

GAME:IT

Bouncing Ball

Objectives:

- Create Sprites
- Create Sounds
- Create Objects
- Create Room
- Program simple game

All games need sprites (which are just pictures) that, in of themselves, do nothing. They are the elements that make up the game parts.

The sprites are later turned into objects (which can be programmed).

Backgrounds are just that—they are what you see in the background.

Backgrounds are colors, or textures, or even pictures.

The backgrounds are later used as part of the rooms.

Sounds are little files (usually .wav files) that add another dimension to games. They are used directly in the programming.

Rooms are the “playing fields” for the games. If you want to change the “playing field”, you change the room. Rooms can be changed to different sizes and detail.

Creating a New Game

Where do I start?

When starting a new program, typically you create the sprites & backgrounds first.

The objects are then created from the sprites. Objects are important because they can be programmed to do things (sprites cannot).

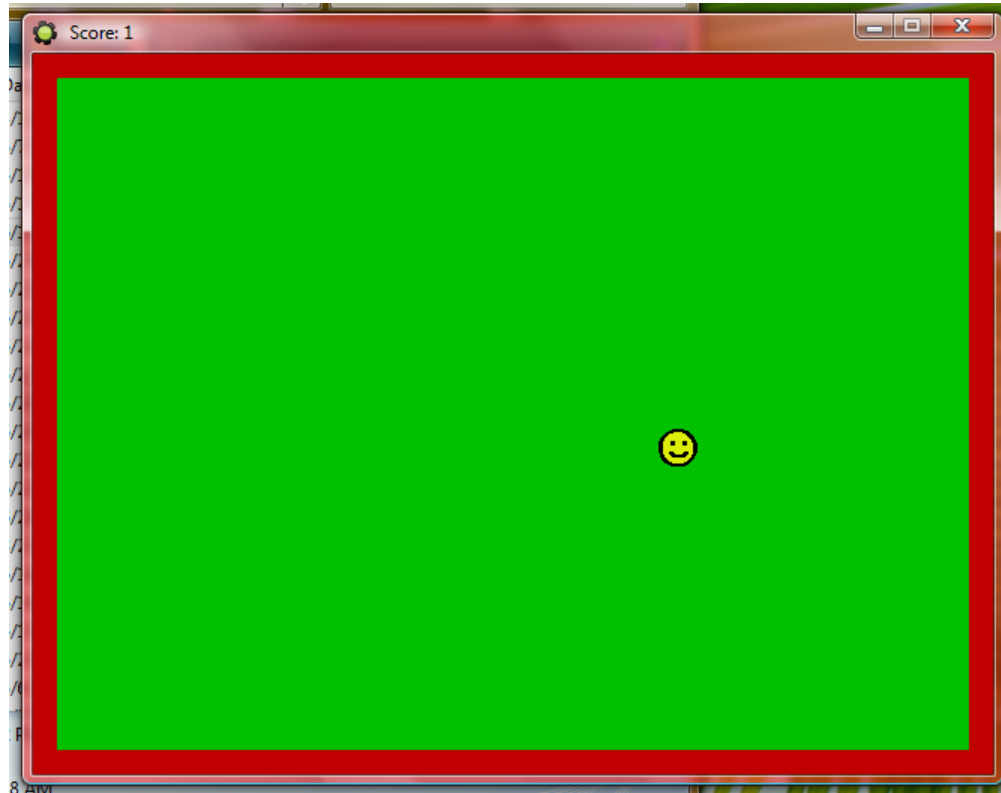
Rooms are created next because objects and backgrounds are needed to create the rooms.

When the room is set up, programming the events and actions of the objects can take place and be tested for errors.

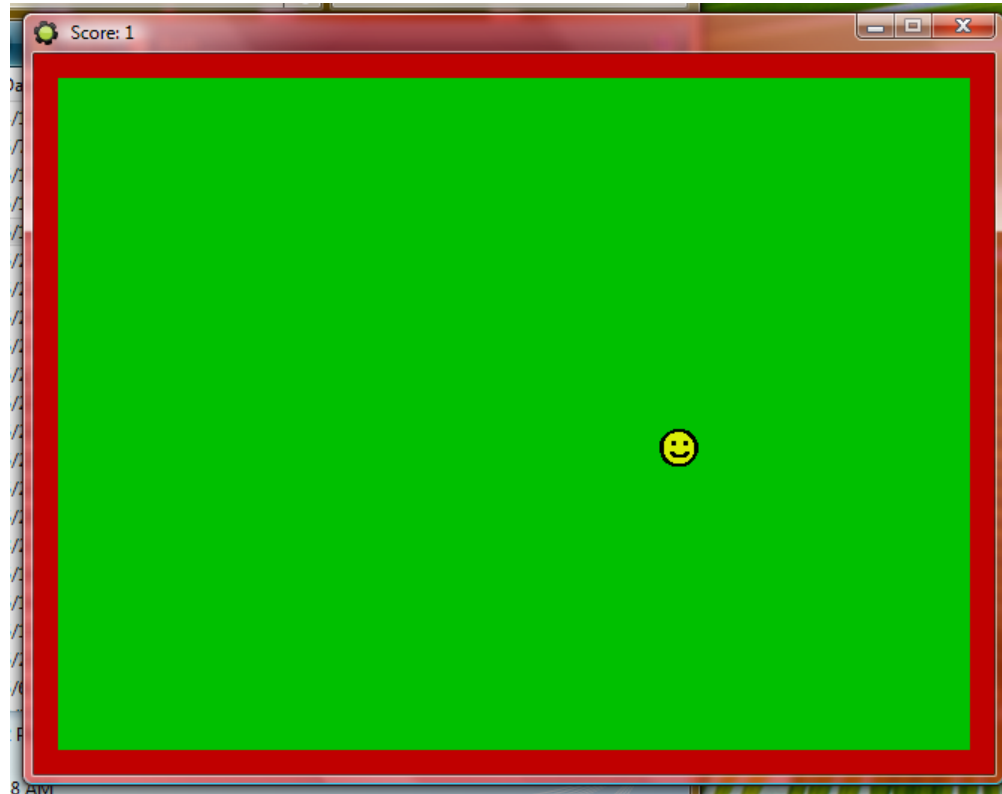
If you know what sounds you are using, they can be created before programming starts or added later to enhance the game.

Your first game will be simple to demonstrate a few of the elements of Game Maker.

The game will involve a ball bouncing around the room and making a sound when it hits the wall.



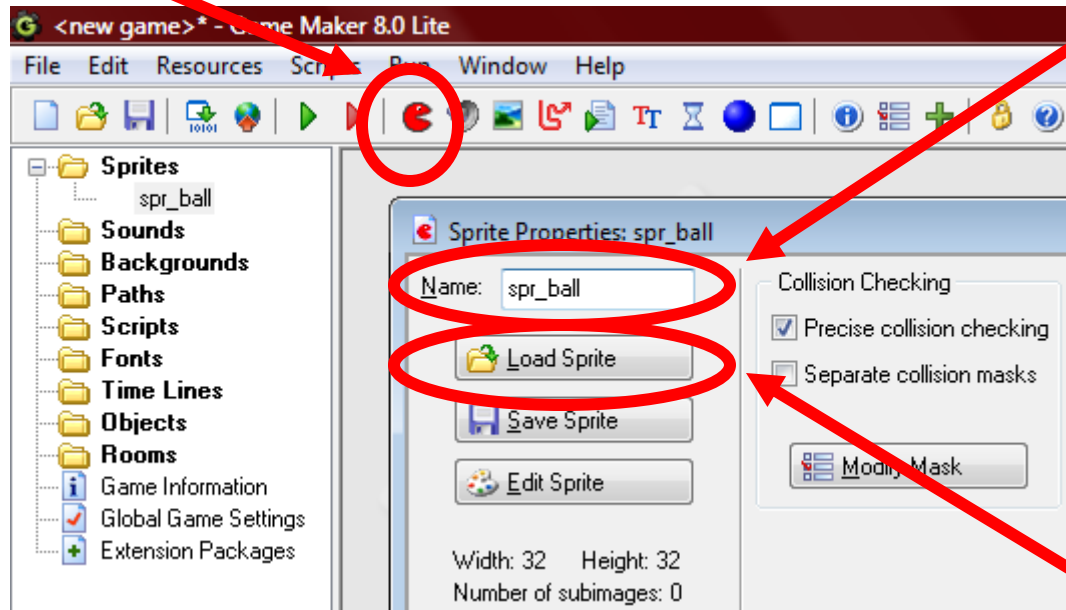
If you look at the game you should notice it has (1) a smiley face, (2) a red border, and (3) a lime green background.



