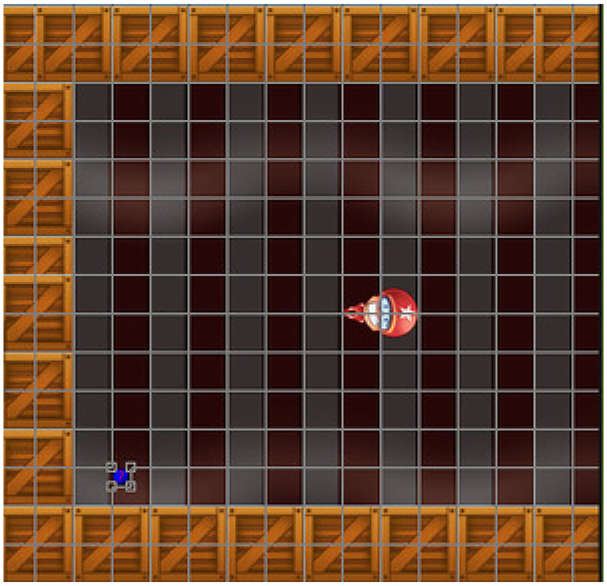
Getting Started with Your First Game



Catch the Clown Tutorial

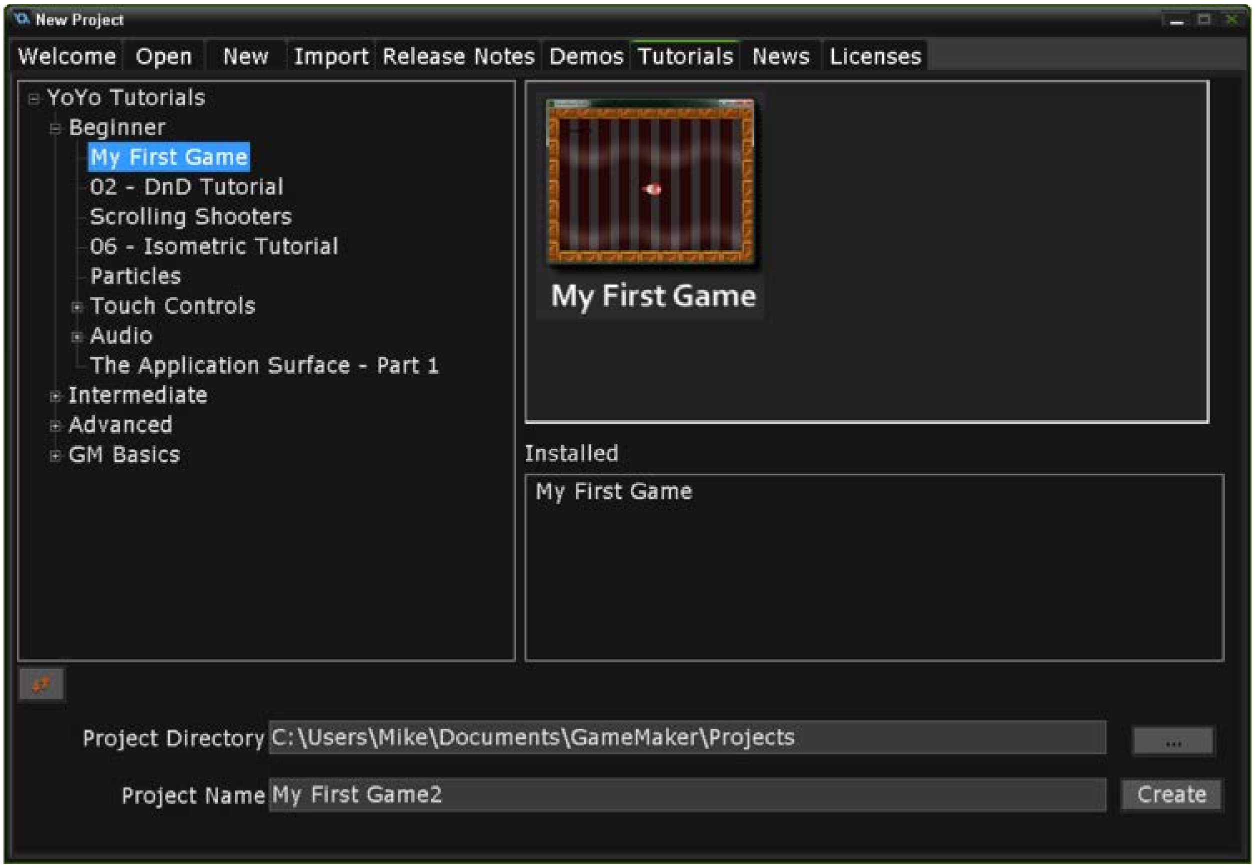
In this tutorial, I will guide you through making your (possibly) first game with GameMaker: Studio. This tutorial is appropriate for those who have never opened the GameMaker software and will show you, step-by-step, how to make a simple game. This tutorial includes:

- Using the Main Menu to Add and Create Assets
- Creating Your First Object with Events and Actions
- Adding Instances to the Game Room
- Saving and Testing Your Game
- Adding an Alarm
- Creating a Score Display and Background Music with a Controller Object

This tutorial is based on the in-software tutorial My First Game, where you learn how to make *Catch the Clown*; this is an action game in which the player tries to click (or tap if playing on a mobile device) on the image of a clown that is bouncing around the screen. The game does get progressively harder each time the player successfully clicks on the clown, and hence, catches the clown.
To open the tutorial, follow these steps:

1. Start the GameMaker software.
The New Project window appears.

2. From the menu, click the Tutorials tab (Figure 1).
A list of tutorials appears on the left.



Project Directory C:\Users\Mike\Documents\GameMaker\Projects

Project Name My First Game2

Figure 1: The New Project Window

3. In the left column, click the + next to Beginner.
A list of Beginner tutorials appear.

*****4. Double-click My First Game.**
GameMaker downloads and installs the tutorial. If you have already installed the tutorial, then the tutorial opens.

Next

Adding Sprites and Sounds to *Catch the Clown*

This section jumps straight to the action to get you started quickly making a game in GameMaker: Studio (GM:S). Here, I show you how to:

- Find the Game Assets Folder: this is where you can find assets, such as Sprites and music, for *Catch the Clown*.

- How to load Sprites and Sounds: GM:S makes it easy to add your assets through the Resources menu.
- *Catch the Clown* uses two different Sprites, a Sprite is basically an image that you can use as an asset in your game. Sprites can be a single image or can consist of several images grouped together, known as sub-images.

You can find all of the assets for the different tutorials and projects within your My Documents folder on your PC. When you installed GameMaker: Studio, the software saved all of the assets for the different tutorials in the following directory: My Documents > GameMaker > Projects. Within the Projects folder, you will find all of the assets for both the projects and the tutorials.

You can quickly access the tutorial assets from the main menu, click, Help > Open Project In Explorer. This simply opens the appropriate folder where the assets are saved depending on which project or tutorial you are working on.

Loading Sprites

To load the clown Sprite, follow these steps:

1. Open the My First Game tutorial.
The tutorial opens with the in-software directions in the right column. Feel free to click Close in the bottom corner. By closing the in-software tutorial, you will free up some screen real estate for you to work (if you don't have a very large monitor). This tutorial covers not only the information from the in-software tutorial, but it also contains additional information from yours truly.

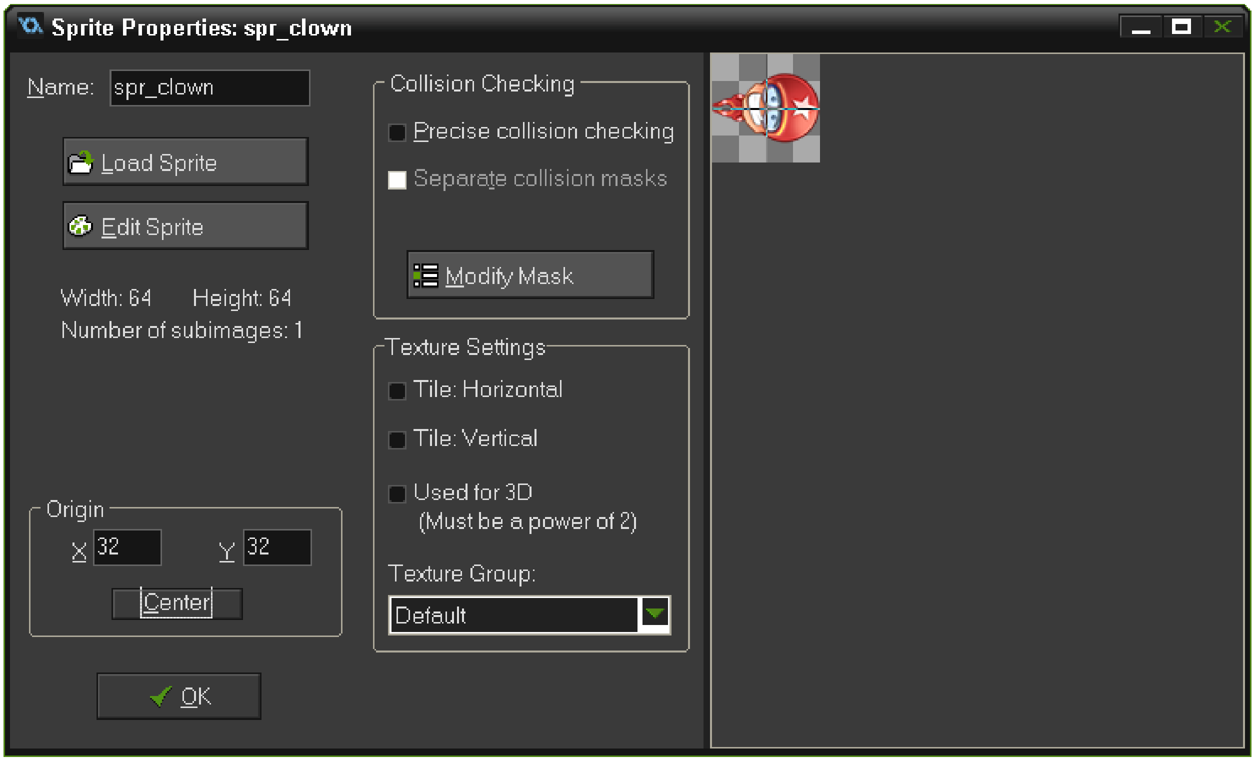
2. From the main menu, click Resources > Create Sprite.
The Sprite Properties appears.

3. In the Name field, type spr_clown.

4. Click Load Sprite.
A navigation window opens.

5. Click assets > clown.png > Open.
You should now see the clown image in the Sprite Properties.

6. Click Center as shown in the figure to set the Origin.
The X and Y boxes populate with 32, as those coordinates represent the center of the image.



