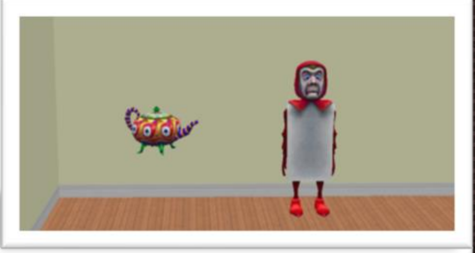
ORACLE

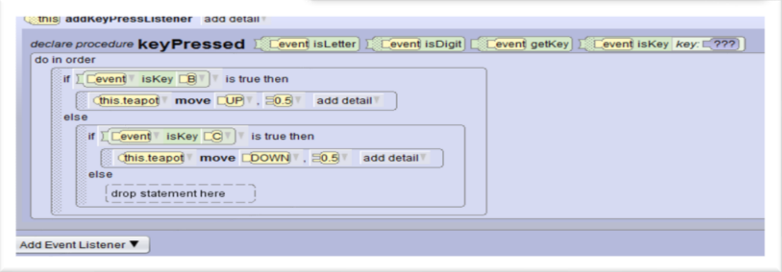
Academy

Java Fundamentals

2-11

Keyboard Controls



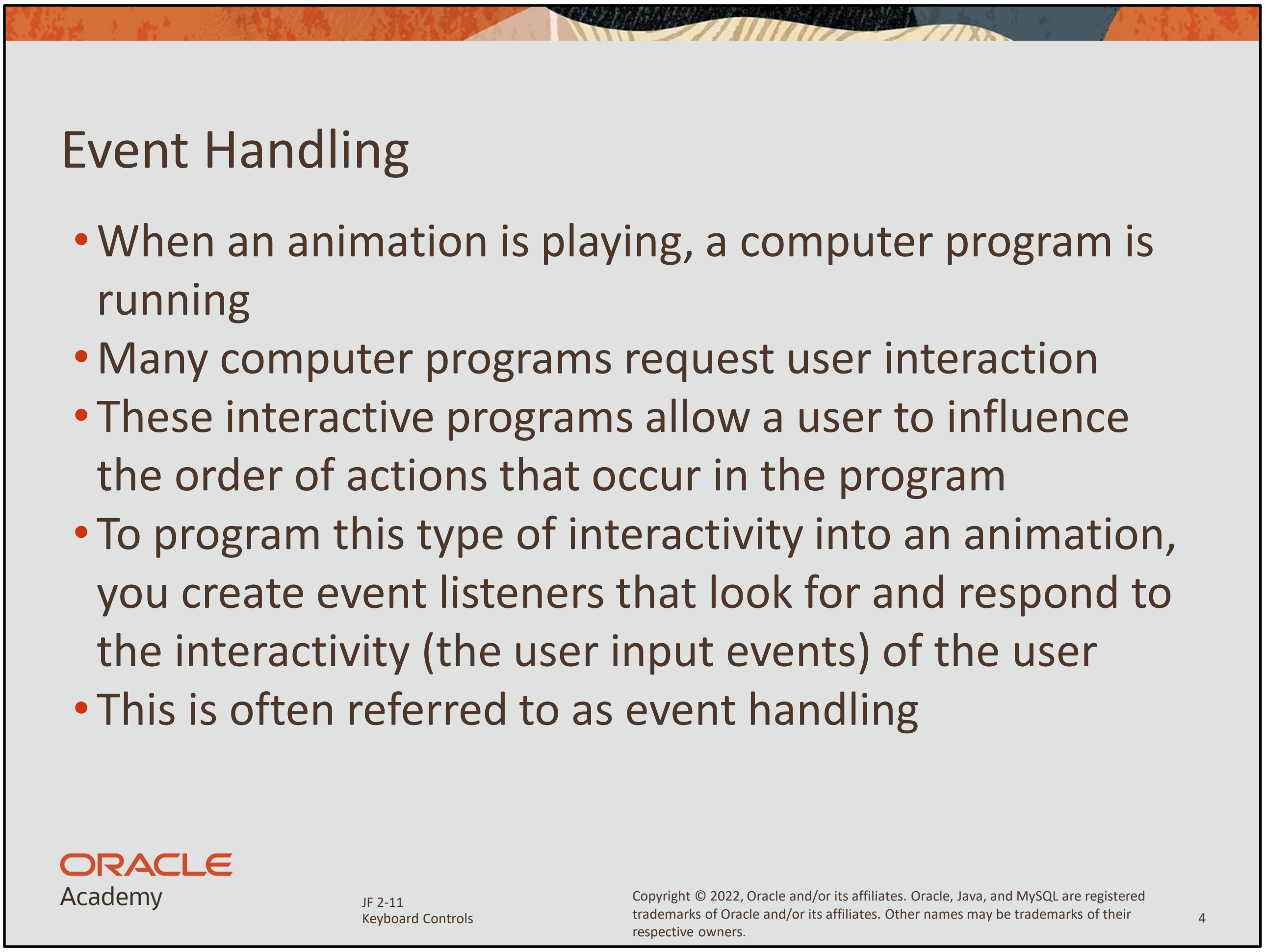


```
this addKeyPressListener add detail
declare procedure keyPressed  event isLetter  event isDigit  event getKey  event isKey key: ???
do in order
    if event isKey B is true then
        this.teapot move UP , 0.5 add detail
    else
        if event isKey C is true then
            this.teapot move DOWN , 0.5 add detail
        else
            drop statement here
```

ORACLE
Academy

Objectives

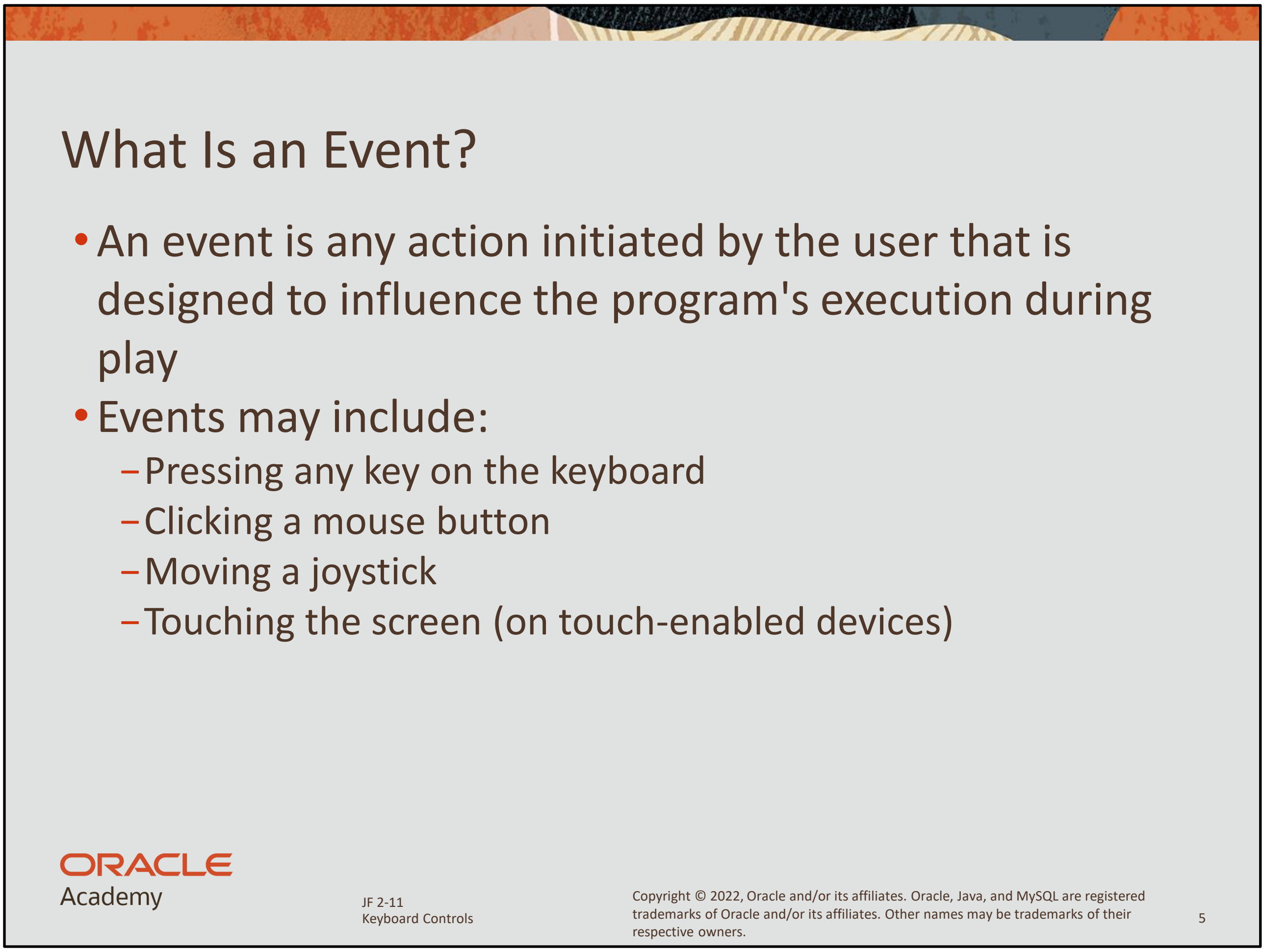
- This lesson covers the following objectives:
  - Create an opening sequence
  - Use keyboard controls to manipulate an animation
  - Save your Class file
  - Using the starter tab
  - Add an existing class file to an animation



Event Handling

- When an animation is playing, a computer program is running
- Many computer programs request user interaction
- These interactive programs allow a user to influence the order of actions that occur in the program
- To program this type of interactivity into an animation, you create event listeners that look for and respond to the interactivity (the user input events) of the user
- This is often referred to as event handling

An animation that does the same thing time after time soon loses its appeal. Creating animations where the user can dictate what happens increases the interest and lifespan of your program.

