



MAKE A HIDDEN OBJECTS GAME IN FLASH MX

In this tutorial, you will learn how to make a hidden objects game in Flash MX. An example of this type of game is the 'Garden Critters' game that is located in the Game Design section of the Flash Classroom gallery.


In this type of game, the user has to locate objects that have been hidden in an environment. The level of complexity of the game will depend on the level of detail the creator has gone to when developing the environment and hiding the objects within that environment. The 'Garden Critters' game has been designed for an early childhood audience and therefore most of the critters are easy to find.



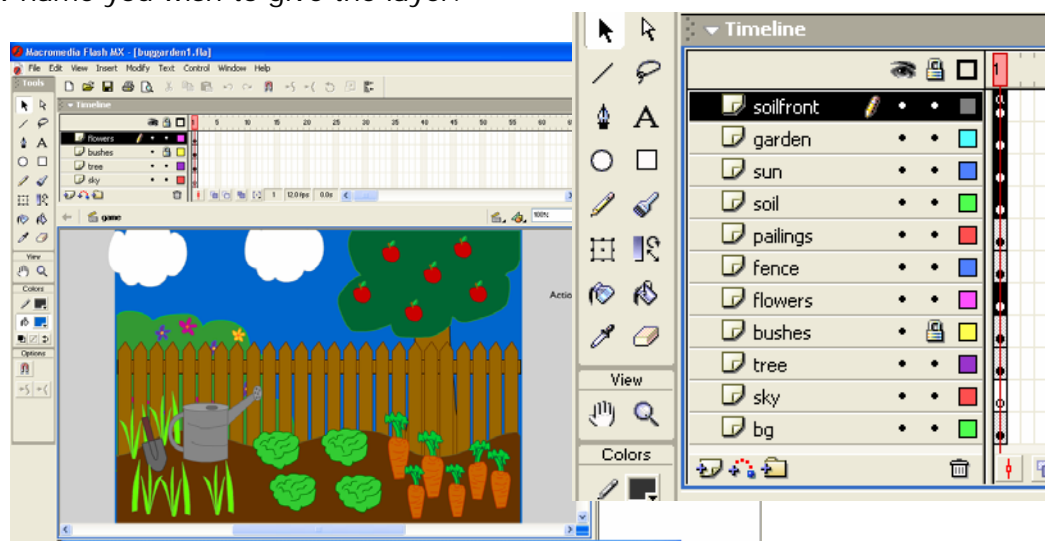
PART 1—MAKING THE ENVIRONMENT

1. Open a new document in Flash by selecting **File > New > Flash Document**.
2. The first part of the game design involves creating the environment where your objects will be hidden in. To do this, draw your background using the **Draw Tools** located on the tools panel on the left-hand side of the Flash interface.

In my 'Garden Critters' game, my environment had 10 layers. These are shown below. If you set up your environment well with multiple layers, you will be able to place your hidden objects partly behind different objects in the scene.

To add layers to your scene, select the **add layers** button  from the bottom of the timeline.

To rename layers, double click on the layer name (e.g. Layer 1) and type in the new name you wish to give the layer.






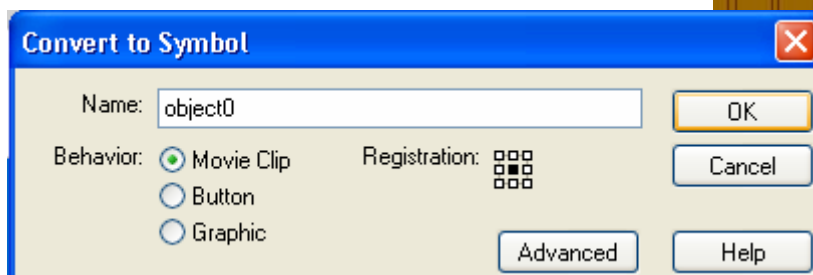
PART 2 — DESIGNING THE HIDDEN OBJECTS

Once you have created the background environment, it's time to develop the objects that you will hide in the environment.

