“How Computers Work”
Instructor notes

Overview:

The purpose of this activity is to give the students a basic sense of how computers work by having them act out a simple computer simulation.

Each student takes on the role of a different part of a simplified computer and they work in groups to run a simple program. The end result of this program is to draw a picture on a simulated computer display.

It is designed for groups of 3 students, although it can be adapted to work as an individual activity or with groups of 2-4.

Prerequisites:

Ability to perform simple addition and plot (x, y) coordinates on a grid.

This activity is designed for students in grades 1-8.

Description:

In this simulation of a (greatly simplified) computer, we consider a computer as being comprised of 3 major components:

- CPU (Central Processing Unit)
  This is the part that executes the program and tells the other components what they need to do.
- ALU/Memory (Arithmetic/Logic Unit & Memory)
  The ALU is the part of the computer that performs all the math and logic operations. The Memory keeps track of information so that it can be recalled later.
- Display
  The Display is the part that shows the results to the person using the computer.

We assign a student to each of these components and give them a simple program to run. The student acting as the CPU processes each instruction in order and tells the ALU/Memory and the Display what to do.

While the program is being run, the Display should hide the image so that the CPU and the ALU/Memory have no idea what it being drawn on the screen.

Once the program is done, the display shows the result to the other members of the group.

Note that these component boundaries used in this activity are slightly different than

what would be found in a real computer. For example, the ALU is actually part of the CPU, and the Memory is generally not bundled with the ALU. These divisions were chosen for this activity to help distribute the work done by each student while keeping the activity mostly accurate.

Setup:

Arrange the students in groups of 3 and assign one student to each of the computer components. If you can't assign each student into a group a three, then consider the following:

- This can be an individual activity by having a single student perform all 3 roles.
- For a group of 2 students, combine the CPU and ALU/Memory tasks together and keep the Display separate.
- For a group of 4 students, the ALU/Memory can be broken into two separate tasks – one student (the ALU) can calculate the results and another (the Memory) can record the values. However, it may be more interesting for the students to be arranged in 2 groups of 2.

The roles for the students:

- CPU (Central Processing Unit)
  This student is given the program to run and is responsible for telling the other components (students) what they need to do.
- ALU/Memory (Arithmetic/Logic Unit & Memory)
  This student keeps track of the current values of x and y and performs any math operations requested by the CPU.
- Display
  This student responds to "plot" commands from the CPU by plotting the x,y values on the display grid.

Example:

This is a simple example with three students:

Student **A** is acting as the **ALU/Memory**

Student **C** is acting as the **CPU**

Student **D** is acting as the **Display**

Initial setup:

The group gets 3 worksheets: one each for the CPU, Display and ALU/Memory. Each student takes the sheet for their task and waits for student C (the CPU) to start.

Student D (the Display) needs to be arranged so that the other students cannot see the Display worksheet.

Student C starts by processing instructions (in order) from the program on the CPU sheet – giving the tasks to the other students as required.

For this example, assume that the selected program begins with the following instructions:

Add 4 to x

Add 6 to y

Plot (x,y)

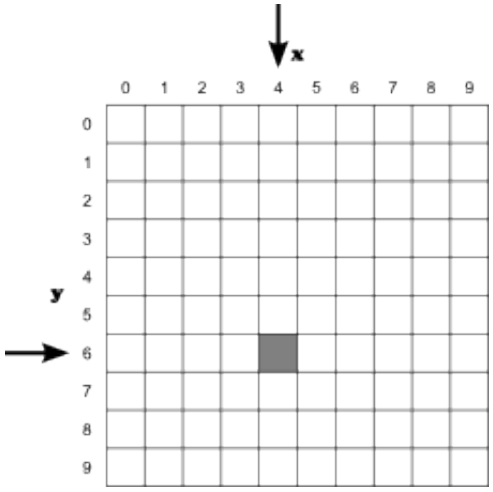
Add 2 to x

Subtract 3 from y

Plot (x,y)

