، نمایش را ذخیره و تست کنید.

۱۰-۵- نحوه استفاده و کنترل کانالهای صوتی:

همانطور که پیش از این نیز ذکر شد، دایرکتور امکان استفاده از ۸ کانال صوتی را به‌طور هم‌زمان در اختیار طراحان و اسکریپت‌نویس‌ها قرار داده است که ساده‌ترین روش استفاده از این کانالها قرار دادن اصوات در کانالهای صوتی موجود در Score است، هر چند الزامی برای این کار وجود نداشته و می‌توان بدون وارد کردن صدا به درون Score به‌وسیله دستورات لینگو، صداهای موجود در Cast را پخش و کنترل کرد. ضمن این که دایرکتور امکان پخش و کنترل فایل‌های صوتی خارجی را داراست. بنابراین نیازی به وارد کردن آنها به درون Cast نمی‌باشد.

به‌طور کلی چه درحالتی که صدا را وارد Cast کرده و داخل یکی از کانالهای صوتی موجود در Score قرار داده باشید و چه این کار را نکرده باشید، جهت پخش و کنترل یک صدا لازم است از کانالهای صوتی برای این منظور استفاده کنید، که این عمل به وسیله خصوصیات متدهای از پیش تعریف شده برای کانالهای صدای دایرکتور امکان پذیر است. برای استفاده از این خواص و متدها از فرم کلی زیر استفاده کنید:

`Sound(Channel Number).Property = Value`
`Sound(Channel Number).Method()`

در فرم کلی فوق منظور از Channel Number، شماره یکی از کانالهای صداست که می‌تواند مقداری بین ۱ تا ۸ داشته باشد و منظور از Property یکی از خواص کانالهای صوتی است که می‌توان مقدار Value را به آن نسبت داده یا مقدار آن را بررسی کرد. مانند خاصیت Volume که حجم صدا را تعیین می‌کند و بالاخره منظور از Method()، یکی از متدهای کنترل صدا مانند Pause() می‌باشد که باعث توقف موقت پخش صدا می‌گردد.

۱۰-۵-۱- متدهای کنترل کانالهای صوتی

جدول ۱۰-۱۱ لیست مهمترین متدهای مربوط به کنترل یک کانال صوتی است:

متد	شرح	مثال
PlayFile (stringFilePath)	پخش فایل خارجی	Sound(1).PlayFile("d:\1.Mp3")
Play(Member (Index OR Name))	پخش یک Member صوتی	S o u n d (1) . Play(Member("Music"))
Play()	ادامه پخش صدای Pause شده	Sound(2).Play()
Stop()	توقف کامل صدا	Sound(3).Stop()

متد	شرح	مثال
Pause ()	توقف موقت صدا	Sound(4).Pause ()
Rewind()	برگرداندن هد صدا به ابتدای آن	Sound(5).Rewind ()
FadeIn({ Milliseconds })	به سرعت ولوم صدا را صفر کرده و آن را در مدت زمان تعیین شده که واحد آن میلی ثانیه است به مقدار فعلی برمی گرداند. پارامتر زمان اختیاری بوده، چنانچه آن را حذف کنید یک ثانیه خواهد بود.	Sound(6).FadeIn ()
FadeOut ({ Milliseconds })	ولوم صدا را در زمان تعیین شده که پارامتری اختیاری است به صفر می رساند. چنانچه این پارامتر را حذف کنید یک ثانیه خواهد بود.	Sound(7).FadeOut(3000)
FadeTo (Volume , { Milliseconds })	ولوم صدا را در مدت زمان تعیین شده به حد دلخواه می رساند.	Sound(8).FadeTo(50,2000)
isBusy()	تعیین می کند کانال صوتی در حال پخش است یا نه؛ خروجی این متد یک مقدار منطقی است.	Put Sound(2).isBusy()

جدول ۱۱-۱۰ متدهای مربوط به کانالهای صدا

۱-۵-۱۰- خواص کانالهای صوتی

جدول ۱۲-۱۰ لیست مهمترین خواص مربوط به کانالهای صوتی را نشان می دهد:

خصوصیت	شرح	مثال
Volume	تست یا تنظیم حجم کانال صدا که مقداری بین ۰ تا ۲۵۵ می باشد.	Sound(1).Volume=255 Put Sound(2).Volume
Pan	تست یا تنظیم بالانس کانال صدا که مقداری بین -۱۰۰ تا ۱۰۰ می باشد. در سیستم استریو مقدار -۱۰۰ باعث پخش کامل صدا از باند سمت چپ و مقدار ۱۰۰ باعث پخش کامل صدا از باند سمت راست می گردد. برای تراز کردن بالانس از مقدار صفر استفاده کنید.	Sound(2).Pan=-100 Put Sound(2).Pan
Member	تعیین عضو صوتی که در کانال صدا در حال استفاده است.	Put Sound(2).Member

جدول ۱۲-۱۰ خواص کانالهای صدا

نکته: خاصیتی به نام `soundLevel` وجود دارد که جزء خواص سطح بالای صداست (`Top Level`) و ولوم صدای خروجی نهایی دایرکتور را تنظیم می‌کند. به عبارت دیگر به کمک آن می‌توان ولوم صدای ۸ کانال را به‌طور هم‌زمان کنترل کرد. این خاصیت می‌تواند مقداری بین ۰ تا ۷ را دارا باشد و به فرم کلی زیر قابل استفاده است:

`_Sound.SoundLevel=Value`
 به‌عنوان مثال فرمان `_Sound.SoundLevel=0` موجب قطع کامل صدای خروجی نهایی و فرمان `_Sound.SoundLevel=7` باعث افزایش کامل صدای خروجی نهایی دایرکتور می‌گردد.
 مثال ۲:

هدف این مثال آشنایی با متدها و خواص کانالهای صوتی می‌باشد. برای این منظور یک پخش‌کننده صدا (`Soundplayer`) با امکان پخش سه موزیک متفاوت، به‌طوری که بتوان پخش صدا را کنترل کرده و امکان تغییر بلندی صدا نیز وجود داشته باشد، ایجاد کنید. توجه داشته باشید با زدن هر یک از دکمه‌های `Play`، صدای مربوطه از کانال صوتی شماره یک (۱) پخش می‌شود.
 مراحل انجام کار:

- ۱- سه فایل صوتی دلخواه را به درون یک نمایش جدید وارد کنید و آنها را به ترتیب "track 1"، "track 2" و "track 3" نامگذاری کنید.
- ۲- تعداد هشت دکمه مطابق شکل ۱۴-۱۰ بر روی صحنه قرار دهید. (راهنمایی: این دکمه‌ها در کانالهای ۱ تا ۸ و از فریم ۱ تا ۵ را اشغال می‌کنند)



شکل ۱۴-۱۰ طراحی صحنه پروژه `Sound Player`

۳- رفتار آماده Hold on current frame را در فریم ۵ از کانال رفتار score قرار دهید.
 ۴- حال نوبت نوشتن رفتارهای دکمه مربوط به پخش می‌باشد. برای نوشتن این رفتارها از دستور کلی ("نام عضو صوتی" member "play (member (شماره کانال) sound استفاده می‌کنیم. در این شکل کلی شماره کانال می‌تواند عددی بین ۱ تا ۸ باشد که معرف شماره کانال صوتی است که می‌خواهد عضو صوتی موجود در Cast در آن پخش شود. بنابراین رفتارهای زیر را برای دکمه پخش ایجاد کنید:

• رفتار play 1 مربوط به دکمه play Track 1:

```
on MouseUp me
  sound (1). play ( member "track 1")
end
```

• رفتار play 2 مربوط به دکمه play Track 2:

```
on MouseUp me
  sound (1). play ( member "track 2")
end
```

• رفتار play 3 مربوط به دکمه play Track 3:

```
on MouseUp me
  sound (1). play ( member "track 3")
end
```

۵- اکنون نوبت به نوشتن رفتار برای دکمه‌های باقی مانده است که بسیار ساده می‌باشند:
 • رفتار stop مربوط به دکمه stop:

```
on MouseUp me
  sound (1). stop ()
end
```

• رفتار Pause مربوط به دکمه Pause:

```
on MouseUp me
  sound (1). Pause ()
end
```

• رفتار Resume مربوط به دکمه Resume:

```
on MouseUp me
  sound (1). Play ()
end
```

نکته: رفتار Resume باعث ادامه دادن به پخش صدای Pause شده می‌گردد، که از همان مدت play ولی بدون پارامتر اضافی استفاده می‌کند.

- رفتار VolUp مربوط به دکمه افزایش صدا (+)

```
on MouseUp me
  sound (1). Volume=Sound(1).Volume+25
end
```

- رفتار VolDn مربوط به دکمه کاهش صدا (-)

```
on MouseUp me
  sound (1). Volume=Sound(1).Volume-25
end
```

۱۰-۶- کنترل صدا در Score:

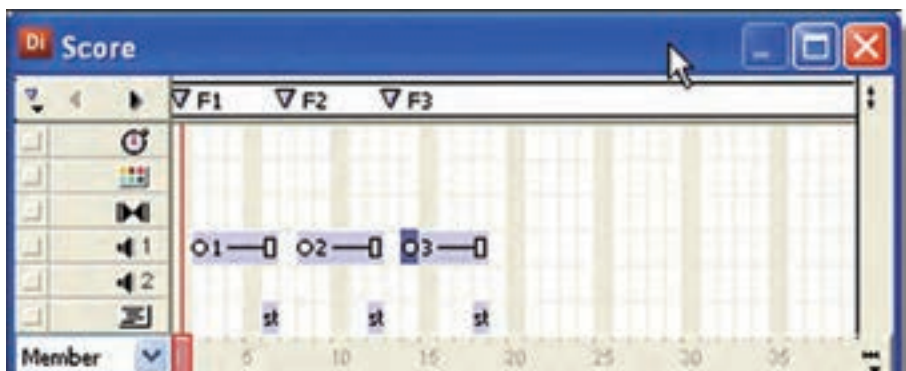
همانطور که در میحث قبلی ذکر شد تمام متدها و خصوصیات مربوط به کانالهای صوتی مانند متد Pause() یا خاصیت Volume از طریق قرار دادن اصوات در کانالهای صدای موجود در Score نیز قابل دسترس هستند، اما استفاده از Score در بسیاری از موارد باعث ساده تر شدن طراحی نمایش، به خصوص در بخش صدا می‌گردد. اکنون مثال زیر را اجرا کنید تا تکنیک‌های استفاده از صدا و کنترل آن در Score را فرا گیرید.

مثال ۳:

مراحل کار:

یک نمایش جدید ایجاد و سه فایل صوتی را وارد کنید.

فایلها را مطابق شکل ۱۵-۱۰ در کانال صدای شماره یک بچینید، سپس در فریم انتهای هر صدا یک فرمان توقف هد درج کنید.



شکل ۱۵-۱۰

برای هر یک از اصوات یک مارکر درج کنید. هر مارکر مطابق شکل ۱۵-۱۰ یک فریم قبل از شروع اسپرایت صدا درج شده است.

