

Unit 4 - The Design Process

The Design Process unit transitions students from thinking about computer science as a tool to solve their own problems towards considering the broader social impacts of computing. Through a series of design challenges, students are asked to consider and understand the needs of others while developing a solution to a problem. The second half of the unit consists of an iterative team project, during which students have the opportunity to identify a need that they care about, prototype solutions both on paper and in App Lab, and test their solutions with real users to get feedback and drive further iteration.

Chapter 1: User Centered Design

Big Questions

- How do designers identify the needs of their user?
- How can we ensure that a user's needs are met by our designs?
- What processes will best allow us to efficiently create, test, and iterate upon our designs?

Week 1

Lesson 1: Analysis of Design

Unplugged

The class explores a variety of different teapot designs to consider how design choices are made and why. Using the teapots as an example, the class will explore the relationship between users, their needs, and the design of objects they use.

Lesson 2: Understanding Your User

Unplugged

Using user profiles, the class explores how different users might react to a variety of products. Role playing as a different person, each member of the class will get to experience designs through someone else's eyes.

Lesson 3: User-Centered Design Micro Activity

Unplugged

In small groups, the class uses the design process to come up with ideas for smart clothing. From brainstorming, to identifying users, to finally proposing a design, this is the first of several opportunities in this unit to practicing designing a solution for the needs of others.

Week 2

Lesson 4: User Interfaces

Unplugged

See how a paper prototype can be used to test and get feedback on software before writing any code. To help out a developer with their idea, the class tests and provides an app prototype made of paper.

Lesson 5: Feedback and Testing

Unplugged

Users have been testing an app, and they have lots of feedback for the developer. The class needs to sort through all of this feedback, identify the common themes and needs, and start revising the prototype to make it better meet the users' needs.

Lesson 6: Identifying User Needs

Unplugged

Up to this point the users that the class has considered have all been remote, and the only information from users has come through text or role playing. Now the class will rely on each other as potential users, and pairs will get to interview each other to identify needs that could be addressed by developing an app.

Lesson 7: Project - Paper Prototype

Unplugged | Project

Using the interview information from the previous lesson, the class comes up with app ideas to address the needs of their users. To express those ideas, and test out their effectiveness, each student creates and tests paper prototypes of their own.

Chapter Commentary

This chapter introduces the design process as a specific version of the problem solving process in which empathy for a user's needs is consistently integrated. Students learn strategies for identifying user needs and assessing how well different designs address them. In particular they learn how to develop a paper prototype, how to gather and respond to feedback about a prototype, and consider ways different user interfaces do or do not affect the usability of their apps.

Chapter 2: App Prototyping

Big Questions

- How do teams effectively work together to develop software?
- What roles beyond programming are necessary to design and develop software?
- How do designers incorporate feedback into multiple iterations of a product?

Week 3

Lesson 8: Designing Apps for Good

Unplugged

To kick off the app design project, the class organizes into teams and starts exploring app topics. Several example socially impactful apps serve as inspiration for the project.

Lesson 9: Market Research

Dive into app development by exploring existing apps that may serve similar users. Each group identifies a handful of apps that address the same topic they are working on, using those apps to help refine the app idea they will pursue.

Lesson 10: Paper Prototypes

Unplugged

Paper prototypes allow developers to quickly test ideas before investing a lot of time writing code. In this lesson teams explore some example apps created in App Lab, using those apps to help inform the first paper prototypes of their apps.

Week 4

Lesson 11: Prototype Testing

Unplugged

In this lesson teams test out their paper prototypes with other members of the class. With one student role playing the computer, one narrating, and the rest observing, teams will get immediate feedback on their app designs which will inform the next version of their app prototypes.

Lesson 12: Digital Design

App Lab

Having developed, tested, and gathered feedback on a paper prototype, teams now move to App Lab to build the next iteration of their apps. Using the drag-and-drop Design Mode, each team member builds out at least one page of their team's app, responding to feedback that was received in the previous round of testing.

Lesson 13: Linking Screens

App Lab

Building on the screens that the class designed in the previous lesson, teams combine screens into a single app. Simple code can then be added to make button clicks change to the appropriate screen.

Week 5

Lesson 14: Testing the App

Teams run another round of user testing, this time with their interactive prototype. Feedback gathered from this round of testing will inform the final iteration of the app prototypes.

Lesson 15: Improving and Iterating

Using the feedback from the last round of testing, teams implement changes that address the needs of their users. Each team tracks and prioritizes the features they want to add and the bugs they need to fix.

Week 6

Lesson 16: Project - App Presentation

Project

Each team prepares a presentation to "pitch" the app they've developed. This is the time to share struggles, triumphs, and plans for the future.

Chapter Commentary

This chapter is focused on a long running group project that allows students to apply all they've learned about User-Centered Design to develop an app prototype. Working in teams, students identify a social issue that they care about and design and prototype an app to address that issue. This is an opportunity for students to explore other roles in software development, such as product management, marketing, design, and testing.



This curriculum 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**.

Lesson 1: Analysis of Design

Unplugged

Overview

To kick off a unit devoted to group problem solving and developing products for other users, students begin by investigating the design of various teapots. Students analyze each teapot, attempting to identify how specific user needs might have informed its design. By considering these design choices, and attempting to match each teapot with a potential user, students can begin to see how taking a user-centered approach to designing products (both physical and digital) can make those products more useful and usable. To conclude the activity, students are asked to propose some changes to one of the teapots that would make it more useful or usable.

Purpose

Students will enter this unit with an understanding of the problem solving process from prior units.

The problem solving process used throughout CS Discoveries is:

- Define
- Prepare
- Try
- Reflect

In this lesson, students look at real world objects to understand how the problem solving process can be applied to help others. Starting with this lesson, we will be reframing this process to include a layer of empathy, encouraging students to consider how others will experience and use the things they create.

We are purposefully starting out by looking at non-technical objects to encourage students to think more broadly about what it means to consider the end user of a product before honing in on how it specifically applies to software design

Agenda

Warm Up (5 min)

Introduction

Activity (35 min)

Who Was this Designed For?
Discussing Design Choices

Wrap Up (5 min)

Improving on Designs

Extension Activities

Exploring Everyday Things

[View on Code Studio](#)

Objectives

Students will be able to:

- Express opinions respectfully and effectively
- Critically evaluate an object for how well its design meets a given set of needs
- Identify empathy for the user as an important component of the design process

Preparation

Print a copy of **Determine the User - Activity Guide** for each student (or prepare to distribute the document digitally).

Links

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

For the Students

- **Determine the User** - Activity Guide
[Make a Copy](#)
- **Problem Solving Process with Empathy** - Image

Vocabulary

- **Critique** - To critically evaluate in a detailed and constructive manner.
- **Empathy** - In design, paying attention to a user's feelings and needs when designing a product.
- **User** - Someone who uses an object, including software and hardware.

Teaching Guide

Warm Up (5 min)

Introduction

Display: Put up the **Problem Solving Process with Empathy - Image** to introduce students to the User-Centered Design Process as an extension of the Problem Solving Process they used in the first three units. The primary difference to note is that we've added **Empathize** to the background of the entire process.

Prompt: What does it mean to empathize? How might the concept of empathizing in a problem solving process be different from in other places you've heard of empathy?

Discuss: Students should discuss their ideas as a class. Eventually direct conversation towards the fact that to empathize with other people means to consider their wants, needs, or concerns.

Remarks

This unit we are going to start thinking a lot more about designing for other people. This is still largely a problem solving process, but we will now need to think a lot more closely about the needs of our users. In other words, empathy will be an important theme as we learn more about design.

Discussion

Goal: This discussion can be fairly free-flowing and open ended. While you want to eventually arrive at a shared understanding of what it means to empathize, students will be working on their understanding of the word over the entire unit. There's no need to settle on a single fixed definition, so treat this discussion as an introduction to a theme of the unit.

Activity (35 min)

Who Was this Designed For?

Distribute: Determine the User - Activity Guide

Determining the User

The first part of this activity asks students to match different teapots with the most likely user. Have students work through this activity in pairs, encouraging discussion about why exactly they are choosing to make each connection. This is a great place to discuss the design of teapots and the needs of users.

While some of the users have an obvious connection to a specific teapot, arguments could be made for any number of connections. There are no right or wrong answers here - the discussion and ability to reasonably back up a decision are what matter.

Critiquing Design

Remarks

We're now going to learn how to critique a design. A critique is a careful criticism in which you give your opinion about the good and bad parts of something, such as a piece of writing or work of art. It's important that we see critiques as feedback towards improvement, as opposed to a final judgement. In this case we're going to critique objects in order to identify for whom they may have been designed.

This second activity gives students an opportunity to specifically critique four of the teapots. The structure of this page reflects the general approach we will be using for critique, using the three statements:

Teaching Tip

Reducing Printed Materials

Unit 4 contains a lot of printed materials. Classrooms with limited ability to print can choose one of the following options instead.

Option 1: Online Activity Guides

All of our activity guides are available in editable Google Doc and Microsoft Doc formats. Classrooms using Google or Microsoft products can distribute digital copies to students, who can complete most of the activity guides entirely online.

Option 2: Student Journals

Many of the activity guides can be converted into journal activities. Teachers can project the instructions and prompts or allow students to view them online, then complete the activity itself in a journal.

Look for teaching tips on relevant lessons to indicate when you may avoid printing by using one of these options.

- I Like... (a strength)
- I Wish... (a weakness)
- What if... (a suggestion)

You may want to ask students to apply this critique process to more of the teapots, or objects around the room, to help reinforce the process.

💡 Teaching Tip

The first step towards creating a design that can be effectively used by many people is to understand that not everyone thinks the same, or looks at an object or design in the same way. This is the first exercise in the students thinking outside of themselves and at how others may perceive items in the world

Discussing Design Choices

Discuss: Students should talk about their answers on the worksheet either in their small groups or as a class. Questions could include:

- Which Teapots did you choose for "Someone who needs to serve tea at dinner party?" Why did you choose those particular teapots?
- Which teapots did you choose for "Someone who likes metallic objects?" Why did you choose those particular teapots?
- Which users were the easiest to find matches for?
- Which users were the hardest to find matches for?
- For page 2, ask which teapot was your favorite? Why?

Wrap Up (5 min)

Improving on Designs

Journal: Pick one or two of the objects you analyzed today (in either activity):

- What could you change to make this object more usable for you? Feel free to use words, pictures, or a combination of both.

Extension Activities

Exploring Everyday Things

Bring in a variety of odd or uncommon objects (old tools, obscure kitchen utensils, and antique gadgets work well for this activity). Distribute the objects around the room and organize students into groups of 3-4. Allow the groups to circulate around the room and examine the objects. For each object, groups will discuss what they believe the object does and what kind of person might use it.

When the class has had a chance to explore all of the objects, reconvene as a group and discuss:

- What teams thought each object was
- What each object **actually** does
- What features of each object gave clues to what it did, or what kind of person might use it.

Poorly Designed Products

Ask students to find things they use in everyday life that they feel could benefit from a design overhaul. As a class discuss what is ineffective about the existing design and how it could be improved.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **CS** - Computing Systems



This curriculum 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**.

Lesson 2: Understanding Your User

Unplugged

Overview

Designers need to understand their users' needs in order to create useful products. This lesson encourages students to think about how to design for another person by role-playing as someone else using a user profile and reacting as that user to a series of products. Each student is assigned a user profile describing a person, which they then use to choose appropriate products, critique product design, and suggest improvements to design.

Purpose

This lesson builds on the previous by asking students to not only consider that there **is** a user that products are designed for, but also empathize with those potential users. Throughout this unit, students will work to apply the problem solving process from Unit 1 to problems for other people. Empathy is a key element of this kind of user-centered design and a difficult skill for students at this age to master.

Agenda

Warm Up (10 min)

Usability vs. Aesthetics
Who Am I?

