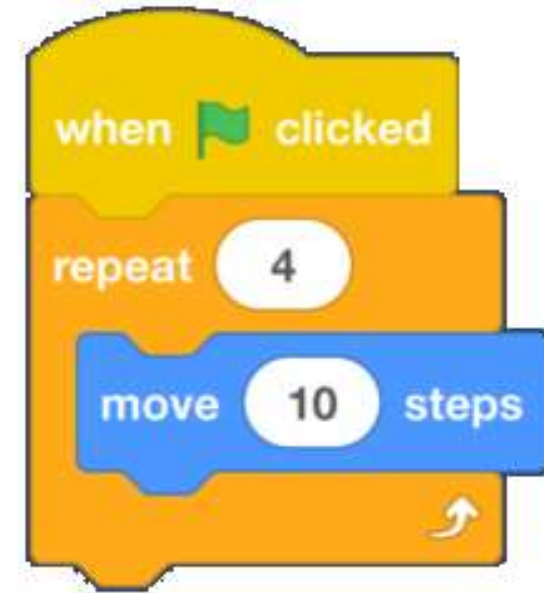


Drag & Drop Programming

COMP360



“Any sufficiently advanced technology is indistinguishable from magic.”

Arthur C. Clark

Remaining Schedule

	Wednesday, April 19 Drag & drop programming	Friday, April 21 Drag & drop programming
Monday, April 24 Concurrent Programming read chapter 13	Wednesday, April 26 Concurrent Programming	Friday, April 28 Concurrent Programming
Monday, May 1 Exam 3	Wednesday, May 3 final review	
Tuesday, May 9	Final Exam 10:30am – 12:30pm	

Friendly

- Many students are intimidated by the syntax of most programming languages
- Drag & Drop or Visual Programming languages attempt to avoid this problem
- Programmers do not have to type the program
- You never have a syntax error

Drag & Drop Languages

- Scratch
- Snap! or BYOB (**B**uild **Y**our **O**wn **B**locks)
- Alice
- MIT App Inventor
- Hopscotch
- Gamefroot

Scratch

- Scratch is a visual programming language, generally aimed at young programmers
- Scratch runs online through a browser
- Designed by Mitchel Resnick of the MIT Media Lab, Lifelong Kindergarten Group
- First available in 2005

Scratch Popularity

- The Scratch online community's slogan is "Imagine, Program, Share"
- The website receives over 125 million page views per month
- As of July 12, 2016, it had 12,561,189 registered members
- Over 15,700,000 projects and growing rapidly

Category		Notes		Category	Notes
	Motion	Moves sprites and changes angles and change X and Y values		Events	Contains event handlers placed on the top of each group of blocks
	Looks	Controls the visuals of the sprite; attach speech or thought bubble, change of background, enlarge or shrink, transparency, shade		Control	Conditional if-else statement, "forever", "repeat", and "stop"
	Sound	Plays audio files and programmable sequences		Sensing	Sprites can interact with the surroundings
	Pen	Draw controlling pen width, color, and shade. Allows for turtle graphics.		Operators	Mathematical operators, random number generator, and-or statements
	Data	Variable and List usage and assignment		More Blocks	Custom procedures (blocks) and external devices control

Try Scratch

- Go to <https://scratch.mit.edu/>
- Write a program to move the creature in a square of 50 units on a side

Alice

- A modern programming tool
 - 🎮 3-D graphics
 - 🎮 3-D models of objects
- Animation
 - 🎮 Objects can be made to move around the **virtual world** (a video game or simulation implemented in 3-D)
- Use the software from the textbook CD
- Can download latest version from the web
www.alice.org



Kinds of Animations

🌐 Two kinds of animations:

🎬 Movie

💡 Passive user watches the animation

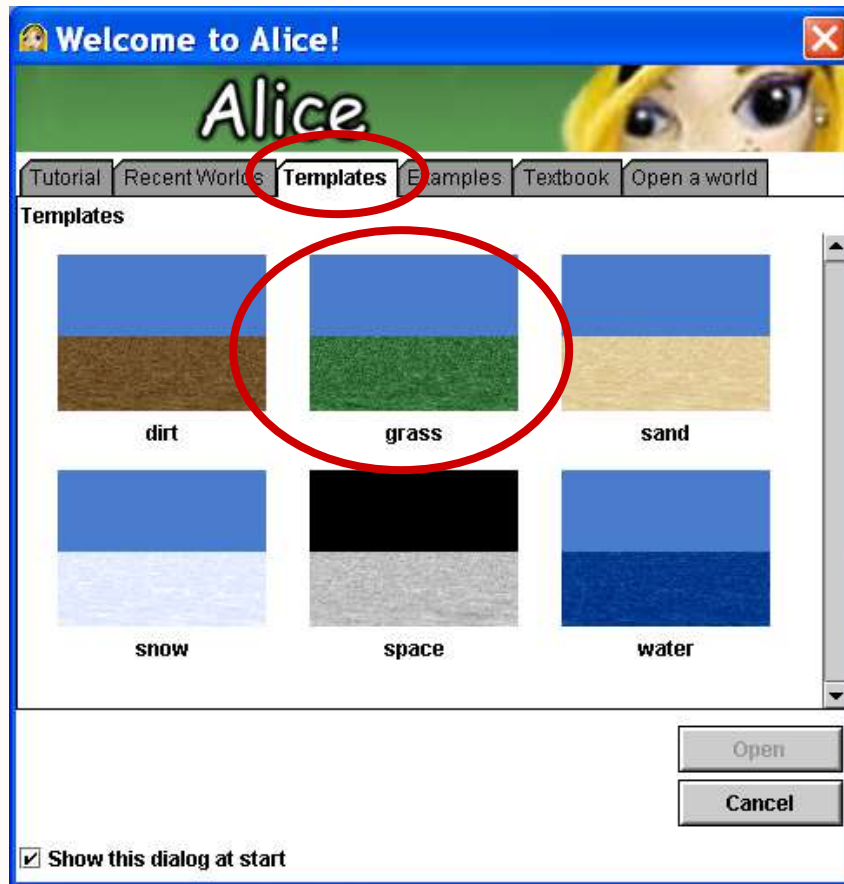
🎬 Interactive

💡 Active user clicks on mouse, types a key on keyboard ...

💡 Actions of user are called **events**



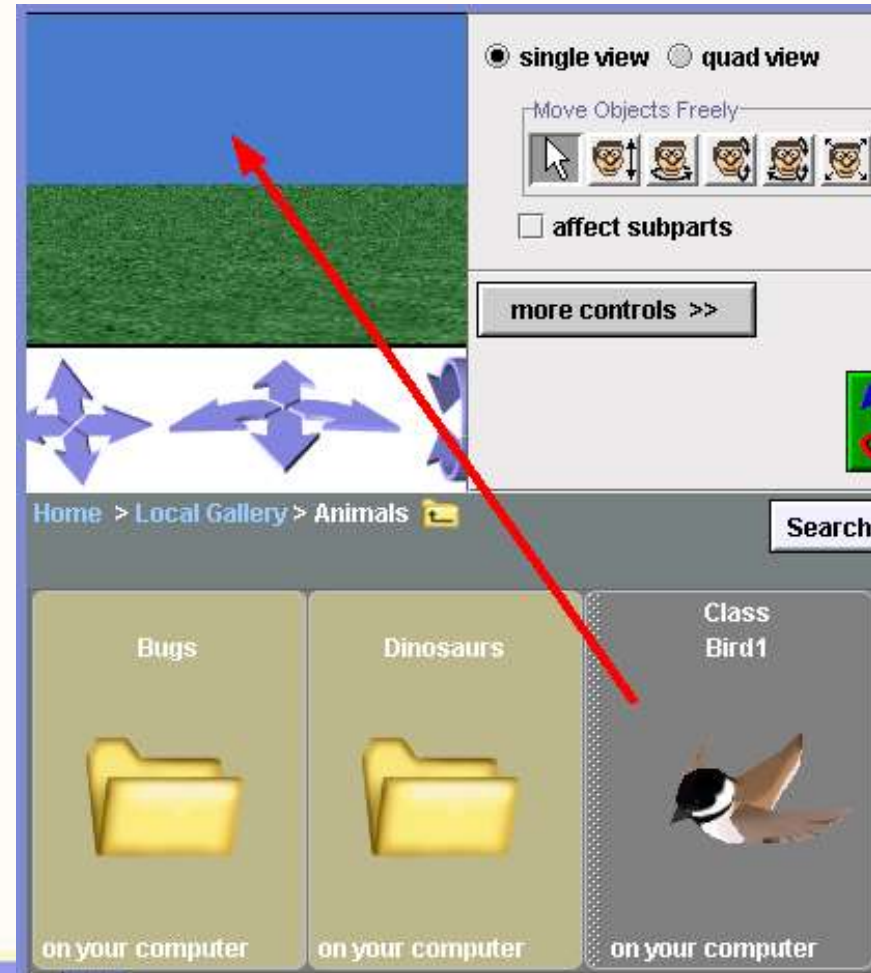
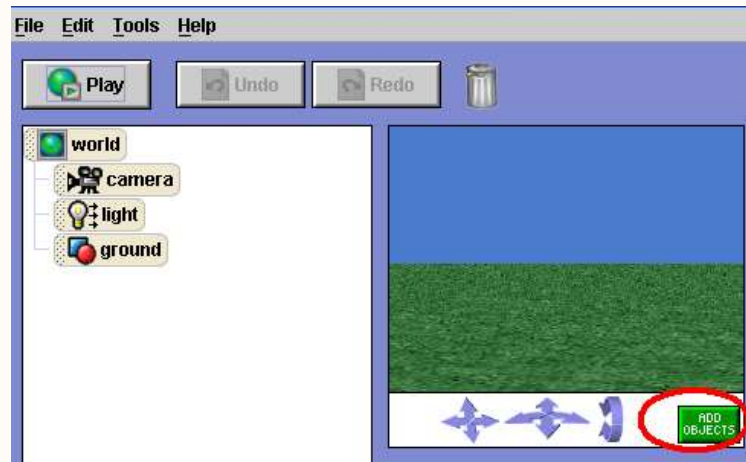
Example: A new world