هد را به فریم اول انتقال داده، نمایش را پخش کنید. دایرکتور پخش فایل شماره یک را آغاز می کند.
 اکنون پنجره Message را باز کرده، فرامین زیر را مطابق جدول ۱۶-۱۰ در آن اجرا کنید تا به عملکرد آنها پی ببرید:

فرمان	عملکرد
Go "F2"	هد به اولین فریم دوم رسیده بنابراین پخش آن آغاز می شود.
Go Next	پخش فایل سوم
Sound(1).Pause()	توقف موقت فایل سوم
Sound(1).Play()	از سر گیری پخش فایل سوم
Sound(1).Rewind()	صدا دوباره از ابتدای فایل سوم شروع به پخش می کند.
Sound(1).Stop()	صدا به صورت کامل متوقف می شود.
Go Loop	هر صدا به مارکر جاری (F3) برگشته و پخش فایل سوم آغاز می شود.
Sound(1).Volume=0	فایل سوم در حال پخش است، اما هیچ صدایی شنیده نمی شود.
Sound(1).Volume=255	به حداکثر رساندن صدای در حال پخش در کانال شماره یک صدا
Put Sound(1).Status	نمایش وضعیت کانال شماره یک صدا (مقدار شماره ۳ به مفهوم «در حال پخش» را نمایش می یابد)

جدول ۱۶-۱۰

۷-۱۰- فشرده سازی صداهای داخلی به فرمت SWA:

دایرکتور از تکنولوژی جالبی به نام Shockwave Audio جهت فشرده سازی صداهای داخلی استفاده می کند که باعث کاهش چشمگیر حجم کلیه صداهای درون Cast می شود. بنابراین حجم فایل نهایی که به کمک عمل Publish تهیه می گردد، کاهش می یابد. بر این اساس کاربر

قادر است در هر لحظه از طراحی نمایش، تنظیمات فشرده‌سازی را به میل خود تغییر دهد، زیرا این تنظیمات زمانی اتفاق می‌افتد که بخواهد نمایش را با Publish به یکی از فرمت‌های Projector - Shockwave Movie یا از طریق فرمان Update Movie تبدیل کند.

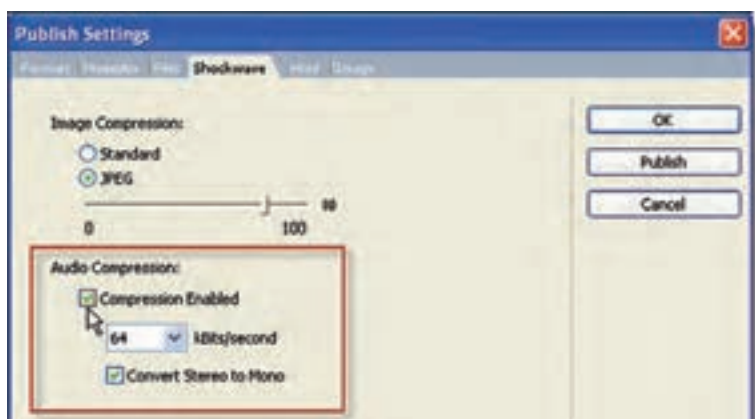
نکته:

فشرده‌سازی Shockwave Audio، فقط قادر به فشرده‌سازی صداهای WAV بوده، روی MP3 اثری ندارد؛ زیرا فایل‌های MP3 فشرده شده هستند و به‌وسیله نرم‌افزارهای ویرایش صدا مانند Sound Forge یا Audition می‌توانید حجم آنها را کم کنید.

مراحل انجام کار:

۱- وارد تنظیمات Publish شوید: File/Publish Setting

۲- زبانه ShockWave را انتخاب کنید. (شکل ۱۷-۱۰)



شکل ۱۷-۱۰

۳- گزینه Compression Enabled را علامت بزینید تا فعال شود.

۴- نرخ انتقال داده صوتی را برحسب کیلوبیت بر ثانیه تنظیم کنید.

۵- جهت تبدیل صداهای استریو به مونو، گزینه Convert Stereo to Mono را علامت‌دار کنید.

۶- کادر محاوره‌ای را تایید کنید.

۸-۱۰- استفاده از یک فیلم دایرکتور درون فیلم جدید

شما قادر هستید از یک فیلم اصلی دایرکتور (پسوند dir) در یک فیلم دیگر دایرکتور استفاده کنید. برای این منظور کافی است فایل مورد نظر خود را Import کنید که در این حالت نمایش وارد شده، به‌صورت یک Film Loop به‌همراه تمامی عناصر وارد Cast می‌شود

و با قرار دادن Film Loop به درون صحنه می‌توانید از آن استفاده کنید. برای تنظیم خصوصیات این حلقه فیلم، کافی است آن را انتخاب و از زبانه Film Loop اقدام کنید. در مورد این تنظیمات قبلاً در فصل ششم به‌طور کامل صحبت کردیم.

۹-۱۰- استفاده از محتویات Flash

در دایرکتور به راحتی می‌توان از فیلم‌های Swf استفاده کرد و آنها را تحت کنترل در آورد. به‌عنوان مثال: عملیات توقف (stop)، پخش مجدد (Play) و برگشت به ابتدا (Rewind) رفتن به یک فریم یا برچسب خاص و بسیاری از عملیات مشابه دیگر نیز در دایرکتور قابل دسترسی می‌باشند.

۱۰-۱۰-۱- وارد کردن فیلم swf به درون نمایش:

برای وارد کردن فیلم‌های فلش به درون نمایش خود به یکی از دو روش زیر عمل کنید: روش اول: استفاده از کادر محاوره‌ای Import روش دوم: از طریق منو Insert / Media Element / Flash Movie ... را انتخاب کنید تا کادر محاوره‌ای آن مطابق شکل ۱۸-۱۰ ظاهر شود:



شکل ۱۸-۱۰ کادر محاوره‌ای انتخاب فیلم swf

به کمک دکمه Browse فایل swf خود را پیدا کرده، آن را باز کنید. سپس تنظیمات زیر در کادر محاوره‌ای Flash Asset Properties انجام دهید: Media (رسانه): این گزینه به‌طور پیش فرض فعال است و باعث برقراری یک اتصال (Link) بین نمایش دایرکتور و فایل swf می‌گردد، اما فایل را به درون نمایش وارد نمی‌کند.

برای اینکه فایل swf به نمایش شما اضافه گردد، این گزینه را غیر فعال کنید. Preload: این گزینه زمانی در دسترس است که گزینه Media تیک خورده باشد و باعث می‌شود که دایرکتور قبل از شروع به پخش فایل swf، آن را به‌طور کامل به حافظه بارگذاری کند، در صورتی که آن را غیر فعال کنید همزمان با بارگذاری پخش خواهد شود. Image: به‌طور پیش فرض فعال بوده، باعث نمایش تصویر فایل swf می‌گردد. در صورتی که آن را غیر فعال کنید، فقط صدای swf را خواهید شنید. Paused: اگر این گزینه را فعال کنید، فایل swf در ابتدای شروع به پخش در حالت توقف قرار گرفته، فقط فریم اول آن پخش خواهد شد. sound: با غیر فعال کردن این گزینه صدای فیلم swf قطع شده، فقط تصویر آن پخش می‌شود.