To create the smiley face, you will need to create a sprite.

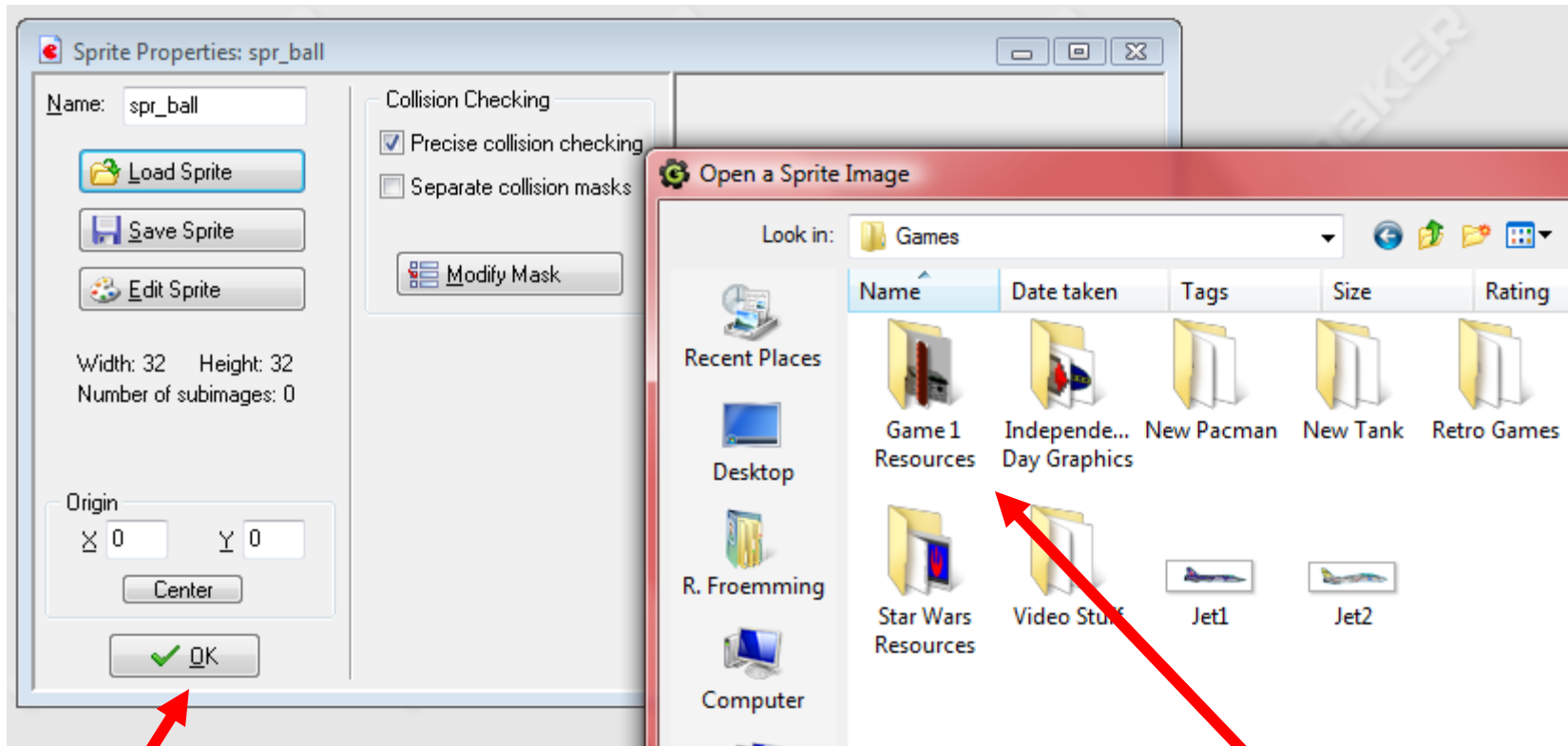
1) Select the sprite icon

2) Enter the name of the sprite—in this case—spr_ball.



3) You now have 2 options: #1-create your own sprite (we will do this later) or #2-load a pre-existing sprite.

To load a pre-existing sprite, click the Load Sprite button.



1) Find the folder that contains you Game 1 Resources and select ball.png

2) Once the ball is loaded, click on the OK button to finish.

Now do the same for wall and create **spr_wall** (this creates a 2nd sprite)



To create a sound

1) Click on the sound icon

2) Name the sound (snd_beep**)**

**3) Load the sound from the
Game 1 Resource file**

4) Click "OK"

By the way

Why is it important to carefully label sprites, sound, backgrounds, rooms, and objects?

It's just GOOD programming—it helps you to save time later in searching for mistakes or making good changes to your game.

It only takes a few seconds to type in something that could cause minutes/hours of frustration.

What are good examples?

`Spr_ball, spr_wall, snd_beep, bckgd_desert, obj_plane`

What are BAD examples?

`Spr01, sound1, background, plane (is it a sprite or object?)`

Now we need to create objects.

But I don't understand—we created sprites -- they look the same?

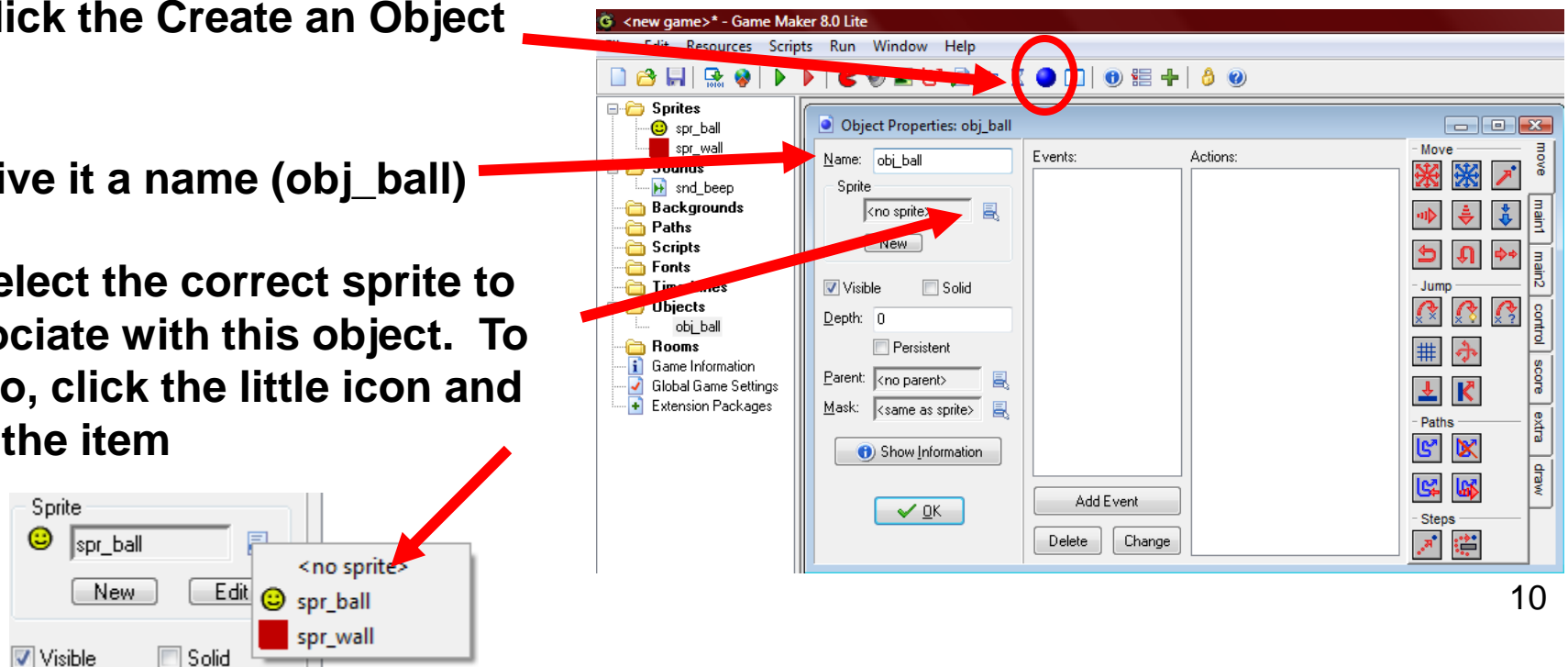
The difference is sprites are just pictures, but objects can be programmed in Game Maker.

