

Making Procedural Methods

(<http://www.alice.org/resources/lessons/making-procedural-methods/>)

Summary

This facilitation guide is intended to guide the instructor through the introduction of creating and using custom procedural methods, provide exercise options for participants to apply the use of procedural methods, and provide assessment and debriefing options for the end of the lesson. This should be one of the early lessons in any Alice curriculum. Creating and using user generated procedural methods is important for teaching several key concept of computer science and is a great way to help students structure their code and benefit from the power of inheritance and code reuse. It can be offered as a stand-alone lesson or can be paired with the lesson on parameters to fully explore the functional usage of custom procedures. This lesson and supporting materials cover most of the issues related to creating custom procedural methods but not all. We selected a set that we thought would be a solid basis and could be covered early in an Alice curriculum. You can modify this lesson to be longer and more advanced by adding in other lessons such as parameters, functions, data types, and variables.

Learning Objectives

- Review what is a method in Alice
- Review the different between a built in procedure and a user generated procedure
- The benefits of creating and using procedural methods
- How to add and use a procedural method
- How to share methods between projects
- Where to find additional resources for building and using procedural methods

Lesson Overview

- Introduction
- Lesson on Procedural Methods

- 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.

Using User Generated Procedural Methods

Creating a procedural method

Using a custom procedural method

Exporting and Importing custom methods

Other Alice Skills

Using the Clipboard

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 shown how to create and use custom procedures in Alice. Describe the class activity and discuss the skills they will acquire in the process. Optionally you can show them a constructed Alice scene that makes use of custom procedures.

Methods and Custom Procedural Methods Lesson (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 methods, 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 creating a custom procedural method and apply the new skills.

Option 1 (Open Student Work Session)

With groups that do well independently you may choose to have them design a scenario or story that would make use of the new skill, 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 (Making Procedural Methods) 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 code editor 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 making procedural methods that points them to the correlating How To... materials at the appropriate points. This exercise also guarantees that they will explore all the different

ways to implement a procedural method in the process of completing the exercise. You can point the students to the web page for the exercise and/or print out the associated directions.

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: Getting Started

Step 1-3. Get everyone to open Alice, open the provided world, or construct a similar scene.

Module 2: Adding an Object level procedural method

Step 4-18. Create and use the Greeting method on an object level class. Ensure everyone has successfully created, programmed, and made use of the custom procedural method at the object class level.

Module 3: Using inheritance with a procedural method

Step 19-29. Create and use the Salutation method on the Biped or similar class level. Ensure everyone has successfully created, programmed, and made use of the custom procedural method at the Biped or similar class level.

Module 4: Using Scene Level Procedures

Step 30-39. Create and use the PartyGreetings method at the scene class level. Ensure everyone has successfully created, programmed, and made use of the custom procedural method at the scene class level.

Module 5: Using Parameters

Step 40-43. If you have also integrated teaching parameters into the lesson you can continue to these steps. Create and use a parameter to customize the party guests greetings. Ensure the students have created a parameter and implemented it in their program.

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 google doc and word document have 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

Check back for updates to this

K-12 Framework Integration

Check back for updates to this

Lesson Material Talking Points

Making Procedural Methods

What is a method? Why make and use them?

Slide 3 What is a Method

- A method is a small defined (named) set of instruction within your program that defines a task to be performed.

Slide 4 Types of Methods in Alice

- Alice separates methods into two tabs or categories of methods Procedures and Functions
- A procedural method is used to give directions for doing something such as move or turn
- A functional method is used to return information such as what key was pressed or how far is one object from another

Slide 5 Built In Procedures

- Alice has already supplied a number of procedures to make basic animations possible such as moving, turning, and rolling.
- Built in procedures will also help you change environmental conditions like lighting, fog, or adding sounds.