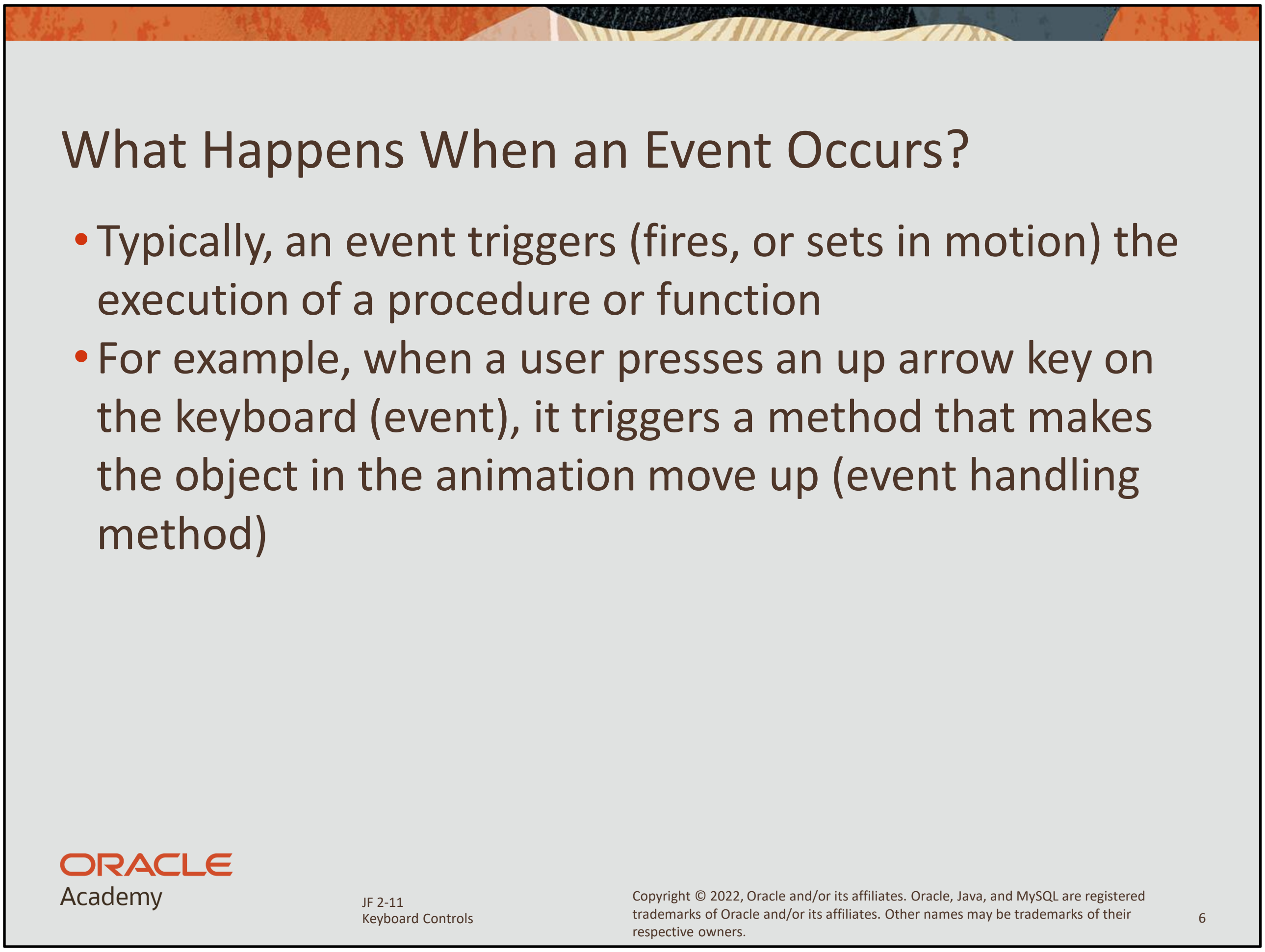


What Is an Event?

- An event is any action initiated by the user that is designed to influence the program's execution during play
- Events may include:
  - Pressing any key on the keyboard
  - Clicking a mouse button
  - Moving a joystick
  - Touching the screen (on touch-enabled devices)

ORACLE Academy

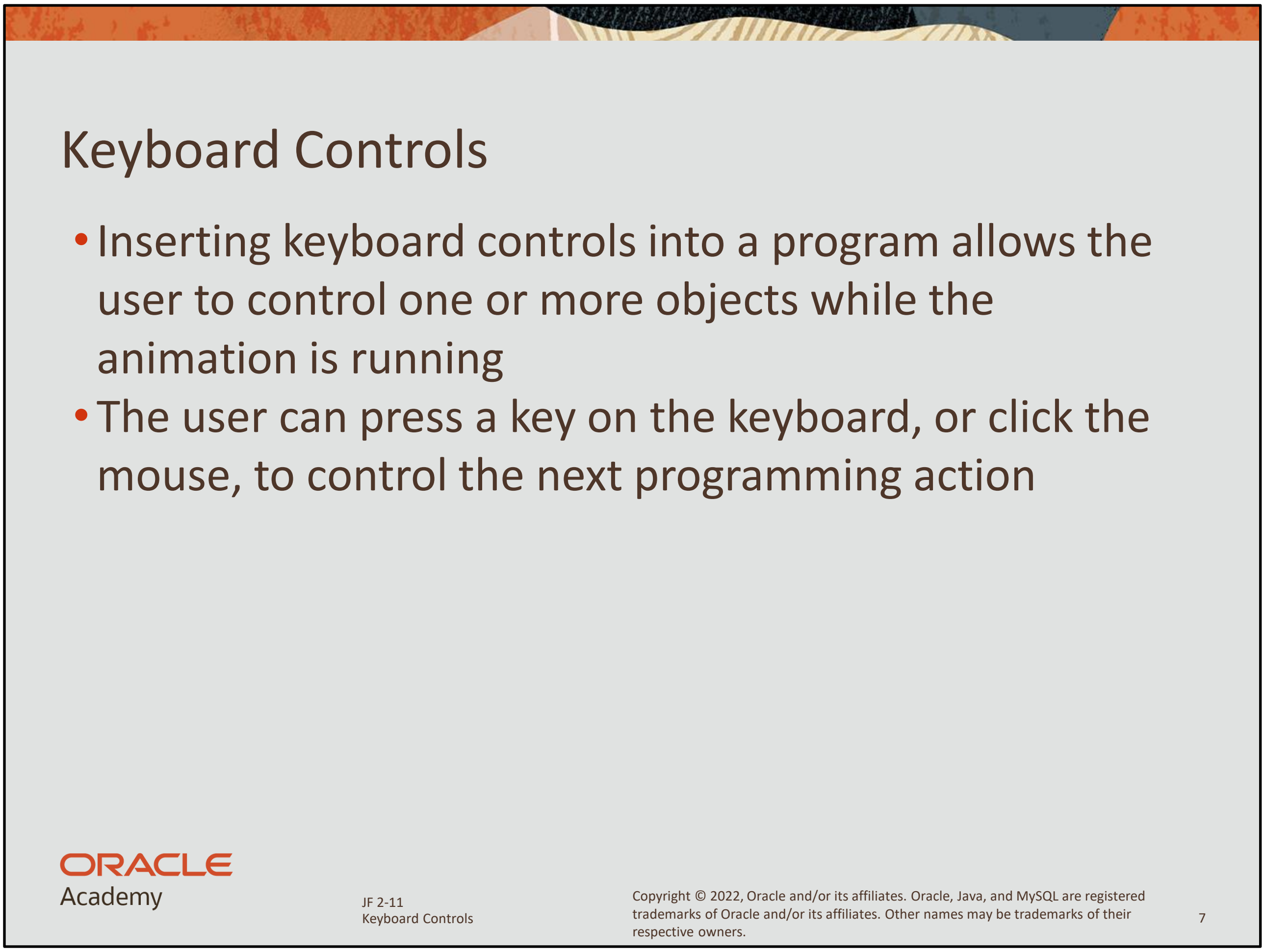
Events are things that occur when a certain condition is met. Either a key press or mouse click/movement or if the object is in a specific place in the animation. An event can also be triggered by the time of the animation.



What Happens When an Event Occurs?

- Typically, an event triggers (fires, or sets in motion) the execution of a procedure or function
- For example, when a user presses an up arrow key on the keyboard (event), it triggers a method that makes the object in the animation move up (event handling method)

How you code an event is entirely up to you. It can be a simple procedure that moves a single object or a complex one that has repercussions for multiple objects within your world.