Sprite Properties: spr_clown

Name: spr_clown

Width: 64 Height: 64

Number of subimages: 1

Origin

X 32 Y 32

Collision Checking

□ Precise collision checking

□ Separate collision masks

Texture Settings

□ Tile: Horizontal

□ Tile: Vertical

□ Used for 3D (Must be a power of 2)

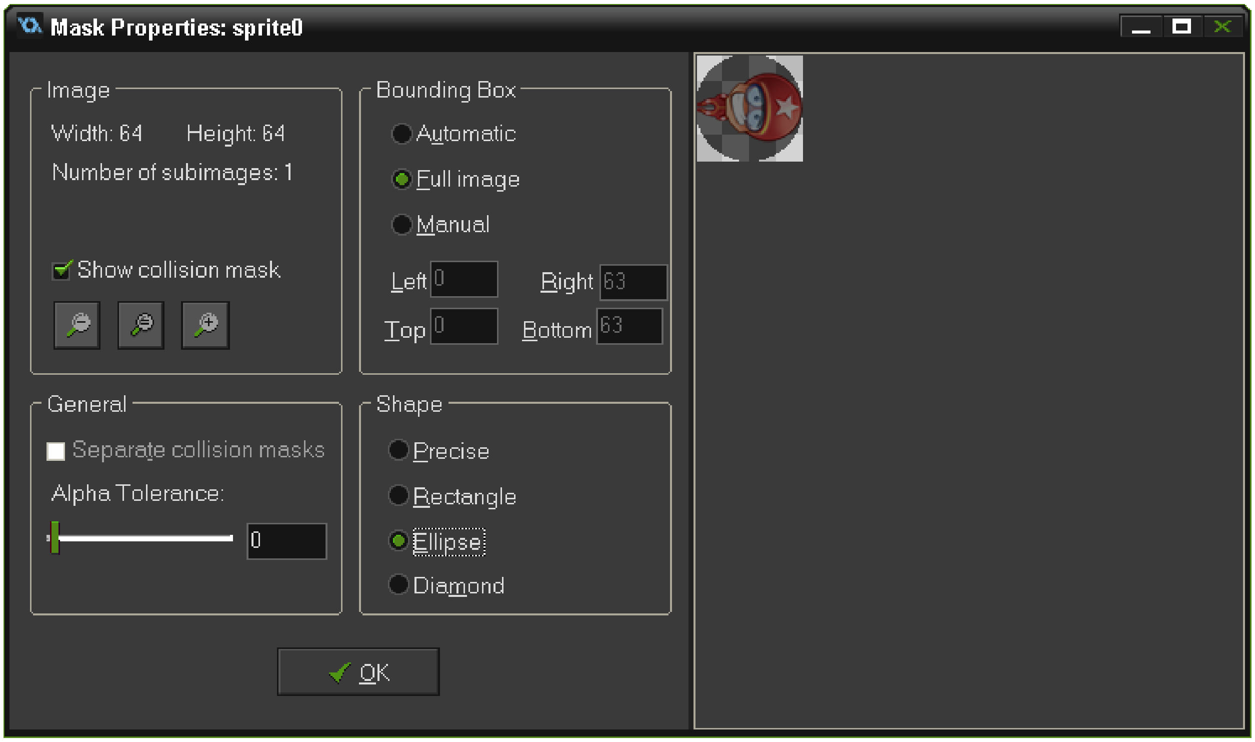
Texture Group: Default

Sprite Properties with Origin Centered

7. In the Collision Checking section, click Modify Mask.
The Mask Properties appear.

8. In the Bounding Box section, select Full Image.
GameMaker uses the entire area of the image for the Mask.

9. In the Shape section, select Ellipsis as shown in the figure on the next page.
GameMaker uses an ellipse shape for the Mask. Changing the collision Mask does affect how the Sprite behaves and reacts during the game.



Mask Properties: sprite0

Image

Width: 64 Height: 64

Number of subimages: 1

☑ Show collision mask

Bounding Box

○ Automatic

● Full image

○ Manual

Left 0 Right 63

Top 0 Bottom 63

General

□ Separate collision masks

Alpha Tolerance: 0

Shape

○ Precise

○ Rectangle

● Ellipse

○ Diamond

10. Click OK from the Mask Properties.
The Mask Properties is saved and closed.

11. Click OK from the Sprite Properties.
The Sprite Properties are saved and closed.

After you complete this procedure to load the clown Sprite, the Resource tree on the left side of the screen should now show your Sprite.

Now, you need to load in the Sprite that will represent the wall in the game. I'll run through the steps again, but this time with fewer details.

Load the wall Sprite by following these steps:

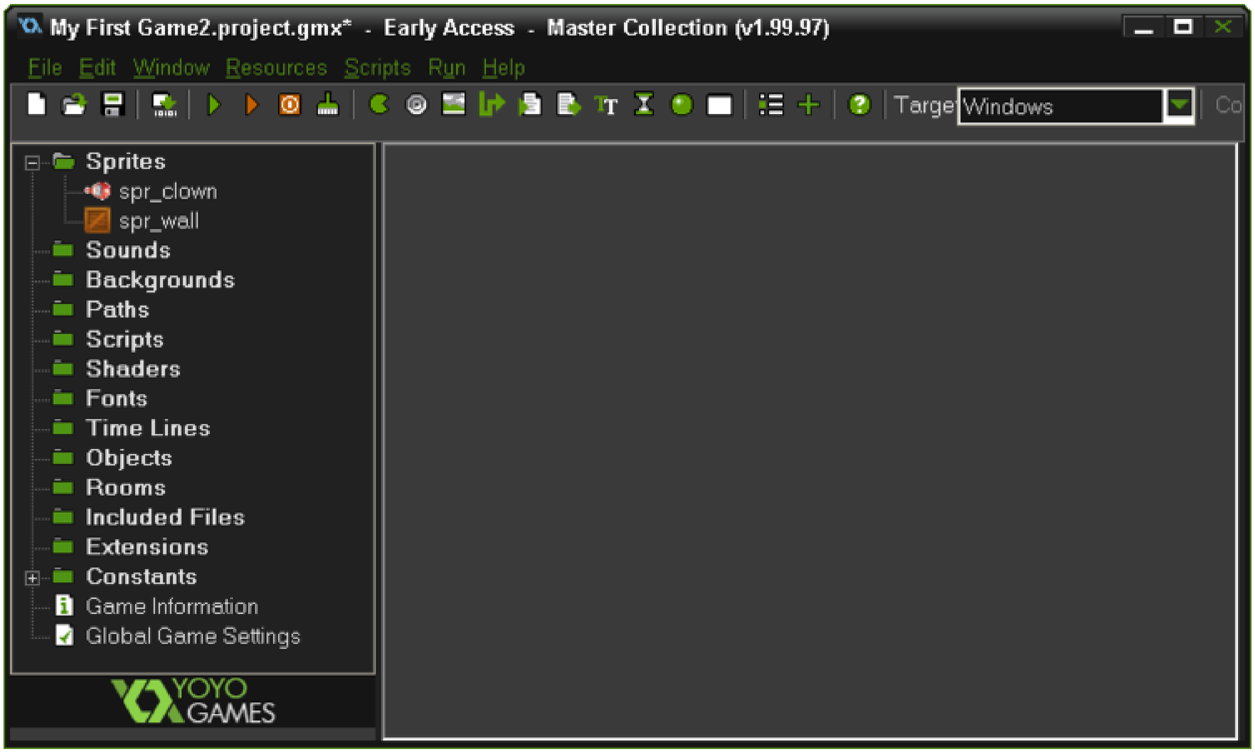
1. From the main menu, click Resources > Create Sprite.

2. In the Name field, type spr_wall.

3. Click Load Sprite, select wall.png, and then click Open.
The wall appears in the Sprite Properties.

4. Click OK.
The Sprite Properties is saved and closed.

You should now see both the clown and the wall Sprites in the Resource tree.
For the wall Sprite, we did not have to change the Origin or the Collision Mask as the default properties were appropriate.



Next

Loading the Sounds

Now that you have the Sprites for the game loaded in, it's time to load in the sound effects. GameMaker can handle multiple formats for sound effects, including:

- wav files – are good for short sound effects, such as guns blasting and explosions exploding.
- .mp3 files – are good for long sound effects, such as background music.
- .ogg files – are good for high quality digital multimedia.

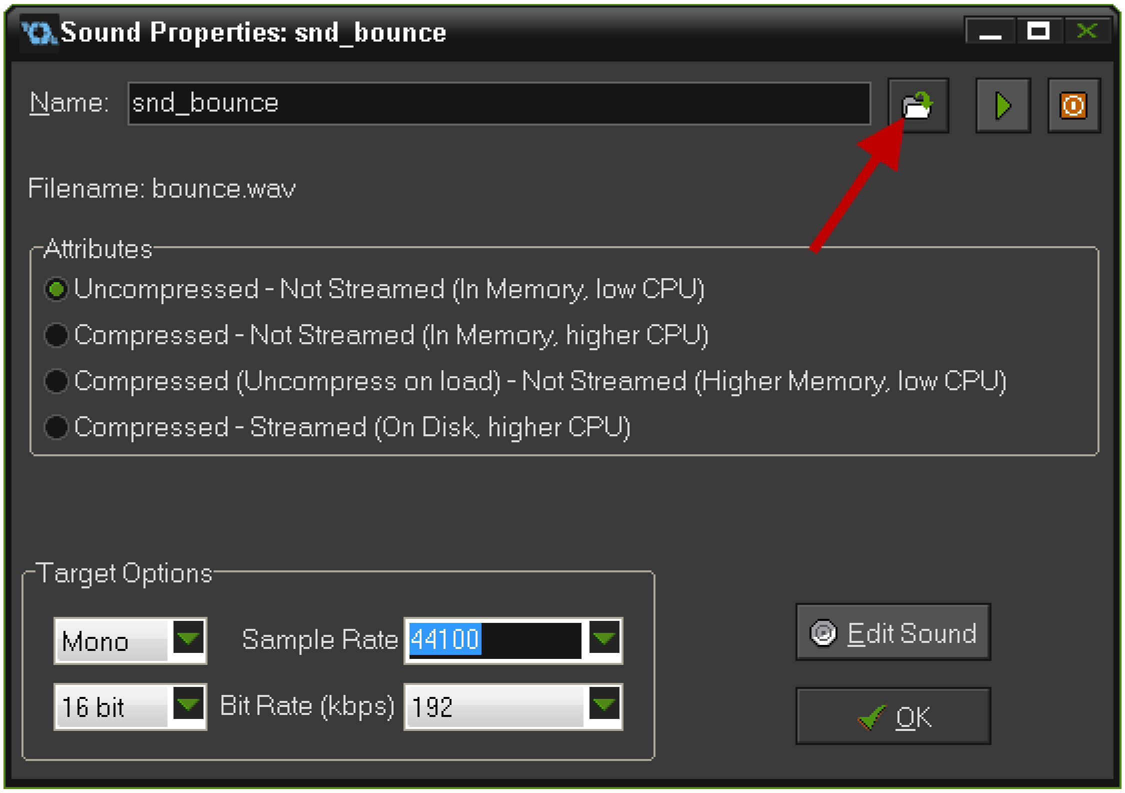
You can load sounds into GM in much the same way as Sprites. For *Catch the Clown*, you will load in both a bounce Sound and, later on, you will add some background music.

