COSC 123
Computer Creativity

Introduction to Alice

Dr. Ramon Lawrence
University of British Columbia Okanagan
ramon.lawrence@ubc.ca

Key Points

1) Learn the basic idea of programming and its key concepts

2) Experiment with the Alice environment and create worlds.

3) Learn about objects, classes, and methods.

4) Set and modify the properties of an object.

5) Create new objects including composite objects.

6) Learn how to animate many objects simultaneously.

Programming

What is programming?

- ***Programming*** is the process of constructing programs in order to instruct a computer on how to solve problems. It is the act of writing out the steps of an *algorithm*.

- A ***program*** is a sequence of simple computer instructions in some *language* which tell the computer the necessary steps to solve a problem or complete a task.

- A ***language*** is the structure and syntax used to communicate to the computer the tasks it is required to perform.

Why do we program?

Electronic devices require instructions to perform their function. Programming is our way of communicating those instructions.

Programs are written to do many things:

- Allow computes to communicate on the Internet
- Control airplanes, factories, cars, and electronics
- Send email, make a YouTube video, send a Twitter message
- Run businesses, handle inventory, trade stocks
- Any millions of others… Your ideas?

The ability to program in a digital society makes you a content ***producer*** rather than just a ***consumer***.

- Producers have the ability to impact others by creating and distributing their creations.

Programming and Creativity

Programming creates digital content. ***Creativity*** is at two levels:

- 1) Programming allows us to express our visions electronically for others to use.
- 2) The act of programming to realize the vision requires creativity and problem solving.

All programs that you use (Internet, email, Microsoft Office, YouTube, Google) are the result of programmer creativity.

- They had the vision to determine what they wanted to build and how that product can impact society.
- They had the ability to realize that vision by creating the necessary programs.

Programming Languages

Often the fun and creativity that programming allows gets lost in the details of the programming language.

The programming language is the format that we express our vision and approach to the computer. Each language has its own features, benefits, issues, and syntax.

The challenge is that to communicate with the computer we need to learn the language and associated tools and rules.

- Learning the language and tools takes practice and patience.

Analogy:

- Writers need to be fluent in the language they write.
- Artists need to know the basic techniques for painting/drawing.

Programming Languages Alice and Java

The two programming languages that we use are very different.

Alice is a graphical language designed to teach programming. All Alice programming is done graphically (very little typing). Alice programs are 3D stories and animations.

Java is a general purpose language used in industry and other programming courses. Java allows you to create anything and runs on most computes and cell phones.

Artistic comparison: Alice is like paint by numbers whereas Java is an open canvas for oil painting.

- Issue: “With great power comes great responsibility.”

Programming Concepts

There are some basic techniques common to all languages. **Learning the concepts is more important than the language.**

Key programming concepts:

- ***data variables*** – storing and using data in named locations
- ***expressions*** – computations on data to produce new results
- ***execution order*** – instructions are given in the correct order
- ***decisions*** – perform different actions based on a condition
- ***iteration*** – repeat a sequence of steps multiple times
- ***methods*** – groups of instructions with a particular purpose
- ***data structures*** – organizations to hold many data items
- ***code organization*** – larger programs need to structure the code so it can be easily created and modified (object-oriented)

Programming - Art or Science?

Is programming an art or a science?

- It is a science because algorithms and data structures can be analyzed for performance and chosen with respect to their relevance to a particular problem.
- It is an art or craft because skills of programmers vary widely, even with similar training, and the "best" solution to the problem is often open to debate.

In computer science, we teach you the "science" component.

- We want you to understand the choices you make and the reasons for them.
- However, students will all have different natural abilities and talents with respect to programming.
- If it is easy or natural for you, great! If not, then fall back on the science and the techniques we teach to help you!

Programming: Art or Science?

Question: What do you think programming is most like?

A) Art (creativity)

B) Science (experimentation)

C) Engineering (construction)

D) All of the above

E) Other or none of the above

Programming Practice

Like arts/sports, programming is a skill that requires practice.

- A musician practices scales to learn the basics and does the same song many times to master the techniques. Each song has its own skills and techniques used.
- A programmer practices by creating programs to perform tasks. The programs require understanding of the language and tools, and the solutions require composing techniques.

Key point: Like an artist, you must commit to practicing the craft. Programming skill comes from practice not memorization.

The labs are designed to give you some practice, but mastery will require more. Practicing is your studying for this course!

The 5 Basic Steps of Software Development

A programmer does ***NOT*** begin creating without a plan.

