



Distance Formula with Alice

prepared by Bill Genereux
Assistant Professor of Computer and Digital Media Technology
Kansas State University at Salina

This worksheet assumes you have already downloaded and installed the free Alice software from Alice.org.

Using the Distance Formula in a Function

When you have completed the tutorial, do the following:

Create a new world with a moon surface.

Under Edit->Preferences, select the *Java Style in Color* view (very important!)

Click *Add Objects*, and add an Astronaut and a Lunar Lander to the scene

Click *Done* when you are done adjusting these objects.

In the Events window at the top right, click the “*create new event*” button.

Select *Let mouse move any object*. This will allow you to reposition your items.

Next, we are going to create a function that will calculate the distance between the Lander and the Astronaut using... you guessed it... the *distance formula*.

First, you need to click on the world object in the object tree. This will show the world's details below. Click on the *functions* tab to show the world's functions. Click the create new function button. Name your new function *landerDistance*.

(Just like math functions, a function in Alice is a special set of instructions that will ultimately return some answer based on some inputs. Our function *landerDistance* will use the positions of the astronaut and the lander as inputs and will output the distance between the two.)

We are going to need some variables to make this function work. Variables in computer programming are similar to variables in Algebra in that they can be any value, however with a computer program, the variables will actually store a value in memory to be used later.

Click *create new variable* to make your first variable. Select *number* as the variable type. (all of the variables in this function will be numbers) Name this variable *xAstr*. It will hold the x coordinate of the astronaut. We need variables for the astronaut's y coordinate, and the x & y coordinates of the lander. So create more variables called *yAstr*, *xLand*, and *yLand*. (The objects actually have three coordinates since they are in 3-D, but we will ignore the up/down coordinate when calculating our distances)

Alice makes it a little difficult to write math formulas, so we will create a few more variables for ease of use in our distance formula. Make variables called *xSquared* and *ySquared* to hold the $(X2-X1)^2$ and $(Y2-Y1)^2$ values of the distance formula. Finally, make one last variable to hold the distance, called *distance*.

Ok, making the variables was the easy part. Now we will put it all together. To use the variables created, we need to drag them from the variable area to the programming area just above the *return* statement.

Drag *xAstr* down first, click on *set value*, and select any numeric value. We will replace that number with the astronaut's x coordinate value. To get this value, click the *astronaut object*, and select the *functions* tab. Scroll down to the astronaut.getPosition() function. Drag that tile up to the xAstr.set(value...) tile and replace the number you previously selected. You will get an error, but click *cancel* to continue on. Select the getDistanceRight() option. We will let the "DistanceRight" value serve as our x coordinate for both objects. Repeat the same process for the other x & y variables using "DistanceRight" for the x coordinates and "DistanceForward" for the y coordinates.

You should now have values set for *xAstr*, *yAstr*, *xLand*, and *yLand*.

Next, we need to calculate the values of xSquared and ySquared, so drag each of those down and select any number. Click the **world object** and select the functions tab. The world functions include the math functions we need. Scroll down to advanced math and drag the Math.pow(a, b) function into the xSquared and ySquared tiles you just brought down, replacing the number. Choose **expressions** and *world.landerDistance.xAstr*. But this only has part of what we want. so click the **xAstr** you just selected and choose *math + xLand*. When you are finished, you should have a Math.pow function that is squaring the difference of the x coordinates. Do the same for the y coordinates.

For the *distance* variable, use the **Math.sqrt** function to get the square root of the sum of the two squares. This is really messy trying to explain on paper, but is really not that bad.