Activity (30 min)

Looking Through a User's Eyes
Responding to Products
Find a Seat

Wrap Up (5 min)

Thinking About Empathy

[View on Code Studio](#)

Objectives

Students will be able to:

- Distinguish between their own needs and the needs of their users
- Critique a design through the perspective of a user profile
- Design improvements to a product based on a user profile (not personal opinions)

Preparation

- Print out enough user profiles for all students, ensuring as much variety as possible
- Print **Reacting as your User - Activity Guide** for each user
- Prepare prompt questions for warm up exercise
- Label four corners of the room as A, B, C, and D
- Arrange room in table groups of 4, and place a number on each table group (1, 2, 3,...)

Links

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

For the Teacher

- **Understanding the User** - Slide Deck

For the Students

- **Reacting as your User** - Activity Guide

[Make a Copy](#)

- **Profile 1** - User Profile [Make a Copy](#)
- **Profile 2** - User Profile [Make a Copy](#)
- **Profile 3** - User Profile [Make a Copy](#)
- **Profile 4** - User Profile [Make a Copy](#)

Vocabulary

- **Usability** - How easy, efficient, and satisfying it is to use a human-made object or device (including software).

Teaching Guide

Warm Up (10 min)

Usability vs. Aesthetics

Prompt: Are good looking products always easy to use? Can you think of any examples of a product that looked good but was hard to use? What about something that was easy to use, but you didn't like how it looked. Write them down and be prepared to share.

- 🗣️ **Discuss:** Have students journal their thoughts. As they share with their neighbors what they wrote ask them to also discuss whether they care about a product being usable or looking good. After a couple of minutes allow a couple of partners to bubble up what they discussed.

🗣️ **Remarks**

Clearly as users of products we have a lot of needs. It's important as a designer to be able to distinguish among these needs. The **usability** of our designs will affect whether a user can use the product in the first place. If a product isn't attractive, however, it may never be used at all. As we think more about designing for other people, we'll want to have many different kinds of needs in mind.

💬 Discussion

Goal: Highlight that there are many kinds of user needs that we need to empathize with as designers. In particular call out that usability of a product is an important and separate consideration from whether it is merely aesthetically pleasing.

Who Am I?

- 📍 **Distribute:** Hand students user profiles as they enter class. Instruct students to begin reading over their assigned profile.

🗣️ **Remarks**

Take a few minutes to read over your assigned user profile - you'll need to "get in the head" of your user for our activity today. In fact, for the rest of the day you will be learning to empathize with you users, and respond to situations as your users might.

💡 Teaching Tip

Reducing Printed Materials

The User Profiles are used for reference only. Students can look at digital versions during this activity. Assign each student a link, rather than handing out the actual profiles.

If you choose to print out the user profiles, they can be reused, as long as the students are told not to write on them.

Circulate: Give students a few minutes to read over their profiles, encouraging them to "get in the head" of their assigned user. For the rest of the day students will be attempting to empathize with their users and respond to situations as their users should

Display: Understanding the User - Slide Deck slides 5-7

For each slide, ask students to choose a corner based on which picture **their user** would be drawn to. Then have students walk to the corner (A, B, C, or D) that they have chosen to discuss with other students.

Share: Have the groups in each corner briefly discuss (1-2 minutes) why they think their user is drawn to this version of the product. Have one or two students or groups share what they discussed.

Activity (30 min)

Looking Through a User's Eyes

Group: Place students in groups of 3-5 based on their assigned user profile - students with the same profile will be working together through the next activity.

- 📍 **Distribute: Reacting as your User - Activity Guide**

Display: Understanding the User - Slide Deck slides 9-11

Responding to Products

For each product in the slide deck, the groups are going to analyze how their user would react to it by filling out a row in the **Reacting as your User - Activity Guide**. The guide asks students to make a distinction between the **usability** of an item for their user and whether a user likes or dislikes it. You may want to go through the first one as a group so you can model that a product may be aesthetically appealing, but not very usable, or usable but not aesthetically appealing.

Share: Have a few groups share out their reactions to each product.

Find a Seat

Display: Understanding the User - Slide Deck slide 12

Students can work individually on the second page, which asks them to consider which of the chairs displayed on slide 12 best fits their user. After reflecting on the features that lead them to choose a specific chair students have a space to design an even more appropriate chair for their user. This is a creative activity, and students can choose to approach it in whatever way is most expressive for them, including drawing their design.

Share: Circle around the room to share some of the chair designs, prompting students to define what specifically makes their design a better choice for their user.

💡 Teaching Tip

Reducing Printed Materials

Online Option: The Activity Guide can be completed online. Students can "circle" their chosen ratings by putting a border around them, or by typing the rating in the box. For the chair design, students can either draw their design online or submit a paper version of their design separately.

Journal Option: This activity can be completed as a journal entry. Students can use a digital version of the Activity Guide as a prompt, copying the charts and questions into their journals.

Tip: Multiple windows or tabs open on the same computer can be confusing. Have groups work together with one computer displaying the User Profile while another displays the Activity Guide.

💡 Teaching Tip

Adapting the Activity: The provided slides include several products already, but you should add some additional products and images that you think will resonate with your students.

Wrap Up (5 min)

Thinking About Empathy

Journal: Paste or copy your designed object into your journal. Take 1-2 minutes to write about what was easy and what was difficult for you to empathize with your user about.

Exit Ticket:

- What were 3 things about your user that were different from you personally?
- What were 2 times that you found it hard to empathize with your user?
- What was 1 thing you think your user would really like about the chair you designed?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **CS** - Computing Systems



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

Lesson 3: User-Centered Design Micro Activity

Unplugged

Overview

This lesson guides students through an abbreviated version of the design process they will be seeing throughout this unit. Students first brainstorm a list of potential users of smart clothing. As a class, they then group these ideas into broad categories and each group will choose one category of user. Groups repeat this process to brainstorm needs or concerns of their user, eventually categorizing these needs and choosing one to focus on. Finally, students design a piece of smart clothing, using the specific needs and concerns they brainstormed to guide their decision making. At the end of the class students quickly share their decision-making process and get feedback on how well their product addresses the user need they selected.

Purpose

This micro activity is the first of three design projects in this unit. It is a fast-paced introduction to the user-centered design process, intended to give students an experience with user-centered design that they can build on in later projects. Certain shortcuts like speculating as to their user's needs rather than confirming them directly will be corrected as they complete more developed versions of this design process. In this lesson the primary goal is to establish that design decisions will be made with the user's needs in mind. The practice of categorizing lots of disparate ideas to help make decisions will also reappear later in the unit.

The activity in this lesson is an adaptation of the **Design Charrette from the University of Washington**.

Agenda

Warm Up (5 min)

Designing for Others

Activity (50 min)

Introduce the Activity

Define

Prepare

Try

Wrap Up (20 min)

Reflect

[View on Code Studio](#)

Objectives

Students will be able to:

- Empathize with a user's needs to design an object
- Create meaningful categories from a collection of ideas, specifically in the context of a brainstorm

Preparation

- Ensure you have plenty of sticky notes, pens and large poster paper for students to work on
- Set up groups with preferably 3 students each

Links

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

For the Students

- [User Centered Design - Activity Guide](#)

[Make a Copy](#)

Teaching Guide

Warm Up (5 min)

🗣️ Designing for Others

Prompt: In the last two lessons we've seen that products are designed with a purpose and that different designs are more useful or pleasing to different people. Since different people have different needs, interests, etc. what might be some of the challenges if you're trying to design a product for someone else?

Discuss: Allow students a minute to think silently before having them share with their tables and then the class as a whole.

🗣️ Remarks

Designing for other people can be challenging for a lot of reasons, but one of the most important is that it challenges us to consider what another person values, likes, or is concerned about. In other words we need to have empathy for someone else. Today we're going to do a mini design activity to help us practice the entire process ourselves.

💬 Discussion

Goal: This should be a very quick introduction to the lesson. You are looking to call out that designing for other people requires you to consider their needs instead of your own, which can often be challenging. You may wish to point back to the Design Process graphic and re-emphasize the importance of empathy when designing for others. In either case use this as a quick hook for the lesson and then move to the main activity.

Activity (50 min)

Introduce the Activity

Group: Students should be in teams of 3 to 5.

💡 **Distribute:** **User Centered Design - Activity Guide** to each student as well as markers, sticky notes, and poster paper for each group.

Overview: As a class read through the "Overview" section to make sure groups understand the goal of the activity.

Define

💡 **Brainstorm Users:** Ask students to list on their activity guides as many different potential users of smart clothing as they can think of. Give students a couple of minutes to brainstorm independently.

Give students a minute to brainstorm as many different people as they can. Once they're done ask them to create a post-it for the two or three user types they think are most interesting.

Categorize Users: Invite students to discuss with their table at least one bigger category of users they see on the board. Eventually bubble up their ideas to a full class discussion. You should aim to create broad categories for every user on the board.

Choose Specific User: Ask groups to pick one of the categories you've created to design for. Do your best to ensure a good mix of users in the classroom but it's not a problem if some groups choose the same user.

💡 Teaching Tip

Reducing Printed Materials

Online Option: The Activity Guide can be completed online. For the drawing, students can either draw their product online or submit a paper version of their product separately.

Journal Option: This activity can be completed as a journal entry. Students can use a digital version of the Activity Guide as a prompt, copying the headers or prompts into their journals.

💡 Teaching Tip

Why Smart Clothes?: This activity can easily be run with a different target product. This type of product was chosen because these are broad applications for combining computing technology with clothing that could benefit many types of users. An additional benefit is that the field is not yet well-defined and so it provides students more leeway to develop ideas of their own for how to solve people's problems rather than rely on more established solutions. Regardless of what product you use, the point is less that the product is feasible and more that students are thinking creatively about how to meet other people's needs rather than their own.

Brainstorm Needs: Students will repeat the brainstorming process to identify a list of potential concerns, interests, and needs of the user they picked. Encourage students to think carefully about what might be important to those people.

💡 Teaching Tip

💡 **Categorize Needs:** Students should repeat the same process of creating scraps of paper for each need, interest, or concern of their user and then grouping them. Students can use the markers and poster paper to do this step if you have provided those materials.

Choose Specific Need: Ask groups to pick the specific need for their user that they want to address. They should try to pick a need they think could be addressed by smart clothing so in some cases they may need to be a little creative in thinking about these needs.

Brainstorm - Categorize - Choose: In this activity students will use this process twice, first to identify a user, then to identify a user's need. The first time through you should model this process more carefully. During the brainstorm emphasize the fact that there are no right or wrong ideas. When creating categories emphasize that again there are no "right" categories. Remind students that the goal here is to Define the problem they will try to solve today and this process is a useful way to focus in on a specific problem.

Prepare

Brainstorm Solutions: Ask students to brainstorm potential ways smart clothing could be used to address the problem they've decided to solve.

💡 **Discuss Pros and Cons:** Once students have brainstormed solutions invite groups to discuss pros and cons of the proposed solutions. Reinforce that they should be having this conversation from the standpoint of their user. Either the specific needs they chose or the broader needs they've brainstormed should guide how they value each idea.

💡 Teaching Tip

Knowing User Needs: The best way to empathize with someone else is to talk to them. In this activity students are brainstorming potential needs of a user, but in subsequent projects they will want to hear directly from users so they don't make assumptions about their needs. For this lesson the goal is to develop a plausibly realistic set of needs and then think through how to design an object for those needs, rather than your own.

💡 Teaching Tip

Empathizing with User's Needs: Reinforce the need to empathize with the user categories students chose. When weighing pros and cons here it should be from the standpoint of the user needs and concerns they identified.

Try

Describe Your Product: Students should write a description of what their product is and how it addresses their user's need on their activity guide.

Draw Your Product: Students should draw and label a picture of their product. Specifically any "smart" features should be labeled with short descriptions. Students can also use poster paper and markers for this portion of the lesson.

Wrap Up (20 min)

Reflect

Present Your Product: Groups should be given a couple of minutes to share what they created. You can structure presentations around the following steps

- Who your user is and what specific need you identified.
- The features of the product designed
- How the features addressed the need they chose
- At least one feature of their product they might not personally have included but have to meet the needs of their user.