Loop: جهت تکرار نمایش در صفحه، پس از پایان یافتن آن از این گزینه استفاده می‌شود..

Direct to stage (DTS): اگر این گزینه را فعال کنید، فیلم فلش شما با نادیده گرفتن و پوشاندن سایر اسپرایت‌ها، به‌طور مستقیم در صفحه به نمایش در می‌آید که مزیت آن نمایش بهتر و ملایمتر آن خواهد بود.

Quality: از این گزینه برای تنظیم کیفیت پخش فیلم استفاده می‌شود. Scale Mode: با استفاده از این گزینه می‌توان مقیاس فیلم در swf را روی صفحه تعیین کرد.

Rate: مربوط به سرعت پخش فیلم می‌باشد که دارای گزینه‌های زیر است: Normal: دایرکتور فایل فلش را با سرعت اصلی خودش پخش کند. Fixed: از این گزینه برای پخش با سرعت ثابت و تعیین شده استفاده می‌شود که برای این منظور شما می‌توانید سرعت مورد نظر خود را در کادر مقابل این گزینه وارد کنید. (مثلاً می‌توانید با سرعت ثابت ۵fps، آن را پخش کنید) Lock step: سرعت پخش فیلم swf با سرعت Tempo در نمایش شما تعیین می‌گردد. بنابراین با این روش می‌توان سرعت‌های متفاوت و متغیری را در طول پخش ایجاد کرد.

نکته: سرعت پخش فیلم فلش نمی‌تواند بیشتر از سرعت Tempo در دایرکتور باشد. بنابراین اگر شما گزینه Normal را انتخاب کرده باشید و به‌عنوان مثال frame Rate فیلم swf شما ۱۵ باشد، برای پخش فیلم Swf با سرعت واقعی، باید Tempo حداقل ۱۵ باشد. در غیر این صورت فیلم swf با سرعت کمتری پخش خواهد شد. scale: از این گزینه برای تعیین مقیاس فیلم swf استفاده می‌شود، به‌عنوان مثال اگر ۵۰ را وارد کنید، اندازه فیلم swf نصف خواهد شد.

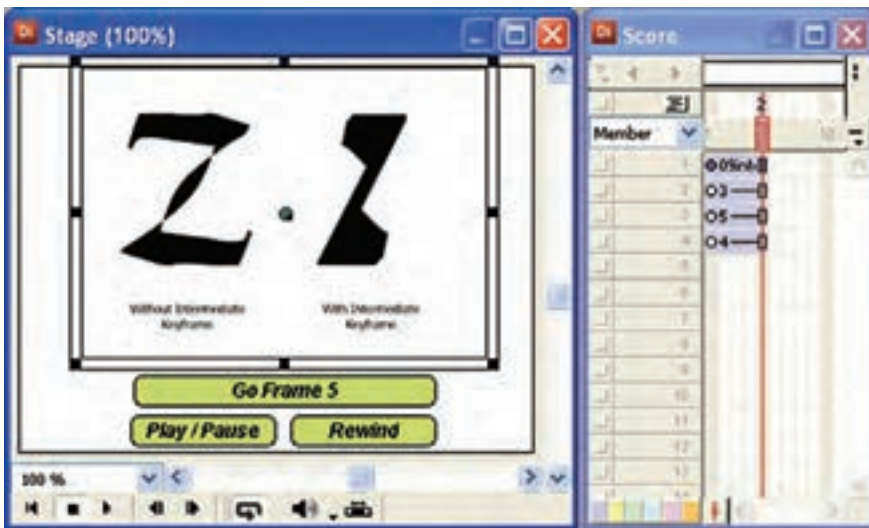
در ناحیه سمت چپ کادر محاوره‌ای فوق که ناحیه Preview می‌باشد، می‌توانید پیش‌نمایشی از فیلم swf را مشاهده نموده، مشخصات آن را شامل تعداد فریمهای آن، سرعت اصلی پخش آن (fps)، اندازه آن به پیکسل و حجم فایل آن را مشاهده کنید. بعد از انجام تنظیمات و مشاهده مشخصات فیلم swf، دکمه OK را کلیک کنید تا کادر محاوره‌ای بسته شده، فیلم swf وارد پنجره Cast شود. اکنون می‌توانید فیلم فلش را درون Score قرار دهید تا در خروجی ظاهر شود. چنانچه نمایش دایرکتور را پخش کنید، فیلم فلش نیز به‌صورت کامل پخش می‌شود.

نکته: بعد از وارد کردن فیلم فلش به درون Cast نیز می‌توان از مشخصات و تنظیمات آن مطلع شد؛ کافی است فلش را انتخاب کرده، از طریق دو زبانه Member و Flash در پنجره Property Inspector به مشخصات فوق دسترسی پیدا کرد.

۱۰-۱۱- کنترل فیلم Flash با Lingo:

برای اینکه بیشتر با کدهای کنترل فیلم Flash در دایرکتور آشنا شوید، به مثال زیر توجه کنید: مراحل انجام کار:

- ۱- یک فلش از نوع swf را به داخل یک فیلم جدید دایرکتور وارد کنید (بهتر است swf شما دارای انیمیشن باشد تا مفهوم کنترل را بهتر درک کنید)
- ۲- فیلم فلش را وارد صحنه کرده، دکمه‌های زیر را در صحنه قرار دهید. توجه کنید که فلش در کانال شماره یک و بقیه دکمه‌ها در سایر کانال‌های زیر آن قرار گیرد. (شکل ۱۹-۱۰)



شکل ۱۹-۱۰ کنترل فیلم Flash توسط Lingo

۳- رفتار توقف را در فریم آخر نمایش درج کنید.

۴- نمایش را پخش کنید، مشاهده خواهید کرد که فیلم فلش به‌طور خودکار پخش خواهد شد. البته در صورتی که گزینه Paused فعال نشده باشد.

۵- حال نوبت نوشتن رفتارها برای دکمه‌هاست، رفتارهای زیر را برای دکمه‌ها بنویسید.

- رفتار دکمه / play Pause

```
on mouseUp me
if sprite(1).playing=False then
Sprite(1).play()
else
Sprite(1).Stop()
end if
end
```

نکته: خاصیت playing تعیین می‌کند فیلم Flash در حال پخش است (True) یا خیر (False)

- رفتار دکمه Rewind

```
on mouseUp me
Sprite(1).Rewind()
end
```

- رفتار دکمه 5 gotoframe

```
on mouseup me
Sprite(1).gotoframe(5)
end
```

این رفتار باعث انتقال کنترل به فریم شماره ۵ فیلم فلش می‌گردد.

چند متد کاربردی دیگر:

متد `findlabel()`: جهت پیدا کردن شماره فریم برچسب دار.
مثال:

انتقال کنترل به برچسب "Menu"

```
n = Sprite (1). findlabel ("Menu")
Sprite (1). gotoframe (n)
```

متد `hold()` مشابه `stop()` می‌باشد، با این تفاوت که `stop` هم صوت و هم تصویر (انیمیشن) را نگه می‌دارد، اما `hold ()` فقط باعث توقف تصویر می‌شود. بنابراین پخش صدا ادامه پیدا می‌کند.

متد `PrintAsBitmap`: جهت چاپ اشیایی که دارای اطلاعات کانال Alpha باشند.

۱۲-۱۰- استفاده از کنترل‌های Activex

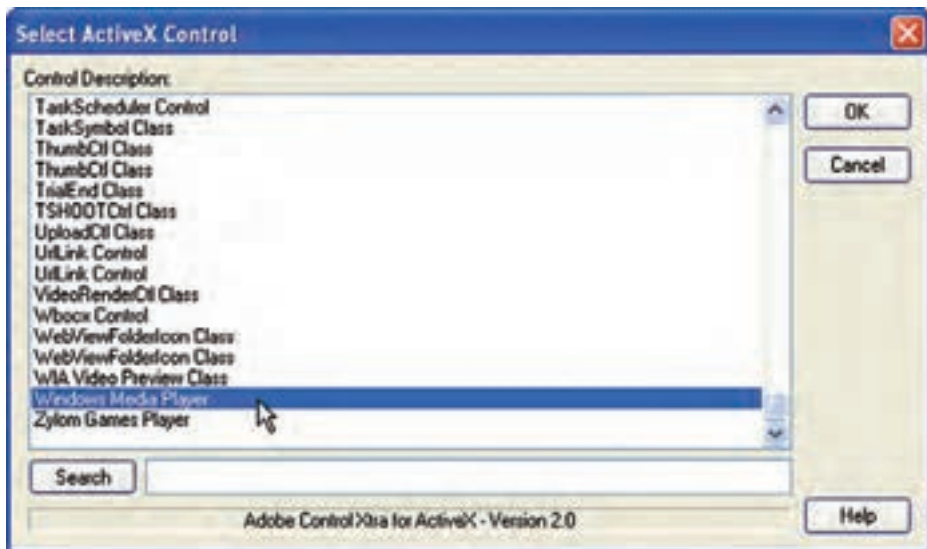
Activex ها در حقیقت اجزای بصری (Visual) تشکیل دهنده فرم‌های ویندوز هستند مانند دکمه‌ها، برچسب‌ها، لیست‌ها و... که در محیط‌های برنامه‌نویسی مانند VB از آنها جهت طراحی فرمها استفاده می‌شود. دایرکتور قابلیت وارد کردن و استفاده از Activex ها را به کاربر می‌دهد و در حقیقت شما می‌توانید همانند VB یا هر محیط برنامه‌نویسی دیگری به کمک این اجزاء برنامه‌های کاربردی بسازید.

مثال ساده زیر نحوه وارد کردن و استفاده از Activex ها را جهت خلق نمایشی که با استفاده از Activex Windows Media Player قادر به پخش انواع فایل‌های رسانه‌ای از روی دیسک می‌باشد را شرح می‌دهد.

مراحل کار:

یک نمایش جدید ایجاد کنید.

از طریق منوها Insert / Control / Activex را کلیک کنید تا کادر محاوره‌ای آن مطابق شکل ۲۰-۱۰ باز شود.

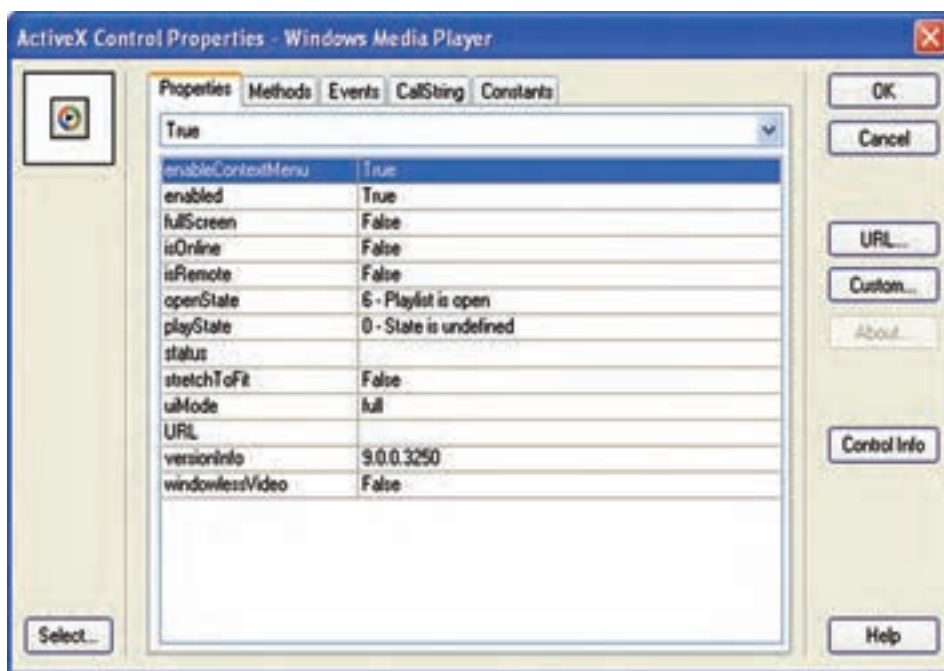


شکل ۲۰-۱۰ کادر انتخاب Activex

در کارد محاوره‌ای Activex گزینه Windows Media Player را انتخاب کرده، تایید کنید.