Developing a program should follow five basic steps:

- 1) **Specification** - Determine the scope of your problem and **what** you want your program to do.
- 2) **Design** - Determine the structures and algorithms necessary (**how**) to solve your problem at a high-level of abstraction.
- 3) **Implementation** - Start writing the code on the computer.
- 4) **Testing, Execution, and Debugging** - Test your program for various cases and fix any problems.
- 5) **Maintenance** - Over time, modify your program as necessary to handle new data or more complicated problems.

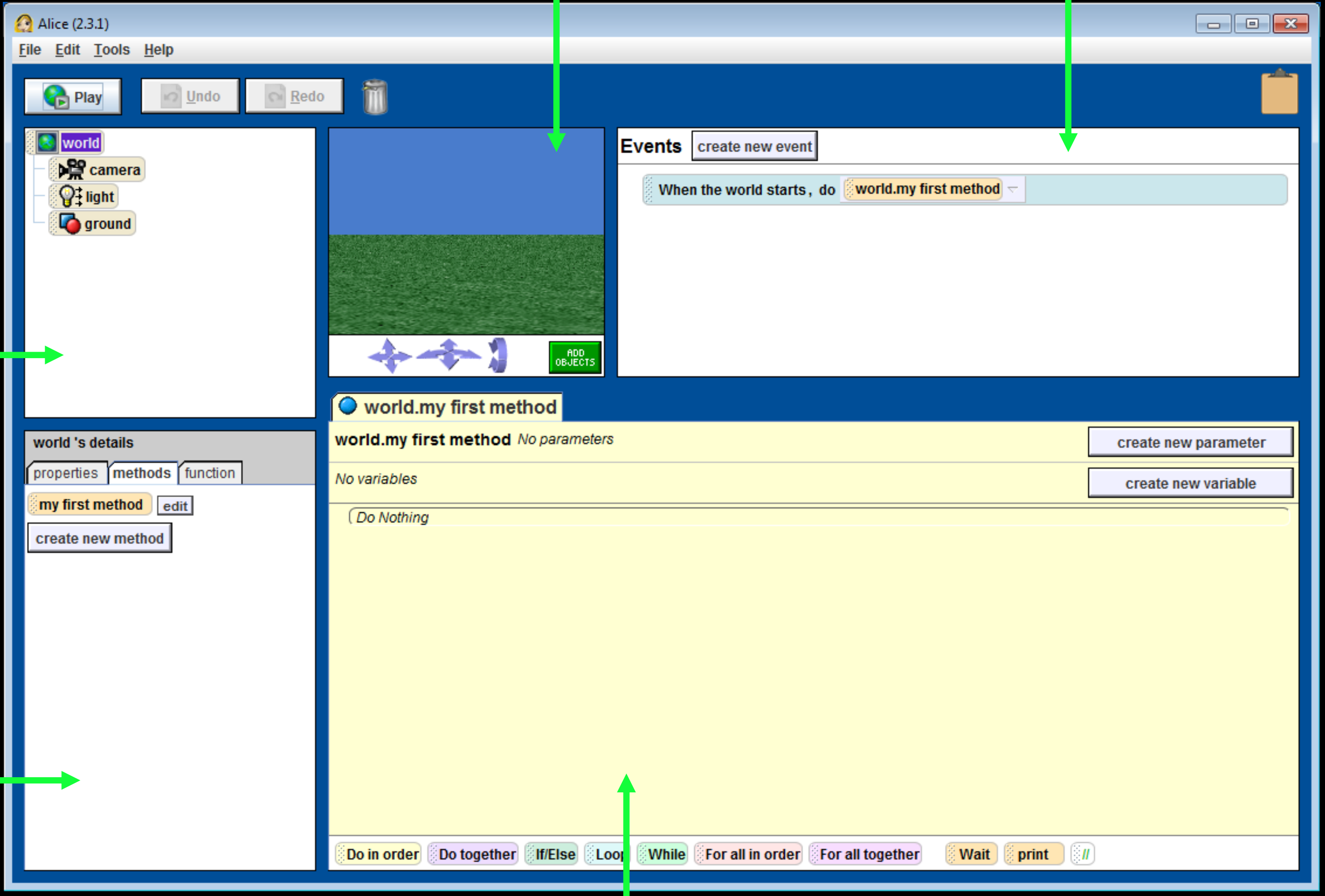
Programming with Alice

Alice is a 3D programming language designed to teach programming.

Alice allows you to compose stories which contain objects and scenery that interact.

Programming a story involves creating objects and scenery (the data), moving and interacting the objects (the instructions and methods), and everything in the story occurs according to a script (set of instructions). The script may involve decisions, loops, and events.

