The Programmable Brain NeuronZoo V0.9

Manual

20170105 Roelh

Introduction

For a long time, people have sought for similarity between the operation of a human brain and a computer. But it is hard to imagine that our brain contains an ALU-like tissue that performs arithmetic or logic operations on rows of bits. It is also difficult to imagine that our brain cells understand hundreds of instructions, like a microprocessor. Registers are easier, they can just represent the 7-or-so things that are in our short-term memory.

But if we remove the ALU and have only three instructions, interpret the register contents as pointers to objects, and imagine that each brain cell is an object, with connections to other brain cells...

We can build a brain model with nice properties:

- have many cells, with connections between them,
- connections between cells can be added or removed (by the user)
- The user can assign a name or meaning to a cell
- cell networks can represent a model of any kind of real-world system

"Thinking" properties:

- there is a short-term memory of 7 items
- cell connections can represent data but can also represent instructions
- cells can execute instructions that change short-term memory, change connections between cells, and check if a connection is present
- these three simple instructions could be used as building blocks for an interpreter that handles more complex instructions
- cell networks can implement any kind of "reasoning"

It is not your typical computer system:

- no built-in knowledge of numbers (not even binary)
- no logic operations like AND, OR
- no separation between "cpu" and memory
- no "data word width" concept

There are examples provided (in the File menu) that show:

- How to program a simple calculator that adds numbers (although the cell instructions can not add)

- That a chess move generator can be programmed in less than 40 cells

Chapter 1: Overview

With NeuronZoo you can build your own cell structures, and program the cells to function like a computer.

The cells understand 3 types of instructions, and can serve as input (when you click on a cell) or as output (by showing a "mental image" inside itself).

The cells can hold information in the way they are connected, and in neurotransmitters that they can contain.

It is available on every PC with an Internet connection. The programmable brain will work in your web browser (Chrome, Firefox), there is no need to download anything. It will run on Windows, on Linux and on Apple (single mouse button not yet supported) .

The brain is presented in 3-D with real-time rendering. All cells, the neurotransmitters within them, and their connections are in real-time simulated and visible on the screen. Because of the 3-D rendering, a reasonable speedy PC or laptop is needed to get acceptable performance.

More details

The brain is built from cells (that might be regarded as neurons). Cells can have connections between them, that are called axons.

The system of cells and axons let us create what mathematicians call a directed graph. This is a way to represent structured information. The cells and the axons are also the long-term memory of the brain.

Within the cells, there can be particular substances called Neurotransmitters. In this brain, they are the short -term memory.

Instruction cells can manipulate the Neurotransmitters and the axon connections , providing the "thinking" of the brain. There are only three instruction types. The execution speed can be controlled and single-stepping is also possible.

The cell configurations that you make can be saved on your own PC.

In the file menu, several examples are available.

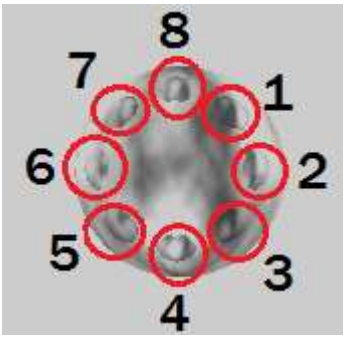
Navigation

The scroll button is used for zooming in and out. Holding the left mouse button down and moving the mouse, changes the viewing angle. Holding the right mouse button down and moving the mouse, shifts the point of view .

Disclaimer: Although this document uses names that are borrowed from neuroscience, the working of this programmable brain is totally different from that of a real brain.

Chapter 2: Cell Structure

A single cell looks like this:

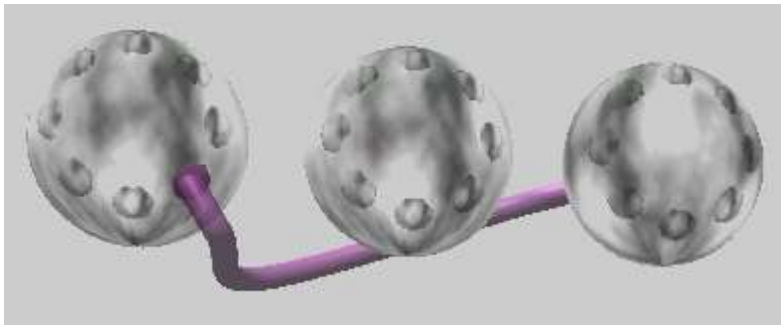


A cell has 8 connection points at its front side. The connection points are numbered in a clockwise way.

At each connection point, an outgoing connection (Axon) can be present. This connects the cell to another cell (at the back side), where it is an incoming connection.

Each cell has a maximum of 8 outgoing connections (one for each connection point). The number of incoming connections is not limited, and for incoming connections, no connection point is needed.

