

## 3.6: First Person Computer Games

### *Projections of 3-D Objects*

---

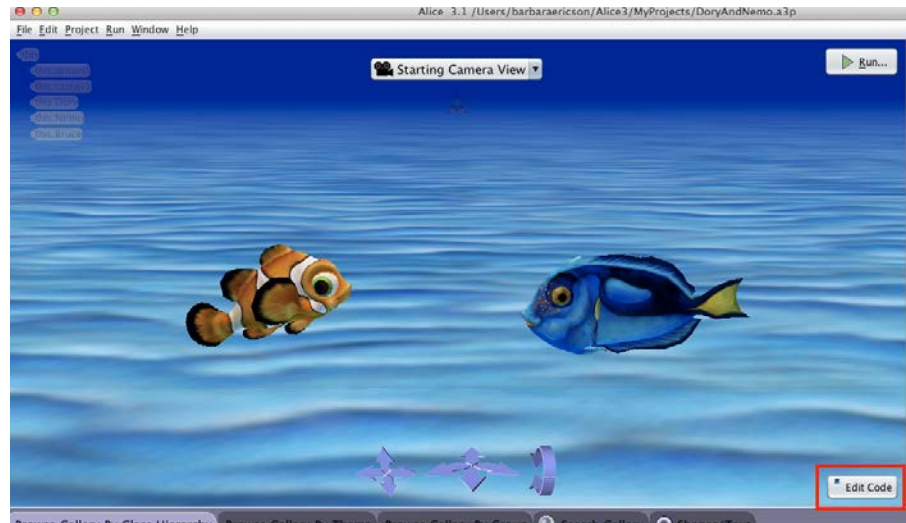
Alice is an educational software program that uses a 3-D environment to teach students programming.



If you have not done so already, please download and install Alice 3.1 on your device.  
Download: [http://www.alice.org/index.php?page=downloads/download\\_alice3.1](http://www.alice.org/index.php?page=downloads/download_alice3.1)

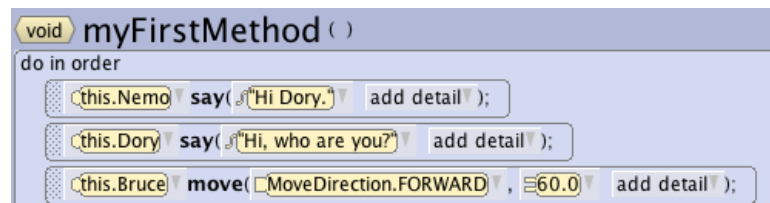
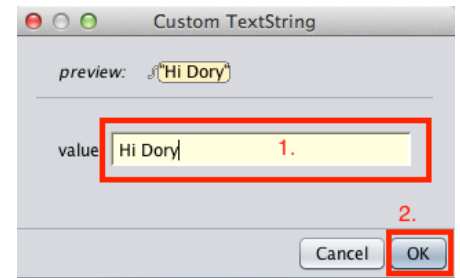
When you first start Alice 3 it will ask you to pick a template for your 3-D environment. There are lots of templates that you can use such as *GRASS*, *ROOM*, *MARS*.

1. Select the **Sea\_Floor** as your background scene, and click on the **Setup Scene** button in the upper left window. From the Gallery By Class Hierarchy (bottom window), select **Swimmer classes**, then **Fish classes**, and **new Blue Tang ( )**. Name your fish Dory and click **OK**.
  - a. Click on Dory, and from the handle style buttons in the top right corner of the screen, select translation. By selecting the arrows on the dolphin, we can move it to different locations in the screen. Move the dolphin left and right, up and down, forward and backward. Then, move it so that its coordinates are (0, 0, 0).
  - b. Use a matrix to describe each of the movements of the dolphin from its location at the origin.
    - i. Move 2 units right.
    - ii. Move 4 units down.
    - iii. Move 3 units forward.
  - c. Translate Dory to a position where you want it to start in the scene.

