Instructor's Handout

(modified from zerorobotics and CS10K Community handouts)

The Peanut Butter & Jelly Computer

OBJECTIVES

In this activity, students will write a set of instructions for the instructor to make a PB&J sandwich. The exercise is an analogy for writing computer software that gives instructions to the hardware.

KEY CONCEPTS

- Software and hardware work together to create a computer, so the software must take the limitations of the hardware into account
- Software is a set of instructions for the hardware, telling it *how* to display and store data.
- Software must provide well-organized, thorough, and *literal* instructions for the hardware.

PREPARATION & MATERIALS

Suggested materials:

jar of peanut butter
jar of jelly
copies of the PB&J student handout
loaf of bread
butter knife

These materials make up the "hardware" components of the PB&J computer. Feel free to modify these components for allergy considerations, or add other materials and tools. Opening the jar of peanut butter by cutting the jar with a pair of scissors or spreading the jelly with the scissor handles can be dramatic, funny, and demonstrate the need for precision and thoroughness in creating software, since hardware doesn't have human "intuition" when carrying out the instructions.

The activity can be completed as a class by following along on the MuddX "What's in a Computer?" page. Students may also complete the student handout individually or in groups.

Instructor's Handout

(modified from zerorobotics and CS10K Community handouts)

Activity:

1. Pass out the PB&J student handouts and set out the "hardware" components in the front of the classroom. Students should complete Part 1, which asks them to list the components in the PB&J computer.
2. Briefly discuss the list. "Instructor" should be on the list of hardware components, because the instructor will be one of the "agents" carrying out the software instructions.
3. Students should move onto Part 2, which asks them to create a set of instructions for the hardware components.
4. Pick a set or multiple sets of instructions to carry out. Follow the instructions in the order they are provided and only perform the actions the instructions explicitly dictate. Computers do what they are told in the order they're told to do it in, and nothing more.

   For example, if the instructions say "put peanut butter on one slice of bread and jelly on the other and put the pieces of bread together," you might set the whole jar of peanut butter on a slice of bread and the whole jar of jelly on the other slice, and then put the pieces of bread side-by-side.

   If the instructions are not specific enough, you can "freeze" or do something ridiculous. A computer doesn't know how to "open the jar of peanut butter" and might get stuck, try to break the jar, or pull the lid off without twisting. However, a computer *could* understand "pick up the jar with your left hand, put your right hand palm down on the lid and squeeze gently. Twist the lid counterclockwise until the lid comes off. Remove the lid and set it on the desk."
5. As you "run the software program" and act out the instructions, invite the students to "provide a software update" and revise their instructions. Repeat for a couple of rounds or until the students are able to provide a set of very specific instructions. Key specifications in the instructions include:
   - The bread must be taken out of the bag
   - The instructor must grab the knife's *handle*.
   - The knife must be inserted into the jar to scoop up peanut butter or jelly (they can define the quantity, too)
   - The knife must be withdrawn from the jar.
   - The instructor should spread peanut butter or jelly evenly across one entire side of the bread.
   - The two slices of bread should be placed so that two two sides with peanut butter and jelly on them are touching.
6. Debrief:
   - Why didn't the first set of instructions work?
     - Not enough specificity, not breaking the instructions into many smaller steps…
   - What kinds of things are humans very good at that computers are bad at?
     - Interpretation, making assumptions…