

Program Description

Blocks Microworld is a microworld. That sounds redundant, but it needs saying. A microworld is different from other types of programs. It does not instruct. In fact, it does nothing but what you or your students tell it.

A microworld provides a structured environment within which students act. They act on the things presented by the program; students' actions upon things are constrained by built-in principles. When the principles constraining students' actions are mathematical, the microworld is a mathematical microworld.

Any program provides enablements and constraints. It enables you to do some things, and it constrains you in doing others. Artful design of mathematical, instructional software involves putting enablements and constraints in places that orient students in productive directions.

Blocks Microworld is designed so that when students can *imagine* accomplishing tasks with it, they no longer need it. It is meant to provide concrete, structured experiences with numeration systems and numeration-based algorithms. However, it does not teach these ideas. A *teacher* teaches; Blocks Microworld provides support.

There is another feature of a microworld that makes it unlike other software. First, it is *not* oriented toward developing students' facility with prescribed activities. If you are looking for a program that will teach a prescribed ritual, this isn't it. Its design is meant to facilitate students' creation of meaningful, and perhaps idiosyncratic, methods—but methods grounded on deep understandings of big, important, mathematical ideas.

The ideas that can be addressed directly by using Blocks Microworld are many: place-value numeration, decimal numeration systems, addition, subtraction, division, multiplication, and base systems. Since Blocks Microworld's design enables it to be used in many areas, it cannot be tied to a

specific grade or a specific topic. In fact, you will often find children raising questions that incorporate ideas from an area typically taught separately and unrelatedly to what you are currently teaching.

Instructional Level

Blocks Microworld can be used in

- Kindergarten for instruction in base-ten numeration
- Grades 1-4 for instruction in base-ten numeration, whole-number addition, whole-number subtraction, and whole-number division
- Grades 3-6 for instruction in decimal numeration and operations with decimal numbers.

Also, Blocks Microworld is designed so a teacher may move a child or group of children in any grade level to whatever level of conceptual sophistication is appropriate for them.

More detail on using Blocks Microworld at these levels is given in the Instructor's Guide.

Developer Information

Meet the Developer

Pat Thompson is Professor of Mathematics in the Department of Mathematical Sciences at San Diego State University, where he teaches mathematics and mathematics education. He also is a senior researcher in the Center for Research in Mathematics and Science Education, San Diego State University, where he directs research projects on learning and teaching mathematics. He uses the software he writes in both his research projects in public schools and in his mathematics and mathematics education courses at San Diego State University.

Program Credits

Program, documentation,
and activities by Patrick W. Thompson

Published by Intellimation Library for the
Macintosh

Acknowledgements

The ideas behind Blocks Microworld would not have reach fruition without the research support provided by the National Science Foundation, Grants MDR 87-51381, MDR 89-50311, and MDR 90-96275, and by Apple Computer, Inc., Office of External Research. (Of course, any conclusions or recommendations stated here cannot be held against NSF or Apple Computer.)

User Guide

- *Using this Guide*
- *What's in this Package*
- *What You Will Need to Use the Program*
- *Installation*
- *How to Use the Program*

Using this Guide

Intellimation will describe briefly the chapters of the User's Guide, starting with an introductory paragraph and then discussing each section. We'll also include the best approach to using the guide for both a new user and an experienced user.

What's in this Package?

Intellimation will list all of the product components including both software and supplemental print materials. The reader should be able to use this list to verify that they have everything.

For example:

The (*put name of program here*) contains the following components:

Software components:

- First item
- Etc.

Printed materials:

- First item
- Etc.

What You Will Need To Use the Program

Intellimation will list the system requirements and any additional requirements needed to run the program. We use the developer information from the Submission Kit as a reference.

System Requirements. This will include the computer hardware and peripherals required to make the program run. We will include the minimum system memory, the system version, the HyperCard version, and the amount of hard drive space required. We will list separately all optional hardware and peripherals.

Additional Requirements. Where appropriate, we will include a list of prerequisite educational or programming skills (if any) necessary to effectively use the program. Also, listed will be any additional software programs or textbooks required, including version numbers or copyright years.

For example:

To run (*put name of program here*) you must have the following:

- Macintosh computer with 1Mb of RAM
- System 6.0.5
- HyperCard version 2.0 (Claris)
- etc.

You will need the following materials:

- First item
- Second item
- etc.

In addition, it is assumed throughout this manual that you are familiar with Macintosh basics. If you are not, you should go through the Macintosh tutorial.

Installation Section

Intellimation will describe the installation procedures for your product. If there are known issues about installing your program, please advise us here.

The topics covered in this section are:

Backup Your Disks

Defective Disks

Installing on Your Hard Drive

Single Disk

Compacted

Using a Macintosh Hard Drive System

Using a Macintosh Floppy Drive System

Installing HyperCard Stacks on Your HomeCard

Network Installation

How To Use the Program

The title of this section presumes that *someone* is using this program. I assume that this someone— you —are a teacher, teacher educator, or an education major. That is, I have not written this section for a school student. Introductory activities for school students, and student-oriented tasks, are included in the appendices.

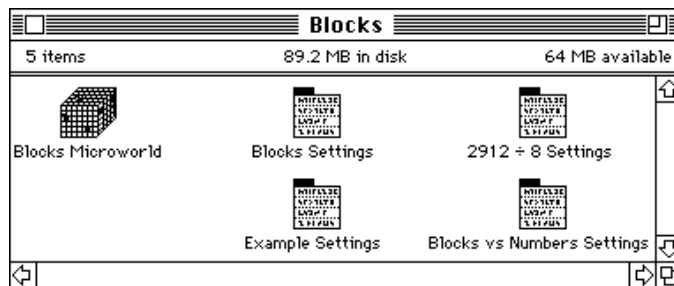
Getting Started

If you haven't already done so, install Blocks Microworld on your computer (see *Installation Section*). Be sure that the file named *Example Settings* is in the same folder as Blocks Microworld.

Start Blocks Microworld

To use Blocks Microworld you must first tell your computer to load it into memory from a disk.

- Open the folder that contains Blocks Microworld. The easiest way to open something is to put the mouse pointer over it and double-click [*click-click*].

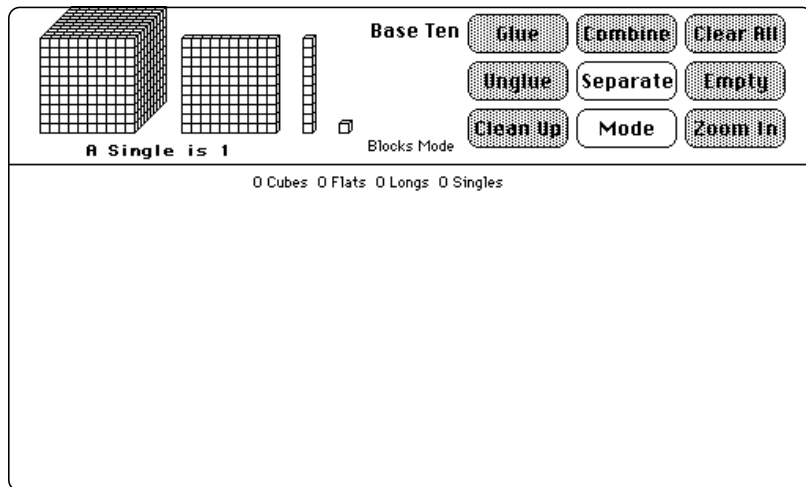


You should see something like this when you open the folder containing Blocks Microworld.

- Put the mouse pointer over the Blocks Microworld icon.

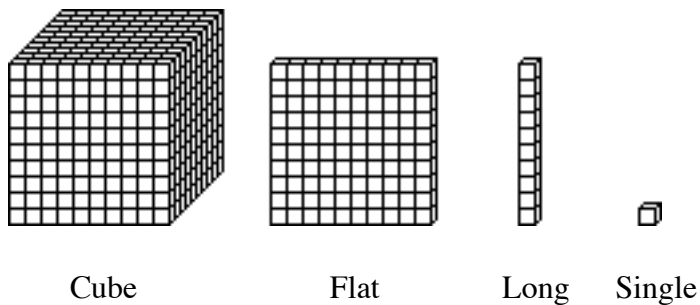
- Double-click [*click-click*] on the Blocks Microworld icon without moving the mouse.

In a few moments you will see a window containing multi-base blocks and nine buttons. What you actually see may differ from the following figure if someone has changed the basic settings since purchasing the program.



A startup screen for Blocks Microworld. The buttons at the top right perform actions. The empty area at the bottom is called the “blocks region.” It is where you store blocks to make numbers.

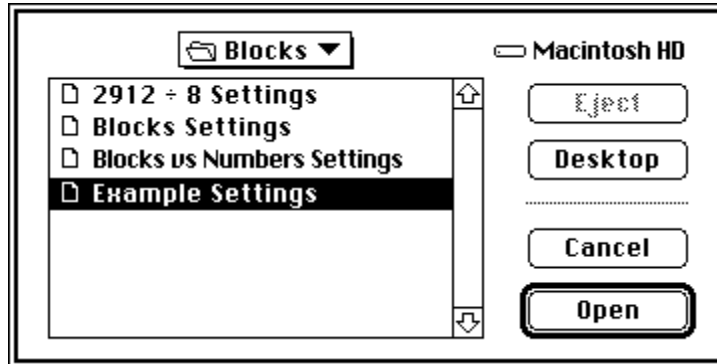
The blocks at the top of the screen (collectively called *master* blocks) are called a *cube*, a *flat*, a *long*, and a *single*, respectively.



Load the settings named *Example Settings*

You can change features of Blocks Microworld to support different aspects of the topics your students are studying. Sets of features can be stored in “settings” files.¹ The settings for this example are stored in the file named *Example Settings*.

- Select *Load Settings...* from the **File** menu.



- Click *Example Settings*
 - Click *Open*
- } Or double-click *Example Settings*.

Bring a block down and move it around.

- Put the mouse pointer over a flat.
- Click the mouse button; hold it down.
- Drag a flat into one of the white spaces at the bottom of the window.
- Move the flat from one place to another to get a feel for how to move blocks.

Select and move a group of blocks

You do not need to move blocks one at a time. You can move several blocks at once.

- Bring down several more blocks of any kind.
- Draw a box around several blocks: Put the mouse pointer over white space (where there are not blocks),

¹ Settings files are useful for structured activities, where you want all computers working alike. They are also useful for tailoring the program for different grade levels.

click, and drag. Draw the box to intersect several blocks.

- Move any of the blocks. The whole selection will move.

Get Rid of a Block.

To get rid of a block:

- Drag it into the master-block region. The cursor will turn into a trash can. When you let go of the block, it will disappear.

To get rid of all blocks at once:

- Click *Clear All*.

Make 3012.

According to the current settings a cube has a value of 1000, a flat has a value of 100, a long has a value of 10, and a single has a value of 1. So, you need to put out 3 cubes, 1 long, and 2 singles.

- Click *Clear All* to start with a clean screen.
- Put the mouse pointer over the master cube. Click the mouse and keep your finger down. Now drag the cube into the left and bottom part of the window. Do this two more times to make 3000.¹
- Drag a long to the same lower-left region as the cubes.
- Drag two singles to the same lower-left region as the cubes.
- Click *Clean Up*.

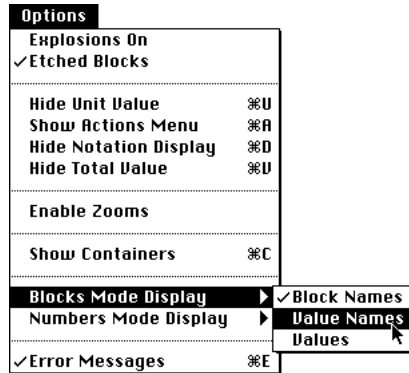
Change the Display

Blocks Microworld's current settings (from *Example Settings*) have it display the number of each kind of block, but not each block's value. After your students are familiar with blocks as

¹ A shortcut: Press the "3" key. Then drag one cube. Three cubes will appear.

screen objects, you will probably want to emphasize the blocks' values.

- Press the mouse button on the **Options** menu.
- Move the pointer down to *Blocks Mode Display*.
- Move the pointer to the right and down to highlight *Value Names*. Now release the mouse button.

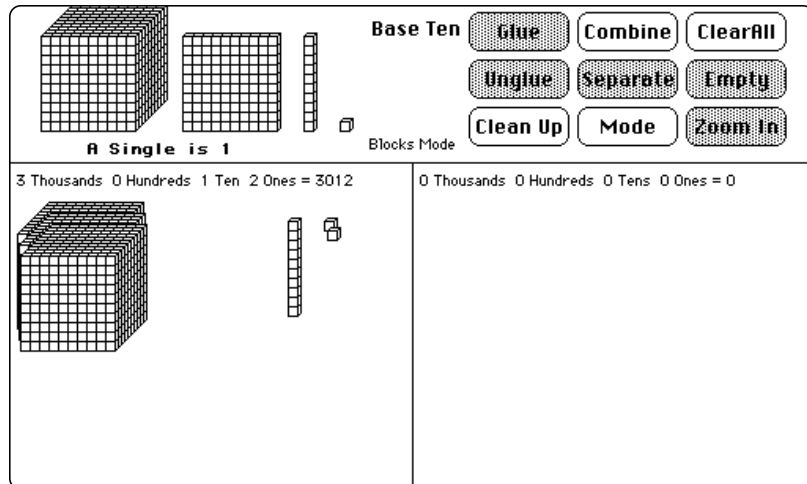


Selecting Value Names as the preferred display in Blocks Mode.

You will notice that the display now says that you have 3 thousands, 0 hundreds, 1 ten, and 2 ones.

Solve a Subtraction Problem

To follow this example, your screen should look as it did when you finished the previous section, showing 3012 on the left hand side, and it should have the settings from *Example Settings*.



Subtract 1024 from 3012.¹

A customary meaning of subtract is “take away.” An image that you hope children have is that “take away” ends up separating an initial amount (a minuend) into two parts—the amount you take away (a subtrahend) and the amount that remains (a difference).