How to create objects:

1) Click the Create an Object icon

2) Give it a name (obj_ball)

3) Select the correct sprite to associate with this object. To do so, click the little icon and find the item



The objects have additional features that need to be addressed—**visible/solid**

? What is visible?

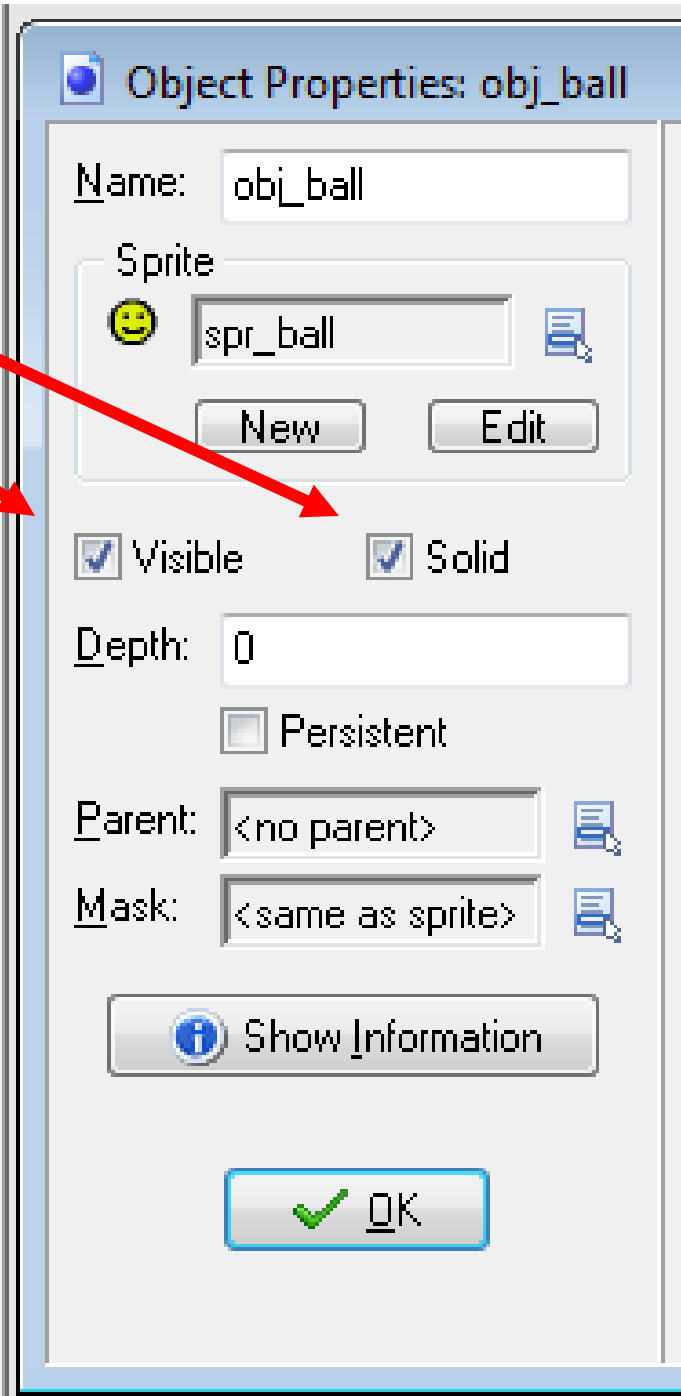
Most items in your games will be visible but sometimes you need **special pieces to be invisible (like walls and floors) to bounce off of or collide with.**

? What is solid?

It is a state of being for an object—solid you can collide with or bounce off of— **not solid you will pass right through.**

Check both for the ball. Click “OK” to save that object.

Next, create the wall object next following the same steps.

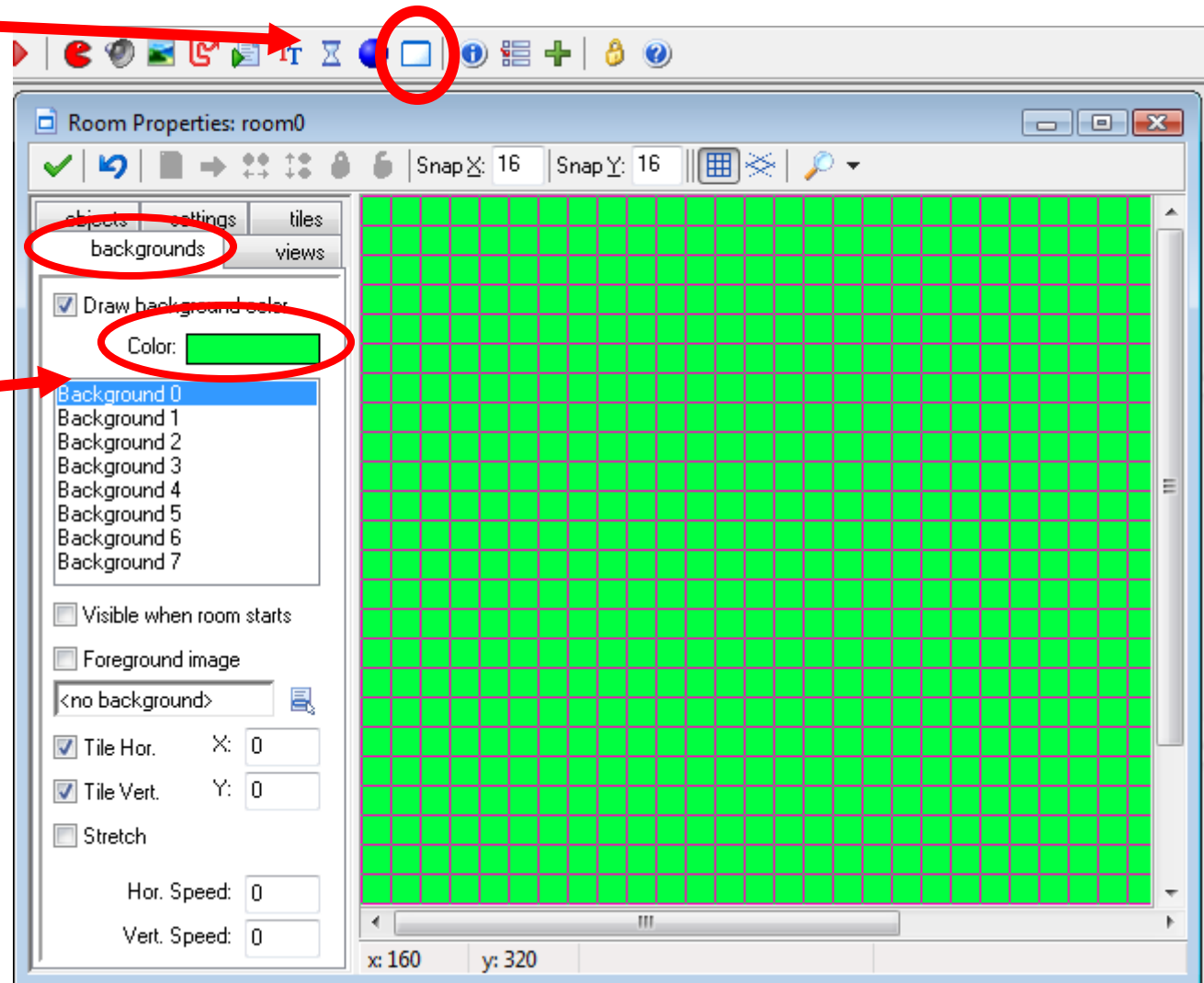


Next—we create our first “room”

