



Programming with Alice

prepared by Bill Genereux
Assistant Professor of Computer and Digital Media Technology
Kansas State University at Salina

This overview assumes you have already downloaded and installed the free Alice software from Alice.org.

All computers can do the same basic functions. These are:

1. Process Input /Output
2. Store information
3. Do repetitive tasks
4. Make decisions
5. Do math computations

These things are what make computers powerful and useful. The Alice environment allows us to do each of these things. Let us examine each of these.

Initial Setup

- Create an Alice world with two of your favorite characters in it.
- Click **Play** to view your world
- Try to use the mouse to move your characters

Did they move?

Why do you think this is so?

Input / Output

“I/O” is the connection between the computer and the user. Examples of I/O devices include keyboard, mouse, monitor, and sound card.

Let’s allow users to interact with your Alice world using the mouse I/O device.

- In the **Events** panel, click **Create New Event** button and select “*Let the mouse move <objects>*”
Notice that an event has been added that will let the mouse move any object.
- Click **Play**
- Try to move your characters with the mouse

Did they move?

Why do you think this is so?

Events are things that happen in the Alice world that Alice can respond to. You can add many more events. For example, you could add events that control certain movements when a specific key is typed.

- Browse the other events found under **Create New Event**.

What are some of the other events that you found?

Store Information

An important part of computing is the ability to store information for later use and processing. Now we will make Alice store some information to use.

- Be sure you are in edit mode with your world not running
- Click “create variable” button at the center right of the screen
- Give it a name of “userInput”
- Select “other” and make sure “String” is showing in the listbox
- Click ok

You have now told Alice to reserve a space in memory to store information entered by the user. Notice the variable “userInput=default string” you just created appears in the variables area.

- Drag the “userInput” variable you made down to where it says “do nothing”.
- Set value to “default string”.

You should now see a line that says “**userInput** set value to **default string**”

- Select (click on) the “world” object at the top of the list of items in the upper left panel.
- In the lower left panel, click on the “functions” tab to see the world’s functions
- Scroll down to the **ask user** functions area.
- Select function “ask user for a string”
- Drag “ask user for a string” right to replace the “default string” of the userInput line
- Select “Enter a String:” then click OK 2x.

You should now see a line that says “**userInput** set value to **ask user for a string question...**”
Strings are a series of typed characters. Any non-numeric text you type is considered a string.

- Click **Play**

When your world runs, you should see a pop-up window asking for a String.

- Type some text into the box and click OK

The text you typed gets saved in memory under the “userInput” variable you created but we haven’t told Alice what to do with the saved text yet. So now let’s do something with that text.

- Select (click on) a character you created in the world preview window.
- Click on the “methods” tab found to the left on your screen (next to functions)
- Drag the “say” method below the userInput statement we just created, choose “Hello”
- Drag the userInput variable (the box that says userInput = default string) down to replace “Hello”

You should have a line that has your character say userInput

- Click **Play**
- Type sample text into the input box
- Your character should repeat what you typed.

The text you typed is saved into memory so it can be reused by your character

Do repetitive tasks

One of the best things a computer can do is simple, repetitive tasks that would be very boring for a person to do by hand. Computers love to do things again and again. The computer science term for doing the same thing over and over again is called doing a “**loop**.” A computer instruction is looped through again and again until something causes the loop to end. We have to be careful with loops. If we forget to cause the loop to end, we can have what is called an “**infinite loop**” or a loop that never ends.

Let us test Alice’s ability to do repetitive tasks with loops.

- Select one of your characters in the world preview window
- Make sure the **methods** tab is selected in the bottom left pane
- Select and drag the “turn” method below the last instruction we made. It should be the say userInput line.
- choose turn left 1 revolution
- Click on **play**

After you type some text in the input box, a character should repeat what you typed, after which a character should spin around one time. In my world, I had one character speak and the other one spin. You might do this as well, or have the same character do both. Now it is time to test the power of the **loop**.