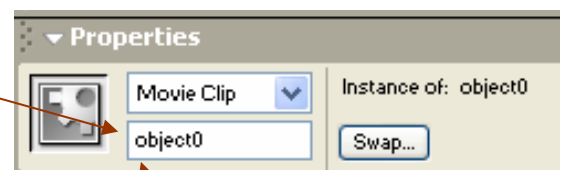
It's important to consider the target audience for your game when you start developing your objects. If your game is for young children, you may wish to ensure that the objects stand out in the environment. If the game is for older students, you may wish to make it harder by having only minimal contrast between the background and the object—this can be achieved by simply using an outline of the object or by choosing very similar colours. In the pictures above, you can see two different approaches to object design. The first features a snail whose major portion has been given the same colour as the fence paling. The second features an outline of a grasshopper that has been placed on a lettuce in the garden. Note that in the game the grasshopper is much harder to spot than the snail, though both are still relatively easy to find as the game has been designed for young children.



3. Create a new layer by selecting the  Add New Layer button. Double click on the Layer name and give it the name **Objects**.
4. In the grey area off the stage, **draw** the objects that are to be hidden. I have drawn 7 objects. You can create as many objects as you like—just remember the number as you will need to edit the actionscript later to make your game work.
5. Select each object individually and select **Insert > Convert to Symbol** or **F8**. Make sure that the symbols have the **Movie Clip** behaviour set and name them **object0**, **object1**, **object2**, **object3** and so on.



6. Each object will also need to be given a matching Instance Name. Select each of the objects you have converted to movie symbols and in the properties panel at the bottom, enter an instance name for that object. (see below)
7. Hide the objects in the environment. You may wish to copy them onto other layers if you wish to partially hide them behind elements of the environment.



Enter Instance Name Here.

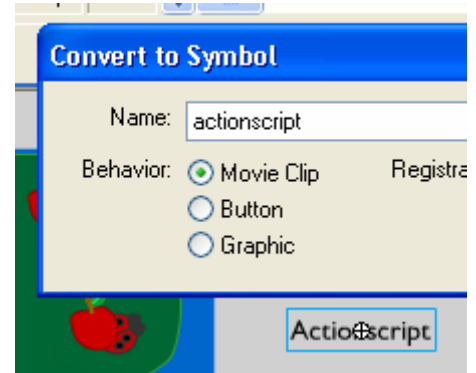


PART 3 — ADDING THE SCRIPT TO MAKE YOUR GAME WORK

If you have created all of the objects and have hidden them, you are now ready to develop the actionscript that will make your game work. This script will be added to a movie clip symbol called 'Actionscript' that we will sit just off the edge of the game.

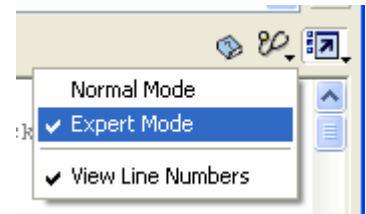
7. Using the text tool **A** type the text **ACTIONSRIPT** in the grey area just off the right hand side of the stage.

8. Select this text and choose **Insert > Convert to Symbol**. Enter the name **actionscript** and check the **Movie Clip** behaviour option. Select **OK** to convert the text to the symbol format.



9. The script that makes our game work is going to be added to this movie clip. To do this, select the Actionscript movie clip and press **F9** to open the **Actions Panel**. Note that you can also open this panel by selecting **Window > Actions**.

10. On the right hand side of the actions panel, you will see a button containing an arrow. Click on this button to open the drop down menu shown here. Select **Expert Mode** so that you can type the script below straight in.



11. Enter the following script into the panel. This is the first part of the script and it is creating an array that will enable Flash to keep track of the number of hidden objects that the game player has located.

```
onClipEvent (load) {  
    found = [];  
    for (i=0; i<7; i++) {  
        found[i] = false;  
    }  
}
```

This script is telling the Flash Player that when it loads the Actionscript Movie Clip, it is to start a loop that will continue to run until the user has clicked on all seven hidden objects.



