

## Java Fundamentals

### 2-1: Getting Started with Alice 3

### Practice Activities

Nama : Ngakan Gde Satria Abirama  
NIM : 2115101056  
Prodi : Ilmu Komputer

#### Lesson Objectives:

- Identify scene components
- Create and save a new project
- Add an object to a scene
- Communicate the value of saving multiple versions of a scene
- Code a simple programming instruction
- Use the copy and undo command
- Understand the value of testing and debugging

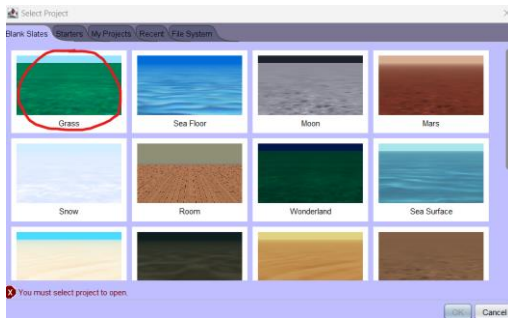
#### Vocabulary:

Identify the vocabulary word for each definition below.

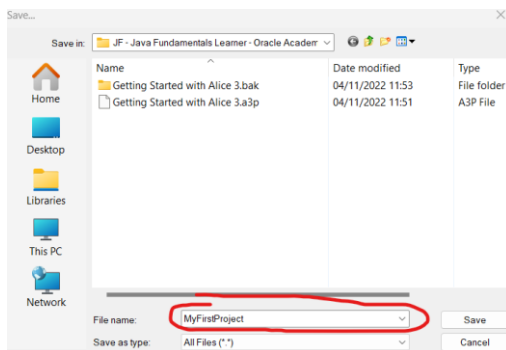
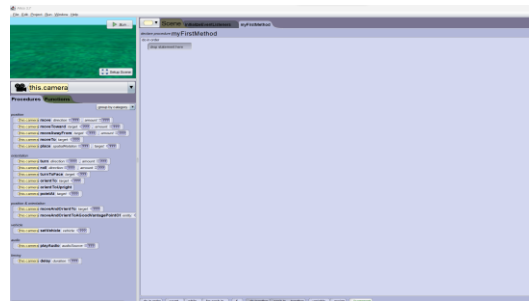
	The place where you add the programming instructions to program your animation.
	When something is broken or does not work as intended in a software program.
	The first scene of an animation where you select the background template and position the objects.
	The process of finding bugs in a software program.
	Contains the instructions that define the appearance and movement of an object.
	A collection of three-dimensional objects that you can insert into the scene.
	A value that is used by the method to perform an action.
	What an object is called after it is added to the scene.
	The place where you can select objects from the gallery to add to the scene, position objects, and edit an object's properties.
	The background of the scene that provides the sky, ground and light.
	A piece of program code that defines how the object should execute a task.

## Try It/Solve It:

1. Create and save a new project.
  - a. Launch Alice 3.
  - b. Select the grass template.
  - c. Save the file using a name such as "MyFirstProject" in a location you can find easily on your computer.
  - d. Ensure that the project name is displayed in the title bar.