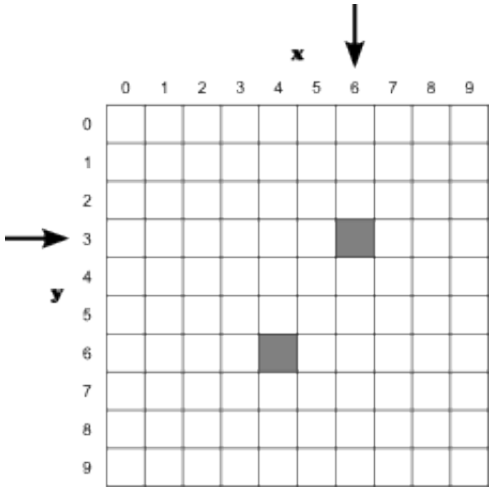
The group should handle these instructions as follows:

- C executes the 1st command by telling A to "Add 4 to x"
- A adds 4 to x and records the updated value in the x-column: 4
- C executes the 2nd command by telling A to "Add 6 to y"
- A adds 2 to y and records the updated value in the y-column: 6
- C executes the 3rd command by:
    - Requesting the current values of x and y from A
    - Telling D to plot the x,y values
- D plots the (x,y) values by:
    - Finding the column that corresponds to the x-value (4) and
    - Finding the row that corresponds to the y-value (6) and
    - Filling the square at the intersection.



The group continues executing the instructions:

- C executes the 4th command by telling A to "Add 2 to x"
- A adds 2 to x and records the updated value in the x-column: 6 (= 4 + 2)
- C executes the 5th command by telling A to "Subtract 3 from y"
- A subtracts 3 from y and records the updated value in the y-column: 3 (= 6 - 3)
- D plots the (x,y) values by:
    - Finding the column that corresponds to the x-value (6) and
    - Finding the row that corresponds to the y-value (3) and
    - Filling the square at the intersection.



...and so on with the remaining instructions.

Follow-up Discussion:

The purpose of the simulation exercise is to give the students a small taste of what it is that computers do.

The single-most important “take-away” from this exercise is the following:

> The computer had no idea what was being drawn on the display - it was just mindlessly following the instructions in the program.

Computers do not “understand” what they are doing. Drawing a picture of a cat is the same (from the computer's perspective) as drawing a picture of a dog - it's just a series of instructions to execute.

Since computers are simply executing the instructions in the program, it is the programmer's responsibility to write the program correctly. If there is a mistake in the program, the computer will still go ahead and try to execute the program as written.

Other possible discussion topics:

- What happens if the CPU tells the display to plot an x or y value that is greater than 9?
    - It could be ignored (not drawn)
    - It could report an error
    - It could be drawn in some other part of the screen
        - On some systems, plotting a large x value would draw the pixel at the beginning of the next row.
- What happens if the CPU is faster than the display?
    - On a real computer system, the CPU generally doesn't wait for the display, so the display will either queue up the drawing commands (and seem sluggish as it struggles to catch up) or it will drop some of the commands (and the display will not be accurate).
- What happens if the display is faster than the CPU?
    - Then it sits around waiting for the next drawing command. This is preferable to the previous situation since the display will always be accurate.
- Note that the computer's display has the origin (x=0, y=0) in the upper left corner and y-values increase as they work down the page. Normal math plots have the origin in the lower left corner and the y-values increase as they work up the page.
    - Why are they different?
        - Traditional math plots work up from the bottom since this is consistent with how we see the world around us. We start from ground-level and measure how tall things are: trees, buildings, ...
        - Computer displays are modeled after how we write on a sheet of paper: we start at the top and move down to the next line when we finish a row.
        - (Of course, computer programs can be written that plot points in the traditional math way.)
    - Which method is better?
        - Ha! Trick question. They both work perfectly fine. You just need to pick one and be consistent.

Credits: