Unit I: Introduction: Installation and Alice Navigation

Chapter 5: Understanding the Alice Virtual World

In this chapter, you will learn how objects change position and move within the Alice virtual world. You will explore how an object's position, orientation, and sense of direction affect the way it moves. You will learn how to use actions such as **move**, **turn**, and **roll** to control the objects that make up a scene. Through these activities, you will develop a deeper understanding of how 3D objects behave and how their movements can be controlled to create effective animations.

Objectives

- Understand the basic principles of object motion and rotation;
- How Objects see the Virtual World:
- Understand that each object has a sense of direction;
- Be able to add code to move, turn and roll;
- Use a sequence of actions to create a dance routine, or other complex action

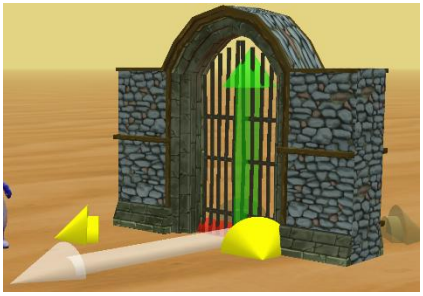
Orientation

As humans, our orientation is towards the earth: moving up is towards the sky, down is towards the ground. In the Alice virtual world, each object's orientation is based on itself, regardless of the direction of the sky and ground.

In setup scene view, if you select an object, you will see arrows at the base of the object. This point where the arrows show is called the objects **pivot point**. All movement is relative to that pivot point. Forward is in the direction of the white arrow and up is in the direction the top of the object is pointing (shown with a green arrow) and down is the opposite direction regardless of the direction of the sky or ground.

Creative Commons BY-NC- SA

All objects have their own pivot point and sense of direction based on that pivot point.



In the human world, if we move towards an object like the ground or a wall, we stop when we hit the ground or wall. In Alice, objects can go through walls, the ground, or intersect with other objects.

Changing Position

Objects in the Alice virtual world can change their position using the procedures **move**, **moveToward**, **moveAwayFrom**, **moveTo** and **place**. Using this picture of the cat and the gate as the starting point, we will illustrate each of these changes in position, always from the same starting position.

this.cheshireCat.move(MoveDirection.LEFT, 1.5)

The cat moves to his left and intersects with the castle gate. Notice that his orientation does not change. He is still facing the same direction.

this.cheshireCat.move(MoveDirection.BACKWARD, 1.5);

The cat moves backward (the opposite direction than he is facing.) His **orientation** (direction he is facing) does not change.

Creative Commons BY-NC- SA

this.cheshireCat.moveToward(this.castleGate, 1.5);

The cat's pivot point moves toward the castleGate's pivot point. His **orientation** does not change.

this.cheshireCat.moveAwayFrom(this.castleGate, 1.5);

The cat's pivot point moves in the opposite direction to the castleGate's pivot point.

this.cheshireCat.moveTo(this.castleGate);

The cat's pivot point moves to the same position as the castleGate's pivot point.

Notice that moving changes the **position**, but not the **orientation** of the object.

Alice Programming for the AI Age. Authored by Janet Joy using ChatGPT.
Creative Commons BY-NC- SA

Place: Spatial Relation

this.cheshireCat.place(SpatialRelation.IN_FRONT_OF, this.castleGate);

this.cheshireCat.place(SpatialRelation.BEHIND, this.castleGate);

this.cheshireCat.place(SpatialRelation.LEFT_OF, this.castleGate);

In the first image, the cat moves **in front of** the castle gate, and his body rotates slightly so that he is facing the same direction as the gate. In the second image he moves **behind** the gate. In the third image he moves to the **left of** the gate, again, he turns to face the same direction as the gate.



You can also move to a spatial relation of **below** (you don't see him because he is underground), or **above** (he is on top of the gate, or to the right. The **place** procedure changes both the **position** and **orientation**.

Orientation

An object in the Alice virtual world can change its orientation using **turn**, **roll**, **turnToFace**, **orientTo**, **orientToUpright**, and **pointAt**.

To illustrate each of these we will start with the cat in the position as shown on the right: facing a tiny bit to the right.