Journal: Based on today's activity what challenges do you foresee in designing software for others?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ CS - Computing Systems



This curriculum 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**.

Lesson 4: User Interfaces

Unplugged

Overview

Following the mini design project, students look towards the next phase of design - prototyping a product that attempts to address user needs. In teams, students examine a paper prototype for a chat app called "Txt Ur Grndkdz". Through using this paper prototype, students get a chance to see how a simple paper prototype can be used to quickly test ideas and assumptions before we ever get to the computer. After "using" the provided prototype students begin to identify ways to improve the next iteration.

Purpose

This lesson introduces students to the concept of a paper prototype as a quick and dirty tool to kick off the development of an app. Paper prototypes will return in both the next few lessons and the chapter two project as well. This prototype will also serve as the context around which students will start to parse and organize feedback from a wide variety of users. While this lesson asks students to work with an existing prototype, they will soon be developing prototypes of their own design.

Agenda

Warm Up (10 min)

What is an App?

Activity (40 min)

**Reading and Using a Paper Prototype
Prototype Debrief**

Wrap Up (5 min)

Critique the Prototype

View on Code Studio

Objectives

Students will be able to:

- Use a paper prototype to test out an app before programming it
- Identify the user needs a prototype was designed to address

Preparation

- Print one copy of the **User Interface Screens - Activity Guide** for each pair of students
- Print one copy of the **User Testing (Computer) - Activity Guide** for each pair of students
- Print one copy of the **User Testing (User) - Activity Guide** for each pair of students
- Cut out the UI Screens or provide scissors for students to do so themselves

Links

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

For the Students

- **User Interface Screens - Activity Guide**
[Make a Copy](#)
- **User Testing (Computer) - Activity Guide**
[Make a Copy](#)
- **User Testing (User) - Activity Guide**
[Make a Copy](#)

Vocabulary

- **Prototype** - A first or early model of a product that allows you to test assumptions before developing a final version.
- **User Interface** - The visual elements of a program through which a user controls or communicates with the application. Often

abbreviated UI.

Teaching Guide

Warm Up (10 min)

What is an App?

Prompt: What is an app? What apps do you or other people you know use?

- 🗣️ **Discuss:** Keep a running list of responses on the board. Once everyone has had an opportunity to share write down a proposed shared definition of an app as "a piece of software designed for a purpose".

🗣️ **Remarks**

When you think of an app, you probably imagine the finished product. In the last lesson you got a chance to experience the earliest phase of development, when we were brainstorming ideas and coming up with potential users and needs. The path between that brainstorming and a finished app, however, is quite long.

- 🗣️ **Discuss:** What do you think goes into the development of an app?

🗣️ **Discussion**

Goal: This discussion aims to introduce the definition of an app. Most suggestions students provide should be able to be tied to this definition. Apps can provide services, organize or provide information, or might just be a game or other form of entertainment. In all cases, you just want to establish an app as a piece of software built for a purpose.

🗣️ **Discussion**

Goal: The purpose of this discussion is to get students to consider the development lifecycle of a software product. You can use the list from the previous prompt to point out how their expectations of apps were likely all focused on the finished product, and not the process of development.

Activity (40 min)

Reading and Using a Paper Prototype

Group: Place students into pairs.

- 📄 **Distribute:** Hand out a copy of **User Interface Screens - Activity Guide** to each pair of students. If you haven't already cut the screens apart, give students a few minutes to do so.

Define: The set screens I just handed you is called a paper prototype. This is one of the earliest forms of a prototype, and it allows a developer to test out their idea before investing a lot of time programming. Paper prototypes are a quick and dirty way to share the user interface of your app with potential users.

- 🗣️ **Discuss:** Based solely on the screens, what do you think this app was designed to do?

Transition: For each pair of students, either assign each student a role of **Computer** and **User**, or allow the students to decide. Once the roles have been assigned, have pairs move so that they are sitting across from each other.

Distribute: Give each of the **Computers** a copy of **User Testing (Computer) - Activity Guide** and each of the **Users** a copy of **User Testing (User) - Activity Guide**

📄 **User Testing**

This activity has pairs of students testing the paper prototype, with one acting as the **User** and one as the **Computer**. The basic process is as follows:

💡 **Teaching Tip**

Reducing Printed Materials

These manipulatives can be reused if students do not write on them. You will need them for the next lesson's activity, too.

🗣️ **Discussion**

Goal: Encourage students to identify specific elements of the prototype that support their arguments, and to consider the user needs that this prototype might have been designed to address

1. The **Computer** places the home screen (the one titled "Txt Ur Grndkds") in front of the **User**.
2. The **User** selects a task from from the table on their activity guide.
3. The **User** attempts to complete the tasking by "clicking" on the paper prototype screen in front of them.
4. Every time the **User** clicks on the screen, the **Computer** consults the Navigation Diagram on their activity guide.
 - If the Navigation Diagram shows a line connected to the element that was clicked, find the screen at the other end of the line and place it in front of the user.
 - If the Navigation Diagram doesn't show a line connected to the element that was clicked, do nothing.
5. When the **User** completes a task (or decides it is impossible in the current prototype), they fill out the "What I Tried" and "My Reaction" columns

There are several additional rows in the tasks table for **Users** to come up with their own tasks to attempt. The first task is filled out as an example, and you may choose to model it for the class.

Prototype Debrief

Share: Have the **Users** share back their experience to the whole class.

- 🗨️ **Discuss:** Discuss some of the common problems that **Users** ran into in this activity, and attempt to identify potential improvements to the prototype.

Wrap Up (5 min)

Critique the Prototype

- 💡 **Journal:** Earlier in the lesson we hypothesized about the user needs that this prototype was address. Now that you experienced the app first hand, as either a **User** or **Computer**, return to that question. In your journal, jot down the following:

- What user needs did this app addresses well (I Like)
- What user needs could this app could do a better job of addressing (I Wish)
- What are a few user needs that aren't addressed at all, but that you thing would make a good addition (What If)

Share: If there's time, have students share out their needs. Consider writing this up on the board or on poster paper where they can be referred to in the next lesson.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

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

💡 Teaching Tip

Reducing Printed Materials

Computer Activity Guide The computer's Activity Guides are used for reference only. Students can look at digital versions during this activity. Assign each student a link, rather than handing out a paper version. If you choose to print out the guide, they can be reused, as long as the students are told not to write on them.

User Activity Guide: The user's Activity Guide can be completed online or as a journal activity. Students can use a digital version of the Activity Guide as a prompt, copying the charts and questions into their journals.

🗨️ Discussion

Goal: The next lesson will delve more deeply into analyzing feedback on this app and proposing improvements, so don't invest too much time in the details of specific improvements here. The goal of this discussion is to get students' gears turning in preparation for the following lesson.

💡 Teaching Tip

Reinforce Vocabulary: Use this concluding conversation to reinforce the vocabulary terms introduced in this lesson, in particular "user interface" and "prototype"



This curriculum 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**.

Lesson 5: Feedback and Testing

Unplugged

Overview

In this lesson students use feedback from "users" of the paper-prototyped app from the previous lesson in order to develop improvements to the user interface of that paper prototype. The lesson begins with a reflection on the fact that designers need to translate human needs with technology into changes to the user interface or experience. Students are then given a collection of feedback and requests from users of the app from the previous lesson. In groups students categorize the feedback and identify ways the needs expressed in the feedback could be met by changes to the interface of the app. Then in groups students will implement some of these changes to meet one of the needs they identified.

Purpose

This lesson introduces several skills that students will need to use later in the unit. They will learn to categorize feedback in order to organize and prioritize their response. They learn to interpret human needs with a piece of technology in terms of the changes to the user interface. They have their first opportunity to create a paper prototype of their own by designing a new screen of an app. As students move towards designing apps of their own from scratch these will be important skills that students use repeatedly.

Agenda

Warm Up (10 mins)

Activity (40 mins)

**Making Sense of User Feedback
Responding to Feedback**

Wrap Up

[View on Code Studio](#)

Objectives

Students will be able to:

- Translate user needs into changes and improvements in the user interface of an app
- Categorize and prioritize user feedback for an app
- Create a paper prototype for the screens of an app

Preparation

Print one copy of **Prototype Feedback - Activity Guide** for each group of 2 or 3

Print one copy of **Improve a Screen** for each student

Colored pencils or markers for the paper prototypes

Links

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

For the Students

- **Prototype Feedback** - Activity Guide

[Make a Copy](#)

- **Improve a Screen** - Activity Guide

[Make a Copy](#)

- **User Interface Screens** - Activity Guide

[Make a Copy](#)

Teaching Guide

Warm Up (10 mins)

Prompt: Think about a piece of software (like an app) that was hard to use. Maybe you couldn't find something you needed, it was poorly organized, or just difficult to use. What specifically didn't you like about it? Why didn't it meet your needs? How could it have been improved?

- 🗣️ **Discuss:** Students should silently write down their ideas. Then move to a full class discussion of the types of issues students have had with apps in the past. Direct conversation towards what specifically about the user interface or user experience of the app was lacking, and follow up by asking how those aspects of the app could be improved.

🗣️ *Remarks*

Technology is made to meet our human needs and that's how we usually think about them. I heard lots of good examples of things you needed the technology to do better. As makers of an app we want to meet our user's needs but we need to think about them in terms of what we have control over in the app. In other words, we need to start thinking about how to translate human needs into the user interfaces and experiences we create.

🗣️ Discussion

Goal: This conversation is aimed at making the point that human needs with software can be solved by changing the software's user experience or user interface, terms students learned in the previous class. This is primarily a brainstorm, and if those points don't come out naturally in conversation make them as you transition to the main activity using the proposed comments below.

Activity (40 mins)

Making Sense of User Feedback

Group: Place students in groups of 2 or 3

- 💡 **Distribute:** Give each group of students a copy of **Prototype Feedback - Activity Guide**. You should also either distribute copies of the **User Interface Screens - Activity Guide** or display on them on the screen.

📝 *Prototype Feedback*

- 💡 **Categorize Feedback:** Working in groups students should create piles of slips of paper so that each pile is a category of feedback. Categories should correspond with similar needs or problems that the feedback is addressing. For example, several pieces of feedback might all be about the fact that the font is too small to read.

Share: Once all groups have categorized their feedback ask for a few suggestions of the types of categories they created.

💡 Teaching Tip

Reducing Printed Materials

The Prototype Feedback and User Interface screens can be reused, as long as students don't write on them. The User Interface Screens should already be printed from the previous lesson.

💡 Teaching Tip

Picking Categories: Circulate the room asking groups to explain how they are forming their categories. Remind students that their user may not always say exactly what they need. You need to interpret or translate what they've said. There's no correct grouping of feedback here, but students should be prepared to justify their categories

Responding to Feedback

Distribute: A copy of **Improve a Screen - Activity Guide** to each group.

📝 *Improve a Screen*

Potential Improvements

- 💡 As a group, students take the feedback that they categorized and propose some potential improvements to the app. Each proposed improvement should be clearly connected to one of the feedback categories that were created in the last activity.

Screen Redesign

Each student will need to draw an improved version of one of the screens in the paper prototyped app. The activity guide contains one page for each screen of the app for the group to divvy up.

Wrap Up

Share: Ask students to briefly present the screen that they have updated. Ask them to describe specifically

- The category of needs they chose to address
- Different ways they considered to address those needs
- The changes to the user interface and user experience they designed to address those needs

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **IC** - Impacts of Computing

💡 Teaching Tip

Reducing Printed Materials

The Activity Guide can be completed online or as a journal activity. Students who complete the activity online may choose to use an online drawing tool to sketch the improved user interface or submit the sketch separately on paper.

💡 Teaching Tip

Choosing Improvements: Students may need help brainstorming good improvements to their apps. For example, if multiple users are complaining about small text then they could try to increase the size of the the font on the screen they are improving. They might also choose to add a "text size" setting in their "Settings" page. There are always many improvements to an app that might have the desired effect. For this lesson it is more important that the change reasonably address the need they chose. There will be more opportunities to investigate what is realistic to change in an app's UI later in the unit.



