

# Chapter 1

## Learning to Program With Alice

(Duke Book)

- \* Rather than typing a difficult, long and obscure computer language in the hopes of getting a calculation, with Alice you will be more like a **director of a play**, where on screen objects act out the script you have created.

### "Why should I even take a programming course?"

Some benefits of taking even 1 programming course are:

- o It is a way of learning a **new way to think**
  - o Increases **problem-solving skills** → Problem-solving is finding an answer to a question or figuring out how to perform a task. Computer programming is a pure, distilled form of problem-solving.
- \* Alice was originally developed as part of a research project in **virtual reality**.
  - \* Alice is different from traditional computer programming because you use "**natural English words**," like "move forward" or "turn right."
  - \* You can create 2 things with Alice:
    - o **Be a director of a 3D animated film**
    - o **Create an interactive video game**
  - \* After you have learned how to use Alice, you will understand all of the fundamental ideas involved in programming without all of the frustration of **the syntax of a computer language**.
  - \* **Computer Program** → is a set of instructions that tell the computer what to do.
  - \* Think of computer programming not only as a way of telling a computer what you want it to do, **but as a way of telling another human being what you want the computer to do**.
  - \* **Elegant** → a program is considered "elegant" if other human beings can easily understand and appreciate the intentions of the original programmer.

- \* One can create an "elegant" program by documentation → comments in the program, a web page for reference, or an accompanying written document like a user's manual that helps someone else understand what you were trying to do.

## All Computer Programs are Made From Very Simple Ideas:

### 1. A list of Instructions:

- Sequential programming → when you perform a task in a specific order

EXAMPLE: a recipe

- Beat eggs
- Mix in flour, sugar, and shortening
- Pour into a baking pan
- Bake at 375 degrees for 45 minutes

### 2. If's:

- Conditional execution → perform an action based on a condition.

EXAMPLE:

- If you have Dove chocolate, you must share with Mrs. Martin.

### 3. Repeating Behavior:

- Looping or Iteration → an action that is repeated for a "given" (numeric or conditional) time
- If a condition is true/false

EXAMPLE:

- For a numeric amount of times:
  - Stomp your feet 5 times
- Conditional:
  - As long as there are M&M's, keep eating them

### 4. Breaking things up into smaller pieces:

- Problem Decomposition; Stepwise Refinement; Top-Down Design; Algorithm → the process of doing a complicated task by breaking the task down into a list of smaller, simpler tasks. Once all of the simpler tasks are done, the complicated task is also accomplished.
- Reductionism → an ancient philosophical approach to the process of doing a complicated task by breaking the task down into a list of smaller, simpler tasks.

5. Compute a Result:

- Perform a **sequence of steps** to obtain a result that is an answer to a question.

EXAMPLE:

- o Look in the phone book and find the phone number for John Brown.
- This action actually asks a question of, "**What is John Browns phone number?**"
- **Function** → in computer programming, a "function" is just a question.
- **Calling a Function** → is asking a question so that you can compute a result.

\* Computer programming is really just using the previous 5 ideas **in various combinations**.

\* In reality, most computers really only understand about **100 different instructions**.

\* The millions of programs that run on computers use the same 100 instructions, but each in different orders and combinations. These different orders and combinations is what **causes the complexity of computer programming**.

- o EXAMPLE: Think of the game of chess
  - There are only 6 kinds of chess pieces
  - Each piece can only move in a simple pattern
  - What makes chess "hard" or "complex" is all of the possible combinations of moves.

\* **Learning how to think about arranging a sequence of instructions** to carry out a task-such as how to design a program-is probably the most valuable part of learning to program.

\* **Object - Oriented Programming (OOP)** → Most modern computer programming languages in which programs are organized into a set of methods that manipulate the properties of objects stored in a computer.

## \* Why did they name this software Alice?

- The Alice system is based on the use of objects. What makes Alice different from traditional OOP languages is that you can actually see the objects on the screen.
- The team that developed Alice named it so in honor of **Charles Lutwidge Dodson**. Dodson was an English mathematician that wrote under the name, **Lewis Carroll**. He wrote, "*Alice's Adventures in Wonderland*" and "*Through the Looking Glass*." Dodson and the developers of Alice both had a common belief: **the most important thing to doing something complex was to make things simple and fascinating to a learner!**

\* **Virtual World** → is a video game or simulation implemented in 3D.

\* All Alice virtual worlds begin with a scene of a **ground and sky**.

