

Lesson 16: Project - Prototype an Innovation

225 minutes

Overview

In this final project for the course, students team to develop and test a prototype for an innovative computing device based on the micro:bit. Using the inputs and outputs available on the board, groups will create programs that allow for interesting and unique user interactions.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems

Agenda

Warm Up (5 minutes)

Review Project Guide

Activity (210 minutes)

Define - Scope Innovation

Prepare - Complete Project Guide

Try - Develop Prototypes

Reflect - Peer Review

Iterate - Revise Prototypes

Wrap Up (10 minutes)

Share

Extensions

Pitch Video

Marketing Website

Crowdfunding Campaign

Objectives

Students will be able to:

- Implement a plan for developing a piece of software that integrates hardware inputs and outputs
- Independently scope the features of a piece of software
- Prototype a physical computing device

Preparation

- Collect materials for physical prototyping, eg.
 - Cardboard
 - Scissors
 - Tape
 - Glue
 - Foil
- Print a copy of the project guide for each pair of students
- Print a copy of the peer review sheet for each student
- Print a copy of the rubric for each student
- Check the "**Teacher's Lounge**" forum for verified teachers to find additional strategies or resources shared by fellow teachers

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the teachers

- **Project: Prototype an Innovation** - Slides [▼ Make a Copy](#)

For the students

- **Peer Review - Prototype an Innovation** ▼ Make a Copy
- **Prototype an Innovation** - Rubric ▼ Make a Copy
- **Prototype an Innovation** - Activity Guide ▼ Make a Copy
- **Prototype an Innovation - Student Checklist** ▼ Make a Copy

Teaching Guide

Warm Up (5 minutes)

Review Project Guide

Group: Place students into groups of 2-4 for this project. At your discretion you may choose to have students form larger groups or work independently.

🔗 Teaching Tip ▲

Facilitating Group Projects: If students are working in pairs or small teams to complete projects, consider showing these two videos to the class:

- **How Teamwork Works**
- **Dealing with Disagreements**

Depending on your goals with this project, consider having teams complete a **Student Guide to Team Planning**, which reinforces the message in the video

Distribute: Provide a copy of the project guide to each group. As a class, review the different steps of the project and where they appear in the project guide.

Distribute the rubric or student checklist to each student. Have them look over this resource so that they know from the beginning what components of the project you will be looking for.

🔗 Teaching Tip ▲

Rubric and Checklist: Students have two resources they can use for self-reflection and making sure they are on the right track: the rubric and the student checklist. We recommend having students use the checklist for their own self-assessment and reflection, since it may be easier to digest and understand when reviewing their own project. However, we recommend teachers use the full rubric for evaluating projects to give more accurate feedback to students. You can see examples of this with the Sample Marked Rubrics resource at the top of the lesson plan (only visible to verified teachers)

Activity (210 minutes)

Define - Scope Innovation

Brainstorm: Students should spend the first 15-20 minutes brainstorming ideas for innovative devices built around the features of the micro:bit. Encourage students to review their prior work in this unit as well as the real world innovations they researched earlier for inspiration.

Prepare - Complete Project Guide

Distribute: Make available any construction materials that students may need for building their prototypes. While the focus of this step isn't to build the actual prototype, having these materials available can help with the brainstorming process.

Circulate: Once students have discussed their ideas for the project they should complete the project guide. While this should be a fairly familiar process, encourage students to make each component as clear and detailed as they can at this point. Planning ahead can help them identify issues in their plan before they'll need to make more significant changes to their code or physical device. Encourage students to use additional paper to sketch out screens for their app or additional views of their physical device.

Try - Develop Prototypes

Distribute: Make available the physical prototyping materials (such as cardboard, tape, scissors, etc). Let students know that, just as we used paper prototypes to quickly test software ideas, hardware developers often used cheap materials such as cardboard and tape to quickly iterate on the design of physical devices. While not all student ideas may require a physically prototyped component, you should encourage students to consider how the shape and design (or form factor) or their innovation could impact its usability.

Transition: Depending on the nature of their innovations, students may need to spend some time building the physical components of their projects before moving online. When students are ready to program, they can transition to Code Studio. These levels provide some guidance on how students may go about implementing their projects, but are left quite open to allow for a broad range of ideas. If they wish, students can work in a different order than the one suggested in these levels.



1-4

Prototype an Innovation

1

2

3

4

Teaching Tip



Debugging Strategies: As students design and implement their own project ideas, they may find themselves with new bugs that they need to untangle and you may find yourself looking at completely unfamiliar code as students look for help troubleshooting their errors. To help smooth out the debugging experience, consider the following strategies:

- Review the **Teacher Guide to Debugging** for some common questions and strategies to help support students in debugging their code
- Have students follow the steps in the **Student Guide to Debugging** and use the **Bug Report Quarter-Sheets** as an initial step in the debugging process. This helps students prepare and communicate their issue before asking for help.

- If students haven't seen it yet, consider showing the [Debugging Video](#) to the class to reinforce debugging best practices.

Digging Deeper: Consider supplying students with an object to talk to as part of the debugging process. This is sometimes known as Rubber Duck Debugging - you can learn more on the website <https://rubberduckdebugging.com/>

Reflect - Peer Review

Group: Pair up groups to review each other's projects.

Distribute: Give each student a copy of the peer review guide. Students should spend 15 minutes reviewing the other group's project and filling out the peer review guide.

Iterate - Revise Prototypes

Circulate: Students should complete the peer review guide's back side where they decide how to respond to the feedback they were given. They should then use that feedback to improve their innovation.

Wrap Up (10 minutes)

Share

Share: Give students a chance to share their innovations, either within the class or to a broader audience. If you choose to let students do a more formal presentation of their projects the project guide provides students a set of components to include in their presentations including:

Teaching Tip

Celebrate: As this is the culminating project for the entire course, consider going big with this share out. Bring in parents and administrators, or even host a after school event to provide students with a real audience to share their accomplishments with.

- The original innovation they set out to build
- A description of the programming process including at least one challenge they faced and one new feature they decided to add
- A description of the most interesting or complex piece of code they wrote
- A live demonstration of the actual innovation

Send students to Code Studio to complete an end of unit survey. Although their answers are anonymous, the aggregated data will be available to you once at least five students have completed the survey.

Extensions

Pitch Video

Record and edit a short video to pitch your innovation.

Marketing Website

Using Web Lab, design a website to market your innovation.

Crowdfunding Campaign

Have students design a crowdfunding campaign (along the lines of Kickstarter or Indiegogo) for their innovations. This could include:

- Design mockups for the final product
- A rough cost analysis for production
- A short pitch video



This work is available under a **Creative Commons License (CC BY-NC-SA 4.0)**.

If you are interested in licensing Code.org materials for commercial purposes **contact us**.