FLASH CLASSROOM TUTORIAL—MAKE A HIDDEN OBJECTS GAME IN FLASH MX

12. Enter the following script under the script you have just added into the actions panel. Note that if you have more than 7 objects, you will need to change all of the 7's in the script to the number of objects you have.

```
onClipEvent (mouseDown) {
    x = _root._xmouse;
    y = _root._ymouse;
```

← This part of the script is tracking the location of the players click.

```
for (i=0; i<7; i++) {
    if (_root["object"+i].hitTest(x, y, false)) {

        myColor = new Color(_root["object"+i]);
        myColor.setTransform({rb:255, bb:255, gb:255})

        found[i] = true;
        break;
    }
}
```

← This part of the script is a looping through all of the hidden objects to check if they have been clicked.

It also changes clicked objects to a new colour. In this script, the colour has been set using RGB colour code to white.

```
gameover = true;
for (i=0; i<7; i++) {
```

← This part of the script is a loop that tests whether all of the objects have been found.

```
if (found[i] == false) {
    gameover = false;
}
}
```

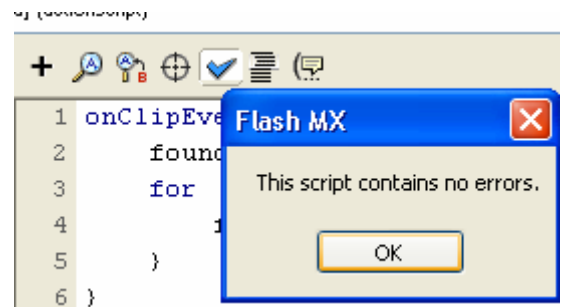
← If some objects are still to be found, the game will continue.

```
if (gameover) {
    _root.gotoAndPlay("feedback");
}
}
```

← If all of the objects have been found, the Flash Player will take the user to the first frame in the next scene. This frame has been given the frame label 'feedback'.

13. Check that your script is correct by clicking on the blue tick button at the top of the panel.

If your script is correct, you will see the box shown at the right. If it isn't an output panel will appear showing you the error. If there is an error, fix it before moving on to the next step.





PART 4 — ADDING ADDITIONAL SCENES FOR THE HOME PAGE & FEEDBACK

At present you have a scene containing your game. If you test it, by selecting **Control > Test Movie**, you should see that your objects change colour when you click on them. Even though you have added the script to get the game to move on to another scene once all objects have been found, this part isn't working because we haven't created that scene yet.

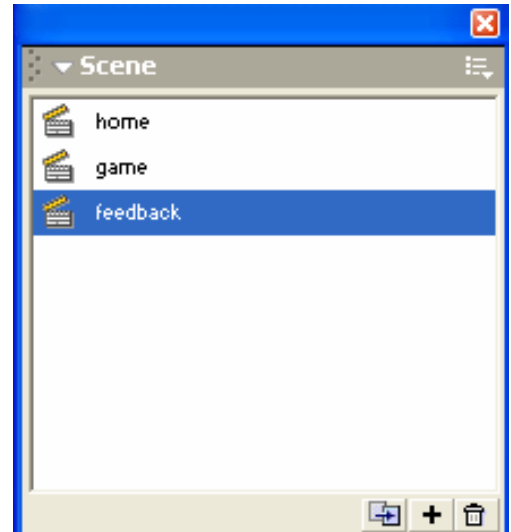
- 14. To add the scene that will let the player know they have successfully finished the game, select **Insert > Scene**.

- 15. In the scenes panel that appears, double click on the scenes and rename them.

Rename the scene with your game in **game**.

Rename the scene you have just added **feedback**.

- 15. Click on the **+** button on the bottom of the scenes panel to add a third scene.