Turn

this.cheshireCat.turn(TurnDirection.LEFT, 0.25);

The cat turns left 0.25, a quarter of a turn. Turning 0.5 would make him face the opposite direction. Turning 1.0, would make him turn a full circle, and back to where he started. You would see him turning, though, so you would see the turn even though he ended up in the same position.

Creative Commons BY-NC- SA

this.cheshireCat.turn(TurnDirection.BACKWARD, 0.25);

The cat turns a quarter turn to lie on his back, but half underground, because the pivot point was touching the ground.

You can add arguments to change the **duration**, **animationStyle** and **asSeenBy. Experiment**!

Roll

this.cheshireCat.roll(RollDirection.RIGHT, 0.25);

Rolling tips the object in the direction specified (left or right): Here the cat rolls to the right 0.25, a quarter of a turn. Notice that he is also half underground, but his **position** did not change: his pivot point is in the same place.

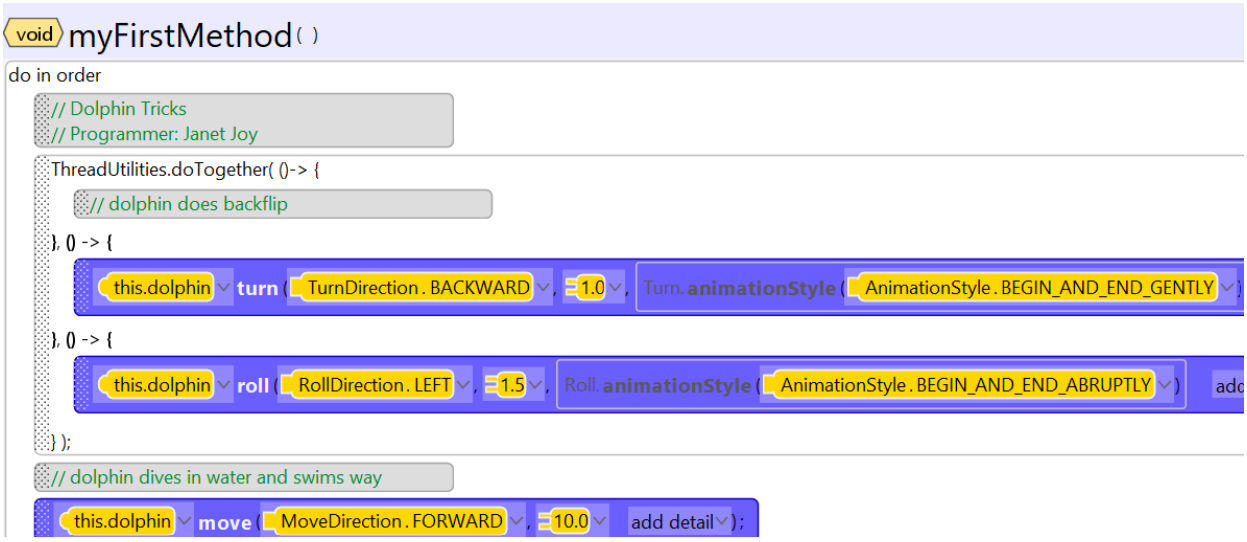
Active Learning

Describing how to create animation on a written page is a bit like enjoying a meal by looking at a picture. To really understand how **move**, **turn** and **roll** really work, you need to get into Alice and experiment. These procedures form the basis for all the animation you will create in Alice. Here are some code snippets for ideas on how to create some stories:

Creative Commons BY-NC- SA

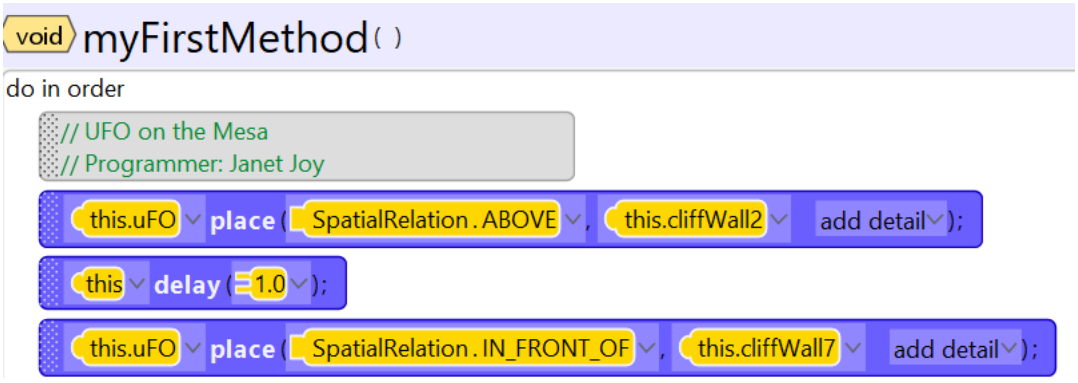
Dolphin Tricks

Start a new program with the ocean and add a dolphin. Make the dolphin do tricks using move, turn and roll.