1) Select the Create a room icon

2) Select background tab

3) Select a color



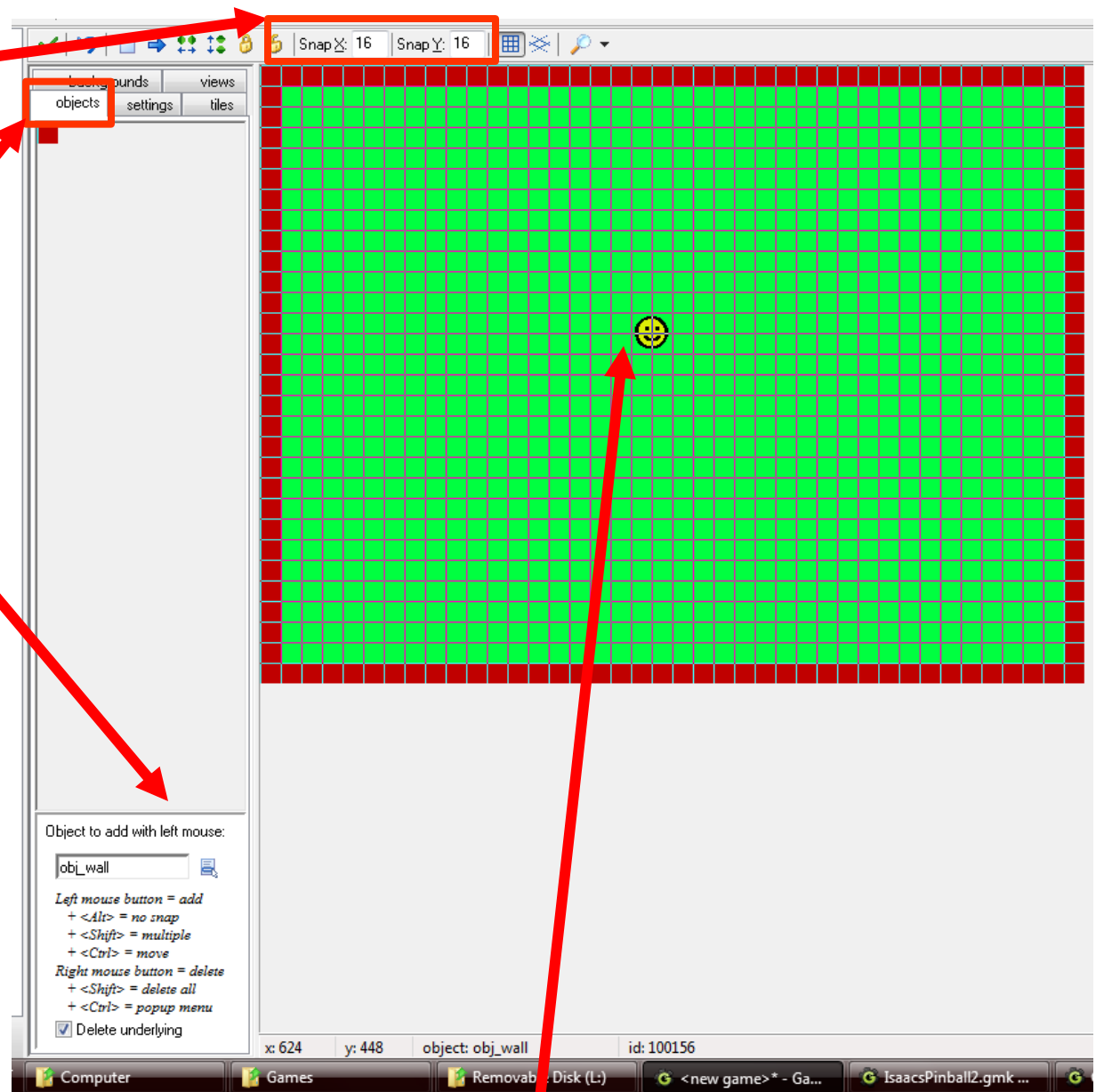
To add objects to the room:

- 1) Make sure Snap X and Y are set to 16
- 2) Click on the objects tab
- 3) Select the object you wish to add (submenu will appear). In this case, we'll add the obj_wall.
- 4) Add the obj_wall all around the outside of the room to create a border.

Left mouse adds

Right mouse removes

HINT – Click & drag with the left mouse key depressed while holding down the Shift key will add the object quickly.



4) Add in ball (anywhere in middle)

Time to Program!

The actual programming occurs in the OBJECTS we have created: **obj_ball** and **obj_wall**.

With the room set up and the objects in the room, what do we want the ball to do?