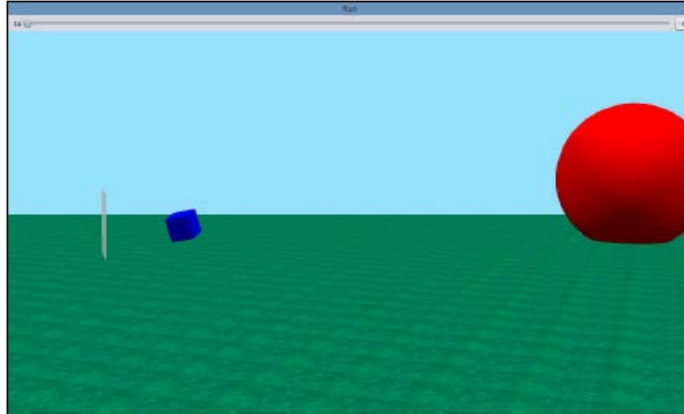


2. Create a clownfish object named "Nemo." You can resize Nemo by selecting the Resize handle style button and moving the mouse cursor.
  - a. Click on the rotation button from the handle style buttons, and practice rotating the dolphin about the three axes through its center. Use a matrix to represent the motion of the dolphin described. Assume the center of rotation of the dolphin is at the origin.
    - i. Rotation counterclockwise one full turn about the  $z$ -axis
    - ii. Rotation counterclockwise one half turn about the  $x$ -axis
  - b. Translate Nemo to a position where you want it to start in the scene.
  
3. Create a shark object named "Bruce" and position it where you want it by the end of the scene. You will move it away from the other fish to start by executing (running) a procedure.
  - a. To start the scene with Bruce far away from Dory and Nemo, you can run a procedure on the shark to move it backward.
    - i. Right click on *this.shark* in the left window and then select *procedures* then *this.shark move* ... then *MoveDirection.BACKWARD* and then *Custom Decimal Number* and type in **60** and hit **Enter**.
  - b. Select Edit Code from the main screen. Click on the down arrow next to *this.Bruce* and pick *this.Nemo* as the current object. This will show the procedures (actions) that Nemo knows how to do.

- i. Click on the left area of the block *this.Nemo say...* and drag it to the *myFirstMethod()* window on the right.
  - ii. Click on *Custom TextString*. This will let you type in a string which is a sequence of characters.
  - iii. Type in what you want Nemo to say, like "Hi Dory" and then click on *OK*.
- c. Dory Procedure: Click on the down arrow next to *this.Nemo* and pick *this.Dory* as the current object. This will show the procedures (actions) that Dory knows how to do.
  - i. Click on the left area of the block *this.Dory say...* and drag it to the *myFirstMethod()* window. Enter a custom string for Dory to say to Nemo.
- d. Bruce Procedure: Click on the down arrow next to *this.Dory* and pick *this.Bruce* as the current object. This will show the procedures (actions) that Bruce knows how to do.
  - i. Click on the left area of the block *this.Bruce move...* and drag it to the *myFirstMethod()* window. Select *MoveDirection.FORWARD* and then *Custom DecimalNumber*. Type in "60" to move Bruce forward 60. The code will look the following picture when you are done.

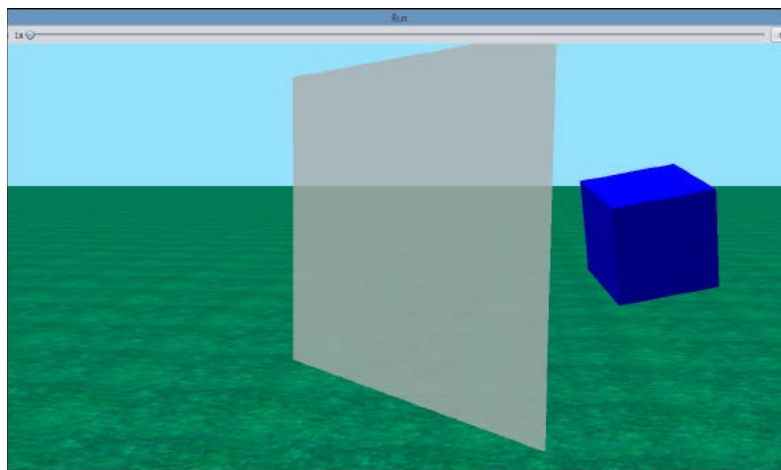


- e. Add other procedures
  - i. Example: Try to turn both Nemo and Dory around and have them flee from Bruce.