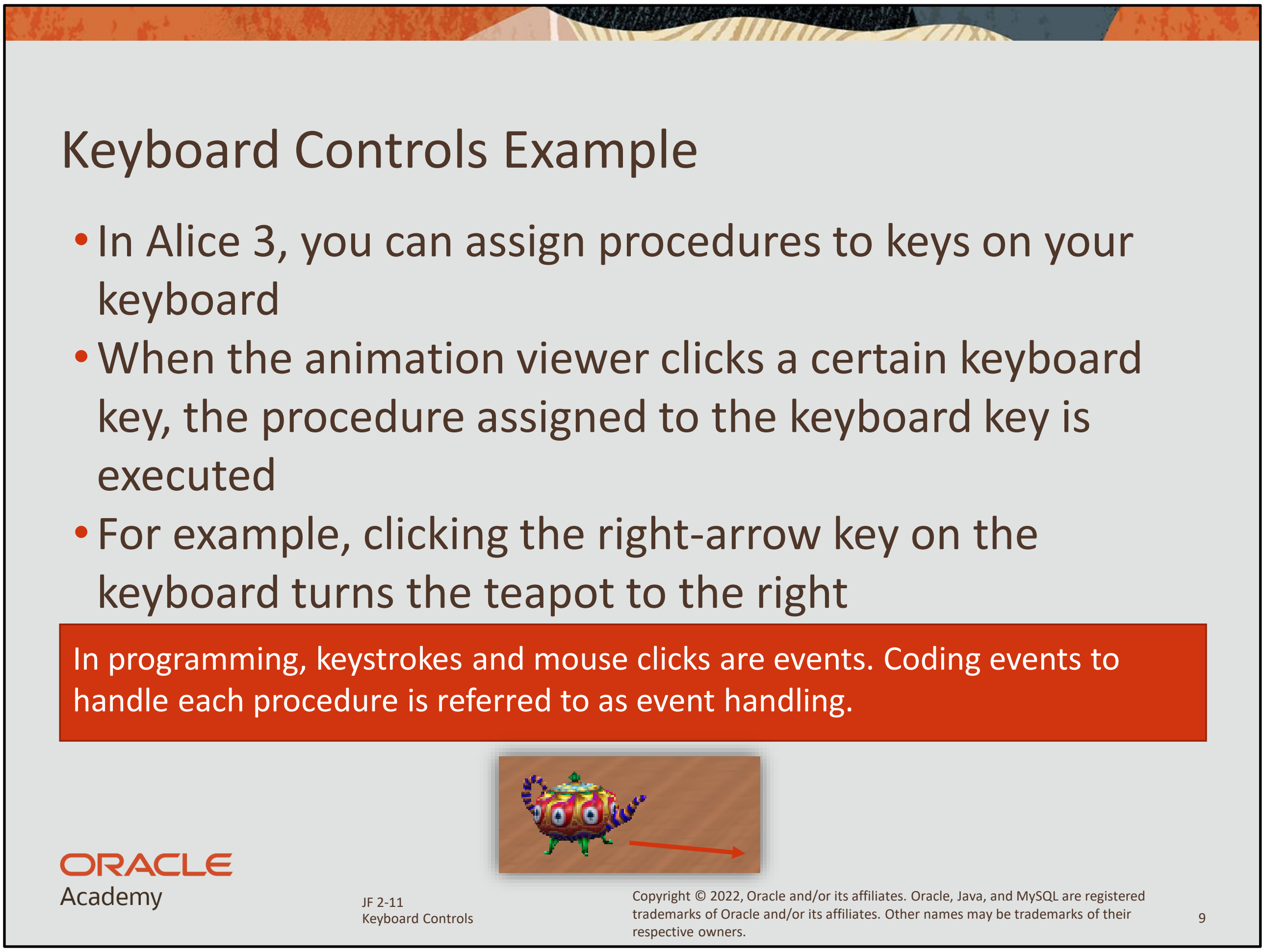
Keyboard Controls

- Inserting keyboard controls into a program allows the user to control one or more objects while the animation is running
- The user can press a key on the keyboard, or click the mouse, to control the next programming action

All of the main keys on a keyboard can be coded to be the trigger for an event.

Keyboard Controls

- With keyboard controls, you can:
  - Create scenes where the user controls an object that interacts with other objects
  - Create animations that execute conditionally, based on a key press or mouse click
  - Create games where the user is required to control an object to win the game



Keyboard Controls Example

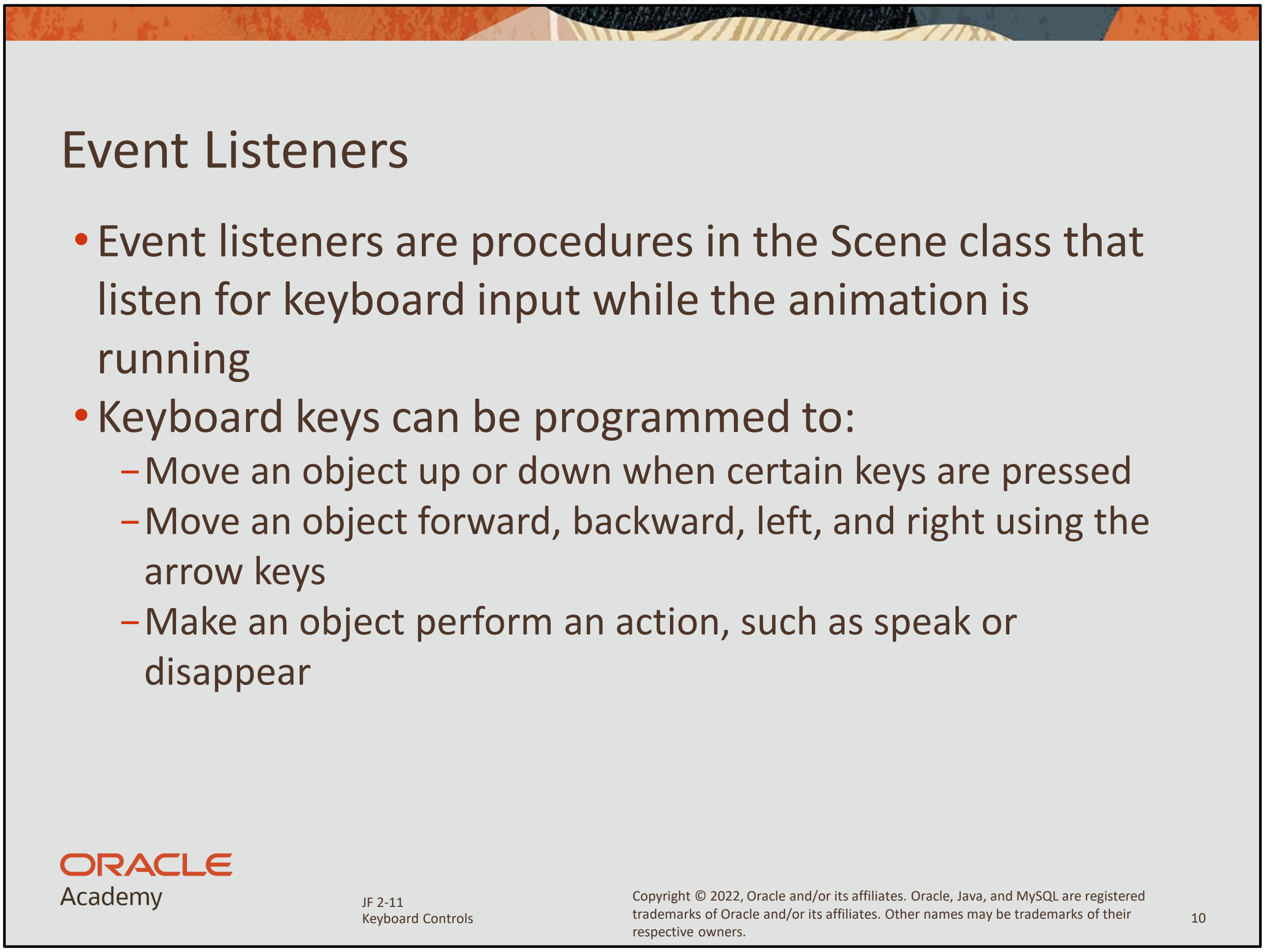
- In Alice 3, you can assign procedures to keys on your keyboard
- When the animation viewer clicks a certain keyboard key, the procedure assigned to the keyboard key is executed
- For example, clicking the right-arrow key on the keyboard turns the teapot to the right

In programming, keystrokes and mouse clicks are events. Coding events to handle each procedure is referred to as event handling.

