

## Tutorial: Pong

Copyright 2006, WPI

First Version created by Eric Sutman

Version 1.1

Task - Create a simple 2-player Pong from scratch

## Getting Started

We'll assume you are familiar with the classic game Pong. If not, go surf the Web and find an online/Web-based version to try out. This tutorial also assumes you have done at least one Game Maker tutorial, such as "Your first game" where you click on the clown. If not, go do that tutorial. Once ready, open up Game Maker, and save a blank file as Pong!.gm6 in your home directory to make it easier to save later on.

## Creating Sprites

Pong is not a game known for its graphical prowess (no parallax-mapping or real-time soft shadows here), hence its choice as a game to make from scratch in a few hours. However, even Pong needs *some* graphics, no matter how simple they are. With many game engines, including Game Maker, often you can draw the graphics in another tool and import the graphics into Game Maker. However, since Pong's graphics are so simple, it's just as easy to create them with Game Maker's built-in paint program.

### Ball

To create our (ironically-shaped) square ball start by right-clicking on the **Sprites** folder and select **Add Sprite**. In the **Sprite Properties** window that pops up, name the sprite "spr\_Ball". We'll also need to unclick the **Transparent** check box in the lower left of the window. We'll deal with transparencies plenty in later labs. Just above the **Transparent** check box, click **Edit Sprite**. When **Sprite Editor** window appears, select **File→New**. Input 16 in both fields and click OK. Double-click on the newly created sprite (probably called "image 0"). From here you'll enter the **Image Editor**, which is a fairly generic painting program. Use the **Fill Bucket** to paint the block white, click the **Ok** check at the left of the top toolbar, and again for the **Sprite Editor** and the **Sprite Properties** windows.

### Red versus Blue

We're going to create 2 more sprites, using the exact same process as we did with the ball, only this time, instead of making them white, we'll make them red and blue. If you really want, you could make them blue or green or mauve electric sunshine orange, but the point is they aren't black or white. Name one of them "spr\_Wall" and the other "spr\_Goal". As you may have already guessed, we'll be using these sprites for the wall and goal objects of the game. They won't be visible in-game, but will be when we place them in our level (room).

### Paddle

To create the paddle, we're again going to use a process similar to the one we used to create the ball. The only difference for this sprite is that when you put in a size for the sprite, you'll want to put in 16 for width, and 64 for height.

**Tip:** Before moving on, click the **Save** button (the disk in the top toolbar), just as a safety. In general, it is a good idea to save your work often, in case of a computer crash or in case you accidentally close Game Maker. You might even rename your previously saved games with a version number (i.e. Pongv0.1.gm6).

## Creating Objects

## Getting the Ball Rolling

Much like when we created our sprites, we start by right-clicking on the **Objects** folder and selecting **Add Object**. At the top-left of the **Object Properties** window, rename the object "obj\_Ball". Below that, select the Ball sprite by clicking on the icon to the right of the text-field and clicking on its name.

Next we will insert the behavior of the ball in the **Events** and **Actions** windows. Simply put, when anything occurs during the game, such as keys being pressed or objects colliding, the object has the ability to react to it. Thus, we program the game by coherently piecing together actions that are called by events.

Click the **Add Event**, and select **Create**. This will allow us to call actions when the object is first created. In the toolbar on the right, click and drag the first icon (red arrows in the 8 basic directions) for "Start Moving in a Direction" over to the Actions window. A new window will appear to set the parameters of the action. By directions, click the 4 diagonal buttons. The depressed buttons represent directions the ball might begin moving. We don't want to allow purely vertical or horizontal directions, because the ball would be stuck on that axis. Set the speed to 4. This will move the ball 4 pixels each "step" (there are 30 steps per second by default in Game Maker).

We will return to the ball object, but we must first create the other objects (since we aren't able to tell it how to interact with objects that don't exist yet).

