Lesson 20: Updating Your Prototype

Overview

Question of the Day: How can I track my work while we make updates to our app?

Teams make a decision about how they will incorporate updates to their app, then they begin updating their app based on their post-its from yesterday's lesson. They use a "To Do / Doing / Done" chart to organize their work and track their progress. Teams spend the majority of the class working on updating their app, with time at the end to re-combine into a single app if necessary.

Purpose

Students should focus the majority of their time on updating their apps based on user feedback, but they should also use the "To Do / Doing / Done" chart to stay organized and track their work. Moving a task across the board to the "Done" stage is a major accomplishment and, by the end of class, teams can celebrate how many tasks they completed in a single period.

Assessment Opportunities

1. Track progress while making changes to their app

Activity Guide: Monitor students as they move post-its across the "To Do / Doing / Done" chart, ensuring that their current work is reflected on the chart.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (5 minutes)

Getting Prepared

Journal

Activity (35 minutes)

Objectives

Students will be able to:

- Track progress will making changes to their app

Preparation

- Print one copy of the activity guide for each team
- (Optional) Prepare poster paper or a large whiteboard space for each team

Links

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

For the teachers

- **CSD Unit 4 - The Design Process** - Slides

For the students

- **Updating Your Prototype** - Activity Guide

Teaching Guide

Warm Up (5 minutes)

Getting Prepared

Distribute: Make sure each team has their materials available, especially their Bugs and Features activity guide and post-its from yesterday.

Journal

Prompt: Look over all the post-it notes your team created yesterday. Which one do you think is the most important to complete for your app? Why?

Discuss: Have students journal individually first, then have students share in their teams before asking a few students to share with the class.

Discussion Goal

Goal: This prompt previews the decisions students will make today, where they will "claim" different post-its to work on for improving their app. This also allows teams to hear different perspectives on how important certain features are to their program - some students may focus on adding new features, whereas others may focus on making the app more usable in the first place.

Remarks

> Today we will update our apps based on the bugs and features we identified yesterday! We'll start by picking the post-its we want to prioritize first, then start updating our apps with new features and big fixes! Throughout today, we'll keep track of our tasks using a new chart so we can see how many tasks we've accomplished by the end of the class period.

Question of the Day: How can I track my work while we make updates to our app?

Activity (35 minutes)

Distribute: Give each team a copy of the Updating Your Prototype Activity Guide. Optionally, students can copy this guide onto a poster paper or a large whiteboard area to use instead.

Display: Read through the Overview of the Updating Your Prototype activity guide. Have each team member select two post-its to work on throughout today (if they finish these, they can always select more). Those team members should write their names on the post-its and put them in the To-Do column of the chart.

Update Your App: Teams will need to decide how they would like to continue working on their app since every team member has their own copy of their app after they imported each other's screens. There are three suggested options below. You can have teams decide which of these three they would like to consider, or you can choose one for the entire class:

- **Stay Together on One App:** Each team chooses one of their team's apps to focus on and add updates to - for example, the app they used for testing. During class, the entire team works together to implement one feature at a time using Pair Programming - one team member is the driver while the rest of the team helps navigate
  - **Pros:** Everyone is working on the same app, and there is one completed app at the end
  - **Cons:** Won't be able to implement as many features since everyone is working together, and can be less engaging for navigators when there are large teams.
- **Split Off For Some Updates, Then Recombine to One App:** Half of the team works together on one app to implement any bug fixes or feature updates on existing screens, and the other half of the team works on new features and new code that requires whole new screens. Before class ends, both teams re-share their project share URL so they can re-import the screens into their projects. They will also have to find a way to share code with each other - this usually involves converting to text mode and sharing with another student, such as via a shared Google Doc.
  - **Pros:** Easier to implement more features and, after the sharing, there is one completed app at the end
  - **Cons:** Re-importing screens and sharing code can get messy and takes time away from implementing new features
- **Everyone Works on Separate Apps:** Each team member works on their own version of the app, implementing their bug fixes and new features just in their app.
  - **Pros:** Easiest way to implement features and most of the class time is spent updating the app
  - **Cons:** There is no "unified" app with all of the features in one place, and there's less opportunity for collaboration and teamwork

Code Studio: Have students log into Code Studio and begin working on the updates to their apps.

1 **Updating Your Prototype**

Circulate: Monitor students as they update their apps, focusing on their use of the "To Do / Doing / Done" chart. 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. Make a big deal and encourage celebration when students move a post-it to the "Done" column.

Teaching Tip

Additional Time: Depending on your school calendar, your student's enthusiasm and how complex their apps are, how much you've involved the school community, etc - you may decide to extend how much time students work on these updates. This is especially true if students want to re-combine their updates back to a single app - the process for re-importing screens and sharing code requires some extra time in AppLab.

If you decide to dedicate more time, continue to use the activity guides from today (To Do / Doing / Done) and yesterday (Four quadrants of Urgent vs Easy) as a way for students to manage their tasks and for you to monitor progress throughout this process. Students can create new post-it notes for new features or bugs, then add them to this chart to track their progress. This can help determine how much time to give students to work on these updates.

Wrap Up (5 minutes)

Collect: Collect all the materials from each team in a safe storage location. Try to keep the post-its organized from today's activity. If teams used a whiteboard for their post-it notes, consider having them

take a picture of how they organized their post-its.

Journal

Prompt: Remind students of the **five practices of CS Discoveries** (Problem Solving, Persistence, Creativity, Collaboration, Communication). Choose one practice you thought was especially important for the activity we completed today. What made it so important?

Discuss: Have students reflect individually, then give them the option to share as a class. Since this is a more personal and introspective prompt, it may not be appropriate to have a full-group conversation.