



- We'll get started promptly at 9:00.

**Menti.com**

**Submit code: 9530 5481**

**Answer the prompt provided**

“It’s not the note you play that’s the wrong note – it’s the note you play afterwards that makes it right or wrong.”

- *Miles Davis*



<https://youtu.be/VRZKKMvQIfk>





# Assessment: Andrew & Judith

A tool for feedback, accountability  
& improvement

# GOALS FOR TODAY



- Explore Module 2 content and activities
- Discuss classroom application of Alice
- Discuss curriculum integration framework

# PURPOSE

- Explore programming in Alice
- Gain ideas on how to incorporate logical thinking in your own classroom.

## NON-PURPOSE

- Become an expert in programming.

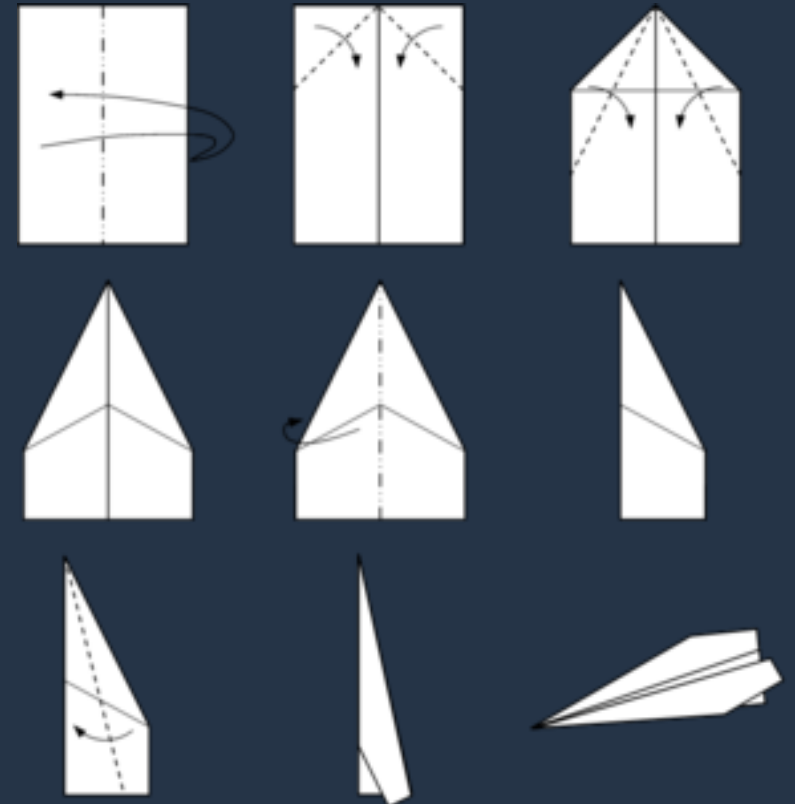


# Paper Folding Activity



1. Get a sheet of paper
2. Fold it in half, lengthwise, to get a center line only
3. Fold the top 2 corners into the center
4. Fold the angled edge into the center
5. Fold along the center line
6. Fold down the flaps

1. Get a sheet of paper
2. Fold it in half, lengthwise, to get a center line only
3. Fold the top 2 corners into the center
4. Fold the angled edge into the center
5. Fold along the center line
6. Fold down the flaps



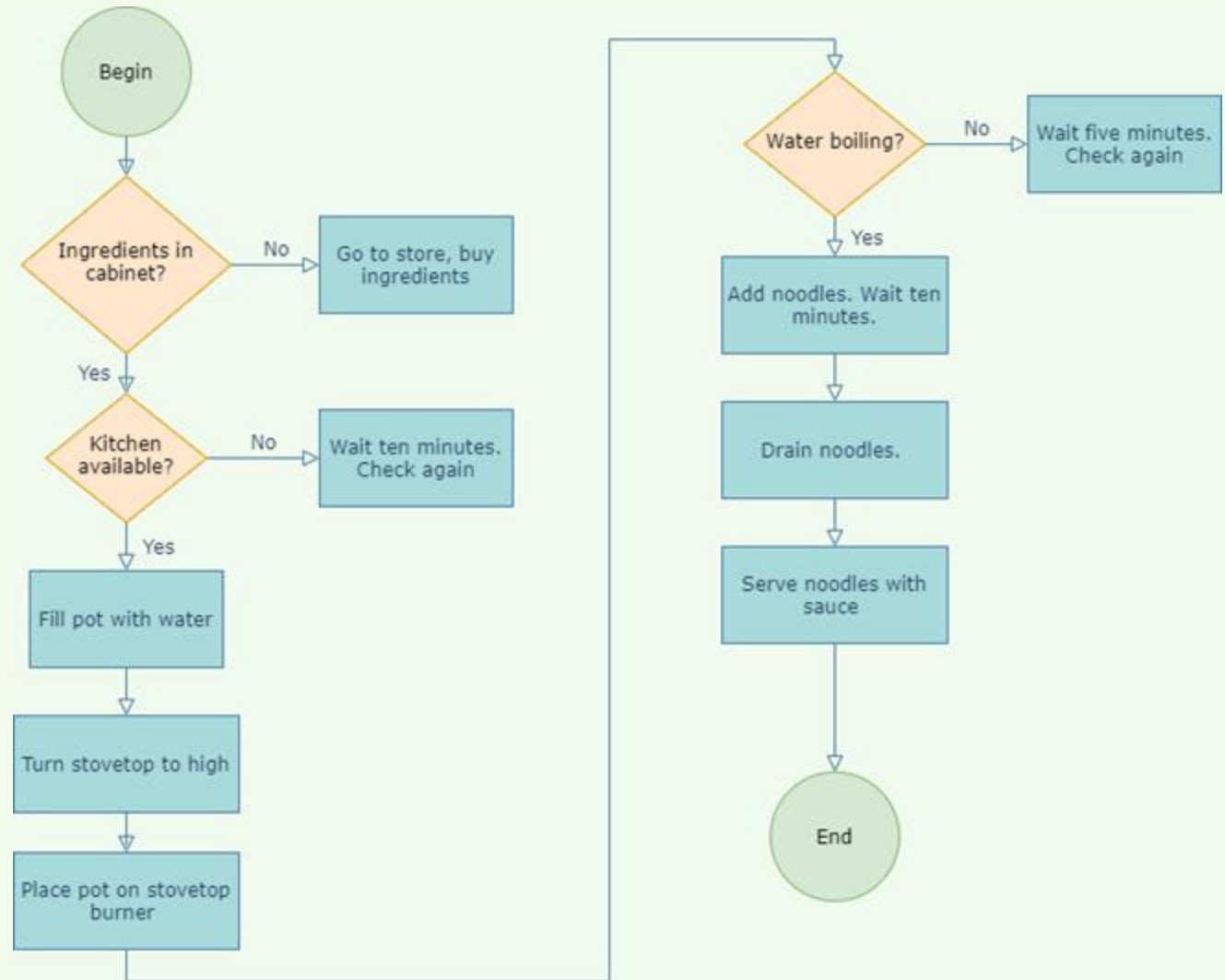


# **CODERS Module 2**

Introduction to Alice

# Connecting the dots ...

**Software:** Set of instructions, procedures, and routines that enable the computer to perform tasks.



# Introduction

Alice is a block-based drag and drop programming environment that makes it easy to create animations, build interactive narratives, or program simple games in 3D.

Alice is hosted by Carnegie Mellon University and sponsored by Oracle, Java, EA, and National Science Foundation (NSF).