### III. Projection of 3-D Objects on a 2-D Plane

Open ALICE 3.1. Select a background and create 3 Geometric Shape objects in the scene: a cube, a larger sphere and a plane locate at  $x = 5$ . Arrange the objects so that the projection screen (plane) is in front of the cube, and the sphere is farther away in the distance.



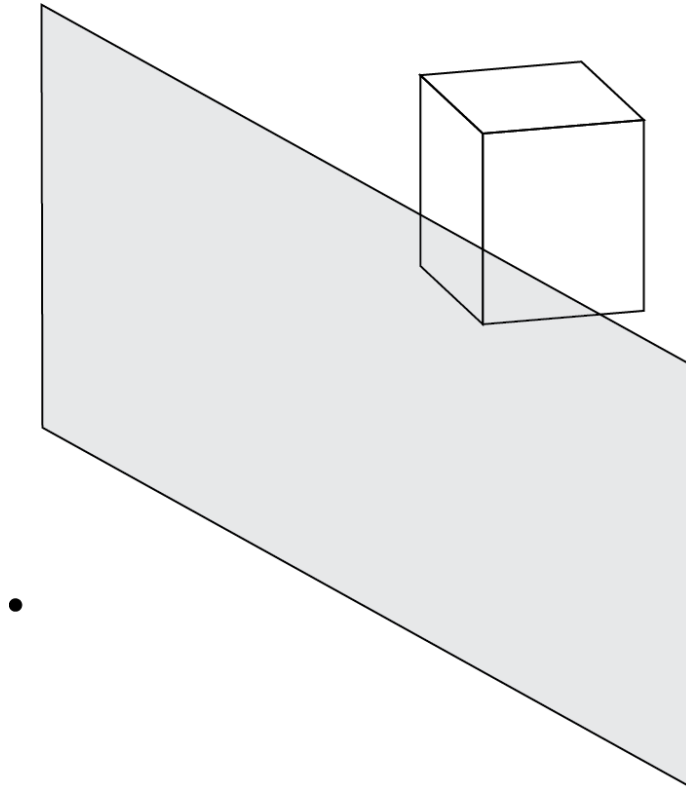
Imagine that the cube is a three-dimensional object that needs to be projected onto a screen represented by the gray rectangle.

4. Draw the image of the cube as it would be projected on the screen.

In this drawing task, the “eye” or the “camera” is the point, and the shaded figure is the “TV screen.”

The cube is in the 3-D universe of the computer game.