Slide 6 User Generated Procedures

- You can create your own procedures in Alice
- The benefits of creating your own procedures are:
 - Code reuse so you don't have to code the same thing over and over again
 - Help you organize your program to make it easier to understand and for you to navigate and edit
 - Share code between projects

Slide 7 Hierarchy and Inheritance

- Where you put your custom method impacts what will be able to use the method
- Alice has an inheritance tree that is strongly influenced by the underlying joints of the classes
- If you create a procedure for a class higher up in the tree other classes under that will be able to use them

Why use methods?

What are the benefits of creating and using custom procedures?

Slide 9 Abstraction

- To use a procedure you only need to know how to use it and not how it was made. You don't need to know how the move procedure was created but you can use it.
- You can create levels of complexity making it easier for you to organize your code. More complicated segments of code can be turned into easy to read and use procedures making your program easier to navigate and read.

Slide 10 Alice Built In Procedures

- A good example of abstraction is the Alice built in procedures
- You don't need to know what was done to create them to be able to use them.
- They also simplify the program you generate a simple back and forth conversation using **say** will only be a couple lines where if all of the underlying directions such as draw bubble, add text, and more this program could be overwhelming long and harder to edit and update.

Slide 11 Reusability

- In programming if you find yourself creating the same code over and over again you are not taking advantage of the computer.
- Once you have created a procedure you can easily use it over and over again without having to create it each time which will save you lots of time.
- You can even share your procedure between projects or import other custom created procedures into your project

Slide 12 Thor Greeting His Guests

- In this example you could either program Thor waving his arm and saying hello every time someone new approaches or you could create one custom procedure called greeting that includes the animation of the arm and the say bubble hello and call it each time.

Slide 13 Debugging and Editing

- If you use one method and call it several times in your program you can easily modify it without having to find and make changes every time you separately created the animation
- The more you separate out sections of your code it also makes it a lot easier to quickly navigate to the section you want to work on or need to debug

Slide 14 Howdy is Better

- In this example we decide that Howdy is more Thor like than hello
- If we had constructed every greeting we would have to update every instance of hello
- In the construction where we made the custom procedure Greeting all we have to do is go in and update hello to howdy and every instance is now updated
- You can also see how much long the code is and how you would need to scroll for such a simple interaction when you implement every instance of the greeting

Creating New Procedures

How to create, write, and invoke your own procedures in Alice

Slide 16 Adding a Procedure

- There are a couple places you can select add procedure:
 - Within the class drop down menu
 - On the class summary page

Slide 17 Where to add your procedure?

- You can create your new procedure at different places within the Alice class hierarchy
- There are different reasons to build them at each level:
 - Scene Class - you want to be able to construct a procedure that uses multiple different objects or use environmental effects
 - Gallery Class (Biped) - You want to use the base joint structure and have the procedure used by all child classes of that gallery class
 - Object Class (Thor) - You want to use unique joints or procedure that are unique to that class such as rabbit ears

Slide 18 Name the Procedure

- Be sure to name your procedure something that will be easy for you and others to understand what the procedure will do
- Naming a procedure follows similar requirements as newly added objects:
 - Begin with a lowercase letter
 - No special characters such as ? or #
 - No spaces so for multiple word names use camelCase

Slide 19 Accessing the New Procedure

- Whenever a new procedure is added the name of that procedure now appears in multiple places:
 - The procedures tab in the methods panel
 - The procedures section of the class summary tab
 - The drop down class menu

Slide 20 What is "This"

- When editing custom procedures the class name will be replaced by "this" when looking at the procedures and methods for that class.
- Think of this as a placeholder representing that once you create this method "this" could be replaced by any class that inherits that method

Slide 21 Build Your Procedure

- The code editor are for the selected procedure tab works just like myFirstMethod so you can construct using all of the available statements and control structures.