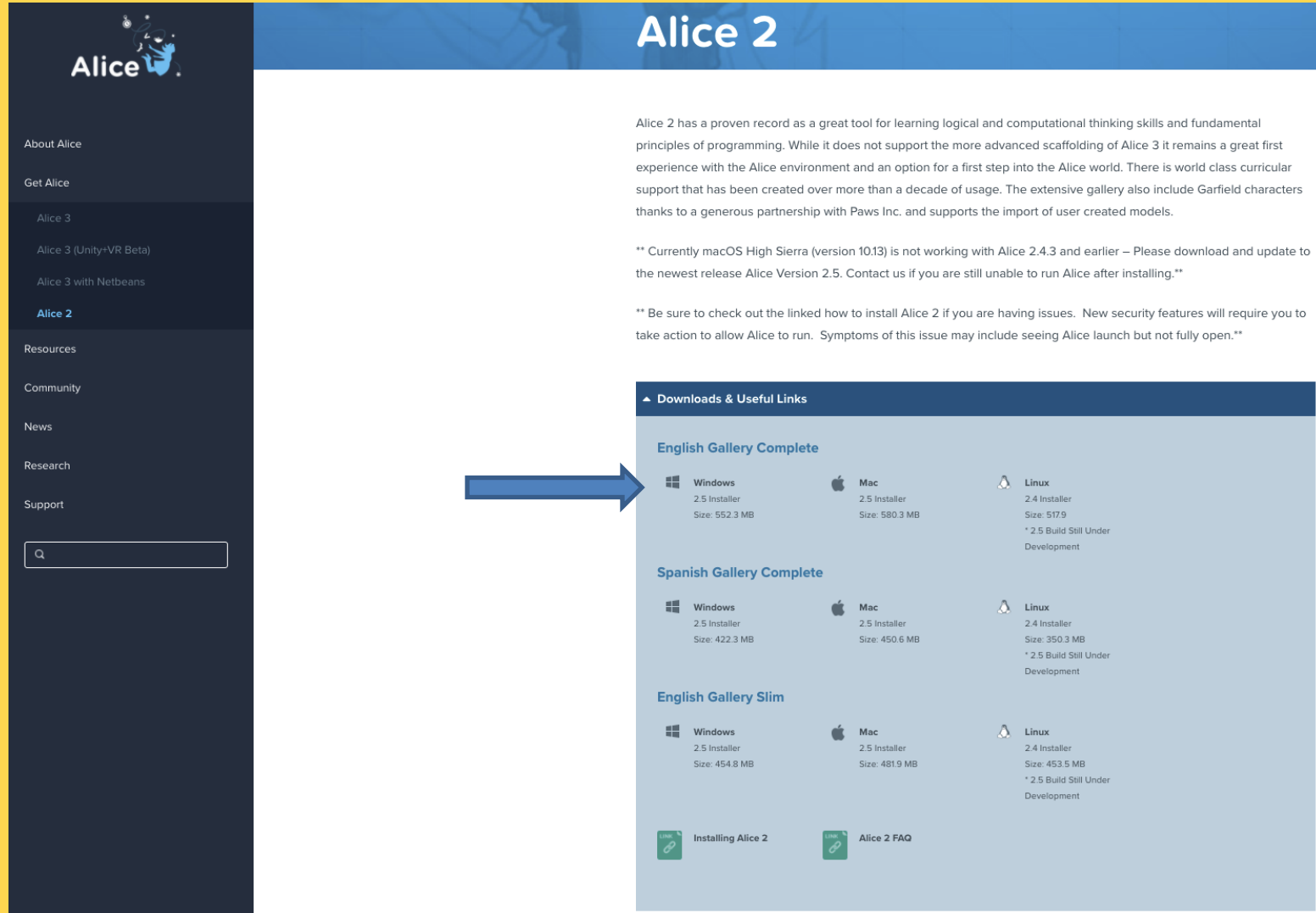
# Download and Install Alice

**Step 1:** Download Alice 2 via <https://www.alice.org/get-alice/alice-2/>

**Step 2:** Go to the folder where you downloaded Alice 2.

**Step 3:** Unzip the downloaded zip file by right clicking it, and click "Extract All"

**Step 4:** Double click Alice.exe



**Alice 2**

Alice 2 has a proven record as a great tool for learning logical and computational thinking skills and fundamental principles of programming. While it does not support the more advanced scaffolding of Alice 3 it remains a great first experience with the Alice environment and an option for a first step into the Alice world. There is world class curricular support that has been created over more than a decade of usage. The extensive gallery also include Garfield characters thanks to a generous partnership with Paws Inc. and supports the import of user created models.

\*\* Currently macOS High Sierra (version 10.13) is not working with Alice 2.4.3 and earlier – Please download and update to the newest release Alice Version 2.5. Contact us if you are still unable to run Alice after installing.\*\*

\*\* Be sure to check out the linked how to install Alice 2 if you are having issues. New security features will require you to take action to allow Alice to run. Symptoms of this issue may include seeing Alice launch but not fully open.\*\*

**Downloads & Useful Links**

**English Gallery Complete**

OS	Version	Size
Windows	2.5 Installer	552.3 MB
Mac	2.5 Installer	580.3 MB
Linux	2.4 Installer	517.9 MB

\* 2.5 Build Still Under Development

**Spanish Gallery Complete**

OS	Version	Size
Windows	2.5 Installer	422.3 MB
Mac	2.5 Installer	450.6 MB
Linux	2.4 Installer	350.3 MB

\* 2.5 Build Still Under Development

**English Gallery Slim**

OS	Version	Size
Windows	2.5 Installer	454.8 MB
Mac	2.5 Installer	481.9 MB
Linux	2.4 Installer	453.5 MB

\* 2.5 Build Still Under Development

[Installing Alice 2](#) [Alice 2 FAQ](#)