To load a Sound effect, follow these steps:

1. From the main menu, click Resources > Create Sound.
The Sound Properties appear.

2. In the Name field, type snd_bounce.

3. Click Load Sound From a File icon as indicated in the figure.
A navigation window appears.


Sound Properties: snd_bounce

Name: snd_bounce

Filename: bounce.wav

Attributes
- ● Uncompressed - Not Streamed (In Memory, low CPU)
- ○ Compressed - Not Streamed (In Memory, higher CPU)
- ○ Compressed (Uncompress on load) - Not Streamed (Higher Memory, low CPU)
- ○ Compressed - Streamed (On Disk, higher CPU)

Target Options
- Mono
- Sample Rate 44100
- 16 bit
- Bit Rate (kbps) 192

The Sound Properties

***4. Click the assets folder > bounce.wav > Open (from the bottom-right corner of the navigation window).**
The navigation window closes. **You** can now click the Play icon to hear the sound you just loaded. To stop the sound from playing, click the Stop icon.

5. Click OK.
The Sound Properties are saved and closed.

Now that you have the bounce Sound loaded, it's time to load in the click Sound. This is the same procedure as what you just did, so I'll step through the process a little quicker this time.

To load the click Sound effect, follow these steps:

1. From the main menu, click Resources > Create Sound.
The Sound Properties appear.

2. In the Name field, type snd_click.

3. Click Load Sound.
The navigation window appears.

4. Double-click, click.wav.
The navigation window is saved and closed.

5. Click OK.
The Sound Properties are saved and closed.

You should now see both Sound files in the Resource tree directly beneath your Sprites.

Now that you have your Sound and Sprite assets loaded into GM, it's time to create some **Objects**.

Creating the Clown and Wall Objects

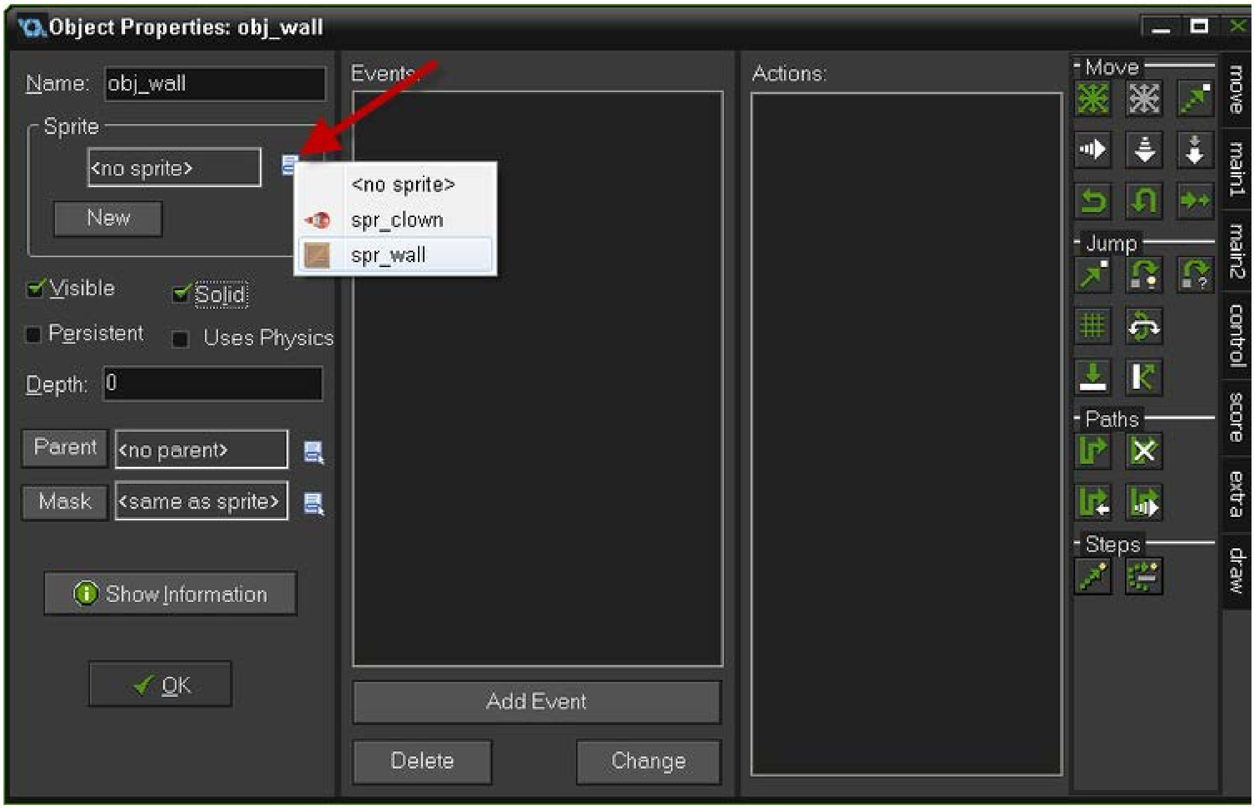
Now that you have your Sprites loaded into GM, it's time to do something with them. After all, Sprites are simply images that don't do anything (besides the collision Mask). You have to assign them to an Object. Basically, Objects are defined by their reactions to Events. An Event helps define the Object, such as how it moves or acts in the game. Each time you place an Object into the Room, it becomes an Instance. Therefore, you can have multiple Instances of a single Object in each game. In *Catch the Clown*, you will have two different Objects – the clown and the wall – and you will place a single Instance of the clown Object into the Room and several Instances of the wall Object in the Room.

To create the wall Object, follow these steps:

1. From the main menu, click Resources > Create Object.
The Object Properties appear.

2. In the Name field, type obj_wall.

3. Click the Sprite drop-down as shown in the figure on the next page.
A list of the Sprites that you loaded previously appears.


Object Properties: obj_wall

Name: obj_wall

Sprite: <no sprite>

☑ Visible ☑ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

<no sprite>
spr_clown
spr_wall

The Sprite drop-down shows available Sprites.

4. Choose spr_wall from the drop-down list.
A thumbnail of the Sprite appears and the name of the Sprite populates the field.

5. Select Solid from beneath the Sprite section.
This setting does not allow anything to penetrate this Object.

6. Confirm that Visible is also checked.
This makes the Object actually visible on the screen as compared to being an invisible Object.

7. Click OK.
The Object Properties are saved and closed.

You should now see obj_wall in the Resource tree in the Objects folder.

Now it's time to create the clown Object, which you will do in a similar manner to the wall Object, but with some different properties.

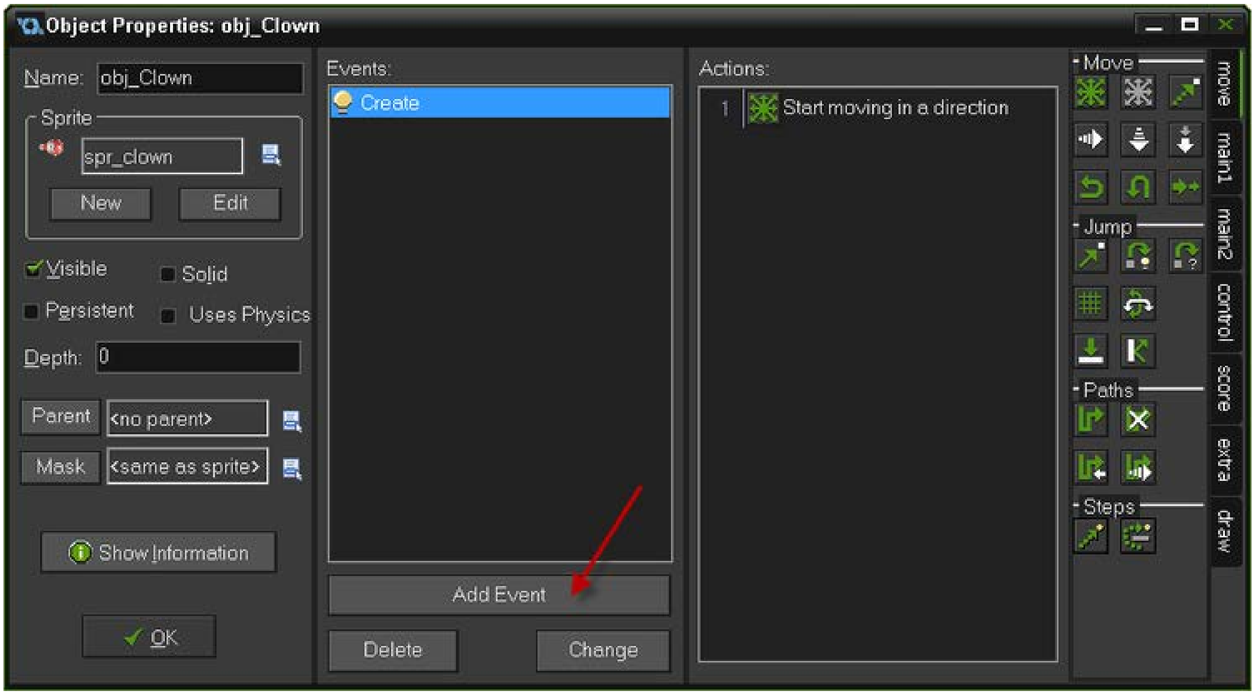
To create the clown Object, follow these steps:

1. From the main menu, click Resources > Create Object.
The Object Properties appear.

2. In the Name field, type obj_Clown.

3. Click the Sprite drop-down and choose the clown Sprite.
The image and name of the clown Sprite appear.

4. Click the Add Event button, found near the bottom middle of the window, as indicated in the figure.
The Choose the Event to Add menu appears.



