

Lesson 9: Project - Make a Game

Overview

For this project, students design and create a game that leverages the new inputs and outputs that are available to them. This project is purposefully left very open-ended to empower students to think broadly about how physical output might be useful in an app, while still giving them a chance to review the program development process and try out the new features available through the Circuit Playground.

Purpose

This end of chapter assessment is a good place for students to bring together all the pieces they have learned (event handlers, using the board as output, etc.) in one place. This project is purposefully left very open to allow students to think broadly about how physical output might be useful in an app - this is a great opportunity to encourage students to revisit programs they've written earlier in this unit or in Unit 4 that would benefit from using the board to output information.

Assessment Opportunities

Use the project rubric attached to this lesson to assess student mastery of learning goals of this chapter.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (5 minutes)

Demo Project Exemplars

Activity (125 minutes)

Unplugged: Program Planning Prototyping the Program

Wrap Up (5 minutes)

Sharing Projects Self Assessment

Objectives

Students will be able to:

- Design a piece of software that uses hardware for non-traditional input and output
- Prototype a program that integrates software and hardware
- Use event handlers to respond to user interaction

Preparation

- Print a copy of the project guide for each group of students
- Print a copy of the rubric for each student

Links

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

For the teachers

- **CSD Unit 6 - Physical Computing**
- Slides

For the students

- **Computer Science Practices** - Reflection
- **Make a Game** - Activity Guide
- **Make a Game** - Rubric
- **Make a Game** - Student Checklist

Teaching Guide

Warm Up (5 minutes)

Demo Project Exemplars

Display: On the projector, show the example project at the beginning of this lesson.

Prompt: Analyze the exemplar for the following elements:

- Where is input being taken?
 - What events do you think are being used for each input?
- What information is being output through the board?
- Where might this program be using a loop?

Discuss: Have students share their observations and analyses of the exemplar.

Distribute: Provide each student with a copy of the rubric. Review the different components of the rubric with them to make sure they understand the components of the project.

Activity (125 minutes)

Unplugged: Program Planning

Group: Place students in groups of 2-3.

Distribute: Hand out the project guide to students. This is the tool students will use to scope out their projects before getting onto the computers. Give students some time to brainstorm the type of program they want to make.

Make a Game Project Guide

Brainstorming

Give groups some time to brainstorm ideas for their project. The goal is to come up with a simple game that can be controlled using elements of the Circuit Playground. Encourage students to think about all of the programs and projects that they've made so far as potential starting points (including the prototypes from Unit 4).

1. Once groups have settled on an idea, they can record it on the activity guide under Project Brainstorming.
2. Next students think through the Events they'll need to respond to.

💡 Teaching Tip

Students may find it particularly difficult to predict *all* of the events that they'll want to respond to or the functions that they'll need to create. Make sure that they know this is an *iterative* process and planning is only the first step. We want to start this project with as clear a plan as possible, but there will likely be things that weren't considered from the start and the plan will need to change accordingly.

3. Finally students consider the functions that they may need to create.

Prototyping the Program

Transition: Once teams have completed their planning sheet, they can head to Code Studio to work on prototyping their projects.

 1

Demo - Bug Grab

 2

Plan Your Project

 3

Screen Design

 4-5

Events

4

5

 6-7

Functions

6

7

 8

Finishing Touches

Wrap Up (5 minutes)

Sharing Projects

Share: Find a way for groups to share their projects with each other. It will likely be helpful to use the share link for the project so that students can share the project with other students.

Self Assessment

Using the rubric, students should assess their own project before submitting it.

Send students to Code Studio to complete their reflection on their attitudes toward computer science. Although their answers are anonymous, the aggregated data will be available to you once at least five students have completed the survey.

 9

Reflection