UFO Explores the Mesa

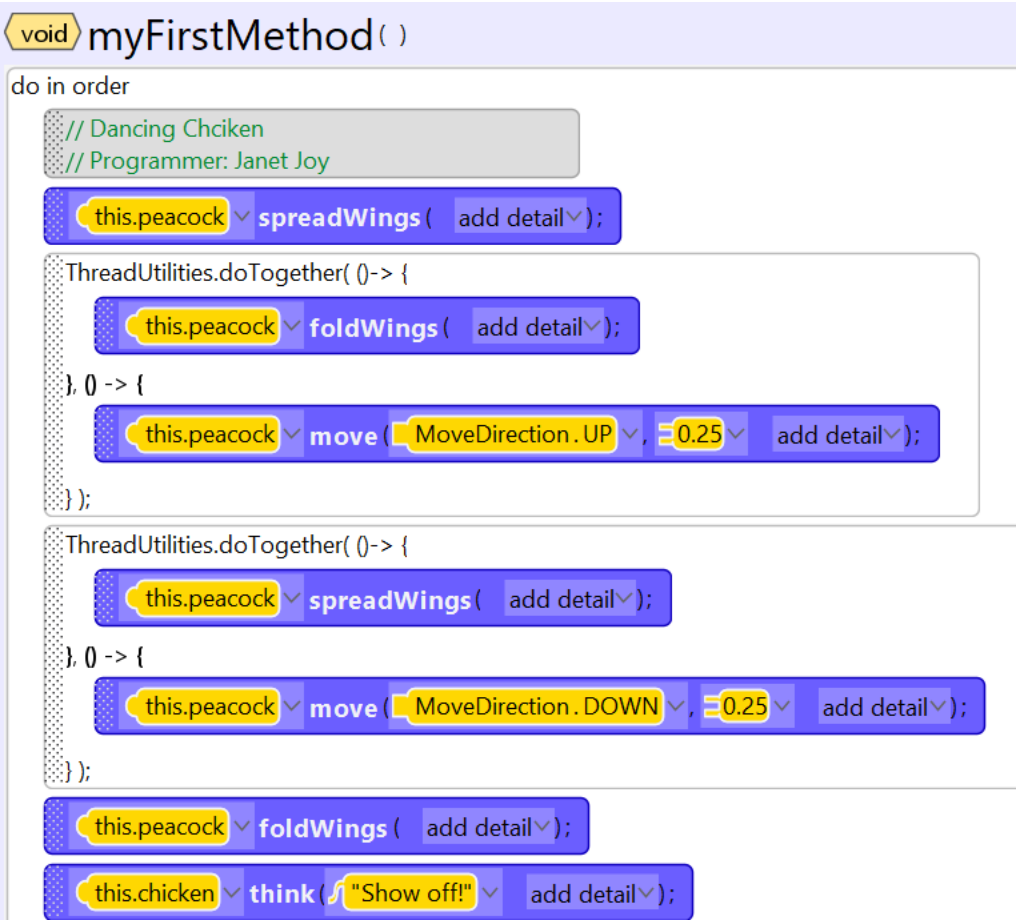
Start a new program with the Mesa minimum. Add a UFO. You can move it off to one side so that it makes an entrance. Fly from one cliff to another. Pause at each one. Add two people. After the UFO comes and goes, one person says, “Wow did you see that?” The second person says, “See what?”



Creative Commons BY-NC- SA

Dancing Birds

Start a new program. Add a chicken and a peacock. Make the peacock spread and fold his wings and move up and down. Add your own code to make the chicken do a little dance using turn and roll.



Coming Soon:

How do you open the gate? How do you bend the arms?

Creative Commons BY-NC- SA