- 🌐 Create a new world
 - 📁 File → New
 - 📁 Choose template



Example: Adding objects



Techniques and Tools

- **Mouse** is used to
 - ▶ approximately position objects in the scene
- **Camera Navigation** is used to
 - ▶ set the camera point of view
- **Drop-down menu methods** are used to
 - ▶ resize objects
 - ▶ more precisely position objects in the scene
- **Quadview** is used to
 - ▶ position one object relative to another object



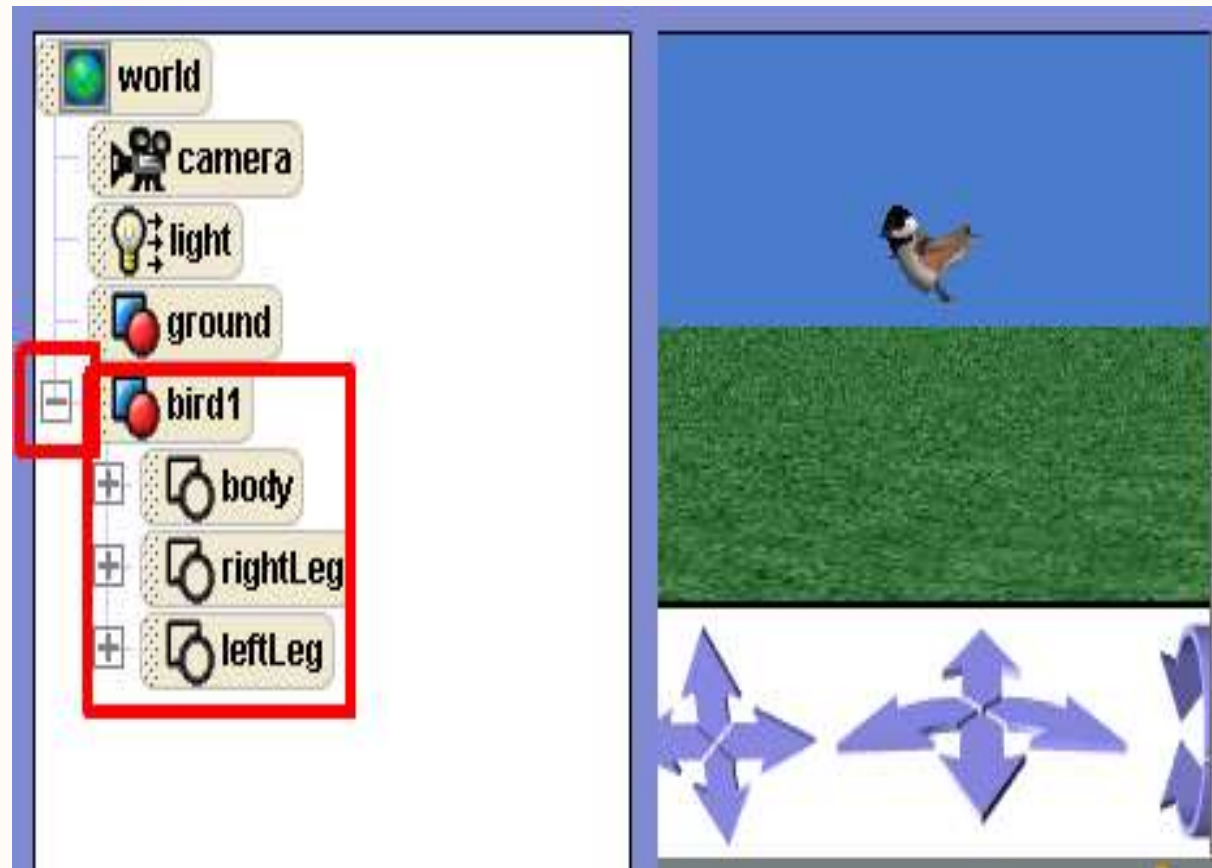
Objects

- 🌐 An "**object**" is
 - 🔧 any thing that can be identified as unique from other things
- 🌐 How is an object unique?
 - 🔧 has a name
 - 🔧 has properties:
 - 💡 width, height, color, location
 - 🔧 can perform actions (methods):
 - 💡 associated actions it can perform
 - 💡 tasks it can carry out



Object Parts

- Objects may be composed of parts



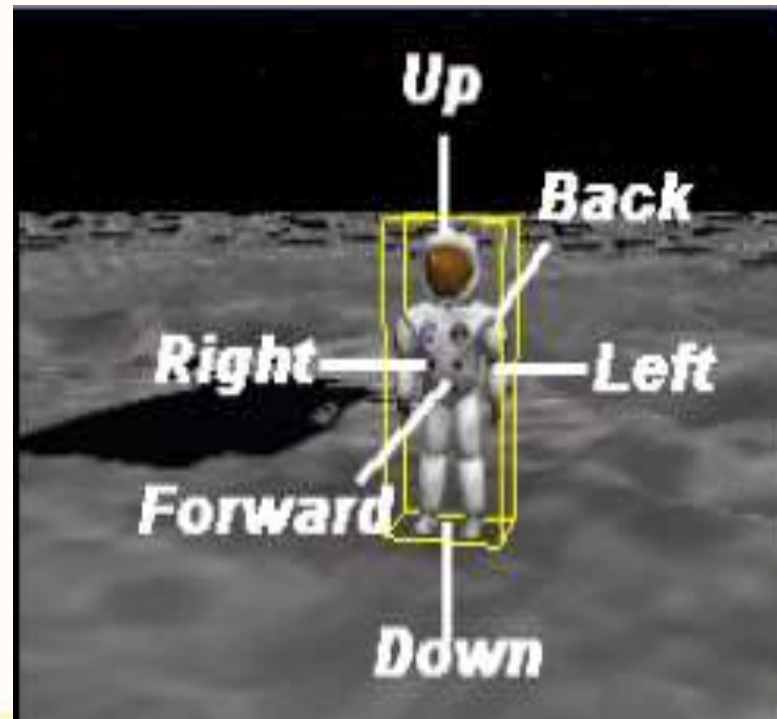
3 Dimensions, 6 Directions

🌐 A 3D object has

📏 3 dimensions

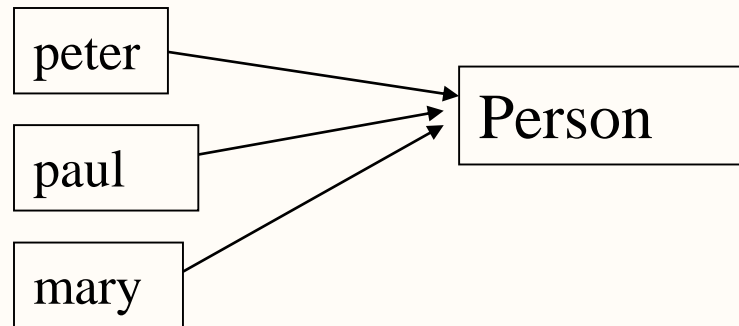
💡 height, width, depth

📏 6 degrees of freedom (directions of movement)

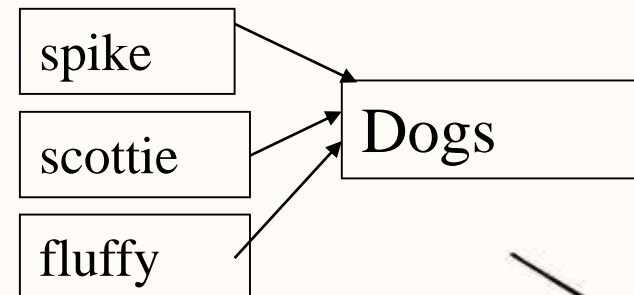


Class

- Objects are categorized into classes



- Each object is an **instance** of the class.



- All objects in a class have similar properties and generally can perform the same tasks.



World.my first method

World.my first method *No parameters*

create

No variables

creat

Do in order

// The snowman tries to catch the attention of the snowwoman by talking and blinking his eye

snowman turn to face snowwoman more...

snowman say Ahem... more...

Do together

snowwoman.head turn to face snowman more...

Do in order

snowman.head.leftEye move up 0.04 meters duration = 0.5 seconds more...

snowman.head.leftEye move down 0.04 meters duration = 0.5 seconds more...

Do in order

snowman.head.rightEye move up 0.04 meters duration = 0.5 seconds more...

snowman.head.rightEye move down 0.04 meters duration = 0.5 seconds more...

// The snowwoman blushes and turns away

Do together

snowwoman.head set color to more...

snowwoman.head turn to face snowwoman2 more...

snowwoman.head set color to more...

snowman.head turn forward 0.15 revolutions more...

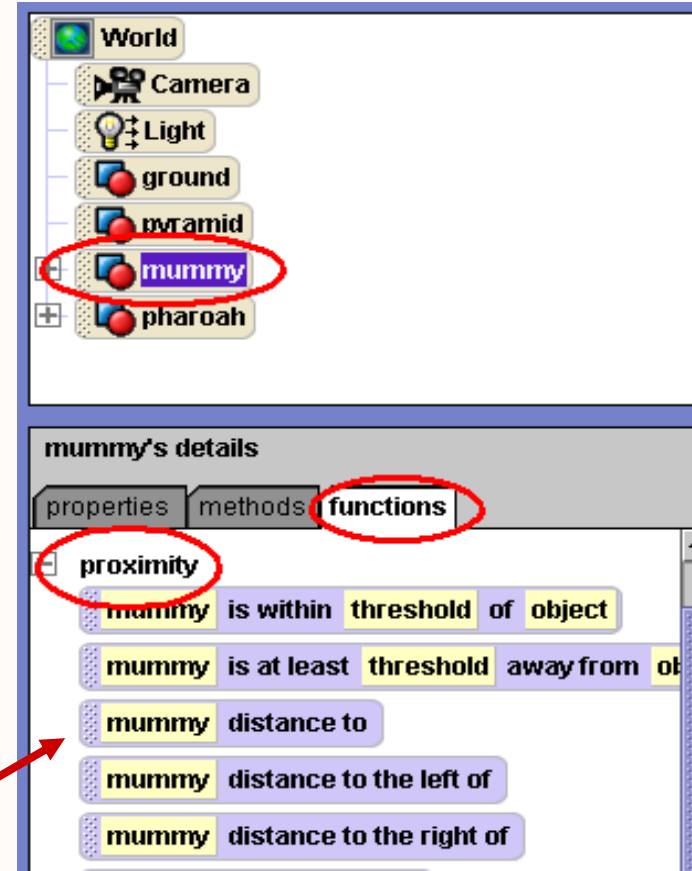
snowman turn right 0.5 revolutions more...



Built-in Functions

Categories

- proximity
- size
- spatial relation
- point of view
- other



This example illustrates some built-in proximity functions.



MIT App Inventor

“MIT App Inventor is an innovative beginner's introduction to programming and app creation that transforms the complex language of text-based coding into visual, drag-and-drop building blocks.”

from the MIT App Inventor website

- Originally developed by Google
- Developed by the MIT Media Lab, MIT Computer Science and Artificial Intelligence Lab

History

- Created by Google in 2010
- Given to MIT which released it in 2012
- On December 6, 2013 (*the start of the Hour of Code*), MIT released App Inventor 2

Try App Inventor

- Go to <http://appinventor.mit.edu/>
- Write a simple Android app