The following example shows an outgoing Axon from connection 3 of the 1st cell (at the front side), going to the 3rd cell (at the backside).

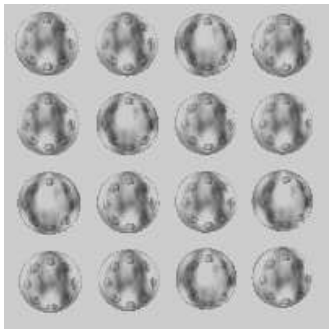


(In most cases the axon is transparent. Here, it has been selected to make it more visible.)

When the mouse goes over the axon, it is highlighted, and a circle appears around the destination cell.

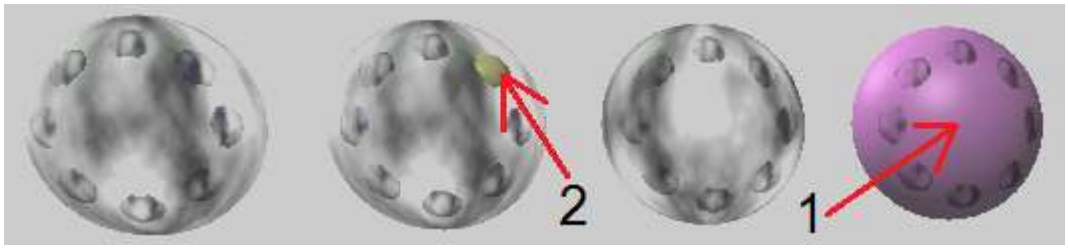
Chapter 3: Manual cell creation and connection

At the right side of the screen you will find a menu. Click on "New cells" to open the sub-menu:

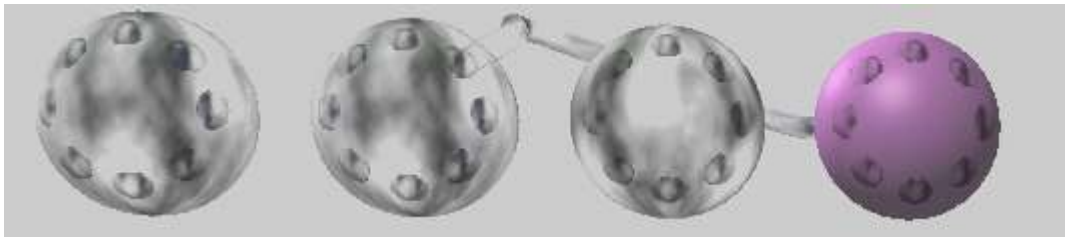


If you click "createGroup", you will get a group of 16 new cells in a 4 by 4 configuration. For other arrangements of cells, you can adjust the blue "cellsX" and "cellsY" sliders.

Now let's make a connection (Axon) between cells. 1) First right-click the destination cell. It will get another color to indicate that it is selected. 2) Then right-click the connection point where the Axon should start.



The result is the (transparent) Axon:



For removing the Axon, right-click on it to select it (it gets another color). Then, in the menu, click "ThisAxon" and then "remove". This will only be needed to correct mistakes.

Again right-clicking a selected item will de-select it again.

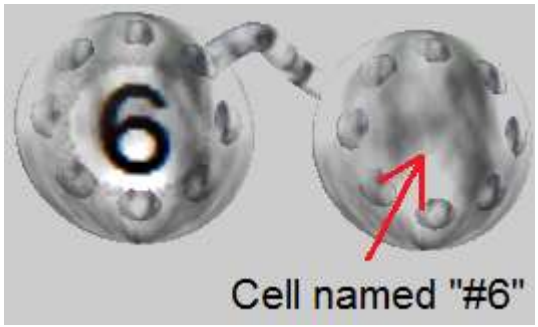
Chapter 4: Mental image

The mental image capability, as described in this chapter, is not needed for the "thinking" of the cell system. It is only for the convenience of us, the observers of the cell activity. It is the main method of providing output to the user.

Every cell can have a name attached to it. Just select the cell and put the name in the "cellName" box (example for a cell called John):

In the current version, the name of the cell is not used, except for one important case. If the name starts with "#", it is a reference to an image. Another cell will display this image inside itself, if it has an **axon at its connection 1**, that connects to the cell that has this special name.

Example: The cell at the right has name #6. The left cell displays the "6" because it has a axon at connection 1 to the cell named "#6".



