

Introduction to Events

(<http://www.alice.org/resources/lessons/introduction-to-events/>)

Summary

This guide is intended to guide the facilitator through the creation of events and using the *initializeEventListeners* method / tab of the **Scene** class to create interactive animations in Alice 3. This includes a basic overview of events, event listeners, and event handlers, provide options for participants to use these events, and debriefing the experience at the end. This presentation is not an exhaustive tutorial on events, but rather a demonstration of some key principles. Hopefully students will then be able to experiment on their own.

Creating interactive animations and games can be very interesting to students, and so this lesson can be used relatively early, but it may be delayed until later, depending on the goals of the course. This lesson can be grouped with other lessons that give a more in depth look at how to create the different types of listeners if more support is needed. This lesson can also be supplemented by pointing students towards specific how to materials for different types of desired behaviors. Some of the concepts and structures needed to create different types of event listeners will require the use of Arrays, Conditionals, and Functions. You may want to cover those lessons in conjunction with or before this lesson but it is not required. This lesson should minimally be offered after a lesson in code creation also available on **alice.org**.

Learning Objectives

- What are events?
- What are event listeners?
- What are event handlers?
- Understand the relationship between *initializeEventListeners*, *myFirstMethod*, and the **Scene** class.
- Understand the four types of event listeners in Alice

Lesson Overview

- Introduction
- Lesson on Events
- Student Work Session
- OR Step-by-Step Work Session
- OR Guided Student Work Session
- Assessment
- Debrief / Students Share Work

Skills Overview

This project was developed for use with Alice 3. The following Alice 3 skills will be learned through the lesson and additional resources for them are linked to in the How to Resources section of the associated webpage for this guide and also linked to in the tutorial exercise again linked to on the webpage associated with this guide. Optional educational activities can be incorporated based on their relevance to the required steps for the project.

Alice Basics

Where to add event listeners

Creating and calling event handlers best practices

Basic Overview of Creating the Following

Scene activation listeners

Mouse Click event listener

Keyboard event listener

Collision/ Proximity event listeners

Default Event Handlers - More depth

Add keyboard move for object

Add object manipulation with mouse

Prep + Materials

Classroom Resources

Computer Access

Each participant should have his or her own computer for the duration of the project. It is also possible to allow pairs of students to work together at a shared computer.

Presentation + Lecturing

Ideally, you should be able to present the lesson ppt in front of the class. Depending on your approach, you may also want to be able to show Alice and be able to demonstrate and guide the class through the exercise. You can also print and distribute these materials if needed.

Supporting Materials

You may want students to have access to the How To... resources that accompany this lesson and are linked to on the webpage associated with this guide. This can be achieved by insuring they have access to the Alice.org website and can play the videos or by downloading and making the videos accessible in another way. You may also wish to download, print, and distribute the accompanying How To... quick reference guides that can be found on the relevant How to webpage. These materials can be downloaded in .doc format to combine several into one hand out.

Software Requirements

This lesson requires each computer to have Alice 3 installed and available.

Time

The lesson is designed to take 45m-90m, depending on:

- The inclusion of the lesson presentation
- Time spent on going over skills training
- Time spent on optional learning activities
- Complexity of exercises or projects assigned
- Time spent debriefing

Suggested Process

Introduction

Tell the students that they will be introduced to events and shown how to use keypress, mouse-click, and collision events to create interactive animations in the Alice development environment. Describe the class activity and discuss the skills they will acquire in the process.

Lesson Introduction to Events (optional)

For this lesson, you can present the lesson materials in a couple different ways due to the inclusion of a lot of Alice skills components in this lesson.

Option 1

Run completely through the supplied power point slides to give an overview of the editor, the concepts, and the skills lessons. More detailed talking points are provided below for highlighting key concepts while giving the presentation.

Option 2

Integrate application demos or the more in depth How To... videos into the presentation at the relevant topic moments. The level of detail and time spent on the skills details may be determined by how you plan to integrate the exercise component (see below), accessibility of the supporting materials to the whole class, and the skill level of your students.

Exercise Facilitation

There are several options for how to allow the participants to explore events and apply the new skills.

Option 1 (Open Student Work Session)

With groups that do well independently you may choose to have them build a scene of their own design, assign them one (or several) of the provided challenge exercises (linked on website page associated with this lesson), or allow them to select an exercise or project from the list provided. Note that the guided tutorial exercise (Introduction to Events) provides the most scaffolding for students that may need that level of support and direction. Additionally, you might want to provide them with access to the events How To... materials to support them.

