An Introduction to Alice 3.1

From the Alice v. 2.4 By Jenna Hayes
Modified for Alice v. 3.1 by Lori Fuller
under the direction of Professor Susan Rodger
Duke University, June 2014

SAVE OFTEN!!!
Hello! I’m Alice, and I’m going to teach you how to use the Alice 3.1 program. With Alice, you can make your own animations, using tons of different characters.
Starting Off

• Our first step is to choose a background.
• When you open Alice 3, a box will pop up that has different choices of background. It looks like the box to the right.
• Select the **MOON** background, because our world will be in space.
• Click on **MOON** and then click **OK**.
After you click **OK**, your screen will look like this:

![Image of a programming interface with text boxes and options]
Panel Sections for Alice 3.1
Saving your world

• Before we do anything else, let’s save our world. You should also always do this before you close out of Alice.

• Click on File at the top left-hand corner of your screen, and then click on Save As.
Saving your world

• In the box that pops up, name your world **spaceWorld**, and save it in a place that you will be able to find again, such as in a MyProjects folder on your Desktop.
Adding objects to your world

• Now, we will add some objects to the world.
• In the **Scene View**, click on the **Setup Scene** button.
Adding objects to your world

A new screen will appear, on which there is a selection of objects below the MOON screen that you can add into your world. This is called the **Gallery**.
Adding objects to your world

• In the list provided, select the Transport classes

• Select the Automobile classes.

• Click on the new Humvee.
Adding objects to your world

• On the box that pops up, you may change the name. Be sure to use unique names if you are going to add more than one instance of the same object.
• Click OK to add the instance to the MOON scene.
• The humvee will appear in the center of the MOON scene. Yes, humvees can drive on the MOON.
Adding objects to your world

The humvee takes up most of your screen, but we will re-size it.
Adding objects to your world

- The humvee takes up most of your screen, but we will re-size it.
- Select the RESIZE button on the right-hand side of the screen.
- Move the cursor to the humvee and click and drag to make it larger or smaller. In this case, make it smaller, we need more space on the MOON.
Adding objects to your world

- The humvee is not facing in the direction we need it to face.
- Select the **ROTATION** button on the right-hand side of the screen.
- Move the cursor to the humvee and click and drag to turn the vehicle from the bottom circle.
Adding objects to your world

• Next, we are going to add an astronaut.
• Click on the All Classes button on the Gallery section.
• Select Biped classes.
Adding objects to your world

• In the window that appears, you will need to build your astronaut. You may even make it look like you...
• Adjust life stage, gender, skin color, outfit should be astronaut suit, hair and face.
• One issue, which has been mentioned to the creators of Alice 3, is that there is no space helmet for the astronaut outfit. Let’s pretend you can now breath on the MOON.
• Click OK button when done.
Adding objects to your world

This window will appear, and you may rename the instance. Remember, if you are using more than one (maybe you’ve been cloned), name each with a unique name. Click OK when done.

SAVE OFTEN!!!
Adding objects to your world

You will need to resize the astronaut to fit the window and have size continuity with the humvee. Just as you did with the humvee, select the RESIZE button and click and drag to the size desired. And then, if necessary, select the ROTATION button and rotate the astronaut as needed. Use the DEFAULT button to move the object around the screen. *Should look similar to the image below.*

![Image of astronaut and humvee](image-url)
Now we have added two objects to our world. The next step is to position them!
Review on positioning and adjusting instances in the scene

• Within the scene, select the object you want to position or adjust.
• The DEFAULT handle style allows you to move the instance on the screen.
• The ROTATION handle style allows you to rotate the instance on the given axis.
• The TRANSLATION handle style allows movement up or down and forward or backward in the scene.
• The RESIZE handle style allows you to make the instance larger or smaller.

SAVE OFTEN!!!
Review on positioning and adjusting instances in the scene

• You may also Position the instance by setting the X, Y, and Z points.
• Reminder, on the Mac, all three points needed to be set before click on Enter/Return button.
• You may also adjust the size of the instance by adjusting the Size: Width, Height, and Depth.
• Again, on the Mac, I had to remain in the size field selected and press the Enter/Return key.
The Undo button is your friend!

• What if you make a mistake, like accidentally clicking on something and moving it or resizing it?
• You can click on the **Undo** button above the Code Editor to undo the last thing you did.
• Use this button whenever you mess up, or want to get rid of something you just did. I think this is redundant.
We have finished positioning our characters, so now we can move on and start to move our camera around!
The Camera Markers

• A **Camera marker** is a bookmark location to save the location of your camera view. This way, if you move your camera around, you can always get back to a certain position by moving to the **Camera marker** location.