ORACLE
Academy

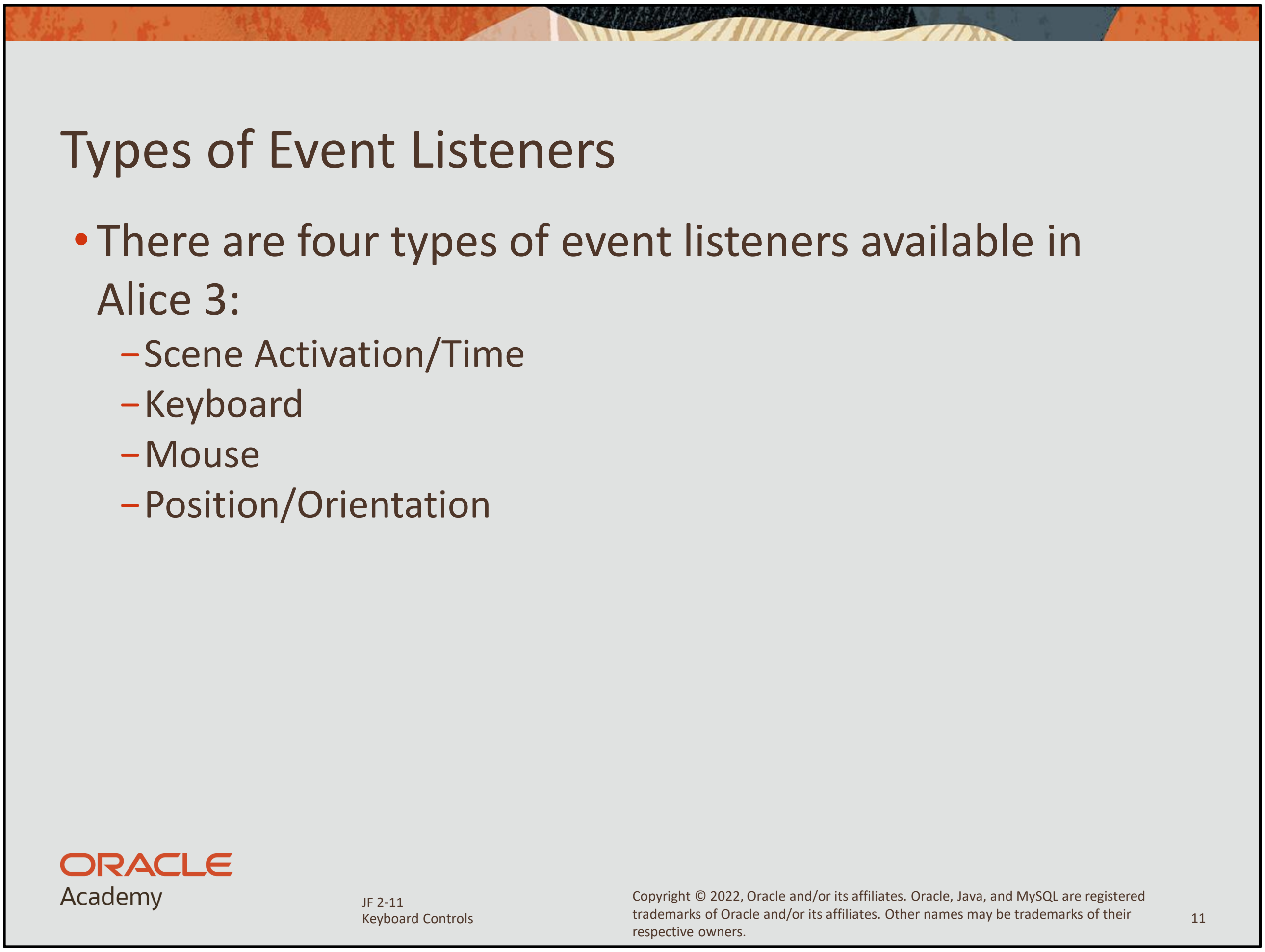
JF 2-11
Keyboard Controls

When coding the arrow keys to control your object bear in mind that the four arrow keys are tied to the W, A, S and D keys. You will not be able to use these two sets of keys for a two player game as they produce the same input.



Event Listeners

- Event listeners are procedures in the Scene class that listen for keyboard input while the animation is running
- Keyboard keys can be programmed to:
  - Move an object up or down when certain keys are pressed
  - Move an object forward, backward, left, and right using the arrow keys
  - Make an object perform an action, such as speak or disappear



Types of Event Listeners

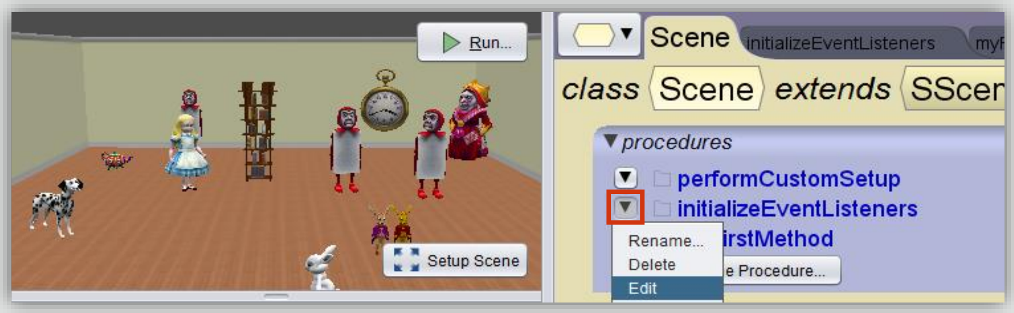
- There are four types of event listeners available in Alice 3:
  - Scene Activation/Time
  - Keyboard
  - Mouse
  - Position/Orientation

ORACLE
Academy

There are multiple sub menus that can be used to specify exactly what the event trigger is. Have a look through the menus and familiarize yourself with all of the options.

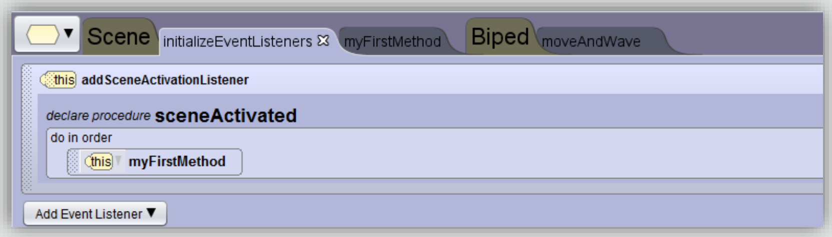
Steps to Access Event Listeners

- In the Code editor, click the Scene tab
- Click the button next to initializeEventListeners and choose Edit
- This opens the initializeEventListeners tab if it is not already open



Event Listeners Tab

- The initializeEventListeners tab is where you can add an event listener to your code



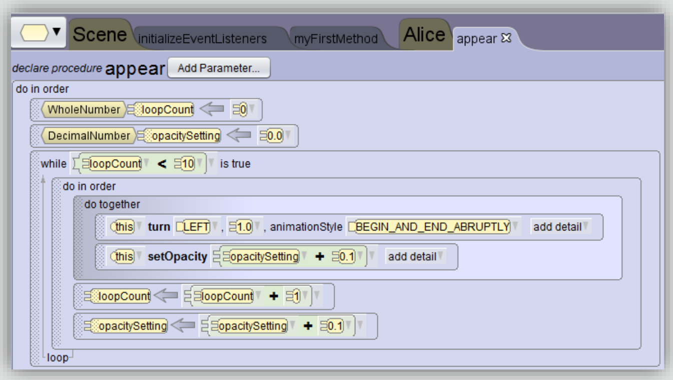
```
this addSceneActivationListener
  declare procedure sceneActivated
    do in order
      this myFirstMethod
```

- The sceneActivated listener is where you can create an animation that will play before the myFirstMethod begins
- This can be used as an opening sequence to your code

The addEventlistener button is where you create all of your events. You can create as many events as you like. Be careful not to code events that will contradict each other if both are executed at the same time.

Scene Activation Listener

- You can create your own procedure or use one of the built in procedures to create an opening sequence
- This procedure named "appear" turns Alice around and makes her visible in our scene



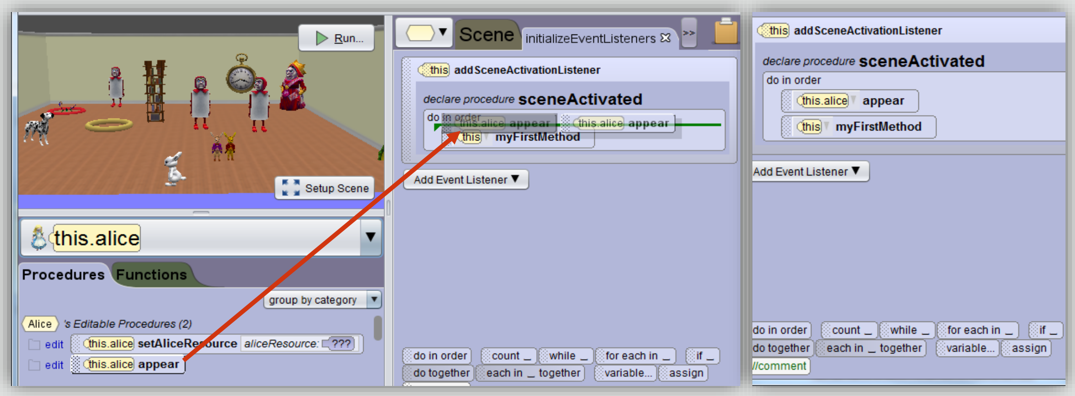
```
declare procedure appear
do in order
    WholeNumber loopCount ⟵ 0
    DecimalNumber opacitySetting ⟵ 0.0
    while loopCount < 10 is true
        do in order
            do together
                this turn LEFT , 1.0 , animationStyle BEGIN_AND_END_ABRUPTLY
                this setOpacity opacitySetting + 0.1
            loopCount ⟵ loopCount + 1
            opacitySetting ⟵ opacitySetting + 0.1
    loop
```

ORACLE
Academy

When creating code blocks that have multiple nested statements in them it is always a good idea to add comments to the code so that it is easier to read and understand.

Scene Activation Listener

- To enable the procedure, drag it from the Alice procedure list and place it before the myFirstMethod call in the sceneActivated listener

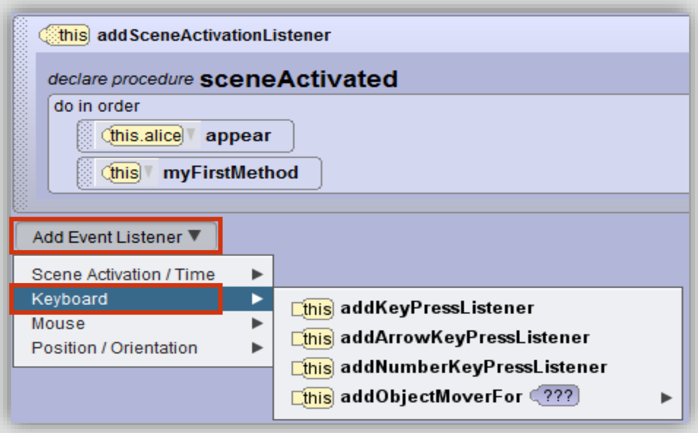


- Test your opening sequence!