There are enough thousands to take away one thousand.

- Move one thousand from the left region to the right region. To do this, click on a cube, then drag it to the other side while still holding the mouse button down.

You have just subtracted 1000 from 3012. Notice how the display has changed.

There are not enough tens to remove two tens. You can change one thousand into ten hundreds.

- Click on a cube without moving it. It should become highlighted.²

¹ The method illustrated here is neither a conventional method nor the only method. The instructional materials, in the Appendices, emphasize repeatedly that there is no “correct” method for operating on numbers. There are customary methods, unconventional methods, and personally meaningful methods. It is a matter of social agreement as to what methods are adopted as customary.

² The buttons *Glue* and *Unglue* are now active. Buttons become active when there is a potential for using them.

- Click *Unglue*.

The cube, which represented 1000, exploded into ten flats, each of which represent 100. One thousand is the same as 10 hundreds. You now have 1 thousand, 10 hundreds, 0 tens, and 2 ones.

There still are not enough tens to take away two tens. You can change one hundred into ten tens.

- Click on a flat without moving it.
- Click *Unglue*.

The flat, which represented 100, exploded into ten longs, each of which represent ten. One hundred is the same as ten tens.

Notice the number display on the left. It now says that 1 thousand, 9 hundreds, 11 tens, and 2 ones is the same as 2012.

There are now enough tens to take away two tens.

- Press the “2” key. (You don’t need to hold it down. Also, nothing will happen until you do the next step.)
- Click on a long. Two longs should become highlighted.
- Drag one of the longs to the right side. Both longs will move.

The screen shows the result of taking away 1020 from 3012. You still need to take away four ones, but there are not enough of them.

- Click on a long.
- Click *Unglue*.
- Press the “4” key.
- Click on a single. Four singles should become highlighted.
- Move one of the singles to the other side. All four singles will move.

You are finished taking away 1024 from 3012. Removing 1024 from 3012 leaves 1988.

Confirm

The idea of “checking” subtraction by adding can be made explicit in Blocks Microworld. We started with 3012 and split it into two parts. If we recombine the two parts, we should get 3012.

- Click *Combine*.

The two collection are now (logically) one collection. When 3012 is separated into two amounts, these two amounts, recombined, make 3012.

- Click *Separate*.

The collection is split again into two parts.

Undo Your *Unglues*

When you combine the two collections, the re-combined collection is not in conventional form. We typically do not represent numerical values using “digits” greater than nine.

- Click *Combine* if it is active. (You should have just one blocks region.)

The display should say that you have 2 thousands, 9 hundreds, 10 tens, and 12 ones.

- Select 10 ones. To do this, press the “0” (zero) key. Then click a single. Ten singles should become highlighted.
- Click *Glue*. The singles are glued together into one long. Ten ones is the same as one ten.

The display should now say that you have 2 thousands, 9 hundreds, 11 tens, and 2 ones.

- Select 10 tens. To do this, press the “0” (zero) key. Then click a long. Ten longs should become highlighted.
- Click *Glue*. The longs are glued together into one flat. Ten tens is the same as one hundred.

The display should now say that you have 3 thousands, 0 hundreds, 1 ten, and 2 ones—the configuration you started with.

Represent Your Subtraction

An intended use of concrete manipulatives, and one of the most important uses, is that they provide a basis in concrete action for children’s use of mathematical notation. The actions you took to solve the subtraction problem $3012 - 1024 = ?$ can be summarized like this:

- Make 3012
- Remove 1 thousand
We have removed a total of 1000 and we are left with 2 thousands, 1 ten, and 2 ones
- Turn 1 of the two thousands into 10 hundreds
We have removed a total of 1000 and we are left with 1 thousand, 10 hundreds, 1 ten, and 2 ones
- Turn 1 of the ten hundreds into 10 tens
We have removed a total of 1000 and we are left with 1 thousand, 9 hundreds, 11 tens, and 2 ones
- Remove 2 of the eleven tens
We have removed a total of 1020 and we are left with 1 thousand, 9 hundreds, 9 tens, and 2 ones
- Turn 1 of the nine tens into 10 ones
We have removed a total of 1020 and we are left with 1 thousand, 9 hundreds, 8 tens, and 12 ones
- Remove 4 of the twelve ones
We have removed a total of 1024 and we are left with 1 thousand, 9 hundreds, 8 tens, and 8 ones

The above “story” is a representation of what you did to solve the problem $3012 - 1024 = ?$. It is an unconventional representation, but one from which children can benefit by writing as they learn subtraction within base-ten numeration.

Another type of representation of what you did is to record actions in a compact form on a standard statement of the initial problem. Since you solved this problem unconventionally (by “starting from the left” so to speak) the notational representation will be unconventional.

The following figure is one way to capture your actions notationally. It is not proposed as an alternative to a standard procedure. Rather, this is one way to represent *the actions you actually took* in solving $3012 - 1024 = ?$. It tells a story about what you did—the same story as given above.

$$\begin{array}{r}
 3 \quad \overset{9}{\cancel{0}} \quad \overset{1}{1} \quad \overset{1}{2} \\
 - 1 \quad 0 \quad 2 \quad 4 \\
 \hline
 \cancel{2} \quad 9 \quad \cancel{9} \quad 8 \\
 1 \qquad \qquad 8
 \end{array}$$

A representation of your actions in solving $3012 - 1024 = ?$. Start from the left. The thousand that you changed into 10 hundreds was one of the 2 thousands that remained after removing 1 thousand. Also, the ten that you changed into 10 ones was one of the tens that remained after removing 2 tens.

I encourage you to come up with your own notational method for telling the story of your actions. Try it now, on the figure below. Just be sure that you are faithful to your actual actions with the blocks.

$$\begin{array}{r}
 3 \quad 0 \quad 1 \quad 2 \\
 - 1 \quad 0 \quad 2 \quad 4 \\
 \hline
 \end{array}$$

Blocks Mode and Numbers Mode

Blocks Microworld operates in one of two modes—*Blocks Mode* and *Numbers Mode*. You switch from one mode to the other by clicking the *Mode* button. If you are continuing from the previous sections, you have been using Blocks Microworld in *Blocks Mode*.

Blocks Microworld's two modes are intended to support two different kinds of interaction with blocks. The different kinds of interaction are designed to develop complementary aspects in students' understanding of numeration and arithmetic operations.

Blocks Mode

When Blocks Microworld is in *Blocks Mode*, you glue and unglue blocks by selecting the blocks directly and then clicking *Glue* or *Unglue*.

Children's attention is oriented explicitly to blocks' structures when using the program in *Blocks Mode*. The blocks' structures support children's constructing images of converting one value into ten one-tenths of that value, and vice versa.

Blocks Mode also has the added convenience that you may do multiple gluings or unglueings simultaneously. You may unglue, for example, three flats into 30 longs at once, or glue 30 longs into 3 flats at once.

Numbers Mode

When Blocks Microworld is in *Numbers Mode*, the buttons named *Glue* and *Unglue* are renamed as *Carry* and *Borrow*. In *Numbers Mode*, you glue and unglue blocks by selecting a *digit* in the expanded-notation display and then clicking *Carry* or *Borrow*.

In *Numbers Mode* children are oriented to constructing relationships between actions on symbols (e.g., "carrying" a 1) and the meanings of those actions (turning 10 of one value into

1 value 10 times as great). When you wish your students to focus explicitly on relationships between symbols and values, you can request that they switch to *Numbers Mode* (and perhaps save their settings so that this is the mode in which their copy of Blocks Microworld begins).

Same Story, Different Themes

This example will have you do the same task in *Blocks Mode* and in *Numbers Mode*. You will solve $19.94 + 3.78 = ?$. Please use the following settings:¹

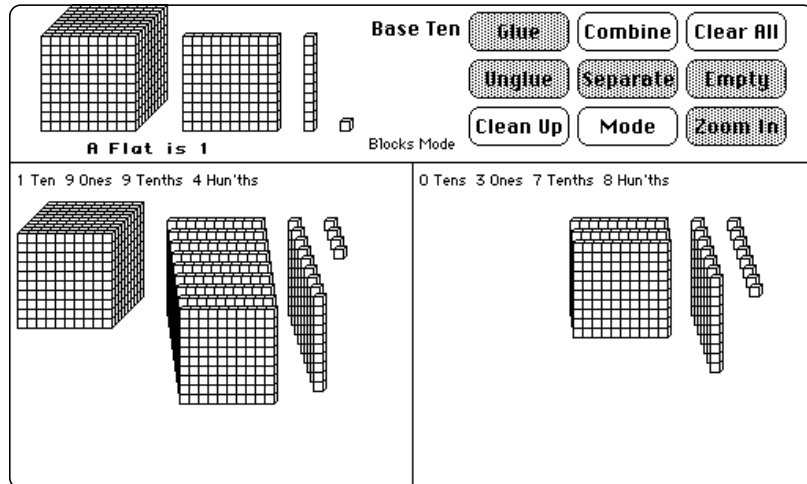
- Click the *Separate* button if it is active.
- In the **Unit** menu, select *A Flat is 1*.
- In the **Options** menu, in the *Blocks Mode Display* submenu, select *Value Names*.
- In the **Options** menu, in the *Numbers Mode Display* submenu, select *Value Names*.
- In **Options** menu: if you see *Hide Total Value*, select it. If you see *Show Total Value*, don't do anything.
- In the **Base** menu: select *Base ten*.

Perhaps you noticed a new feature—*A Flat is 1*. By stating that a flat has a value of one, you have implied that a cube has a value of ten, a long has a value of one-tenth, and a single has a value of one-hundredth.

Solve $19.94 + 3.78 = ?$ In Blocks Mode

- Put one ten, nine ones, nine tenths, and four hundredths in the left-hand region. (*Recall a shortcut: Press a number key, then drag a master block.*)
- Put three ones, seven tenths, and 8 hundredths in the right-hand region.

¹ Or, load the settings named *Blocks vs Numbers Settings*.



- Click *Combine*.

The display says that you have a total of 1 ten, 12 ones, 16 tenths, and 12 hundredths. Now, to standardize the result.

- Select 10 hundredths. (Press “0” and click on a single.)
- Click *Glue*.

Ten hundredths were turned into one tenth. You now have 1 ten, 12 ones, 17 tenths, and 2 hundredths.

- Select 10 tenths. (Press “0” and click on a long.)
- Click *Glue*.

Ten tenths were turned into one one. You now have 1 ten, 13 ones, 7 tenths, and 2 hundredths.

- Select 10 ones. (Press “0” and click on a flat.)
- Click *Glue*.

Ten ones were turned into one ten. You now have 2 tens, 3 ones, 7 tenths, and 2 hundredths. The sum of 19.94 and 3.78 is 23.72.

Solve $19.94 + 3.78 = ?$ In Numbers Mode

In *Numbers Mode*, you act on the digits of a number in order to glue or unglue blocks. By solving $19.94 + 3.78 = ?$ in *Numbers Mode* you will get a sense of how children’s will experience the same problem differently than in *Blocks Mode*.

- Click *Clear All* to begin with a fresh screen.
- Click *Mode*.

The “mode” notice, in the center of the screen, should say “Numbers Mode.” Also, the *Glue* and *Unglue* buttons changed names. They are now *Carry* and *Borrow*.

Start the same way as before. Put 19.94 on one side and 3.78 on another.

- Click *Combine*.

The display says that you have a total of 1 ten, 12 ones, 16 tenths, and 12 hundredths. Now, to standardize the result.

- Click on the “12” next to the word “Hun’ths.”

The “12” should become surrounded by a box, . This is to show you that your click was on target. Notice that the *Carry* and *Borrow* buttons are now active.

- Click *Carry*.

Ten hundredths were turned into one tenth. You now have 1 ten, 12 ones, 17 tenths, and 2 hundredths.

- Click on the “17” next to the word “Tenths.” You should see 17 in a box, .
- Click *Carry*.

Ten tenths were turned into one one. You now have 1 ten, 13 ones, 7 tenths, and 2 hundredths.

- Click on the “13” next to the word “Ones.” You should see 13 in a box, .
- Click *Carry*.

Ten ones were turned into one ten. You now have 2 tens, 3 ones, 7 tenths, and 2 hundredths. The sum of 19.94 and 3.78 is 23.72.

What I Hoped You Noticed

In *Blocks Mode*, children's attention is oriented to blocks and actions on them as they work. This is proper when the base-ten structure of the blocks has yet to become second nature for them.

In *Numbers Mode*, children's attention is oriented more to numerical symbols as they work. When they are thinking about numerical values of blocks, but are constrained to work on symbols, they have many occasions to make explicit connections among blocks, values, and symbols.

Quick Reference

In this section you will find explanations of Blocks Microworld's features: its menus, buttons, and command keys.

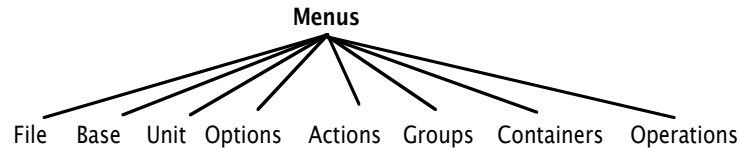
In designing Blocks Microworld, I had to decide what to put in menus and what to put in buttons. This was my rule of thumb: Those things a student is most likely to use routinely while solving a problem are in buttons. Things about which a student must make a decision, such as what block has a value of 1, or how many blocks to put in each container, are in menus. My intention for putting "decision" items in menus is that, because students must break from routine activity to select a menu item, it causes them to make their assumptions public. When students make their assumptions public, the likelihood that they notice their importance increases.

Another class of items in the menus are there primarily for pedagogical purposes. Students probably will not choose to hide the notation display or hide total values, but *you* may want them to do so in order to focus their attention on something important. Pedagogical issues are mentioned briefly in this section. They are addressed more broadly in the *User's Guide* and the *Appendices*.

Menus

Menus are where you set the main features of Blocks Microworld's behavior.