This curriculum 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**.

Lesson 6: Identifying User Needs

Unplugged

Overview

Up to this point students have focused on designing for users who are, to some degree, distanced from them. Whether through brainstorming, profiles, or text feedback, the connection to an end user has never been direct. This distance is designed to help students get outside their own head when thinking about users, but in order to get information more directly from an actual user, students need to rely on their classmates. In this lesson students pair up to become users (and designers) for each other, allowing everyone to directly interview their end user and ask questions to better inform their design. Each student pair interviews each other, attempting to identify a specific need that could be addressed by an app.

Purpose

Up to this point students have explored users and empathy at a distance - whether it's analyzing a user profile, brainstorming potential users and needs, or responding to written feedback from users. In this lesson we transition to a more direct interaction with users, through interviews.

Agenda

Warm Up (5 min)

Personal Goal

Activity (40 min)

Listening to Users

Brainstorming Solutions

Wrap Up

Project Preparation

[View on Code Studio](#)

Objectives

Students will be able to:

- Interview a peer to learn about their needs
- Analyze interview notes to develop follow-up questions
- Brainstorm potential solutions to a specific problem

Preparation

Print one copy of **User Interview - Activity Guide** for each student

Print one copy of **Paper Prototype - Project Guide** for each student

Gather brainstorming materials (pens, sticky notes, etc)

Links

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

For the Students

- **User Interview** - Activity Guide

[Make a Copy](#)

- **Paper Prototype** - Project Guide

[Make a Copy](#)

Teaching Guide

Warm Up (5 min)

💡 Personal Goal

Journal: What is something you've always wanted to learn (or learn to do better)? This will be shared with a peer, so make sure this is something you are comfortable shared. What has gotten in the way of you learning this?

💡 Teaching Tip

Push students to think beyond learning information or academic subjects, but instead to focus on more aspirational skills. Playing an instrument, doing a craft, or making something are all good fodder for the following activities.

Activity (40 min)

Listening to Users

Group: Pair students as heterogeneously as possible. The goal in pairing is for students to attempt to solve a problem that is not their own, so to the extent possible you want students to be paired with someone of different interests and background.

💡 Teaching Tip

Reducing Printed Materials

This Activity Guide can be completed online or as a journal activity.

💡 **Distribute:** Give each student one copy of the **User Interview - Activity Guide**.

🗣️ Remarks

Until this point the information that you have received about users has always been at best second hand and one directional (from the user to you). Often this is the only type of information we can get about users, but whenever possible it's best to go directly to the source. You each are now paired with your new user, and your job is to design an app for this user. Your apps will be designed to remove a barrier to learning something new - the same something new that you each identified in the warm up activity.

📝 User Interview

This interview activity is broken into two parts - two initial questions that appear on the first page, and three follow-up questions that students will create on the second page.

Initial Questions

These first questions are focused on learning what specifically each user wants to learn, and what they think is in the way of that. This should give each student some explicit information to work off. Give each student roughly three minutes to complete this portion of the interview with their partner, and push them to come up with follow-up questions on the spot that will keep the user talking for the full three minutes.

Developing Follow-up Questions

After everyone has had a chance to ask the initial questions, give students a few minutes to read back through their notes. Students should seek out and highlight any statements that they would like to know more about and any ambiguous statements that need to be clarified. The goal of this time is to "read between the lines" and try to figure out what your user has not yet stated, or what assumptions that you may have that need to be validated.

Once students have reviewed the responses to the initial questions, they should each write down three follow-up questions.

Give students another 2-3 questions each to ask their follow-up questions.

Identifying Barriers

The final goal of this interview is to identify a handful of barriers to the user learning what they wanted. As students complete this section, encourage them to point out specifically what in their interview notes lead to identify each barrier.

Share: Give each student a chance to share back a few of the barriers that they identified, keeping track of all of them on the board.

Discuss: As a class talk about how we could categorize all of these barriers. On the board, create 4-5 general categories as identified in the discussion (see discussion goal for tips) and sort the barriers from earlier into these categories. It's not essential that **every** barrier fits neatly into one of the categories, but the class should come up with a broad enough set of categories to cover most of their identified barriers.

Brainstorming Solutions

Share: Have each student select one of their identified barriers as the focus of their app. The barrier that they chose should fall under one of the categories created in the previous discussion. Students then share out what their category is (not the specific barrier).

Group: Based on the categories chosen, group students into common categories.

Brainstorm: In groups, brainstorm potential app-based solutions to the barrier category. While each student should keep their specific barrier mind during this activity, the goal is to consider solutions to the broader category of barriers - this will give students the change to see potential solutions that they may not have considered for their specific problems.

Wrap Up

Project Preparation

Distribute: Give each student a copy of **Paper Prototype - Project Guide**

Prompt: Look over this project guide. Tomorrow we will kick off this project based on the interview and brainstorming you completed today. On the first page, complete the **Define** section.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **IC** - Impacts of Computing

Discussion

The goal here is to guide the students towards general types of barriers that will allow them to work together to brainstorm around a broad category of barrier, without worrying about the specifics on their user. Ideally these categories look something like:

- Time management (finding the time to learn something)
- Resource acquisition (getting resources needed to learn something)
- Mentorship (finding human support to learn something)
- Prerequisite skills (building the base skills to learn something)

If your students struggle to identify barriers at this level, consider providing these categories for them and having them look through their interview notes to find examples of each kind of barrier.

Teaching Tip

Reducing Printed Materials

This project guide can be completed online or as a journal activity.



This curriculum 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**.

Lesson 7: Project - Paper Prototype

Unplugged | Project

Overview

Based on the peer interview from the previous lesson, each student comes up with an idea for an app that will address their user's problem. Students then get to create their own paper prototype of their app ideas by drawing "screens" on individual notecards. A project guide directs students through the process including building the app and testing it with their user to see if their assumptions about the user interfaces they created are accurate.

Purpose

This lesson concludes the first chapter of Unit 4 and is an opportunity to showcase the skills student have developed. They will develop a user interface based on the needs of their classmate that they identified in the previous lesson. Additionally they practice getting user feedback by testing their paper prototype. In the next chapter students will continue to use many of these skills but will be programming their prototypes in App Lab.

Agenda

Warm Up (5 mins)

What Makes a Good User Interface

Activity

Getting Started
Paper Prototyping
Prototype Testing

Wrap Up

Share Your Work
Reflect on the Project

[View on Code Studio](#)

Objectives

Students will be able to:

- Design the functionality of an app to address the specific needs of a user
- Identify improvements to an app based on user testing
- Design the user interface of an app

Preparation

- Print one copy of **Paper Prototype - Project Guide** for each student if you have not already
- Print one copy of **Paper Prototype - Rubric** for each student
- 6 notecards for each student
- Drawing supplies (colored pencils, markers, rulers, etc.)
- Poster paper for each student

Links

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

For the Students

- **Paper Prototype** - Rubric
[Make a Copy](#)
- **Paper Prototype** - Project Guide
[Make a Copy](#)

Teaching Guide

Warm Up (5 mins)

What Makes a Good User Interface

Prompt: Today we will start making the interfaces of our paper prototypes. Before we get started reflect on the "Txt Ur Grndkdz" app we improved or other apps you've used in the past. What features make a good user interface? How can we make sure apps are easy to use and address our user's needs?

- 🗨️ **Discuss:** Have students share their ideas. You can record them on the board for reference throughout the class.

Activity

Getting Started

Group: If they are not already, place students with the partner they interviewed in the previous lesson

- 💡 **Distribute:** If you collected them after the last class give each student a copy of the **Paper Prototype - Project Guide** and **Paper Prototype - Rubric**

Paper Prototype Project

Overview: As a class read through the overview of the project reviewing what they will need to produce and how it will be assessed.

Define: The first several steps of this project ask students to record the results of their interview and subsequent class discussion from the previous lesson. If this is not completed then complete it now.

- 💡 **Prepare:** Students should brainstorm the minimum set of features their app would need in order to address their user's needs. This can be completed as a bulleted list on the activity guide.

- 💡 **Decide Your Screens:** Students are provided space on their activity guide to sketch out their ideas for the screens of their app. This is an opportunity to make a quick first draft of their app. This should take about 10 minutes and focus primarily on

- How many screens they'll need
- What information or functionality appears on each screen
- How screens will be connected to one another

Paper Prototyping

Distribute: Give each student up to six note cards for creating their screens as well as drawing supplies.

- 💡 **Create Screens:** Students should create the screens of their app.

Create Navigation Diagram: Students can make their navigation diagram whenever they want but in general it will make sense to wait until their screens are finished.

Discussion

Goal: Before sending students off to design their paper prototypes give them an opportunity to brainstorm attributes of effective designs. For example they may bring up

- Simple interfaces
- Reusing design elements
- Placing common elements in the same place on each screen
- Draw attention to what's important

Teaching Tip

Reducing Printed Materials: Both the Project Guide and the Rubric can be completed online or as a journal activity.

Teaching Tip

Scoping Projects: Encourage students at this point to scope the functionality of their app to the minimum amount necessary to address their user's needs. As they work they'll likely discover new components they need to add so it's best to start with a small set early on.

Teaching Tip

Why Six Screens?: There's nothing magical about the number six. It's a rough estimate of the number of screens most students should be able to work within for their ideas and is a concrete way to help scope students' apps. It also just fits nicely on one piece of paper. Feel free, however, to impose any other limit you like.

Prototype Testing

Test Your App: Students should regroup with the partner they designed their app for. Using the same structure from Lesson 4 they should test the app. Each should act as the "computer" for the app they designed and allow their user to interact with the different screens of the app. Further instructions exist inside the project guide.

On the project guide students have space to record what they observed from testing their app. Students can and should also ask their user if there's anything else they would like to see in the app.

Interpreting User Feedback: Students have a T chart on the project guide to interpret their user feedback and identify potential changes or improvements. On the left side of the T chart, students list all of the interesting observations they made during testing (such as assumptions the user had, ways they tried to interact with the app, or statements they made while using the app). The right side of the chart is an area to interpret what each of those observations might mean. This process will help students identify what changes should drive their next iteration.

Teaching Tip

How Much Time: Let students know early how much time they'll have to work on their projects. It's recommended students have at least an hour to design their screens and they will likely need more. Help students plan by reminding them how much time they have to work on each individual screen.

Wrap Up

Share Your Work

Create a Poster: As an optional extension of this project students can create a poster of their project to present. The poster could include:

- The screens of their app
- Some way of demonstrating the navigation diagram, perhaps this time using the screens themselves
- A description of what the user is trying to learn and their barrier
- A description of how the app works and how it addresses the user's needs

Reflect on the Project

Journal: Think about the entire development process for your paper prototype, from the initial interview to designing the prototype, to testing, to brainstorming further improvements.

- What were the biggest challenges involved in designing a piece of software for a different user?
- What skills from this project do you think will be useful when programming app prototypes later in this unit?

Discuss: Ask students to share their reflections with the class.

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.

Discussion

Goal: Highlight the difficulty in viewing projects from another person's perspective, especially when you are the one doing the creating. This is a skill students will need to continue to use in this unit. Call out skills like interpreting user needs from interviews and live tests, scoping large projects, designing user interfaces, and general organization that students will continue to use in the second chapter of the unit.

Code Studio levels

- Levels
- 2

Student Instructions

[View on Code Studio](#)

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

► AP - Algorithms & Programming



This curriculum 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**.

Lesson 8: Designing Apps for Good

Unplugged

Overview

To kick off the app design project that lasts through the end of the unit, students first explore a number of apps designed for social impact that have been created by other students. The class then reviews the Define, Prepare, Try, and Reflect steps of the Problem Solving process as they develop an idea for an app of their own with social impact. Finally, students will form project teams and lay out a contract for how the team will function throughout the development of their app.

Purpose

While this lesson spends some time reviewing and reinforcing the user-centered design process that students will use throughout their project, the primary goal is to give students a chance to see how other people their age have used technology to address issues they care about. The resources provided are a great starting place, but you can make this lesson pack a bigger punch if you find examples (even those from your own previous classes) that demonstrate local examples of students designing apps for social good.