By placing it before the myFirstMethod call the appear procedure will execute fully before the animation begins. This is useful when creating an opening sequence for your animation.

Keyboard Listeners

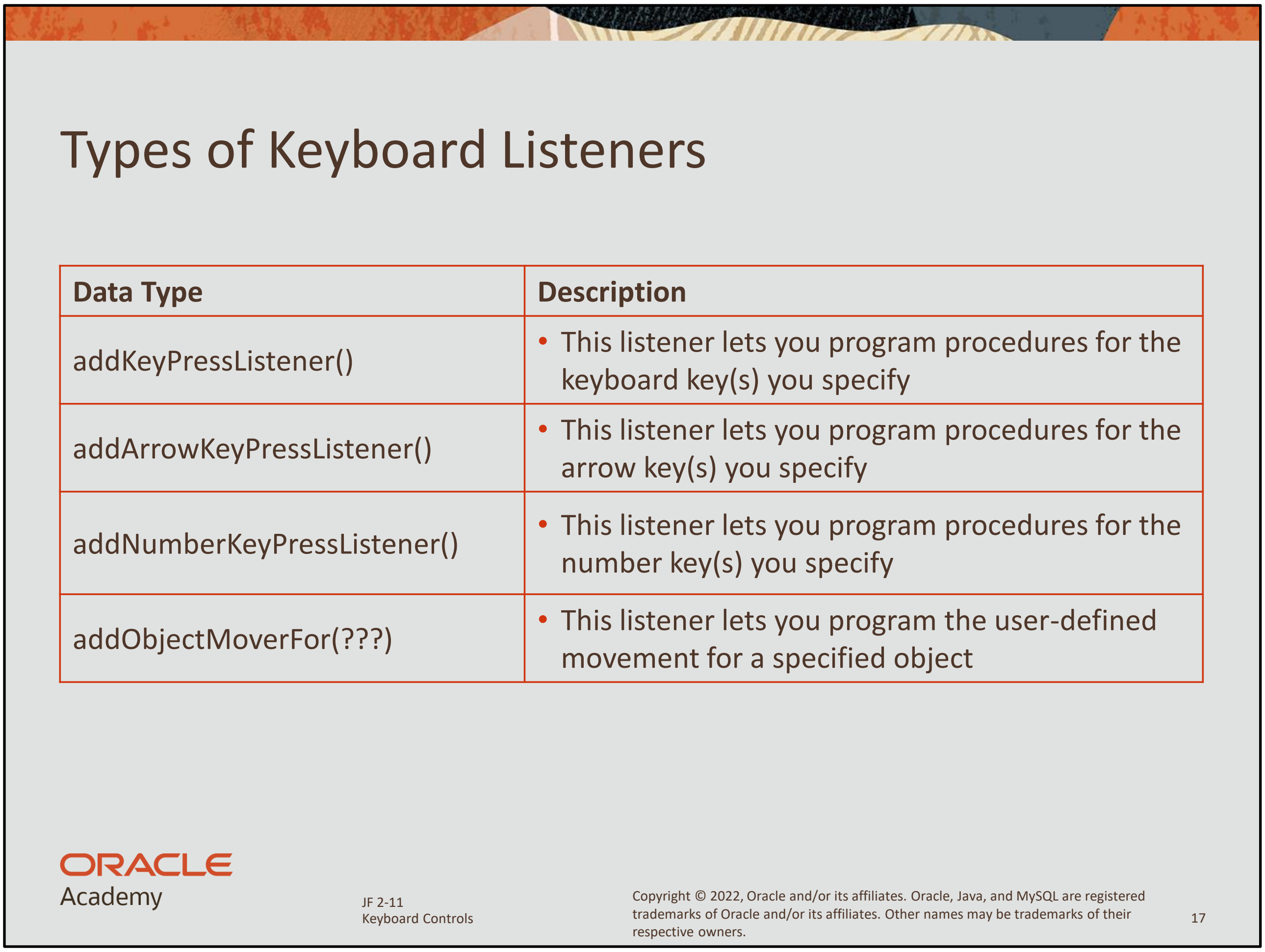
- Keyboard listeners:
  - Are found in the Add Event Listener drop-down menu
  - Listen for, and react to, a keyboard keystroke that you specify



```
this addSceneActivationListener
  declare procedure sceneActivated
    do in order
      this.alice appear
      this myFirstMethod
```

Add Event Listener ▼
Scene Activation / Time ▶
Keyboard ▶
Mouse ▶
Position / Orientation ▶
this addKeyPressListener
this addArrowKeyPressListener
this addNumberKeyPressListener
this addObjectMoverFor ??? ▶

From this list you can choose from most of the keys available on a keyboard to be your event trigger.

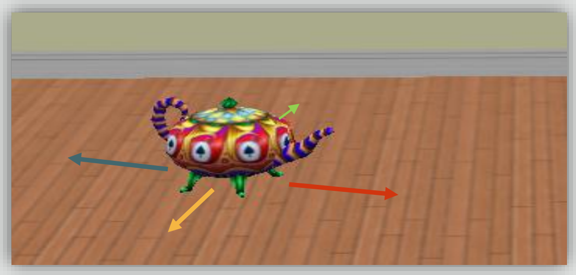


Types of Keyboard Listeners

Data Type	Description
addKeyPressListener()	• This listener lets you program procedures for the keyboard key(s) you specify
addArrowKeyPressListener()	• This listener lets you program procedures for the arrow key(s) you specify
addNumberKeyPressListener()	• This listener lets you program procedures for the number key(s) you specify
addObjectMoverFor(???)	• This listener lets you program the user-defined movement for a specified object

Program Keyboard Event Listeners

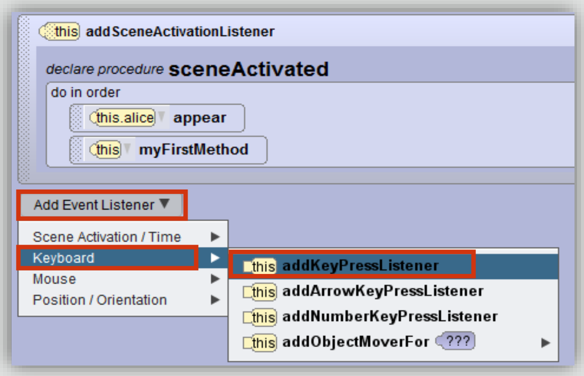
- For example, we will program keyboard event listeners to command the teapot to move up and down using the B and C keys, and move left, right, forward and backward using the arrow keys
- We are in wonderland after all!



ORACLE
Academy

Steps to Add Keyboard Event Listener

- Select the Add Event Listener drop-down list
- Select Keyboard
- Select addKeyPressListener



this addSceneActivationListener

declare procedure **sceneActivated**

do in order

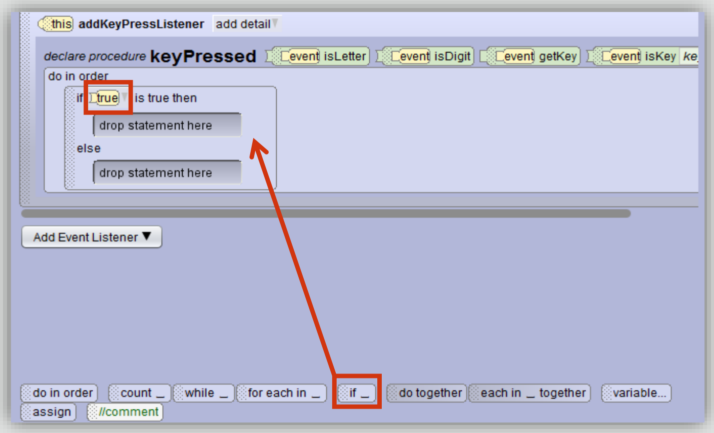
this.alice appear

this myFirstMethod

ORACLE Academy

Steps to Program the IF Control Structure

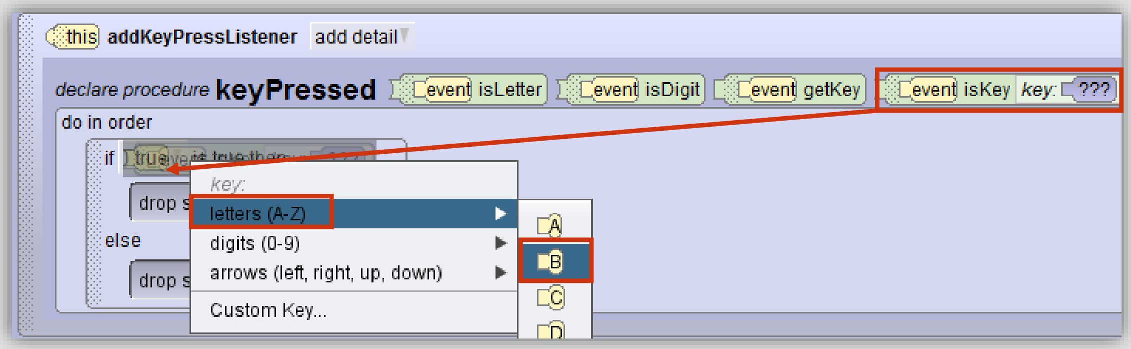
- Drag the IF control structure into addKeyPressListener
- Select the true condition



True is used as a placeholder value here.