• Look to the right side of your screen, and find the **Camera Markers** arrow. Click the arrow to reveal the **Add Camera Marker** button.
The Camera Markers

• Alice has only one camera in a scene. The camera is moved around and repositioned for close-up views and special effects. Because moving the camera is a common action when setting up a scene in the Scene editor, Alice provides a way to mark a camera position so as to create known camera positions in the scene.

• Each scene and camera angle is setup and rehearsed until you are happy with the arrangements. The camera positions (location and orientation angle) are marked before any actual filming begins. Below is an example of the camera position as set for the initial scene.
The Camera Markers

- It is **highly recommended** marking the starting location of the camera before moving the camera around in the scene. The camera can then be moved freely around the scene and can always be returned to its original position, using the marker.

- To create a starting location camera marker click on the Add Camera Marker button. A dialog box will pop up. Enter a meaningful name for the marker, for example `startView`.

- When a name is entered, press **OK** and Alice will automatically create a camera marker object at the current location of the camera. The marker remembers not only the location but also the camera’s orientation (the direction and angle at which it is pointed). This information is commonly known as the camera’s viewpoint. Also, color coded.
Moving the Camera

• Let’s try moving the camera to get a close up of the astronaut.

• Under your viewing screen, you should see a bunch of blue arrows.

• If you click on the arrows and drag your mouse in different directions, the camera will move in different ways. *Use small adjustments.*

  *I found that the trackpad on the Mac works also...*
The Camera Marker

• Move your camera until it is close to the astronaut’s face. Use the **Undo** button if you make any mistakes or don’t like the result. Not too close...

• Let’s drop another Camera marker at the camera’s new position, using the same steps as before, and label it **Astronaut HeadShot**.
The Camera Marker

• Add another Camera Marker for the humvee.
• Move the cameras so you can only see the humvee in the camera.
• Add another Camera Marker and name it `humveeView`.
The Camera Markers

• To move from one camera view to the next using the Camera Marker Window...
• As you can see they are color coded.
• Click on the view you want to go to and click on the left camera button at the top left of the window. Then, click on the color of the camera view. Practice a few times and then return to the startView.
Camera Marker Conclusion

• Now, right click on camera in the object list and set its view back to startView. Yes, that was mentioned on the previous slide.
• As the camera moves back into the start position, you can see the other camera placements. And they are color coded as they are in the Camera Marker window.
Now that we are done setting up our cameras, we can start to animate the characters in the world!
Methods/Procedures

• Click on the **Edit Code** button.
• It should bring you to the main windows.

SAVE OFTEN!!!
Methods/Procedures

- **The Code Editor** panel is where you will add the coding to animate the objects.
- **The Methods panel** is where you will find the pre-existing Procedures and Functions for the objects.
- And the **Scene View** is where you will see the animation evolve.

SAVE OFTEN!!!
Selecting the object

• The **method editor** is where you can get the pre-existing procedures and functions to make your characters do things.

• Your characters already know how to do certain things.

• To select an object to animate, either click on the instance in the **Scene View**, or use the drop down menu next to ‘this’ (in this case) to select it.
Adding Methods/Procedures

• To tell your astronaut to do something, click on one of these methods, hold down your mouse button, and drag and drop the procedure into your Coding Editor.

• Click and drag the astronaut (this.adultPerson) ‘say’ procedure to the drop statement here position. Select ‘hello’ as the default.

• Add a move and set it for up 1 meter.

• Add a turn right 1.

• Add a move down 1 meter.

• Press the Run button.

See the sample programming statements in the Code Editor

SAVE OFTEN!!!
Adding an astronautWave Procedure

• Click on the Class Tab.
• From the list, select the AdultPerson, which in this case is the astronaut.
• Click on the Add AdultPerson Procedure.
Adding an astronautWave Procedure

• In the window, name this astronautWave.
• Click OK.

• This should bring you to the window now named astronautWave. See tabs at the top of the Code Editor.
Adding an astronautWave Procedure code

1. From the Procedure tab under the Methods panel, click and drag ‘this turn...’ to the Code Editor
   • From the ‘this’ click and locate the getRightShoulder, and
   • set the direction to Left at .4 meters.
   • Set the duration to 1 second and
   • set the animation style to Begin and End Gently.