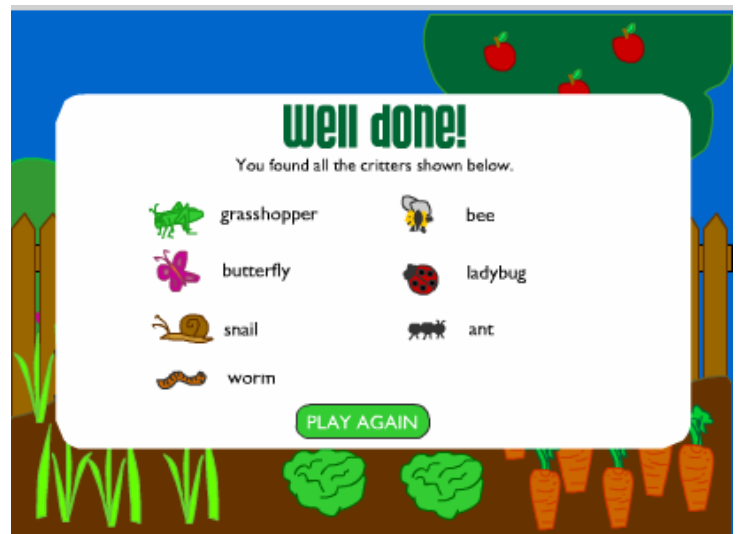
Rename this scene **home** and click on it and drag it to the top. Your scenes panel should look like the one shown above.

Making the Feedback Scene

- 16. Select the **feedback** scene from the Scenes panel. This will make this the current scene.
- 17. Use the drawing and text tools to create the feedback you want the player to receive once they have successfully located all of the hidden objects.

The picture at the side shows the feedback provided to players who successfully find all of the hidden critters in the 'Garden Critters' game.

Note that I copied and pasted the background from the game scene and then added a new layer to put the rounded rectangle, text and play again button on.

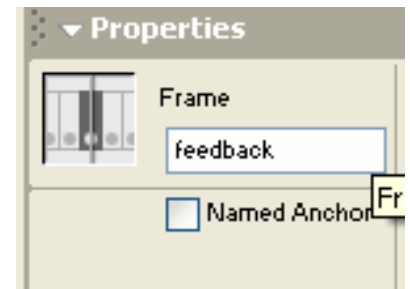




FLASH CLASSROOM TUTORIAL—MAKE A HIDDEN OBJECTS GAME IN FLASH MX

18. Select the first keyframe on the timeline in the Feedback Scene and in the Properties panel, give this frame the name **feedback**.

Note that this name is referenced in the last part of the actionscript that you added to the Actionsript Movie Clip in an earlier step. These two names must match in order to make your game work.



19. To enable the player to stop and read the Feedback you have put in this scene, add a **stop action** to the **first keyframe** on the timeline.

To do this, select the first keyframe and then open up the Actions Panel by selecting **F9**. Type in the stop action below to make it work:

```
stop();
```

20. To enable the player to play the game again, add a **PLAY AGAIN** button to the scene.

To create a button, follow these steps:

- Use the draw and text tools to create the button design.
- Convert the button design to a button symbol by selecting the design and choosing **Insert > Convert to Symbol**. Give your button a name and select the **Button** behaviour. (shown right)



At this point, if you know how to edit buttons to give them multiple states for a rollover effect, you may do so. If you are new to Flash and button design, continue on with just this simple button for now. You can learn more about making your buttons interactive by taking some of the **Button Design tutorials** that are available in the Flash Classroom Getting Interactive Tutorial Collection.

21. To make this button work, select the button and then enter the following script in the actions panel:

```
on(release){  
    gotoAndPlay("home", 1);  
}
```

Your button should now take you to the home scene when clicked. However, you probably haven't set this scene up with it's own stop action or button yet—so let's do this before you test your game.



FLASH CLASSROOM TUTORIAL—MAKE A HIDDEN OBJECTS GAME IN FLASH MX

22. You are now going to design your home scene. In this scene, you should put the name of the game, simple instructions on how to play the game and a button that allows the user to play the game.