# Break Time



# Alice Programming

Story Telling



# The Unexpected Journey

Video

# Getting Started

## Object Tree

Lists objects in World

## Scene editor

Adjust object placement in World

## Events

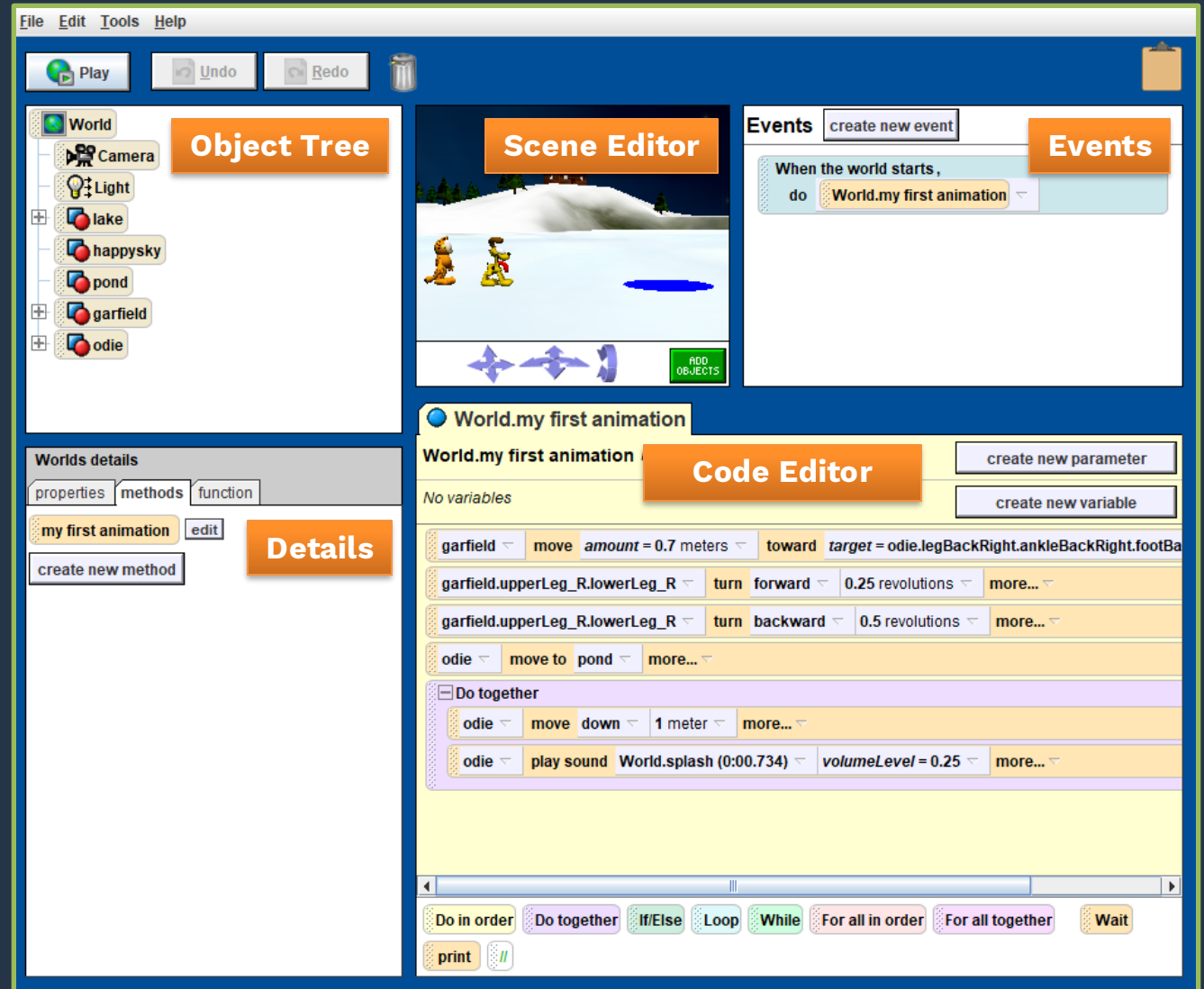
Associate methods with system or user input (mouse click, key stroke, etc.)

## Details

View methods, functions, and properties for object selected

## Code Editor

Build programs by dragging methods from Details area



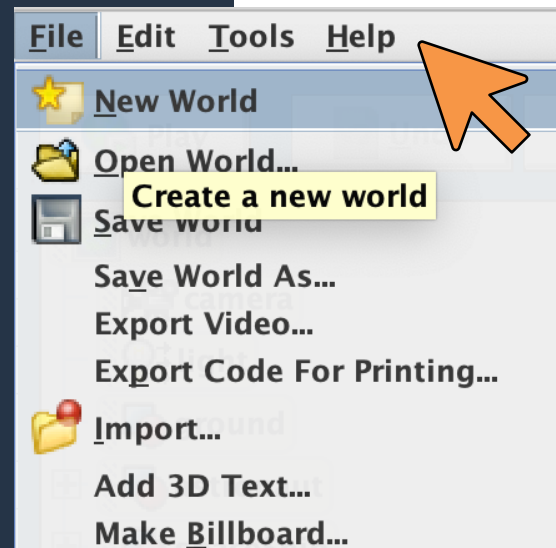
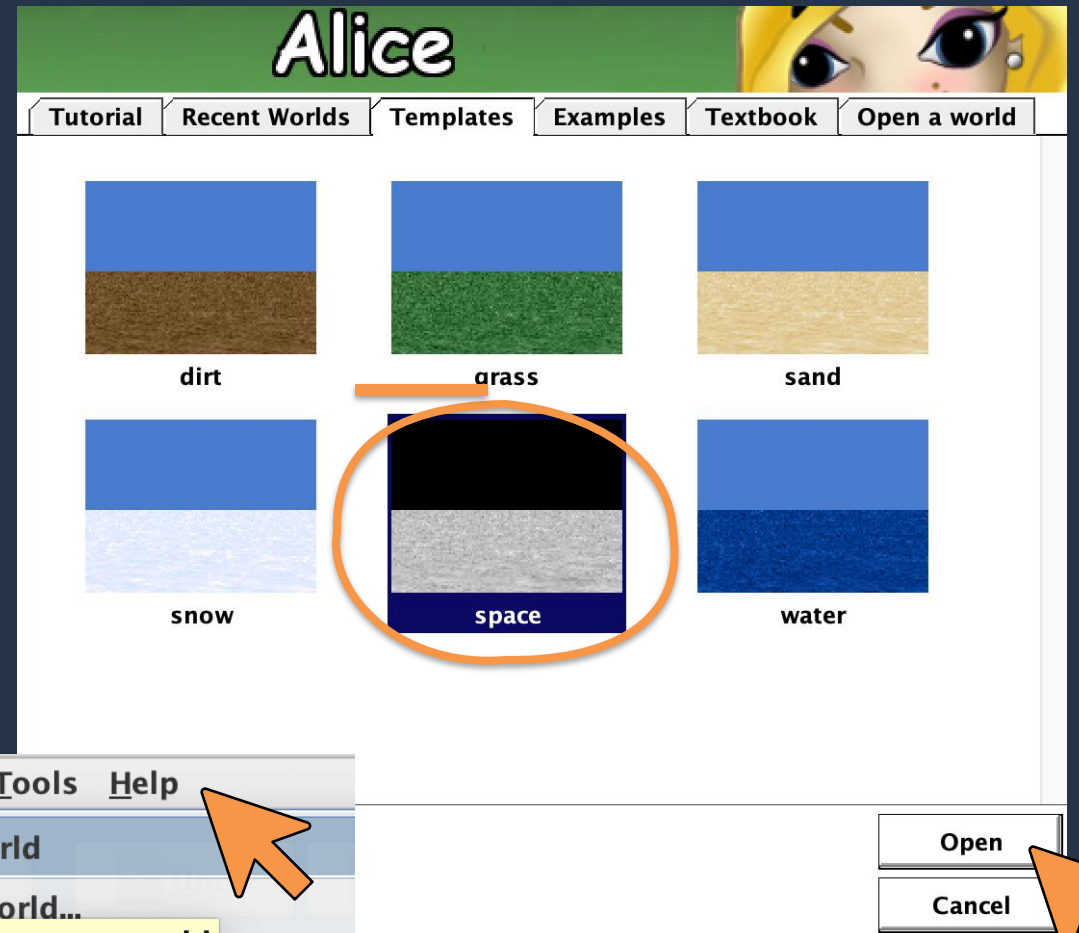
# Create New World

By default, this window appears when program starts.

Click **space**, then **Open**.

Or, from the menu bar:

**File > New World > Templates tab > space > Open**



# Add Objects



Click **ADD OBJECTS**

Scroll right (at the bottom)

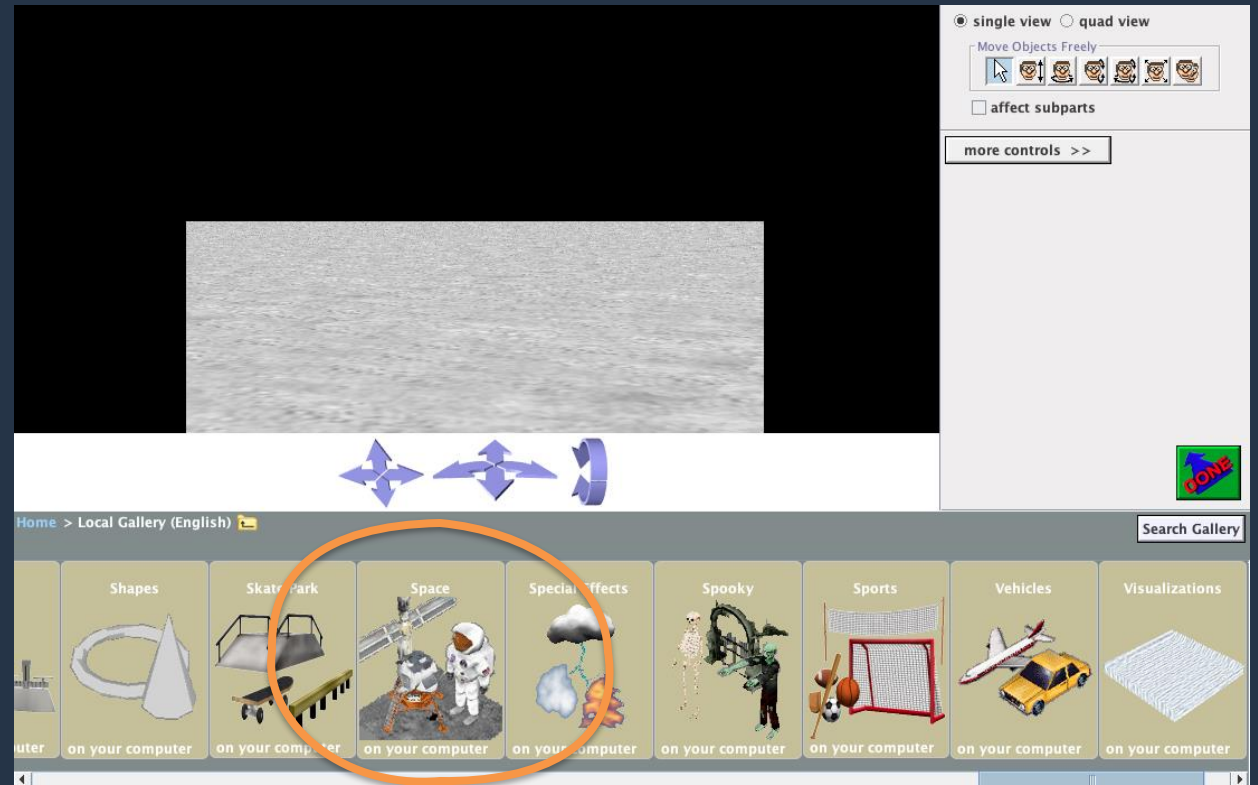
Select **Space**

Scroll right, select **Class Astronaut**

Click **Add instance to world**

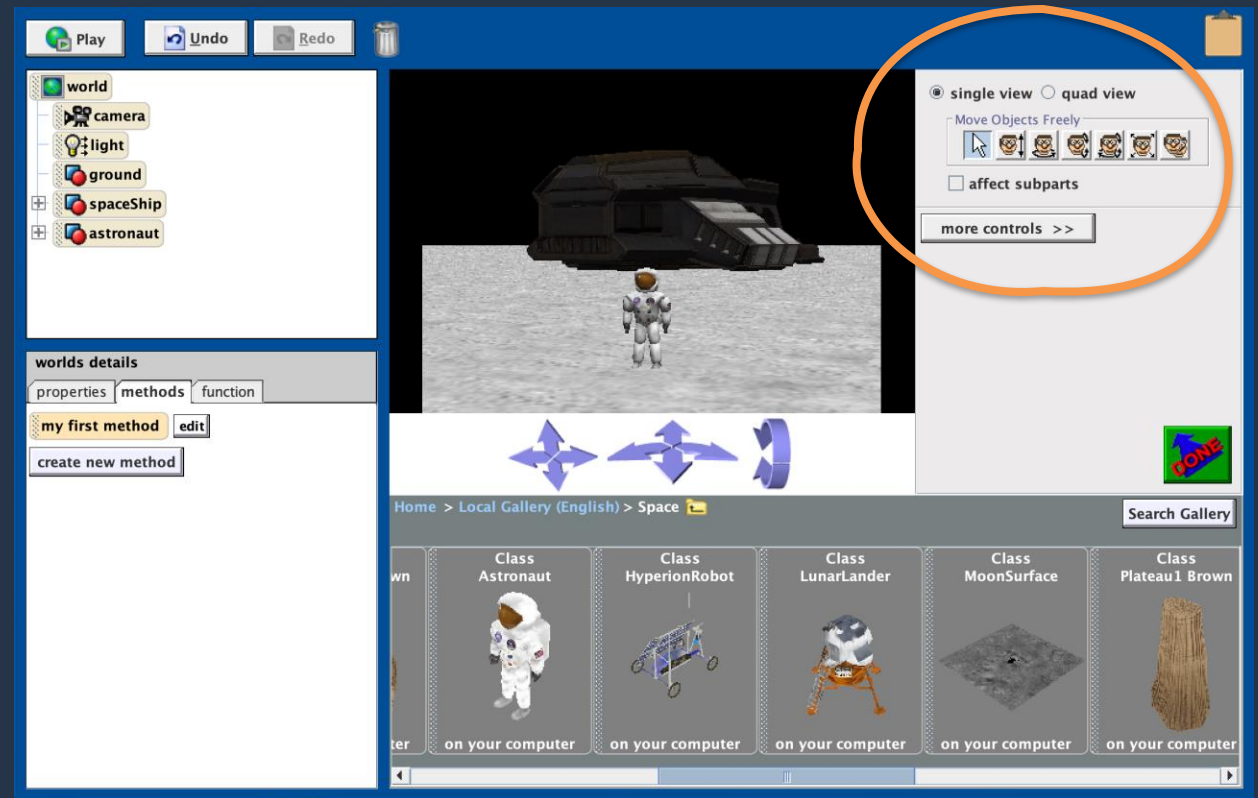
Select **Class SpaceShip**

Click **Add instance to world**



# Object Position

Before leaving the add objects menu, notice the various object positioning options. Take a few minutes to familiarize yourself with these toggles.