2. From the Procedure tab under the Methods panel, click and drag ‘this turn...’ to the Code Editor
   • From the ‘this’ click and locate the getRightWrist, and
   • set the direction to Forward to .25 meters.
   • Set the duration to 1 second and
   • set the animation style to Begin and End Gently.
Adding an astronautWave Procedure code

3. From the Procedure tab under the Methods panel, click and drag ‘this turn...’ to the Code Editor
   • From the ‘this’ click and locate the getRightWrist, and
   • set the direction to Backward at .25 meters.
   • Set the duration to 1 second and
   • set the animation style to Begin and End Gently.

4. From the Procedure tab under the Methods panel, click and drag ‘this turn...’ to the Code Editor
   • From the ‘this’ click and locate the getRightShoulder, and
   • set the direction to Right at .4 meters.
   • Set the duration to 1 second and
   • set the animation style to Begin and End Gently.

SAVE OFTEN!!!
Adding an astronautWave Procedure code

You should now have 4 programming statements in the Code Editor.

Select the myFirstMethod tab at the top of the Code Editor panel.
Adding an astronautWave Procedure

- Above the Methods panel, you should see the panel with ‘this’ in it.

- Click on the down arrow and select **this.adultperson** from the list.

- You should now see the **astronautWave** method now listed under the Procedure tab.
Adding an astronautWave Procedure

• Click and drag the astronautWave procedure to the Code Editor, and place it under the existing code.

![Image of Code Editor with astronautWave procedure added]

• It should look like this... press the run button to try it out.
Adding Keyboard Controls

• Now we want to add keyboard controls to control the action in the animation.
• The first keyboard to add is for when the SPACE bar is pressed, the astronaut will perform the astronautWave procedure.
• To start, click on the initializeEventListeners tab at the top of the Code Editor panel.
• Click on the Add Event Listener tab.
Adding Keyboard Controls

• Select **Keyboard** and then **addKeyPressListener**.
Adding Keyboard Controls

• The window should look like the one below.
Adding Keyboard Controls

• Click and drag an **If statement** to the Drop Statement Here area.
• Select true as the placeholder.
Adding Keyboard Controls

• Click and drag onto the true to replace it.

• Select the Custom Key… and then press the SPACE bar to add it.

• The SPACE bar will be pressed for the astronaut to wave.
Adding Keyboard Controls

• Click on the ‘this’ just above the Methods Panel and click the drop down arrow to select this.adultPerson.
Adding Keyboard Controls

• You should now see the astronautWave procedure listed.

• Click and drag the astronautWave procedure into the spot in the If (top) drop statement here area.

• Should look like the image to the right. Test your program. Click Run and then press the SPACE bar.
Add float Methods/Procedure

• Let’s write another method. This method will have the astronaut go up in the air and then float in a circle around the humvee.

• From the Class drop down menu, select ‘Scene’. It is the area where we need to build the float procedure.

• Select Add Scene Procedure from the list.

• Name the new procedure float.

SAVE OFTEN!!!
Add float Methods/Procedure

• From the ‘this’ drop down menu, select `this.adultPerson` from the list.
Add float Methods/Procedure

• Click and drag the `this.adultPerson` move programming statement into the Code Editor panel.

• Set the direction as Up 1.0 meters, duration to 1 second, and the animation style to Begin and End Gently.

• Drag a Do Together from the bottom panel below the Code editor and place it below the this.adultPerson move line.
Add float Methods/Procedure

• In the Do together click and drag a `this.adultPerson say` statement. Type in “Wheeeeeee!!!” using a custom string. (no quotes)

• Click and drag a `this.adultPerson turn` statement from the Methods panel under the Orientation section.

• For the turn, set the direction to Left 1.0 meters, `asSeenBy` click and select `humvee` from the list, and set the animation style to Begin and End Gently.

SAVE OFTEN!!!
Add float Methods/Procedure

• For the last line of code for the float procedure, click and drag a `this.adultPerson move` statement below the Do Together in the Code Editor.

• Set the direction as Down 1.0 meters, animation style as Begin and End Gently.

• The coding should look like the image below.
Add float Methods/Procedure

• Click on the myFirstMethod tab at the top above the Coding Editor.

• To add the float procedure to the myFirstMethod panel, go to the drop down panel above the Methods Panel and select ‘this’ from the list.

• You should see the float procedure listed. Click and drag it to the Code Editor under the last line of code.

• Test your animation, press Run.
Add a rideHumvee Method/Procedure

• Just like we did with the float procedure, we are going to add procedure for the astronaut to get into the humvee to get ready to ride in it.

• From the Class drop down menu, select ‘Scene’. It is the area where we need to build the rideHumvee procedure.

