Learning for Industry 4.0

LESSON PLAN FOR 3D MODELING PROJECT: LET'S BUILD A DRONE!

COURSE DESCRIPTION

Create a drone frame in ten easy steps with NCLab's 3D Modeling app. The resulting frame can be exported as an STL file to be 3D printed, then used to build a real drone. You can even customize it with your initials! Complete the activity on your own, or as part of a class.

This is a "real world" introduction to 3D Modeling. 3D Modeling is used in engineering, manufacturing, art, science, medicine, and much more. Think about your favorite animations, or the advances in artificial limb design. Need a part? Design and print it on a 3D printer!

As you perform your coding magic, watch the drone frame come to life. The ten steps of the assembly process all begin with a short tutorial and clear instructions. Then, simply complete the code and press the play button. The results will show in the viewer. You will immediately see if your code is correct, or if you need to adjust it. Don't be afraid to make mistakes! Try each step as many times as you like.

You will practice many modeling skills and improve your understanding of geometry. Learn how to:

- Create basic geometric shapes.
- Move and rotate them.
- Create complex objects as unions of simple shapes.
- Copy objects, and mirror them across lines of symmetry.
- Extrude 2D objects to 3D.

Some basic background knowledge is helpful. By middle school, most students have learned the properties of angles, basic shapes, the XY coordinate plane, and how to move, rotate and reflect a shape. Therefore, we recommend 3D Modeling to students in Grades 6-8 and 9-12 (or older!).

Modeling is a vital part of the engineering cycle. For a few dollars, a 3D model can be printed, tested, and even redesigned. Once your frame is printed, you can quickly assemble a working drone using our downloadable instructions and suggested parts. The design is Plug and Play – no soldering or hardware needed!

Fly it for a while, then – jump back into the world of design. Do you want your model to use less materials, fly faster, lift more payload, or look more attractive? Try modifying the code to meet these goals. That's the beauty of 3D Modeling: it is easy to make changes.

Enjoy the drone project, and we hope you catch the 3D Modeling bug!

OBJECTIVES

By working through the steps in "Let's Build a Drone!", students will become familiar with several aspects of 3D modeling, the design process itself, and how models can be created with a script (code). Students will model 3D objects using PLaSM (Programming Language of Solid Modeling). Some of the skills practiced:

- Define, display, and color objects
- Create objects from geometric shapes, specifying the dimensions.
- Move and rotate objects in the XY coordinate plane.
- Create union sets of objects
- Extrude 2D objects to 3D
- Copy objects across planes of symmetry
- (In supplement) Add initials using Python

BACKGROUND KNOWLEDGE AND SKILLS

Geometry. Students in middle and high school grades should have enough experience with geometry to grasp what is taking place in the game. Helpful understandings:

- The XY coordinate plane: X and Y axes, grid spacing, coordinate pairs (including negative and decimal values).
- Properties of shapes: dimensions (lengths in x and y directions, radius, base x height).
- Angles and degrees of turn
- Transformations: translation, rotation, and reflection.

Coding. In this game, students are completing or modifying pre-existing code with very few changes. In other words, they do not need a strong coding background to complete the project. The language being used is PLaSM (Programming Language of Solid Modeling). It is based on Python and will accept Python code and libraries. Unlike menu-driven or drag-and-drop GUI CAD programs, PLaSM uses script to create and modify objects. Not only does this introduce students to what goes on behind the scenes in a GUI program, but it also links math and language to images and is a powerful way to teach visual-spatial reasoning.

BEFORE YOU START

Hardware. **Both PC and Mac platforms are supported.** Students can use desktop computers, laptops, and Chromebooks, and tablets. Phones are NOT supported.

Browsers. The most compatible browsers are Firefox, Chrome, Edge, and Safari. Internet Explorer is not compatible with certain graphic components.

Hour of Code sign on requirements. If you are doing this activity as part of Hour of Code, visit https://hourofcode.com/ for information on how to set up your students to play Hour of Code games, including this one. You can also go to the game directly at https://hoc.nclab.com/3d/

NCLab requirements. **The tutorial game itself is free and requires no registration.**

After completing the tutorial, students can access a set of instructions on how to add their initials to the drone frame. The free app at nclab.com will allow them to modify the frame design. To access the free app, they will need to sign up for a free account. Please note: Students under the age of 14 must use a parent email address or be part of a school account with signed parent permission, in accordance with COPPA rules. If they are 14 or older, they may use their own email address.

Internet connection. NCLab is cloud-based and most of the calculations are performed on the cloud-based server. Make sure that your local internet can support simultaneous use by all your users.

3D Printer (optional). If you wish to print the STL files for the drone frames, you will need a 3D printer. The frame is nearly flat and take approximately one hour to print.

Building drones (optional). Instructions are available at nclab.com for constructing real drones using the printed drone frames. You may want to build one or two for the class for demonstration purposes. Parts, including motors, propellers, control board, batteries, chargers, and a controller to fly the drone cost approximately $60 to $100 for everything. It is exciting to see the drone in action, and students can be encouraged to design and test improvements.

Color printer (optional). Print the posters to display in your classroom. Students will receive a printable certificate at the end of the game from Hour of Code which they may want to print.

NCLab's 3D Model gallery. To inspire students, check out the models created using NCLab's PLaSM. Share the link with your students: https://nclab.com/free-apps/ . Scroll down the page to see the gallery for 3D Modeling.

Print the notetaker and exit ticket for each of your students (at the end of this document).

TIPS FOR SUCCESS WITH ALL STUDENTS

Try the game yourself ahead of time to become familiar with the interface (see diagram in this document). **You do not need to be an expert in coding or modeling. Each step includes a tutorial and is auto-graded.**

"Navigator and Pilot" Partners: This is a tried and true technique that increases engagement and learning. There are ten steps, and students can switch roles so that they each have five turns at being the Navigator and five turns at being the Pilot. Alternatively, they can repeat the same step on the other student's device so that both students get credit for the work. Ideally, set up the pairs within the same zone of proximal development (ZPD).

> **The Navigator** reads the instructions and informs the Pilot. The Navigator stands next to the Pilot.
>
> **The Pilot** controls the mouse and the keyword, doing the writing and execution tasks. The Pilot sits at the computer.

This structure can help ease pressure on students who have difficulty reading and writing due to language or disability. Reading aloud helps with oral language skills. Our experience has been that dialogue ensues and the students help each other by asking and answering pertinent questions, and solving the problem together.