Option 2 (Guided Work Session – Directions)

For more structure, you may decide you want to assign the students the tutorial exercise for this lesson. The tutorial exercise provides step-by-step directions for guiding the participants through building an interactive Alice world that points them to the correlating How To... materials at the appropriate points. This exercise also guarantees that they will explore all the different methods that can be used for each skill in the process of completing the exercise. You can point the students to the web page for the exercise or print out the associated directions. Depending on the lessons given before this lesson you may want to assign some but not all of the sections from the exercise as the later sections require an understanding of conditionals, functions, and arrays.

Option 3 (Guided Work Session – Instructor Led)

For the most structured and supported format, you may break the session up into smaller segments or Modules. Each module includes demonstrating more in depth each skill before having the participants apply them through the activity. This format also provides more break points to check in with participants. Detailed step-by-step guide provided below. You can approach this in two ways

- Play the more general How To... videos or demonstrate the skills and then allow participants to then follow the step-by-step directions for the module,
- Or demonstrate the exact steps from the exercise to the class and then have them do the step demonstrated.

A guided facilitation guide is provided below with more details. The session would follow this basic flow:

Module 1: Setting up the Scene

Step 1-6. Get everyone to open Alice, select a Scene template, and construct a scene that will meet the requirements outlined.

Module 2: Setting Up an Interactive First Person Camera

Step 7-9. Create an interactive first person camera. Ensure everyone can successfully add a basic keyboard move for object event listener.

Module 3: Creating and Using Event Handlers (End of Basic Overview)

Step 10-14. Creating a custom scene procedure and then call it using a basic event listener.. Ensure everyone has successfully created and called a procedure through a mouse click event.

Module 4: Using Add Details on Event Listeners - Optional Requires Basic Arrays

Step 15-17. Use add details on an event listener to modify their functionality. Ensure everyone has successfully added a modifier to the mouse click listener to limit the clickable objects.

Module 5: Add Keypress Listener

Step 18-22. Set up a general keypress listener. Ensure everyone has successfully created a handler and tied it to a keypress event listener.

Module 6: Add Individual Key Press Conditional - Optional requires conditionals and functions

Step 23-27. Set up a conditional block in a keypress listener to define an outcome for a specific key. Ensure everyone has modified the keypress listener to work with only one specific key.

Module 7: Passing Event Data to the Handler Parameters - Optional requires functions and parameters

Step 28-35. Use parameter on an event handler to pass eventlistener data to an event handler. Ensure everyone has successfully modified an event handler to connect to and accept an event listener function return.

Assessment (Optional)

You can use the supplied bank of assessment questions, challenges, and exercises to quiz your students on the retention of their new skills. These materials are provided in a separate document that can be downloaded from the webpage associated with this guide. A word document has been provided to allow you to customize as needed.

Class Regroup + Summary

We recommend regrouping as a class to discuss challenges and successes, and to offer feedback, both among the participants and about the curriculum itself. There are provided reflection questions found below.

Standards and Integration

Interim 2016 CSTA K-12 CS Standards

Under review

K-12 Framework Integration

Under review

Lesson Material Talking Points

Introduction to Events

What are events in programming and how do I use them in Alice.

Slide 2 Interactivity

Almost all programs that we use daily on our computers are interactive. We interact with computers and mobile devices in a variety of ways, mouse-clicks, keyboard presses, voice commands. Computers also respond to network messages, internal (error) messages, the built-in clock. These are all examples of events, things that happen that the computer, or a program on the computer, is listening for, and will respond to.

Slide 3-5 Events, Event Listeners, and Event Handlers

- An **event** is when something special happens.
 - buttons being clicked
 - the mouse moving
 - text being entered into fields
 - the program starting
- An **event listener** is a special procedure that waits for and responds to a specific event to occur while the program is running
- An **event handler** is the routine or procedure that is activated by the event listener when the event occurs

Event Listener Types

The general types of Alice event listeners and what they listen for. The following slides show the different options for each type of event listener. The following includes general descriptions, not specific details. Other lessons will explore many of these event listeners in more detail.

Slide 7 Scene Activation/ Time Events

- Clicking on the **Run...** button in Alice generates a *Scene Activated* event
- Every new project in Alice, when it is created, adds this event listener to the *initializeEventListeners* procedure of the **Scene** class
- This listener has an event handler method called *SceneActivated*, which will launch the *myFirstMethod* procedure of the Scene class, starting the program.
- This *SceneActivated* procedure can be edited so code can be added, modified, or deleted, as any procedure in Alice
- The time listener uses the internal clock of the computer to trigger handlers based on a duration of time.