These images are fixed, so it is not (yet) possible to create your own images. The available images are at this moment:

#0 .. #9	Digits 0 .. 9
#=, #+, #-, #*, #/	Calculator keys: equals sign, plus, minus, multiply, divide
#dr, #dg, #db, #dy	Colored discs: disc red, disc green, disc blue, disc yellow
#wp, #wn, #wb, #wr, #wq, #wk	Chess pieces: white pawn, knight, bishop, rook, queen, king
#bp, #bn, #bb, #br, #bq, #bk	Chess pieces: black pawn, knight, bishop, rook, queen, king
#a1..#a8	For instruction cells: connection or axon numbers
#ntw, #ntr, #ntg, #ntb, #nty, #nto, #ntp, #ntc	For instruction cells: cells representing "Neurotransmitters"

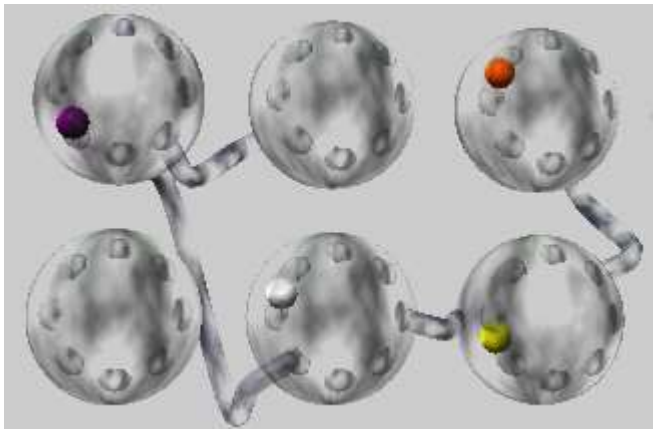
Note that the image can also be displayed in the named cell itself, by having an axon at its connection 1, that connects to itself.

Chapter 5: Neurotransmitters

Within the cells, neurotransmitters can be present. The cell can receive them from an axon of another cell that is connected to it. A cell can also destroy them. Neurotransmitters can be manipulated by executing an instruction (it can also be placed manually in a cell, by "placeNT" in the cell menu).

If a neurotransmitter is present in a certain cell, it is shown as a moving colored particle within the cell.

There are several neurotransmitters. They have a difficult name, so for simplicity we only use the first character of the name. Fortunately, by coincidence, the first character of the name is also the first character of its color. The following screenshot shows cells with a purple, orange, white and yellow neurotransmitter:



	Neurotransmitter list	**Special function**
W	White	Indicates a clicked cell
R	Red	
G	Green	
B	Blue	
Y	Yellow	
O	Orange	
P	Purple	Activates instruction execution
C	Cyan	

Two of these neurotransmitters have a special function:

- The Purple neurotransmitter causes the cell that contains it to execute the instruction of this cell.

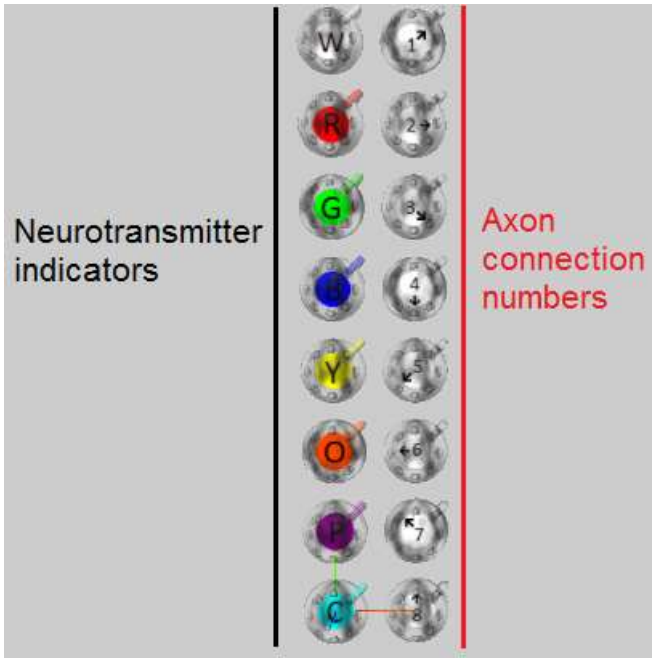
- The White neurotransmitter can be used to trace what cell was left-clicked to start an instruction flow (only if the clicked cell has an axon at connection 5 that connects to a valid instruction).

Chapter 6: Instruction principles

A cell that represents an instruction needs the following:

1. Connections that represent the neurotransmitters and axon connection that are involved in the instruction (see below)
2. A connection to the next instruction. At connection 3 there must be an axon leading to the next instruction. This next instruction will be executed after the current one. If there is no next instruction, execution will stop.
3. A method to activate the first instruction. If you left-click a certain cell, and if this cell has at connection 5 an axon leading to an instruction, this instruction will be executed. This action will also put the white neurotransmitter in the clicked cell.

Most instructions need connections to two Neurotransmitter indicators and a single Axon connection number. These are also cells, and appear as follows:



All eight neurotransmitters and all eight connection numbers are available.