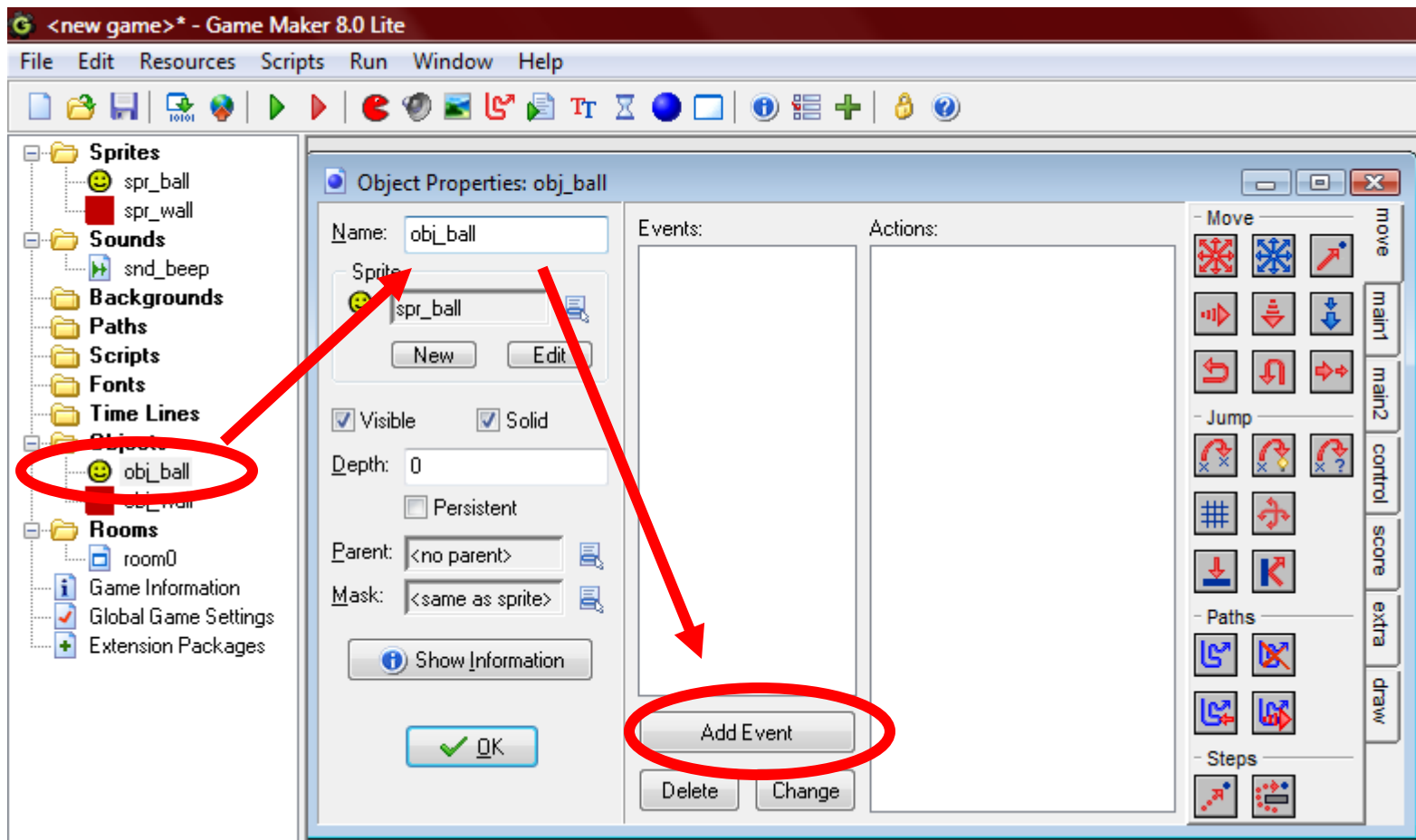
1) Bounce off the walls

2) Make a “beep” when it hits the wall.

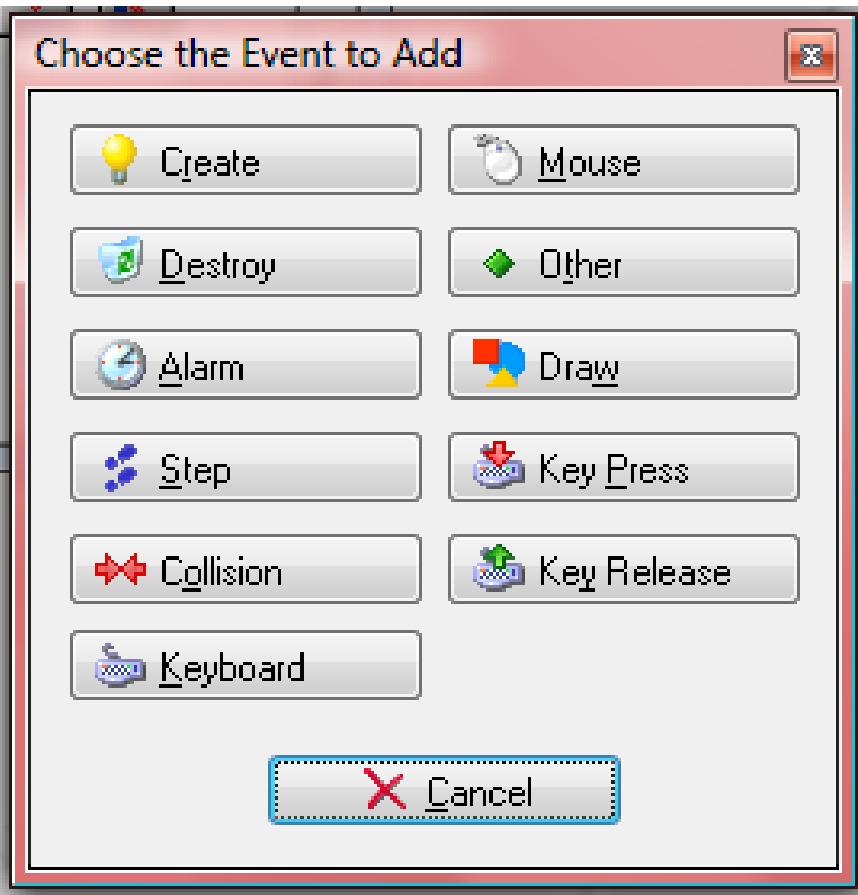
In this case, all the programming takes place in the **obj_ball**.

1) Recall you obj_ball by double clicking on it in the the side bar to open the Object Properties window for obj_ball.

2) Click on the “Add Event”



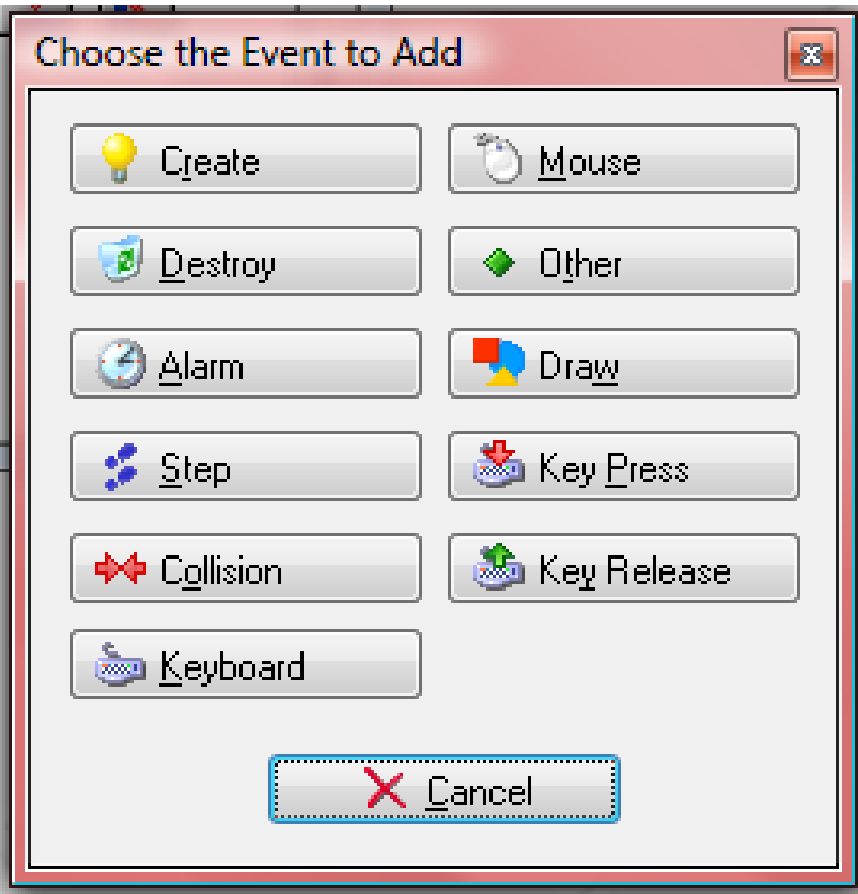
What are EVENTS?



Create Event: You use this when you want an object to be created at the beginning of the game or during (as opposed to placing the object in the room during room creation)

Destroy Event: Whenever you destroy this object, it would set in motion any actions that accompany it (seldom used).

What is ?