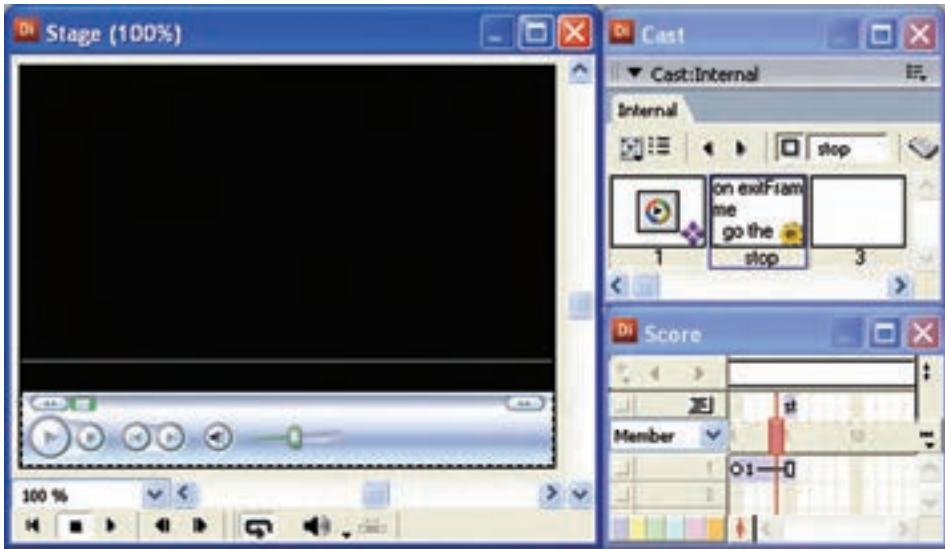
کادر محاوره‌ای جدیدی باز می‌شود که کلیه خصوصیات (Propertise)، متدها (Methods) و رویدادهای (Events) Activex انتخاب شده را نشان می‌دهد (شکل ۲۱-۱۰) و شما قادرید مقادیر اولیه را تنظیم کنید (مانند Properties در محیط VB، کافی است خاصیت مورد نظر را انتخاب و مقدار آن را در بخش بالایی کادر تنظیم کنید).

یکی از خواص مهم این Activex؛ URL می‌باشد که به وسیله آن می‌توان یک فایل رسانه‌ای را جهت پخش معرفی نمود و در این نمایش از آن استفاده خواهیم کرد. این کادر محاوره‌ای را نیز بدون تغییری در خصوصیات آن تایید کنید تا Activex وارد Cast شود.



شکل ۲۱-۱۰ تنظیم خصوصیات Activex

حال مطابق شکل ۲۲-۱۰ Activex را در کانال شماره یک Score قرار داده، اندازه آن را برابر Stage تنظیم کنید. (طول اسپرایت اهمیتی ندارد، اما برای آن ۵ فریم در نظر بگیرید و در آخرین فریم نمایش فرمان توقف هد را قرار دهید.)



شکل ۲۲-۱۰ قراردادن Activex در صحنه

اکنون نوبت به آخرین مرحله از طراحی نمایش می‌رسد و نیاز به رفتاری داریم که یک فایل صوتی یا تصویری را درون Activex باز کند. در یک پروژه واقعی لازم است فایل‌های مورد نظر را درون یک پوشه قرار داده، آن‌را به مسیر پروژه یعنی محلی که فایل اصلی دایرکتور قرار دارد کپی کنیم سپس رفتار زیر را برای یکی از فریم‌های ۱ تا ۴ یا دکمه‌ای دلخواه بنویسیم (با فرض اینکه پوشه Media حاوی فایل ویدئویی AV02.dat است و در مسیر پروژه قرار دارد، رفتار زیر را در فریم ۳ کانال اسکریپت درج کنید).

```
on exitFrame me
Sprite(1).URL=_Movie.path&"Media\AV02.dat"
End
```

خاصیت Path یکی از خصوصیات کاربردی مربوط به شیء _Movie می‌باشد که مسیر فایل دایرکتور را در خود داراست و به کمک آن می‌توانید به مسیر پروژه دسترسی پیدا کنید. برای تست بهتر این خاصیت، کافی است در یک نمایش ذخیره شده، فرمان زیر را در یک اسکریپت یا پنجره Message تایپ کنید:

```
Put _Movie.path
```

همانطور که در رفتار فوق مشاهده می‌کنید جهت استفاده از خواص و متدهای یک Activex، کافی است از همان فرم کلی خواص و متدهای مربوط به اسپرایت‌ها استفاده کنید. یعنی:

```
Sprite(Index OR "Name").Property
```


Sprite(Index OR “Name”).Method()

و بالاخره می‌توانید علاوه بر رویدادهای دایرکتور از رویدادهای اصلی Activex نیز، هنگام نوشتن اسکریپت‌ها استفاده کنید، که به فرم کلی زیر قابل استفاده هستند.

یادآوری: جهت مشاهده لیست رویدادهای یک Activex در کادر محاوره‌ای Activex Control Properties از زبانه Events استفاده کنید. همچنین جهت آشنایی با خصوصیات یک اکتیوکس از زبانه CallString کمک بگیرید.

On Event
Statements

End

به‌عنوان مثال Activex Microsoft Form2 CommandButton دارای رویدادی به‌نام DbClick می‌باشد که می‌توانید از آن در یک مدیر رویداد به فرم مثال زیر استفاده کنید:

on DbClick

put “ DoubleClick Occured”

end

نمایش را ذخیره و تست کنید.