Adding Comments

For your convenience, a comment can be added to each cell. For instruction cells, this can be a short description of the action that the instruction performs.

When the mouse is above a cell that has a comment, this comment will be displayed in the menu (as the last item). If the comment is long, it will be split in up to three menu lines.

It can be entered in the "ThisCell" menu (max approx 120 chars) . Press ENTER when you are done (otherwise this field in the menu will not be updated if another cell is selected afterwards). *Do not use too much capitals in long comments, because the line break system is tuned to the width of lowercase characters.*

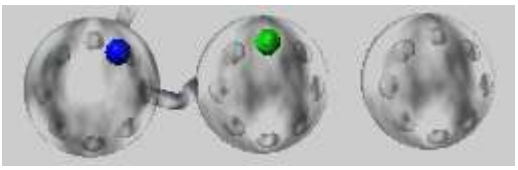
Chapter 7: Instruction NT

The first instruction that we introduce is the NT (Neurotransmitter) instruction. It can create a neurotransmitter (at the startpoint of an axon) and move it through the axon into a certain cell. If another cell has a neurotransmitter with the same name (and color), this is destroyed.

The following screenshots show an example of what the instruction does:

Before the instruction:

After execution of the instruction:

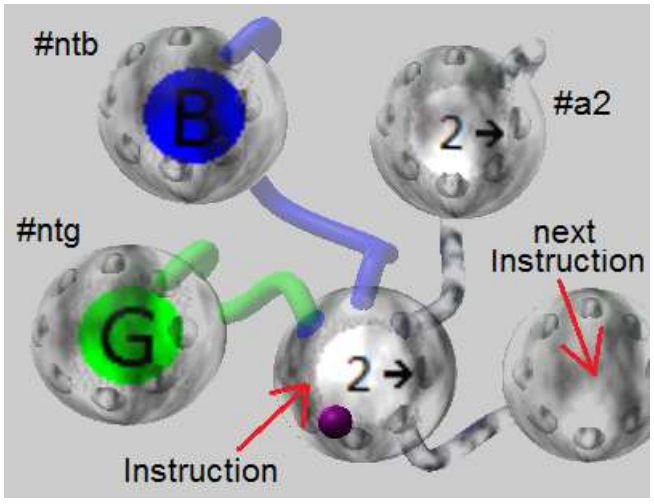


In this example, the instruction does the following:

1. Locate the cell with the blue neurotransmitter.
2. Locate the axon at connection 2 of this cell.
3. Create a green neurotransmitter within the axon, put it in the cell that is at the destination of the axon, and destroy the green neurotransmitter in other cells.

Note, that the blue and green particles seem to be at a fixed position in the cell here. But in reality they have no fixed position, they are moving in circles within the cell that contains them.

The next picture shows the instruction cell that belongs to this example:

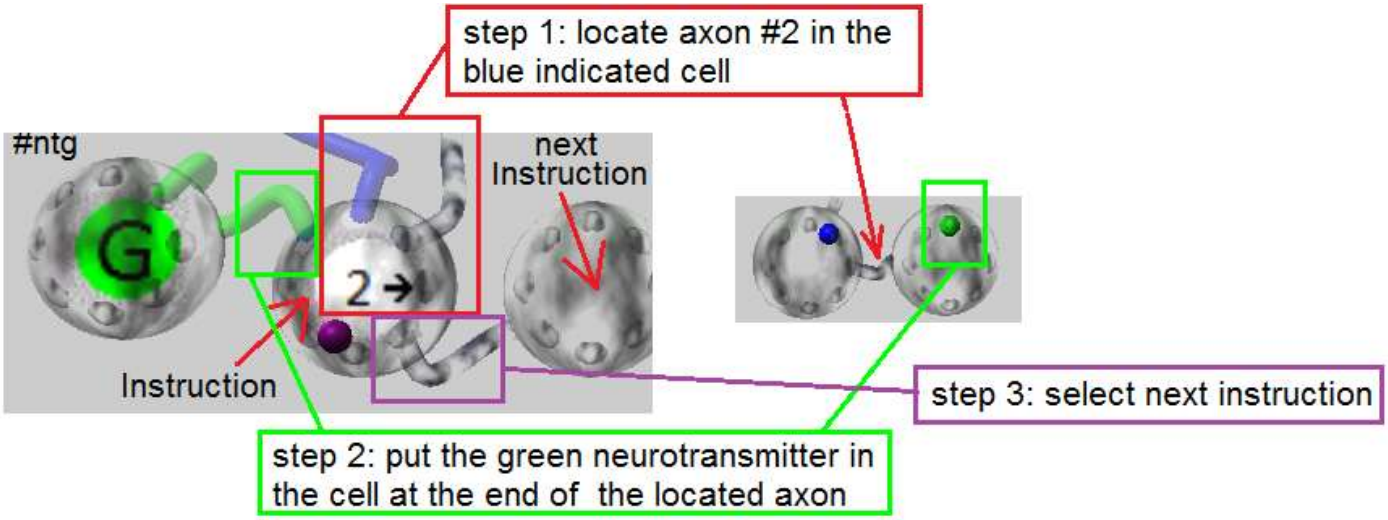


Note that the cells #ntb, #ntg and #a2 are already on the screen, but are not as close to the instruction as shown here. Also note that these cells display their own image because they each have an axon at their connection 1, that connects to themselves.