Agenda

Warm Up (10 min)

Apps Created by Students

Activity (45 min)

What's in an App?

Team Contract

Brainstorming

Wrap Up (5 min)

Planning for Research

[View on Code Studio](#)

Objectives

Students will be able to:

- Identify ways in which apps can effect social change

Preparation

Assign teams of 4-5, ideally based on similar interest in an app topic

Print one copy per team of **App Design**

Kickoff - Activity Guide

Choose 1-3 of the apps from **Example Apps for Good - Teacher Resource** to share with the class

Arrange classroom seating for pre-selected teams of 4-5 students

Put out sticky notes for each team

Links

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

For the Teacher

- **Example Apps for Good** - Teacher Resource [Make a Copy](#)

For the Students

- **App Design Kickoff** - Activity Guide [Make a Copy](#)

Teaching Guide

Warm Up (10 min)

Apps Created by Students

💡 **Group:** Students should be seated in their pre-assigned teams of 4-5 students, but don't tell them that they are in their team assignments yet.

Display: Share with students the example apps you've selected, either from **Example Apps for Good - Teacher Resource** or elsewhere. For each app:

- **Discuss:** What is the social issue that this app was designed to address, and how was it designed to do so.
- **Prompt:** Are there other apps or pieces of technology you're aware of that work do address a social issue?

💡 Teaching Tip

The reason for not revealing the team assignments at the beginning of the lesson is so students can begin to form their own ideas about apps they may want to work individually before bringing their ideas to the team. Alternately, you can have students move to their team tables at the beginning of Activity 2.

Activity (45 min)

What's in an App?

Discuss: Ask students what they think an app is. If the discussion stalls, try asking some of the following prompts:

- What do you think the difference is between an application (like Microsoft Word) and an app like those we looked at earlier?
- What are some of the things you noticed the students did to build their app?
- Which apps did you think were the most creative? The most useful? The most surprising?
- Which apps do you think would be difficult to create? Why?

Review: Show the Problem Solving Process graphic, and review the Define, Prepare, Try, and Reflect steps. Let the students know that the micro and mini projects they worked on previously focused on the Define and Prepare steps. With this project, they will also work through these two steps, but then continue on to the Try and Reflect steps. They will also cycle back through them at least once to refine their app.

Transition: If students are not already seated in their app teams, have them do so now.

💡 **Distribute:** One copy per team of **App Design Kickoff - Activity Guide**

App Design Kickoff

Let students know that they are currently seated with the team that they will be working with throughout this app design project.

Team Contract

💡 In order to support a positive and productive team work environment, each team will start by developing and signing a contract. You can use this contract throughout the course of the project to address and mediate issues among team members. Go through each section of the contract with the whole class.

Communication: The communication section covers

💡 Teaching Tip

Reducing Printed Materials This guide can be completed online or as a journal activity.

💡 Teaching Tip

The rules in the team contract are general enough to work for most classrooms, but to make it more impactful you should consider customizing the contract to include wording from your classroom or school rules.

basic respectful communication guidelines. There are two open spaces for groups to add rules of their own. Ask groups to consider the modes of communication that they'd like to use and what the expectations should be. For example, **we will use email to communicate outside of the classroom and team members will respond within 24 hours**

Decision Making: While other sections are just bullet pointed lists, the decision making section is numbered to reinforce the priority of the decision making steps.

Participation: The participation section is meant to prevent students from spinning their wheels in situations where they are unsure how to proceed. Consider adding rules to this section as a whole class.

Once you've confirmed that all students understand and agree to the terms of the contract, have team members sign their names at the bottom.

Distribute: Pass out sticky notes to each group.

Transition: If space allows, give each team some space to work on the next activity where they can spread out.

Brainstorming

The second page of **App Design Kickoff - Activity Guide** provides space for teams to brainstorm about their apps.

Review: Remind students of the brainstorming activity they did in the User-Centered Design Micro Activity where they spent a few minutes coming up with as many user types as they possibly can. Let them know they will be doing the same exercise again, but this time they will be focused on writing down as many possible users for their app topic as they can.

In particular, remind them:

- One user per sticky note
- No bad answers
- Build on others' suggestions with "Yes, and..."

Finally, remind them that after they brainstorm on sticky notes, they'll have a few minutes to sort and analyze them, but their first goal is to write down as many ideas as possible, no matter how crazy.

Team Name: To warm up and practice brainstorming, give teams one minute to come up with a team name and record it on the activity guide

Topic: Give teams roughly five minutes to choose a general topic for their app. It's important to remind them that they aren't trying to come up with what the app will do or any specifics at this point, but just agreeing on a general topic that it should address.

User Groups: Give students some time to generate ideas for users, and then go through the process of user grouping. Remind students that when grouping users they don't have to make a decision about their target user during this step, just make the groupings. They should arrange the sticky notes on their tables into categories, and if there is time, come up with a short name for the grouping. Tell them they will have 1-2 minutes to come with with users, and an additional 1-2 minutes to group those users.

Once teams have a group of possible users for their topic, they should work together to narrow down their potential target user group to one or two general categories. Tell students that they will have time during the next two lessons to further research and refine their target user group, but they want to have a clear starting point to guide that research.

Share: Ask each team to briefly share their name, topic, and user group.

Wrap Up (5 min)

Planning for Research

Discuss: Given that groups are tackling big real world problems with these apps, it's unlikely that they'll have direct access to potential users to interview. Discuss with the class how they think they can learn more about who their users are, and what their needs might be.



Discussion

In the following lesson we will use market research (researching existing solutions to each group's problem) as a way to learn about how others have addressed user needs already. While that will be a common way that all groups can learn about their users, use this discussion to help students consider ways to learn about their users outside of the lessons provided.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ IC - Impacts of Computing



This curriculum 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**.

Lesson 9: Market Research

Overview

In this lesson students research apps similar to the one they intend on creating to better understand the needs of their users. Students work within their teams to search the Internet for other apps, then evaluate the ones they find interesting. By the end of the lesson, each team will have a clearer idea about the type of app they want to create and further refine who their target users are. Each team will maintain a list of citations for all the apps they examined for use in their final presentation.

Purpose

This lesson is the primary opportunity students have to get user input into the original conception of their app. By looking at apps similar to the ones they have proposed needing they may be inspired to change the structure of their app or may even identify an aspect of their problem they had previously ignored. In later lessons students will test out prototypes of their app to perform further refinements.

Agenda

Warm Up (5 min)

App Brainstorm

Activity (40 min)

Brainstorming App Ideas

Wrap Up (5 min)

Share Findings

[View on Code Studio](#)

Objectives

Students will be able to:

- Locate apps that address a specific user group or need
- Identify the user needs being addressed by an app

Preparation

Arrange classroom seating for teams of 4-5 students

Print one copy per team of **Market Research - Activity Guide**

Links

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

For the Students

- **Market Research** - Activity Guide

[Make a Copy](#)

Teaching Guide

Warm Up (5 min)

App Brainstorm

Group: Have the students sit at their team tables. This will be the default seating arrangement for the rest of the unit.

Prompt: Today we will be trying to learn more about the needs of the users of your app. One way that we can learn about our user's needs is by seeing how other apps are trying to meet them. Before we go look online, discuss with your group the types of apps you might expect to find. Be prepared to share.

Discuss: Give groups a minute to brainstorm the types of apps they expect to find. Ask them to share out their thoughts. Use this opportunity to expand the scope of solutions they look for. Ideally hearing other groups thoughts will provide new ideas.

Review: Remind the class they are still in the Define phase of their project. Their goal today is to find and analyze other apps in their topic area. They should work within their teams to continue thinking about their target user group and to move towards consensus as to the type of app they want to make. They should not be planning out what the individual screens of their app will look like. That will happen during the next phase - Prepare.

Activity (40 min)

💡 **Distribute:** Hand out one copy of **Market Research - Activity Guide** to each team.

Market Research

In this activity teams will be searching for apps in their chosen topic. Every student is expected to find at least one app for the team to analyze.

As the team finds an app, they should work together as a group to determine if they are inspired by it and want to reference it as they create their own. It is important that they record the name and location of the app they research, so they can later create a citation in their final presentation.

💡 In addition to researching other apps, each team should be brainstorming ideas for their own apps. On the back of the Activity Guide there is space for them to write down these ideas as team members propose them. Emphasize that they will have time in the next lesson to make a final choice, plus time in future lessons to design the app. Their goal today is to just brainstorm a few ideas as they are doing their research.

Suggested places for students to search:

- **Code.org Public Gallery**
- Mobile app classes or competitions like:
 - **Apps for Good**
 - **Technovation**
 - **Verizon App Challenge**
- Mobile app stores like Apple App Store or Google Play
- General web search using Google or Bing
 - Try phrases like “app competition” or “apps for social good”
 - Include your state name for local ideas or competitions

💡 Teaching Tip

Reducing Printed Materials: This Activity Guide can be completed online or as a journal activity.

💡 Teaching Tip

Focus on the User: Make sure students stay focused on the research portion of this activity. Some teams will want to begin drawing out screens for their own apps. Encourage them to take notes on what they find inspiring in other apps, but remind them they are not to start their design process yet.

Brainstorming App Ideas

After teams have found and reviewed enough apps, give them roughly 10 minutes to complete the Brainstorming section of this activity. This brainstorming activity can also be done with Post It notes. Students can write down their ideas like they have done in previous grouping exercises, then as a team select the top 3-6 ideas and record them on their activity guide.

Wrap Up (5 min)

Share Findings

Share: Ask each group to prepare to share at least one app they found that inspired an idea for their project.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ IC - Impacts of Computing



This curriculum 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**.

Lesson 10: Paper Prototypes

Unplugged

Overview

Before starting to design apps, we need to help students to better scope their expectations. Because students will eventually be prototyping these apps in App Lab, they will be in better shape if their ideas align with the kinds of apps that are easily prototyped in App Lab. Teams start this scoping by looking through several example apps designed to demonstrate apps that can be created with App Lab. Teams then can chose one (or more) of the apps as a basis for their own. From there, teams have some time to discuss the basic functionality of their app before using 3x5 index cards to develop paper prototypes.

Purpose

This lesson kicks off the actual design phase of this project. In order to simplify the app design process students are presented several broad categories of apps to choose from in developing their own idea. Students last created paper prototypes in Lesson 7, but at that point very few constraints were placed on the prototype user interface. Because these prototypes will eventually be converted into App Lab apps, it's important to better scope the types of apps teams design and the interface elements they use.

Agenda

Warm Up (5-10 min)

Our Apps

Activity 1 (30 min)

Activity 2 (45 min)

Prototyping Design

Discuss: Exemplar App

Preparing for Paper Prototypes

Paper Prototyping

Wrap Up

Collect Prototypes

View on Code Studio

Objectives

Students will be able to:

- Communicate the design and intended use of program
- Demonstrate the user flow through an app's design using a paper prototype

Preparation

Collect 3x5 index cards (or copies of **Phone Paper Prototype - Template**), sticky notes, colored pens and pencils, scissors for each team

Print a copy of **Prototype Brainstorming - Activity Guide** for each team

Links

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

For the Students

- **Prototype Brainstorming** - Activity Guide
- **Phone Paper Prototype** - Template
- **What's for Lunch** - Prototype

Vocabulary

- **Prototype** - A first or early model of a product that allows you to test assumptions before developing a final version.

Teaching Guide

Warm Up (5-10 min)

Our Apps

Group Make sure students are sitting with their teams.

Prompt: With your team, discuss what you know about your app so far.

- Who will use the app?
- What will it look like?
- How will the user interact with it?

You may have decided on some things, but not others.

Make sure you know what your team has already agreed on, and what you still need to discuss.

Discussion

Goal: In a few minutes, students will be looking at examples of other apps. AS they do, they should be keeping their own app in mind, both how they want to implement the ideas that they already have, and how they might use aspects of the example apps in their own.