To begin, use the **draw** and **text tools** to create the design of your choice. Remember to include a name for the game and some basic instructions. The instructions shown here are those from the home scene of the Garden Critters game.



23. Create a **Start** button by following these steps:
- Use the draw and text tools to create the button design.
 - Convert the button design to a button symbol by selecting the design and choosing **Insert > Convert to Symbol**. Give your button a name and select the **Button** behaviour.
24. To make this button work, select the button and then enter the following script in the actions panel:

```
on(release){  
    gotoAndPlay("game", 1);  
}
```

25. You will also need to add a stop action to the first keyframe in this scene. To do this, **select the first keyframe** and enter the following action into the actions panel:

```
stop();
```

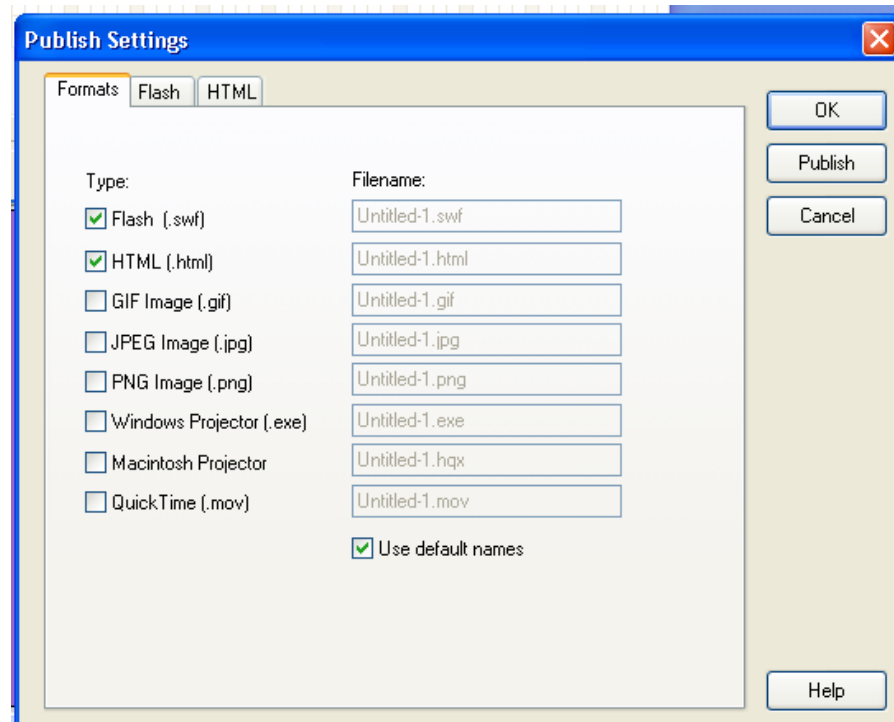
That's just about it! To test your game, select **Control > Test Movie**. You should be able to move between the scenes and find the hidden objects. If not, go back over this tutorial to check where you went wrong. If you're still having problems, email your file to the address below and we'll check it for you.

If you're happy with your file, move on to the next page to learn how to publish and share your game.



PUBLISH & SHARE YOUR WORK

26. Save your work by selecting **File > Save**.
27. Turn your flash file into a game that can be played on any computer by publishing it in different file formats. To do this select **File > Publish Settings**. The box shown below will appear.



28. Tick the file formats you want and click on the **Publish** button. These files will be saved in the same location you saved your original file. If you want your game to be a standalone file that can be played on Windows or Macintosh machines—ensure you check the Windows Projector (.exe) and Macintosh Projector (.exe) format options.

ENHANCE YOUR GAME

If you've got to this point, why not learn more about game design and how to enhance your game with a customized mouse or sound by completing the following Flash Classroom Tutorials

Working with Sound Changing the Mouse Cursor tutorial