## Player Paddles

Next, we'll make our players' paddles. Create a new object, and call it "obj\_Player1". For the sprite, select our Paddle sprite. Click **Add Event**, and select **Keyboard→Letters→W**. Select the icon on the right for "Jump to a Given Position" (a blue 'x' with a red arrow pointing to another 'x'). In its parameters window, input 0 for **x**, and -4 for **y**, and click the **Relative** check box at the bottom. This will move the paddle up 4 pixels at a time while the 'w' key is being held. The **Relative** check box tells Game Maker that we want the paddle to move those distances from its current location. Otherwise it will read them as coordinates. The reason we put a negative number to move up is because in Game Maker (as in many graphics applications), our origin (0,0) is on the top-left of the screen, and the y axis grows down from it. To finish this event, drag over the icon for "Bounce against solid objects" (a red arrow bouncing off a dark blue line). This will tell Game Maker that while we're holding the 'w' key to keep the paddle from going through other solid objects.

Now right-click on the W-Key Event, click **Duplicate Event** and select **Keyboard→Letters→S**. With the new event highlighted, double-click the "Jump to a Given Position" action to bring up its parameters, and change -4 to 4. Close the parameters window and Object windows by clicking **Ok**. **Tip:** In general, you should always close the object windows this way as it will save the changes you have made. If you close the windows with the 'x' in the right corner, the changes will not automatically be saved (although Game Maker will ask you).

Next, right-click on "obj\_Player1" in the Objects folder and duplicate the object. Right-click on W-Key and select **Change Event**. Change it to **Keyboard→<Up>** and then change S-Key to **Keyboard→<Down>**. Then click "Ok" and our paddles are complete!

Once again, this is a convenient place to click **Save**.

## Building the Wall

On to creating the bounding walls of the game: first create a new object, and name it "obj\_Wall", and assigning it the Wall sprite. Next, unclick the **Visible** check box under the sprite section, click **Ok**, and that's all we really need to do for our Walls.

**Goooooooooooooooooooooal**

To create the Goals, just make a new object, name it "obj\_Goa;" and uncheck **Visible**. Click **Add Event** and choose **Collision→Ball**. On the right-side of the window, click on the **Main1** tab of Actions, and drag over "Restart the current room" just above the bottom-right corner (a program window with a green arrow pointing up). This will restart our Pong game every time the ball reaches the left or right side.