Alice Environment



Start Program

World View

Events Editor

Object Tree

Details Pane

Code editor

When the world starts, do world.my first method

world.my first method No parameters

No variables

Do Nothing

Objects

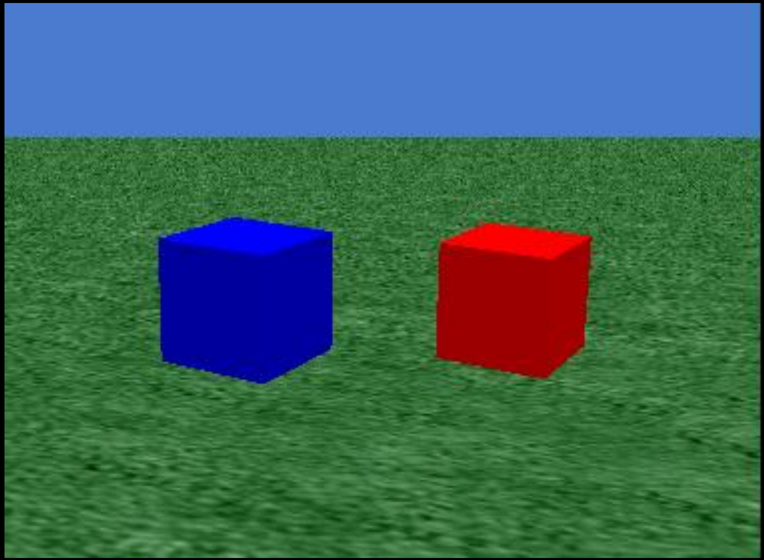
All ***objects*** in the Alice world are listed in the ***object tree***.

Objects are elements in the world that have a name.

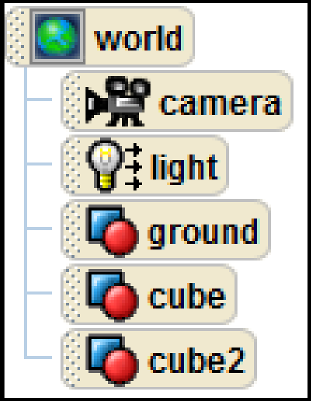
Two standard objects are the camera and the light source. Most worlds have a ground surface which is also an object.

The camera controls set the point of view.

World View

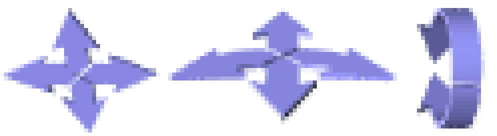


Object Tree



world
- camera
- light
- ground
- cube
- cube2

Camera Controls



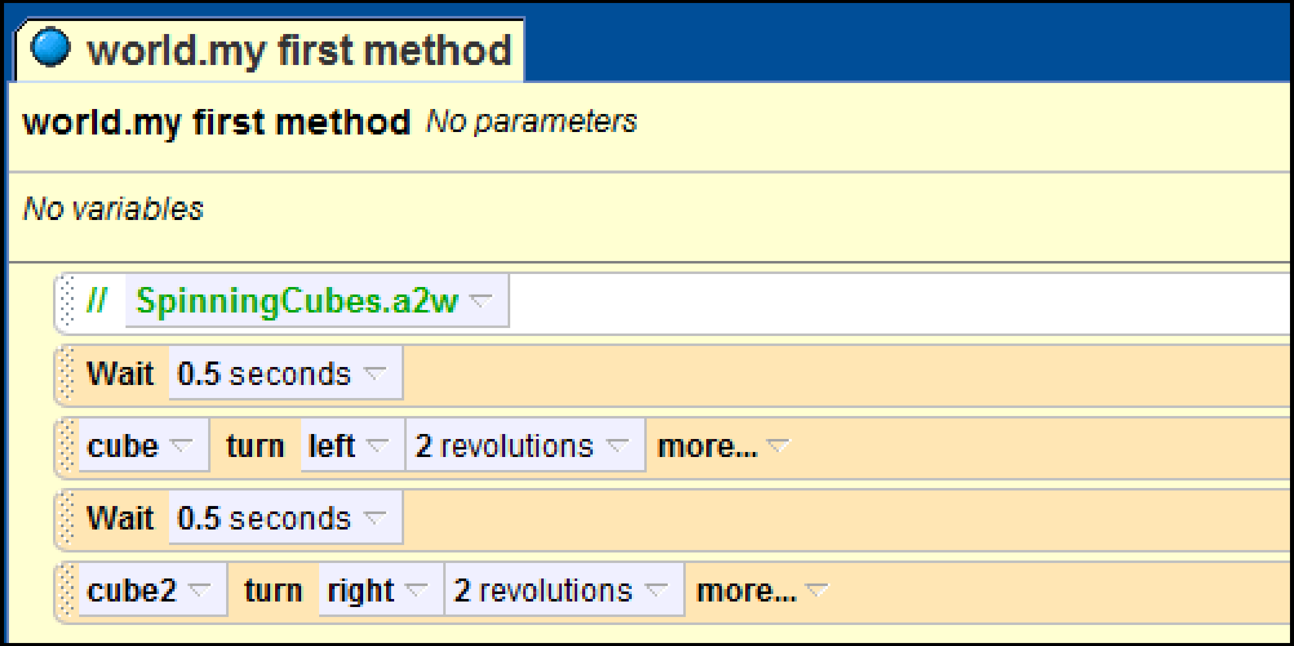
This world has five objects: camera, light, ground, cube, and cube2.

