

Lesson 27: Project - Design a Game

Overview

Question of the Day: How can the five CS practices (problem solving, persistence, communication, collaboration, and creativity) help programmers to complete large projects?

Students will plan and build their own game using the project guide from the previous two lessons. Working individually or in pairs, students will first decide on the type of game they'd like to build, taking as inspiration a set of sample games. They will then complete a blank project guide where they will describe the game's behavior and scope out the variables, sprites, and functions they'll need to build. In Code Studio, a series of levels prompts them on a general sequence they can use to implement this plan. Partway through the process, students will share their projects for peer review and will incorporate feedback as they finish their game. At the end of the lesson, students will share their completed games with their classmates. This project will span multiple classes and can easily take anywhere from 3-5 class periods.

Purpose

This lesson is the culmination of Unit 3 and provides students an opportunity to build a Game Lab project of their own from the ground up. The scaffolding provided by the project guide and the practice they have using it are intended to assist students in scoping their projects and seeing their ideas through to completion. This project is an opportunity to showcase technical skills, but they will also need to collaborate with their partner, provide constructive peer feedback, and repeatedly use the problem solving process as they encounter obstacles along the way. This project should be student-directed whenever possible, and provide an empowering and memorable conclusion to this unit of CS Discoveries.

Assessment Opportunities

Use the project rubric attached to this lesson to assess student mastery of learning goals of this unit. You may also choose to assign the post-project test through Code Studio.

Standards

Full Course Alignment

Objectives

Students will be able to:

- Create a plan for building a piece of software by describing its major components
- Implement a plan for creating a piece of software
- Independently scope the features of a piece of software

Preparation

- Print copies of the project guide, one for each student / pair of students
- Print copies of the rubric, one for each student / pair of students
- Print copies of the peer review guide, one for each student / pair of students
- Review sample games in Code Studio

Links

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

For the teachers

- **CSD Unit 3 - Interactive Animations and Games** - Slides
- **Extra Code in Challenge Levels** - Resource

For the students

- **Computer Science Practices** - Reflection
- **Make Your Own Game** - Project Guide
- **Make Your Own Game** - Peer Review
- **Make Your Own Game** - Rubric

Agenda

Lesson Modifications

Warm Up (5 minutes)

Activity (215 minutes)

Review Project Guide

Define - Scope Game

Prepare - Complete Project Guide

Try - Write Code

Reflect - Peer Review

Iterate - Update Code

Share

Wrap Up (5 minutes)

Journal

Reflect

End of Course Survey

Post-Project Test

Teaching Guide

Lesson Modifications



Attention, teachers! If you are teaching virtually or in a socially-distanced

classroom, please **click here** to access modifications that can be used during this lesson.

Warm Up (5 minutes)

Prompt: Today, you'll start the final project of the unit, in which you will design and code your own game. Before you start, what are three skills or qualities that you think will be important as you complete this project?

Share: Allow students to share out their ideas.

Discussion Goal

This discussion serves to set cultural norms for the project. Students should expect that the project will be challenging, but that they will have plenty of opportunities to iterate on their work as they work together to learn as a community.

Remarks

There are lots of practices that programmers use when working on large projects. As you design and build your game, try to reflect on how you are using the five practices of problem solving, persistence, communication, collaboration, and creativity.

Question of the Day: How can the five CS practices (problem solving, persistence, communication, collaboration, and creativity) help programmers to complete large projects?

Activity (215 minutes)

Review Project Guide

Group: This project can be completed individually or in pairs. At your discretion, you may choose to have students form larger groups as well.

Distribute: Each student or group of students should be given a copy of the project guide. As a class, review the different steps of the project and where they appear in the project guide.

Distribute: Give each student a copy of the rubric or student checklist so that they know from the beginning what components of the project you will be looking for.

Define - Scope Game

Circulate: Students should spend the first 15-20 minutes playing the sample games, reviewing past work, and discussing as a group the type of game they'd like to build. If they want they can sketch ideas on scratch paper or in their journals.



Sample Games

Prepare - Complete Project Guide

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. Planning ahead can help them identify issues in their plan before they'll need to make more significant changes to their code.

Try - Write Code

Transition: Students are now ready to program their games on Code Studio. These levels provide some guidance on how students may go about implementing their project guide. None of the steps are significantly different from what students have seen from the previous two lessons. If they wish, students can work in a different order than the one suggested in these levels.

💡 Teaching Tip

New Code and Previous Challenge Levels: Throughout the unit, students may have learned additional code in the challenge levels of different lessons. As students approach this project, they may want to revisit the code they learned in these levels or visit them for the first time to learn new codes to use in this project. To help guide students back to previous levels, you can use the **Extra Code in Challenge Levels** as a resource to quickly find where new codes were introduced in earlier challenge levels.

**2-5**

Project - Background and Variables

2

3

4

5

**6-10**

Project - Sprites and Interactions

6

7

8

9

10

Reflect - Peer Review

Distribute: Give each student a copy of the peer review guide.

Students should spend 15 minutes reviewing the other group's game and filling out the peer review guide.

**11**

Review Your Game

Iterate - Update Code

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

Students should refer to the rubric or student checklist to ensure they meet the requirements of the project.

💡 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)

Share

Share: Give students a chance to share their games. 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:

- The original game 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 game

Wrap Up (5 minutes)

Journal

Question of the Day: How can the five CS practices (problem solving, persistence, communication, collaboration, and creativity) help programmers to complete large projects?

Prompt: Have students reflect on their development of the **five practices of CS Discoveries** (Problem Solving, Persistence, Creativity, Collaboration, Communication). Choose one or more of the following prompts as you deem appropriate.

- Choose one of the five practices in which you believe you demonstrated growth in this unit. Write something you did that exemplified this practice.
- Choose one practice you think you can continue to grow in. What's one thing you'd like to do better?
- Choose one practice you thought was especially important for the project we completed today. What made it so important?

Reflect

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.



End of Course Survey

If this is the last unit of CS Discoveries that you are teaching, also have students take the end-of-course survey. See the **CSD Instructions resource** for more information about the End-of-Course survey and how to assign and see the results.

Post-Project Test

The post-project test is found at the bottom of the Interactive Animations and Games unit overview page on Code Studio.

This test is locked and hidden from student view by default. In order for students to see and take this test, you'll need to unlock it by clicking the "Lock Settings" button and following the instructions that appear.