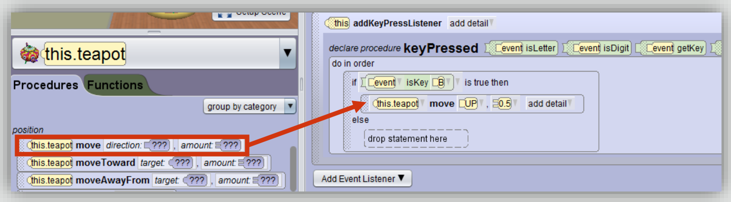
Steps to Select the Keyboard Key to Activate an Object's Motion

- Drag the isKey: ??? tile onto the true condition
- A key menu appears
- From the drop-down menu, select the keyboard key that you want to use to control the motion

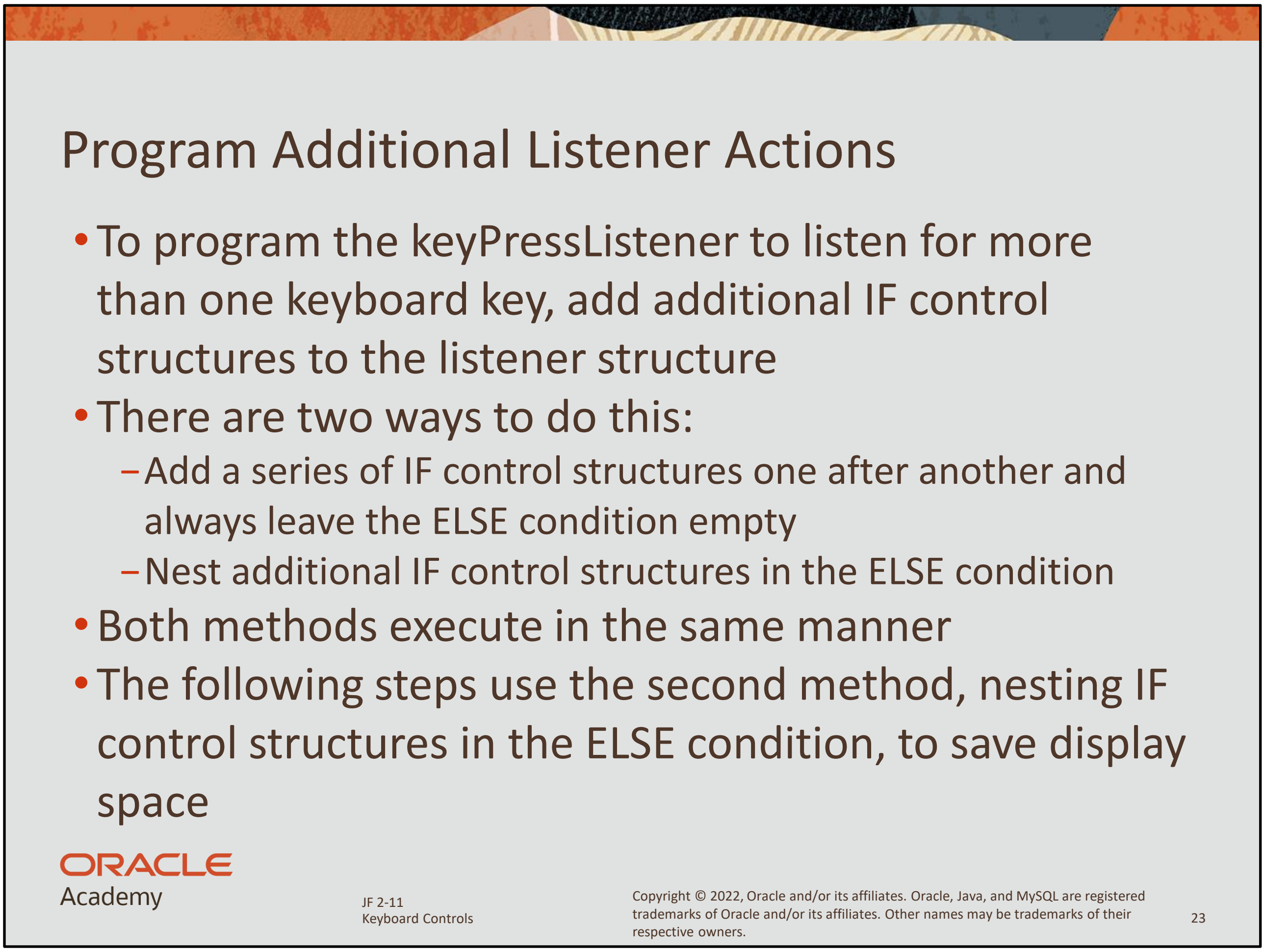


Steps to Program Motions Activated by Key Press

- From the Instance drop-down menu, select the object controlled by the keyboard key
- Drag the procedure that should be activated by the keyboard key into the IF control structure and select the arguments
- You could drag multiple procedures and control structures into the IF control structure
- For example, when the B key is pressed, the teapot moves and then turns


this.teapot
Procedures
Functions
group by category
position
this.teapot move direction: ??? , amount: ???
this.teapot moveToward target: ??? , amount: ???
this.teapot moveAwayFrom target: ??? , amount: ???
Add Event Listener

```
this addKeyPressListener add detail
declare procedure keyPressed  event isLetter  event isDigit  event getKey
do in order
    if event isKey B is true then
        this.teapot move UP , 0.5 add detail
    else
        drop statement here
```



Program Additional Listener Actions

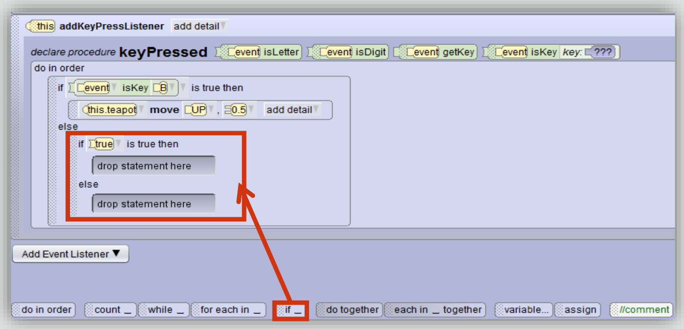
- To program the keyPressListener to listen for more than one keyboard key, add additional IF control structures to the listener structure
- There are two ways to do this:
  - Add a series of IF control structures one after another and always leave the ELSE condition empty
  - Nest additional IF control structures in the ELSE condition
- Both methods execute in the same manner
- The following steps use the second method, nesting IF control structures in the ELSE condition, to save display space

ORACLE
Academy

The difference between the two methods here is that by placing the IF statements in the ELSE part of the statement as soon as a condition is true the rest will not be checked. If you create separate IF statements then all of them will be checked regardless of a condition having been met. The first method only reads a single key. The second one allows for multiple keys to be pressed at the same time to accommodate diagonal movement for instance.

Steps to Program Additional Listener Actions

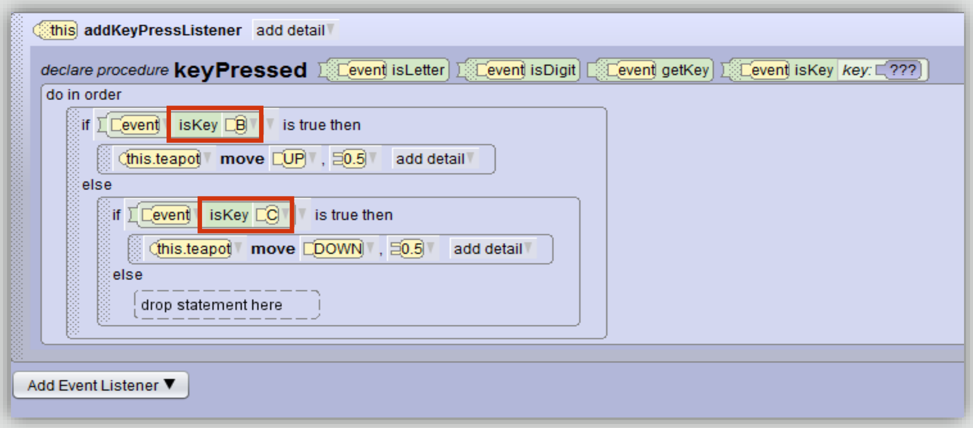
- Drag an IF control structure into the ELSE condition of an existing IF control structure and select the true condition
- Drag the isKey ??? variable onto the true argument
- Specify the keyboard key to listen for
- Specify the programming statements to execute



```
this addKeyPressListener add detail
  declare procedure keyPressed  event isLetter  event isDigit  event getKey  event isKey key: ???
  do in order
    if event isKey B is true then
      this.teapot move UP , 0.5 add detail
    else
      if true is true then
        drop statement here
      else
        drop statement here
```

Completed Programming Instruction Example

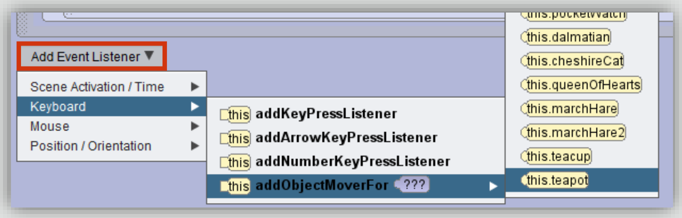
- Below is an example of a teapot programmed to move up and down using the B and C keyboard keys



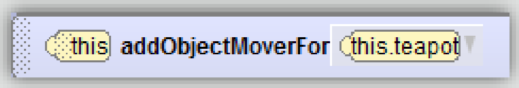
```
this addKeyPressListener  add detail
  declare procedure keyPressed  event isLetter  event isDigit  event getKey  event isKey key: ???
  do in order
    if event isKey B is true then
      this.teapot move UP , 0.5 add detail
    else
      if event isKey C is true then
        this.teapot move DOWN , 0.5 add detail
      else
        drop statement here
```

