Lesson 3: Happy Maps

53 minutes

Overview

This **context-setting** lesson brings together teams with a simple task: get the "flurb" to the fruit. Students will practice writing precise instructions as they work to translate instructions into the symbols provided.

Purpose

The bridge from algorithms to programming can be a short one if students understand the difference between planning out a sequence and encoding that sequence into the appropriate language. This activity will help students gain experience reading and writing in shorthand code.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm Up (5 minutes)
- **Introduction**

Main Activity (40 minutes)
- **Happy Maps Programming**

Wrap Up (8 minutes)

Extended Learning

Objectives

Students will be able to:

- Articulate an algorithm for a daily task
- Decode and run a program created by someone else
- Identify and address bugs or errors in sequenced instructions
- Translate an algorithm into a program

Preparation

- Print one ***Worksheet** and one set of ***Manipulatives** per group of 2-3.
- Make sure each student has a Reflection Journal.

Links

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

For the teachers

- **CSF Course A Lesson Slides** - Slides (**Download**) ▾ Make a Copy
- **Happy Map Cards** - Worksheet Answer Key ▾ Make a Copy

For the students

- **Feeling Faces - Emotion Image** - Resource ▾ Make a Copy
- **Happy Map Cards** - Worksheet ▾ Make a Copy
- **Happy Map Game Pieces** - Manipulatives ▾ Make a Copy
- **Happy Maps** - Unplugged Video (**Download**)

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (5 minutes)

Introduction

Display: Show "Step-by-step directions" slides

This warm-up is meant to get the class thinking about how to break a big problem down into a list of individual steps.

Do This: Before students give you any instructions, ask them to describe the **goal** of the program ("Walk to the chalkboard and draw a smiley face") and the **expected outcome** ("A smiley face will be drawn on the chalkboard"). Remind students that it is good practice to articulate the goal and outcome of the programs they will create before beginning to write their algorithms.

Model: Start by asking the class for step-by-step directions on how to get to the chalkboard. Make sure that you ask students to break apart any large instructions, like "Walk to the chalkboard," into smaller instructions like "Step forward."

When you reach the board, ask for instructions to draw a smiley face and try to get them to keep their instructions equally small.

Say: *Well done! You just gave me a list of steps to finish a task. In computer science, that's called an* ***Algorithm****!* (Follow up with your typical method of introducing vocabulary: word wall, repeat-after-me, etc.)

Teaching Tip

If the class starts shouting simultaneously, explain that you can only hear one instruction at a time. Call on students individually if that helps.

Display: Show "Vocabulary" slide

Say:

- **Algorithm** - A list of steps to finish a task.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Main Activity (40 minutes)

Happy Maps Programming

Unplugged Activity **Video: Happy Maps - Unplugged Activity**

2 **Happy Maps - Student Resources**

In this exercise, the class will get map cards that have a pre-defined start space (flurb) and end space (fruit). Students will need to get the flurbs to the fruit on each card, using the arrows provided.

Model: Select one of the intermediate maps from the Happy Maps Cards worksheet (e.g., #3). Display it for the class and work through the puzzle together.

Have students look at the puzzle, then think-pair-share their solution for how they would get the flurb to the fruit.

Display: Show “Decompose it!” slide

Remind students to break each puzzle down into smaller steps.

Display: Show "Get the Flurb to the banana" slide

Think: This flurb needs to take two steps to get to the fruit. Work with your elbow partner to decide what you think those two steps are.

Pair: Have students discuss with neighbors for about 90 seconds.

Share: Ask a few students to describe their algorithm to the class. Move your finger along the displayed map as the students read their steps. Once you have a solution, ask if anyone else came up with a different idea that also works.

Display: Show "Using Symbols" slide

Now, share with the students that the magic step of changing an algorithm into a program happens when the code is written down using symbols. Do the students see any symbols on the display?

Think: Challenge students to encode the algorithm that they came up with before into symbols, and to write those symbols down in their journals (or on a piece of paper.)

Pair: Once students have written down their symbols, ask them to swap with a partner to see if they can follow each others' instructions.

Share: Ask for volunteers to come draw their arrows on the board. If the original code doesn't work, spend some time debugging as a class. Students should be familiar with the idea of "debugging" from previous lessons, so be sure to use the vocabulary to get them comfortable with it.

Once the code has been successfully written on the board, congratulate the class on writing their first program together!

Students' Turn

Display: Show "Your Turn" slide

Group: If your class is comfortable, place students into small groups of 2-3. Otherwise, you can continue solving these problems as a class and having them think-pair-share to write programs.

Distribute: Pass out one of the images from the ***Worksheet** to each group as needed.
(Optional) If you start noticing that students are ready for more, use the provided ***Manipulatives** and let students choose their own start and finish destinations on the blank map.

Encourage the students to follow these steps:

- Discuss an algorithm to get the flurb to the fruit.
- Encode the algorithm into arrows in their journals.
- Try their code to see if everything works as expected.
- Debug any issues and fix their code until it works correctly.

Share: When the lesson is done, offer to let groups share out the most difficult maps that they solved. If you have time, ask them to share their solutions as well.

Wrap Up (8 minutes)

Flash Chat

Display: Show "Flash Chat" slide

Discuss: When it's time to wind down, ask students if they can tell the difference between an algorithm and a program. Both are a list of steps, but a program (*code*) has been encoded in a way that can be run by a machine (or a kindergartener!)

- *Do you think that someone who speaks another language would be able to run your program?*
- *Why or why not?*

Reflection

Display: Show "Reflection" slide

Prompts:

- What was today's lesson about?
- Draw one of the ***Feeling Faces** that shows how you felt about today's lesson in the corner of your journal page.
- Can you draw your own flurb map?
- What would the code be to solve your map?

Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

Stuffie Maps

- Create a life-size grid on the rug with tape and have student bring stuffed animals to school.
- Now students can program friends to move their actual stuffies as directed in the programs.

Create Your Own

- Have students create their own maps.
- Have other students solve them using programs.


This work is available under a **Creative Commons License (CC BY-NC-SA 4.0)**.
If you are interested in licensing Code.org materials for commercial purposes **contact us**.