Blocks Microworld's menus are:



Not all of Blocks Microworld's menus appear automatically. Some menus may be installed so that they appear whenever the program starts up (e.g., **Actions**); others appear only under certain circumstances (**Groups**, **Containers**, and **Operations**).

File Menu

File	
New	⌘N
Close	⌘W

Load Settings ...	
Save Settings	
Save Settings As ...	

Page Setup ...	
Print...	⌘P

Copy Window	

Quit	⌘Q

The File menu is where you find regular Macintosh stuff, like loading and saving settings, printing, and quitting. It also is the menu that allows you to copy Blocks Microworld's screen to paste it into other programs.

New

Select *New* to make a new Blocks World window. You can have up to five windows open at one time on a big-screen

Macintosh, fewer on smaller-screen Macintoshes. You will need to “zoom” windows to make multiple windows fit a smaller-screen Macintosh (see the **Options** menu).

My intention for allowing multiple windows was for comparing different units and different bases (see discussion of *Zooms Enabled* in the **Options** menu).

To highlight the effect of different units you might have students make two windows; in one window make a flat have a value of 1 and in the other make a cube have a value of 1/10. Then ask them to put out blocks in each window so that both windows show a value of 2.42.

To highlight issues surrounding different bases, you might, for example, create two windows, both in base ten and both having a single be 1. Put the same number of blocks out in each window, then change one window to base seven and the other to base nine. The blocks in each window have identical total values (they have the same number of singles), but they “look” different. Another activity might be to have students make two windows in different bases (e.g., six and eight), tell them to put out 2 cubes, 3 flats, 5 longs, and 3 singles in one window, then ask them to put out enough blocks in the other window so that they will be identical when they change them both to base ten.

Close

Select *Close* to close the active (front most) window. You cannot close the Blocks World 1 window.

Load Settings...

Select *Load Settings...* to bring up the standard Macintosh file dialog.

This item, in conjunction with *Save Settings As...*, allows you to tailor Blocks Microworld for different activities. You can choose the menu options you wish students to use and then save them under the activity’s name (see *Save Settings As...*). You can then copy the settings file onto students’ work disks, or store the settings in a folder that all students can access over a network.

Save Settings

Select *Save Settings* to make the current settings become Blocks Microworld's default settings. When you select *Save Settings*, Blocks Microworld stores the current settings in a file named *Blocks Settings* in the program's home folder.¹

When you start Blocks Microworld, it looks in its home folder. If it finds a file named *Blocks Settings*, it reads it into memory and uses those settings as its default settings.

One way to customize Blocks Microworld for your class is that you run the program, choose the settings you want the program to start with and then select *Save Settings*. Then copy the file *Blocks Settings* onto each students' work disk, putting the file into the same folder as Blocks Microworld.

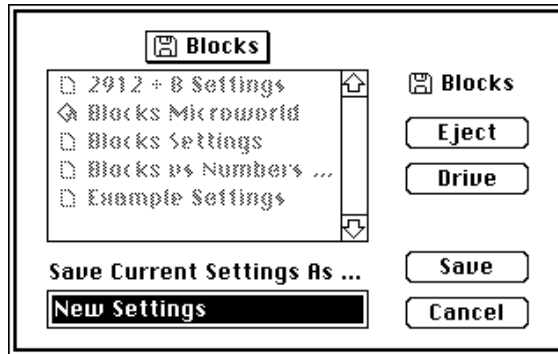
If your students are using a network, then put the settings file where they can all access it. Have them each load the file (see *Load Settings...*), then have each student select *Save Settings* to store their personal version of these settings.

A low-tech, but efficient way, to customize each students' version of the program is to recite the settings you want them to select while they are at their computer. Then tell them to select *Save Settings*.

Save Settings As ...

Select *Save Settings As...* when you want to save the current settings, but not make them the program's default settings. You will see a dialog box like in the following figure.

¹ If the program is on a floppy disk, but not in a folder, Blocks Microworld sees the floppy disk as a folder and puts *Blocks Settings* onto the disk.



Type the name you want for this settings file. The highlighted text (“New Settings”) will be replaced by what you type. Press Return, or click **Save** when you are done.¹

Page Setup ...

Select *Page Setup* to choose whatever options provided by your printer.

Print ...

Select *Print...* to print the active window at your printer.

Copy Window

Select *Copy Window* to copy the current window to your computer’s clipboard. If you are running Multifinder, you can switch to your word processor or graphics program and paste the screen into a document. The pictures of Blocks Microworld’s screens in this manual were all gotten by using *Copy Window*.

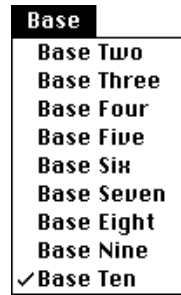
A hint: If you want nice, crisp reductions of a Blocks Microworld screen, don’t reduce screens in a graphics program. Instead, reduce them in your word processing program or page-layout program. Reducing a Blocks Microworld screen in a graphics program usually produces ugly, ragged pictures.

¹ Be sure to notice ahead of time the drive and folder in which you are saving your settings. If you are like me, you can easily “misplace” a file by saving it somewhere other than where you thought.

Quit

Quit the program.

Base Menu



A numeration system is a method for representing numerical values. In base-ten numeration, the position of a digit tells you the value of that place. Reading from right-to-left, the first place in a numeral has a value of 1, or 10^0 . The second place has a value of ten, or 10^1 . The third place has a value of 100, or 10^2 . And so on. Each succeeding position has a value 10 times as great as the position to its right. The digit in a position tells how many of that position-value are in the number.

Base-ten blocks model this structure. The smallest block is a single. A long is ten singles, or ten ones, or 10^1 . A flat is ten longs, or ten tens, or 10^2 . A cube is ten flats, or ten hundreds, or 10^3 .

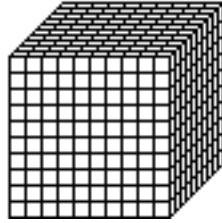
There is nothing unique about ten as a base. We can use this same principle with, say, nine as a base. The single is still one, but now a long will have nine singles. However, we would still represent a long as 10—1 group of nine and 0 ones. A flat will be nine longs, or 10 10's (remember, in base nine); a cube will be nine flats, or 10 100's (again, in base nine).

The **Base** menu allows you to select a base from two to ten. Whatever blocks are present on the screen are converted to the new base. The number of singles will be preserved; they will be regrouped according to the base you select.

My editor will cringe at putting an example in a “quick reference” guide. Nevertheless ...

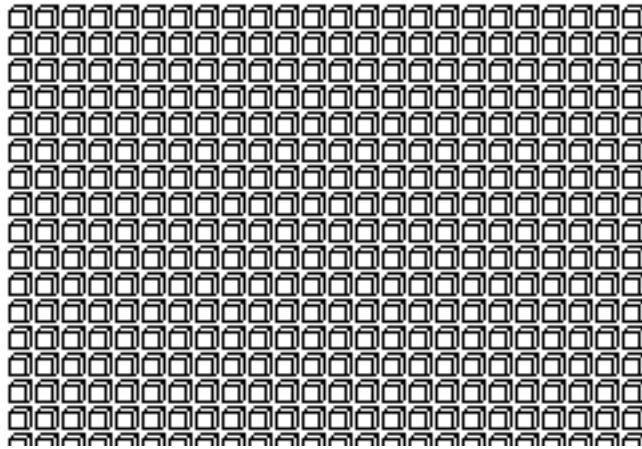
Here is a cube in base ten. Its value is represented “1000.”

1 Cube 0 Flats 0 Longs 0 Singles = 1000



To change it to base nine, first think of the cube as nothing but a bunch of singles stacked neatly. Then, think of all those singles laid out separately.

1 Cube 0 Flats 0 Longs 0 Singles = 1000

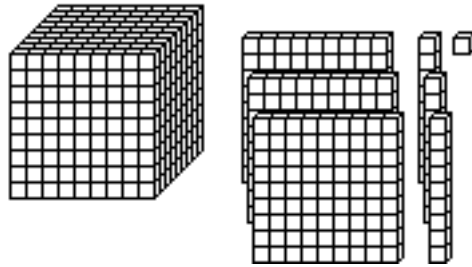


Now, regroup these loose singles according to the conventions of base nine. One base-nine cube has $9 \times 9 \times 9$, or 729 (speaking in base ten) singles. So, we can make one base-nine cube. That leaves 271 loose singles.

A base-nine flat has 9×9 , or 81 singles. We can make three base-nine flats out of what's left. That uses 243 of the 271 loose singles, so we are left with 28 loose singles.

A base-nine long has nine singles. So, we can make three base-nine longs, leaving one loose single.

1 Cube 3 Flats 3 Longs 1 Single = 1331



We started out with 1000 in base ten. We ended with 1331 in base nine. But we still have the same *collection* of singles. They are just grouped differently.

You should note that your choice of base is independent of your choice of unit. Blocks Microworld can represent “decimals” in bases other than ten. Let a flat have a value of 1 in base nine. Then 0.01 in base nine means you have 0 flats, 0 longs, and one single. A flat has nine longs, so a long has a value of one-ninth. A flat has eighty-one singles, so a single has a value of one eighty-oneth. So, 0.01 in base nine stands for what we would normally call $\frac{1}{81}$.

This last example raises an interesting point. It is only when a single is one that the display’s value remains unchanged when changing bases. If a single is not one, then the number of singles remains unchanged, but the display’s total value does change. To understand, in principle, how the total value changes is nontrivial.

If you are an elementary school or middle school teacher, you might use these features in enrichment activities, or for your own explorations. If you are a teacher educator, you and your students probably will make considerable use of this menu.

One final note: You may have noticed that Blocks Microworld says “ten” for “10” regardless of the base. There are two good reasons for this. First, what else might we call it? “One zero?” Second, if you’ve ever taught number bases, you’ve seen students continue to say “ten” when reading “10,” to say “one hundred” when reading “100,” and so on. So, rather than fight students’ natural inclinations, I accompany number-base instruction with a story that builds on a well-known aspect of

children’s ability to speak several languages and not get them confused. They do not think of speaking different languages. Instead, they contextualize their language use. They know to speak one way to grandmother, another way to peers, and perhaps yet another way to parents.

It is in this vein that I tell a story, like this.

There is a far-away land where people speak English, just like us, and their culture is very similar to ours. There is one very big difference between them and us, though. They have eight fingers, and they count like this: One, two, three, four, five, six, seven, ten, eleven, twelve, thirteen, fourteen, fifteen, twenty, twenty-one, and so on.

Now, pretend that a visitor, named Fred, came from this land to the United States, and you are Fred’s host. Fred says he is twelve years old. How old would you say Fred is?

Fred loves jelly beans, which he cannot get at home. You and he are in a candy store, and Fred asks for thirty-six jelly beans. What should you say to the store owner about how many jelly beans to give Fred?

And so on. The idea is that we have different ways of naming amounts—our customary way, and Fred’s way. If students become confused about what, for example, “1237” stands for in base eight, ask them whose 1237 it is—ours or Fred’s?

Unit Menu

Unit
✓ Single is 1 Single is 10 Single is 100 Single is 1000
Long is 1 Flat is 1 Cube is 1
Cube is 1/10 Cube is 1/100 Cube is 1/1000

You can use the **Unit** menu to change blocks' values. This feature allows you to highlight the ratio nature of numbers.

All numerical values are in comparison to a unit. So, the value of any block depends on what you assume has a value of one.

- If a flat has a value of one, then a cube has a value of ten, a long has a value of one-tenth, and a single has a value of one-hundredth.
- If a cube has a value of one-hundredth, then a flat has a value of one-thousandth, a long has a value of one ten-thousandth, and a single has a value of one hundred-thousandth.
- If a single has a value of one hundred, then a long has a value of one thousand, a flat has a value of ten thousand, and a cube has a value of one hundred thousand.

The **Unit** menu allows you to use Blocks Microworld in instruction on decimal numbers. The Appendix contains a detailed example of introductory instruction on addition and subtraction of decimal numbers.

The **Unit** menu also allows you to move young children to deep understandings of the numeration system. Young children often reason about blocks as blocks, not as things having numerical value. This is natural and productive, at first. But this view will persist unless children have occasions to think beyond the concrete level.

A productive use of Blocks Microworld with young children is to suggest tasks to them that use, say "Single is 10," to assign numerical value to a display. You might have them hide the display's total value (see **Options** menu), or hide the notation display completely, and have them show the display only after they, themselves, have committed to an interpretation of the number represented by their display.

Options Menu

Options	
Explosions On	
✓ Etched Blocks	

Hide Unit Value	⌘U
Show Actions Menu	⌘A
Hide Notation Display	⌘D
Show Total Value	⌘U

Enable Zooms	

Show Containers	⌘C

Blocks Mode Display	▶
Numbers Mode Display	▶

✓ Error Messages	⌘E

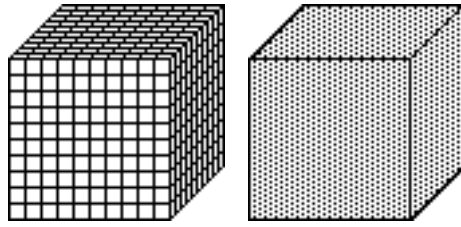
The **Options** menu allows you to control the way Blocks Microworld operates and how it displays information. It is the most diverse menu, containing mostly pedagogically-oriented features. It also provides Blocks Microworld's entry into concepts of division and, indirectly, multiplication.

Explosions On

Explosions have to do with changing from one base to another. If explosions are on, Blocks Microworld will “explode” the current blocks into a collection of singles before regrouping them according to the new base (see the discussion of the **Base** menu). If explosions are not on, Blocks Microworld skips the step of laying out loose singles and goes directly to putting out the re-structured singles.

Etched Blocks

Blocks can be drawn in either of two formats: etched or plain.