Methods

A ***method*** is a set of statements that can be ***called***.

- Methods perform actions and are associated with objects. The methods define an object's behavior (what it can do).

The ***world*** object has a method called ***my first method***. This method is called when the animation starts.



world.my first method

world.my first method *No parameters*

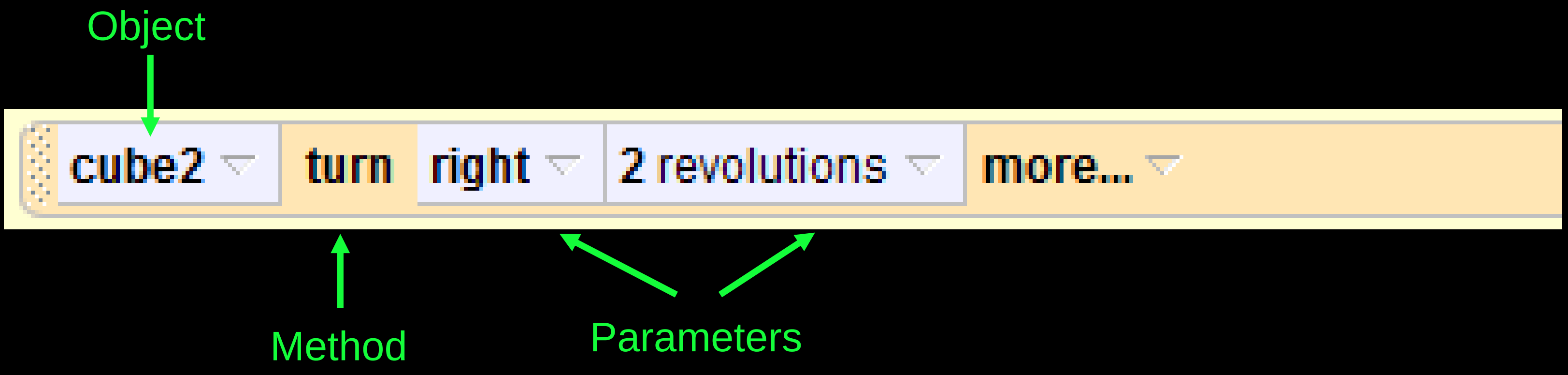
No variables

```
// SpinningCubes.a2w
Wait 0.5 seconds
cube turn left 2 revolutions more...
Wait 0.5 seconds
cube2 turn right 2 revolutions more...
```

Methods (2)

Calling a method is the act of running a method of an object.

Methods can accept ***parameters*** which provide input data for the method to use.



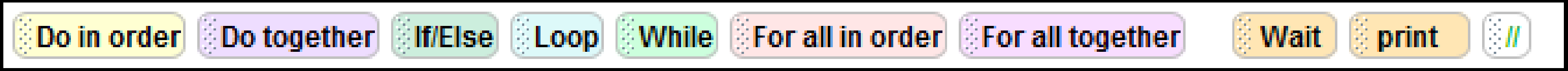
Object

cube2 turn right 2 revolutions more...

Method

Parameters

A method consists of a sequence of statements. Statements may be calls to other methods or statements to perform decisions, loops, or calculations. Statements in Alice:



Built-in Methods

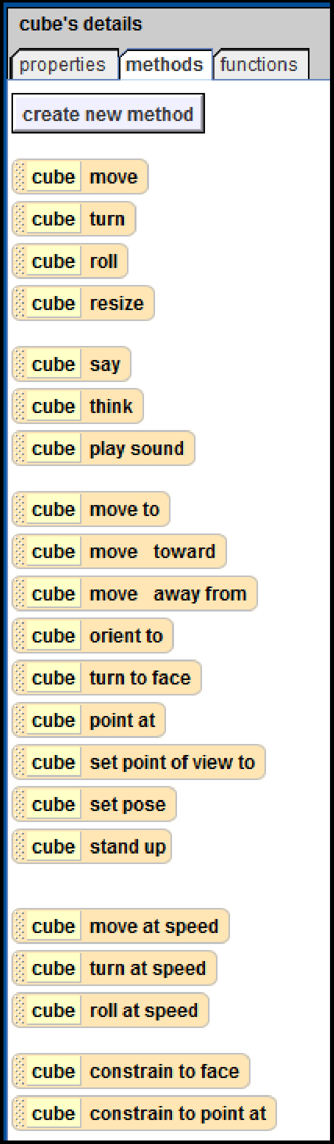
Built-in methods exist for almost all objects in Alice.