Object Properties: obj_Clown

Name: obj_Clown

Sprite: spr_clown

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Events: Create

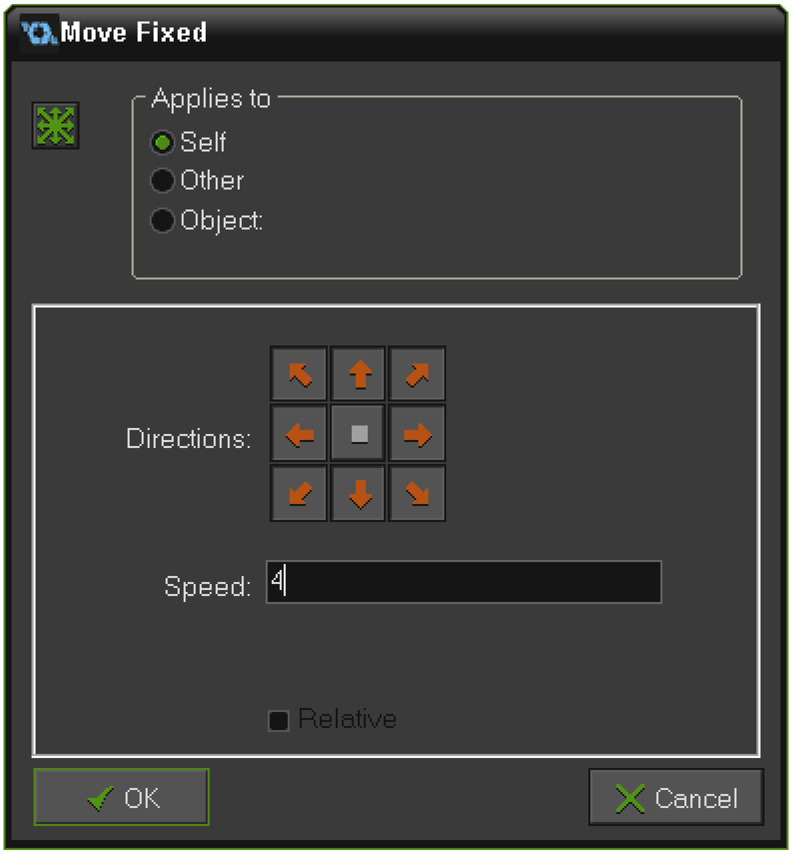
Actions:
1 Start moving in a direction

Click Add Event to open the Events menu.

5. Click Create from the menu.
A Create Event now appears in the Events section of the Object Properties.

6. From the far right of the Object Properties, find the Move tab and select it (if it’s not already selected).
The Move Actions appear along the right column.

7. Drag and drop the Fixed Move Action into the Actions section as shown in the figure.
The Move Fixed Action properties appear, as shown on the next page.



Move Fixed

Applies to

● Self

○ Other

○ Object:

Directions:

Speed: 4

□ Relative

8. Select each of the eight arrows as shown in the figure.
The Object will move in a random direction based on the arrows selected.

9. In the Speed field, type 4.
This sets how fast the Object moves during the game.

10. Click OK.
The Move Fixed properties are saved and closed.

The direction arrows indicate which direction the Object moves. When more than one direction is selected, the Object will move in a random direction, based on which arrows you selected. The middle square indicates no motion at all. In this case, do not select the middle square as you want this Object to move.

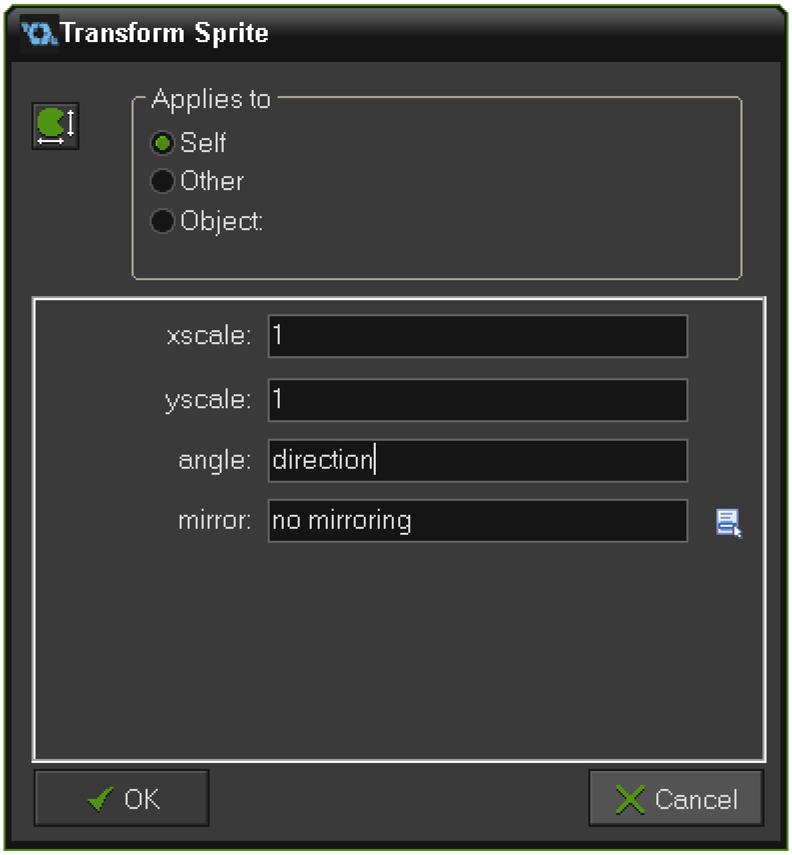
Now, you need to configure the Transform Sprite Action.

1. Drag and drop the Transform Sprite Action from the Main1 tab to the Actions section.
The Transform Sprite properties appear.

2. In the Applies to section, select Self.
This Action will apply to this Object, and not another Object.

3. For the Xscale and Yscale fields, leave the default of 1.
You are not going to change the scale of the Sprite.

4. In the Angle field, type *direction* as shown in the figure.
The Sprite will now face the direction it is moving. *Direction* is a special variable that GM uses to determine the movement of an Instance.



Transform Sprite

Applies to
● Self
○ Other
○ Object:

xscale: 1
yscale: 1
angle: direction
mirror: no mirroring

5. For the Mirror field, leave the default of No Mirroring.
The Sprite will not have a mirror image.

6. Click OK to save and close the Transform Sprite properties.

You just added an Action that changes the angle the clown Sprite is drawn at every time it changes direction while moving. Remember, in this game, the clown moves about the game Room in random directions and it's up to the player to try and click on it, hence, *Catch the Clown*.

After you create an Action, you can copy and paste it to apply to other Events. This is useful in that you don't have to keep recreating the same Actions over and over. In this example, you can copy the Transform the Sprite Action and paste it into new Events that you will create next. I'll remind you about this at the appropriate time.

****Next***

Adding a Collision Event

At this point in the tutorial, you have set the behavior of the clown Object by adding an Event, an Action, and setting the Action properties. You are now ready to add a Collision Event. In this case, the Collision Event triggers the bounce Sound to play when the clown bounces against a wall.

To add a Collision Event, follow these steps:

1. With the obj_clown properties open (see the last procedure), click Add Event > Collision > obj_wall.
The Collision Event appears in the Events section, directly under the Create Event. You might notice that the Actions box is now empty. That's because you haven't added any Actions for the Collision Event yet. If you're wondering what happened to the Actions for the Create Event, just click on Create from the Events section and the Actions for that Event appear. Now it's time to add a Bounce Action to the Collision Event.

2. With the Collision Event highlighted, drag and drop the Bounce Action from the Move tab into the Actions section.
This opens the Bounce Action properties.

3. Leave the default values.
The Object will bounce against solid objects and won't act precisely, as that isn't necessary for this game.

4. Click OK.
The Bounce Action properties are saved and closed.

Now set up the Play Sound Action.

1. Drag and drop the Play Sound Action from the Main1 tab into the Actions section.
The Play Sound properties appear.

2. Click the Sound drop-down and choose snd_bounce.

3. Leave the default property for the loop field as False.
The sound will only play once when triggered.

4. Click OK to save and close the Play Sound properties.

You have now finished setting the properties for the Collision Event for the wall Object.

The loop property is used to play the sound either once or continuously. To have the sound play continuously, type in “true” to have the sound play once, type in “false.”

You can edit both Events and Actions after you have worked with them. Simply right-click on an Event or Action to open a menu.

Now you need to copy the Transform the Sprite Action from the Create Event and paste that into the Collision Event. That way, the Sprite will change its angle when it bounces off the wall Object. I’ll step you through that next.

To copy and paste an Action from one Event to another, follow these steps:

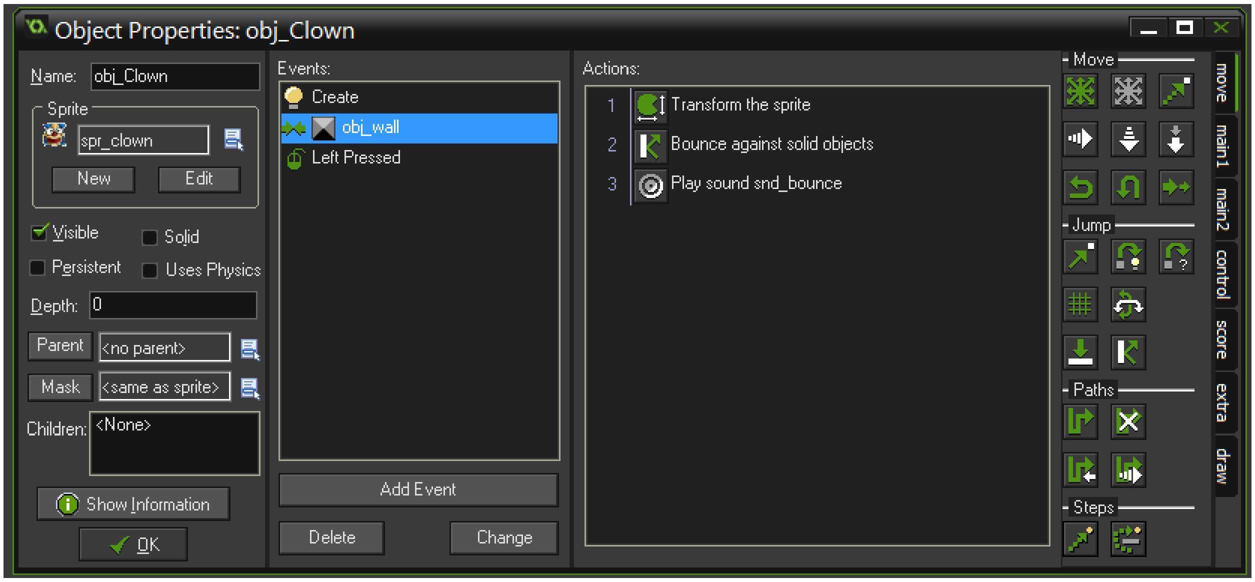
1. Select the Create Event.
The Create Event Actions appear.

2. Right-click the Transform the Sprite Action and choose Copy.

3. Select the obj_wall Collision Event.
The Collision Event Actions appear.

4. Place your cursor over the Actions section, right-click and choose Paste.

The Transform the Sprite Action appears. If you do not copy and paste this Action into the Collision Event, then the angle of the clown Sprite will not change when it hits a wall.