Keep it Student-Centered: Encourage students to answer questions among themselves (ask your partner, ask "three before me"). The game includes **tutorials** which explain each step, and **the coding part is simple**. Students only need to type in a few characters or numbers to run the program. **Students can also play with the values before submitting the "correct" answer, just to see what happens.**

Notetaker and Exit Ticket: Encourage **diagrams and drawings** as part of the response, especially for non-verbal students. Use a **Reader/Paraphraser** approach to increase participation: students take turns reading or describing their response and paraphrasing the other person's response.

COMPUTER SCIENCE STANDARDS

"Let's Build a Drone!" demonstrates three key understandings in computer science: **decomposing problems**, **using procedures with parameters**, and **building a prototype based on a computer program**.

Keep in mind that this activity guides students through a **tour** of applied 3D Modeling. Students complete the 10 steps according to explicit directions. Specific values must be entered at the end of each step to create a frame that will work with specific motors, propellers, and control boards, while limiting the weight so the drone can fly.

The standards require that students create their own programs, and this end will not be accomplished within an hourlong exercise with well-defined parameters. However, during the activity, you can encourage students to play with the values to see their effect, THEN put in the correct values before pressing the Submit button.

After completing the activity, students can access the entire script and, with the free NCLab application, modify parameters and components. If drones do not appeal to them, they can experiment with one of the many student-created models in the NCLab Gallery, or venture out and create their own from scratch.

Becoming expert at 3D modeling will not happen overnight of course. Some students pick up skills quickly and learn by experimenting with the application, using the list of commands as a guide. For most students, we recommend taking the 3D Modeling course, which teaches each modeling skill systematically.

Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

… break down problems into **subproblems**, which can be further broken down to smaller parts. Decomposition facilitates aspects of program development by allowing students to focus on one piece at a time (e.g., getting input from the user, processing the data, and displaying the result to the user) … Practice(s): Recognizing and Defining Computational Problems: 3.2

Create procedures with parameters to organize code and make it easier to reuse.

… create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. These procedures can be generalized by **defining parameters that create different outputs for a wide range of inputs**. For example, a procedure to draw a circle involves many instructions, but all of them can be invoked with one instruction, such as "drawCircle." By adding a radius parameter, the user can easily draw circles of different sizes. Practice(s): Developing and Using Abstractions: 4.1, 4.3

Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

A prototype is a computational artifact that demonstrates the core functionality of a product or process. Prototypes are useful for getting early feedback in the design process, and can yield insight into the feasibility of a product. The process of developing computational artifacts embraces both creative expression and the exploration of ideas to create prototypes and solve computational problems. Students create artifacts that are personally relevant or beneficial to their community and beyond. Students should develop artifacts in response to a task or a computational problem that demonstrate the performance, reusability, and ease of implementation of an algorithm. Practice(s): Creating Computational Artifacts: 5.2

(c) 2016 Computer Science Teachers Association https://sites.google.com/site/cstastandards/standards, viewed 10/24/2017

ACADEMIC STANDARDS

MATH: "Let's Build a Drone!" demonstrates key concepts in Geometry. This activity practices the following content standards in a simple and visual way.

Use the coordinate plane to represent real world problems:

Represent real world and mathematical problems by graphing points on the coordinate plane, and interpret coordinate values of points in the context of the situation. CCSS.MATH. CONTENT.x.G. A.2

Classify two-dimensional and three-dimensional figures into categories based on their properties.

Understand that attributes belonging to a category of 2D and 3D figures also belong to all subcategories of that category. *(Note that computer programs take advantage of these attributes. For example, all squares have equal sides and equal, 90-degree angle. Therefore, the only parameter needed to define a specific square is the length of one side.)* CCSS.MATH.CONTENT.x.G.B.3

Experiment with transformations in the plane:

Know precise definitions of angle, circle, perpendicular line, parallel line, and line segment, based on the undefined notions of point, line, distance along a line, and distance around a circular arc.

Given a geometric figure and a **rotation, reflection,** or **translation,** draw the transformed figure using, e.g., graph paper, tracing paper, or geometry software. CCSS.MATH.CONTENT.HSG.CO. A.1-5

Apply geometric concepts in modeling situations: Use geometric shapes, their measures, and their properties to **describe objects** (e.g., modeling a tree trunk or a human torso as a cylinder). CCSS.MATH.CONTENT.HSG.MG. A.1

© 2017 Common Core State Standards Initiative http://www.corestandards.org/Math/
Viewed 10/24/2017

STEM: "Let's Build a Drone" walks students through building a working prototype, a key part of the engineering design process. The technology in "STEM" is the software itself, and the option to print the frame on a 3D printer, assemble components to make a working drone, and revise the model's design as needed. In this activity, the design has already been made. However, students may wish to modify the design based on new criteria, their flying experience, or aesthetic preferences.

Define the criteria and constraints of a design problem. MS/HS-ETS1-1.

Next Generation Science Standards: For States, By States: MS-ETS1 Engineering Design

© NGSS 2013 http://www.nextgenscience.org/dci-arrangement/ms-ets1-engineering-design
, Viewed 10/24/2017

ENGLISH LANGUAGE ARTS: The activity itself requires reading and scripting. Students can write notes and a reflective piece, and engage in discussion around the essential questions.

ONE HOUR LESSON PLAN

The one-hour lesson plan includes an introduction, the activity itself, two response worksheets, and a celebration at the end. If you prefer, simply let the students go through the steps on their own, then regroup for discussion and further exploration.

Ideas for **project-based learning** are listed under Extensions. 3D Modeling and drones both lend themselves to follow up skills acquisition and project-based learning. We hope that this hour will whet the appetite for both!