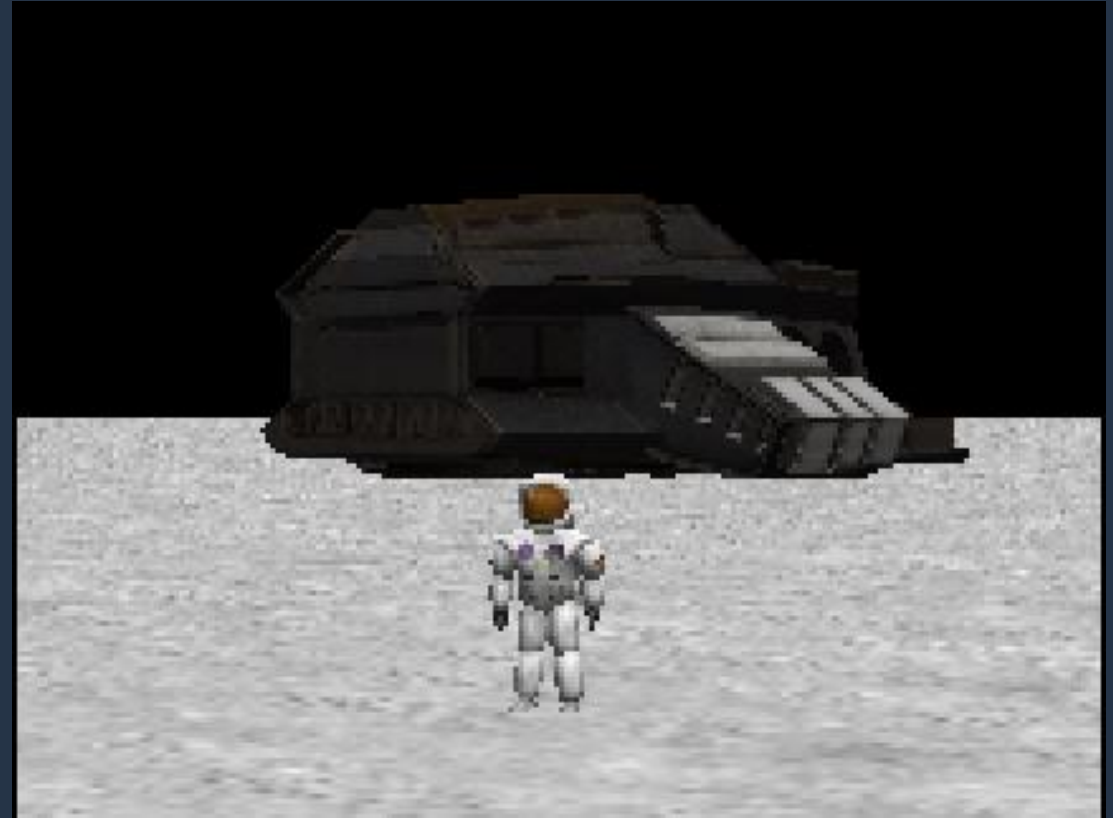


Click **DONE** when you are finished adding objects to the scene.



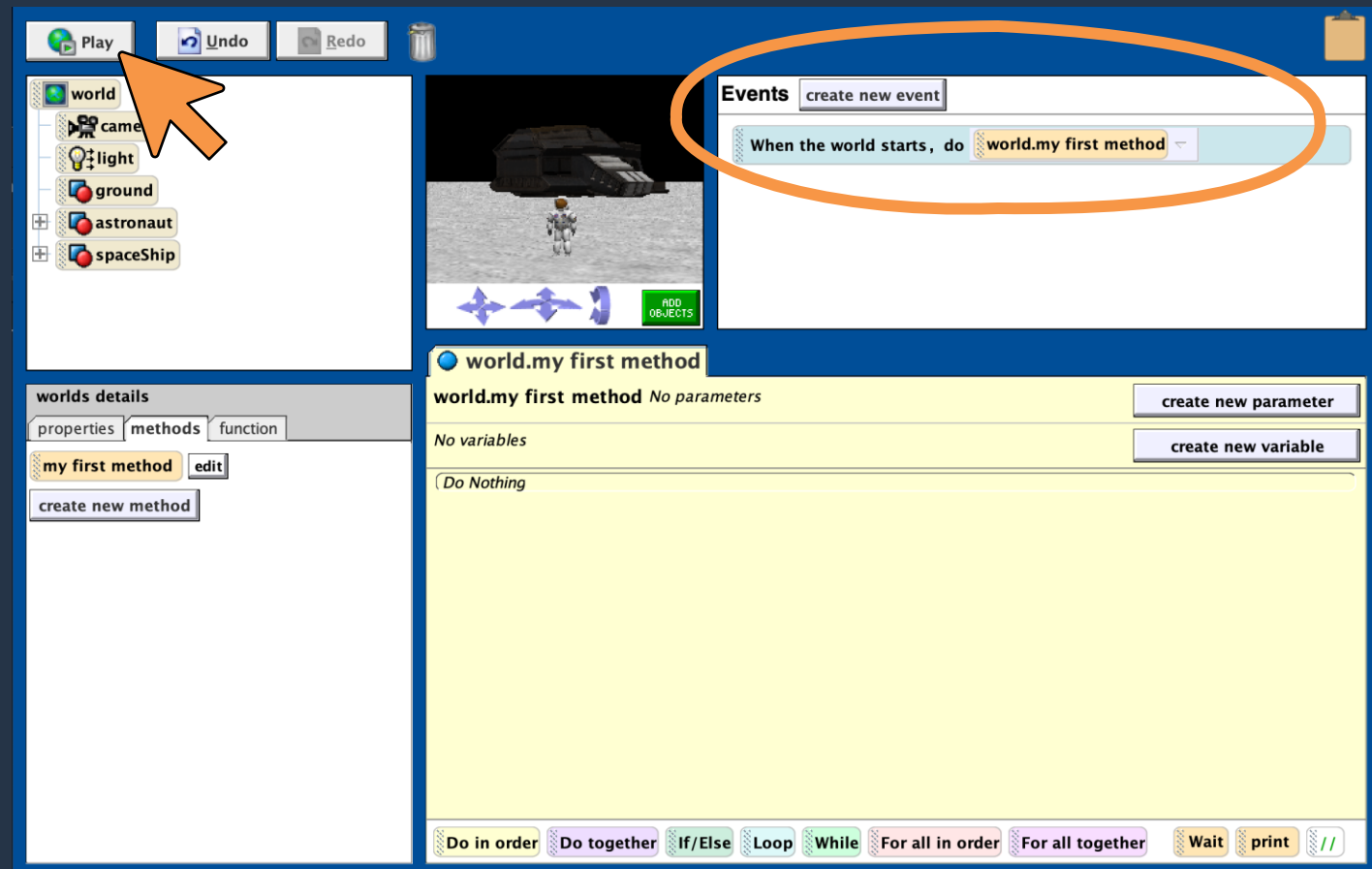
# Objects in Scene

Your scene editor should look like this, with a spaceship object and an astronaut object.