Object Properties: obj_Clown

Name: obj_Clown

Sprite: spr_clown

☑ Visible □ Solid

□ Persistent □ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:
- Create
- obj_wall
- Left Pressed

Actions:
1 Transform the sprite
2 Bounce against solid objects
3 Play sound snd_bounce

Next

Adding a Mouse Event

During the game, when the player clicks the clown (clicks work as taps on mobile devices) four different Actions occur: the player's score increases; the click sound plays; the clown jumps to a new direction; and the clown's speed increases. The following procedures show you how to set up these Actions.

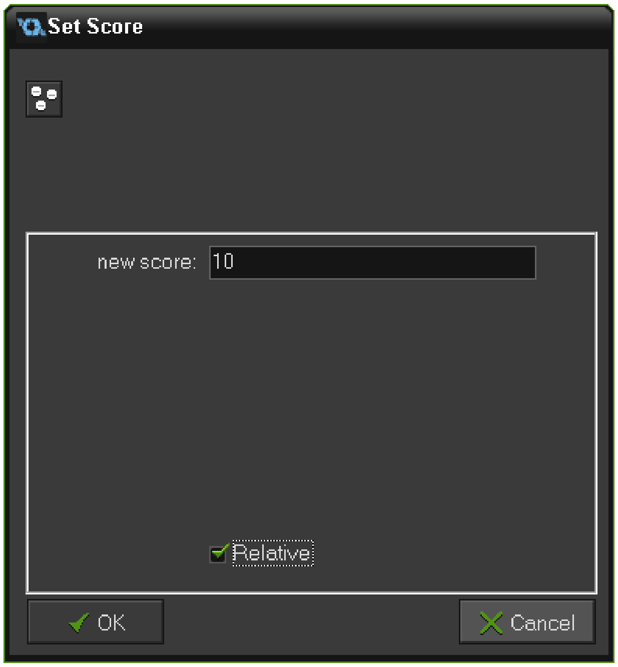
To add the Mouse Event, follow these steps:

1. With the obj_clown properties open, click Add Event > Mouse > Left Pressed.
The Left Pressed Event appears in the Events section.

2. Drag and drop the Set Score Action from the Score tab into the Actions section.
The Set Score properties appear.

3. In the New Score field, type 10.
Now, when the player clicks the clown with the left mouse button, the score increases by 10.

4. Select Relative as shown in the figure.
This adds 10 points to the player's score each time they catch the clown (otherwise, the score would stay at 10).



new score: 10

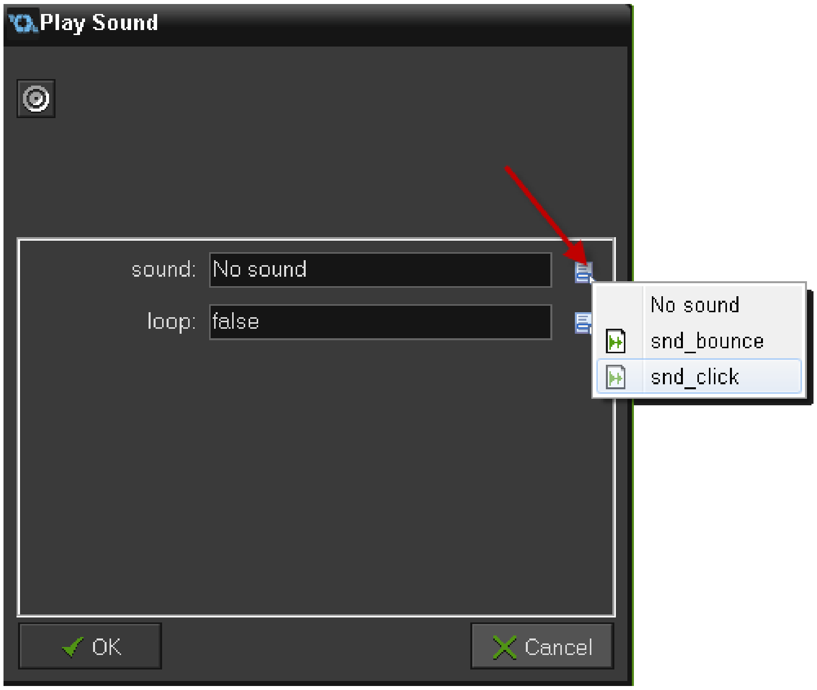
☑ Relative

5. Click OK to save and close the Set Score properties.

Now configure the Play Sound Action.

1. Drag and drop the Play Sound Action from the Main1 tab into the Actions section.
The Play Sound properties appear.

2. Click the Sound drop-down and choose snd_click as shown in the figure.



Play Sound

sound: No sound

loop: false

3. For the Loop field, leave the default of False.
Now, the click sound will play once when the clown collides with the wall.

4. Click OK to save and close the Play Sound properties.

Next, configure the Jump to Random Action.

1. Drag and drop the Jump to Random Action from the Move tab to the Actions section.
The Jump to Random properties appear.

2. Leave the default parameters.
The Object will jump to a random position in the Room when the player clicks (or taps) on the clown during the game.

3. Click OK to save and close the Jump to Random properties.

Now to configure the Move Fixed Action.

1. Drag and drop the Move Fixed Action from the Move tab into the Actions section.
The Move Fixed properties appear.

2. In the Applies to section, select the Self radio button.

3. Select all eight arrows, remembering not to select the middle square.

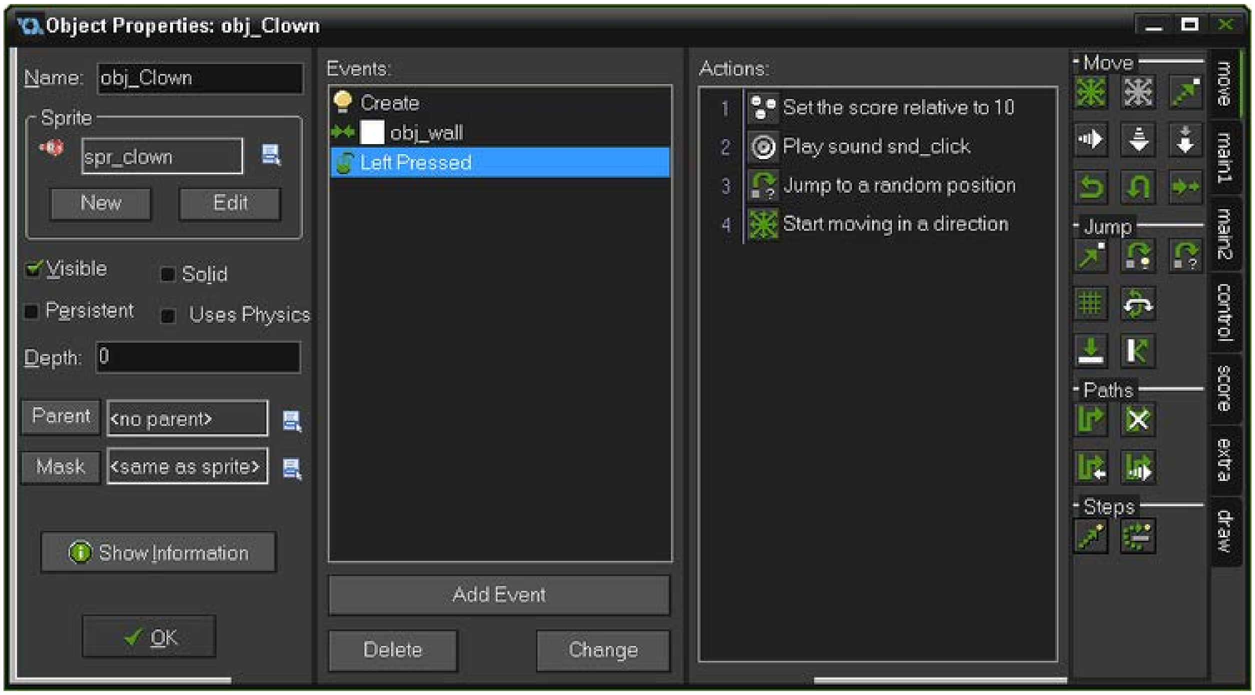
4. In the Speed field, type 0.5.

5. Select Relative.
Now, each time the player clicks the clown, the Speed will increase by 0.5.

6. Click OK to save and close the Move Fixed properties.

Originally, you set the Speed of the clown Object to 4. Now, when the player clicks on the clown during the game, the speed of the clown increases by 0.5. Therefore, the first time you click the clown, the speed increases to 4.5, the second time the speed increases to 5, the third time to 5.5 and so on.

You now have a Left Pressed Event with four defined Actions that sets the score, plays a sound, makes the clown jump, and change direction as shown in the figure.


Object Properties: obj_Clown

Name: obj_Clown

Sprite: spr_clown

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

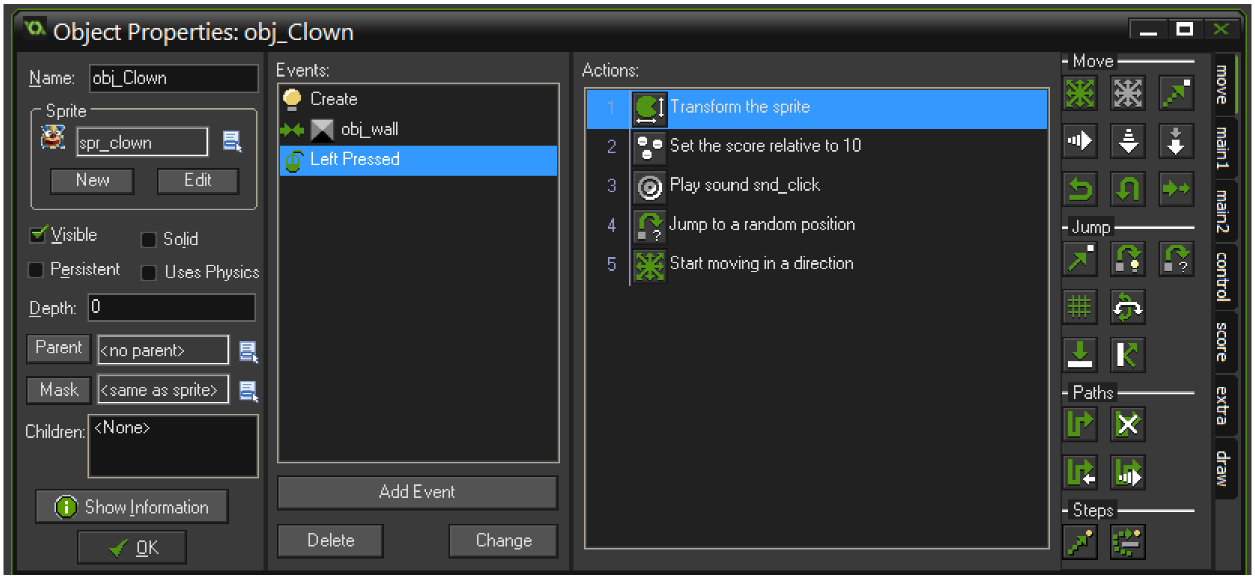
Mask: <same as sprite>

Events:
- Create
- obj_wall
- Left Pressed

Actions:
1. Set the score relative to 10
2. Play sound snd_click
3. Jump to a random position
4. Start moving in a direction

After completing these steps, you are ready to move on to the next step of creating the game Room.

