

# Tutorial: Introduction to Events

<http://www.alice.org/resources/exercise-and-project/tutorial-introduction-to-events/>



## Introduction – Lost and Found

This tutorial exercise provides step-by-step directions for guiding you through setting up some basic event listeners and event handlers. Here you will explore how to construct and use a couple different event listener types as you follow along. This will provide an overview of event listeners in Alice and teach you the basic skills to get you started through the creation of a simple interactive world that allows the user to move around and find an object. There are optional steps at the end if you are comfortable using arrays, conditionals, and functions.

You will need to access other printed materials or have access to the [alice.org](http://www.alice.org) website to view the how to content called out within these materials. These materials can be downloaded and printed for offline use from [Alice.org](http://www.alice.org)

Don't forget to save your projects frequently

## Setting Up the Scene

First you will need to set up a scene that will be interesting to navigate around and where you can have one character hide from another. For the following steps, you may wish to watch the Video or check the Quick Reference Guide associated with the How To: Scene Editor Overview

1. Start Alice

2. Select a blank slate world of your choice
3. Click on the *Setup Scene* button to go to the **Scene Editor** of Alice.
4. Save the starting camera marker position
5. Use the different camera views (the top view will be especially helpful) to add terrain and props in a way that will make the world interesting to navigate and provide somewhere to hide a character
6. Add a character to the scene hidden from the starting camera view

## Setting Up an Interactive First Person Camera

For the following steps, you may wish to watch the Videos or check the Quick Reference Guides for the How To: *Using Events Overview*

7. Go to the initializeEventListeners tab and add Event Listener Keyboard/ add object move for camera
8. \*Optionally you can create a third person camera
  - a. add a main character and situate the starting camera over their shoulder or/
  - b. add a vehicle and move the camera above and behind
  - c. Set the camera property *vehicle to* to the character or vehicle you desire
9. Save and run your program and test the ability to navigate around the world and see if you can find your hidden character

## Creating and Using Event Handlers

It seems like something should happen when you find the hiding character. Let's set up the ability to click on the character when we find them and have them react. For the following steps, you may wish to refer to the image above, or use your own ideas. You may wish to watch the Videos or check the Quick Reference Guides for the How To: *Creating Custom Procedures*.

10. Create a new scene procedural Method **found**
11. Add an animation of your choosing or do the following:
  - a. Have the lost character **say** "You found me"
12. Go to the initializeEventListeners tab and add Event Listener Mouse/**mouseClickedOnObjectListener**
13. Add the **found** scene procedure to the event listener
14. Save and run your program

## Using Add Details On an Event Listener

If you didn't click on any object other than the target object at the end of the last steps run your program and do so now. The mouse click will work on every object. This counts on the user

not clicking all over the place for the experience to work correctly. Let's extend the listener through the *add detail* conditions to only register if we select the right object.

15. Select add detail **setOfVisuals** and select custom array
16. Add the hidden character to the custom array
17. Save and run your program and attempt to click on other object before finally finding your lost character and clicking

## Add Keypress Listener

It seems odd that we can't call out to the lost character and have them respond. Let's set up a call and response animation to help the user find the character. For the following steps, you may wish to watch the Videos or check the Quick Reference Guides for the How To: *Creating Custom Procedures*.

18. Create a new scene level method **callAndResponse**
19. Add procedures to have the lost character call out to the search party however you would like or add the following:
  - a. If you created a third person character or vehicle add a **say** "Where are you?"
  - b. Add a **say** for the lost character "I'm over here"
20. Go to the initializeEventListeners tab and add Event Listener Keyboard/**addKeyPressListener**
21. Add the **callAndResponse** scene procedure to the event listener
22. Save and run your program and click any key to receive a hint as to the location of the missing character

## Add Individual Key Press Conditional

After the above steps you will be able to select any key on the keyboard to have the call and response happen but what if you wanted it to only work for the spacebar?

23. Add an **If\_** block into the **keyPressed** listener
24. Drag the **callAndResponse** procedure into the **If\_** block
25. Drag the keyPressed built in function evenGetKey??? into the **if** condition
26. In the key select choose custom key and press the spacebar and confirm
27. Save and run your program and click any key to see what happens

## Passing Event Data to the Handler Parameters

What if there are multiple lost characters? How would you determine which character was found? To do this we will need to extend the found procedure to be extendible depending on which object is clicked.

28. Add another lost character to your scene
29. Update the click event add details setOfVisual to allow for both characters by replacing the custom array to support both lost characters
30. Add a parameter to **found**
  - a. Set value type to SModel - This is required for it to support object function as any object could be clicked
  - b. Name the parameter something that makes sense such as whichCharacter
  - c. Since you have already used this method you will need to acknowledge that you will have to update any instance that it has been called by clicking the radio button
31. Update the event listener by dragging the **mouseClicked** custom function eventGetModelAtMouseLocation into the now red parameter for **found**
32. Create conditional statements in **found** to have a different outcome for each potential parameter
  - a. Add an **if\_** block with the default true
  - b. Add the relationalSThings function check from the drop down to replace the True and add whichCharacter and one of the lost characters
  - c. Drag the "You Found Me!" say procedure into the block or create something different
33. Create a second conditional block for the other character
34. Edit the callAndResponse to then also include a response from the second character.
35. Save and run the project.