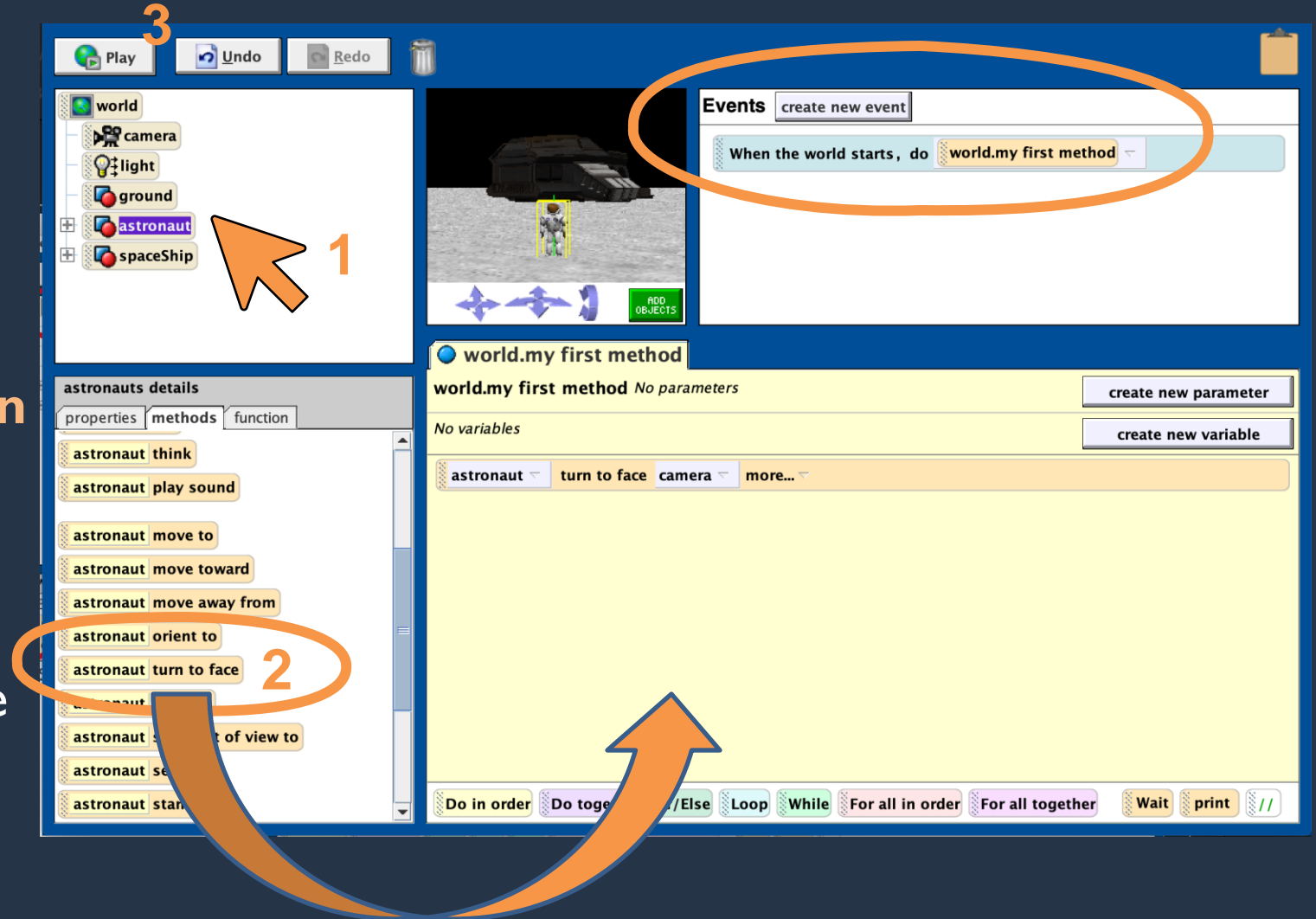
# Methods & Event Handling

- A **method** is used to invoke a set of instructions by name.
- An **event** is used to manage program flow and execution.
- Examples of **event** trigger:
  - A mouse click
  - Pressing a key on keyboard



# Add Method to Event

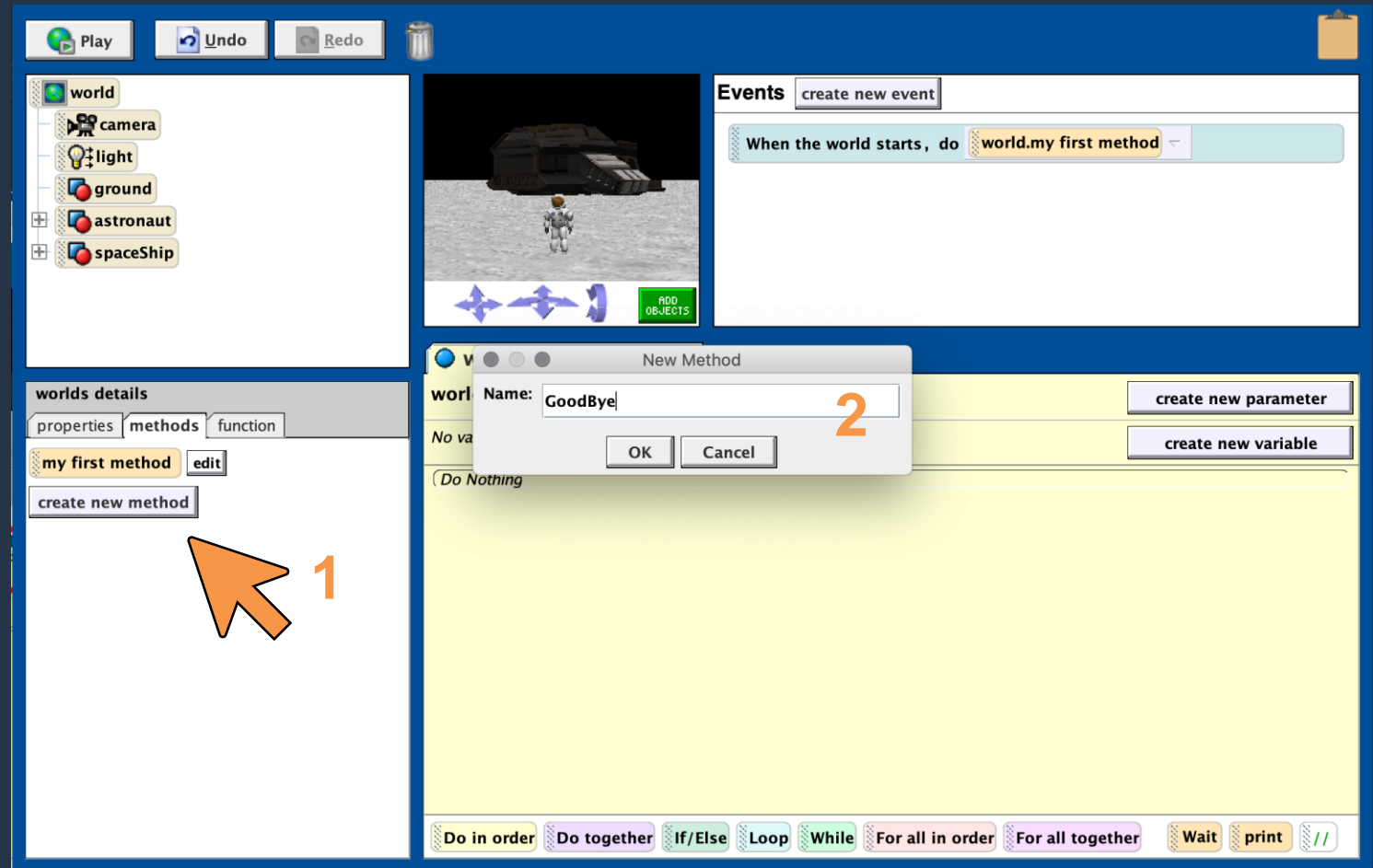
1. In **object tree**, click the astronaut object, and go to methods tab.
2. Click and drag the Method **turn to face** into Code Editor, and select camera.
3. Click **Play**. On initialization, the astronaut will turn to face the camera (if not already facing camera).