Steps to Move Objects Using Arrow Keys

- Select the Add Event Listener drop-down menu
- Select Keyboard
- Select addObjectMoverFor
- Select the entity, or object, to control



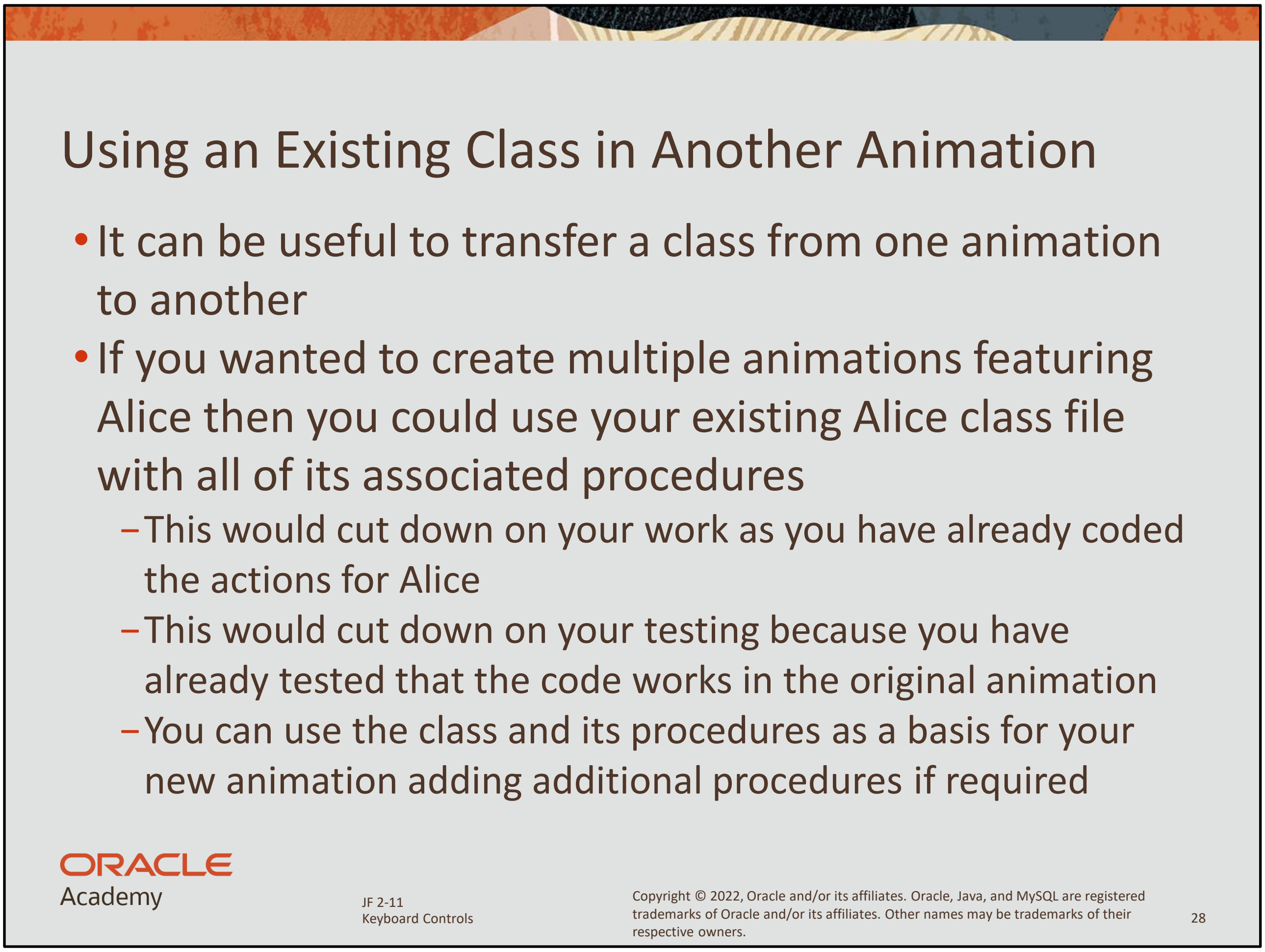
- Creates the following line of code



```
this addObjectMoverFor this.teapot
```

Steps to Test Event Listeners

- Run the animation
- Click inside the animation window with your cursor
- Use the keyboard keys (specified in addKeyPressListener) to make the object perform the procedure (move up and down)
- Use the arrow keys on your keyboard to move the object forward, backward, right, and left



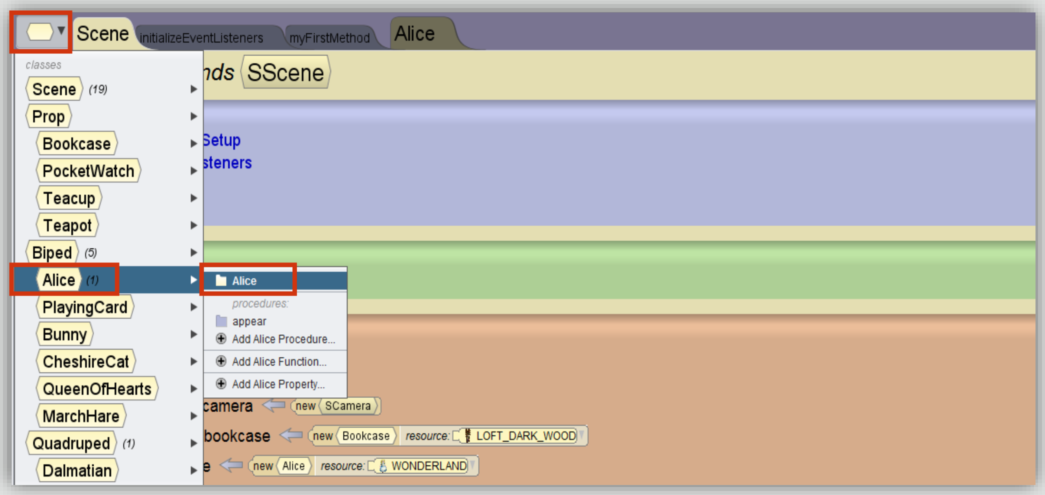
Using an Existing Class in Another Animation

- It can be useful to transfer a class from one animation to another
- If you wanted to create multiple animations featuring Alice then you could use your existing Alice class file with all of its associated procedures
  - This would cut down on your work as you have already coded the actions for Alice
  - This would cut down on your testing because you have already tested that the code works in the original animation
  - You can use the class and its procedures as a basis for your new animation adding additional procedures if required

As you have seen creating detailed animations takes a lot of work. The transfer of a class's code from one animation to another is an incredibly useful feature of Alice 3. A class can be transferred fully or in part allowing you to take only the parts that you need.

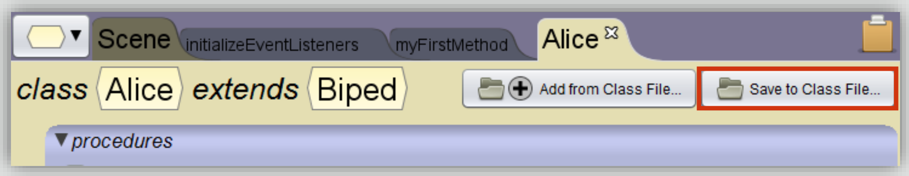
Saving a Class File in Alice 3

- Using the class list button, choose Alice from the list and then click on the Alice class

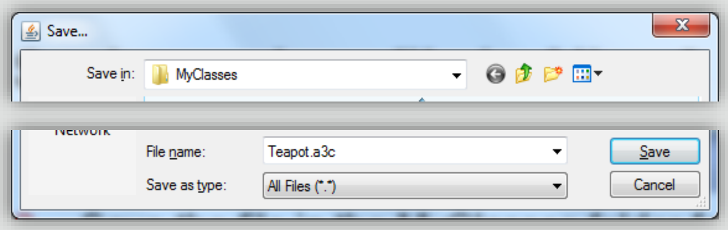


Saving a Class File in Alice 3

- Click on the Save to Class File button



- Save the file in the MyClasses folder for easy access when adding it to other animations and name the class with an appropriate name



Save in: MyClasses

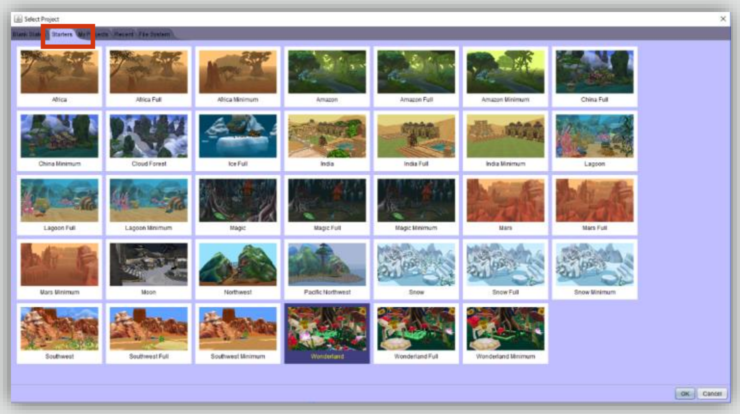
File name: Teapot.a3c

Save as type: All Files (*.*)

By saving your class in the MyClasses folder it will appear in the scene editors gallery under the My Classes tab.