Remember to copy and paste the Transform the Sprite Action from the Create Event into the Left Pressed Event (see the previous procedure). Otherwise, the angle of the clown Sprite will not change when the player clicks the clown. At this point, you can click OK to save and close the Object Properties. In the next section, I show you how to create the Room.



Name: obj_Clown

Sprite: spr_clown

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent: <no parent>

Mask: <same as sprite>

Children: <None>

Events:

- Create
- obj_wall
- Left Pressed

Actions:

1 Transform the sprite
2 Set the score relative to 10
3 Play sound snd_click
4 Jump to a random position
5 Start moving in a direction

Click the OK Button. Now click on File and Save.

Next

Creating the Game Room

So far in this tutorial, you have made the Objects that you need to place into the game Room. Now is the time to create the Room itself. This includes adding a background image, background music, and a score display. After this process, I will suggest that you save and test your game. When you test your game, you might find that it's rather easy to play. That's why I'll add some extra steps to add a little more difficulty to Catch the Clown.

To add a Room to your game, follow these steps:

1. With your Catch the Clown tutorial open, and picking up from the last section, from the main menu, click Resources > Create Room.
The Room Properties window appears.

2. Click the Settings tab.
The Settings properties appear for the Room.

3. In the Name field, type in rm_main.
Note that the default value of Snap X and Snap Y are set to 32.
So let's change them X 64 Y 64.

By setting the Snap value to the same as the clown Sprite, it makes it easier to place the clown Instance at the correct location in the Room. **If you change the Snap values to say, 64, you will see the size of the grey squares become much bigger**. In the next step, you will place an Instance of the clown Object in the game Room. When you do this, you will notice how the clown Object snaps to the grid lines.

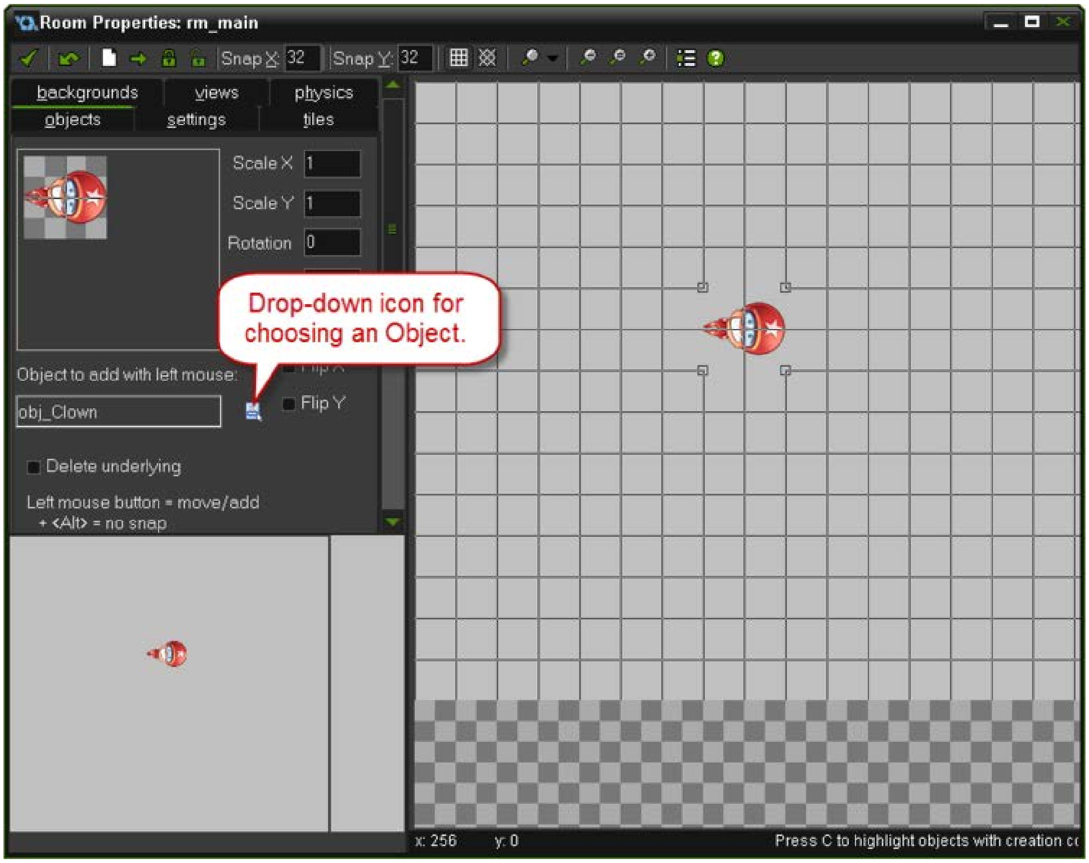
Adding Instances to the Game Room

Now that you have a game Room set up, you can add Instances of your Objects to the Room. The following procedure picks up from where you left off in the last section.

To add Instances to the game Room, follow these steps:

1. With the Room Properties window open, click the Objects tab.
The Objects properties for the Room appear.

2. Click the drop-down icon as indicated below and select obj_clown.
The Sprite for obj_clown appears in the top-left corner.


Room Properties: rm_main

Snap X: 32 Snap Y: 32

Scale X 1

Scale Y 1

Rotation 0

Object to add with left mouse: obj_Clown

Drop-down icon for choosing an Object.

x: 256 y: 0

Press C to highlight objects with creation co

3. Place one Instance of the clown Object into the game Room by clicking in the center of the grey squares as shown above.
The Sprite for the Object appears in the Room and it is now considered an Instance of that Object. You could place more than one clown Object in the Room, if you did, then you would have two Instances of the clown Object in the Room.

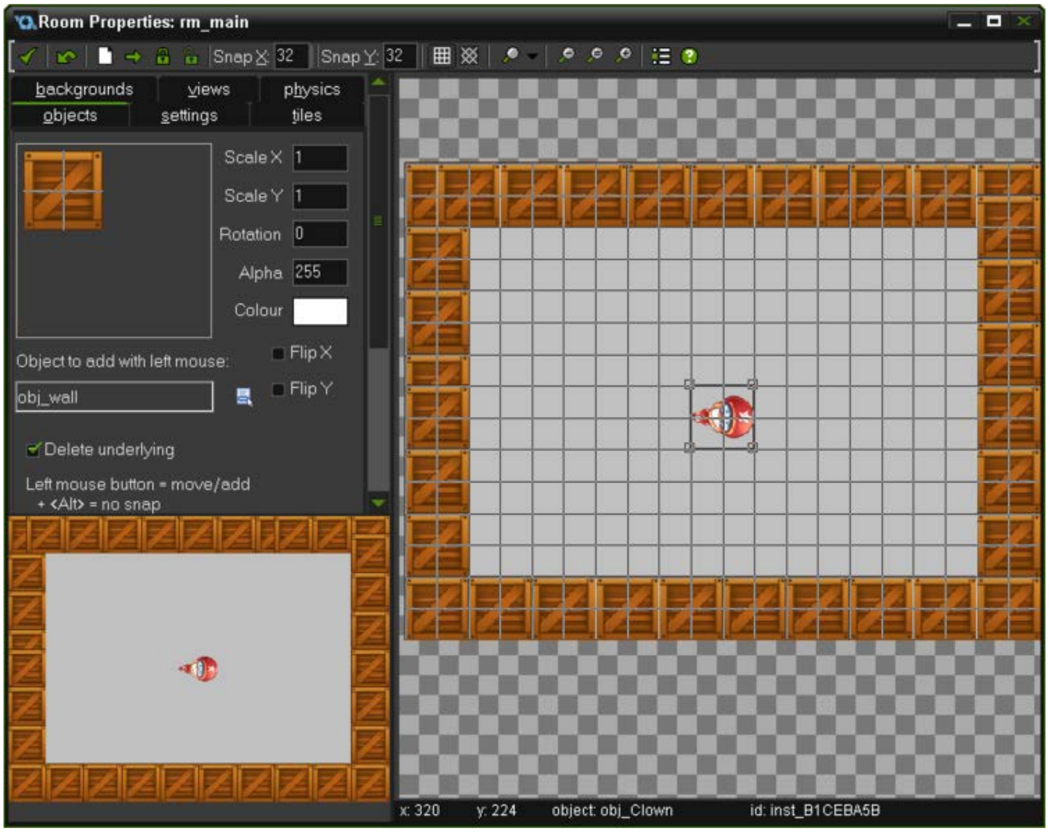
Notice the small grey box in the bottom left corner. This is the mini-view and it provides a small-scale view of the entire room, while a more detailed view is on the right. You can navigate the Room on the right by clicking and dragging within the mini-view.

Now, you need to place several Instances of the wall Object in the Room for the clown to bounce off of during the game.

To add the wall Object to the Room, follow these steps:

1. With the Room Properties window open from the last procedure, and the Objects tab selected, click the drop-down and then choose obj_wall.
The wall Sprite appears in the top left corner.

2. Click on the cells that border the Room to place multiple Instances of the wall Object.
This creates a walled border. Your game Room should now look like the figure.



Snap X 32 Snap Y 32

Scale X 1

Scale Y 1

Rotation 0

Alpha 255

Colour

Object to add with left mouse: obj_wall

□ Flip X

□ Flip Y

☑ Delete underlying

Left mouse button = move/add
+ <Alt> = no snap

x: 320 y: 224 object: obj_Clown id: inst_B1CEBA5B

To place multiple Instances of the same Object at once, you can hold down Shift+Ctrl, then left click and hold with the mouse as you drag. Select Delete Underlying to avoid placing more than one Instance in the same spot. If you accidentally place an Instance where you don't want it, you can click and drag to move it. To delete the Instance, right click and choose Delete.

4. Click the green check mark to save and close the Room Properties.
The Room now appears in the Resource tree on the left.

Now it's time to add a background to the game Room.

Adding a Background Image

Currently, your game Room consists of a grey background. That isn't too appealing, is it? In the following procedure, I show you how to add a nice image as the background.

To add a background image, follow these steps:

1. From main menu, click Resources > Create Background.
The Background properties appears.

2. In the Name field, type bck_main.

3. Click Load Background.
The navigation window appears.

4. From the navigation window, open the assets folder > select background.png > Open (from the bottom right of the window).
The image now appears in the Background Properties window.

5. Leave the default settings.
This background image will tile both horizontally and vertically, meaning, the image will fill the Room.