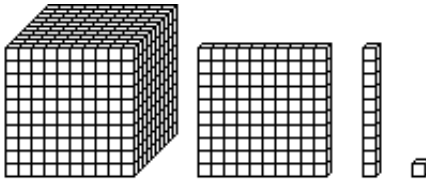
Etched Block Plain Block

Etched blocks suggest that blocks larger than a single are made of smaller blocks, so children interpret etched blocks as having numerical value more easily than they can interpret plain blocks as having numerical value. In regard to number, plain blocks are more abstract—they demand that *seeing* blocks as having numerical value be second nature to students.

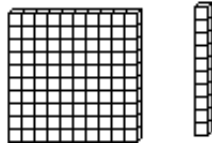
Some teachers prefer etched blocks, some prefer plain blocks. You must judge for yourself which is more appropriate for your students at any given moment.

Hide Unit Value/Show Unit Value

Sometimes it is desirable to have students reason inversely about value in relation to units. For example, imagine that you are using Blocks Microworld with an overhead projector and the following screen is what your students see. Ask them, “What unit value have I selected?” (Answer: *A single is 100.*)



0 Cubes 1 Flat 1 Long 0 Singles = 11000



What is the unit value I selected?

Show Actions Menu/Hide Actions Menu

Select this item to show or not show the **Actions** menu (which is explained later in this section). If the **Actions** menu is showing, this item reads *Hide Actions Menu*.

Hide Notation Display/Show Notation Display

Sometimes students use the notation display as a crutch, looking at it immediately without interpreting it in relation to blocks and values of blocks. If you notice this, then you might request that they hide the notation display while working. Or, you might force the issue by using Blocks Microworld in a demonstration with its notation display hidden.

Show Total Value/Hide Total Value

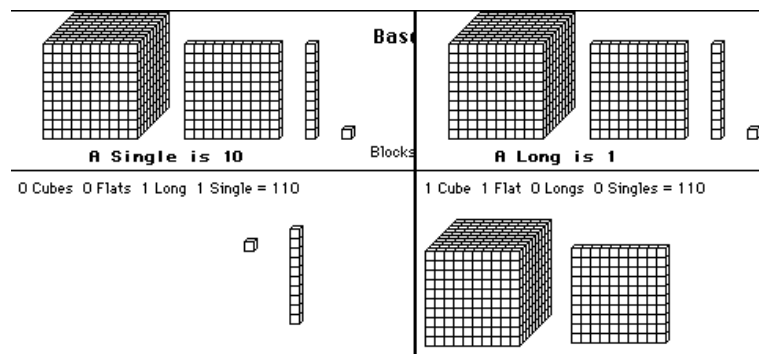
Sometimes students discover prematurely that they can ignore the expanded notation display and look immediately at the total values. If you decide that you want them to focus on the expanded notation display, then you might request that they hide total values while working. Or, you might force the issue by using Blocks Microworld in a demonstration with total values hidden.

Enable Zooms

Select *Enable Zooms* to activate the *Zoom In* buttons in all Blocks World windows.

To “zoom” a window (this is Mac jargon) means that, with one click, you cause a window to shrink or enlarge automatically.

The *Zoom In* button, when active, causes all open Blocks World windows to “zoom” to a reduced size and to stack neatly across the screen, as shown in the next figure.¹



Two Blocks World windows that have been zoomed-in. Both are in base ten. In one, a single is 10. In the other, a long is one. The displays in both windows represent the number 110.

When you zoom a window in, the window’s buttons become hidden—you can’t click any of them. It is for this reason a new menu appears—the **Operations** menu. Its purpose is merely to allow you to do what you would normally do with buttons while a window is zoomed in. The **Operations** menu will disappear when you zoom your windows out.

Show Containers/Hide Containers

Select this item to show Blocks Microworld’s containers if they are hidden, or to hide them if they are shown.

¹ See the discussion of *New* in the **File** menu for the reason for including a “zooming” feature.

When you show Blocks Microworld's containers, two additional menus will appear: **Groups** and **Containers**. I need to discuss those menus in order to talk about containers and their uses, so I'll forward you to those menus for further information.

Blocks Mode Display/Numbers Mode Display

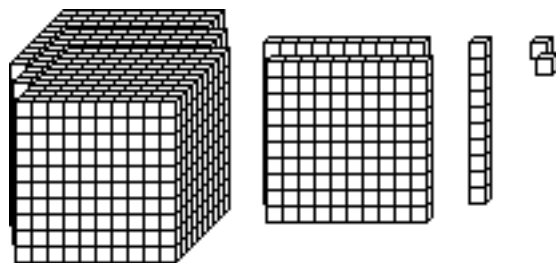
These are submenus. When you select either, a new menu will appear to the side.



The submenu items are *Block Names*, *Value Names*, and *Values*. Both submenus act the same. The *Blocks Mode Display* submenu controls how the notation display is presented in Blocks Mode. The *Numbers Mode Display* submenu controls how the notation display is presented when Blocks Microworld is in Numbers Mode.

The next figure illustrates the three different notation displays.

A Long is 1



Block Names display : 3 Cubes 2 Flats 1 Long 2 Singles = 321.2
Value Names display: 3 Hundreds 2 Tens 1 One 2 Tenths = 321.2
Values display: 300 + 20 + 1 + 0.2 = 321.2

Here is a brief history of these display options. Blocks Microworld initially provided only a Block Names display. In two research projects I found that both students and teachers often followed the program's lead and referred to the blocks by

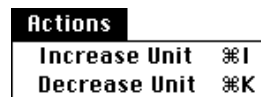
their shape, not by their value. The effect of their orientation toward shape instead of value showed up later. Those students who infrequently attended to matters of numerical value learned a lot about blocks, but very little about number and representations of numerical value.

This motivated me to change the program so that it presented only the *Value Names* option. In subsequent research I found that this change was often beneficial, but not always. Some students needed to remain with block names, and some students quickly surpassed the need for value names. So, the current design incorporates the flexibility for a student, or you, to decide what display best fits the moment.

Error Messages

When this item is checked, Blocks Microworld will beep and display an error message. When this item is unchecked, Blocks Microworld will just beep.

Actions Menu

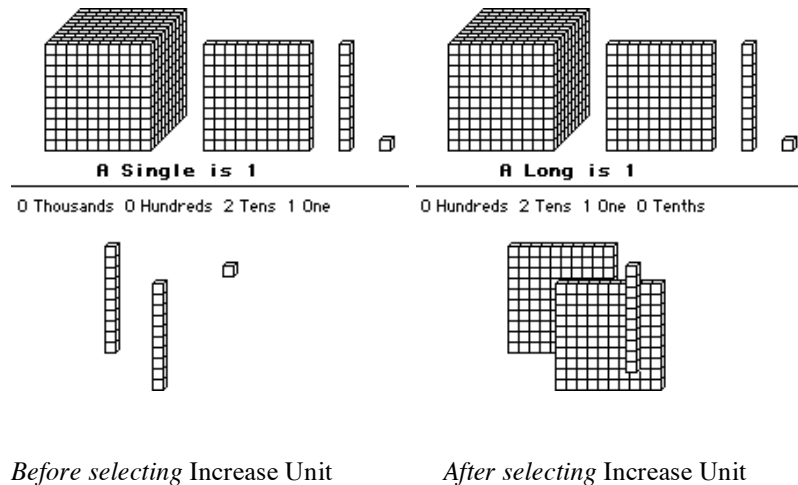


Select *Show Actions Menu* in the **Options** menu to see the **Actions** menu. The **Actions** menu also appears any time you show containers.

The **Actions** menu allows two very special operations. It allows you to increase the unit by a factor of 10, and it allows you to decrease the unit by a factor of 10—both operations resulting in a display having the same value as before the operation.

Suppose the current unit value is *A single is 1*, and that the current display shows 2 longs and 1 single. To use this setup in solving $21 - 2.1 = ?$ is problematic. You can't unglue a single, and if you make a long be 1, then you have blocks worth 2.1 and not 21.

Select *Increase Unit*. The unit is increased from a single to a long, each long is replaced by a flat, and each single is replaced by a long. You can now unglue one one into ten tenths, and the blocks still have a total value of 21. The effect of *Increase Unit* is illustrated in the next figure.

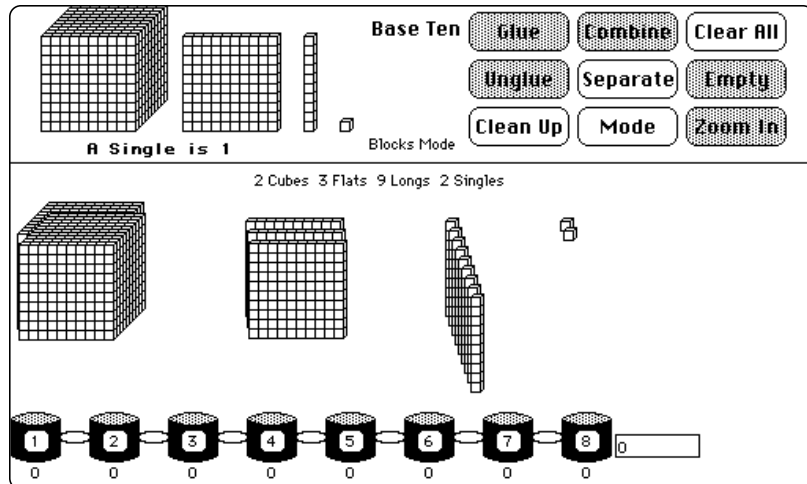


Groups Menu and Containers Menu

Groups	Containers
<input checked="" type="checkbox"/> Do Not Group	<input checked="" type="checkbox"/> Link Containers ⌘L
<hr/>	<input type="checkbox"/> Hide Total ⌘T
1 per Container ⌘1	Show 2 Containers
2 per Container ⌘2	Show 3 Containers
3 per Container ⌘3	Show 4 Containers
4 per Container ⌘4	Show 5 Containers
5 per Container ⌘5	Show 6 Containers
6 per Container ⌘6	Show 7 Containers
7 per Container ⌘7	Show 8 Containers
8 per Container ⌘8	<input checked="" type="checkbox"/> Show 8 Containers
9 per Container ⌘9	Show 9 Containers
10 per Container ⌘0	Show 10 Containers

These two menus act in concert to influence Blocks Microworld's behavior, so I will discuss them together instead of separately.

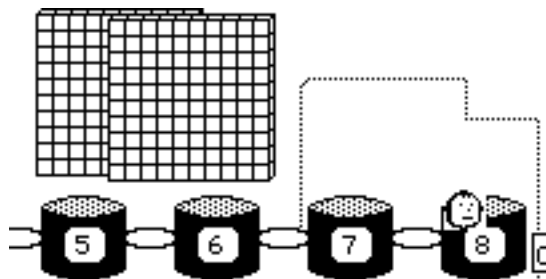
The **Groups** menu and the **Containers** menu appear only when you select *Show Containers* in the **Options** menu. A typical screen is shown in the next figure.



Blocks Microworld's window when you select Show Containers in the Options menu. The purpose of linked containers is to have children develop methods to share blocks among them so that each container gets the same amount.

Containers are things you drop blocks into. Select a collection of blocks, then drag them to a container. Blocks Microworld watches the mouse pointer when you are dragging blocks, so you "hit" a container when the *mouse pointer* is over a container.

If the containers are linked (as shown above), then Blocks Microworld must be able to share the blocks evenly among containers before it can put any blocks into a container.



I tried dropping two flats into eight linked containers. As soon as I moved the mouse pointer over a container, Blocks Microworld hinted, by shaking its head "no," that it could not share two blocks evenly so that each container gets a whole block.

If the containers are not linked, then all the blocks you are dragging will go into only the container you dragged them to.

The **Groups** menu is where you set the number of blocks that must go into each container if any container is to accept blocks.

The **Containers** menu is where you set the number of containers to display and whether or not they are linked.

An Example

The interaction between the two menus is intuitive in action, and complex in description. So, first a brief example, then a general description.

- Combine the screen, go into *Blocks Mode*, and clear all blocks.
- Show the containers (select *Show Containers* in the **Options** menu, or press \square -C).
- Select *Show 8 Containers* in the **Containers** menu.
- Link the containers if they are unlinked (*Link Containers* in the **Containers** menu).
- Put out 2 flats.
- Drag a flat to any container. To drag a flat to a container, drag it so that the *mouse pointer* is over a container.

A “shaking head” tells you that Blocks Microworld cannot share this one block among 8 containers. You need a number of blocks that is a multiple of 8.

- Select the two flats (draw a box around them, or press “2” and click on one).
- Click *Unglue*.

Now there are 20 tens where there were 2 hundreds. We can share 16 of these tens by giving 2 tens to each of the 8 containers.

- Select *2 per Container* in the **Groups** menu (or, press \square -2).

- Click on any long. Blocks Microworld selects 16 longs.

Your settings say you want to put 2 blocks into each container, and there are 8 containers. Blocks Microworld anticipates you will need 16 of the type of block you clicked on.

- Drag the selected blocks to any container. Click on any block, then drag *the mouse pointer* over any container. Release the mouse button when the containers become highlighted.

The containers became highlighted to indicate that Blocks Microworld could share your selection evenly among the containers. Each container received 2 tens. The value of each container's contents is 20; the total value of all containers is 160. Thus, $8 \times 20 = 160$, and $160 \div 8 = 20$.

There aren't enough tens remaining to share evenly among the containers.

- Select the remaining tens. (Draw a box around them, or press "4" and click on one.)
- Click *Unglue*.

Where there were 4 tens there are now 40 ones. We can share 40 ones among 8 containers by putting 5 into each container.

- Select *5 Per Container* in the **Groups** menu.
- Click on any single. Blocks Microworld should select 40 singles.

Your settings now say you want to put 5 blocks into each of 8 containers. Blocks Microworld anticipates that you will need 40 blocks of the kind you clicked.

- Drag the selection to any container. Release the mouse button when the containers become highlighted.

Each container received 5 ones. The display shows that each container now has received a total of 25, and that the total shared among the containers is 200. Thus, $8 \times 25 = 200$, and $200 \div 8 = 25$.

A General Description

When containers are linked and there are x containers:

Set y *Per Container* in the **Groups** menu. Blocks Microworld will attempt to select x times y blocks of the kind you click. This enables you to drag the precise number of blocks required to share y blocks among x containers.