Using the Starter Tab to Create a World

- Alice 3 has pre-built worlds that can be used to quickly create a full and interesting animation
- Use the starters tab from the new project interface to choose your world

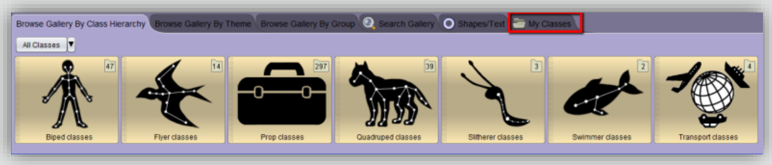


ORACLE Academy

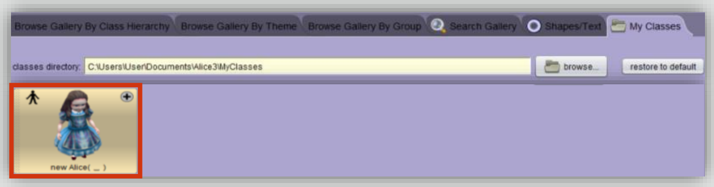
The starter tab is a way of creating animations in a pre-populated world so that you don't have to add and position the scenery objects.

Adding a Class File in Alice 3

- Create a new project in Alice and go to the scene editor to access the gallery
- Choose the My Classes tab from the gallery



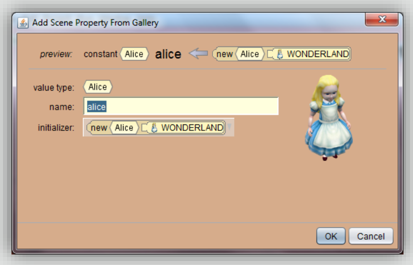
- This gives you access to the MyClasses folder where your classes should be saved


classes directory: C:\Users\User\Documents\Alice3\MyClasses

The My Classes tab gives you easy access to the classes that you have saved in the myClasses folder. If you have saved your classes somewhere else then you can browse to them by using the browse button.

Adding a Class File in Alice 3

- Add the class as you would any other class from the gallery



Add Scene Property From Gallery

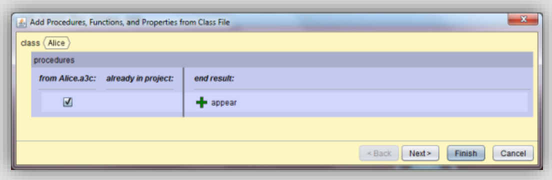
preview: constant Alice alice ← new Alice WONDERLAND

value type: Alice

name: alice

initializer: new Alice WONDERLAND

- When you click OK you are presented with a list of the procedures associated with the class file



Add Procedures, Functions, and Properties from Class File

class Alice

procedures

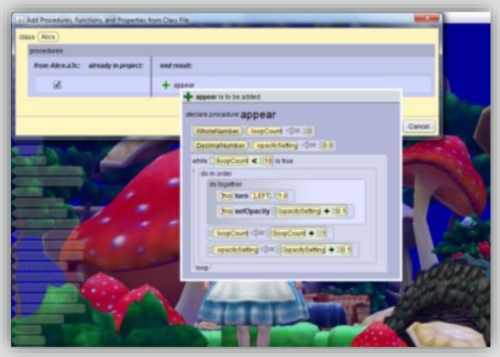
from Alice.a3c:	already in project:	end result:
☑		+ appear

ORACLE
Academy

The tick box in the left column allows you to choose what procedures you want to bring across to your new animation.

Adding a Class File in Alice 3

- Placing the cursor over the green cross expands the code for the procedure



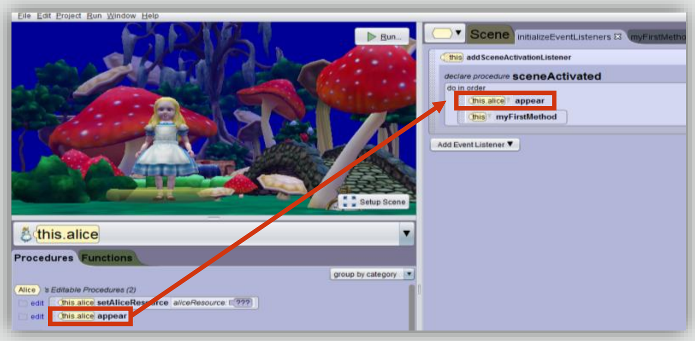
appear is to be added

```
declare procedure appear
WhileNumber loopCount ⇦ 0
DecimalNumber opacitySetting ⇦ 0.0
while loopCount < 10 is true
  do in order
    do together
      this turn LEFT, 1.0
      this setOpacity opacitySetting + 0.1
    loopCount ⇦ loopCount + 1
    opacitySetting ⇦ opacitySetting + 0.1
loop
```

- Clicking finish will add the Alice object into your new animation

Adding a Class File in Alice 3

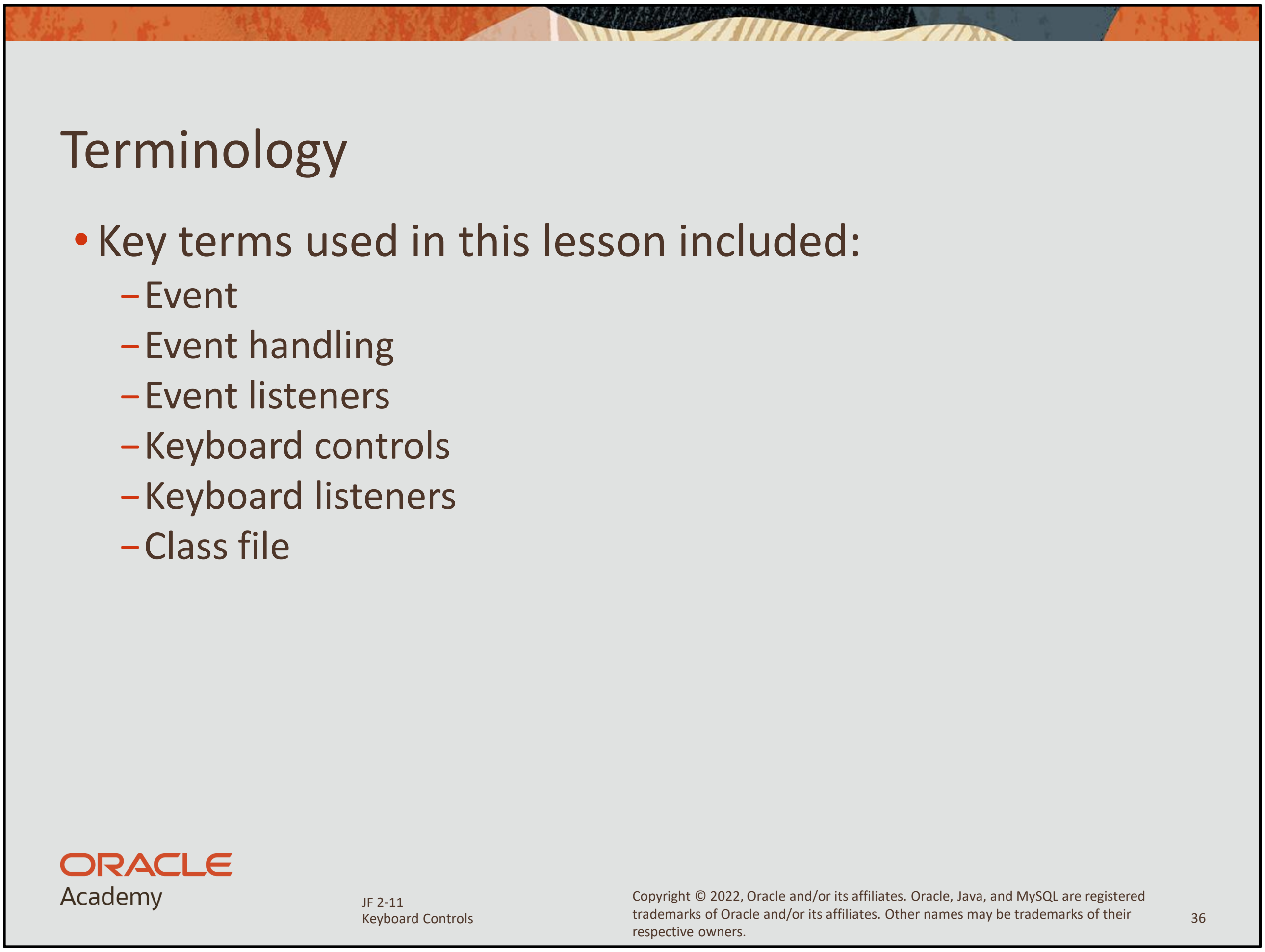
- You can now use the Alice character and her procedures in your new animation
- You can add the appear method to the new sceneActivated listener so that Alice appears in a consistent way across all of your animations



```
this addSceneActivationListener
  declare procedure sceneActivated
    do in order
      this.alice appear
      this myFirstMethod
```

```
Alice 's Editable Procedures (2)
edit  this.alice setAliceResource aliceResource: ???
edit  this.alice appear
```

This lets you quickly create an animation using all of your existing code. This is known as a programmers code library and is common practice within industry.



Terminology

- Key terms used in this lesson included:
  - Event
  - Event handling
  - Event listeners
  - Keyboard controls
  - Keyboard listeners
  - Class file

ORACLE
Academy

Summary

- In this lesson, you should have learned how to:
  - Create an opening Sequence
  - Use keyboard controls to manipulate an animation
  - Save your Class file
  - Using the starter tab
  - Add an existing class file to an animation

ORACLE

Academy