Slide 22 Procedure is Ready To Use

- Remember that until the new procedure is used it will not actually run when the play button is selected you will need to minimally call it in myFirstMethod
- You can now drag the newly created procedure into your program just as you would with any of the other built in statements

Sharing Custom Procedures

Exporting and importing custom procedures

Slide 24 Exporting Class Files

- Custom procedures and methods are shared by saving and sharing the whole class that contains it
- To share a custom procedure you will navigate to the class summary tab that contains the custom methods and select save Class File

Slide 25 Importing Class Files

- Custom procedures and methods are imported by adding them from a saved class file that contains it
- To import a custom procedure you will navigate to the class summary tab where you want to import the procedure and select add from Class File

Slide 26 Merge Class Files

- When adding from a class file you can select which methods you want to add
- The pop up interface will allow you to check and uncheck the methods to determine what you want to import or overwrite

Tips and Tricks

Slide 28 Using the Clipboard

- The clipboard doesn't make a copy it "cuts" the selected code so if you want it to be in both places copy it first before dragging it up to the clipboard
- The clipboard can hold multiple items at a time. A number will appear on the clipboard showing how many items are being stored. They are pulled out in the opposite order of how they were added
- You can also drag nested statements as one block into the clipboard and pull them all out as one. A trick is to create a nested block to move large chunks of code in one item and then dissolve the code block once moved.

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 1-3 of Tutorial Exercise

Students open Alice and establish the required starting scene for the exercise.

Talking Points

- If you are going to create your own scene be sure to use only characters from the same gallery class hierarchy such as Bipeds or Swimmers.

Module 2: Creating a Custom MadHatter Procedure

Goal – Complete Steps 4-18 of Tutorial Exercise

Students will be able to add and make use of a custom procedure at the object class level

Media

- Play the video: Creating a Procedure
- OR demonstrate creating a procedure

Talking Points

- A custom procedure can be created at several different class levels
- For this example we will create it for the object class level
- Be sure to name your procedure something that will make it easy for you to know what it is going to do
- Programming your custom procedure allows you to use all of the same program construction functionality as myFirstMethod
- Notice that when creating a procedure for an object class you can't select other objects for the methods panels

- Note that to access and use the procedure you need to have the MadHatter object selected for the procedure to be visible

Module 3: Creating a Custom BiPed (or similar) Class Procedure

Goal - Complete Steps 29-39 of Tutorial Exercise

Students will be able to add and make use of a custom procedure at the object class level

Media

- Replay the video: Creating a Procedure
- OR Demonstrate creating a procedure pointing out the different location

Talking Points

- Notice that the scene view shows the biped image and not an object instance
- Don't be confused by "this" showing in all of the statement blocks this is just referencing this Biped and can also be thought of as a placeholder that can be replaced by all the different child classes and objects that will replace this when the procedure is actually used
- Look at the different objects that now show the ability to call this method and how the procedures panel of the methods panel of the madHatter now shows a custom biped procedure and a custom madHatter procedure.

Module 4: Creating a Custom Scene Procedure

Goal - Complete Steps 30-39 of Tutorial Exercise

Students will be able to use the clipboard to move code within their program between method tabs in the process adding and making use of a custom procedure at the scene class level.

Media

- Replay the video: Creating a Procedure and Using the Clipboard
- OR Demonstrate creating a procedure pointing out the different location and using the clipboard to move code

Talking Points

- Sometimes you don't realize till after you have created methods that you will want to reuse it over and over again

- Use the clipboard to move code from one method to another
- The clipboard allows you to add multiple individual statements or you can use a trick to put larger blocks of code inside a nested code block and move them as one large chunk
- Note that a scene level method can contain statements for different scene objects

Module 5: Using Parameters

Goal - Complete Steps 40-43 of Tutorial Exercise

Students will be able to create and use a parameter to customize instance of the custom procedure

Media

- Play the video: Adding and Using Parameters
- OR Demonstrate the different techniques for positioning a camera in the Scene

Talking Points

- Parameters can be added to user generated procedures to allow them to be modified much the same way that built in procedures have parameters that allow them to be modified
- Once you add a parameter you will need to go back and set that parameter for any procedures that have already been implemented
- The data type of the parameter will impact the way that you will have to input the parameter when using the procedure.