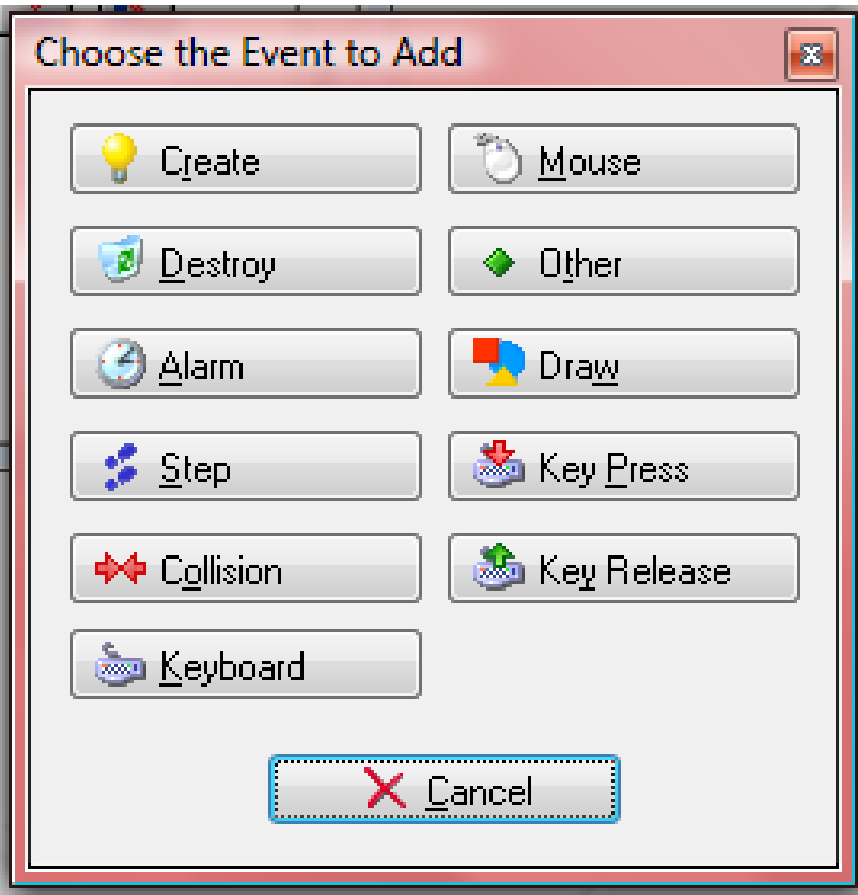
Alarm(s): Each object has 12 alarms. The alarm counts down from whatever you set and when it reaches 0, the action occurs. Example: Object changes direction every 30 steps.

FYI: Action must reset alarm.

Step Event: Happens every step of the game. Can be used to continuously check to see if “something” has happened.

Examples: When score exceeds 10 do “this”, when score exceeds 100 do “this”

What is ?



Collision Event: When 2 objects collide, you want these actions to occur.

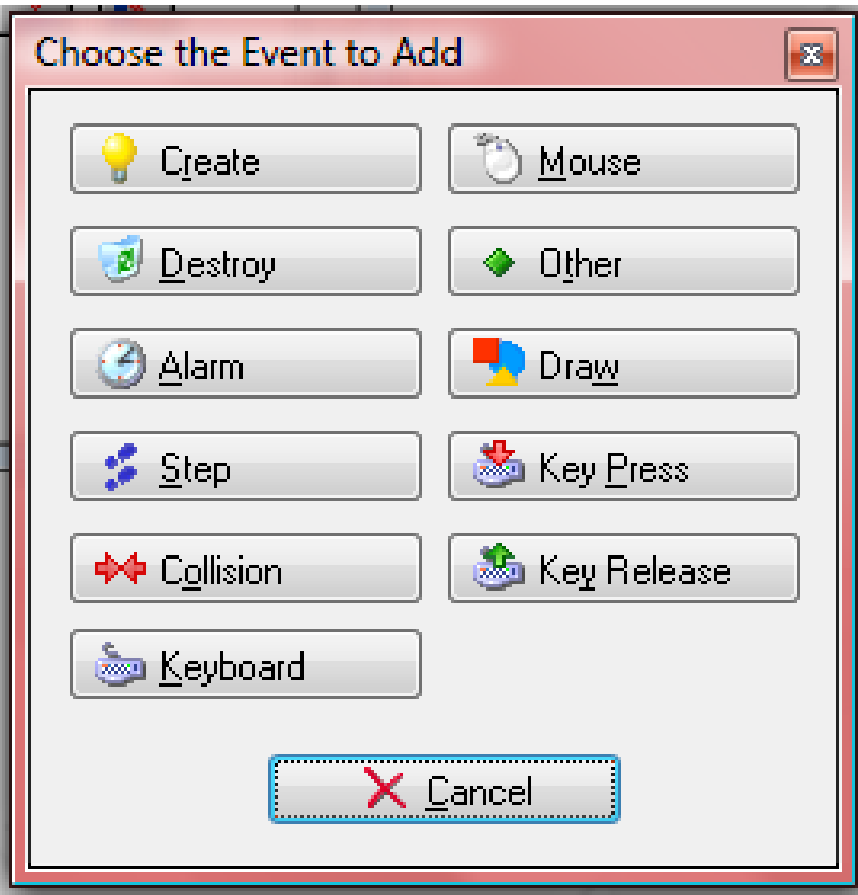
Example: When bomb collides with object—create an explosion sprite . . .

Keyboard Event: When a player presses a certain key on the keyboard, the actions will occur.

You get the action in every step as long as the key is depressed.

There is a <no key> event and an <any key> event.

What is ?



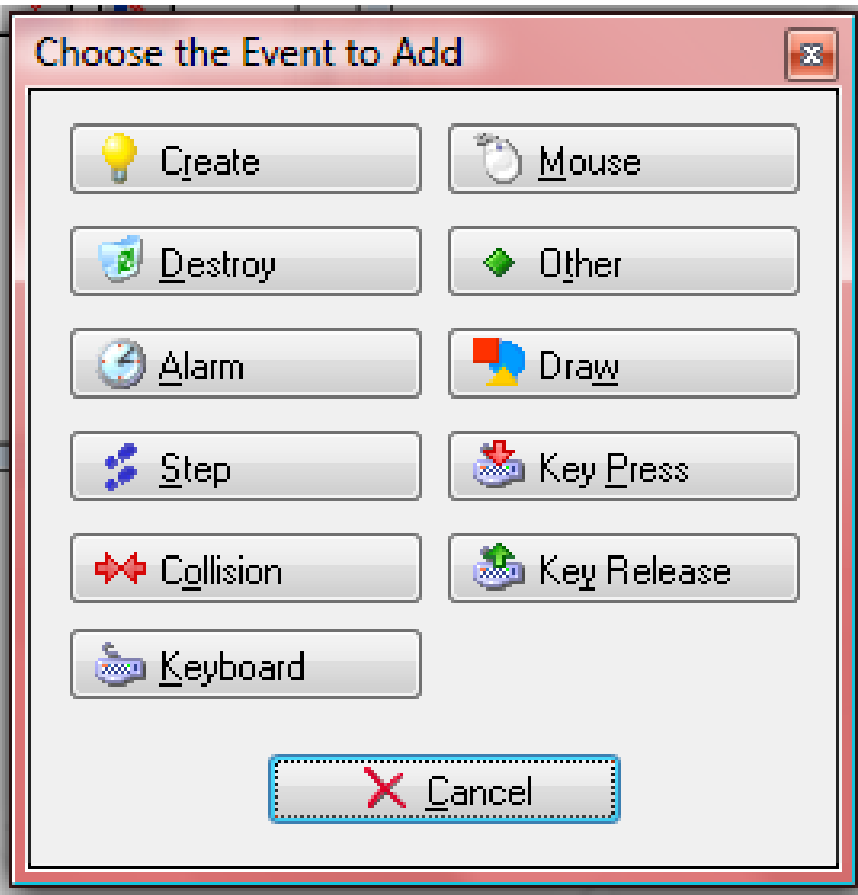
Mouse Event: This would occur when the mouse cursor touches the object.

You can select left, right or middle buttons.

Other Events: Miscellaneous collection of “special” purpose events.

Example: Outside room (when the object goes outside of the room—do “this” action.

What is ?



Draw Event: Mostly used to display score or lives remaining or other information relevant to the game.

FYI: Object must be checked visible!

Key Press: Only happens once at key press (not continuously).