In order to help the observer to see what the instruction does (without having to follow all connections), the axons to #ntb and #ntg will get the color of the associated neurotransmitter, and the image of connection #a2 is displayed within the instruction cell (because #a2 is at connection 1 of the instruction).

Another diagram might explain the NT instruction better. The instruction does 3 steps:

1. Locate a certain axon (here: #2) in a cell that contains a certain neurotransmitter (here: blue).
2. Create another neurotransmitter (here: green) at the start of the axon, transfer it through the axon and put it in the cell at the end of the axon, and destroy the green neurotransmitter in other cells.
3. Select the next instruction (it is found by following the axon at connection #3 of the current instruction).

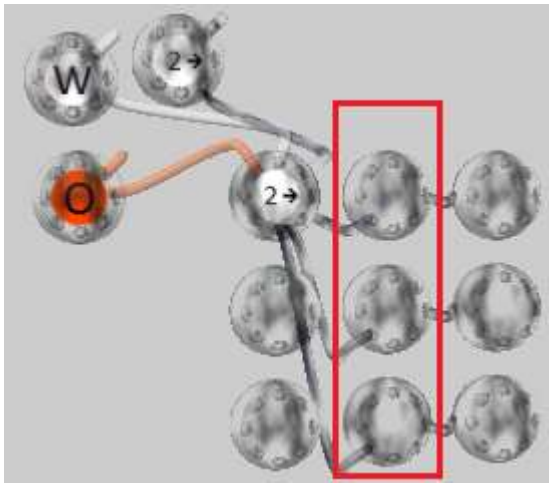


Of course, the instruction can operate on other neurotransmitter colors and other axon connection numbers.

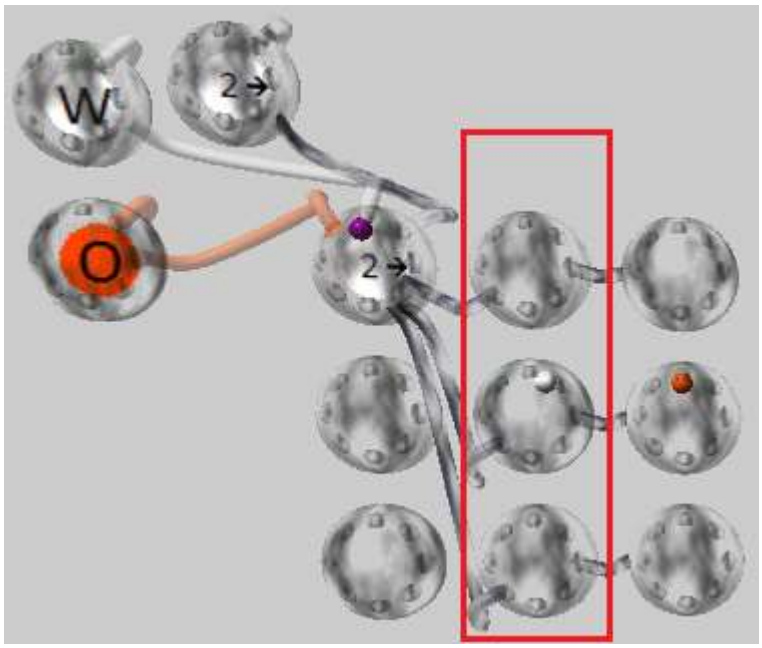
The instruction can also be used in some special ways, that will be discussed later.

EXAMPLE for the NT instruction.

The following example shows a simple 1-instruction program with 3 input cells. The input cells are indicated with the red rectangle.



After left-clicking the middle of the input cells, you see the following:



What do we see:

- The input cells have an axon at connection 5, leading to the instruction
- The white neurotransmitter was placed in the clicked cell
- The instruction was activated
- The instruction found axon #2 at the cell with the white neurotransmitter
- The instruction placed the orange neurotransmitter in the cell to which the axon #2 connects
- The instruction has no next instruction (hard to see), so execution stops, leaving the purple neurotransmitter in the last executed instruction (that is, in this case, also the first one).