Other methods can be developed and added.

Some useful methods are:

- say
- think
- sound

Add a method call to your code by clicking on the object, selecting the methods tab, then dragging method into the code area.

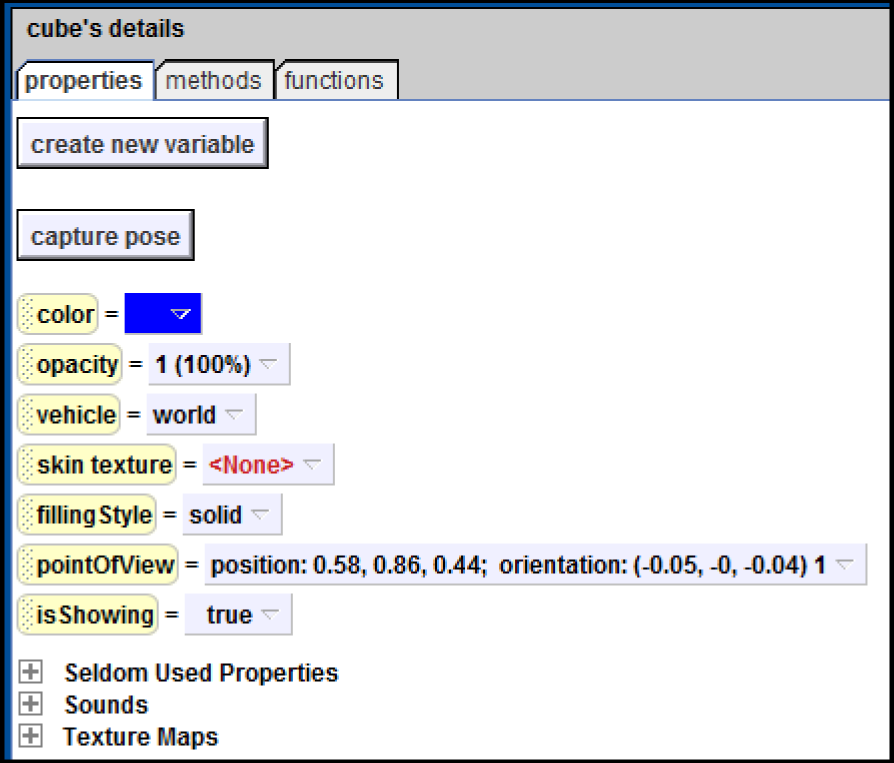


Properties

Properties describe an object’s state at any point in time.

- e.g. color, texture

The value of the property can be changed directly or during animation using a method call.



color = 🟦

opacity = 1 (100%)

vehicle = world

skin texture = <None>

fillingStyle = solid

pointOfView = position: 0.58, 0.86, 0.44; orientation: (-0.05, -0, -0.04) 1

isShowing = true

Classes

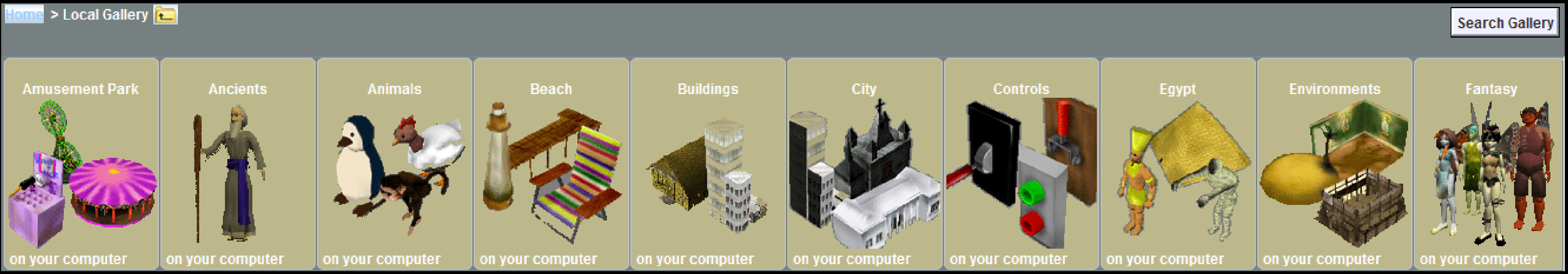
A ***class*** is a template for an object.