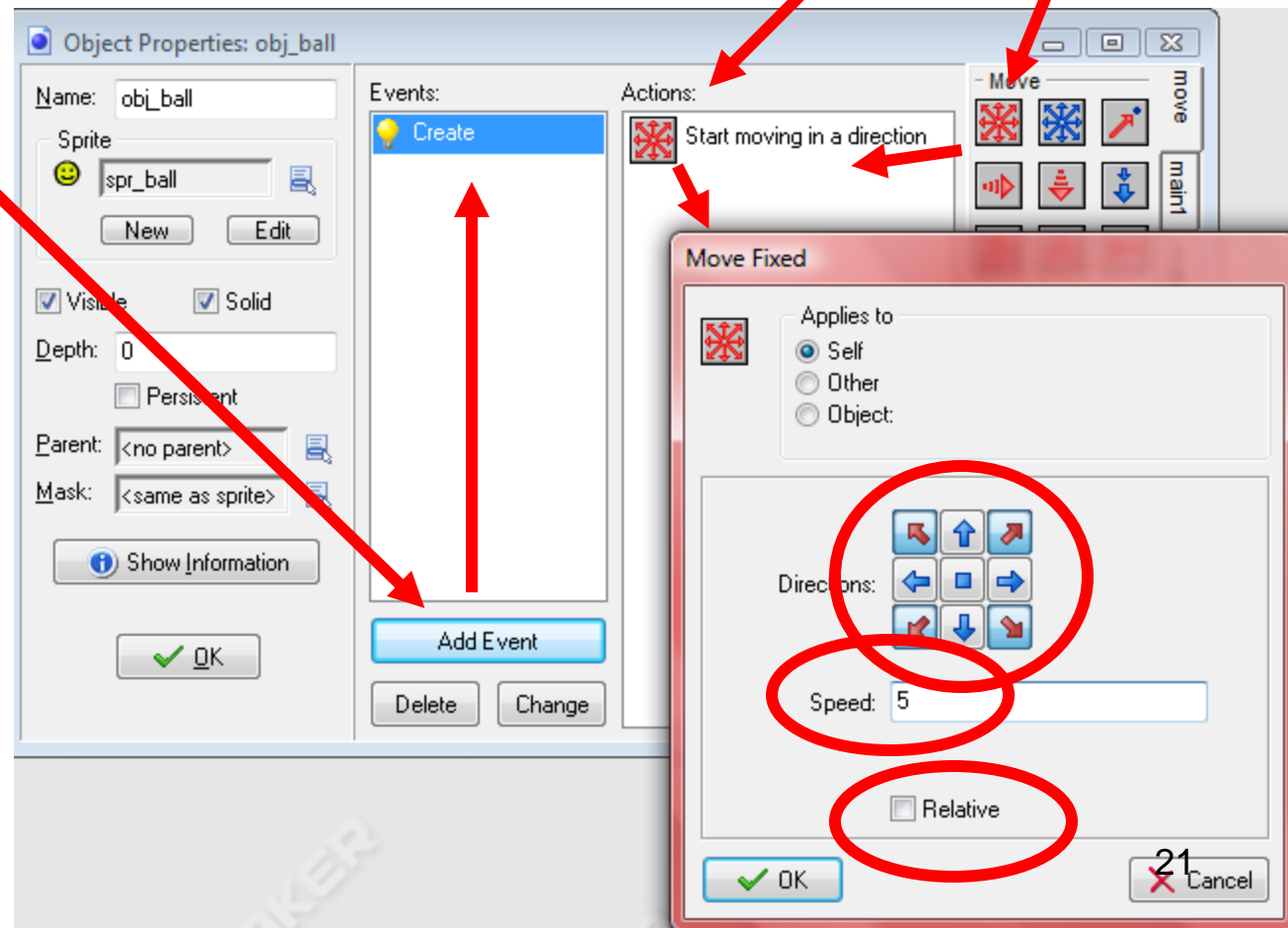
Key Release: Only happens once at key release

1) Select the “Create” event from the Add Events menu.

2) Select the “Move Fixed” from the actions menu (Move tab) and drag it into the Actions area.

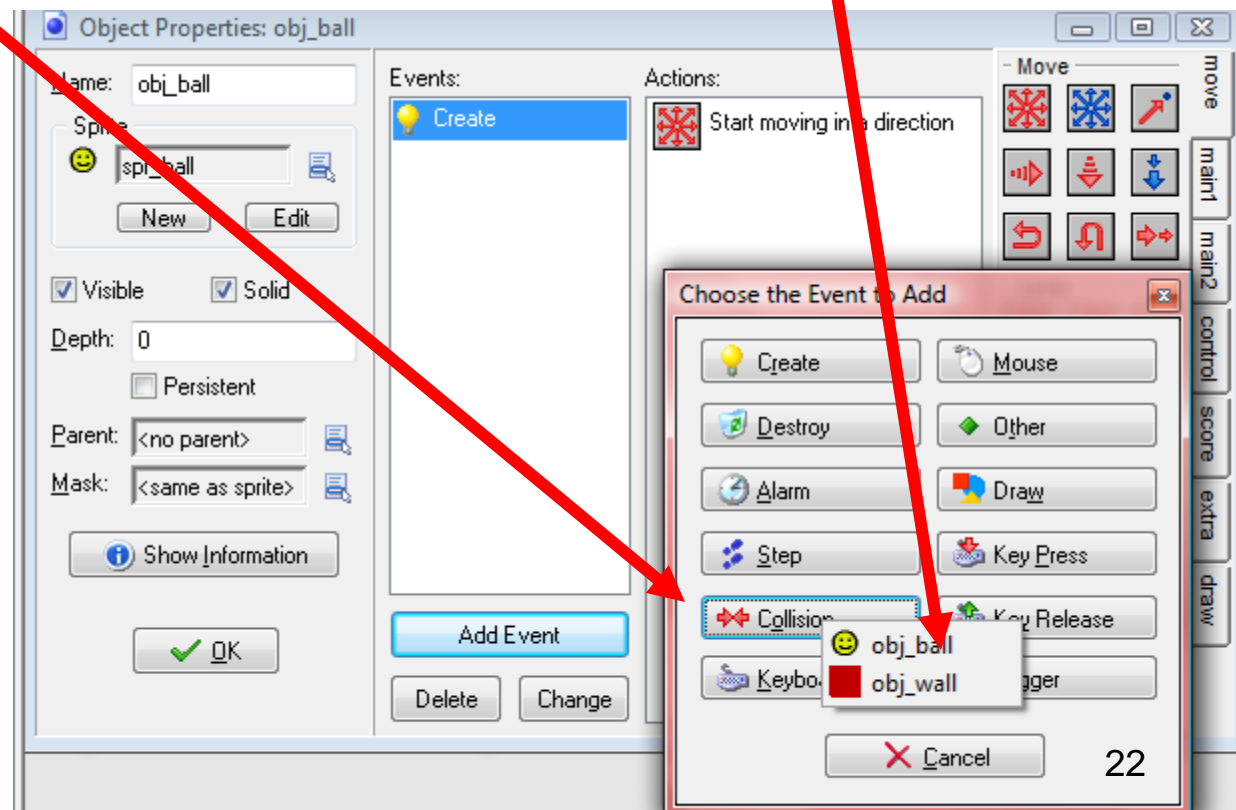
3) Double click on the icon in the actions area and a sub-menu should pop up.

4) Select the 4 corners of the directions and enter 5 as the speed. Leave the relative button unchecked at this point.



Next, select another add event—this time Collision

The collision should be with **obj_wall**.