Activity 1 (30 min)

Transition: Go to Code Studio to explore the different app types. For each, discuss with the group how it might apply to their problem.

Code Studio levels

Lesson Overview 

Student Overview

Example App Types 

Teacher Overview

Student Overview

In this activity students will be introduced to a small number of relatively constrained apps: [View on Code Studio](#)

- Quiz Me
- Decision Maker
- List Manager
- Crowdsourcing

For younger or more inexperienced students you may want to restrict them to **just** these app types. We've specifically chosen these examples because they are relatively easy to prototype in App Lab.

If you allow students to stray from these, be aware that some types of apps are more difficult to create in App Lab and may require more creativity in developing a prototype. For example, App Lab has no mapping functionality, so if students wanted to create a mapping app, they would need to find or create a static image of a map to use in their app.

Demo Apps

3

4

5

6

(click tabs to see student view)

Distribute: Prototype Brainstorming - Activity Guide

UI Elements: The activity guide contains a table of design elements available in App Lab. For each element, teams will discuss what they think it does and how it could be used in their app. Teams don't need to (and likely won't) incorporate all of the elements, but they should consider how each could be used in the context of their app.

Share: After giving teams time to fill out the table, go through the list asking a group to share out how they might use each of the elements. Use this as an opportunity to ensure that the class has a clear understanding of what each element is and does.

Discussion: Now that they've seen more examples of apps, teams should discuss any new ideas that they have for the app, or things that they might change based on what they saw.

Teaching Tip

Reducing Printed Materials: The Activity Guide can be completed online or as a journal activity. The 3x5 notecards can be substituted with paper that has been cut into 4-6 rectangles.

Activity 2 (45 min)

Prototyping Design

Define: Review the term prototype. To set expectations it's important to continually remind students that they are developing prototypes throughout the entirety of this unit, each one adding more functionality on top of the previous.

Remarks

To better flesh out your app idea, each team will create and test a paper prototype. As with the prototypes you created earlier in this unit, this will allow for quick exploration with minimal investment. Designers can quickly iterate and change sketches based on feedback from users.

Discuss: Engage the class in a discussion about why this process might be preferable to just jumping into programming the apps.

Prompt: As you're doing this, you'll also want to create a Navigation Diagram to describe the flow between the screens. This will be helpful for your user testing tomorrow as well as used in your final presentation.

Content Corner

The process students are engaging in is known as participatory design - a design philosophy for computer systems that originated in Scandinavia in the 1980s. The goal is to get the end users of the design involved in the process early and often.

Discussion

Students may come up with any number of valid reasons, but you'll want to make sure to highlight at least the following:

- It's easier and faster to sketch out an idea
- Testing ideas on paper lets you find problems in your assumptions **before** you spend time programming
- Users are more likely to give feedback on something that looks less polished / finished

Discuss: Exemplar App

Display: What's for Lunch - Prototype . Here is an example of the low fidelity prototype for our example app "What's for Lunch?"

Discuss: What do you notice about this prototype?

Preparing for Paper Prototypes

Prototype Brainstorming

Suggested Screens: The activity guide contains a list of suggested screens for the app types that students experimented with in the warm up. Teams should have selected an app type at this point, so they can use this as a starting point to determine which screens they'll need.

Our App's Screens: The final page provides a table for teams to list all of the screens they'll need. Let students know that they will likely need more screens than are listed in the "suggested screens", and it may not include every screen they eventually create, but it should cover all of the features that they currently plan to incorporate.

Paper Prototyping

Distribute: 3x5 index cards (or copies of **Phone Paper Prototype - Template**), sticky notes, colored pens and pencils, scissors for each team.

Prompt: Using this activity guide as a guideline, sketch out each of your apps screens on a separate index card.

Circulate: During this extended work period, teams will need to divide and conquer while still ensuring that they're working towards a common goal. Watch to make sure that teams continue communicating as they develop their screens, and encourage groups to frequently see how their screens are working together.

Wrap Up

Collect Prototypes

Collect: Gather the paper prototypes so they can be used for testing in the next lesson. Access to these prototypes will be essential for the next few lessons, so consider making a dedicated location in the classroom for teams to store them.

Discussion

Encourage connections to the previous discussion. How does this prototype achieve (or fail to achieve) the goals of a low fidelity prototype?

- Are these sufficiently detailed to know what each screen will do?
- Do these look very polished? Or sketched?
- Are they relatively neat? Or at least neat enough that your test user could read what they need to?
- What User interface features are on each screen? (looking for things like buttons, images, text, etc).
- Do you think that a child could click on these items with their small fingers? Do you think an adult's fingers could fit on the important features?
- Is the flow through the screens clear?
- If you were a user testing this app, what comments might you make to the developers?
- If you were the developers of the app, what might you want to know from the test users?

Teaching Tip

We introduced the App Lab elements earlier in the lesson so that students would have an understanding of the tools they'll have available. Students can stray from the provided elements, but when it comes time to develop their digital prototype, they'll need to either switch to one of the supported elements or use an image as a stand-in.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming



This curriculum 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**.

Lesson 11: Prototype Testing

Unplugged

Overview

The primary purpose of developing paper prototypes is that they allow for quick testing and iteration before any code is written. This lesson is focused on giving teams a chance to test their prototypes before moving to App Lab. Teams develop a plan to test with users before running prototype tests with multiple other students in the class (and potentially outside the class). In order to test the prototype with the users, the students will have to assign roles in the testing (the “narrator”, the “computer” and the “observers”) as well as have some questions prepared for the user to answer after the test is complete.

Purpose

The goal of this lesson is to give students a clear format for testing and iteration of their apps. This will be the first of multiple opportunities teams have to test various stages of their prototypes, and each stage will serve a different purpose. At this point the primary purpose is to gut check assumptions about how the app should be laid out and navigated - this is **not** the time for students to be overly concerned about fine details.

Agenda

Warm Up (5 min)

Getting Prepared

Activity 1 - Testing (45 min)

**Preparing for Testing
User Testing**

Wrap Up (5 min)

Reflection

[View on Code Studio](#)

Objectives

Students will be able to:

- Test a prototype with a user, recording the results
- Analysing a user test to identify potential issues or improvements

Preparation

Either have other people lined up to test each team’s paper prototypes, or schedule enough time for teams to test each other’s prototypes

Print a copy of **Paper Prototype User Testing - Activity Guide** for each team

Links

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

For the Students

- **Paper Prototype User Testing** - Activity Guide [Make a Copy](#)
- **What’s For Lunch Testing** - Video

Teaching Guide

Warm Up (5 min)

Getting Prepared

Distribute: Make sure each team has their prototypes in hand.

Prompt: Before considering testing with other users, take a moment as a team to work through their screens.

Activity 1 - Testing (45 min)

Praparing for Testing

Remarks

When you are running an experiment in science class, you do your best to test one hypothesis at a time. For example, if you want to prove a theory that food will spoil more quickly out of the refrigerator than inside, you wouldn't want to test with a warm fridge that is say 50 degrees fahrenheit and and outside temperature of 50 degrees as well. Nor would you want to test with either case having the temperatures swing wildly - that might affect the results of which spoils first! So you would want to control the temperature "variable" of the food both inside and outside of the refrigerator and see what happens.


It's the same thing when you're testing software. Even though people are very variable, you want to eliminate as many "variables" as possible.

One way to do this is to make sure you're asking the same questions each time you test a piece of software. So we're going to work on a list of questions to ask our users when they are done testing our low fidelity prototypes - so we could compare the different users' reactions to your apps and their answers to your same questions.

Display: Show **What's For Lunch Testing - Video** .

Discuss: what did you notice about how this test was run? Specifically dig into to following roles that were played:

- **The "user"** is the person who is testing the app in the form of the low fidelity prototype. The user should pretend to execute the "app" by pressing on the prototype with their fingers in the way that makes most sense. The most important part is that the user should speak out loud what they are thinking as they do their actions and ask lots of questions if there are things they don't understand. They can also offer helpful suggestions in our critiquing form with sentences starting with "I like...", "I wish...", and "I wonder..."
- **The "computer"** is the person who is manipulating the fidelity prototype based on what the user is doing. For instance, if the user presses a button that should make the app go to another screen, the "computer" would take away the mock up of the old screen and replace it with the the mock of the next screen. The "computer" starts the test by presenting the user with the first screen of the app.
- **The "narrator"** is the person who is running the test. This person will introduce the team members, app and it's purpose. This person will also remind the user to talk out loud as they are manipulating the app and will remind the "computer" and the "observers" to keep from trying to steer the user in what they think the right way to use the app is, unless the user asks for help.
- **The "observers"** are the other students in the team. They will watch the interaction and write down in their notes what they see the user do in response to the computer.

 **Distribute:** One copy of **Paper Prototype User Testing - Activity Guide** for each team.

Paper Prototype User Testing

Overview: As a class review the goals for the user test. In particular respond to any questions about the different roles.

Teaching Tip

Reducing Printed Materials: This Activity Guide can be completed online or as a journal activity.

Assign Roles for Testing: Ask groups to assign roles for their testing. If they wish, however, roles can be switched between tests.

💡 **Identify Users:** Groups should either be paired with another group to test out their app.

💡 Teaching Tip

Testing Outside of Class: If you wish, ask groups to run a version of this test with a member of the community, school, or their families who might be a likely user of the app. If students are testing outside of class, it's recommended that they make copies of the prototypes to use so that the "master" copies stay safe and available to the team.

User Testing

Set Up: Decide how groups will pair up for testing and place the arrangement where students can see.

Prompt: Using the **Paper Prototype User Testing - Activity Guide**, test your app with a user.

Circulate: Students will start their tests which should run for about 5-7 minutes each. Encourage students to keep on task, and encourage the observers to write as much as they can. After students are done, have them move back to their original team.

Summarize Findings: Have groups discuss what they observed and record their findings on the first page of the activity guide. In particular ask them what their observations mean in terms of changes they'll need to make for the user interface of their prototype.

Wrap Up (5 min)

Reflection

Journal: Write in your journal the answer to this question:

- Was there a difference between testing with a user that was involved in the development of your low fidelity prototype (what we did yesterday) vs testing with a user who had never seen this app before?
- What were some of the similarities between the two types of users?
- What difference did you see between the two types of users?

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems
- ▶ **DA** - Data & Analysis
- ▶ **IC** - Impacts of Computing



This curriculum 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**.

Lesson 12: Digital Design

App Lab

Overview

Having collaboratively developed a paper prototype for their apps, groups now divide and conquer to begin work on an interactive digital version based on the paper prototype. Using the drag-and-drop design mode of App Lab, students individually work through a progression of skill-building levels to learn how to build digital versions of a paper prototype. From there, each group member builds out at least one page of their app in App Lab, to be later combined into a single app.

Purpose

This lesson introduces students to the design mode interface of App Lab, and to the process of translating a paper prototype to digital. They'll use this as a tool to iterate on their paper prototypes, first by building up an exemplar app screen, and eventually building up to an interactive app prototype by the end of the lesson.

Agenda

Warm Up (10 min)

Divide and Conquer

Activity (40-60 min)

**Design Mode in App Lab
Sharing Designs**

Wrap Up (5 min)

Group Debrief

[View on Code Studio](#)

Objectives

Students will be able to:

- Translate a paper prototype into a digital format
- Select the appropriate input element for a given type of information

Preparation

- Print one copy of **Screen Ownership - Activity Guide** for each group
- Print one copy of **Screen Design - Activity Guide** for each student
- (Optional) Print one copy of **Model Screen Design - Activity Guide** for each group

Links

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

For the Students

- **Screen Ownership** - Activity Guide
[Make a Copy](#)
- **Screen Design** - Activity Guide
[Make a Copy](#)
- **Model Screen Design** - Activity Guide
[Make a Copy](#)

Teaching Guide

Warm Up (10 min)

Divide and Conquer

Remarks

Today we're going to take the low-fidelity prototypes that you've been working on and begin developing the digital versions that will lead to your final prototype. With your groups, evenly divide up the screens among your group members.