هد را به فریم اول انتقال داده، نمایش را اجرا کنید. به محض عبور هد از فریم ۳ فایل ویدئو در Activex باز شده، شروع به پخش می‌کند. می‌توانید از کنترل‌های موجود در Activex به آسانی استفاده کنید حتی ویژگی Full Screen نیز در دسترس می‌باشد. کافی است روی ویدئو دوبار کلیک کنید.

خلاصه مطالب

به کمک فرمان Import می‌توان صدا را نیز مانند سایر رسانه‌ها به دو شکل Standard و Link وارد Cast کرده، مورد استفاده قرار داد. جهت نمایش و تنظیم خصوصیات صدای وارد شده، کافی است پس از انتخاب آن در Cast وارد زبانه Sound از پنجره Property Inspector شوید.

یک روش آسان استفاده از صدا، قرار دادن آن در کانالهای صدای موجود در Score می‌باشد.

با اضافه کردن CuePoint ها به فایل صوتی Wav می‌توان عمل هماهنگ‌سازی صدا و تصویر را به آسانی اجرا کرد، هر فایل صوتی به‌طور پیش فرض دارای یک CuePoint است که در انتهای آن قرار دارد.

دایرکتور مجهز به ۸ کانال صوتی است که به کمک زبان لینگو قادر است به‌طور همزمان

۸ صوت متفاوت را کنترل و پخش کند، این کانالهای صدا دارای متدها و خواصی هستند که به آسانی امکان پخش و کنترل صدا، همچنین دسترسی به خصوصیات آن را فراهم میکنند. به عنوان مثال متد Play() جهت پخش یک صدا و خاصیت Volume جهت تست و تغییر حجم صدا به کار می رود.

خاصیتی به نام soundLevel وجود دارد که جزء خواص سطح بالای صداست (Top Level) و ولوم صدای خروجی نهایی دایرکتور را تنظیم می کند. به عنوان مثال فرمان Sound.SoundLevel=0_ موجب قطع کامل صدای خروجی تمامی کانالها می شود. دایرکتور از تکنولوژی جالبی به نام Shockwave Audio جهت فشرده سازی صداهای داخلی استفاده می کند که باعث کاهش چشمگیر حجم کلیه صداهای درون Cast می شود که جهت دسترسی به تنظیمات آن لازم است از زبانه ShockWave کادر محاوره‌ای Publish Setting استفاده کنید.

امکان استفاده از یک فایل دایرکتور درون فایل دیگر وجود دارد، کافی است توسط Import فایل را وارد کنید.

- در دایرکتور به راحتی می توان از فیلمهای Swf استفاده نمود و آنها را تحت کنترل در آورد. به عنوان نمونه: عملیات توقف (stop)، پخش مجدد (Play) و برگشت به ابتدا (Rewind). جهت وارد کردن محتویات flash کافی است آن را در یکی از کادرهای محاوره‌ای Import و یا Flash Asset Properties وارد کنید.

Activexها در حقیقت اجزای بصری (Visual) تشکیل دهنده فرمهای ویندوز هستند. مانند دکمه‌ها و برچسبها که به وسیله فرمان Activex → Control → Insert می توان آنها را وارد کرد.

امکان مشاهده متدها و رویدادهای اکتیوکس، همچنین مشاهده و ویرایش خصوصیات آن از طریق کادر Activex Control Properties وجود دارد. به کمک لینگو می توان از رویدادهای Activex جهت نوشتن هندلرها و از متدها و یا خصوصیات آن جهت کنترل Activex استفاده نمود.

Learn in English

Synchronizing media

To pause the playhead until a specified cue point in a sound or digital video is reached, use the Wait For Cue Point option in the Tempo dialog box. You can also use this method to wait for the end of the sound or digital video, even if it has no cue points. Cue points can also be used to trigger events that scripts can interpret

For example, you can use cue points to make text appear in time with narration. First, use a program such as Sony SoundForge to place cue points in the sound file that correspond to the times when you want the text to appear on Stage. In Director, use the Tempo dialog box to pause the playhead at the frame where the corresponding text appears until the voice-over reaches the proper cue point.

واژه نامه تخصصی	
Asset	دارایی
Busy	مشغول
Hold	نگاه داشتن
Inspector	ناظر
Occur	اتفاق افتادن
Pan	میزان کردن
Resume	از سر گرفتن
Rewind	باز پیچیدن

خودآزمایی:

- ۱- چگونه می‌توان به خصوصیات صدای وارد شده، مانند عمق بی‌تی آن دسترسی پیدا کرد؟
- ۲- چگونه می‌توان عمل هم‌زمان‌سازی صدا و تصویر را انجام داد؟
- ۳- شکل کلی فرامین مربوط به کنترل کانالهای صدا، مانند فرمان پخش یا توقف چگونه است؟ با دو مثال شرح دهید.
- ۴- مقادیر مورد استفاده جهت تنظیم حجم صدا (Volume) و بالانس صدا (Pan) در چه محدوده‌ای هستند؟ مفهوم آنها را شرح دهید.
- ۵- به کمک کدام خاصیت می‌توان به وضعیت یک کانال صدا دسترسی پیدا کرد؟ فرمان مربوطه را بنویسید.
- ۶- تفاوت دو خاصیت SoundLevel و Volume چیست؟
- ۷- فرمت SWA را شرح داده کاربرد آن را توضیح دهید.
- ۸- مفهوم Activex را به همراه یک نمونه از کاربرد آن در دایرکتور بیان کنید؟

پرسش‌های چهار گزینه‌ای

- ۱- دایرکتور قادر به پخش همزمان..... صدا می‌باشد.
(الف) ۲
(ب) ۴
(ج) ۸
(د) ۱۶
- ۲- کدامیک از کانالهای Score قابلیت تشخیص CuePoint ها را داراست؟
(الف) Sound1
(ب) Sound2
(ج) Tempo
(د) Transition
- ۳- کدامیک از گزینه‌های موجود در لیست CuePoint ها، قادر به نگهداری هد دایرکتور تا پایان پخش صدا می‌باشد؟
(الف) End
(ب) End Marker
(ج) Next
(د) Prev