A class determines an object's methods and properties.

In Alice, classes are organized into galleries.

There are built-in (local) galleries and galleries on the Web.

Exploring the Local Gallery of Classes



Terminology Summary

An ***object*** is an instance of a class that has its own properties and methods. Properties and methods define what the object is and what it can do.

A ***class*** is a generic template (blueprint) for creating an object. All objects of a class have the same methods and properties (although the property values can be different).

A ***property*** is an attribute or feature of an object.

A ***method*** is a set of statements that performs an action.

A ***parameter*** is data passed into a method for it to use.

Objects

Question: Which of the following is not an object?

A) camera

B) world

C) wait

D) cube

Objects and Methods

Question: True or false: It is possible to have a method with no parameters.

A) true

B) false

Classes

Question: True or false: The two cube objects have the same class.

A) true

B) false

Classes and Objects

Question: True or false: Two objects that have the same class have the same methods.

A) true

B) false

Classes and Objects (2)

Question: True or false: Two objects that have the same class may have different values for their properties.

A) true

B) false

Demonstration Exercise Classes, Objects, Methods, Properties

Start Alice and open up `SpinningCubes.a2w`.

- Save a version of the file in your own directory on F:.

Items to try:

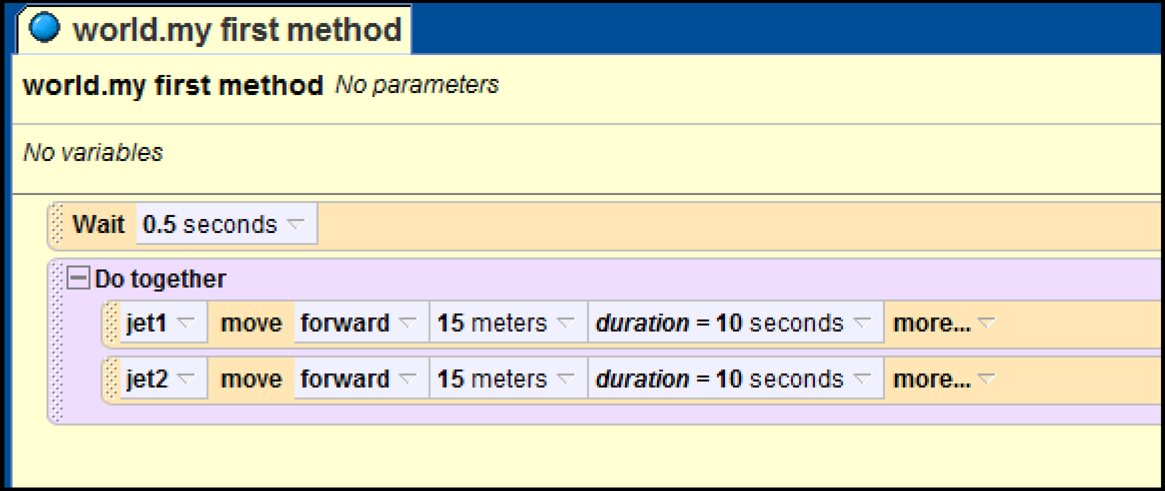
- Play the animation. Then close the animation.
- Try moving the camera using the camera controls.

Change the program to have these steps in order:

- 1) Make `cube` turn left once and `cube2` turn right 5 times.
- 2) Make `cube` go up 5 meters after it spins.
- 3) Change the color of `cube2` to yellow. (`properties` tab)
- 4) Call `resize` method on `cube` to make its ½ its size.
- 5) Add any object from the gallery to the world and make it move up 5 meters.

Do Together Statement

The ***Do Together*** statement allows several things to be done simultaneously.