- Find the “Loop” command box near the bottom of the screen, and drag it up and beneath the turn 1 revolution command you just added.
- Select 5 times to tell the loop to execute 5 times then quit
- Drag the “turn” command that is just above the loop to where it says “do nothing” inside of the loop.
- Click on **play**

Your character should now spin five times instead of once. Alice uses the loop structure to repeat the same command over and over again.

- Change the “Loop 5 times” command to say “Loop infinity times”
- Click on play
- Use your mouse to move the spinning character around.

Does the character stop spinning when the mouse moves it?

Why or why not?

How can you get the character to stop spinning?

Making decisions

Another very powerful thing computers do is make decisions based upon information that is provided to them. For example, when you step on the brakes of an automobile equipped with an anti-lock braking system, the auto’s computer will sense if the wheels are slipping while braking. If the wheels slip, then the ABS will engage, or else the braking will work normally.

Alice can also make decisions based upon information that is provided. Let us experiment with decision making in Alice.

- Click and drag the If/Else command at the bottom of the screen into your Loop you just created
- Select **true**
- Move the “turn” command that is already in the Loop but not yet in the If/Else block into the first “Do Nothing” area of the “If/Else” block.
- Click “world” object in the list at the upper left
- Click the “functions” tab in the lower left pane
- Drag the “a < b” function over to replace the “true” statement in the If/Else block.
- Set value for a and b to “1”.

You should now have an If/Else block with $1 < 1$ in it.

- In the world preview window select the character that turns
- Click on the “functions” tab to the left
- Find the “distance to” function box and drag it over to replace the first “1” of the “<” comparison in the If/Else block
- Set the “distance to” to check the distance to the other character you created

The “distance to” function calculates the distance from the current object to whatever other object you specify. This function is very useful in game programming because we often need to know the distance between two objects. My If/Else block now says If Chicken distance to bunny < 1.

- Click the **Play** button
- Move the two characters around to see how they interact.

Can you make a character turn around? What condition has to happen for the character to turn?

Why?

What are some other examples of a computer making a decision?

Math Calculations

Math calculations are what computers excel at doing. Computers were originally built to solve math problems. There are built in functions in Alice that help us to do math. Let us explore some of these.

- Click the “create new variable” button
- Type “a” as the variable name
- Choose “Number” as the variable type
- Repeat the first three steps, but make the variable’s name “b”
- Create another variable with the name “sum” and “Number” as the type

You should have two new variables, a and b.

- Drag variable “a” from the variable area down and place it above the Loop box we previously made
- Set the value to “2”
- Drag variable “b” from the variable area down and place it above the Loop box we previously made
- Set the value to “2”
- Drag variable “sum” down and place it above the Loop
- Click “set value”, click “expressions”, click “a”

There should now be a box that says “sum set value to a”

- In the “sum set value to a” box, click the triangle to the right of the “a”
- Click “math”, click “a +”, click “expressions”, click “b”

The box should now say “sum set value to (a + b)”

- Click on one of your characters in the world preview window
- Click the methods tab
- Drag a “say” box over so it is placed above the “Loop” we have previously made
- Select “hello”
- Click on the “world” in list at the upper left of the screen
- Click the “functions” tab at the left
- In the “string” section of the functions, drag “what as a string” over to replace the “hello” of the “say hello” box we just made
- Click “expressions”, click “sum”

Characters using “say” can only take string values, not numbers. It is confusing. It adds extra steps. It is just how it is. And that is why we had to add the “what as a string” box to convert the number “sum” to a string.

- Click the **Play** button

Your character should say the sum of 2 + 2. Because you have variables a and b, you can go back and change the “2” values to “ask user for a number” like we did earlier for user input. But that is for another day.

- Scroll down in the functions window to the “advanced math” section

What are some of the advanced math capabilities of Alice?

Feel free to experiment with some of these when you have the time.

Congratulations, you have now experienced all of the basic operations of computing using Alice!