INTRODUCE THE LESSON "3D Modeling can make some fun objects, but they aren't always useful. Today, we are going to build a drone frame on the computer. This frame can be 3D printed and used to build a drone that really flies. People often used the computer to create models. It can help solve problems while designing a product. We will be using scripted CAD, Computer Aided Design, which might be different from the model building programs you have used before. Scripted CAD combines coding with modeling." *Use the* ***posters*** *to introduce ideas about the essential questions.* **Why Learn 3D Modeling?** **Why Learn Scripted CAD?** **Why Build a Drone Frame?** *Hand out* ***notetakers*** *if they are to be used during the activity.* *Direct students to* https://hoc.nclab.com/3d/, *or they can select the activity from the Hour of Code list.*	5 minutes
ACTIVITY: BUILD A DRONE FRAME IN TEN STEPS *Review* **Game Layout, Using the Viewer** *if needed.* *Use* **NCLab "Let's Build a Drone!" Notetaker** *if desired.* *Encourage peer support so that all students complete the activity.* *See* **Lesson Extensions – For Speedy Students or Follow Up** *for students who complete the activity quickly.*	45 minutes
REVIEW LEARNING *Discussion and Exit Ticket: Review Essential Questions (refer to posters).* *Hand out the* **NCLab "Let's Build a Drone!" Reflection / Exit Ticket** *for students to complete as a review document and artifact.*	5 minutes
WRAP IT UP *Explore student 3D models in the* ***NCLab gallery*** *at nclab.com* ***Fly a mini-drone!*** *If you have a premade drone (preferably one made from the drone frame just printed), demonstrate it as a grand finale!*	5 minutes

ESSENTIAL QUESTIONS

WHY LEARN 3D MODELING?

Coding writes the story; 3D Modeling shows it. Like coding, 3D modeling is used in almost any field. This might be your future job! Here are some examples of how 3D modeling is used:

- Manufacturing: rapid prototyping, small-scale production
- Engineering: design, model-making, testing
- Construction: prefabrication, new techniques
- Applied Arts and Fine Arts: digital art, sculpture, industrial design, architecture
- Animation: entertainment, education, marketing
- Medicine: surgery, training, even making "parts" such as ears and kidneys!
- Science: modeling everything from molecules to the universe

It's fun to create and share your models with friends. Design a wedding cake, build a submarine, recreate your favorite game character.

Need a part? Build it yourself! Maybe all you need is one fastener, or a simple custom fitted plate. Create the model and print it on a 3D printer. If you don't have a printer, ask at your school or local library. You may be able to print your file there.

WHY LEARN SCRIPTED CAD?

NCLab's 3D Modeling uses a script to create 3D Models. Why bother? Most programs use menus or drag and drop. Writing code sounds like work. Well, here are some reasons why you want to learn scripted CAD.

- **Be a Code Wizard!** When we use a graphic user interface (GUI), we only see the surface. Someone wrote code behind those easy-to-use modeling tools. This is like pulling back the curtain in the Wizard of Oz. We want to be the wizards!
- **Brain Power!** You get to use more of your brain, which leads to better understanding. Reading and writing are powerful learning tools.
- Scripting is fast once you get used to typing the commands.
- A script shows your whole thought process. You can review, edit, and improve the script, or use it as a basis for another model.
- With NCLab's PLaSM language, you can import **Python libraries** to create some cool stuff once you have developed the basic skills.

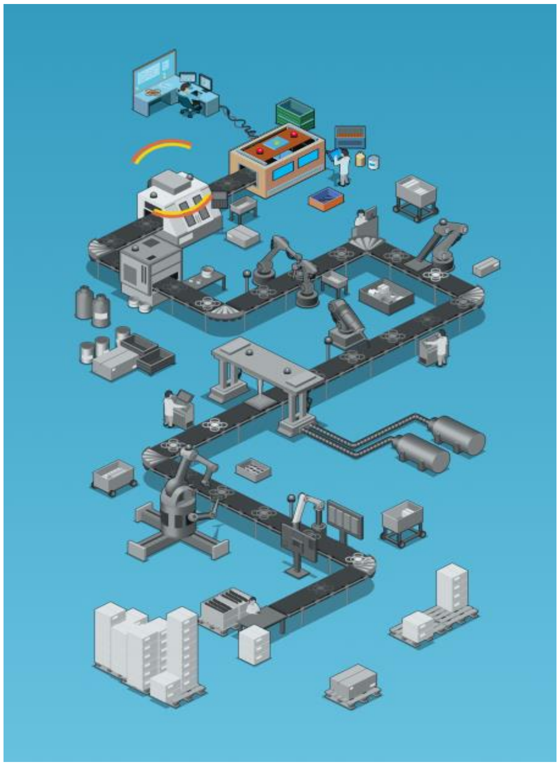
FINALLY, WHY BUILD A DRONE?

Drones are used everywhere: scientific surveys, cinematography, the military, delivery systems, and more.

You can buy a premade drone, and the price keeps going down. However, **building one yourself will teach you some valuable design and modeling skills**. It is very satisfying to make a working model. We guarantee that after you have gone through this design process, you will look at commercially available drone frames with fresh eyes. How do different designs and materials affect the performance or intended purpose?

Drones are fun to fly! Whether racing or doing stunts, drones are great fun. Remember to be kind to people and animals.

GAME LAYOUT, USING THE VIEWER



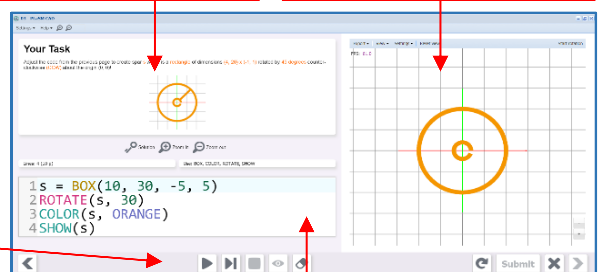
The game is laid out like an assembly line. Each step builds a part of the drone frame and moves it along to the next step. Once a step has been successfully completed, it will be marked with a green checkmark.

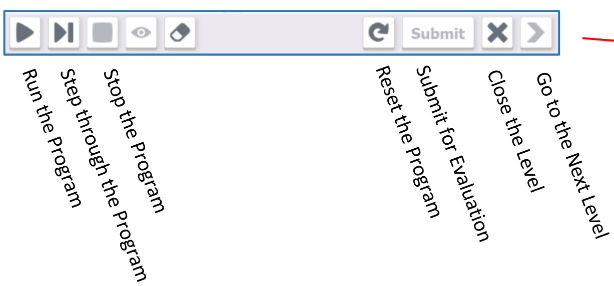
Notice the robotic arms and the highly trained technicians controlling the process! Encourage your students to see themselves as designers and experts, communicating with the instruments and machines that build the components.

The task window for each step looks like this:

The Tutorial and Instructions explain concepts and procedures, and provide detailed instructions.