world.my first method

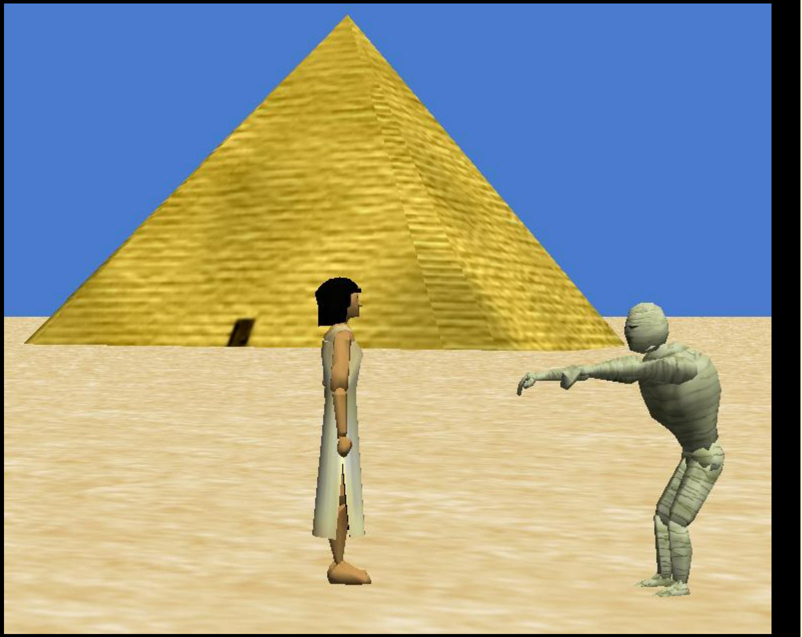
world.my first method *No parameters*

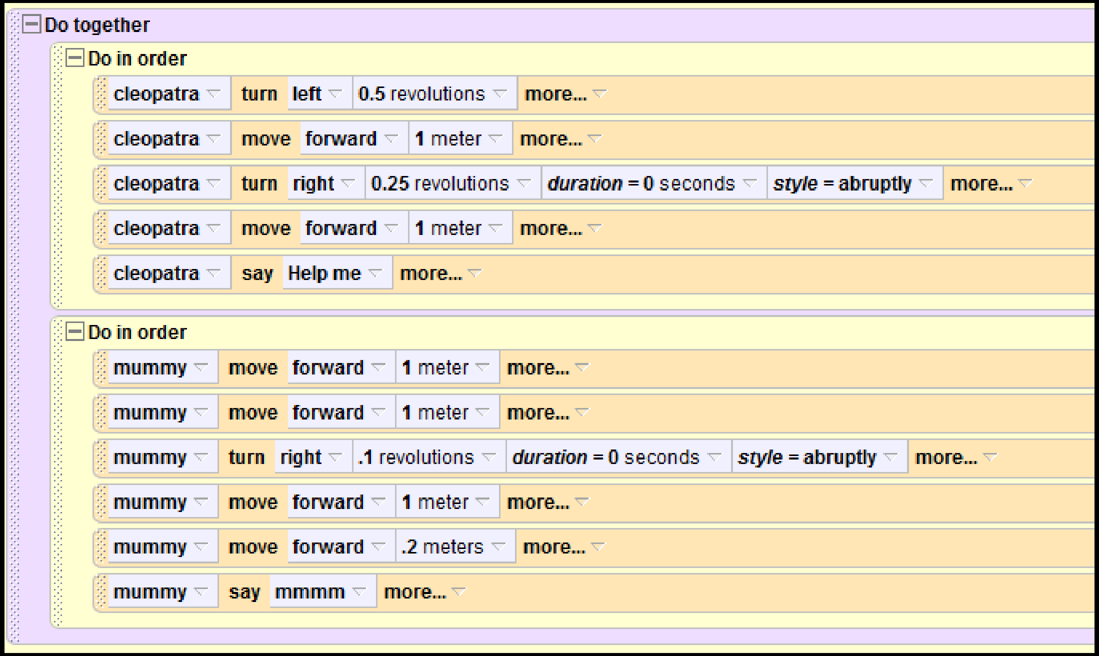
No variables

```
Wait 0.5 seconds
Do together
    jet1 move forward 15 meters duration = 10 seconds more...
    jet2 move forward 15 meters duration = 10 seconds more...
```

Do In Order Statement

The ***Do In Order*** statement forces the statements it contains to be executed in order.



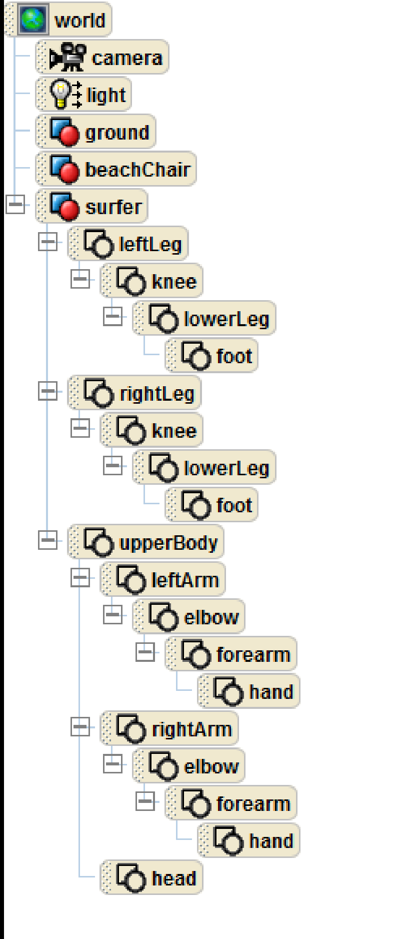


```
Do together
    Do in order
        cleopatra  turn  left  0.5 revolutions  more...
        cleopatra  move  forward  1 meter  more...
        cleopatra  turn  right  0.25 revolutions  duration = 0 seconds  style = abruptly  more...
        cleopatra  move  forward  1 meter  more...
        cleopatra  say  Help me  more...
    Do in order
        mummy  move  forward  1 meter  more...
        mummy  move  forward  1 meter  more...
        mummy  turn  right  .1 revolutions  duration = 0 seconds  style = abruptly  more...
        mummy  move  forward  1 meter  more...
        mummy  move  forward  .2 meters  more...
        mummy  say  mmmm  more...
```