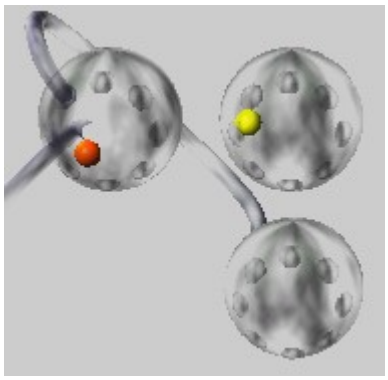
This can be executed as often as we want. Every time we click one of the input cells, the orange neurotransmitter will be placed in the cell at connection #2 of the clicked cell. We could call it "Radio buttons".

This also shows that an instruction cell can be activated by several sources (three "buttons"in this case).

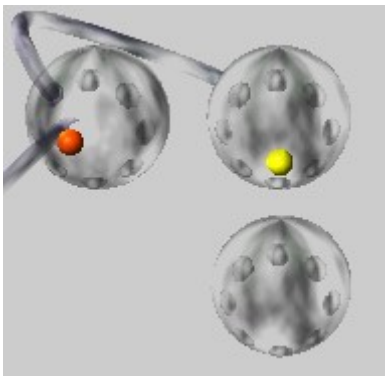
Chapter 8: Instruction AX

The second instruction that we introduce is the AX (Axon) instruction. It can create an axon from one cell to another. The following screenshots show an example of what the instruction does:

Before the instruction:



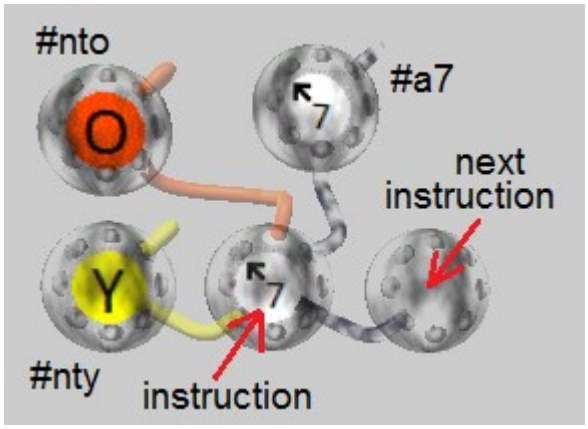
After the instruction:



In this example, the instruction does the following:

1. Locate the cell with the orange neurotransmitter.
2. If there is an axon at connection 7 of this cell (it is), remove that axon
3. Create a new axon, from connection 7 of the located cell, to the cell with the yellow neurotransmitter
4. At the end of the instruction, the next instruction is selected

The next picture shows the instruction that belongs to the example:

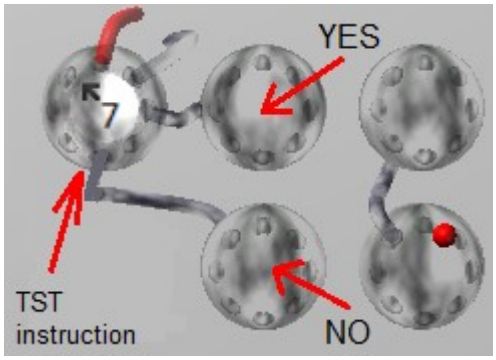


The AX instruction looks like the NT instruction, but the AX instructions uses other axon connections than the NT instruction (see instruction overview in one of the following pages).

Chapter 9: Instruction TST

The third, and also last instruction, enables the program to branch to different sections.

This is an example:



The TST instruction needs a neurotransmitter and connection number (here these are RED and 7).

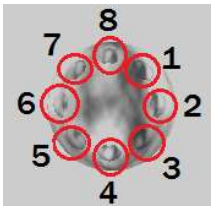
The instruction checks if there is an axon at the indicated position.

In more detail:

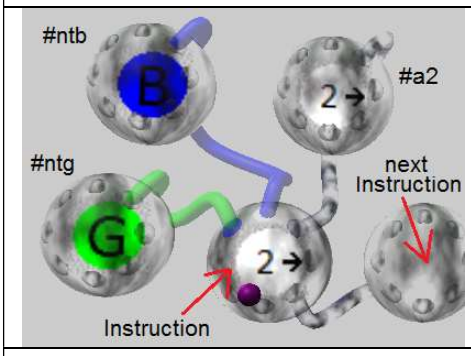
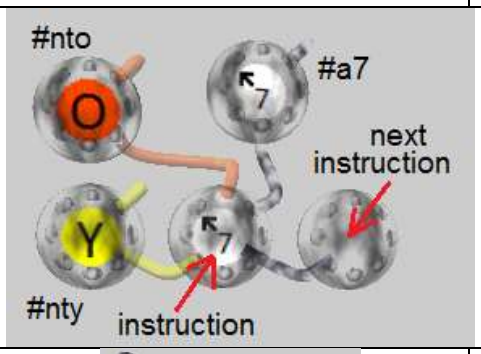
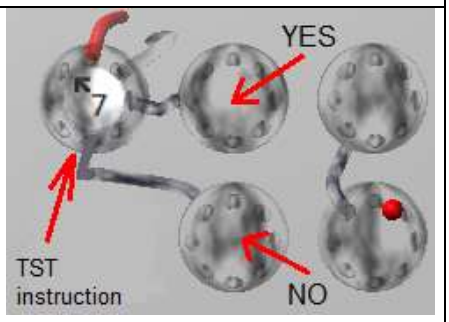
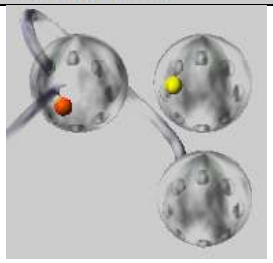
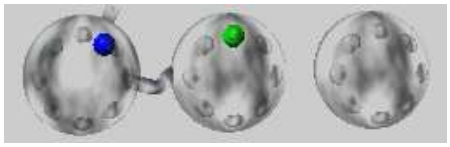
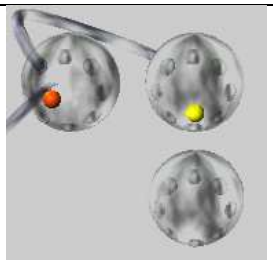
1. Locate the cell with the RED neurotransmitter.
2. If there is an axon at connection 7 of this cell (in this example, it is), continue with the YES instruction
3. If there is no axon at connection 7 of this cell, continue with the NO instruction

Note that for instructions it is not needed to indicate what kind of instruction it is (NT, AX or TST). This will be decided automatically based on the connections that the cell has.

Chapter 10: Instruction overview



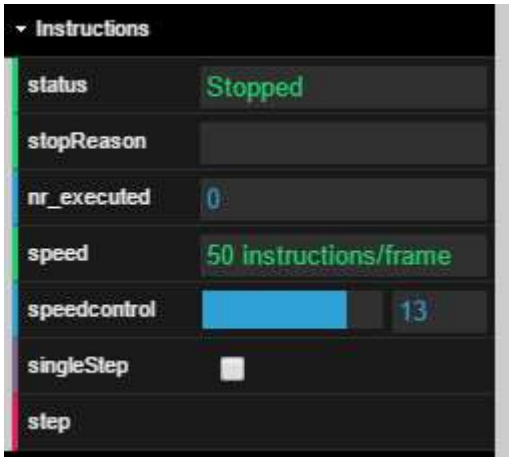
Connection numbering

NT	AX	TST
		
 Before	 Before	
 After	 After	

	Instruction NT	Instruction AX	Instruction TST
Axon 1 *Mental image*	Connection number at located cell	Connection number at located cell	Connection number at located cell
Axon 2			Next instruction if axon present (YES)
Axon 3	Next instruction	Next instruction	
Axon 4			Next instruction if axon not present (NO)
Axon 5 *At left-clicked cell, link to instruction*		NT Color for locating destination of the created axon	
Axon 6			
Axon 7	NT color to place at end of located axon		
Axon 8	NT Color for locating cell	NT Color for locating cell	NT Color for locating cell

Chapter 11: Debugging

There is an "Instructions" menu that has several options that can be of use for debugging the program.



- status: Shows if the program is running or stopped.
- stopReason: Shows why the program stopped
- nr_executed: Number of executed instructions since last activation (due to clicking a cell)
- speed: approximate speed
- speedcontrol: The slider controls the execution speed
- singleStep: Check this box to enable step-by-step execution
- step: In singlestep mode, click to execute instruction. You can also left-click the current instruction (indicated by the purple neurotransmitter) to execute it.

Setting a breakpoint

This can be done without special commands.

Locate the instruction (A) after which you want the program to stop (this must be a NT instruction).

Remove the axon (at connection 3) that goes from A to the next instruction (B), and place a new axon from instruction A connection 5, to instruction B.

As a result, the program will stop after executing instruction A because there is no next instruction. Now, by left-clicking the instruction A, it will activate the instruction that is at its connection 5, which happens to be instruction B, so execution will continue.