The Viewer displays the model generated by the code. It also displays the components which have already been built.

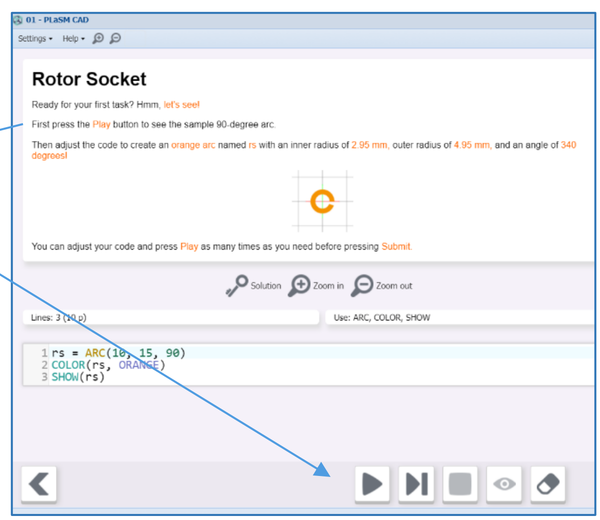




The Code Cell provides a place to write the script. Most of the code has already been written. Students modify the existing code to create the required drone frame component.

Your students will get a better feel for the effect of their code if they **play the sample code first** when instructed to do so.

First press the Play button to see the sample 90-degree arc.



Other features include a **list of PLaSM keywords** (top menu), **zoom buttons** to increase or decrease text size in the code cell (above the code cell and at the top), and **viewer control buttons** (above the viewer).

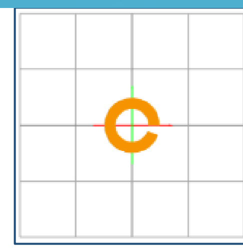
ACTIVITY: BUILD A DRONE FRAME IN TEN STEPS

The "Let's Build a Drone!" activity is simple to execute and does not require depth of knowledge. Most levels only require a few basic actions such as replacing some text or numbers in the code. However, if you or your class wish to know more, here is a detailed explanation for each step and keyword used to build the drone frame.

Step 1. Let's Build a Drone! / Rotor Socket

3D Modeling Skill: Create an object. A rotor socket is created using the keyword ARC.

Design Purpose: A 6 mm diameter rotor motor can be press fit into a close-fitting rotor socket. This eliminates the need for fasteners. The arc has a "spring" action.



In Step 1, construction begins with the rotor socket, which will hold a rotor motor in place.

We start with a simple line of code that defines **object name** equal to a **keyword**.
Objects are named using letters, letters and numbers, or even words. Letters should be lower case so that they aren't confused with KEYWORDS, which are upper case. Choose a name that is concise and representative. In this step, the object name "rs" stands for "rotor socket".
KEYWORDS call a **function** and are always **capitalized** (upper case). The keyword is followed by **parameters** in parentheses. Think of parameters as information the keyword function needs, such as dimensions or color. If more than one parameter is needed, a comma separates each parameter in the list.

In Step 1, the keyword **ARC** creates an arc. ARC() is written as:

```
ARC(2.95, 4.95, 340)
ARC(inner radius, outer radius, degrees of turn)
```

*Note: The default turning direction for the arc is **counterclockwise**, starting at 0 degrees in the x-axis in the first quadrant.*

Writing the line of code correctly: Syntax!
Spelling, punctuation, capitalization, and grammar all provide meaningful information and must be written correctly, or the program will not run. In this introductory Hour of Code activity, **most of the code is prewritten** to keep the tasks simple.

In the PLaSM Python-based language, the line of code needed to define an object follows this structure:

```
object_name = KEYWORD()
```

In this example the object name is **rs**, and the keyword is **ARC()**. The sentence is written

```
rs = ARC(2.95, 4.95, 340)
```

Try reading the command in natural language:

"The object **rs** is an **arc** with a 2.95 mm inner radius, a 4.95 mm outer radius, and 340-degree turn."

*Note on dimensions: **each grid interval in the viewer represents ten units**. The downloadable STL file (3D Modeling print file), specifies millimeters, so we can infer that each interval represents **10 millimeters** when looking at the code results in the viewer.*

Coloring and displaying objects: This time, think of the line of code as a command, starting with the verb.

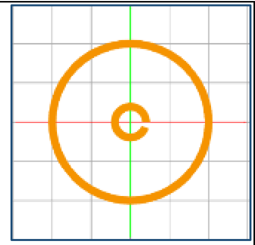
```
COLOR(rs, ORANGE)
SHOW(rs)
```

reads as

"Color the object rs orange."
"Show the object rs."

For fun: try replacing ORANGE with another color name such as MAGENTA or WOOD. Remember to change it back to ORANGE before submitting.

Step 2. Making Rings



3D Modeling Skill: Create an object. A rotor ring is created using the RING keyword.

Design Purpose: An outer ring is required to protect the propellers.

Step 2 creates the next shape needed for our rotor duct assembly – the rotor ring. The size of the ring is constrained by these factors:

- It must be larger than the propellers.
- It must accommodate the length of the leads running from the control board to the motors.

Fortunately, NCLab has already made the necessary calculations!

The keyword **RING** only requires two parameters:

RING (2.95, 4.95)

RING (inner radius, outer radius)

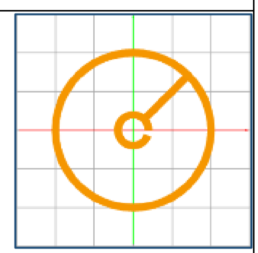
Note the similarity with ARC. Since a RING always turns 360 degrees, we do not need the third parameter.

Most levels start with a sample code to illustrate the new keyword. The required action is to modify this code so that it fits the drone frame requirements.

The instructions encourage pressing the PLAY button before modifying the code. This procedure helps students see the impact of entering different values for the parameters.

Play with it! Try different values for the inner and outer radius to see the effect. Then, put in the required values before submitting the code.

Step 3. Making Rectangles/ Rotating Objects



3D Modeling Skill: Create a rectangle using the keyword BOX(). Rotate the rectangle using the keyword ROTATE().

Design Purpose: Create a spar to hold the rotor socket in place and connect it to the outer ring.

Step 3 creates the next shape needed for our rotor duct assembly – the spar. Two actions are needed to create the spar:

- Create a rectangle s using the keyword **BOX.**
- Rotate s using the keyword **ROTATE**.