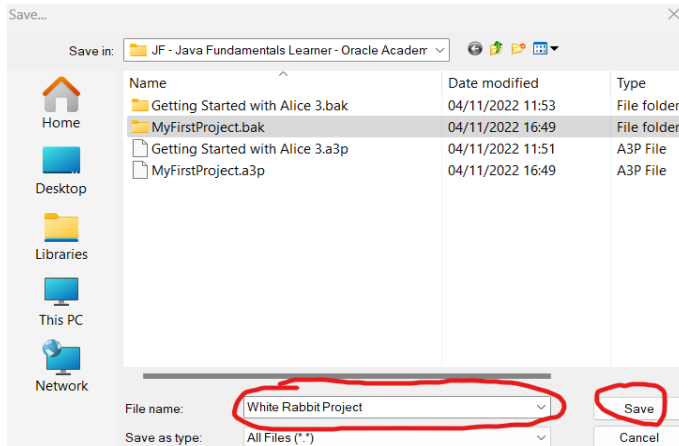
Grass template



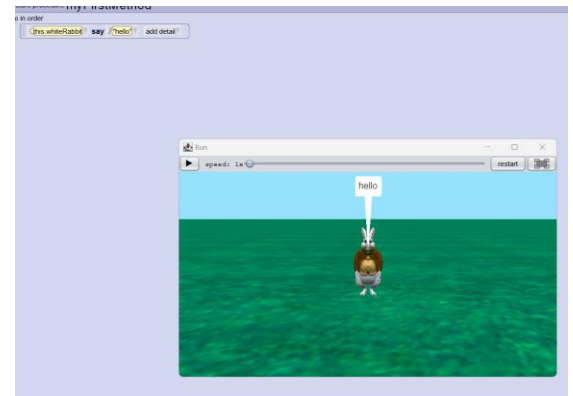
2. Add an object to a scene.
  - a. Launch Alice 3.
  - b. Open the MyFirstProject project file you saved in the previous activity.
  - c. Add an instance of a White Rabbit to the grass scene.
  - d. Save the project.



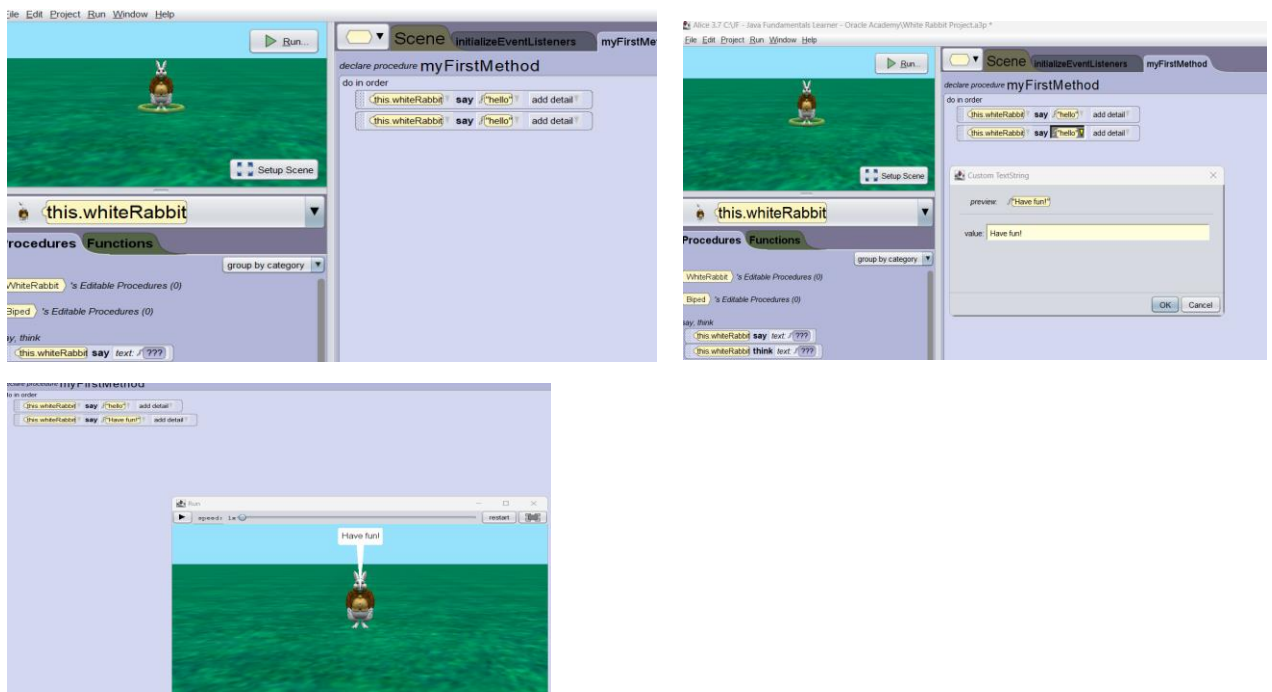
3. Save a new version of a project.
  - a. Open the MyFirstProject project you saved in the previous activity.
  - b. Save a new version of the project and name it "WhiteRabbitProject".  
Note: save several versions of your project so you can use the White Rabbit scene to create different animations.



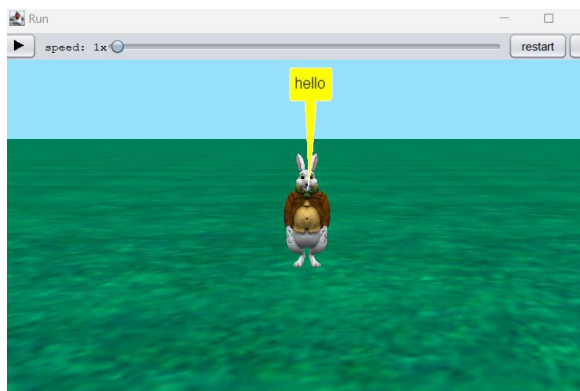
4. Code a programming instruction.
  - a. Open the WhiteRabbitProject you saved in the previous activity.
  - b. Add a programming instruction so that the White Rabbit says "hello".
  - c. Save the project.



5. Copy, edit, and undo programming instructions.
  - a. Open the WhiteRabbitProject you saved in the previous activity.
  - b. Using two different copying techniques make two copies of the programming instruction that makes the White Rabbit say "hello".
  - c. Edit one instance of the "hello" instruction so that the White Rabbit says "Have fun!"
  - d. Using Undo, delete the extra instance of the "hello" instruction.
  - e. Save your project.



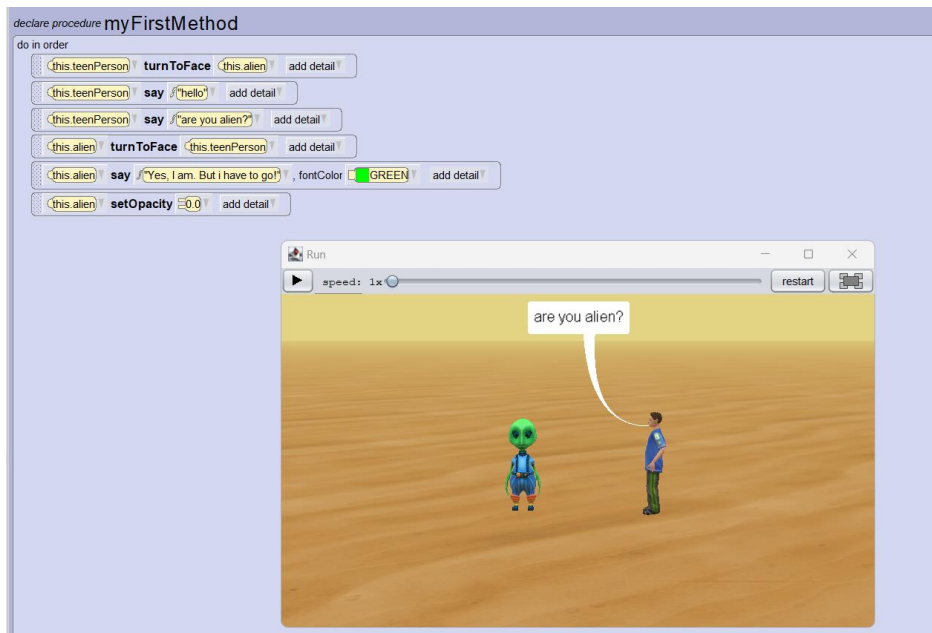
6. Test and debug the animation.
  - a. Open the WhiteRabbitProject you saved in the previous activity.
  - b. Test, edit, and debug your animation to adjust the timing of the "hello" and "Have fun!" instructions. Edit the code so there is enough time to read the text in the speech bubbles.
  - c. Save your project.



## Optional Activities:

Complete the following optional practice activities below to continue practicing the concepts you learned in this lesson.

1. Set up a scene with the Mars template, one alien object, and one person object. Add the following programming instructions to the Code editor.
  - a. The person turns to face the alien. Hint: Use the turnToFace procedure. Select the alien as the target object.
  - b. The person asks the alien, "Are you an alien?"
  - c. The alien turns to face the person.
  - d. The alien says "Yes, I am. But I have to go!"
  - e. The alien disappears. Hint: Select the alien from the Instance menu in the Code editor. Then, drag the setOpacity procedure into the Code editor. Select 0.0 for the argument value. This will make the alien disappear.



2. Set up a scene with the Wonderland template, one madHatter object, and one mushroom. Add the following programming instructions to the Code editor.
  - a. The Mad Hatter re-positions to stand in front of the mushroom. Hint: Use the place procedure with the argument value "in front of" and the target object as the mushroom.
  - b. The Mad Hatter says "If I think hard enough, I can make myself bigger."
  - c. The Mad Hatter becomes two times larger. Hint: Use the resize procedure. Select 2.0 for the argument value.
  - d. The Mad Hatter says "Oh no! Now I am too big!"