• Select Add Scene Procedure from the list.

• Name the new procedure rideHumvee.
Add a rideHumvee Method/Procedure

• From the ‘this’ drop down menu, select this.adultPerson from the list.
Add a rideHumvee Method/Procedure

• Click and drag the `this.adultPerson moveTo` programming statement into the Code Editor. See settings below.

• Add a `this.adultPerson` move statement under the previous `moveTo` statement and use the settings listed below.

• Add another `move` statement and use the settings below.

• Add a `this.humvee getFrontLeftDoor` turn statement and use the settings below.

• Check your coding, it should look like the image below.
Add a rideHumvee Method/Procedure

• Add a Do Together as shown below. This part of the procedure will have the astronaut getting into the humvee. You may need to adjust the settings in order to get the astronaut driver in place.

```java
ThreadUtilities.doTogether() -> {
  this.adultPerson.move(MoveDirection.RIGHT, 0.25, Move.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)) add detail;
}

ThreadUtilities.doTogether() -> {
  this.adultPerson.move(MoveDirection.BACKWARD, 0.1, Move.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)) add detail;
}

ThreadUtilities.doTogether() -> {
  this.adultPerson.getLeftHip().turn(TurnDirection.BACKWARD, 0.25) add detail;
}

ThreadUtilities.doTogether() -> {
  this.adultPerson.getRightHip().turn(TurnDirection.BACKWARD, 0.25) add detail;
}

ThreadUtilities.doTogether() -> {
  this.adultPerson.turn(TurnDirection.RIGHT, 0.25) add detail;
}

ThreadUtilities.doTogether() -> {
  this.adultPerson.move(MoveDirection.FORWARD, 0.25) add detail;
}

ThreadUtilities.doTogether() -> {
  this.adultPerson.move(MoveDirection.RIGHT, 0.25) add detail;
}
```

SAVE OFTEN!!!
Add a rideHumvee Method/Procedure

• These last lines of code close the door to the humvee, and set the vehicle to the astronaut so when the humvee moves, the astronaut moves with it, thus making the procedure complete.

```java
void rideHumvee() {
    do in order:
    (this.humvee) getFrontLeftDoor();
    turn(TurnDirection.LEFT, 0.25),
    Turn.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)
    add detail();

    (this.adultPerson) moveTo((this.humvee) getFrontLeftDoor());
    MoveTo.pathStyle(PathStyle.SMOOTH),
    MoveTo.duration(1.0),
    MoveTo.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)
    add detail();

    (this.adultPerson) move(MoveDirection.DOWN, 0.5),
    Move.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)
    add detail();

    (this.adultPerson) move(MoveDirection.LEFT, 0.6),
    add detail();

    (this.humvee) getFrontLeftDoor();
    turn(TurnDirection.RIGHT, 0.25),
    Turn.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)
    add detail();

    ThreadUtilities.doTogether(0-> {
        (this.adultPerson) move(MoveDirection.RIGHT, 0.25),
        Move.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)
        add detail();

        (this.adultPerson) move(MoveDirection.BACKWARD, 0.1),
        Move.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)
        add detail();

        (this.adultPerson) getLeftHip();
        turn(TurnDirection.BACKWARD, 0.25)
        add detail();

        (this.adultPerson) getRightHip();
        turn(TurnDirection.BACKWARD, 0.25)
        add detail();

        (this.adultPerson) turn(TurnDirection.RIGHT, 0.25)
        add detail();

        (this.adultPerson) move(MoveDirection.FORWARD, 0.25)
        add detail();

        (this.adultPerson) move(MoveDirection.RIGHT, 0.25)
        add detail();

    });

    (this.humvee) getFrontLeftDoor();
    turn(TurnDirection.LEFT, 0.25),
    Turn.animationStyle(AnimationStyle.BEGIN_AND_END_GENTLY)
    add detail();

    (this.adultPerson) setVehicle(this.humvee);
}
```

• Full code should look like the image below:
Add the rideHumvee Procedure

• Click on the myFirstMethod tab at the top above the Coding Editor.

• To add the rideHumvee procedure to the myFirstMethod coding panel, go to the drop down panel above the Methods Panel and select ‘this’ from the list.

• You should see the float procedure listed. Click and drag the rideHumvee to the Code Editor under the last line of code.

• Test your animation, press Run.