The keyword **BOX** requires four parameters. The first two represent the dimension and position in the x direction (xmin and xmax), and the second two represent the dimension and position in the y direction (ymin and ymax). These can also be thought as the product of x and y (xmin, xmax) x (ymin, ymax). The keyword BOX allows placement of a rectangle anywhere in the xy plane.

BOX (4, 20, -1, 1)

BOX (xmin, xmax, ymin, ymax)

Note: negative values can be used, since we are placing the rectangle anywhere on the xy plane.

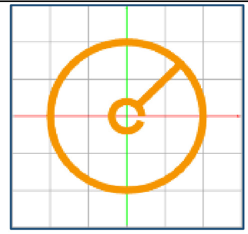
Next, we rotate the rectangle s so that it radiates from the center of the frame. The keyword **ROTATE** requires two parameters in this application. This is a verb command, so the first parameter is the **name of the object**, the second is the **degrees of rotation**. The default rotation is counterclockwise about the origin, which we are using in this case.

ROTATE (s, 45)

ROTATE (object_name, degrees)

Play with it! Try different values for each keyword to see the effect. Then, put in the required values before submitting the code.

Step 4. Create Unions of Objects



3D Modeling Skill: Create an object rd (rotor duct). rd is the UNION of the rotor socket rs, rotor ring rr, and spar s, and is the set of all points contained in the three objects. "Gluing" the parts together simplifies the remaining design tasks.
Design Purpose: Join the rotor socket, rotor ring, and spar together to form an assembly called the rotor duct. We need four rotor ducts for the drone frame, one for each propeller.

Step 4 We use UNION() to build the rotor duct.

The keyword **UNION** requires a list of the two or more objects to be included in the new set. In this case, we are naming an new object rd (rotor duct) as this union.

```
rd = UNION (rs, rr, s)

object_name = UNION (object1, object2, …, objectn)
```

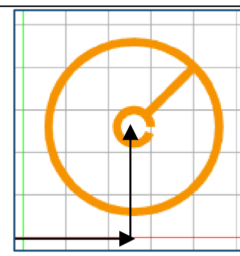
UNION() can also be embedded in a command. For example:

```
ROTATE(UNION(rs, rr, s), 30).
```

This is a handy shortcut if a particular UNION is only required one time. Otherwise, it is still better to create an object from the union, then act on it in the following lines of code.

We will use rd in the next two Steps.

Step 5. Translating (Moving) Objects



3D Modeling Skill: Move an object using the keyword MOVE(). Most objects are created in a default location, either centered on the origin (0,0), or in the first quadrant, using (0,0) as the lower left vertex. After objects are created, we can move them anywhere in the XY plane using MOVE.

Design Purpose: The rotor duct is moved to its final position in the frame design.

Step 5 moves the rotor duct assembly into its final position in the frame.

The keyword **MOVE** can be written different ways. In this case we will use two parameters, one for the movement in the x direction (horizontal), and one for the movement in the y direction (vertical).

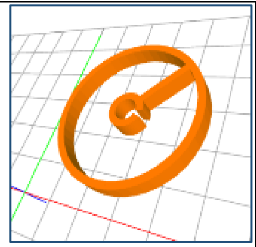
```
MOVE(rd, 26, 26)

MOVE(object_name, x_distance, y_distance)
```

Note: negative values can be used to move x or y in the negative direction on the xy plane.

Observe the changing relationship to the red X axis and green Y axis after the MOVE command.

Play with it! Try different values for x and y to see the effect. Then, put in the required values before submitting the code.



Step 6. Extruding Objects to 3D

3D Modeling Skill: Extrude an object from 2D to 3D using the keyword PRISM().

Design Purpose: The frame is a printable 3D object. All components must be converted to 3D to print. We have completed what we need to do on the XY plane for the rotor duct (create the socket, ring, spar, glue them together, and move the rotor duct assembly into position). The rotor duct is now ready to be extruded.

Step 6 extrudes the rotor duct to 3D.

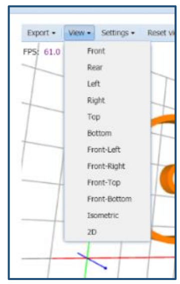
A new object, in this case rd1, is created using the keyword **PRISM,** which takes two parameters. The first parameter is any 2D object, in this case, rd. The second parameter is the height of the prism.

rd1 = PRISM(rd, 5)

3Dobject_name = PRISM(2Dobject_name, height)

Changes in the viewer:

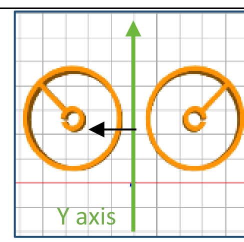
When you run a program that has created a 3D object, the resulting image will be displayed in 3D instead of 2D. **The z axis represents the 3rd dimension and is colored blue.**



The image can be viewed in from different 3D perspectives, and in 2D., selected from the buttons at the top of the viewer.

You can also move, rotate, and zoom the image with your mouse.

Play with it! Try different values for the height to see the effect. Then, put in the required values before submitting the code.



Step 7. Exploiting Symmetry

3D Modeling Skill: Symmetry is used to create a reflection of an object across a line or plane. Since we want to preserve our first object (rd1), we start by making a copy (rd2) using the keyword COPY(). We then reflect the copy across the Y axis using the keyword MIRROR().

Design Purpose: The rotor ducts are arranged symmetrically about the center of the frame. Step 7 creates the 2nd rotor duct.

Step 7 makes a copy of rd1 (the 3D rotor duct assembly). This new object, rd2, is reflected across the Y axis using the keyword MIRROR().

The keyword **COPY()** only need one parameter, which is the name of the object being copied.

rd2 = COPY(rd1)

new_object_name = COPY(original_object_name)

The keyword MIRROR() needs three parameters, starting with the name of the object being mirrored. The second and third parameter can be read together. To reflect across **the line made by the Y axis**, we are actually reflecting across **all points where X = 0**.

MIRROR(rd2, 0, X)

MIRROR(object_name, value, axis)

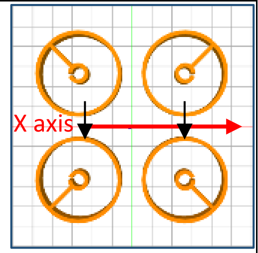
Play with it! Try different values and axes (X, Y, or Z) to see the effect. Then, put in the required values before submitting the code.

