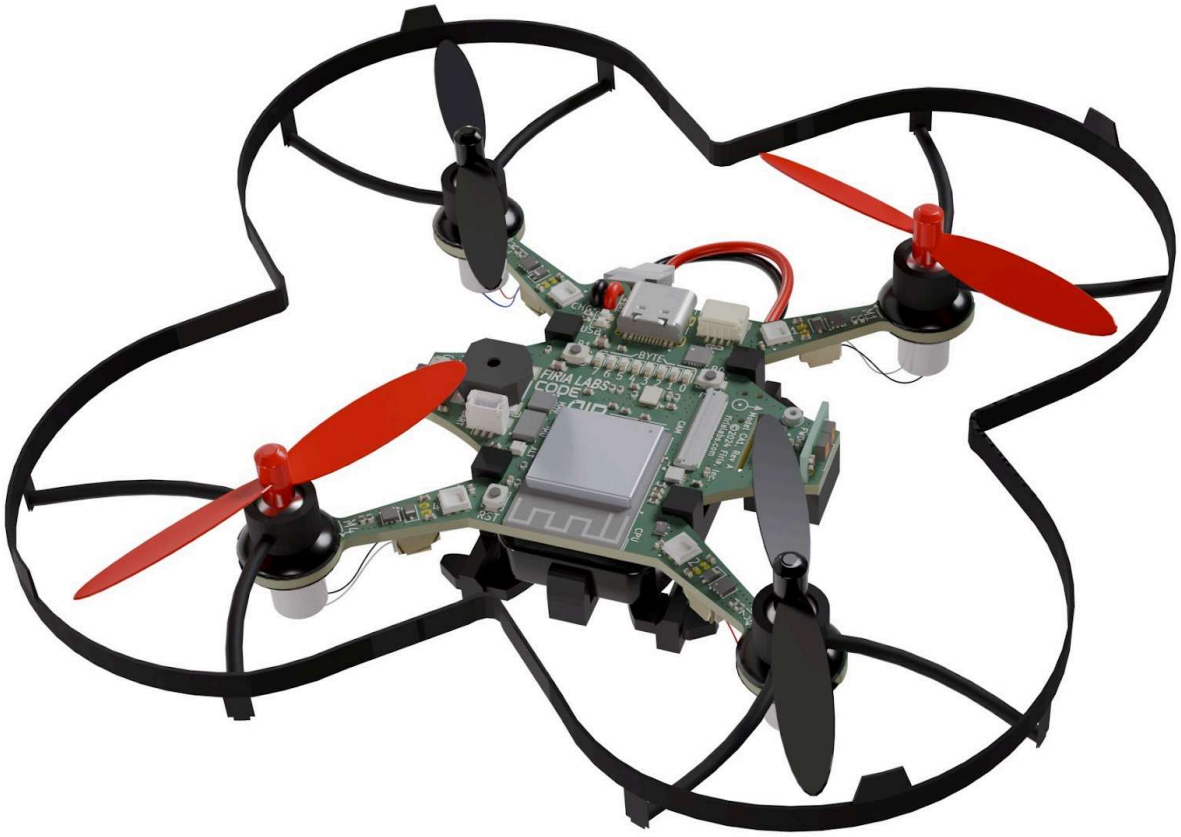
FIRIA LABS

Curriculum Guide

Preview



Mission Pack:

Fly with Python

Table of Contents

Fly with Python Overview

This mission pack teaches Python coding by controlling our next-generation autonomous educational drone, designed to integrate AI and Python programming into the classroom. This curriculum will inspire students with engaging STEM projects that blend advanced technology with hands-on learning.

CodeAIR and the Fly with Python Mission Pack revolutionize learning with its onboard neural network and camera. This allows students to deploy machine learning models for real-time object and pattern recognition. CodeAIR operates autonomously, providing a seamless and interactive coding experience without the need for constant radio control.

Pre-Mission Assignment

If your students come with no Computer Science background, it is important to start by building a foundation of computational thinking. Dedicate some time for students to learn basic terms, such as algorithm, program, and debug. See the Firia Labs collection of Unplugged Activities at https://learn.firialabs.com/curricula/cs-unplugged.

Mission 1: Welcome

Take a tour of the CodeSpace Development Environment. Learn how to use the text editor, hints, CodeTrek and tools.

Mission 2: Introducing CodeAIR

Students get an introduction to CodeAIR. The device gets connected, and students run some code!

Mission 3: Pre-Flight Check

All systems GO for takeoff? Test out some basic Python coding to control CodeAIR's lights and sounds!

Mission 4: Flight Safety

Good drones don't jump into the sky unexpectedly! Code a safety interlock to be sure your CodeAIR stays put until you are ready to fly.