- 💡 **Distribute:** Give each team a **Screen Ownership - Activity Guide**

Screen Ownership

- 💡 Have groups look through their paper prototypes to identify all of the unique screens they will need to create. From there, they can divvy up screens evenly among the group members and document it on the chart.

If there are more screens than students, have groups consider the following factors when dividing up content:

- Is every page **actually** unique, or do some represent the same page with slightly changed content?
- Are there similar pages that could be easily created by the same person?
- Are there multiple simple pages that could be grouped together?

Teaching Tip

Reducing Printed Materials: This Activity Guide can be completed online or as a journal activity.

Teaching Tip

Rules for Screen Names: For each screen, groups should assign a short and clear screen name. This name will be used in App Lab to identify the screen, so they should contain no spaces or special characters.

Activity (40-60 min)

💡 Design Mode in App Lab

Display: Put **Model Screen Design - Activity Guide** up where students can see it, or distribute printed copies. This prototype will also be available to them during the level progression.

Remarks

Before you start working on designing your own apps in App Lab, we're all going to work on converting this paper prototype first. Look over the prototype to make sure you know what you are building toward. Notice that most of the elements on the screen are labelled - you'll want to make sure you use the exact same labels as you build out this screen in App Lab.

- 💡 **Distribute: Screen Design - Activity Guide** when students reach the final level of this lesson. Students can use this activity guide to sketch out their design and document the IDs for each design element.

Teaching Tip

Students will be using App Lab to create a simple prototype for their app. Students who want to go further in making their apps functional can learn more about App Lab, watch demo videos, and work through tutorials at <https://code.org/educate/applab>.

Teaching Tip

Reducing Printed Materials: This Activity Guide can be completed as a journal activity. Classrooms that have chosen the online option may also do this activity on a separate sheet of paper, or use an online drawing tool to design the screen.

Code Studio levels

Lesson Overview 

Student Overview

Tour of Design Mode

[Student Overview](#)

Intro to Design Mode

[Student Overview](#)

Build a Screen

[4](#)

[5](#)

[6](#)

[7](#)

(click tabs to see student view)

Design Mode Elements

[Teacher Overview](#)

[Student Overview](#)

Developing Good Programming Habits

[View on Code Studio](#)

Following clear and consistent naming conventions is an important habit to develop, but the benefits are rarely immediately obvious to students. It really pays off in the long term to be a stickler for well named IDs, particularly because it will save your own sanity when it comes time to read your students' programs, but it will also simplify the process of combining each student's screen into a full team app.

Finish a Screen

[9](#)

[10](#)

[11](#)

(click tabs to see student view)

Design a Screen for your App_2018

[Teacher Overview](#)

[Student Overview](#)

Namespacing

[View on Code Studio](#)

Ensuring that all students are both creating unique IDs for all elements and properly namespacing them will go a long way towards avoiding issues in the next lesson, so it pays to be a stickler here. You can provide students with the optional Screen Design Activity Guide to keep track of their element IDs.

Working together, apart

Even more so than the the paper prototyping activity, it is easy for students in this lesson to forget that they need to consider how their choices affect their team. Encourage active discussion among teams as they design screens to ensure that they are all working towards a common goal.

App Project: Screen Design

[13](#)

(click tabs to see student view)

Sharing Designs

Discuss: How did your initial paper designs change when you recreated them in App Lab. What kinds of adjustments did you have to make?

Share: Have a few students share their screens with the class, focusing on showcasing a variety of designs.

Wrap Up (5 min)

Group Debrief

Discuss: Give teams a few minutes to debrief their experience converting their paper designs to digital. In particular, focus on changes that needed to be made in the conversion so that the whole team knows about changes that had to be made, and which may need to be accounted for elsewhere.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems
- ▶ **IC** - Impacts of Computing



This curriculum 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**.

Lesson 13: Linking Screens

App Lab

Overview

In this lesson teams combine the screens that they designed in the previous lesson into a single app, which they can then link together using code. Students learn basic event driven programming by building up the model app that they started in the previous lesson. In addition to the screen that students designed yesterday, they'll learn how to create additional screens and even import screens made by others.

Purpose

This lesson and the next are the only two in this unit that focus on programming concepts, and the extent to which they do so is minimal. The goal of this lesson is not to teach programming in App Lab (there will be time for that in Unit 6), but rather to teach the minimum amount of programming skills to allow students to make their apps interactive.

Agenda

Warm Up (5 min)

Reacting to Events

Activity (45-75 min)

Linking Screens

Combining Project Screens

Wrap Up (5 min)

Team Debrief

Extension Activities

Screen Pair

[View on Code Studio](#)

Objectives

Students will be able to:

- Write programs that respond to user input
- Integrate screens designed by others into an app of their own
- Collaborate with others to develop an interactive prototype

Introduced Code

- `onEvent(id, type, function(event)){ ... }`
- `setScreen(screenId)`
- `console.log`

Teaching Guide

Warm Up (5 min)

Reacting to Events

Prompt: Think back to when you were programming in Game Lab. When you wanted to respond to user interaction, how did you do it?


Remarks

In Game Lab we used conditionals inside the draw loop to check if a user had interacted with the game thirty times every second. This technique works really well in games and animations where there are a lot of other things that you're also telling the computer to do. Most phone apps, however, **aren't** doing things constantly. In fact, a lot of apps do absolutely **nothing** but wait for the user to click on something.

For apps like this it's better to use a programming concepts called an "Event," which lets you define which code should run when different events occur. In App Lab, "event handlers" can be attached to elements that you created in Design Mode where they can watch for various events to occur (such as clicking on a button).

Activity (45-75 min)

Linking Screens

 This lesson picks up with the model app that students started in the previous lesson. They will add and import additional screens, and then write code that links their screens to buttons.

Teaching Tip

During this lesson, and for the rest of this unit, students will need to share project links with each other in order to import their shared work. Consider providing each group with a shared space (such as a message thread or Google Doc) to track their project links in order to ease this process.

Code Studio levels

Lesson Overview

Student Overview

Adding Screens

 2

 3

 4

 5

 6

 7

(click tabs to see student view)

Events

 8

 9

(click tabs to see student view)

Linking Screens

 10

 11

 12

(click tabs to see student view)

App Project: Screen Import

Teacher Overview

Student Overview

Breaking for the Day

Depending on how much time you have in class, this can be a great point to break for the day. This and the next level can be quite time consuming, so if you are running short on time, consider jumping to the Screen Pair activity described in the Extension Activities for this lesson.

Team Time

Though students are working on their individual copies of their team app at this point, encourage teams to be active and vocal through this portion of the lesson. Any issues that students encounter when importing their teammates' screens can only be resolved through teamwork, and will likely be a problem for all members of the team.

[View on Code Studio](#)

App Project: Linking Screens

[Student Overview](#)

Combining Project Screens

Review: In the previous lesson each student developed a screen for their team app. Give teams a few minutes to review the state of their screens, identifying any remaining work to be done before combining them. The final level in this lesson asks students to import and link each of the screens in their team apps.

Wrap Up (5 min)

Team Debrief

Discuss: Give teams a chance to share their prototypes with each other. Because every student is now working on a separate version (or "fork") of the team app, each app will likely be slightly different. Encourage team members to discuss their different solutions to the same problem - what are the benefits or trade offs?

Extension Activities

Screen Pair

Give students a chance to practice rapid design and prototyping by pairing them to quickly design and merge two screens for a new app.

Brainstorm: As a class, brainstorm a list of app ideas. These can be serious or silly, but encourage a good variety.

Grouping: Pair students, preferably with partners from a different team.

Give each pair a few minutes to choose an app from the list and agree on the screen that each will design, then move start a new **App Lab project**. After a limited time devoted to designing their screens, pairs can then swap share codes and attempt to import each others' screens. The goal here is not to create a working app, but to practice rapid prototyping and iteration. Through this practice students can start to better prioritize what should and shouldn't go into a first iteration.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **IC** - Impacts of Computing



This curriculum 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**.

Lesson 14: Testing the App

Overview

By the end of the previous lesson each team should have a minimum viable prototype of their app. The primary purpose of this lesson is to have the team actually test the app with other people, preferably from the target audience the app is intended for, or from different teams in the class while observers from the team will record the results on the worksheets they used in the planning phase. As with testing the paper prototypes, teams will start by planning for the specific scenarios before running and observing tests.

Purpose

By now students should be fairly practiced at testing with users. This time around, however, the role of the computer is actually done **by the computer!** This means that the team running the test needs to focus their energy on getting carefully watching what the user does, and encourage the user to say aloud what they are experiencing. While it may seem at first that testing with an interactive prototype is easier (since no one has to manually change screens), it can actually be significantly more difficult to observe the user's experience.

Agenda

Warm Up (5 min)

Review Testing Process

Activity (45 min)

Preparing for Testing
Running User Tests

Wrap Up (5 min)

Team Debrief

[View on Code Studio](#)

Objectives

Students will be able to:

- Write out a detailed plan for how they will test their low fidelity prototype with other people
- Run a user test on an app and record what users say about their minimum viable product

Preparation

Print one copy of **App User Testing - Activity Guide** for each student

If possible, have outside users available to test the teams' apps prior to this testing day

Links

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

For the Students

- **App User Testing** - Activity Guide

[Make a Copy](#)

Teaching Guide

Warm Up (5 min)

Review Testing Process

🗣️ **Discuss:** Ask the students to recall the process used in the testing of the paper prototype. Good questions to discuss with the class include:

- What were the steps we used we used in testing the low fidelity prototype? (Expected answer: planning, testing, analyzing)
- What were some of the roles our team mates played in the our testing?
- Do you think we need the role of the “computer” in this testing scenario?
- What is a Task and how does it relate to a Test Case?
- What were some of your Tasks that revealed valuable information in testing?
- What were some of the good questions you asked at the end of the testing session?
- What do you think will be different about testing with the app vs testing with the prototype?

Discussion

Mostly this review is to switch perspective from implementing and back to testing. It's also to show them how similar this testing session will be with what the class did with paper prototypes.

Activity (45 min)

Preparing for Testing

📄 **Distribute: App User Testing - Activity Guide**

App User Testing

This activity guide is very similar to the one students used for testing of their paper prototypes. The primary difference is that there is no "Computer" role, as the actual computer will take care of that. This also allows for the group to run more tests at once, so consider having groups divide and conquer in the testing of this iteration.

💡 Teaching Tip

Reducing Printed Materials: This Activity Guide can be completed online or as a journal activity.

Transition: When teams have had sufficient time to prepare their test cases, transition to the testing phase.

Running User Tests

👥 **Set Up:** Decide how groups will pair up for testing and place the arrangement where students can see.

Prompt: Using the **App User Testing - Activity Guide**, test your app with a user.

Circulate: Students will start their tests which should run for about 10 minutes each. Encourage students to keep on task, and encourage the observers to write as much as they can. After students are done, have them move back to their original team.

💡 Teaching Tip

This test should be longer than testing with the low fidelity prototype as you want the user to be as detailed as possible.

If students get done early, have them go back and do a second round with the same person, perhaps to see if that person missed anything or is willing to “say aloud” more information about their thoughts as they use the app.

Wrap Up (5 min)

Team Debrief

Discuss: Teams review the findings from today's testing and plan for testing opportunities outside the classroom.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems
- ▶ **DA** - Data & Analysis
- ▶ **IC** - Impacts of Computing



This curriculum 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**.

Lesson 15: Improving and Iterating

Overview

Teams have at this point developed an app prototype that has gone through multiple iterations and rounds of user testing. With the information and guidance gained from the last round of user testing, each student has the opportunity to plan for and implement improvements to the team app. Depending on the time you have available, and student interest, you can run the cycle of testing and iteration as many times as you see fit.

Purpose

While it is tempting to make this lesson a deeper dive into programming for all students, encourage each student to find the elements of app design that interest them and focus on that for their personal improvements. While improvement **can** mean adding programmatic functionality, it can also mean a more eye-catching design, more informative text copy, better uniformity of iconography, or any number of other non-programming related features.