Note: rd1 is already displayed, so the SHOW() command only needs to show rd2.

Step 8. More Symmetry

3D Modeling Skill: As in Step 7, we make copies of rd1 and rd2 and reflect them across the X axis this time using the keyword MIRROR().

Design Purpose: Using symmetry to reflect the rotor ducts ensures that the specifications are the same for all four, and that they are positioned correctly.



Step 8 is the second step using COPY() and MIRROR(). This step creates the last two rotor ducts.

The third rotor duct (rd3) is created as a copy of object rd1, and then mirrored across the X axis (the line where Y = 0).

The fourth rotor duct (rd4) is created as a copy of object rd2, then mirrored across the X axis.

To reflect across **the line made by the X axis**, we are actually reflecting across **all points where Y = 0**.

MIRROR(rd3, 0, Y)

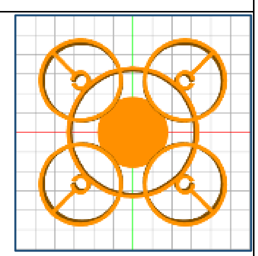
MIRROR(object_name, value, axis)

Note: rd1 and rd2 are already displayed, so the SHOW() command only needs to list rd3 and rd4.

Step 9. Making Cylinders

3D Modeling Skill: Create a 3D object using the keyword CYLINDER(). We also create an extruded ring.

Design Purpose: The cylinder connects the rotor ducts and provides a platform for the receiver board. The ring reinforces the frame, and balances the spars in the rotor duct design.



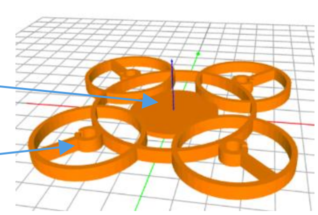
Step 9 creates a **cylinder** using the keyword **CYLINDER().** Notice that a cylinder is a 3D shape by definition. We do not need to extrude it using PRISM(). To define the cylinder, we need two parameters: the **radius** of the base circle, and the **height:**

c = CYLINDER(18, 2)

object_name = CYLINDER(radius, height)

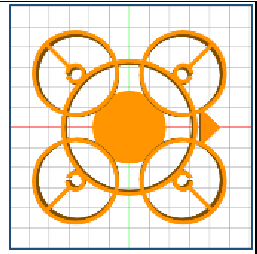
The height of the cylinder c is 2 units, which nests it inside the frame.

The frame height is 5 units.



The extruded ring is created like the one in Step 2, using different values for the inner and outer radius, and a different value for height in the keyword PRISM.

Step 10. Direction of Flight



3D Modeling Skill: Create a triangle by defining XY coordinates for the vertices, and extrude it to 3D.

Design Purpose: The arrow marks the front of the drone. This needed is for flying (pointing the drone away from the controller), and for mounting the clockwise and counterclockwise motors correctly. It also serves as a handy hook for the rubber band used to secure the control board and battery.

Step 10 creates a triangle using the keyword **TRIANGLE(),** which includes the keyword **POINT()** used to define each vertex. To make a triangle, we need three parameters, one for each vertex, and each of those parameters contains an x and y value.

```
a0 = TRIANGLE(POINT(35, -9.5), POINT(35, 9.5), POINT(45, 0))

object_name = TRIANGLE(POINT(x1, y1), POINT(x2, y2), POINT(x3, y3)
```

The code cell may wrap the text on a line of code this long. This does not affect how it runs.

Use PRISM() to extrude the triangle. Notice that "arrow" is used for the object name. Descriptive names are useful.

The drone frame is now complete!

On the next pages, you will see a link to download the STL file for the drone frame, which you can send to a 3D printer. This frame has been tested many times and stands up well!

You can also download a document with:

- Instructions on how to customize the lettering using a Python library.
- Instructions on how to assemble the drone, with suggested parts. There are many manufacturers of blades, motors, control units, batteries, and chargers. You may have your own favorite sources. Don't forget to purchase a flight controller as well to guide your new drone!

The next few screens suggest extensional activities, including:

Downloading the STL file for 3D printing — STL File for NCLab Drone Frame

Personalizing the drone frame with initials, and building a real drone using the printed frame — Personalize and Assemble Your NCLab Drone Frame

The game ends with a video showing our first NCLab drone in flight at the Innevations Center Maker Space!

YouTube Video: First NCLab Flying Drone

These extensional activities are described in more detail under Extensions

REVIEW AND ASSESSMENT

DISCUSSION AND EXIT TICKET

As a teacher, you know your students best. Will your students discuss as pairs, small groups, or whole class? Do you assign roles? Do you use rules or game structures in your discussion? For this one-hour lesson, discussion will necessarily be brief. If you have a wireless microphone and recording device, try passing the mic around and recording responses. The exit ticket can be completed during or after discussion of the essential questions.

Here are the topics again:

WHY LEARN 3D MODELING?

Coding writes the story; 3D Modeling shows it.

Job Possibilities: manufacturing, engineering, construction, applied arts, fine arts, animation, medicine, science.

It's fun to build stuff and share your models with friends. What would you design?

Need a part? Build it yourself! What parts could you build?

WHY LEARN SCRIPTED CAD?

Be a Code Wizard! Go "behind the curtain": what is taking place at the coding level?

Use more of your brain: Reading and writing are powerful learning tools.

Use Python libraries with PLaSM: Python is a powerful programming language with a great community and lots of downloadable libraries. These are compatible with NCLab's PLaSM.

WHY BUILD A DRONE?

Drones are used everywhere and will be used in many professions.

DIY Drones. Learn more by building a drone yourself than by simply purchasing one. **Finding your own engineering "special powers".** What part of the design process is appealing to you?

Learn to fly! These drones are fully functional. Learning to fly is fun, but requires a lot of practice. Be mindful of people and animals.

WRAPPING IT UP/CELEBRATION

FLY A DRONE OR PLAN B: WATCH NCLAB FLY THE FIRST DRONE MADE WITH THIS DESIGN

End the day with a few minutes of fun. Mini-drones are fast and exciting to fly. If you built one ahead of time, or have a mini-drone of your own, fly it around the room and create some "buzz". Word of caution: take turns flying one drone and schedule some turns for another day. The batteries typically only last about 2 to 3 minutes, and that provides a nice, short exit.