Slide 8 Keyboard Events

- Listeners for different types of key presses
 - For any keypress event

- For any arrow key press event
- For any number key press event
- *addObjectMoverFor* maps the four arrow keys to a specific object in the Scene,
 - moving the object forward or backward, turning the object left or right
 - this listener will be demonstrated later in the lesson

Slide 9 Mouse Events

- Listeners for 3 types of mouse events
 - Click on an object
 - Click on the screen – demonstrated later in this presentation
- *addDefaultModelManipulation*
 - Click and drag any object in the scene while the program is running
 - this listener will be demonstrated later in the lesson

Slide 10 Position / Orientation Events

- Listeners to
 - detect if objects are colliding or about to collide with each other – optionally demonstrated later in this presentation
 - detect when an object enters into or exits from the view of the camera.
 - detect when an object is about to get between the camera and another object (occlude)
 - detect when an object changes its point of view (location and / or orientation).
- Almost all of these listeners are paired
 - To listen for an entry condition
 - To listen for an exit condition

Creating an Event

A walkthrough of creating an event listener, an event handler, calling the handler, and modifying the listener.

Slide 12 Navigate to Initialize Event Listener

- In Alice 3, all event listeners are initialized in a special procedural method of the Scene class, named appropriately, *initializeEventListeners*.
- We find this procedure by clicking on the *initializeEventListeners* tab in the Alice 3 editor window.

Slide 13 Add an Event Listener

- Selecting the listener type you want to add will cascade and bring up required fields or parameters similar to other built in methods in Alice
- Similar to other methods in Alice in some cases you may want to populate them with place holders and go back and edit or replace them once it has been created.

Slide 14 Create Event Handler

- It is possible in Alice 3 to add more complicated code directly into the event listener but we recommend you keep a separation between the listener and the handler methods

- Create a scene or object level procedural method to call when the listener is triggered
- This will make editing and navigating your code much easier in the future

Slide 15 Call Event Handler

- Similar to the myfirstmethod the method won't be called until you have the listener call the method when triggered. Navigate to the scene or object and drag your method into the listener similar to other control block.
- Note that the default behavior in the event listeners is also the do in order structure

Slide 16 Modify Event Listeners

- There are a couple things that extend listeners in unique ways to help modify them for informing or constraining the conditions of the listener or to provide more information from the event.

Slide 17 Add Details to Listeners

- Similar to our built in procedures there are optional parameters in the form of Add Details that can be added depending on the listener:
 - Set Of Visual - This will affect the object that can be affected by the event listener - object array
 - Multiple Event Policy - This will affect how your program handles if multiple events of this type are triggered at the same - combine, ignore, or que
 - Held Key Policy - What to do if the key is held down - multiple triggers or one trigger on press or release

Slide 18 Using Listener Functions

- In the header of the event listener block you will find event listener specific functions that will return data from the event itself.
- This data can be used to inform the listener or the handler

Built in Event Handlers

We have a couple event listeners that come in with fully formed event handlers already included that are made to help you easily add interactivity to an Alice world.

Slide 20 Add Object Move For

- Allows an object to be moved and turned throughout the scene using the arrow keys of the keyboard
 - This listener fires when one of the arrow keys on the keyboard is pressed
 - The listener continues to fire and the object will continue to move and / or turn as long as the key is pressed
- The arrow controls
 - UP arrow: moves the object forward in the direction it is facing
 - BACK arrow: moves the object backward from the direction it is facing
 - LEFT arrow: turns the object to its left
 - RIGHT arrow: turns the object to its left

- The user will not create handler code for this event – the Alice system has the handler built-in
- If there are multiple *addObjectMoverFor* listeners created for a project to control different objects
 - Every object that has a listener will move at the same time
 - The built in handler for this listener cannot be modified to discriminate between different objects

Slide 21 Default Model Manipulation

- Allows the user to use the mouse to drag an object around the screen.
 - This listener fires when the mouse is clicked and held on any object in the runtime window.
 - The user will not create handler code for this event – the Alice system has the handler built-in
 - Any object within the scene can be pulled around the scene as the animation is running
 - The modifier keys also work
 - move an object up and down (**shift + click + drag**)
 - turn an object left and right
 - **Control + click + drag** – Windows
 - **Option + click + drag** – Mac
- Possible uses
 - Resetting a character or prop that has gotten into a bad position
 - Creating different scenarios by altering the scene
- It is important to note that you can't modify this to limit the manipulative objects so all objects will react

Tips and Tricks

Slide 23 First Person Camera

- The shortest route to an Alice interactive world is to add object move for camera. This will create the ability for you to use the arrow keys to move around in the scene you have created.
- This can be turned into a third person camera for things like a driving game by using the *setVehicle* property to group the camera to the object and applying the object move to the object

Slides 35 Using Say to Debug

- If you are not sure if your event is being triggered a quick way to visualize the event happening is to put an object in the world or use an existing object and attach a say statement to the trigger.