\* Some objects such as trees and houses provide a **setting**, while other objects such as people and animals play the **role of actors** in your script.

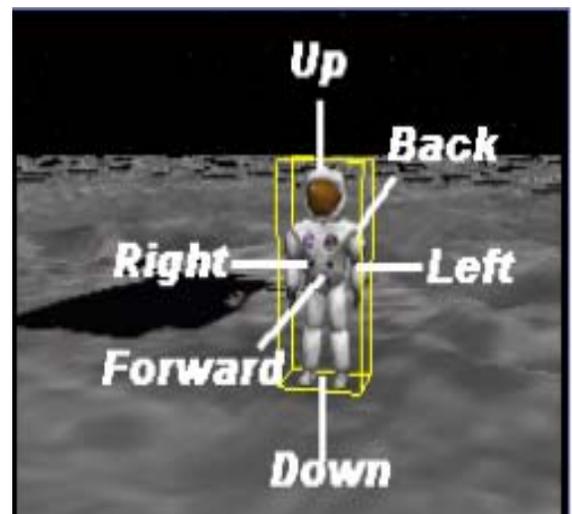
\* Alice has a large number of **3D models** → these models are like a blueprint used to design a house. The blueprint provides the size, color and other attributes of the model.

## 3 Dimensions and 6 Directions

\* Objects in Alice are **3 dimensional**.

\* Each object has **width, height, and depth**; these properties are in relation to the object, not in relation to the camera's view.

- **Height** → vertical, top to bottom
- **Width** → horizontal, left to right
- **Depth** → front to back, forward and backward



- \* Each object in Alice has what is known as **6 Degrees of Freedom** → ways it can move around in the world; possible directions of motion.
  - Each object knows the 6 directions of movement in relation to itself.
- \* **Orientation** → the 6 degrees of freedom in relation to the object, not the camera.
- \* **Bounding Box** → is a yellow box that is displayed when you mouse-click on an object.  
(See picture above)

## **Center of an Object**

- \* Each object in Alice has a unique **center** → it is not based on a calculation, but rather determined by the **graphic artist** when they first created the 3D model.
- \* The center point provides a reference for a **pivot** or **spin** type of movement, so not all "centers" for an object are at the center of mass.
- \* Objects that sit or stand have their center located at the **bottom of the bounding box**; for people it would be between their feet.
- \* Objects that are held also do not have their centers at the center of mass. This is so that when you rotate the object, it will **swing, or rotate**, about that point.



## **Distance**

- \* **Distance** → is measured from one object's center to another object.

## **Position**

- \* **Position** → is the point used from the center of an object within a world.

- \* Alice automatically puts the center of the ground at the center of the world at position **(0,0,0)**.

Exercise:

- o Select a template to start a world
- o In the "Object Tree," select the "ground" object
- o In the "Details Area," select the "properties" tab
- o If you look at the "pointOfView" you will see the coordinates, or position, as (0,0,0), which is the center of the world.

- \* Any object in the world is located relative to **the center of the world**.

## **Animation**

- \* **Animation** → is a fantasy of vision, an illusion.
- \* In Alice, you move objects about creating an illusion of movement, and Alice **renders**, or creates; the animation.

## **3D Text**

- \* In "Scene Editor" mode click on "Create 3D Text" in the Local Gallery.
- \* A text dialog box pops up and lets you choose **font**, **bold**, **italics** and a box to type in the text you want.
- \* Once you click "OK" **the text is entered into the world as an object** and is also displayed in the Object tree.
- \* To change a text object:
  - o Click the object in the object tree
  - o Click the properties tab in the Details Area
  - o Click the text and then a pop-up box will appear and you can change the text.
- \* Modifying a String in the text object **does not modify the name of the object**.

## Billboards

- \* You can create **flat 2D images** in any paint tool program and then import them into Alice. **Billboards** → The flat images you created elsewhere and imported into Alice.
- \* The images must be saved as either; **.gif**, **.jpg**, or **.tif**
- \* Steps to import an image into Alice:
  - Create an image using a paint tool program
  - Save the image with any of the 3 extensions listed above
  - Open Alice
  - Go to File/Make Billboard
  - Navigate to wherever you saved your image
  - Click "Import"
- \* One use of billboards is an "**instruction box**" providing information to the user about how to play a game or simulation.

**Instructions:**

**Up Arrow - - forward**

**Down Arrow - - backward**