Mission 5: Hovering Flight

Take off! Get your drone in the air for an autonomous hover, and learn more Python coding as you explore the Laser Range sensors.

Mission 6: Navigate

Navigate the skies by getting a fix on the ground with CodeAIR's flow sensor. Explore the capabilities and limitations of onboard positioning systems.

Mission 7: Attitude Control – coming soon!

Check your attitude, dude! No, not your demeanor...I'm talking about your drone's airframe here!

Suggested Pacing Guide

The way you present the Mission Pack and its missions, objectives and assignments is up to you. Students can work independently and at their own pace to complete the missions. Or you can work together as a class and engage in extensions and cross curricular activities with the missions. A suggested timeline is provided below. It includes time for students to explore and investigate the code. Completing the mission pack can take more or less time, depending on your students, their interests and previous experience, the time you have to spend on it, and many other factors. It is up to you!

This timeline is based on a 5-day, 45-minute class period.

Week 1	Unplugged activities: See CS Unplugged in the learning portal				
Week 2	Mission 1 Mission 1 Review	Mission 2 Mission 2 Review	Mission 2 Extensions or cross curricular	Mission 3 Obj 1-4 & Q2	Mission 3 Obj 5-6
Week 3	Mission 3 Obj 7 and Q3	Mission 3 Obj 8	Mission 3 Extensions or cross curricular	Mission 3 Review Unit 1 Remix Step 1	Unit 1 Remix Step 2-3
Week 4	Unit 1 Remix Step 4	Unit 1 Remix Step 5	Unit 1 Review / Assessment	Mission 4 Obj 1-4	Mission 4 Obj 5-6 / Review
Week 5	Mission 4 Extensions or cross curricular	Mission 5 Obj 1-2	Mission 5 Obj 3	Mission 5 Obj 4-5	Mission 5 Obj 6
Week 6	Mission 5 Obj 7, Q2	Mission 5 Obj 8	Mission 5 Obj 9, Q3	Mission 5 Obj 10	Mission 5 Extensions or cross curricular
Week 7	Mission 5 Review Unit 2 Remix Step 1	Unit 2 Remix Step 2-3	Unit 2 Remix Step 4	Unit 2 Remix Step 5	Unit 2 Review / Assessment
Week 8	Mission 6 Obj 1	Mission 6 Obj 2, Q1	Mission 6 Obj 3	Mission 6 Obj 4, Q2	Mission 6 Obj 5
Week 9	Mission 6 Obj 6, Q3	Mission 6 Obj 7	Mission 6 Extensions or cross curricular	Mission 6 Review Extensions	
Week 10					
Week 11					
Week 12					
Week 13					
Week 14					
Week 15					
Week 16					

Mission 1: Welcome	**Time Frame:** 20-30 minutes
Mission Goal: Students will learn about the CodeSpace learning environment. **Learning Targets** • I can navigate CodeSpace. • Identify major parts of the Codespace interface: Mission Bar, Objective Panel, text editor, CodeTrek, Toolbox, and Lesson Navigation Controls	**Key Concepts** • Follow instructions in the Lesson Panel carefully. There is a lot of important reading! • Look for "tool icons" to collect tools in your Toolbox as you go.
Assessment Opportunities • Quiz after Objective 4 • Add tool to toolbox (Objective 3) • Review questions	**Success Criteria** ☐ Complete each Objective Goal ☐ Complete Mission 1 Assignment

Standards

CSTA Standards Grades 9-10	CSTA Standards Grades 11-12	AI4K12 Standards Grades 9-12
• 3A-IC-27	• 3B-AP-20	

Student Materials	**Teacher Resources**
• Laptop/computer with Chrome browser • Student account / Flying with Python license • Getting Started with CodeSpace (slides) • CodeAIR Mission 1 Assignment (PDF)	• License and Dashboard Resources • CodeAIR Mission 1 Assignment Answers • CodeAIR Mission 1 Review Questions

Vocabulary

Objective	The steps in the mission; has a goal to accomplish.
Text editor	Where you type the code.
Debugging	The process of understanding what the computer is actually doing and then changing the code to do what you want it to do.
Toolbox	A place in CodeSpace to keep information you learn about programming concepts so you can use it later when you need the information.
Simulation	A 3D environment that lets you see the robot move and interact in a virtual world.

New Python Code

Teacher Notes

- Create a class on the teacher dashboard. Generate a join code for the class section to give students.
- This lesson is the first lesson in all the mission packs. If your students have completed other Firia mission packs, they will already know the information. You can choose to have them complete the mission as a review and refresher, or you can unlock the next mission.
- Review questions can be used as a class review, made into a Kahoot!, or used to create an exam in your learning management system.

Extensions	**Cross-Curricular**
• none	• none