Exercise Facilitation Step-by-Step

These step-by-step directions are for the guided facilitation option 3 that uses the Scene Editor Tutorial as a basis for the hands-on experience for the session. They can be followed in addition to having first gone through the whole ppt lesson.

Module 1: Setting up the Scene

Goal – Complete Steps 1-6 of Tutorial Exercise

Get everyone to open Alice, select a Scene template, and construct a scene that will meet the requirements outlined.

Media

- Play the video: Scene Editor Overview
- OR Demonstrate selecting a template and navigating to the Scene Editor

Talking Points

- For this exercise we are going to be making a lost and found interaction so think about interesting environments or character relationships to fill out the narrative
- The major goal of this section is to set up an interesting world that can be navigated at the ground level and hide a character from view of the starting camera
- You can add more detail and obstructions later so just get the basics set up
- Use props and terrain for large obstructions

Module 2: Setting Up an Interactive First Person Camera

Goal – Complete Steps 7-9 of Tutorial Exercise

Create an interactive first person camera. Ensure everyone can successfully add a basic keyboard move for object event listener.

Media

- Play the video: Events Overview
- OR demonstrate adding an event listener

Talking Points

- The object move for event listener is not just an event listener it has the full event handler already built into it.
- `setVehicle` is a very useful property for grouping objects in Alice. It is important that you vehicle the object that will follow along to the object that will be moved.

Module 3: Creating and Using Event Handlers (End of Basic Overview)

Goal - Complete Steps 10-14 of Tutorial Exercise

Creating a custom scene procedure and then call it using a basic event listener.. Ensure everyone has successfully created and called a procedure through a mouse click event.

Media

- Play the video: Creating Custom Procedures
- OR Demonstrate creating a custom scene procedure

Talking Points

- It is important to try to keep your project easy to navigate and edit. Be sure to name your event handler something that will make sense when you or someone else is looking at your code.

Module 4: Using Add Details on Event Listeners - Optional Requires Basic Arrays

Goal - Complete Steps 15-17 of Tutorial Exercise

Use add details on an event listener to modify their functionality. Ensure everyone has successfully added a modifier to the mouse click listener to limit the clickable objects.

Media

- Demonstrate modifying `setOfVisuals` of an event listener by selecting add details and adding an array of objects

Talking Points

- The default state of many of the event listeners will generally need to be modified to get the desired functionality
- An array can consist of one object

Module 5: Add Keypress Listener

Goal - Complete Steps 18-22 of Tutorial Exercise

Set up a general keypress listener. Ensure everyone has successfully created a handler and tied it to a keypress event listener.

Media

- Demonstrate adding a keypress listener

Talking Points

- Follow the same process as for the mouse click event to create a handler and attach it to the listener
- Without adding more detail the keypress listener will accept any key press
- You can select custom key if you can't find the key you want to use from the list such as space bar or if using different language keyboards.

Module 6: Add Individual Key Press Conditional - Optional requires conditionals and functions

Goal - Complete Steps 23-27 of Tutorial Exercise

Set up a conditional block in a keypress listener to define an outcome for a specific key. Ensure everyone has modified the keypress listener to work with only one specific key.

Media

- Demonstrate adding an `if_` control block and adding the `isKey ??` function to the block logic

Talking Points

- An easy way to add specific functionality for a specific key is to set it up as a conditional control block for if the key press is a specific key
- You can then add your event handler to this structure to only trigger if the specific key listed is pressed

Module 7: Passing Event Data to the Handler Parameters - Optional requires functions and parameters

Goal - Complete Steps 28-35 of Tutorial Exercise

Use a parameter on an event handler to pass event listener data to an event handler to map differentiated outcomes of the same handler based on details of the event such as which model was clicked. Ensure everyone has successfully modified an event handler to connect to and accept an event listener function return.

Media

- Play the video: Adding and Using Parameters and Using Functions Overview
- OR demonstrate setting up a parameter on an event handler and mapping an event listener function to the parameter

Talking Points

- An easy way to reuse the same event handler with the same trigger but using different details is to add a parameter to your event handler
- You can then map the specific event listener function directly to the parameter
- It is important to note that the Data Type of the data that can be returned in the function will have to be used in the parameter.
- The `getModelAtMouseLocation` returns the type `sModel` because it can return any clicked object so you will need to use that as the data type if you want to pass it through the parameter.

Reflection Questions

Still in Development Check Back