# Creating a New Method

1. Click **create new method**.
2. You can give it any meaningful name, for example "GoodBye" and click **OK**.



# Creating a New Method – Contd.

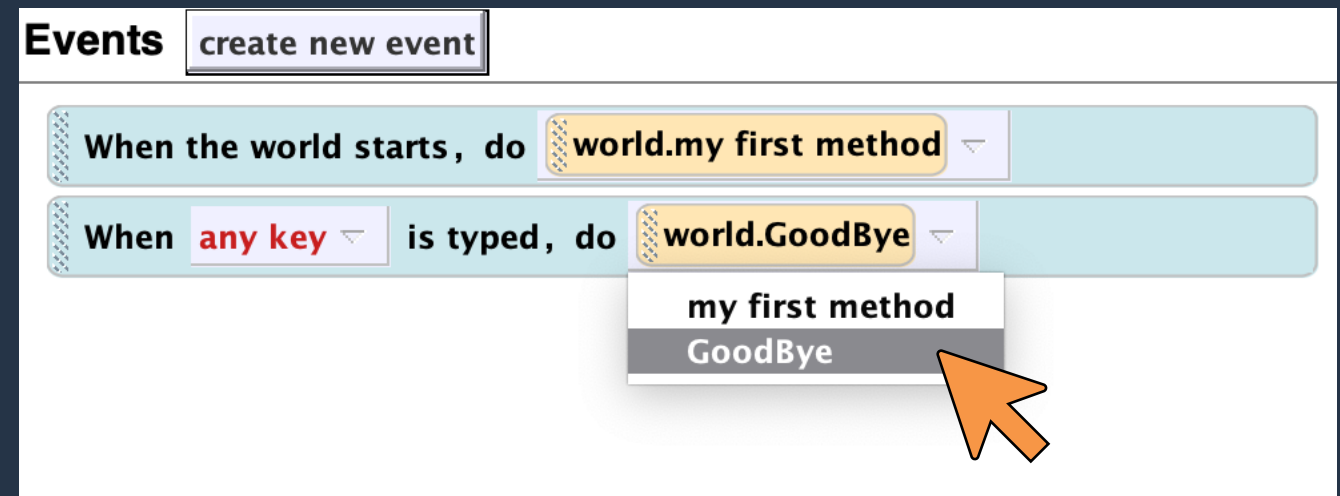
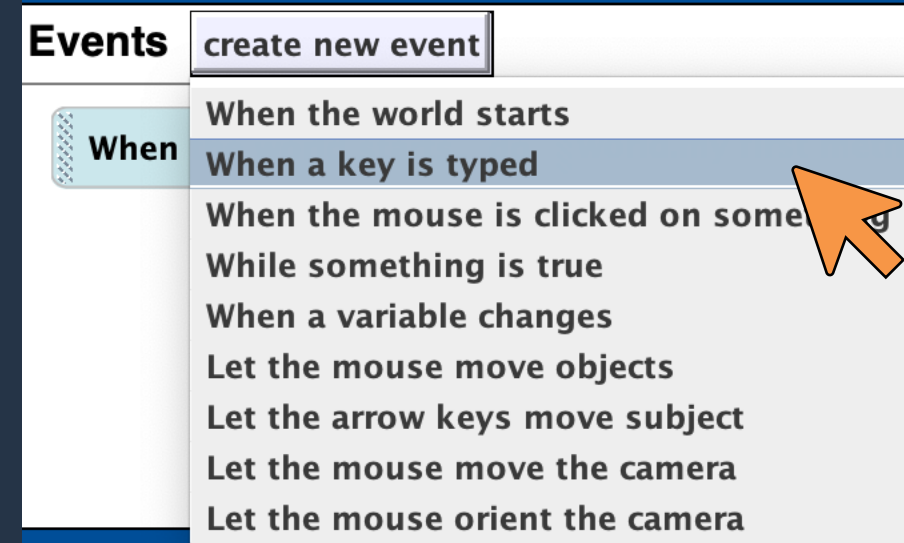
1. In objects tree, make sure astronaut object is selected.
2. In astronauts details, under methods tab, click and drag method **astronaut say** into the Code Editor, choosing the method you created (Goodbye) from dropdown menu.

The screenshot displays a programming environment with several panels:

- Objects Tree:** Located in the top-left, it lists objects: world, camera, light, ground, astronaut, and spaceShip. An orange arrow labeled '1' points to the 'astronaut' object.
- Stage:** The top-center panel shows a 3D scene with a spaceship and an astronaut on a moon-like surface.
- Events:** The top-right panel shows an event trigger: 'When the world starts, do world.my first method'.
- astronauts details:** The bottom-left panel has tabs for 'properties', 'methods', and 'function'. The 'methods' tab is active, showing a list of methods. The 'astronaut say' method is circled in orange and labeled '2'. A large orange arrow points from this method to the code editor.
- Code Editor:** The bottom-right panel shows the 'astronaut.GoodBye' method being edited. It has 'No parameters' and 'No variables'. The code block contains 'astronaut say Goodbye more...'. A dropdown menu is open, showing options: 'what', 'Hello', 'Goodbye' (highlighted in grey and labeled '4'), and 'other...'. A label '3' is placed near the dropdown.
- Bottom Panel:** Contains various control blocks like 'Do in order', 'If/Else', 'Loop', 'While', 'For all in order', 'For all together', 'Wait', 'print', and '//'. A large orange arrow points from the 'astronaut say' method in the details panel to the 'Goodbye' option in the dropdown menu.

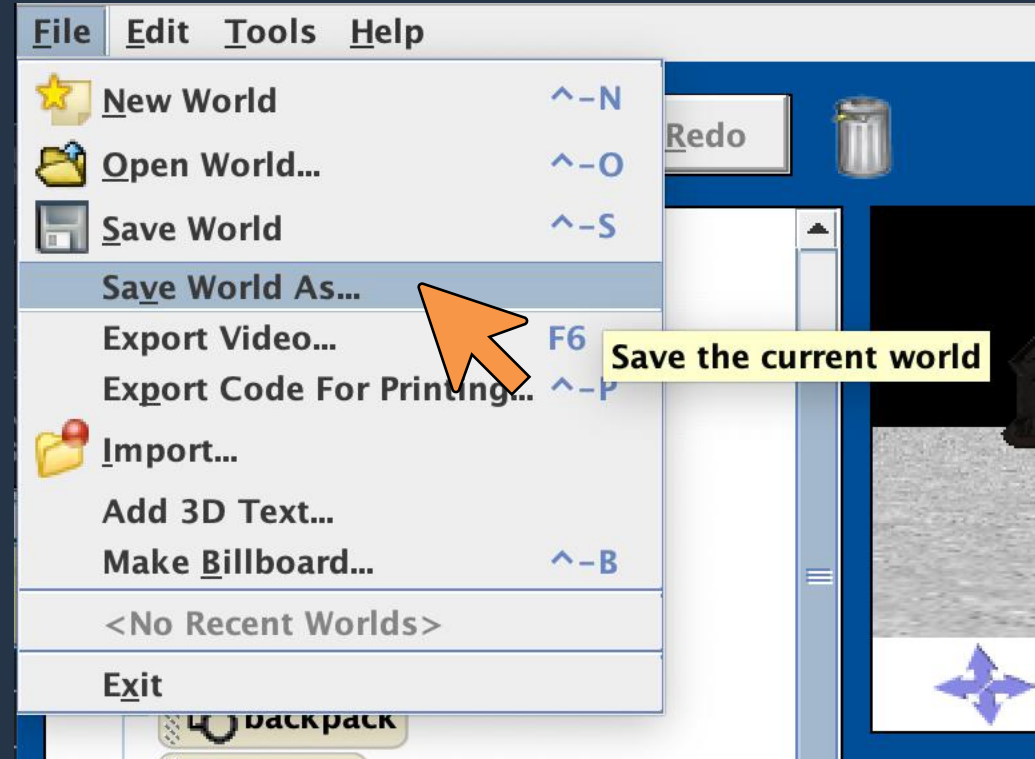
# Create Event

1. In Events pane, click **create new event**.
2. Select **When a key is typed**, making sure **any key** is configured.
3. Add method to event.
4. Add **GoodBye** method to this Event by using the dropdown menu.