۴- جهت پخش صدایی که در دومین سلول Cast ذخیره شده، کدام فرمان صحیح است؟

الف) Sound(1).Play(2)

ب) Sound(2).Play()

ج) Sound(1).PlayFile(2)

د) Sound(1).Play(Member(2))

۵- جهت پخش صدای Pause شده کانال هشتم، کدام فرمان صحیح است؟

الف) Sound(8).Play()

ب) Sound(8).Resume()

ج) Sound(8).FadeIn()

د) Sound(8).PlayNext()

۶- کدام دستور در مدت دو ثانیه حجم صدا را به تدریج کم می کند تا قطع شود؟

الف) Sound(1).FadeIn(2000)

ب) Sound(1).FadeOut(2000)

ج) Sound(1).FadeTo(0,2000)

د) موارد ب و ج

۷- یک صدا در کانال چهارم در حال پخش است، چنانچه دو فرمان Sound(4).Volume=255 و

و Sound.SoundLevel=0 اجرا شود، نتیجه چیست؟

الف) صدا با بالاترین حجم به گوش می رسد.

ب) هیچ صدایی به گوش نمی رسد.

ج) صدا با نصف حجم کامل به گوش می رسد.

د) پیغام خطا

۸- به وسیله کدامیک از زبانه های موجود در Publish Setting می توان عمل فشرده سازی

صداها را انجام داد؟

الف) Files Projector (ب)

ج) ShockWave Audio Compression (د)

۹- کدام گزینه در کادر محاوره ای Flash Asset Properties موجب پخش فایل Flash با

سرعت Tempo می گردد؟

الف) Rate → Lock Step

ب) Rate → Normal

ج) Rate → Fixed

د) DTS

۱۰- جهت انتقال هد کنترل به فریم شماره ۲۰ فیلم فلش موجود در کانال شماره ۵، کدام دستور صحیح است؟

الف) `Sprite(5).go(20)`

ب) `Sprite(5).goFrame(20)`

ج) `Sprite(5).Frame(20)`

د) `Sprite(5).gotoframe(20)`

۱۱- جهت وارد کردن یک ActiveX کدام گزینه صحیح است؟

الف) `Insert` → `Control` → `ActiveX`

ب) `Insert` → `Media Element` → `ActiveX`

ج) `Insert` → `ActiveX`

د) `Control` → `ActiveX`

۱۲- جهت بارگزاری فایل `Video.mpg` موجود در مسیر `E:\:` به وسیلهٔ اکتیوکس Windows Media Player کدام فرمان صحیح است؟

الف) `Sprite(1).Filename="e:\Video.mpg"`

ب) `Sprite(1).Path="e:\Video.mpg"`

ج) `Sprite(1).URL="e:\Video.mpg"`

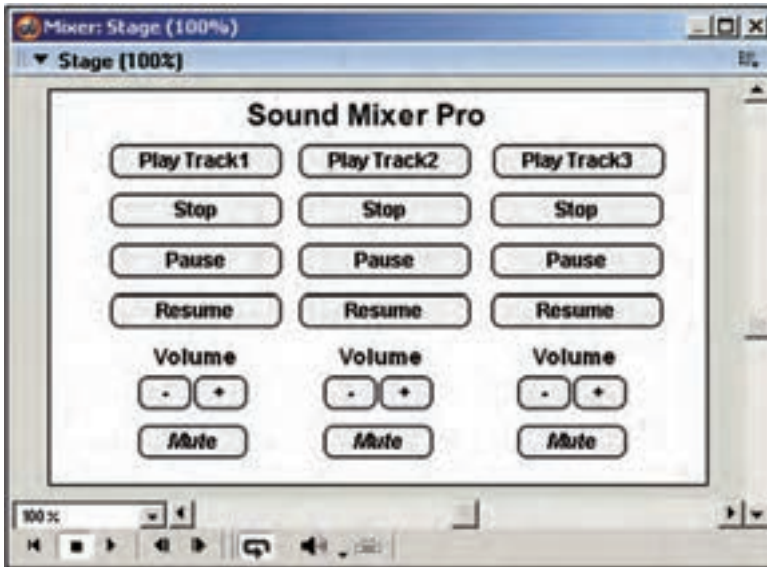
د) `Sprite(1).Link="e:\Video.mpg"`

13- Wait For Cue Point option is used to:

- a) Pause the playhead until a specified cue point in a sound is reached.
- b) make text appear in time with narration.
- c) wait for the end of the sound or digital video, even if it has no cue points.
- d) a, b And c.

کارگاه چندرسانه‌ای

۱- نمایش `Control sound` را طوری تغییر دهید که هر یک از صداها در یک کانال صوتی مجزا به همراه ولوم مجزا پخش شود. در ضمن هر یک از صداها دارای یک دکمه `Mute` جهت قطع صدا نیز باشند. (در واقع صداها باهم مخلوط شوند، مانند شکل ۲۳-۱۰)



شکل ۲۳-۱۰

۲- با استفاده از ActiveX Microsoft Web Browser نمایشی طراحی کنید که یک URL را در Stage نشان دهد. راهنمایی: از متد ("URL") Navigate استفاده کنید و جهت تست نمایش کافی است یک فایل Html ساده را ایجاد کرده، آن را به ActiveX معرفی کنید مانند فرمان `Sprite(1).Navigate("D:\Index.html")` یا در صورت اتصال به اینترنت از آدرس یک URL استفاده کنید. این ActiveX همچنین قادر به باز کردن مسیرهای محلی کامپیوتر می‌باشد مانند درایو D: می‌توانید مثال زیر را تست کنید.
`Sprite(1).Navigate("C:\")`

۳- یک مکالمه به همراه متن آن تهیه کرده و عمل همزمانی صدا و متن را در دایرکتور انجام دهید، سپس یک خروجی پروژکتور، (Exe) ، از آن تهیه کنید.