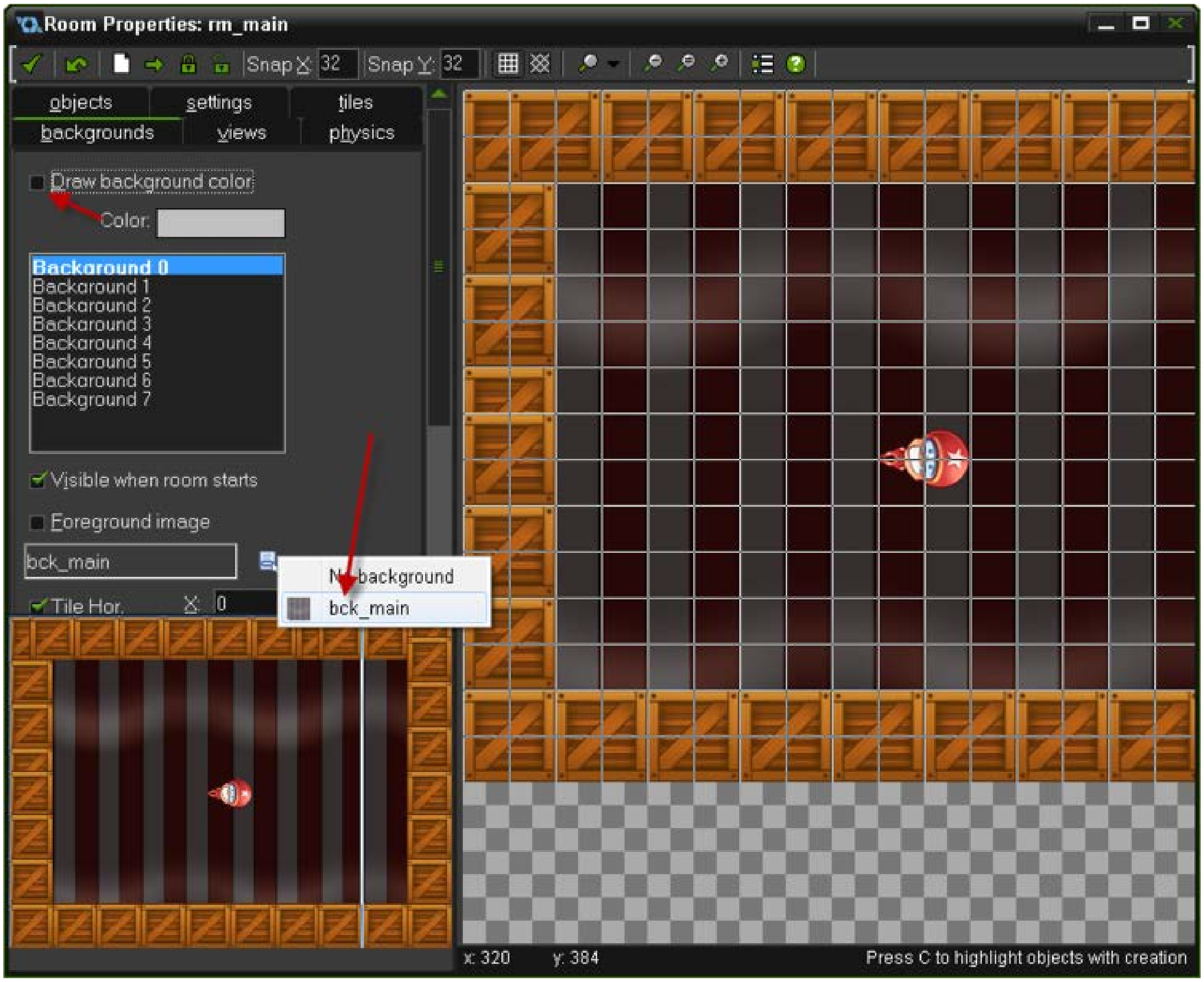
6. Click OK to save and close the Background properties.
You should now see the background in the Resource tree.

Now to add the background to the Room.

1. From the Resource tree, in the Room folder, double-click the rm_main to open it.
The Room Properties appear.

2. Within the Room Properties, select the Backgrounds tab.
The Background properties for the Room appears.

3. Untoggle Draw background color (so that it's not checked) as shown below.
GameMaker will not draw a background color for the Room.



Room Properties: rm_main

Snap X: 32 Snap Y: 32

□ Draw background color

Color:

Background 0
Background 1
Background 2
Background 3
Background 4
Background 5
Background 6
Background 7

✓ Visible when room starts

□ Foreground image

bck_main

✓ Tile Hor. X: 0

x: 320 y: 384

Press C to highlight objects with creation

4. Click the drop-down and choose bck_main.
The background image should now appear in the game Room.

5. From the icon menu, click the green check mark to save and close the Room Properties.

By default, the Tile Hor and Tile Ver settings should already be checked. If not, go ahead and check them. By doing so, even if your background image is small and does not fill the Room, by enabling the Tile Hor and Tile Ver settings, the background is filled by the image. To experiment, uncheck these two boxes and see the effect it has on the background.

Now that you have a game Room set up with Instances of your Objects in place, you can save and test your game.

Saving and Testing Your Game

At this point in the tutorial, you have created enough of Catch the Clown to save it and test it. Saving and testing games is important to catch errors as you work.

To save and test your game, follow these steps:

1. From the main menu, click File > Save.
Follow the prompts to save your game.

2. From the main menu, click Run > Run Normally.
GameMaker then takes a moment to compile the game and it should start running.

You can now play the game as you have created it so far. Basically, the clown should start moving and when it hits a wall, it should bounce off and change angle and direction. If you click the clown, it should change angle, direction, and the speed should increase. If you have your sound turned on, you should hear the sound effects when the clown is clicked and when it bounces off a wall.

This is all fine and good, but you have more work to do to complete this game. In the following sections you will learn how to create an Alarm to increase the difficulty of the game, and how to add background music and a score display.

Adding an Alarm to the Clown

In this section, I show you how to create an Alarm. Alarms are Events that can contain Actions that you set to trigger for something to happen at a predetermined time. In this case, the Alarm changes the direction of the clown every two seconds regardless if it hits a wall or not. This should add some difficulty to the game to provide a greater challenge to the player.

To add an Alarm to an Object, follow these steps:

1. Open the clown Object. From the Resource tree, in the Objects section, double-click obj_clown.
The Object Properties appear for obj_clown.

2. Select the Create Event.
The Actions for the Create Event appear in the Actions section.

3. Drag and drop the Set Alarm Action from the Main2 tab to the Actions section.
The Set Alarm properties appear.

4. In the Applies to section, select the Self radio button.

5. In the Number of Steps field, type 60.
This triggers the Alarm after two seconds. The default Room speed is for 30 steps to occur each second.

6. For the In Alarm No: field, leave the default of Alarm 0, as that is the Alarm you want to trigger.

7. Click OK to save and close the Set Alarm properties.
You just configured a Create Event to trigger Alarm 0 after 60 steps of the game.

Now to create Alarm 0.

1. Click Add Event > Alarm > Alarm 0.
The Alarm 0 Event appears in the Events section.

2. Drag and drop the Move Fixed Action from the Move tab into the Actions section.
The Move Fixed properties appear.

3. In the Move Fixed properties, select all of the arrows.

4. In the Speed field, leave the default value of 0.
You only want to change the angle and direction of the clown with this Alarm and not increase the speed.

5. Select Relative.
The Object will maintain its relative Speed.

6. Click OK to save and close the Move Fixed properties.
The Move Fixed Action window is saved and closed.

When Alarm 0 is triggered after 60 steps, the clown will change direction. Next, set up another Alarm, to trigger Alarm 0 once every 60 steps (so that the clown keeps changing direction every 60 steps).

1. Drag and drop the Set Alarm Action from the Main2 tab to the Actions section.
The Set Alarm window appears.

2. In the Applies to section, select the Self radio button.

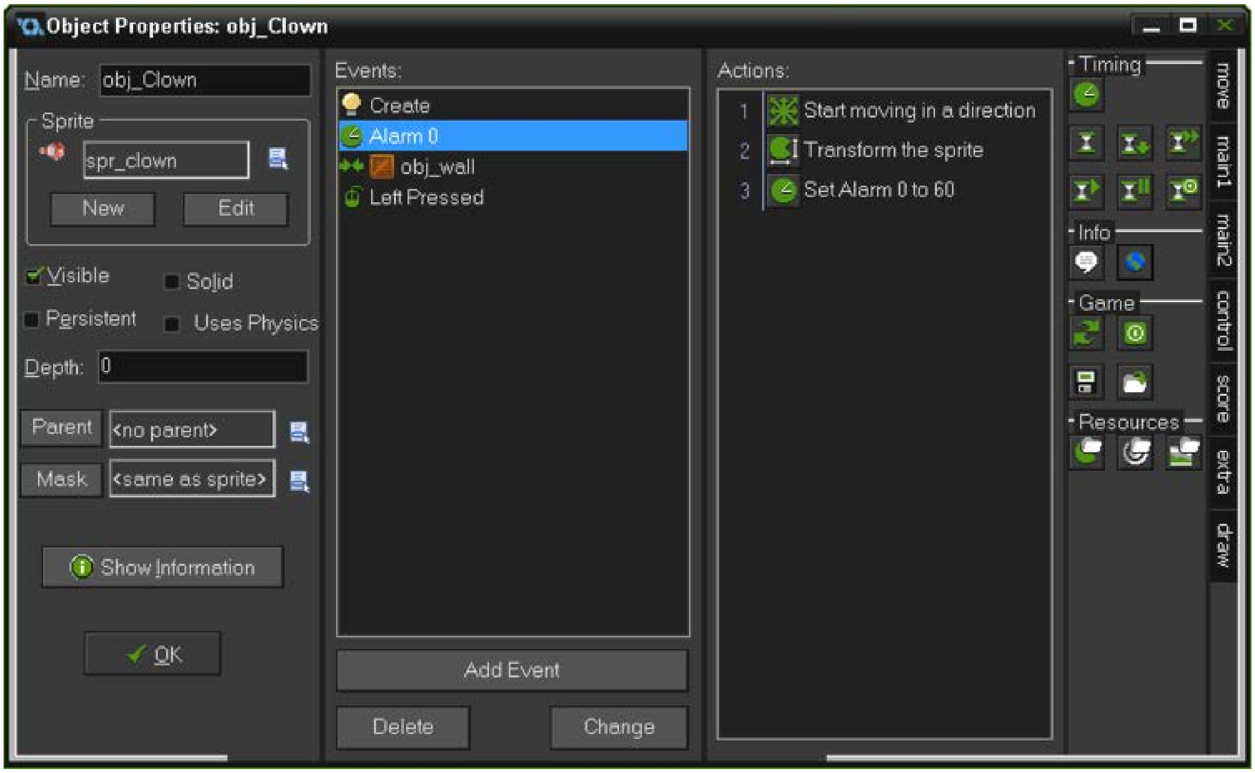
3. In the Number of Steps field, type 60.
This creates a loop in the Alarm so that it is triggered every two seconds.