If you don't have one to fly, you and your class can watch the maiden flight of our NCLab drone again with this link:

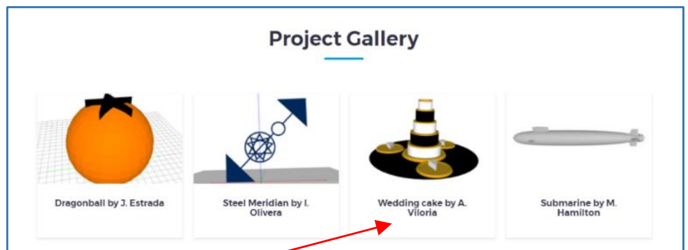
[YouTube Video: First NCLab Flying Drone]

EXPLORE STUDENT MODELS IN NCLAB'S GALLERY

Check out some student work at [NCLab's gallery.] The actual website address is https://nclab.com/free-apps/.

Scroll down the page to see the gallery.

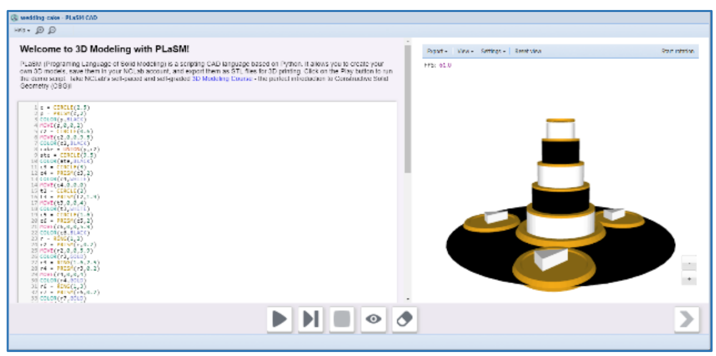
Click on a design to open the viewer which will show the code. You can run the program to produce the model, and even modify it to create something new.



For example, let's open the Wedding Cake.

Run the code to produce the wedding cake in the viewer.

The code can be edited. Change colors and shapes, or add details. With an account, these files can be saved and worked on later.



If your students take the 3D Modeling course, they will learn all a full range of techniques and tools that they can use to create or modify designs. For now, encourage them to experiment with the file to see what happens – they can't hurt the original!

It is also fun just to browse the different models and see how they are built.

LESSON EXTENSIONS – FOR SPEEDY STUDENTS OR FOLLOW UP

These extensions build on the one-hour lesson, or provide avenues to explore for students who finish quickly.

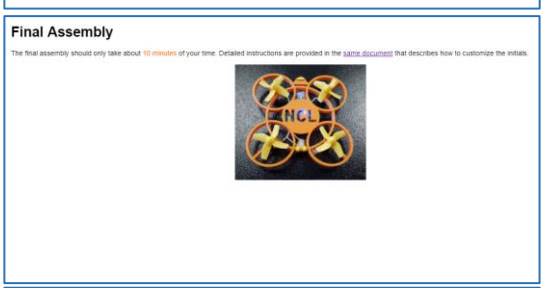
"Let's Build a Drone!" can be used as an introduction to a more in-depth look at the wide range of applications for 3D Modeling and drone technology.

USE THE LINKS IN THE COURSE TO DO THE FOLLOWING:

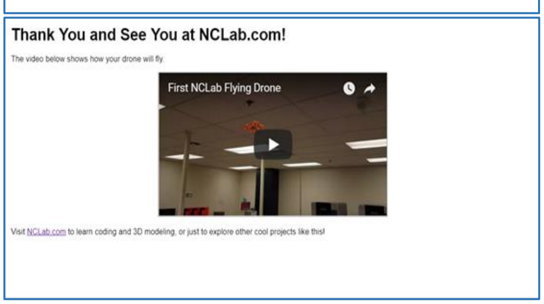
- **3D Lab:** Print the frame, buy the suggested parts or equivalents, and assemble a drone using the instructions.
- **3D App:** Customize frame with initials using PDF link. Play with the parameters to modify the design (requires free NCLab account to access the free app and save files).

Here are links to the resources which can also be accessed from Step 10 of the Hour of Code.

STL File for NCLab Drone Frame



Personalize and Assemble Your NCLab Drone Frame



YouTube Video: First NCLab Flying Drone

PROJECT BASED LEARNING:

Engineering Design: Your students have walked through the creation of a functional model. This is a great opportunity to expand their understanding of engineering design principles in a real-world context. Some possibilities:

- **Research drone types, designs, prices, and purposes.**
  - Possible outcomes:
    - Essays
    - Posters
    - Dramatic presentations (songs, theater)
    - Demonstrations and labs
    - Videos, slide shows
- **Customize the NCLab design using the app. Think about:**
  - Aesthetics – the look and style
  - Specialized purpose – what will you use the drone for?
  - Best use of materials – can you change the dimensions or type of plastic?
  - Aerodynamics – how can we improve flight performance?

Physics and Math: The way a drone navigates 3D space is a fascinating look at physics and math, again in a real-world context. Research topics:

- **Physics of flight:** lift, drag, thrust, weight
  - Describe in general terms (labeled diagrams, demonstrations).
  - Model these characteristics in a lab.
  - Research and examine the equations involved. Even if these are too complex for your students to solve, it is worth a look. They may recognize certain portions of the equations.
- **Optimizing geometry.** Specific shapes affect
  - Flight (aerodynamics).
  - Stability and durability (stress, strain).
- **Using rotation to control direction.** Rotation is a difficult but extremely useful concept to master.
  - The motors and propellers have a specific **counterclockwise and clockwise layout** to control the direction of flight.
  - Model the **air flow** based on which propellers are being engaged by the controls.
  - Try changing the configuration to see what happens.

Learn to Fly a Drone: There are many learn-to-fly videos and flight simulators available for free or a modest fee. Consider starting a flying club.

Design, Modeling, and Drone Citizenship: this is a rich area to research.

- **What are the federal, state, and local regulations?**
  - Which drones are affected?
  - Do the regulations make sense? Can they be improved?
- **Privacy concerns**
  - What are the limits on invading someone else's space, especially with a camera?
- **Sharing airspace**
  - Airports and flight paths

    - Emergency situations. Firefighters are very concerned about drones taking video footage of fires while they are using fire-fighting aircraft in the same space.
    - International restrictions.
- **Use in military operations**
    - Pros and cons in reconnaissance, surveillance, combat, security.
- **Safety**
    - Propeller design; guards
    - Lithium Polymer Battery care - how are LiPo batteries different from other battery types
    - Assembly procedures
    - Static electricity

RESOURCES

On the following pages you will find:

Student Note Taker

Student Reflection/Exit Ticket

Poster: Why Learn About 3D Modeling?

Poster: Why Learn About Scripted CAD?

Poster: Why Learn About Drones?

There are many great videos and articles on the Internet. Use keywords suggested in the essential questions and projects to learn more about 3D Modeling, scripted CAD, and drones.

NCLAB COURSES AND SUPPORT

Of course, 3D Modeling can't be learned in one hour! We hope you enjoyed the activity and invite you to learn about NCLab's courses and apps in 3D Modeling at

https://nclab.com/

For support issues, please email support@nclab.com .

We look forward to hearing from you!

NCLAB "LET'S BUILD A DRONE!" NOTETAKER

NAME: _______________________________

NOTETAKERS ARE HANDY WHEN YOU ARE LEARNING SOMETHING NEW AND IT IS FRESH IN YOUR MIND. WRITE A NOTE OR QUESTION FOR EACH STEP AS YOU GO.

Step 1. Let's Build a Drone! / Rotor Socket
Step 2. Making Rings
Step 3. Making Rectangles/ Rotating Objects
Step 4. Create Unions of Objects
Step 5. Translating (Moving) Objects

Step 6. Extruding Objects to 3D
Step 7. Exploiting Symmetry
Step 8. More Symmetry
Step 9. Making Cylinders
Step 10. Direction of Flight
Step 10. Extras (instructions on building the drone and adding initials, video of the first flight)

NCLAB "LET'S BUILD A DRONE!" REFLECTION/ EXIT TICKET

NAME: ________________________________

A REFLECTION OR EXIT TICKET HELPS YOU THINK ABOUT WHAT YOU HAVE LEARNED AND WHAT YOU WOULD LIKE TO LEARN NEXT.

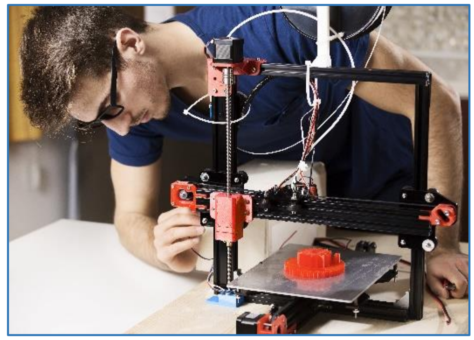
What would I like to model? If I had to choose a job using 3D Modeling, what would it be?
In this exercise, the drone frame was built using a scripted CAD language. How does typing the commands change the 3D modeling experience?
Drones are used for many purposes. How would I would use a drone?
I would like to learn more about:

WHY LEARN 3D MODELING?

Coding writes the story; 3D Modeling shows it.

Like coding, 3D modeling is used in almost any field.

This might be your future job!



- **Manufacturing: rapid prototyping, small-scale production**
- **Engineering: design, model-making, testing**
- **Construction: prefabrication, new techniques**
- **Applied Arts and Fine Arts: digital art, sculpture, industrial design, architecture**
- **Animation and Game design: entertainment, education, marketing**
- **Medicine: surgery, training, even making "parts" such as ears!**
- **Science: modeling everything from molecules to the universe**

It's fun to build and share your own models.

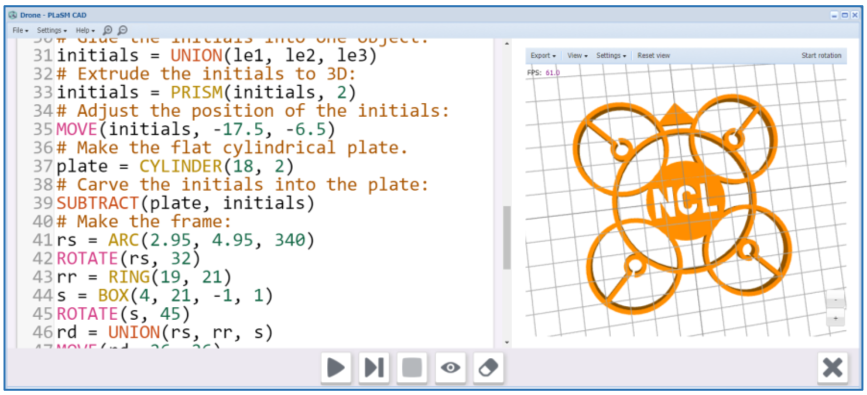
Need a part? Build it yourself!

Maybe all you need is one fastener. Create and print it on a 3D printer.

WHY LEARN SCRIPTED CAD?

We want to be Code Wizards!

When we use a graphic user interface (GUI), we only see the surface. Someone **wrote code behind those easy-to-use modeling tools**. This is like pulling back the curtain in the Wizard of Oz. It may seem boring at first to write code, but it is a **very powerful.**



Brain Power!

You get to use more of your brain, which leads to better understanding. Reading and writing are powerful learning tools.

- Scripting is fast once you get used to typing the commands.
- A script shows your whole thought process. You can review, edit, and improve the script, or use it as a basis for another model.

Python Power!

NCLab's PLaSM language accepts Python libraries and commands. Once you have developed some basic skills, Python can extend your powers.

WHY LEARN ABOUT DRONES?

Drones are like robots, except for one thing.

They fly.

Guess what? Drones may be part of your future job, too. They are incredibly useful and can be programmed to do many tasks!

Photography and Cinematography: surveillance, real estate, scientific studies, news, and sports coverage.

Delivery systems: regular parcel delivery, supplies in disaster areas

Military systems: reconnaissance, combat, defense.

Measurement: drones can be equipped with sensors that measure everything from spatial dimensions, to pressure and temperature.

Entertainment and Sports: racing, acrobatics, and other sports.

Building your own drone teaches you about design and engineering – even physics.

There's nothing like building your own! You will appreciate all the details and problem-solving that go into designing and engineering.

Drones are fun to fly, once you know how!

You can race drones and do tricks with them. Please be respectful of people and animals. Be patient: start with a small, inexpensive drone and be prepared to crash!

nclab
Learning for Industry 4.0