# *Creating Drag and Drop Elements with Collision Detection*

We are going to design an interface that requires items to be dragged onto another item as a point of navigation. In this example we are going to drag cd's onto an iMac, but these can be substituted for whatever project you wish.

•   The first thing to do is to create the graphic of the imac but exclude the cd drive call this symbol iMac. Then create the cd loading bay on a new symbol and call this cd drive.
•   Place both of these elements on layer 1 of the stage and label the cd drive in the instance panel "macload".
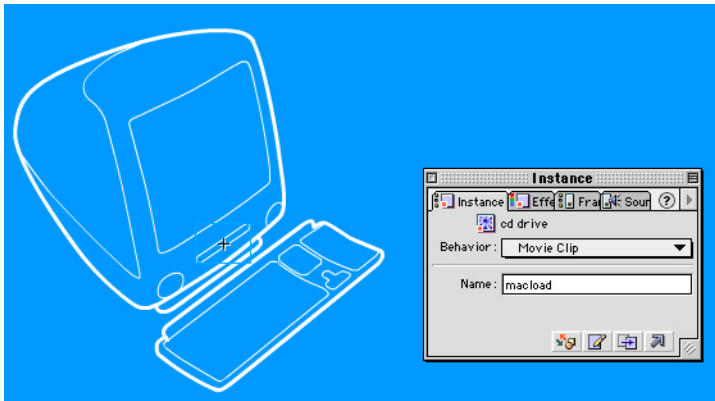
•   Now it's time to create the cd graphic, You can use a character from the font webdings, break it apart and skew it slightly to give it an isometric look. Copy this image.
•   Create a new button and in the hit state (so it's an invisible button) Paste the image, use the fill tool to fill it in.
•   Create a new movie clip symbol and call this "cd clip". In layer 1 drag the cd graphic off the library and label this layer cd.
•   Create a new layer and call this button, then drag your invisible button off the library, select it and bring up the actions by clicking the arrow in the bottom right of the main window. Select expert mode (Apple + E) and paste this text in:

```
on (press) {
        snapBack = false;
        startDrag ("");
}
on (release, releaseOutside, rollOut, dragOut) {
        stopDrag ();
        if (hitTest(_parent.macload) && !snapBack) {
                snapBack = false;
                play ();
        } else {
                snapBack = true;
        }
}
```

The on(press) command calls for this symbol to be able to start dragging it also switches snap back off. The on (release etc) command calls for this symbol to become undraggable. It then looks to see if it's over macload which is the name of the instance of the cd loading bay. If it is then snapback becomes false and the movie continues using the play(); command. If it is not over the hit area of macload then snapback becomes true and it will run the snap back behaviour.

•   We are now going to create the snap back behaviour. Create a new movie clip symbol and double click frame 1 to bring up the frame actions. Once again select expert mode and cut and paste the following text in:

```
xHome = _parent._x;
yHome = _parent._y;
```

This sets up the home location for where the snapback is to go to. In this case it will be wherever the parent is placed on the stage. The parent being the finished cd clip.

- Create a new keyframe in frame 2 and double click it to bring up frame actions again. Paste this text into the actions:

```
if (_parent.snapBack) {
        x = _parent._x;
        y = _parent._y;
        if (Math.abs(x-xHome)>1) {
                x = x+((xHome-x)/2);
                _parent._x = x;
        }
        if (Math.abs(y-yHome)>1) {
                y = y+((yHome-y)/2);
                _parent._y = y;
        }
}
```

This is the actual snap back code. It asks do we need snap back so the code looks for the 'flag' variable that the parent movie sets, so it uses _parent.snapBack to get it. The rest of the code is made to run if this is true. As soon as it is true we need to move the parent movie clip back home. The two if statements look to see if the clip within 1 pixel from home, if we're further away then the behaviour moves the parent closer to home.

The parent is moved by x=x+((xHome – x)/2) and the y=y+((yHome – y)/2). These look at the difference we want to be and move us one-second of that distance each time they're executed. As the clip gets closer to home the distance decreases and caused deceleration as if it's coming to a stop. The 2 in the expression affects the speed setting it higher will slow the snapping and vice versa.

- To ensure this behaviour is constantly monitored create key frame 3 and add this:

```
gotoAndPlay (2);
```

- This clip is now finished so go back to the cd clip and add a new layer called behaviour. Lock the other layers and drag the behavior onto the graphic (it can actually go anywhere). It will show up as a white spot.
- Add another layer and call this actions. In frame 1 of actions double click to bring up the frame actions and add this:

```
snapBack = false;
stop ();
```

Your cd clip should look like this:

- Now we need to add some animation to make the cd look like it's interacting with the cd drive. Place a key frame on the cd layer at frame 5. In the effect palette select alpha and set it to 0%. Then motion tween these frames so that the cd looks as if it disappears.
- Now add a frame at frame 2 to the button layer (F5).

- Now add a frame at frame 5 of the behaviour layer
- Now add a Keyframe at **frame 6** of the actions layer. Paste the following text into frame 6 actions:

```
getURL ("your link here");
stop ();
```

obviously make the link whatever it happens to be or alternatively you can go to another scene of your movie, load another movie in, whatever seems to be appropriate.

Your cd clip should look like this now:
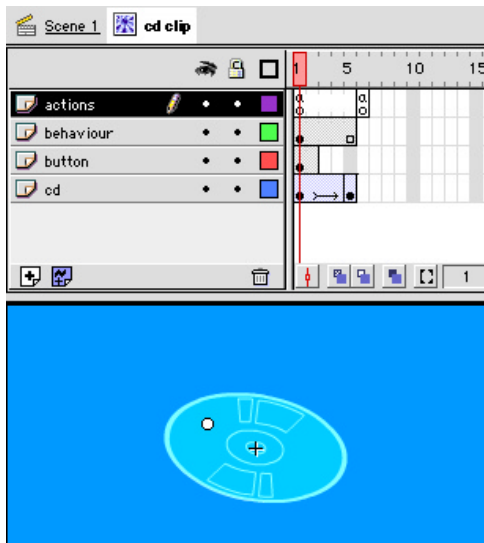
- Go the main movie, scene 1 and create a new layer to place your cd on. Now test your movie and it should work. If you need to add more cd's just duplicate the cd clip and add a different URL (because all the elements in the clip are from the library it's not going to take up much extra space.

Check out the file dragndropfinal.fla if you're unsure of anything.