When containers are unlinked and there are x containers:

Set y *Per Container* in the **Groups** menu. Blocks Microworld will attempt to select y blocks of the kind you click. This enables you to drag the precise number of blocks required to put y blocks into a single container.

Rationale

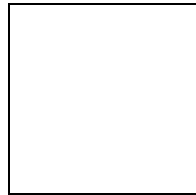
The rationale for linked containers is to provide a concrete model of division as “equitable sharing.” The reason for forcing students to choose an appropriate number of blocks per container is that they must *think ahead* while engaged in the process of sharing. In the process of thinking ahead (“How many of these can I put into each container?”) students will need to think about multiplication. This raises the likelihood that they will come to see multiplication and division as being integrally related—as two sides of a coin, so to speak.

Early versions of Blocks Microworld did almost all the decision-making as to appropriate numbers of blocks to select. This turned out to be counter-productive. Since Blocks Microworld made all the decisions, students didn’t need to think about what they were doing. As a result, they started with a number of blocks, ended with a per-container value, and reasoned not a whit in the interim.

The rationale for unlinked containers is that students sometimes need to see the consequence of *not* imagining equitable sharing. This is a weak rationale, but it has been productive to at least show students what happens if they don’t link containers.

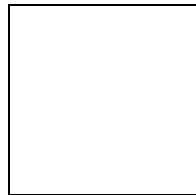
Buttons

Once you have the settings you want, Blocks Microworld's nine buttons are what you and your students will use most in any one task. Most buttons work the same regardless of your menu settings.



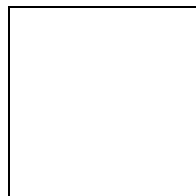
(Blocks Mode)

Select 10, or a multiple of 10, of the same kind of block. Click *Glue* to turn each 10 blocks into one block of the next larger kind.



(Numbers Mode)

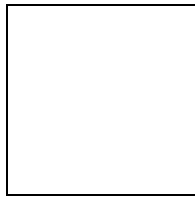
Same as *Glue*, but instead of selecting 10 blocks you click on a digit in the notation display.¹ Blocks Microworld will select 10 blocks represented by that digit, if there are enough, and glue them.



(Blocks Mode)

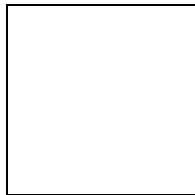
Select one or more blocks of the same kind. Click *Glue* to turn each block into 10 of the next smaller kind.

¹ I use “digit” to mean any number of a place value. So, if you have 12 tens, “12” is a digit.



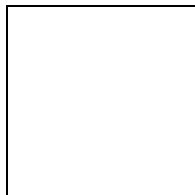
(Numbers Mode)

Same as *Unglue*, but instead of selecting a block you click on a digit in the notation display. Blocks Microworld will select a block represented by that digit and unglue it.



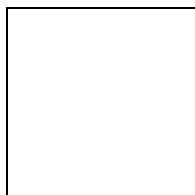
(Active only when a block is in the blocks region.)

Click on *Clean Up* to tidy the screen. Blocks will be organized within blocks regions by size.



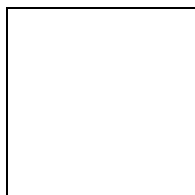
(Active only when the screen is split.)

Unite the two blocks regions to make one. All blocks on the screen are counted as one collection.



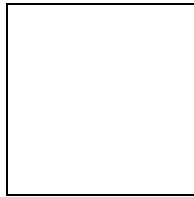
(Active only when the screen is combined.)

Split the current blocks region into two parts. Blocks in different parts of the screen are counted as in two different collections. Each part maintains its own total value.



(Always active)

If in *Numbers Mode*, then switch to *Blocks Mode*. If in *Blocks Mode*, then switch to *Numbers Mode*.



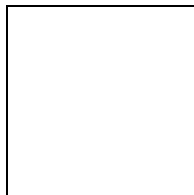
(Active only when a block is in the blocks region.)

Clear all blocks from the blocks region. If containers are showing, then empty all containers.



(Active only when containers are showing and at least one container holds one or more blocks.)

Empty all containers of any blocks they contain.



(Active only when “Zooms Enabled,” in the **Options** menu, is checked.)

When you select *Zooms Enabled* in the **Options** menu, Blocks Microworld allows you to “zoom” windows in and out. Zooming-in allows you to see more than one window on small-screen Macintoshes.

Command Keys

File Menu Command Keys

- ⌘ - N Create a new Blocks World window.
- ⌘ - W Close the current window. (The startup window, *Blocks World I*, cannot be closed.)
- ⌘ - P Print the current window. The window will print in color on an Imagewriter II with a color ribbon.
- ⌘ - Q Quit the program.

Options Menu Command Keys

- ⌘ - U Show/Hide the unit display, normally found

beneath the master blocks. Hide the unit display if it is shown. Otherwise, show it.

- A Show/Hide the **Actions** menu. Show it if it is hidden. Otherwise, hide it. (The **Actions** menu cannot be hidden if containers are showing.)
- D Show/Hide the notation display. Hide the display if it is showing. Otherwise, show it.
- V Show/Hide blocks regions' total values. Hide the total values if they are showing. Otherwise, show them.
- C Show/Hide containers. Show containers if they are hidden. Otherwise, hide them.
- E Turn on/Turn off error messages. When checked, Blocks Microworld displays error messages. When unchecked, Blocks Microworld merely beeps.

Actions Menu Command Keys

- I Increase the unit by a factor of 10; replace every block with a block having a value 10 times as great.
- K Decrease the unit by a factor of 10; replace every block with a block having a value one-tenth as great.

Groups Menu Command Keys

- 1 One block per container:
- 2 Two blocks per container
- 3 Three blocks per container
- 4 Four blocks per container
- 5 Five blocks per container
- 6 Six blocks per container
- 7 Seven blocks per container
- 8 Eight blocks per container
- 9 Nine blocks per container
- 0 Ten blocks per container

Containers Menu Command Keys

- L Link containers

□ - T Show/Hide total value held by containers

Troubleshooting

There are two kinds of “troubleshooting.” One is technical—you have a problem with the program. Another is pedagogical—things aren’t going quite right. This troubleshooting section is for technical problems. Pedagogical troubleshooting is discussed in the instructors’ guide.

Blocks Microworld Won’t Start Under System 7.0

The current implementation of Blocks Microworld will not run under System 7.0 with virtual memory on or 32-bit addressing on. Check your *Memory* control panel. Turn off virtual memory and 32-bit addressing. Then restart your computer.

Blocks Microworld Won’t Start From Another Computer

Blocks Microworld will begin unpredictably if you try to start it across a network (the program is on one computer and you are on a different computer, but have access to the other’s hard drive). I recommend against trying to use Blocks Microworld across a network.

Blocks Microworld Won’t Begin With The Settings I’ve Saved

Make sure the file containing your startup settings is alongside the Blocks Microworld program. It cannot be in a different folder. If you can see the two icons (Blocks Microworld and Blocks Settings) side-by-side, they are in a folder together.

Also, to save your current settings as Blocks Microworld’s default settings, use *Save Settings*. Don’t use *Save Settings As...*

The Screen Is Not Redrawing Properly

Some screen accelerators, especially Apple's 8•24GC, do not like Blocks Microworld. If you have a screen accelerator, turn it off.

Load Settings or Save Settings Won't Work

On an SE/30 (Systems 6.0.7, 6.0.8) under Finder: The menu items *Load Settings*, *Save Settings*, and *Save Settings As* (**File** menu) have been unpredictable. No crashes, just nothing happens.

On an SE/30 (Systems 6.0.7, 6.0.8) under MultiFinder: *Load Settings* and *Save Settings* work, but *Save Settings As* has been unpredictable.

A work-around when none of these items work is to change all settings manually.

A work-around when *Load Settings* and *Save Settings* work, but *Save Settings As* doesn't work, is to save your current settings using the *Save Settings* menu item, then quit Blocks Microworld. You may then duplicate, copy, or rename the file Blocks Settings. Load this file later using the *Load Settings* menu item.

Another work-around when *Save Settings As* doesn't work is to create settings files on a non-SE30 and copy them to your computer.

My Computer Crashes

Whew. This is a tough one—it could crash for lots of different reasons.

Too Many Blocks

One sure-fire crash is to try to make a thousand blocks, and every child at some time wants to make a thousand blocks.

Each block takes up computer memory, so too many will take up too much memory. The number of blocks that will crash

your computer depends on how much memory your computer has. There should not be a problem during typical operations. Let each student crash his or her computer once; it's an initiation rite.

Copy Window Doesn't Work

Shhh, (this is a secret) Blocks Microworld sometimes doesn't get along with System 7.0's clipboard. It seems to crash unpredictably—but (I am thankful) infrequently—when you select *Copy Window* in the **File** menu.

If Copy Window doesn't work properly on your computer, then:

a) Use the Macintosh system to create snapshots of your screen by pressing Command-Shift-3 (⌘-Shift-3).

Under System 7.0 this will create a Teachtext document, named "Picture 1" (or Picture 2, etc.), containing a picture of your screen. Open this Teachtext document; you will be able to copy portions of the picture directly to the clipboard by drawing a rectangle around the part you want. Refer to your Macintosh System 7 Reference, pages 186-187.

Under System 6.0.x ⌘-Shift-3 will create a MacPaint document, named Screen 0 (or Screen 1, etc.), containing a picture of your screen. Open this MacPaint document; you will be able to copy or modify it like any MacPaint document.

b) Use a screen-snapshot utility, like FlashIt or Snapjot.

MacsBug Repeatedly Registers a Crash

Do not use MacsBug while running Blocks Microworld. MacsBug is an intrusive observer of Blocks Microworld's inner workings, and it will sometimes indicate a system crash when, in fact, there was none.

Inexplicable Crashes

For any application there is always a startup program (INIT) with which it cannot coexist. If your computer crashes

frequently, try taking all your INIT's from your system folder. If Blocks Microworld still crashes, I don't know what to say. If it does not crash, then put your INIT's back into your system folder one at a time, restarting your computer each time and testing Blocks Microworld. You've found the guilty INIT when you start having crashes again.

I Can't Select My Printer

On a Mac Plus (System 6.0.7, 6.0.8) with 1 megabyte of memory, there is not enough system memory to open the Chooser. Use the Chooser to select your printer *before* starting Blocks Microworld.

Dialog Boxes Contain Gibberish

On some computers under System 7.0, error messages will contain gibberish. This has happened on a Mac IIsi after a system crash. Shutting the computer down (via Shut Down in the Special menu), and then restarting it (using the keyboard "on" button) seems to fix this problem.

My Computer Runs REALLY Slow

Check your computers RAM cache (in pre-system 7) or disk cache (system 7). In System 6.x.x, this setting is found in the General control panel. In System 7.0.x, it is found in the Memory control panel. If its setting is very large (greater than 2% of available memory), reduce it or turn the cache off.

My Screen Will Not Print

- Make sure your printer is on
- Make sure you've selected your printer in the Chooser (the Chooser is in the apple menu).
- Make sure you've connected your printer cable to the proper port (printer or phone)—the same one indicated in the Chooser's dialog box.
- Make sure your printer cables are well-connected at both ends.

Instructor Guide

- *Educational Impact*
- *Curriculum Integration*
- *Management Tips*
- *Implementation Models*
- *Samples of Student Work*

Educational Impact

Multi-base arithmetic blocks have been popular ever since the 1950's when Zoltan Dienes invented them to support children's meaningful learning of base-ten numeration and numeration-based algorithms. Despite their promise, it became well-known that children often developed what Robert Davis calls "wooden algorithms"—ways to solve arithmetic problems using Dienes' blocks that have few meaningful connections with symbolic methods. Children's understandings of blocks and their understandings of symbolic arithmetic often stood in splendid isolation.

Blocks Microworld began in 1985 as an attempt to overcome children's difficulties in making meaningful connections between their actions with base-ten blocks and their actions with numerical symbols. It has been used in a number of research studies to isolate features of children's engagement that, when supported, can orient them to making important connections between concrete representations of numerical value and traditional symbolic arithmetic. The program was modified repeatedly over the succeeding six years to accommodate findings from this research and to support teachers in addressing these important areas.

In particular, Blocks Microworld is designed so that children can:

- Experience relationships between concrete and symbolic methods;

- Develop images that give meaning to symbolic methods—they can work with symbols while thinking in images.
- Experience the opportunity to express their personal, concrete meanings and methods in a standardized form;
- Experience the unity of a powerful model—using the same model in kindergarten through sixth grade for whole-number and decimal operations.

Curriculum Integration

To integrate Blocks Microworld with ideas of the traditional curriculum is easy. To integrate it with traditional textbooks can be problematic if you expect it to provide direct support for children's memorizing prescribed procedures. It won't help, and the logistics of using it in brief spurts probably won't be worth the effort.

The major requirement Blocks Microworld has made of teachers using it is that many have had to think hard about what they are teaching and how Blocks Microworld supports or doesn't support it. Put simply, if what you have in mind is that students memorize a specific way of writing symbols on paper, then you might experience significant tension between what you want to teach and what Blocks Microworld supports.

On the other hand, if what you have in mind, or come to have in mind, is that students learn to use notation thoughtfully and creatively, that unconventional symbolic methods are acceptable as long as they are valid and meaningful, that what really counts is valid reasoning and creative ways of expressing it, that conventions are merely agreements to do something one way and not another equally valid way, then Blocks Microworld will support you.

The following sections on management tips and implementation models will give specific examples of how Blocks Microworld might be used to support traditional mathematics curricula and how it might be incorporated into your instruction.

Management Tips

Introducing the program

Consider introducing students to the program in much the same way the user guide introduced you to the program. If your students are at computers, have them “mess around” with Blocks Microworld—making blocks, moving them around, glueing and unglueing them, and getting rid of them. If you are using a projection system, you do the same thing while narrating your actions extensively and while inviting comments and questions from your students.