4. For the In Alarm No: field, leave the default of Alarm 0.
You want to trigger Alarm 0 every 60 steps to change the angle and direction of the clown.

5. Click OK to save and close the Set Alarm properties.
The Set Alarm Action window is saved and closed.

You have now created an Alarm Event with two Actions. The Alarm Action triggers the Alarm Event every 60 steps of the game so that the clown changes direction.

Don't forget to copy and paste the Transform the Sprite Action into the Alarm 0 Event. Otherwise, the clown's angle will not update when it changes direction when the Alarm is triggered. When you have done this, the Alarm 0 Event should have three different Actions as shown in the figure.



Object Properties: obj_Clown

Name: obj_Clown

Sprite spr_clown

☑ Visible ☐ Solid

☐ Persistent ☐ Uses Physics

Depth: 0

Parent <no parent>

Mask <same as sprite>

Events:
- Create
- Alarm 0
- obj_wall
- Left Pressed

Actions:
1. Start moving in a direction
2. Transform the sprite
3. Set Alarm 0 to 60

Now is a good time to save and test your game. This time, don't click the clown, but rather wait two seconds (60 steps of the game) to see if the clown changes direction without hitting a wall.

Next

Using Controller Objects

A controller Object is an Object that does not have a Sprite assigned to it. Controller Objects are used to control elements of the game that do not affect game play. Two excellent examples for using a controller Object is to create background music and display the score.

In addition to controller Objects, there are controller variables. Controller variables, much like controller Objects, control different aspects of the game, such as when different Events and Actions are triggered.

It is a good idea to limit the number of Objects you have in each game. The more Objects you have, the slower your game might run. That's why using controller Objects are a good idea. You can use a single controller Object to control multiple Objects of a game, such as music and score displays.

Creating Background Music

The following two procedures show you how to create a Sound for use as background music and how to create a controller Object that will play the background music for Catch the Clown. The procedure after that will teach you how to add a score display into that same controller Object.

To create a Sound for playing background music, follow these steps:

1. From the main menu, click Resources > Create Sound.
The Sound properties appear.

2. In the Name field, type something such as snd_music.

3. Click Load the Sound From a File.
A navigation window appears.

4. Click the Assets folder to open it, choose background_music.mp3, and then click Open (from the bottom right of the navigation window).
The filename of the music file you chose appears under the Name field.

5. Click OK to save and close the Sound properties.
You should now see your background music listed in the Resource tree under Sounds.
After completing this procedure, you now have a Sound created for use as background music.

Assigning the Sound to a Controller Object

Now you need to assign the Sound to a controller Object. The next procedure describes how to create the controller Object with the Sound assigned to it.

1. From the main menu, click Resources > Create Object.
The Object Properties appear.

2. In the Name field, type something such as obj_control.

3. Click Add Event > Create.
The Create Event appears in the Events window.

4. Drag and drop the Play Sound Action from the Main1 tab to the Actions section.
The Play Sound properties appear.

5. Click the Sound drop-down and choose snd_music.

6. Click the Loop drop-down and choose True.
This enables the background music to play continuously throughout the game.

7. Click OK to save and close the Play Sound properties.

After you create the controller Object with the background music assigned, you might get the urge to save and run the game to hear the music. Don't be disappointed if the music doesn't play quite yet. You still have to add an Instance of this Object to the game Room. I will show you how to do that after you create the score display.

Next

Creating the Score Display

Now it is time to create a Draw Event to display the player's score as the game is played.

To create a score display, follow these steps:

1. With the controller Object from the last procedure still open, click Add Event > Draw > Draw.
The Draw Event appears in the Events section.

2. Drag and drop the Draw Score Action from the Score tab (it's the one with the green border around it), into the Actions section.
The Draw Score properties appear.

3. In the X field, type 64.
This determines the X coordinate in the Room where the score display is drawn.

4. In the Y field, type 64.
This determines the Y coordinate in the Room where the score display is drawn.

5. In the Caption field, you can leave the default of Score:, or you can type in something you prefer.

6. Click OK to save and close the Draw Score properties.

7. Click OK to save and close the Object Properties.

With the score display defined in the Draw Event of the controller Object, you are now ready to place an Instance of the controller Object into the game Room. If you’ve been following along through the whole tutorial, you should also have the background music defined in the Create Event.

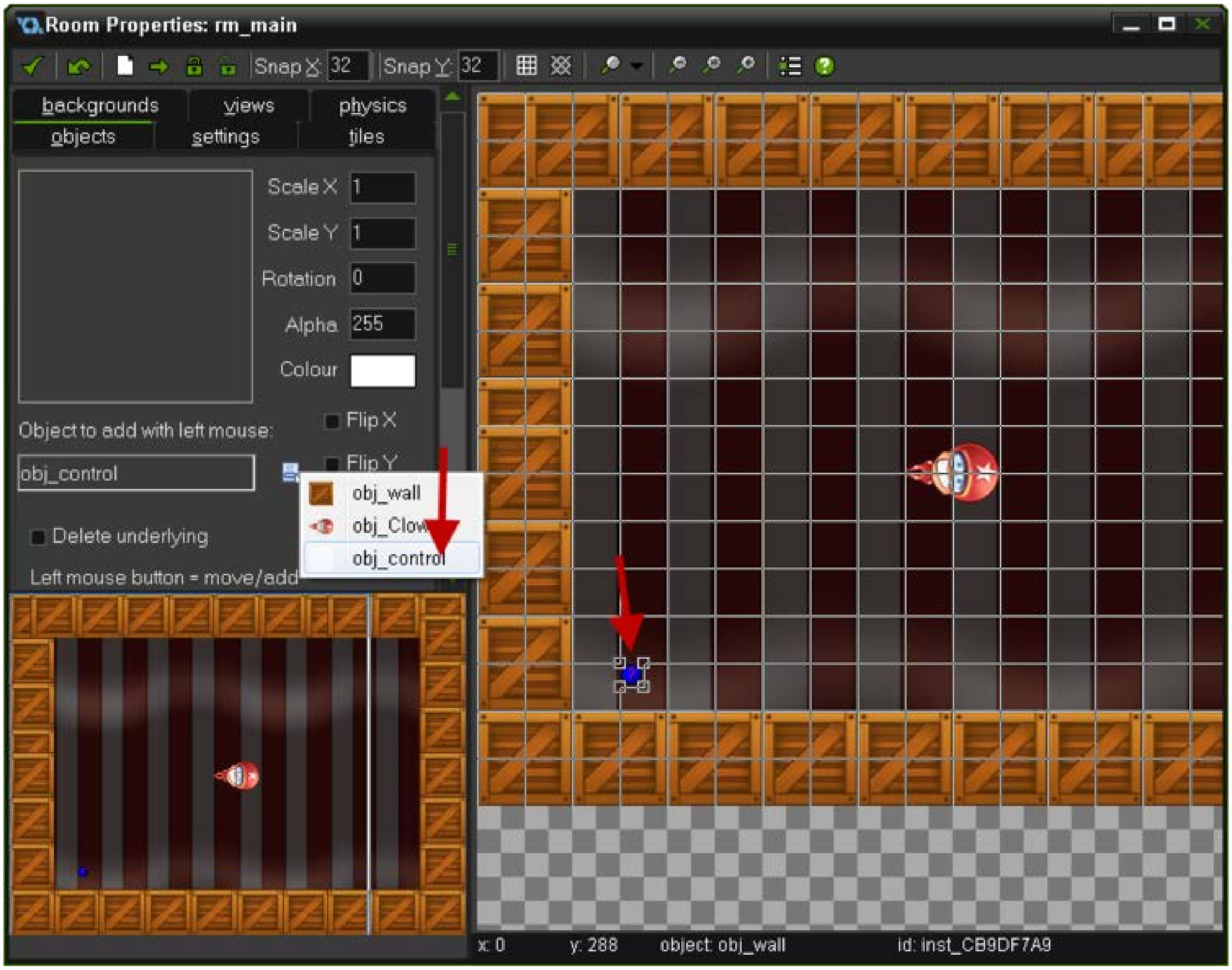
Now you'll place the controller Object in the Room.

1. Double-click rm_main from the Rooms folder in the Resource tree.
The Room Properties appear.

2. Select the Objects tab.
The Objects properties for the Room appear.

3. Click the Object drop-down and choose obj_control.
The name of the Object you chose appears in the field.

4. With obj_control selected, click anywhere in the game Room to place an Instance of the controller Object in the Room.
Objects without assigned Sprites appear as a blue circle with a question mark inside of it when placed in the Room.



You defined where the score display is drawn within the Create Event for the Draw Score Action, that's why it doesn't matter where you place the controller Object within the game Room (it won't effect where the score appears during the game).

4. Click the green check mark to save and close the Room Properties.

This is an excellent time to save and test your game of *Catch the Clown*.

To test your game, from the main menu, click Run > Run normally. GameMaker then compiles your game. When the game is compiled, it appears in a Window (because you most likely have the Window Target selected) and then you can play it.

If you're like me, you might find the score display hard to read as it's in black font on a very dark background. You might find the score easier to read by placing it over the wall. The following procedure shows you how to move the score display.

To change the location of the score display, follow these steps:

1. From the Resource tree, in the Object section, double-click the controller Object.
The Object Properties appear.

2. In the Depth field, type -100.
The Depth determines which layer GameMaker will draw the score. A value of -100 will place the score over the wall Instances instead of under them. If you don't set the Depth, then you won't see the score.

3. From the Events section, select the Draw Event.
The Draw Event Actions appear in the Actions section.

4. From the Actions section, double-click the Draw Score Action.
The Draw Score properties appear.

5. In the X and Y fields, change the value to 32.
Now the score will appear at X: 32 and Y: 32, based on the coordinates of the game Room.

6. Click OK to save and close the Draw Score properties.

Now save and run the game again. The score display should now appear higher and more to the left in the game Room. By placing the score display over the wall, it should be a little easier to read.

All of the following aspects of the game should now work correctly:

- **Music**: Provided your speakers are turned on, you should hear background music, a sound effect for when the clown bounces off the wall, and another sound effect for when you click the clown.
- **Score display**: A working score display should appear in your game, which increases by 10 points every time the player successfully catches the clown.
- **Sprites**: You should see the clown Sprite and the wall Sprites.
- **Clown actions**: the clown should change directions every 60 steps of the game and when it bounces off a wall. The clown should always face the direction it is moving and it should change direction and increase speed when the player clicks on it.

Catch the Clown contains the basic concepts of many games, such as action, scores, and increased difficulty. However, it’s still missing many things, like bad guys and lives. After all, in *Catch the Clown*, the game never ends, it just gets progressively harder.

Or next game will contain some additional items.