Composite Objects

A ***composite object*** is an object that contains other objects.

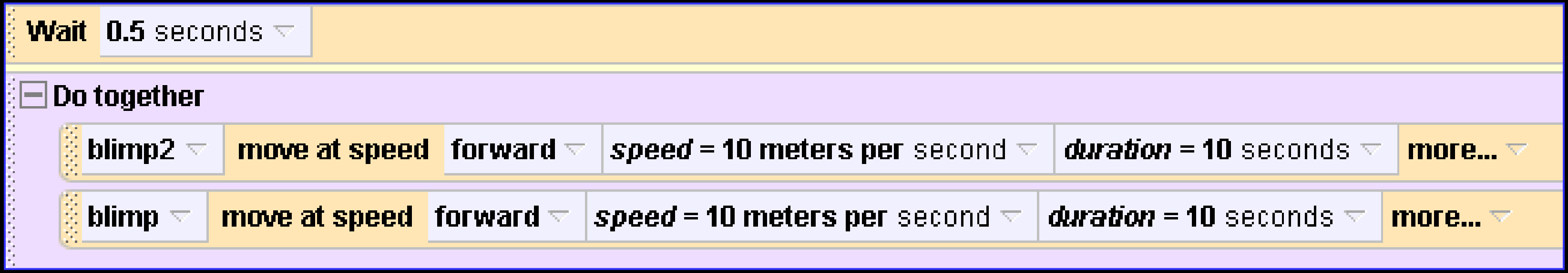
It is possible to control the whole object or any of its parts.



- world
  - camera
  - light
  - ground
  - beachChair
  - surfer
    - leftLeg
      - knee
        - lowerLeg
          - foot
    - rightLeg
      - knee
        - lowerLeg
          - foot
    - upperBody
      - leftArm
        - elbow
          - forearm
            - hand
      - rightArm
        - elbow
          - forearm
            - hand
      - head

Do Together and Do In Order

Question: What will this code do?



```
Wait 0.5 seconds
Do together
    blimp2 move at speed forward speed = 10 meters per second duration = 10 seconds more...
    blimp move at speed forward speed = 10 meters per second duration = 10 seconds more...
```

A) Move blimp then blimp2.

B) Move blimp2 then blimp.

C) Move blimp and blimp2 at the same time.

Demonstration Exercise Do Together and Do In Order

Start Alice and open up `SpinningCubes.a2w`.

Change the program to have these steps in order:

- 1) Make `cube` turn left once and `cube2` turn right once at the same time.
- 2) Make `cube` turn move up 5 meters and `cube2` move left 5 meters.

Demonstration Exercise Composite Object

Start Alice and open up **`SurferWave.a2w`**.

New ideas:

- The **`surfer`** is a composite object.
- To capture a pose, move the object into a certain position then under **`properties`** click **`capture pose button`**. Then to make the person go into that pose again use **`set pose`** method.

Change the program to have these steps in order:

- 1) Make the surfer say "**`Hello`**" while waving.
- 2) Make the surfer's arm go back to normal after he is done waving.
- 3) Using capture pose and set pose, make a pose with the arms spread out from the body parallel to the ground (looks like a T). Then put character in that pose and put him back again.
  - ⇨ Make sure to capture original standing pose.

Conclusion

Object-oriented programming uses:

- ***Objects*** – are instances of a class that have their own properties and methods.
- ***Classes*** – are generic templates (blueprints) for creating objects.
- ***Methods*** – contains statements that perform an action.
- ***Parameters*** – are data passed into a method.
- ***Properties*** – are attributes/features of objects.

Object-oriented programming involves defining objects and manipulating their properties and methods to perform useful actions.

Summary of Alice

In Alice:

- `Galleries` contain ***classes*** of objects.
- An ***object*** is created from a class when it is put into the world.
- Calling ***methods*** on objects make the objects do things.
- A ***property*** is a feature of an object such as its color.
- ***Composite objects*** contain other objects.
- ***Do Together*** makes actions occur simultaneously.
- ***Do In Order*** makes actions happen sequentially.