Also note that the user guide moved quickly to a task. This is advisable, for the program’s relevance only becomes clear in the context of tasks. Make the task mildly demanding relative to the conceptual sophistication of your students. With first-graders you might ask, “Can anyone figure out how many tens there are in two hundreds?”¹ With fourth-graders you might ask them to do the task used in the user guide: Put out 3012; now take away 1024. Go to it; do it any way that makes sense.

It is very important that you and your students discuss your introductory activities. Students must realize that they are free to reason idiosyncratically and to express their reasoning in any way that is meaningful to themselves and to others. But this requires an atmosphere that fosters the expression of personal reasoning and personal opinion.

Classroom Demonstrations

If you are lucky enough to have a projection system, then use it extensively. Have students discuss strategies for action and strategies for representing actions in notation. You can control the projection computer, or you might have students take turns controlling it.

¹ You can do this with first graders once they know about unglueing and after they realize that a flat is made of longs.

If you do not have a projection system, then you can still hold whole-class discussions that are focused on Blocks Microworld screens by printing screens and making a transparency of them.

Laboratory Use

Even if you have one computer for every student, it is productive to have students work in pairs or triplets—as long as they each participate in the tasks. Their task-oriented discussions can be quite valuable, as such discussions can reveal to each student possible variations in thinking about a problem and representing a solution.

Also, in laboratories it often happens that students lose sight of a major objective—that they develop methods to represent their actual actions. They will often push blocks around until they “get an answer” without having reflected on their method or its representation. The best way to handle this is simply to ask them to produce a record of their actions. It is quite reasonable to ask third graders or capable second graders to provide an explanation, in prose, of what their record shows. Lab activities should be followed-up with class discussions of students’ strategic and notational methods.

It is imperative that students have a task when they go to a lab. The task may be as simple as “solve these problems using blocks (and these settings).” Or, it may be a “thinking” task, such as “put out blocks that have a value of 0.300.” The latter example is one of my favorites. Some students will put out 3 tenths, some will put out 30 hundredths, and some will put out 300 thousandths. All will probably be bothered by the fact that the computer *never* says “0.300”. Instead, it will display a total value of “0.3”. In the follow-up discussions in class you can bring up the questions, “Are these different solutions or are they the same?” and “Does it matter that the computer never said that you had a total value of ‘0.300’? Why?”

Implementation Models

These examples are offered as prototypes to stimulate your imagination. They are not offered with any prescriptive intent.

Introduction to Program/Whole Number Numeration¹

Context: Teacher has a Macintosh with projector panel; students are in pairs at computers. Each computer is running Blocks Microworld.

Settings: Base 10
Single is 1

Separated screen
Blocks Mode
Show Block Names

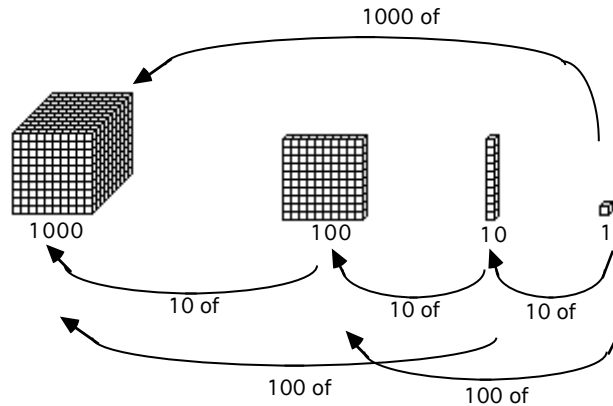
The Mouse

- Show students the mouse. Indicate the mouse pointer. Have them move their mice and watch the mouse pointer.
- Have them put the mouse in each of the four corners of the screen. Then have them put the mouse pointer over the *Clean Up* and *Clear All* buttons.

Making numbers

- Name the blocks. Discuss “ten of” relationship among them. Use projected screen as your “blackboard.” Use mouse as your pointer.

¹ This is taken from a transcript of an actual fourth-grade lesson. Students had never seen Blocks Microworld before, and they had never used a Macintosh before. Much of this could be used, with some modification, at any grade level as an introduction to Blocks Microworld and the basic ideas of numeration.



- Write **4123** on blackboard. Ask students what blocks we would use to show 4123.



Do not ask students how to make 4123 on the computer. Rather, lead them to saying that 4123 would be made of 4 cubes (thousands), 1 flat (hundred), 2 longs (tens), and 3 singles (ones).

- *“This is how we make numbers with the computer.”*
Make 4123 by dragging blocks. Have students mimic your actions on their computers.



Point out the dual representational systems (blocks and notation) as you drag master blocks.

The notation display

This part is for them to understand that the notation display always gives reliable information, while the “blocks” display does not.

- Tell students to put their heads down and cover their eyes; you are going to do “something tricky.”
- Move a flat so that it covers all the singles.
- Ask, “How many longs do I have now?”
“How many singles do I have now?”

Point out the notation display, and the fact that *it* says that you still have 3 singles. Uncover the singles. Encourage students to

pay attention to the notation display for amounts, not just on what they see.

Dragging a block

- Demonstrate how to move a block around.



Say that to move a block, they must put the *tip* (pointy end) of the mouse-pointer on a block.

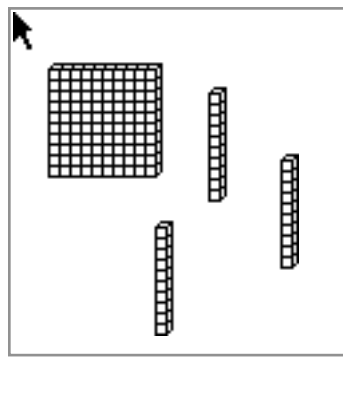
Have students move several blocks from one region to another. Mention that the notation display always correspond to what is on the screen.

Dragging several blocks

- Mention to students that they can move more than one block at a time.

Have them click and drag so that they draw a rectangle around all the blocks in the right region. When they release the mouse button, all the blocks intersecting the interior of the rectangle will be selected (blackened).

Click Here and Hold Mouse Button Down



Drag to here an
release mouse
button.

Or, students can hold down a digit key and Blocks Microworld will automatically select that many blocks of the kind they clicked, if there are enough.

Moving any one of the selected blocks will move all the selected blocks.

- Tell students to move all blocks back to the *right* region.
- Tell students to move all blocks back to the *left* region.
- Tell students to click *Clean Up*.

Unglueing and Numeration

This is an introduction to numeration and representations of numerical value in Blocks Microworld. You lead the discussion with your computer. Students will follow your actions when you tell them to do it themselves.

- Mention blocks' values again; quick review of earlier discussion of relationships among blocks (see figure).
- Set *Value Names* as the Blocks Mode Display (**Options** menu).
- Mention the change in the notation display. Be alert for confusions

Your computer should still show 4 thousands, 1 hundred, 2 tens, and 3 ones.

- Ask again about 1 thousand in relation to hundreds: How many hundreds are in 1 thousand?
- Click on a thousand
- Click *Unglue*
- Let children describe what happened. Make sure the question of whether the total value has changed gets discussed.

After discussing what happened, raise the issue of how you might represent what you now have. If children have no idea, suggest something like this.

- Before: 4123
- After: $\begin{array}{c} 3 \ 11 \\ \cancel{4} \cancel{1} 23 \end{array}$



Be sure that students relate the “before” and “after” of the change. Also, be alert to their speaking about blocks and not about values. You should make it clear that you want them to focus on values. One example: “Started with 4 thousands, 1 hundred, ..., unglued one thousand, now have 3 thousands, 11 hundreds, ...”



It is important that students understand that a collection’s total value is not changed by ungluing a block. Try a story like this:

Suppose we ask a farmer to deliver 4123 apples. He brings the apples in 3 boxes of 1000, 11 boxes of 100, 2 boxes of 10, and 3 loose apples. Did we get the correct number of apples?



The moral of this example is that even though the numeral is “funny looking,” it names the same number as does the “better” looking numeral “4123.”

Repeat the above example several times, each time asking for a representation of the current state in relation to what has gone on before, such as:

- Unglue another thousand.

$$\begin{array}{r} 2 \ 21 \\ 3 \ 11 \\ \hline \end{array}$$
 Represent—such as, ~~4~~123 .

Discuss.

- Unglue a hundred.

$$\begin{array}{r} 20 \\ 2 \ 21 \\ 3 \ 11 \ 12 \\ \hline \end{array}$$
 Represent—such as, ~~4~~1~~2~~3 .

Discuss.



It is important that you have them read the *entire* story from a record, such as “started with 4 thousands, 1 hundred, 2 tens, 3 ones. Unglued a thousand; now have 3 thousands 11 hundreds, 2 tens, and 3 ones (where does the record contain this information?). Unglued another thousand, ... and so on.

- Ask, “Look at this record (the one shown immediately above). We unglued 1 thousand and had 11 hundreds.

Could someone repeat our first step, but unglue 1 of these 11 hundreds as their next step, and still end with this as a record of their actions?"

- Open discussion.

Division

This set of three examples is meant to illustrate how Blocks Microworld can be used in teaching division. The model of division incorporated into Blocks Microworld is division as sharing—sharing some amount evenly among a number of recipients. It is also meant to highlight the influence that base-ten numeration has on the symbolic algorithms we use.

$$2912 \div 8 = ?$$

Context: You are at a computer running Blocks Microworld.

Settings: Load *2912 ÷ 8 Settings* if it is on your disk. Otherwise, use the following settings.

Single is 1	Show Containers
Combined screen	Show 8 Containers
Blocks Mode	Link Containers
Display Value Names	Show Containers' Total

- Goals:
- 1) To answer this question, “If we share blocks having a value of 2912 among 8 containers so that each container gets the same amount, then how much does each container receive?”
 - 2) To establish a method that connects the situation of sharing 2912 ones among 8 containers to a set of principles we can apply more generally.
 - 3) To record the actions we take in answering this question in a way that suggests a general notational method.

The method I will illustrate is:

- a. Start with some amount.

- b. Share what we can, perhaps ungluing some blocks to enable sharing.
- c. Record how what was received by each container.
- d. Record how much was shared in all.
- e. Determine how much remains to be shared; think of this as a new amount-to-be-shared.
- f. If the new amount to be shared is zero, stop. Otherwise, start again at step *a*.

I will also attempt to orient you to having a conversation with yourself. The aim of this self-conversation (soliloquy) is to maintain firm connections between the initial problem, the current situation, the actions you've taken between then and now, and what you might do next.

Put out blocks representing our amount to be shared among the 8 containers.

- Put out 2 thousands, 9 hundreds, 1 ten, and 2 ones.
- Try to put a cube into a container. To put a block into a container, drag the block so that *the pointer* is over the container.

The pointer turns into a “shaking head.” This indicates that Blocks Microworld cannot share one block among 8 containers. It needs a collection of blocks that has 8, 16, 24, or etc. blocks in it.



Ask: *What can I do to share 2 thousands among 8 containers?*



Aha! *Unglue them.*

- Select 2 thousands. (Such as, draw a box around them.)
- Click *Unglue*.

Where I had 2 thousands and 9 hundreds, I now have 29 hundreds. *Now, I can give 3 hundreds to each of the 8 containers.*

- Select *3 Per Container* in the **Groups** menu.

- Click on a hundred. Blocks Microworld will select 24 hundreds.
- Drag any of the selected blocks to any container. Drag the selection so that the *mouse pointer* is over a container.
- Release the mouse button when the containers change color.



Ask: *How much did each container receive?* 3 hundreds, or 300.

Ask: *How much was shared altogether?* 24 hundreds, or 2400.

Ask: *How much remains to be shared?* 2912-2400, or 512.¹



Try to share the hundreds. *But, I can't share 5 hundreds among 8 containers.*



Unglue the hundreds!

- Select 5 hundreds.
- Click *Unglue*.

The 5 hundreds turned into 50 tens. So, where I had 5 hundreds and 1 ten, I now have 51 tens.



There are enough tens to give at least one to each container. *But how many can I give?*



There are 8 containers and 51 tens. If I give 5 tens to each container, that will use up 40 tens. I can give 6 tens to each container—that will use 48 tens.

- Select *6 Per Container* in the **Groups** menu.
- Click on a ten. Blocks Microworld will select 48 tens.
- Drag any ten to a container.
- When the containers change color, release the mouse button.

¹ Notice that this is where we “start over” again. 512 is our new amount-to-be-shared.



Ask: *How much did each container receive?* 6 tens, or 60.

Ask: *How much has each container received in all?* 3 hundreds and 6 tens, or 360.

Ask: *How much was shared in all?* 48 tens, or 480.

Ask: *How much remains to be shared?* 512-480, or 32.¹



Aha! I can see what I'm going to do next! Change the 3 remaining tens to ones, then give each container 4 ones!

(Go ahead and do it.)

Looking Back

Informally

Here is a summary of the steps in your solution. The summary tells a story, but it tells it in a special way. It highlights three things in relation to one another: How much you have shared, how much each container has received, and how much remains to be shared. It also tells about those times you changed a numbers' representation without changing its value (e.g., 5 hundreds changed into 50 tens).

We started with 2 thousands, 9 hundreds, 1 ten, and 2 ones.

You changed 2 thousands to 20 hundreds. You then had 29 hundreds, 1 ten, and 2 ones.

You gave 3 hundreds to each of the 8 containers. In this step you shared a total of 24 hundreds. This left you with 5 hundreds, 1 ten, and 2 ones to share.

You changed 5 hundreds to 50 tens. You then had 51 tens and 2 ones.

You gave 6 tens to each of the 8 containers. In this step you shared a total of 48 tens. This left you with 3 tens and 2 ones to share. Each container has received 3 hundreds and 6 tens.

¹ This is the second time that we “start over” again. 32 is our new amount-to-be-shared.