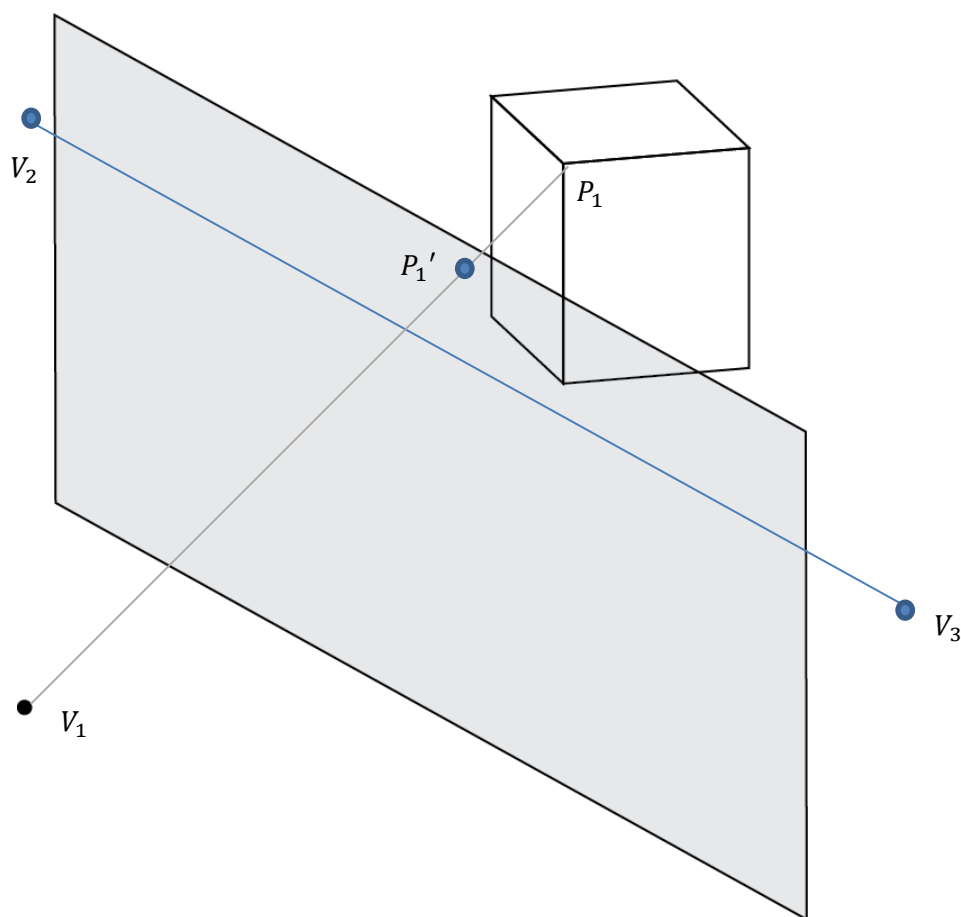
5. By using lines drawn from each vertex of the cube to the point, draw the image of the 3-D cube on the screen.



9. When three-dimensional objects are projected onto screens with finite dimensions, it often limits the field of view (FOV), or the angle the scene represents. This limiting effect can vary based on the size of the screen and position of the observer.
- Sketch a diagram that could be used to calculate a viewer's field of view  $\theta$  in relation to the horizontal width of the screen  $w$  and the distance the viewer is from the screen  $d$ .
  - Assume that a person is sitting directly in front of a television screen whose width is 48 inches at a distance of 8 feet from the screen. Use your diagram and right-triangle trigonometry to find the viewer's horizontal field of view  $\theta$ .
  - How far would a viewer need to be from the middle of a computer screen with a width of 15 inches to produce the same field of view as the person in front of the television?
  - Write a general statement about the relationship between screen size and field of view.

10. In this drawing task, the “eye” or the “camera” is the point, and the shaded figure is the “TV screen.” The cube is in the 3-D universe of the computer game.

By using lines drawn from each vertex of the cube to the point, draw the image of the 3-D cube on the screen. A horizon line and two additional vanishing points have been included to help you. The image point of the first vertex is shown.



11. Let's assume that the point  $V_1$  in our projection diagram is at the origin and the upper right vertex of the cube is located at  $\begin{pmatrix} 5 \\ 8 \\ 4 \end{pmatrix}$ . If our screen represents the plane  $y = 2$ , use matrix multiplication to determine the vector that represents the line of sight from the observer to the projected point on the screen. Explain your thinking.

12. Describe the points or projections:

- a. Describe the set of points  $(8t, 3t)$ , where  $t$  represents a real number.
- b. Project the point  $(8, 3)$  onto the line  $x = 1$ .
- c. Project the point  $(8, 3)$  onto the line  $x = 5$ .
- d. Project the point  $(-1, 4, 5)$  onto the plane  $y = 1$ .
- e. Project the point  $(9, 5, -8)$  onto the plane  $z = 3$ .