At this point in the development process, you can choose to spend as much time as you like iterating and testing. There is no further time budgeted for testing after this iteration, but feel free to use the previous two lessons as a guideline for cycling through further iterations.

Agenda

Warm Up (5 min)

Bugs and Features

Activity 1 (30 min)

Interpreting User Feedback

Brainstorming Session

Bug and Feature Analysis

Activity 2 (30 - 90 min)

Fixing Bugs and Adding Features

Wrap Up (15 min)

Journal

Extension Activities

Merging New Features

[View on Code Studio](#)

Objectives

Students will be able to:

- Analyze the user feedback from the previous lesson and determine a list of bugs (flaws) that need to be fixed and features that could be added to the app
- Prioritize the bugs and features according to impact and ease of implementation

Preparation

- Print one copy of **App User Testing Analysis - Activity Guide** for each team
- Set out sticky notes and poster paper for each team

Links

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

For the Students

- **App User Testing Analysis - Activity Guide**

[Make a Copy](#)

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Feature** - An individual functional element of a software item, such as the ability to do something new, work across multiple platforms, or perform more efficiently

Teaching Guide

Warm Up (5 min)

Bugs and Features

Prompt: Based on your user testing, what are some of the bugs you discovered in your apps? Keep track of these on the board.

- 💡 **Discuss:** You likely found plenty of bugs, but what other feedback did you get from users? Help they class identify feedback that implies the need for new functionality. We're going to call these [feature] requests. Today teams will be working to identify the bugs and features discovered through testing, prioritize them, and plan for improvement.

💡 Teaching Tip

Bugs are often easier for students to identify from feedback than features. Encourage students to really consider how users might have **assumed** the app should work, or places where they needed help or direction to successfully navigate.

Activity 1 (30 min)

- 💡 **Distribute:** Poster paper, sticky notes, and a copy of **App User Testing Analysis - Activity Guide** for each team.

📄 *App User Testing Analysis*

💡 Teaching Tip

Reducing Printed Materials: This Activity Guide can be completed online or as a journal activity.

Interpreting User Feedback

Teams start by completing a T chart that will help connect specific user testing observations to the potential bugs or missing features they reveal. This information will be the basis of a brainstorming session to create list of bugs found in the existing app and features the team would like to add to our next iteration of the app in the next activity.

Brainstorming Session

Once teams have organized all of their feedback into the T chart, they can move into the brainstorming phase. Fill out a sticky note for each bug or feature identified.

The top of the sticky note should say **BUG** or **FEATURE**. The middle of the sticky note should be a description of what the bug or features is - including steps to reproduce the issue if necessary. The bottom of the sticky should have a quick estimate of how long (in minutes) it will take to fix this bug or implement this feature. Let students know that they will have a chance to refine this estimate later in the lesson - all that is necessary right now is a ballpark estimate.

💡 Teaching Tip

It's tempting to focus solely on bugs that are the cause of, or can be solved with, code. Remind students that there are many roles and skillsets on a software development teams, and not all bugs and features are dealt with by programmers. Inconsistent color, confusing text, and counterintuitive layouts are all potential bugs that are important to deal with.

This is an opportunity to let students with other skillsets shine and make a strong contribution to their apps, so make the extra effort to help highlight those students.

Bug and Feature Analysis

On a sheet of poster paper, draw the two charts on page two of this activity guide.

For each of the sticky notes generated in the previous phase, discuss whether it is urgent or not, and whether it seems to be easy or difficult to implement. Based on that discuss, place the sticky in the appropriate quadrant.

Discuss: Now that you have your bugs and features categorized, which of the four categories should be the first that you tackle. Which should be the last?

Activity 2 (30 - 90 min)

Fixing Bugs and Adding Features

Tracking Progress

As students move to Code Studio to begin iterating on their programs, they will use the **To Do, Doing, Done** chart at the bottom of their poster to track the state of each bug and feature. While each student will implement only a select set of these in their own app, the group as a whole will be able to explore a wide variety of features through several versions of the same app. Each time a student tackles a new task they

should first move the sticky from **To Do** to **Doing**, and then finally to **Done** when they have finished implementation. You can use this chart for an at-a-glance view of how each team is progressing.

Teaching Tip

This activity can take as much or as little time as you want, as long as each student is allowed the time to implement at least one fix or feature. You may even consider returning to this activity later in the year when you have some free time - distance from a problem can add useful perspective.

Code Studio levels

Lesson Overview

Student Overview

App Project: Bugs and Features

Teacher Overview

Student Overview

Implementing Features and Fixes Individually

Because App Lab is difficult for large groups of students to work on simultaneously, we have students work on implementing features and fixes individually or in pairs. The benefit of this approach is that it allows the team as a whole to explore more potential features than if they were working all on the same app, but it also means that each team member's app will diverge further from the others as more changes are made.

As an optional extension activity you could provide a day of work for students to share their changes with each other and attempt to integrate all of the added features into a single app.

Keeping Track of Changes

The more features students add to their app, the more likely they are to introduce new unforeseen bugs. To ensure that students always keep the most recent working copy of their app, encourage them make a remix before adding any new features. Make sure students leave the remix alone and return to this level to do their programming - this will ensure that when they go to submit their app for your review it is the most recent version.

[View on Code Studio](#)

Wrap Up (15 min)

Journal

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

- Choose one of the five practices in which you believe you demonstrated growth in this lesson. 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 activity we completed today. What made it so important?

Extension Activities

Merging New Features

In this phase of iteration each student has been working on their own copy of the team's app. If you have the time available, consider having teams reconvene to merge their updates into one master app. This can be pretty tricky, particularly when students may have added conflicting code or design elements, but it's an excellent challenge to build problem solving skills and strengthen team dynamics.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems
- ▶ **DA** - Data & Analysis
- ▶ **IC** - Impacts of Computing



This curriculum 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**.

Lesson 16: Project - App Presentation

Project

Overview

At this point teams have researched a topic of personal and social importance, developed and tested both a paper prototype and a digital prototype, and iterated on the initial app to incorporate new features and bug fixes. Now is the time for them to review what they have done and pull together a coherent presentation to demonstrate their process of creation. Using the provided presentation template, teams prepare to present about their process of app development, including the problem they set out to solve, the ways in which they've incorporated feedback from testing, and their plans for the future.

Purpose

This lesson is the culmination of the last several weeks of project work. In preparing to present their work, it's important that students see this as more than just the programming element of their app - the goal of the presentation is to acknowledge the research, design, testing, and iteration that teams have gone through.

Agenda

Warm Up (5 min)

Inspecting the Exemplar

Activity 1 (40 - 50 min)

Presentation Prep

Activity 2 (10 min per team)

Presentations

Wrap Up (5 min)

Extension Activities

App Celebration Night Submit to Competitions

View on Code Studio

Objectives

Students will be able to:

- Present technical information clearly to non-technical users
- Reflect on the development of an ongoing project

Preparation

- Determine how much time each group has to present. Typically this is 8-10 minutes per team, but remember to leave time for transitioning between teams
- Create a copy of the presentation template slide deck for each team
- Print one copy of **App Presentation - Rubric** for each team.

Links

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

For the Students

- **App Presentation Template** - Slide Deck
- **App Presentation** - Exemplar
- **App Presentation** - Rubric

[Make a Copy](#)

Teaching Guide

Warm Up (5 min)

Inspecting the Exemplar

Discuss: Discuss with the class what all they have completed so far. This has been a large project, and students may not fully realize how much they have accomplished so far. As the class comes up with accomplishments, ask them how they might present the fairly technical things they have done to a non-technical audience (such as potential users of their apps).

Display: Walk through the **App Presentation - Exemplar**, explaining that this is the general structure of the presentation each team will be making.

Content Corner

Communication is an often overlooked crucial skill in computer science, both internally (with other people on the team) or externally (to others in the company or even outside of the company). The students are practicing these skills in this presentation, and these skills will be transferable to many of their other classes.

Activity 1 (40 - 50 min)

Presentation Prep

Distribute: Hand out the **App Presentation - Rubric** and have each team make a copy of **App Presentation Template - Slide Deck** to build their presentation in.

Presentation Template

Most of the content of this slide deck should have already been completed in previous parts of this project. The last few slides (particularly Future Work and Reflection) will require more group thought or group work, so students should address those slides first.

Slide 1: Application name and team roster

Slide 2: A brief Introduction of the problem the app is intended to address and how it does so, as well as the social impact of this app

Slide 3: Describe the market research that was done, including specific apps that were used as inspiration

Slide 4: Show images of the paper prototype and describe the state of the app during paper prototyping

Slide 5: Show images of the digital prototype and describe changes introduced in this iteration

Slide 6: Demo the app, attempting to address all major functionality. Teams may choose to focus on the most functional version of their app, or they could demonstrate functionality across multiple app versions.

Slide 7: List out the key feedback found in each round of user testing. Encourage students to focus on the feedback that most directly affected following iterations.

Slide 8: Describe what the next 3-5 changes would be if the team were to pursue an additional iteration. These changes should be driven by user feedback that the team didn't have time to address.

Slide 9: Reflect on the process as a whole. What went well? What didn't? What did you learn and what would you do differently?

Slide 10: Document any resources used during this project, including the apps found during market research, the source of any pictures used, and any other resources or websites used.

The **App Presentation - Rubric** focuses on the specific content of these slides, but encourage teams to really make

Teaching Tip

Reducing Printed Materials: The Rubric can be completed online or as a journal activity.

this presentation their own - this is their opportunity to "pitch" their app effectively.

Circulate: As teams work on developing their presentations, push them to write in a way that is accessible to a non-technical audience. It often helps to "play dumb" while asking them to present slides to you in order to point out where their presentation may be difficult to understand for some audiences.

Share: If time, pair teams up to practice giving their presentations. Remind teams that their goal is to keep their presentation no longer than 10 minutes.

💡 Teaching Tip

if students are "stuck" about what went right or wrong in the project, prompt them to think about the communication between team members, or the lack of time to complete the task. Usually these are the two major things that can be problematic on a project.

Activity 2 (10 min per team)

Presentations

Transition: Remind each team how many minutes they have for their presentation. Demonstrate how you will be keeping time.

- 💡 **Prompt:** Call up each team individually and allow them to present their work. Each team should allow a few seconds (30) to allow for questions from the other students.

💡 Teaching Tip

it is really important that you keep the class on time with the presentations not only so all of the presentations can get done, but also so the students can see the value of presentation preparation.

Wrap Up (5 min)

Journal: Present students with the following journaling prompts:

- Which presentation did you find the most interesting?
- Which presentation did think would be the most effective at solving the problem they set out to solve?
- For the next questions, write down the phrase "Not Yet", "Almost" or "Got it" depending on where you feel on these topics
 - I feel comfortable researching user wants and needs. (Not Yet, Almost, Got it)
 - I feel comfortable with the research phase of this project. (Not Yet, Almost, Got it)
 - I feel comfortable with the design and low fidelity prototyping phase of a project (including the testing and iteration) (Not Yet, Almost, Got it)
 - I feel comfortable programming in app lab (Not Yet, Almost, Got it)
 - I feel comfortable with testing our app with users and making changes based on those tests. (Not Yet, Almost, Got it)
 - I feel comfortable putting together a technical presentation. (Not Yet, Almost, Got it)
 - I feel comfortable giving a technical presentation on my work. (Not Yet, Almost, Got 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.

📄 Code Studio levels

- Levels
- 📄 2

Student Instructions

[View on Code Studio](#)

Extension Activities

App Celebration Night

Invite parents, other members of the school, and anyone who was interviewed outside of class to come in for a celebration night. Teams can set up booths where they can present their apps and talk about the experience.

Submit to Competitions

Look into having students submit their creations into one of the many youth app competitions. National competitions such as the **Verizon App Challenge** and the **Congressional App Challenge** are available to pretty much everyone, but if you look around you may also find smaller local competitions to participate in.

Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems
- ▶ **DA** - Data & Analysis
- ▶ **IC** - Impacts of Computing



This curriculum 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**.