### **Finishing our Ball**

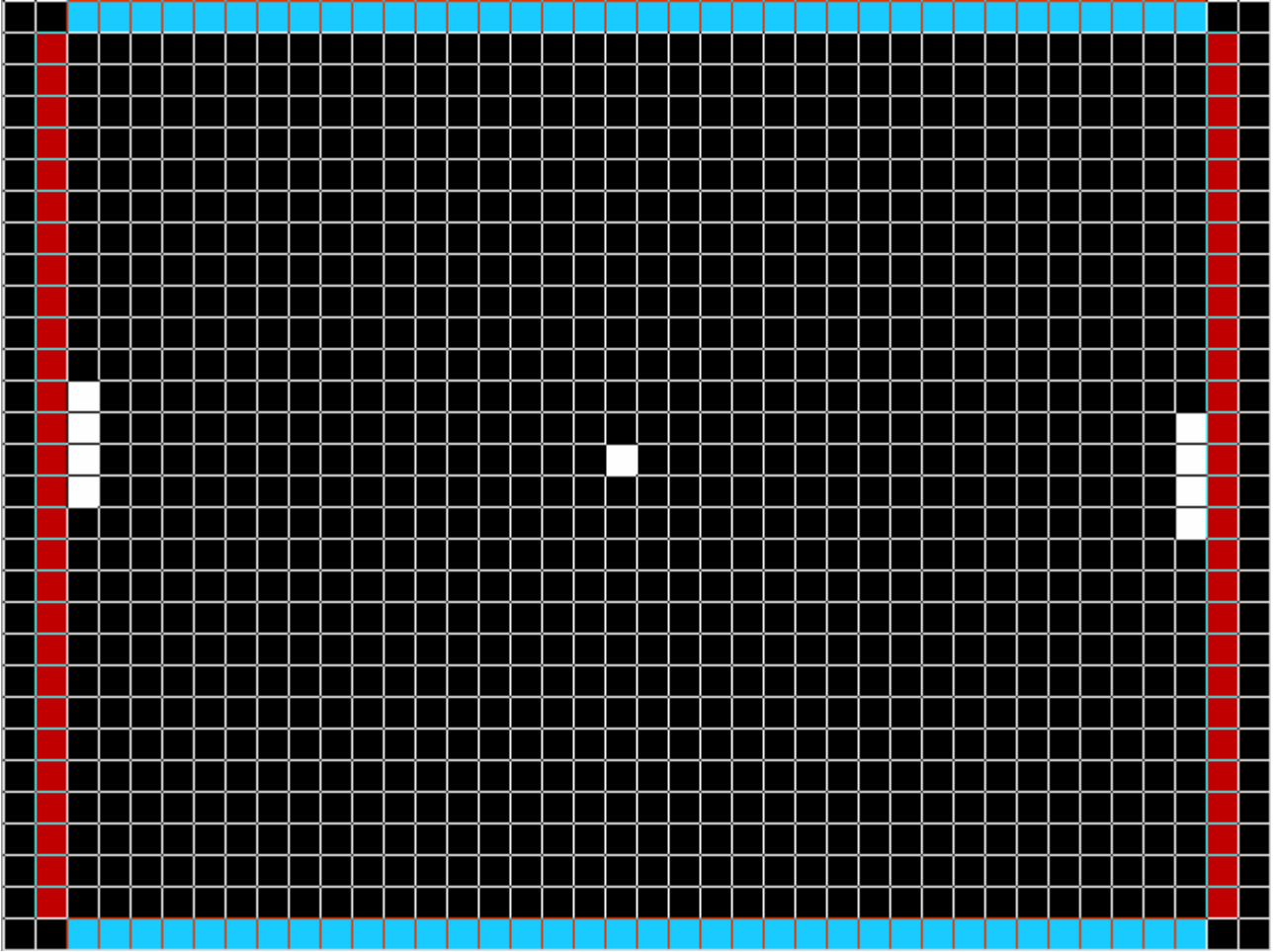
The last thing we need to do before we run our Pong game is to set-up the ball's relationship to the paddles and the walls. Double-click on the Ball object and add the event **Collision→Player1**. Now drag over the Action "Reverse horizontal direction" (a red horizontal arrow turning around), click Ok, and then pull over "Set the horizontal speed" located directly above "Reverse...". In the "Set..." parameters window, check the **Relative** box, and input "0.5:" into the field. This will increase the speed of the ball by .5 pixels per step. Duplicate the event for **Collision→Player2**, but change the .5 to -.5, because the ball is going in the opposite direction now. The cumulative effect of the increases will speed the ball up over time, so that game becomes more challenging. (There are certainly other ways to make Pong more challenging.)

Next, add the event for **Collision→Wall**. This time pull over "Reverse vertical direction", and repeat the above. That's it, we've finished our objects!

Now's another good time to **Save** your work so far.

### **Setting up the Room**

We're now ready to lay out our actual level. Right-click on the **Rooms** folder and click **Add Room**. You won't have to change any parameters of the new room before placing your objects. First click on the **Backgrounds** tab of your **Room Properties** window, and change the color box to black. Switch to the **Objects** tab, select each object and lay them out as pictures below, with our Player1 on the left and Player2 on the right, and with the Goals in red, and the Walls in blue. After the level's laid out, give it a quick **Save** and press the **Run Game** button (the green triangle) on the top toolbar to test it out!



## Extensions

There are numerous ways you might consider extending your basic Pong game. Suggestions include adding different, angled bounces depending upon where on the paddle the ball hits, adding additional rooms with obstacles, having two (or a variable number) of balls or even paddles for each player.