SAVE OFTEN!!!
Pulling it all together – Add arrow keys to move humvee around the scene

• To add the arrow keys to move the humvee around the scene.
• Click on the Initialize Event Listener tab above the Code Editor panel.
• Click on the Add Event Listener button.
• Select Keyboard and then addObjectMoverFor and humvee from the list.
• This adds the ability to move the humvee around the scene using the arrow keys.
Pulling it all together – Add arrow keys to move humvee around the scene

• Before we run our animation, we need to have the camera stay with the humvee as it moves.

• From the ‘this’ menu, click the down arrow and select this.camera from the list.
Pulling it all together – Add arrow keys to move humvee around the scene

• Find the vehicle section in the Method Editor panel, and click and drag `this.camera setVehicle` statement to the Code Editor panel just above the `rideHumvee` code.

• Select `this.humvee` from the list to set the camera’s vehicle to the humvee.

• Now, test your animation, press the run key.

SAVE OFTEN!!!
Pulling it all together – Adding instructions

• Now, drag an astronaut say method at the bottom of your code and just above the rideHumvee programming statement. Select and click on Custom textString.... Type in, “Now press the arrow keys to see me move, and to drive me around.” You may change the text as long as it give the correct instructions. Change the duration on the command to make the speech stay on the screen longer. To do this, click on add detail... on the astronaut say line of code, then choose duration, other..., and then type in 4 or 5.
• Your final code for `myFirstMethod` should look like this:

```
this.adultPerson.say("Uh, hello... woman gone before...");
this.adultPerson.move(MoveDirection.UP, 1.0); 
this.adultPerson.turn(TurnDirection.RIGHT, 1.0); 
this.adultPerson.move(MoveDirection.DOWN, 1.0); 
this.adultPerson.astronautWave();

float();
this.adultPerson.say("After I get into the moon rover humvee, press the arrow keys to make the humvee move...");
this.camera.setVehicle(this.humvee);
this.rideHumvee();
```

Press Run again to test out your world.
Notes for later

• If you later want to unglue the astronaut from the humvee, set the `vehicle` property of the astronaut from humvee back to ‘this’. Don’t forget to make the astronaut stand back up. Just reverse the coding you used to make the astronaut sit in the first place.

• If you later want to un-glue the camera from the humvee, also set its `vehicle` property back to ‘this’.

• Adding `delay statements` helps control the flow of the animation. Sometimes it just needs to slow down a bit.
And now for some fun...

• Let’s make a new introduction for our animation...

• First, you will need to have a 15 – 20 second mp3 audio file ready to go. Download one legally from the Internet. Have it ready on your desktop.

• Click on the myFirstMethod tab.

• From the ‘this’ drop down menu above the Methods panel, be sure to select ‘this’ from the menu.

• From the Control Tiles below the Code Editor, click and drag a doTogether command to the top of the coding in the code editor.
And now for some fun...

• In the Methods Editor, scroll down and locate the audio section. Click and drag the `this.playAudio` statement and place it into the `doTogether`.

• Choose to import a sound file. From the window, locate the mp3 file you downloaded legally previously.

• Now go to the Control Tiles located under the Code Editor, and click and drag a `doInOrder` command under the sound file within the `doTogether` code box.

• Now go to the ‘this’ drop down menu located just above the Methods panel, and select `this.camera`.
And now for some fun...

• Scroll to the bottom of the procedures and find the `this.camera delay` statement and drag it into the `doInOrder` part of the coding for the introduction. Click on the add Detail and set the duration between 2 – 5 seconds depending on your intro music.

• Add `this.camera moveAndOrientTo` statement to the `doInOrder`, and select `this.astronautHeadShot` from the list. This will change the camera to the astronaut head shot position.

• Add a `this.camera delay` and make the duration 2-5 seconds.
And now for some fun...

• Add `this.camera moveAndOrientTo` statement into the `doInOrder`, and select `this.humveeView` from the list.

• Add `this.camera` delay and set the duration for 2-5 seconds.

• Add `this.camera moveAndOrientTo` statement to the `doInOrder` and select `this.startView` from the list.
And now for some fun...

• Your introduction code should look like the following:

```java
ThreadUtilities.doTogether( () -> {
    this.playAudio( new AudioSource( "Opening Remastered.mp3 (24.58s)" ) );
}, 0 -> {
    /* do in order */ {
        this.camera delay( 5.0 );
        this.camera moveAndOrientTo( this.astronautHeadShot add detail );
        this.camera delay( 5.0 );
        this.camera moveAndOrientTo( this.humveeView add detail );
        this.camera delay( 5.0 );
        this.camera moveAndOrientTo( this.startView add detail );
    }
});
```
Congratulations! You have just made your first Alice world. There are many more things that you can do with Alice, so keep exploring it!