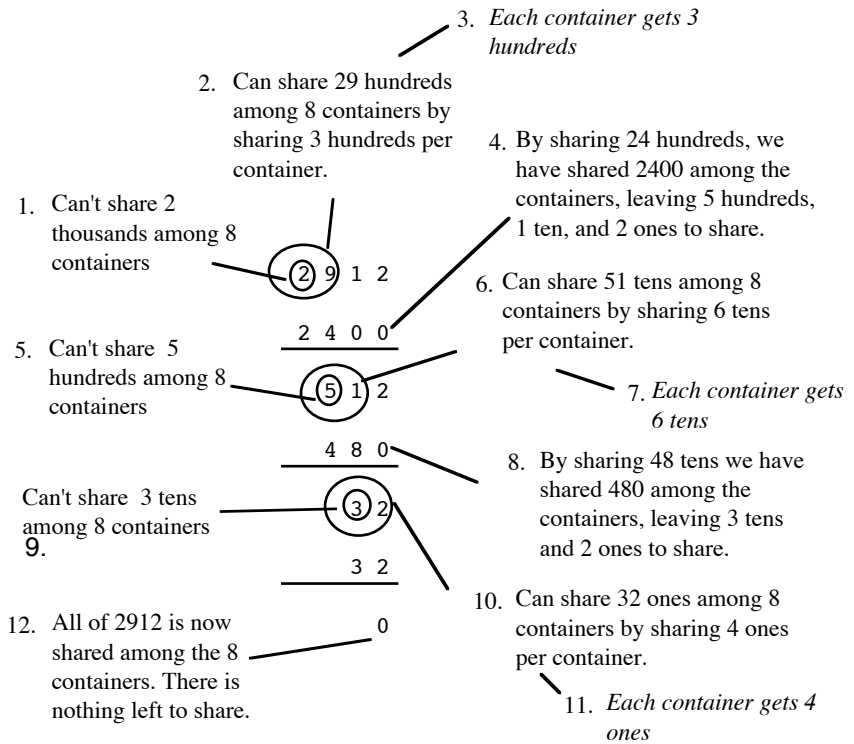
You changed 3 tens to 30 ones. You then had 32 ones.

You gave 4 ones to each of the 8 containers. This left you with nothing left to share. Each container has received 3 hundreds, 6 tens, and 4 ones, or 364.

So, 2 thousands 9 hundreds 1 ten and 2 ones shared among 8 recipients gives 3 hundreds, 6 tens, and 4 ones to each recipient. Put another way, $2912 \div 8 = 364$.

Notationally

Here is another summary of the steps in your solution. The summary tells the same story as was just told, but it tells the story very compactly, using conventions of base-ten numeration. It highlights the same features as the previous story, but you can't read the story from left-to-right.



This notational scheme may look complicated, even if it does reflect your actual actions precisely. However, compare the

above figure to the same diagram, but with the annotation removed and 2 small lines added. Look familiar?¹

$$\begin{array}{r}
 3 \ 6 \ 4 \\
 8 \overline{) \textcircled{2} 9 \ 1 \ 2} \\
 \underline{2 \ 4 \ 0 \ 0} \\
 \textcircled{5} \ 1 \ 2 \\
 \underline{4 \ 8 \ 0} \\
 \textcircled{3} \ 2 \\
 \underline{3 \ 2} \\
 0
 \end{array}$$

2912 ÷ 8 = ? (Unconventionally)

This example repeats the same problem as previously illustrated, but presents one student's unconventional solution and her record of it. The purpose of this example is to illustrate a point made repeatedly in the user guide: As long as a student's method is valid and is based in reasoning, then it should be valued for its insight and creativity.

Another purpose of this example is to show how creative students can be when a teacher makes a sincere effort to convince his or her class that their creativity *is* valued.

- Start with 2912 and 8 containers.
- Unglue 1 ten. This makes 2 thousands, 9 hundreds, and 12 ones.
- Give 1 one to each container.

That leaves 2 thousands, 9 hundreds, and 4 ones remaining to be shared.

$$\begin{array}{r}
 1 \\
 8 \overline{) 2912} \\
 \underline{8} \\
 2904
 \end{array}$$

¹ This should alert you to how packed with meaning symbolic methods tend to be. We need to remind ourselves of this continually when teaching students who do not have all these meanings “automatized,” ready to inject into the notations they see and use.

- Give 1 hundred to each container.
This leaves 2 thousands, 1 hundred, and 4 ones remaining to be shared.

$$\begin{array}{r}
 11 \\
 8 \overline{) 2912} \\
 \underline{8} \\
 2904 \\
 \underline{800} \\
 2104
 \end{array}$$

- Unglue 1 hundred. You now have 2 thousands, 10 tens, and 4 ones to be shared.
- Give 1 ten to each container.
This leaves 2 thousands, 0 hundreds, 2 tens, and 4 ones remaining to be shared.

$$\begin{array}{r}
 111 \\
 8 \overline{) 2912} \\
 \underline{8} \\
 2904 \\
 \underline{800} \\
 2104 \\
 \underline{80} \\
 2024
 \end{array}$$

- Unglue 2 tens. This makes 2 thousands and 24 ones.
- Give 3 ones to each container.
This leaves 2 thousands remaining to be shared.

$$\begin{array}{r}
 3 \\
 111 \\
 8 \overline{) 2912} \\
 \underline{8} \\
 2904 \\
 \underline{800} \\
 2104 \\
 \underline{80} \\
 2024 \\
 \underline{24} \\
 2000
 \end{array}$$

- Unglue 2 thousands. This makes 20 hundreds.
- Give 2 hundreds to each container.
This leaves 4 hundreds remaining to be shared.

$$\begin{array}{r}
 2 \ 3 \\
 111 \\
 \hline
 8 \overline{) 2912} \\
 \underline{8} \\
 2904 \\
 \underline{800} \\
 2104 \\
 \underline{80} \\
 2024 \\
 \underline{24} \\
 2000 \\
 \underline{1600} \\
 400
 \end{array}$$

- Unglue 4 hundreds. This makes 40 tens.
- Give 5 tens to each container.

This leaves a total of 0 remaining to be shared.

Each container has received a total of 3 hundreds, 6 tens, and 4 ones.

$$\begin{array}{r}
 \textcircled{25^3} \\
 111 \\
 \hline
 8 \overline{) 2912} \\
 \underline{8} \\
 2904 \\
 \underline{800} \\
 2104 \\
 \underline{80} \\
 2024 \\
 \underline{24} \\
 2000 \\
 \underline{1600} \\
 400 \\
 \underline{400} \\
 0
 \end{array} = 364$$

Decimal Division: $2 \div 3 = ?$

This example is meant to illustrate two things: the mechanics (and conceptual importance) of the *Increase Unit* and *Decrease Unit* items in the **Actions** menu. It also illustrates how Blocks Microworld can be used to make the idea of “repeating decimals” concrete.

- Show 3 containers.
- Select *A single is 1*.
- Put out 2 singles.

You have two ones. But you can't share 2 ones among 3 containers, nor can you unglue singles.

- Select *Increase Unit* in the **Actions** menu (or press \square - I).

The unit changed from a single to a long; the 2 singles were replaced by 2 longs. Thus, you still have 2 ones, but the blocks representing 1 can be unglued into tenths.

- Unglue 2 ones. You now have 20 tenths.
- Give 3 tenths to each container.

Each container has received a total of 3 tenths; 2 tenths remain to be shared.

- Select *Increase Unit* from the **Actions** menu (or press \square - I).

The unit changed from a long to a flat. The two singles, which stood for 2 tenths, have changed to two longs, which still stands for 2 tenths.

- Unglue 2 tenths. You now have 20 hundredths.
- Give 3 hundredths to each container.

Each container has received a total of 3 tenths and 3 hundredths; 2 hundredths remain to be shared.

- Select *Increase Unit* from the **Actions** menu (or press \square - I).

The unit changed from a flat to a cube. The two singles, which stood for 2 hundredths, have changed to two longs, which still stands for 2 hundredths.

- Unglue 2 hundredths. You now have 20 thousandths.

- Give 3 thousandths to each container.

Each container has received a total of 3 tenths, 3 hundreds, and 3 thousandths; 2 thousandths remain to be shared.

- Select *Increase Unit* from the **Actions** menu (or press \square - I).

The unit changed from a cube being 1 to a cube being $1/10$. The two singles, which stood for 2 thousandths, have changed to two longs, which still stands for 2 thousandths.

- Unglue 2 thousandths. You now have 20 ten-thousandths.
- Give 3 ten-thousandths to each container.

Each container has received a total of 3 tenths, 3 hundreds, 3 thousandths, and 3 ten-thousandths; 2 ten-thousandths remain to be shared.

In principle, this set of actions can repeat indefinitely. We will always come to the point where we have 2 singles to share; we increase the unit, break up two longs to 20 singles, give 3 singles to each container, and we are again left with 2 singles to share. But each time, the 2 singles are worth $1/10$ th of what they were worth before.

Articles

<<Do you really want to include my articles – for elementary school teachers? My articles tend to be long and (people tell me) dense.>>

Appendix

- *Index*
- *Glossary*
- *Bibliography*
- *Blackline Masters/Templates*

Index

<<*I'm assuming you will do this.*>>

Glossary

<<*Fill in later.*>>

Bibliography

The design of Blocks Microworld is informed by Jean Piaget's genetic epistemology, by Les Steffe's theory of number development, Ernst von Glasersfeld's writings on constructivism, and my own research. Here is a brief reference list.

Cobb, P., & von Glasersfeld, E. (1983). Piaget's scheme and constructivism. *Genetic Epistemology*, 13(2), 9-15.

Dreyfus, T., & Thompson, P. W. (1985). Microworlds and Van Hiele levels. In *Proceedings of the Ninth Annual Meeting of the International Group for the Psychology of Mathematics Education*.

Piaget, J. (1950). *The psychology of intelligence*. London: Routledge & Kegan-Paul.

Piaget, J. (1965). *The child's conception of number*. New York: W. W. Norton.

Piaget, J. (1971). *Genetic epistemology*. New York: W. W. Norton.

Piaget, J. (1980). *Adaptation and intelligence*. Chicago: University of Chicago Press.

- Piaget, J. (1980). Schemes of action and language learning. In M. Piatelli-Palmarini (Ed.), *Language and learning: The debate between Jean Piaget and Noam Chomsky* (pp. 164-167). Cambridge, MA: Harvard University Press.
- Piaget, J., & Garcia, R. (1989). *Psychogenesis and the history of science* (H. Feider, Trans.). New York: Columbia University Press.
- Steffe, L., Thompson, P., & Richards, J. (1982). Children's counting in arithmetical problem solving. In T. Carpenter, T. Romberg, & J. Moser (Eds.), *Addition and subtraction: A cognitive perspective* (pp. 83-97). Hillsdale, NJ: Erlbaum.
- Steffe, L. P. (1991, April). *Composite units and their schemes*. Paper presented at the Annual Meeting of the American Educational Research Association, Chicago, IL.
- Steffe, L. P., Cobb, P., & von Glasersfeld, E. (1988). *Construction of arithmetic meanings and strategies*. New York: Springer-Verlag.
- Steffe, L. P., von Glasersfeld, E., Richards, J., & Cobb, P. (1983). *Children's counting types: Philosophy, theory, and application*. New York: Praeger Scientific.
- Thompson, P., & Thompson, A. (1990). Salient aspects of experience with concrete manipulatives. In *Proceedings of the 14th Annual Meeting of the International Group for the Psychology of Mathematics* (pp. 46-52). Mexico City.
- Thompson, P. W. (1982). *A theoretical framework for understanding young children's concepts of whole-number numeration*. Unpublished Doctoral dissertation, Department of Mathematics Education, University of Georgia.
- Thompson, P. W. (1984). Microworld environments for teaching mathematics to future elementary school teachers. In *Proceedings of the De Anza Conference on Computers in Higher Education* Vol. 1 (pp. 5-11). .
- Thompson, P. W. (1985). Computers in research on mathematical problem solving. In E. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 417-436). Hillsdale, NJ: Erlbaum.

- Thompson, P. W. (1985). Experience, problem solving, and learning mathematics: Considerations in developing mathematics curricula. In E. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 189-243). Hillsdale, NJ: Erlbaum.
- Thompson, P. W. (1985). A Piagetian approach to transformation geometry via microworlds. *Mathematics Teacher*, 78(6), 465-472.
- Thompson, P. W. (1987, February). "Direct Engagement" software for teaching mathematics. Paper presented at the Computers in Education VII, Chicago, IL.
- Thompson, P. W. (1987). Mathematical microworlds and intelligent computer-assisted instruction. In G. Kearsley (Ed.), *Artificial Intelligence and Education* (pp. 83-109). New York: Addison-Wesley.
- Thompson, P. W. (1989, July). *Mathematics software*. Paper presented at the Apple Computer Higher Education Conference on "Designing for Learning", Cupertino, CA.
- Thompson, P. W. (1991). To experience is to conceptualize: Discussions of epistemology and experience. In L. P. Steffe (Ed.), *Epistemological foundations of mathematical experience* (pp. 260-281). New York: Springer-Verlag.
- Thompson, P. W. (1991, April). *Representations, principles, and constraints: Contributions to the effective use of concrete manipulatives*. Paper presented at the Annual Meeting of the American Educational Research Association, Chicago, IL.
- Thompson, P. W. (1992, February). *Constructivism, cybernetics, and information processing: Implications for research on mathematical learning*. Paper presented at the International Symposium on Alternative Epistemologies in Education, Athens, GA.
- Thompson, P. W. (1992, August). *Imagery and the development of mathematical reasoning*. Paper presented at the Seventh International Congress on Mathematical Education, Quebec City, Canada.
- Thompson, P. W. (1992). Notations, conventions, and constraints: Contributions to effective uses of concrete materials in elementary mathematics. *Journal for Research in Mathematics Education*, 23(2), 123-147.