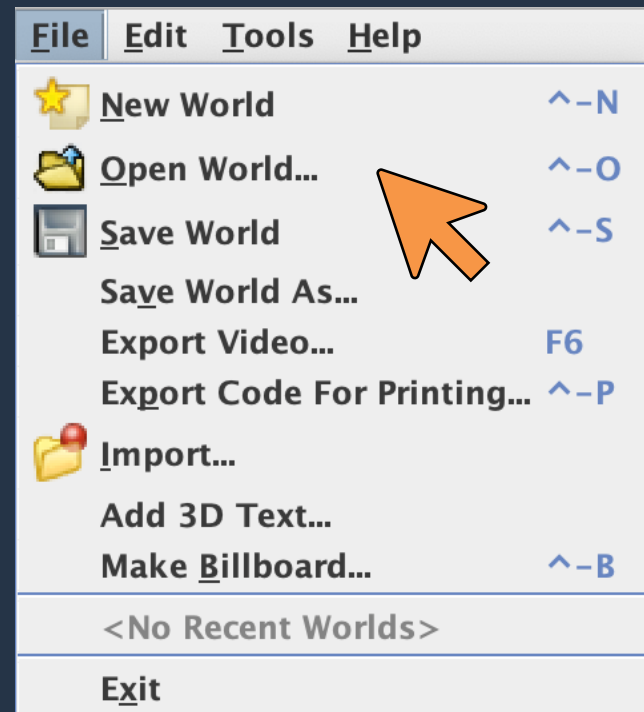
# Save World

File > Save World As... >



# Load World

File > Open World...



# Unexpected Journey

- Download the zip file from, and unzip it:

<http://razib.info/alice.zip>

- Load the **Unexpected Journey** Alice project.
- Take **15-20 minutes** to add what happens next in the story.



# **TEAM TIME:** **Classroom Application of Alice**

Facilitated by Dr. Dave Cornelison

**K-W-L-S**



# What are essential characteristics about a person who wants to learn about STEAM?

1. Write 3 words on a Post-It note
2. Place your Post-It on a Parking Lot

# **Integration of Alice & STEM:**

## **Gravity**

# Lunch

Food Vendors in Plaster Student Union (PSU)

# Code Stars



# Q&A with Amber

# **Alice Activity – Hour of Code**

After Lunch



# TEAM TIME: Curriculum Integration

Facilitated by Dr. Diana Piccolo

TEAM  
TIME



# Alice application ideas for your classroom

- Think-pair-share with your neighbor, 3 ways you can apply/teach/integrate Alice into your classroom.
- Be prepared to share:
  1. Grade level and content area for your Alice idea
  2. Describe your Alice application
  3. Get feedback from group...additional ideas?



# Alice application ideas:

## **Group 1:**

1. Grade level and content area for your Alice idea: 6-8; 4-5 science/contained
2. Describe your Alice application: Study the rock cycle; a history lesson on biographies; choose your own adventures-start character and then continue; math shapes 2-D/3-D

## **Group 2:**

1. Grade level and content area for your Alice idea: coding class(3-8); science –6th
2. Describe your Alice application: science-creating models based on the solar system; math-building visual representations for word problems (multi-step).

## **Group 3:**

1. Grade level and content area for your Alice idea: 9th ELE; 4th grade
2. Describe your Alice application: take a story, such as Romeo & Juliet and create your own version of this play; create own "world", such as a culminating activity to studying US states is to create your own "Jennifer state."

## **Group 4:**

1. Grade level and content area for your Alice idea; K-4
2. Describe your Alice application: Introduce vocabulary, such as sink, float; 3rd-phases of the moon;

# Alice application ideas:

## **Group 5:**

1. Grade level and content area for your Alice idea: 3-6th grade/ all subjects
2. Describe your Alice application: Plan writing in Matt's class and then actually code in a Special's class; or do both in your own class. Give students what if scenarios for students to predict and then run the actual Alice program with the results.

## **Group 6:**

1. Grade level and content area for your Alice idea
2. Describe your Alice application: multi-media class on video editing; expand into animation-storyboarding, story writing; create a fictional story in Alice and then present.

## **Group 7:**

1. Grade level and content area for your Alice idea
2. Describe your Alice application

## **Group 8:**

1. Grade level and content area for your Alice idea
2. Describe your Alice application

# GOALS: Curriculum Integration

- Introduce our framework for integration throughout the academic year.
- Describe integration support from the **Professional Development Field Initiation and Innovation Team (PD-FIIT)** cohort.
- Next steps in preparing for the academic year using Planbook.

# OUR FRAMEWORK FOR INTEGRATION

## How will **CODERS** affect what I'm teaching?

- Within your standard work week, you will add a pre-created STEM or CODING lesson into one of your lessons or assignments for that week, such as writing.
- Your goal is to work toward using at least one STEM or coding pre-created lesson a week-refer to Planbook for lesson bank.
- These pre-created lesson plans can be revised to better fit into your appropriate subjects, such as math, writing or science.

# OUR FRAMEWORK FOR INTEGRATION

## How will I access CODERS curriculum?

- Examples of pre-created lessons will be demonstrated during the Planbook session on Thursday.
- During the academic year, there will be weekly Zoom help desk times available, staffed by graduate students.
- The MSU PD FIIT cohorts will have monthly de-briefing sessions, either in person at the school or through Zoom, as needed.

# OUR FRAMEWORK FOR INTEGRATION

## Visitation Schedule Format

- All teachers from the same school will be visited on the same day by the MSU PD FIIT cohort leader.
- Teachers from each school will coordinate and use the CODERS scheduling calendar located on the CODERS website.
- The PD FIIT cohort is comprised of the CODERS team leader and assigned teachers from each school. For example, Dr. Piccolo's cohort is comprised of teachers from the Neosho, Ava, Mtn. Grove & Logan-Rogersville schools.
- You will have visit(s) by the MSU PD FIIT cohort leader each semester.

# OUR FRAMEWORK FOR INTEGRATION

## Remember...

- The CODERS project this year is in its initial year one implementation stage.
- We encourage you to provide feedback to your MSU PD FIIT cohort leader on ways to improve any part of this project curriculum delivery throughout the academic year (we always like to hear what you like, too).
- This week's CODERS Launch workshop is just the beginning. We are excited to work with you in your classroom and collaborate with others at your school.
- An action plan of "to do" items for you (and us) will be discussed Friday. We want you to leave this week knowing the expectations required of you and for the MSU PD FIIT cohort.

# Integration ideas and questions

**1. If I have 90 students and just teach science, do I teach these concepts to just one class or all of them?**

**A: You don't have to grade anything to turn into us but the assessment group will collect data. We will further discuss ways to integrate STEM and coding to one subject classrooms.**





# **WWC Strategies: Writing Integration**

Facilitated by Dr. Franklin and Dr. Davis



# Assessment: Andrew & Judith

A tool for feedback,  
accountability & improvement

# Plus & Delta Assessment Tool

## Template:

[https://fhop.ucsf.edu/sites/fhop.ucsf.edu/files/custom\\_download/ACPS\\_Plus\\_Delta\\_Template.pdf](https://fhop.ucsf.edu/sites/fhop.ucsf.edu/files/custom_download/ACPS_Plus_Delta_Template.pdf)

## Today's Plus & Delta:

<https://bit.ly/3xgsdGP>



**QR: Today's  
Plus & Delta**

# QUESTIONS? SUGGESTIONS?

- How do you add curriculum ideas into your lessons?
- Answer this question on a sticky note and place on the front whiteboard...near Judith (aka...Vanna White)