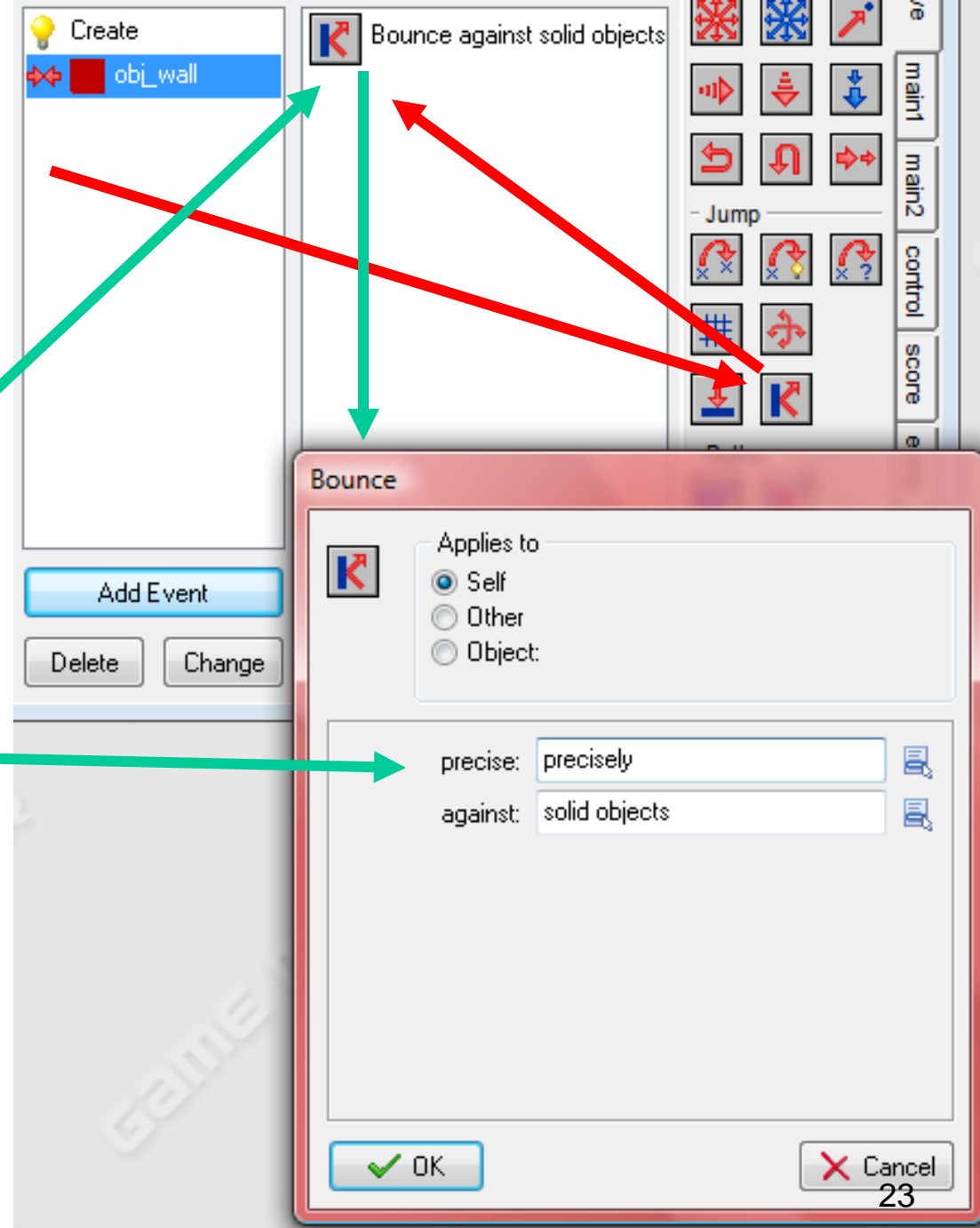
When it collides with **obj_wall** you want it to bounce off—so select the Bounce action and drag it into the action section.

It applies to Self

Precise: Precisely

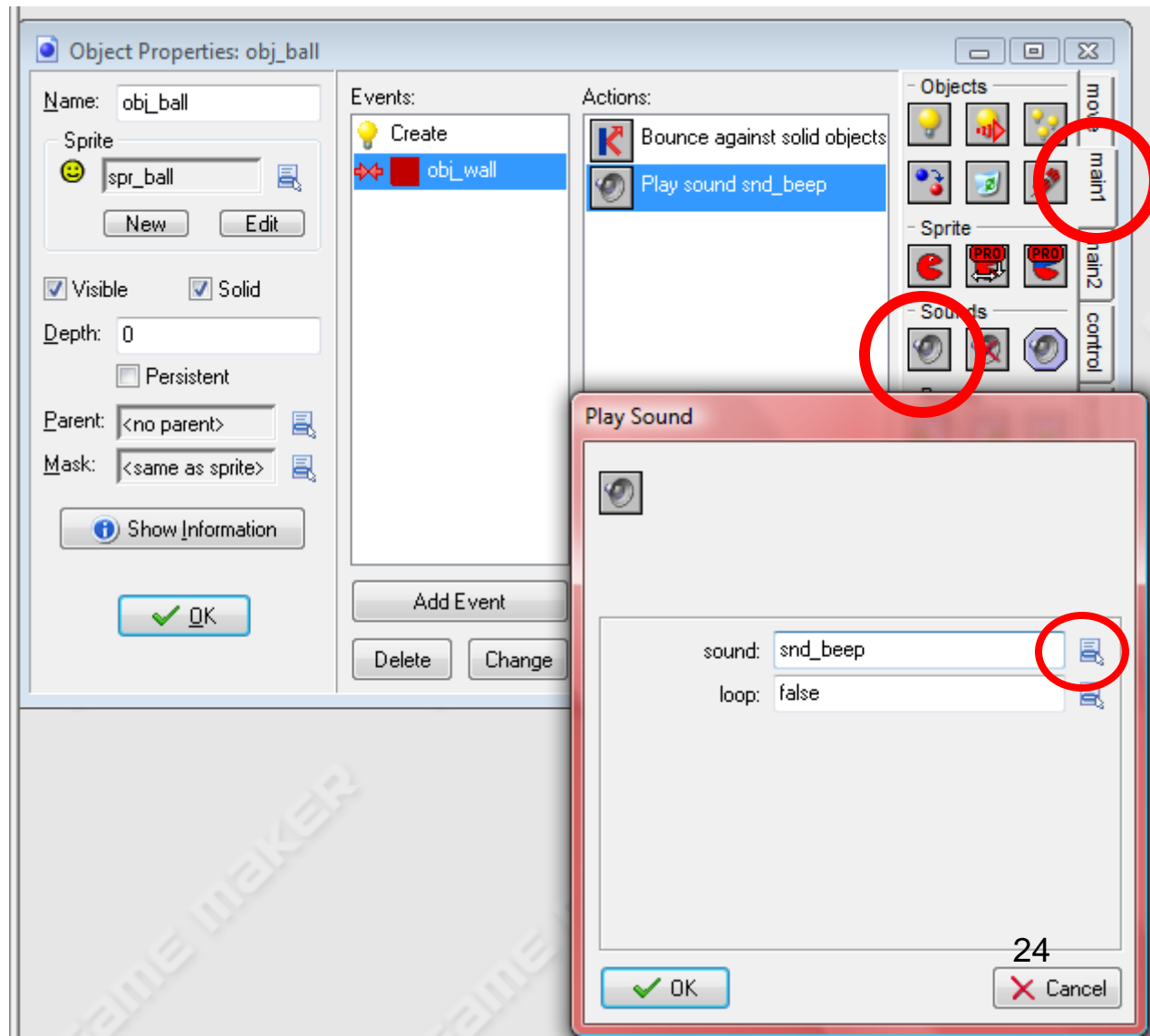
Against: solid objects

Click on “OK” to save



To get the sound to work:

Still in collision with wall—add a **Play Sound** action from the Main 1 folder. Select the sound: **snd_beep** (loop stays at false).



Save the Game— name it Game1_yourname

It's is now ready to play! Click the green arrow to test and play your game.



Assignments:

- 1) Modify the game so that the when you click your left mouse on the ball, it will disappear. Save it as Game1A_yourname

- 2) Modify the game so that the when you click your left mouse on the ball, it changes direction and increases speed (hint: relative button). Save it as Game1B_yourname

- 3) Modify the game so that the when you click your left mouse on the ball, it changes direction and increases speed and if you right click on it, the ball changes direction and slows down. Save it as Game1C_yourname