- Thompson, P. W., & Thompson, A. G. (1987). Computer presentations of structure in algebra. In N. Herscovics & C. Kieran (Ed.), *Proceedings of the 11th Annual Meeting of International Group for Psychology of Mathematics Education* Vol. 1 (pp. 248-254). Montréal: University of Quebec, Montréal.
- von Glasersfeld, E. (1976). Cybernetics and cognitive development. *American Society of Cybernetics Forum*, 8(3 & 4), 115-120.
- von Glasersfeld, E. (1978). Radical constructivism and Piaget's concept of knowledge. In F. B. Murray (Ed.), *Impact of Piagetian Theory* (pp. 109-122). Baltimore: University Park Press.
- von Glasersfeld, E. (1981). An attentional model for the conceptual construction of units and number. *Journal for Research in Mathematics Education*, 12(2), 83-94.
- von Glasersfeld, E. (1984). An introduction to radical constructivism. In P. Watzlawick (Ed.), *The invented reality* (pp. 17-40). New York: W. W. Norton.
- von Glasersfeld, E. (1987). Learning as a constructive activity. In C. Janvier (Ed.), *Problems of representation in teaching and learning mathematics* (pp. 3-18). Hillsdale, NJ: Erlbaum.
- von Glasersfeld, E. (1988). Reluctance to change a way of thinking. *The Irish Journal of Psychology*, 8(1), 83-90.
- von Glasersfeld, E. (1990). Aspects of constructivism: Vico, Berkeley, Piaget. In *Proceedings of the Conference on Evolution and Cognition--The Heritage of Jean Piaget's Genetic Epistemology*. Bergamo, Italy.
- von Glasersfeld, E. (1991). Abstraction, re-presentation, and reflection: An interpretation of experience and Piaget's approach. In L. P. Steffe (Ed.), *Epistemological foundations of mathematical experience* (pp. 45-65). New York: Springer-Verlag.
- von Glasersfeld, E. (1991). *A constructivist's view of learning and teaching*. Paper presented at the Research in Physics Learning International Workshop, Bremen, Federal Republic of Germany.

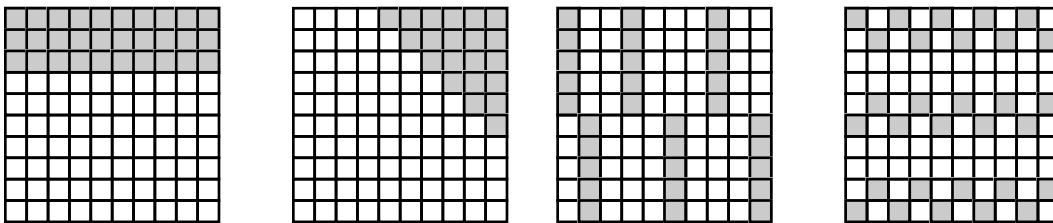
Blackline Masters and Transparency Templates

Mary: Next File

Sample Discussion Items and Test Items (Decimal Numbers)

These test items were used in with fourth-graders during and after a unit on decimal numbers and operations on decimal numbers. Please use these items as prototypes for constructing your own discussion items and test items.

3. Put a circle around each diagram that has $\frac{3}{10}$ of its area shaded.



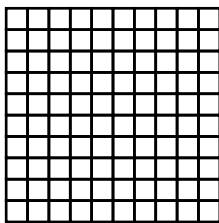
4. Each line has three numbers. Circle the largest number in each line.

(a) $\frac{15}{75}$.123 .0947

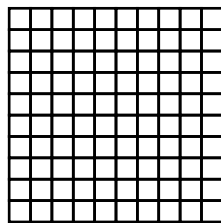
(b) $3\frac{1}{4}$ 3.7198 3.72

(c) 8.35 .972 $8\frac{1}{2}$

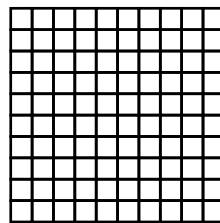
5. The grids shown below are each 10 squares by 10 squares. Shade each grid so that the shaded part goes with the fraction below the grid.



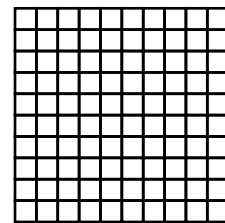
$\frac{3}{10}$



$\frac{2}{5}$



$\frac{5}{20}$



$\frac{41}{100}$

6. Calculate the answers to each of these arithmetic problems:

$$3004 - 286 = \underline{\hspace{2cm}}$$

$$10.02 - 1.034 = \underline{\hspace{2cm}}$$

$$7914 + 2648 = \underline{\hspace{2cm}}$$

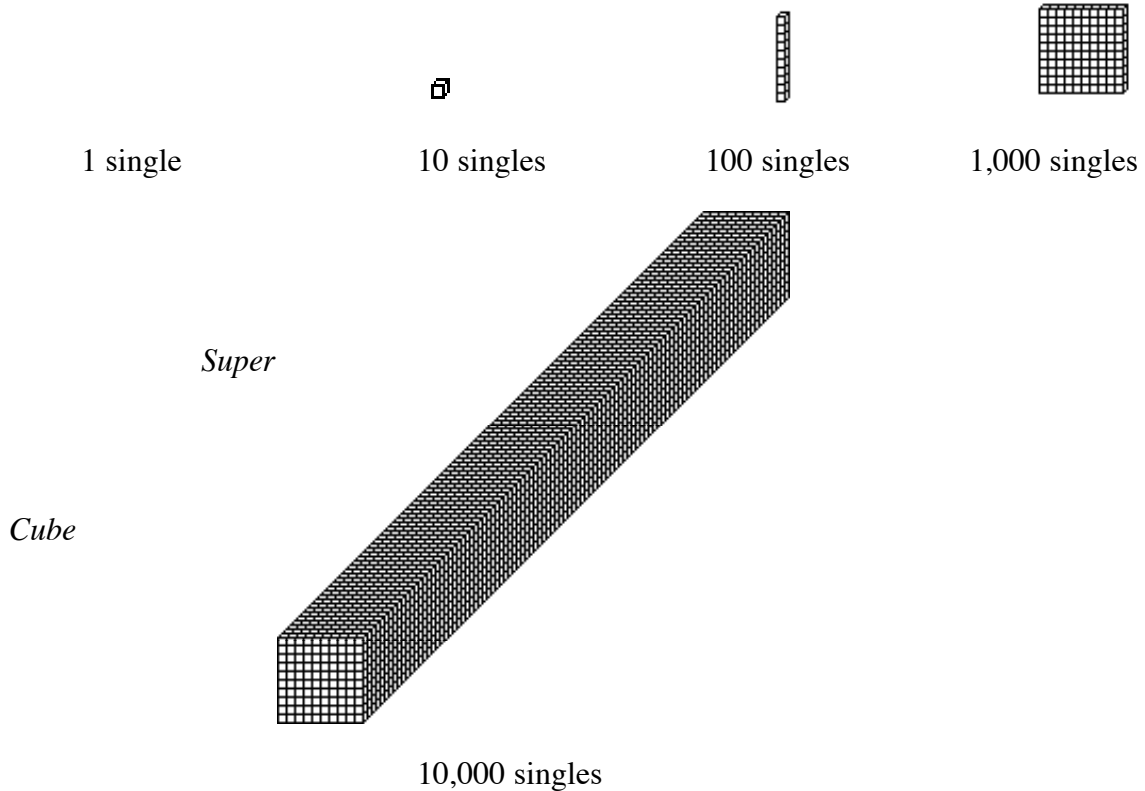
$$.642 + .89 = \underline{\hspace{2cm}}$$

$$5002 - 493 = \underline{\hspace{2cm}}$$

$$1.020 - .35 = \underline{\hspace{2cm}}$$

7. The pictures shown below are of blocks. Larger blocks are made of “single” blocks glued together.

Underneath each block is a number. The numbers show how many “singles” are inside the blocks.



Marianne did not like to count singles. She liked to count the cubes (1,000 singles). In the way Marianne thinks of the blocks, a cube stands for the number 1.

Pretend that you are Marianne, and that the cube stands for the number 1.

- a. Since a cube stands for 1, what does a super cube stand for?

Answer: _____

(Question 7 is continued on next page)

(Question 7, continued from previous page)

- b. Since a cube stands for 1, what does a long stand for?

Answer: _____

- c. Since a cube stands for 1, what does a single stand for?

Answer: _____

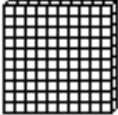
- d. Since a cube stands for 1, what does a flat stand for?

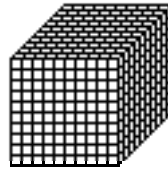
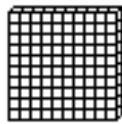
Answer: _____

8. Tom and Sally each gave a value to a block. Their friend, James, put out some blocks. Tom said that Jame's blocks made 4.025. Sally said Jame's blocks made 402.5.

a. Who required more singles to make 1 (one)? Tom Sally (*Circle one*)

Explain your answer:

- b. If Sally's unit was a , then what would Tom say is $\frac{1}{10}$? (*Circle your choice.*)



Explain your answer:

9.a Do this subtraction problem:

$$\begin{array}{r} 2.000000001 \\ - 1.97989 \\ \hline \end{array}$$

.b Do this subtraction problem:

$$\begin{array}{r} 2.001 \\ - 1.97989 \\ \hline \end{array}$$

10. Circle "Same" if an item describes a number that is the same as **4.325**. Circle "Different" if an item describes a number that is different from **4.325**. Circle "Can't Tell" if you cannot tell if the number is the same or different.

$4 + \frac{32}{100} + \frac{5}{1000}$ Same Different Can't Tell

Four ones, three tens, two hundreds and five thousands Same Different Can't Tell

$4 \frac{325}{1000}$ Same Different Can't Tell

Four ones, three tenths, two hundredths, and five thousandths Same Different Can't Tell

$3.\overset{1}{3}1\overset{1}{5}$ Same Different Can't Tell

11. Calculate the answers to each of these arithmetic problems:

$$\begin{array}{r} 12.27 \\ + 5.84 \\ \hline \end{array}$$

$$\begin{array}{r} 8.03 \\ - 2.9 \\ \hline \end{array}$$

$$7.31 - 6.4 = \underline{\hspace{2cm}}$$

$$14.8 + 7.23 = \underline{\hspace{2cm}}$$

13. Tom and Sally each solved $45.32 - 32.37$. Tom's work is shown on the left. Sally's work is shown on the right.

Put a circle around whomever's work is correct.

Put an **X** through whomever's work is incorrect.

Tom

$$\begin{array}{r}
 45.32 \\
 -32.37 \\
 \hline
 13.05 \\
 \begin{array}{r}
 2 \\
 9
 \end{array}
 \end{array}$$

Sally

$$\begin{array}{r}
 14 12 \\
 3 15 13 12 \\
 -45.32 \\
 -32.37 \\
 \hline
 12.95 \\
 12
 \end{array}$$

- b. Tom and Sally actually solved $45.32 - 32.37$ using blocks. Their work, as shown above, was done as they recorded their steps while using blocks.

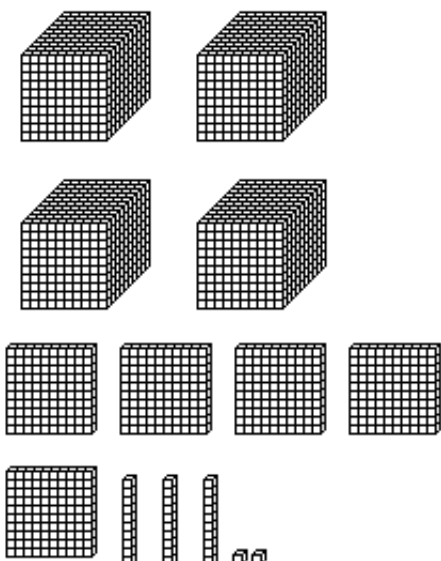
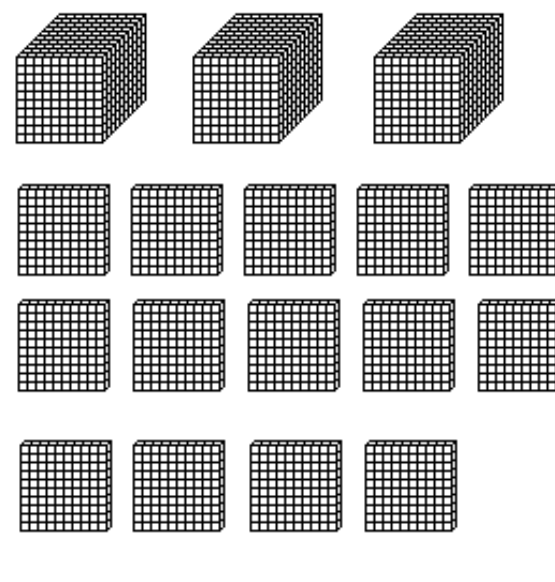
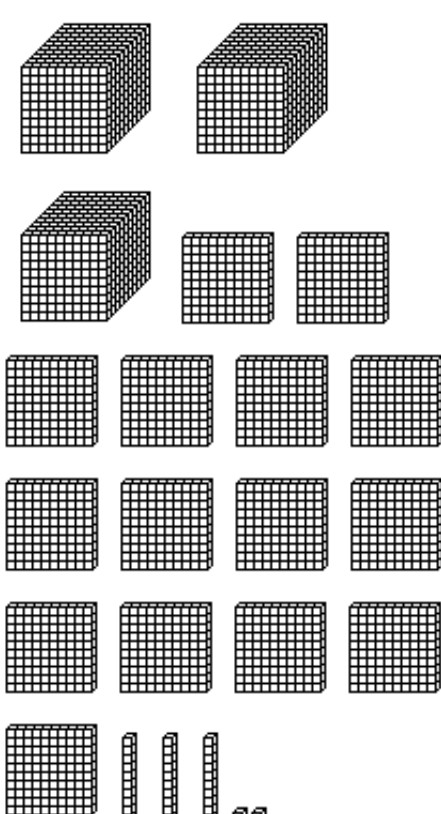
The blocks on the next page show the first four steps in Tom's or Sally's solution.

With whose solution do these steps belong?

Tom's Sally's (*Circle one*)

Explain your decision.

Sample Discussion and
Test Items
(Decimal Numbers)

<p>Step 1</p> 	<p>Step 3</p> 
<p>Step 2</p> 	<p>Step 4</p> 