

Map What Happens

Idea/Introduction

Algorithms have two requirements. A state diagram should visually show whether or not the app is a proper algorithm.

Algorithm Requirements:

- Must be finite - have a start and finish and one or more ways to get to and from these states (no infinite loop that will prevent a user from reaching the finish)
- Must be definite - each step should be very clear (not ambiguous)

Learning Outcome

By the end of this activity participants should have a more vetted idea of their app and a diagram to show that their app is fully thought out.

Time

10 min

Definitions

Algorithm: A step-by-step set of instructions for solving/performing a task.

States: A snapshot of the app frozen in time. Each app should have multiple states.

Ambiguity: Unclear step with multiple possible interpretations.

Activity

Step 1. Explain ambiguity

After the participants have their app idea, do this mini activity to explain ambiguity. Give them a vague sentence and ask for creative interpretations of that sentence (examples below). Then go over the sentences and interpretations with the group out loud and show how silly some of them can be. This activity will help illustrate what ambiguity is so that you can tie it back to how the app will process information. If it doesn't understand a step in the code, it won't function.

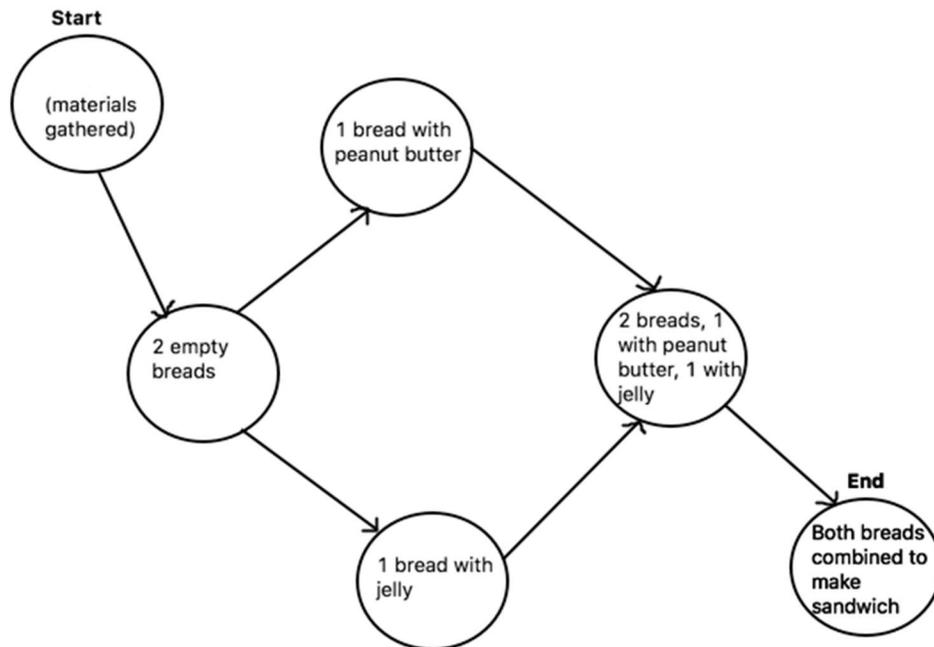
An example sentence: Time flies like arrows

Possible interpretations: Time goes as fast as an arrow, a type of fly (the time fly) is fond of arrows, etc.

Step 2. Explain states

Hand out a state diagramming worksheet and have the participants show a clear start, stop, and other states. Each state should have at least one transition arrow entering or leaving it. There should be no island states that leave someone stranded without a way back or forward.

Example Peanut Butter & Jelly Sandwich State Diagram



Have the participants write down the states for their app and show transitions between each state with arrows.

Framing for slower-paced participants

For slower-paced participants, walk them through the peanut butter and jelly example to help them get started. Ask them ridiculous questions about the order you would make a sandwich; perhaps pantomime trying to spread peanut butter with your hand if they forget to specify that you need a knife.

Framing for faster-paced participants

Challenge faster-paced participants to break down their program steps, and match those steps with App Lab connecting blocks.