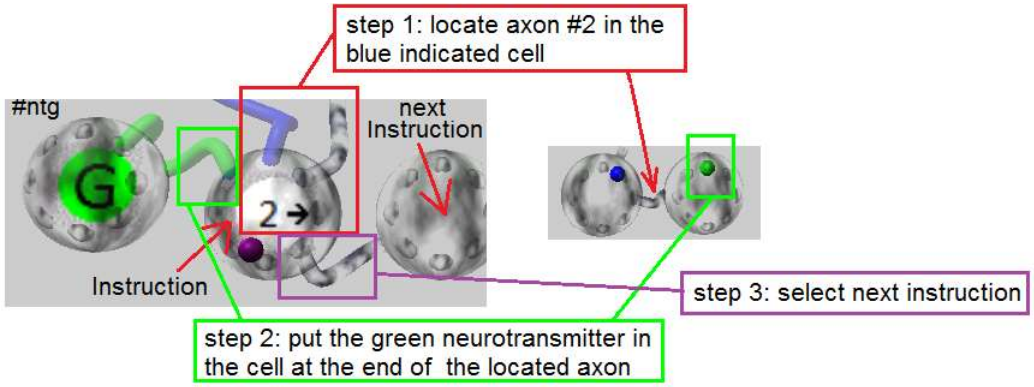
Chapter 12: Special instruction use

Suppose we want to built a calculator. A group of 8 cells is created to represent the display:

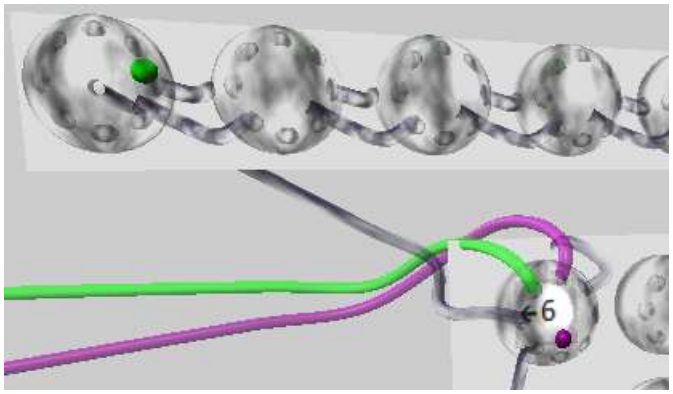
The cells are double-linked for easy access.

Now, in order to do operations on the display, a neurotransmitter must be placed in one of the cells. Once it is there, we can easily access the other cells because they are linked. But how do we get the first neurotransmitter there ?

This is the normal NT instruction again:



The normal NT instruction locates a cell with a certain neurotransmiter (here: blue). But what happens if we specify Purple as the neurotransmitter to locate ? The instruction will locate itself because the purple neurotransmitter is always located in the current executing instruction! If there is an axon at connection 2 of the instruction, that points to the destination (first cell of the display), we're all set. The instruction will put the green neurotransmitter in the first cell of the display. The following picture shows the result (connection 6 (instead of 2) was used to connect to the display):

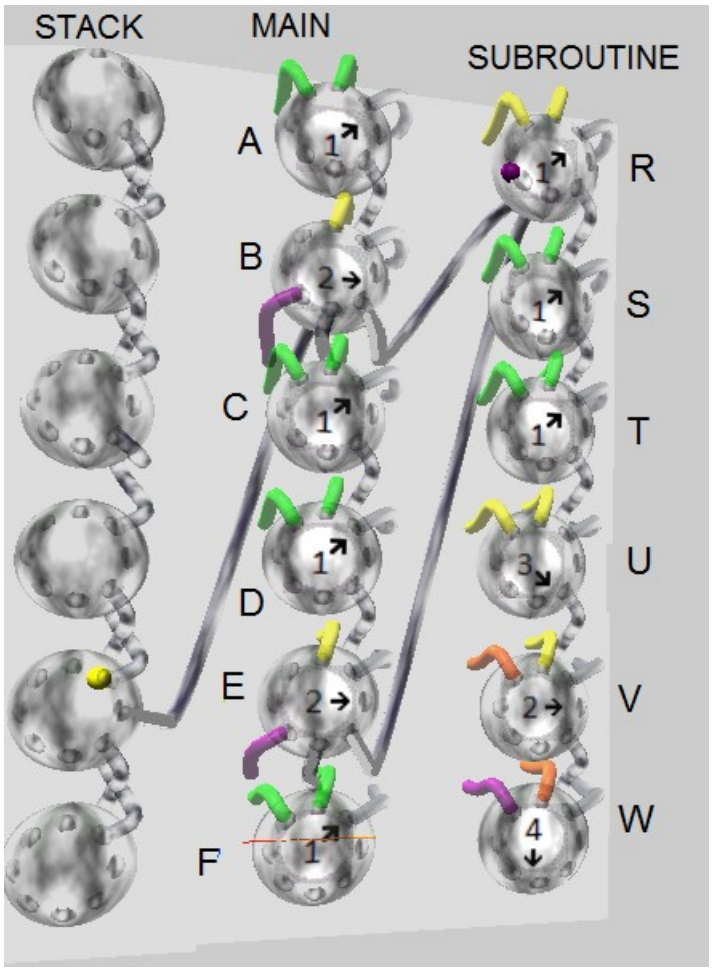


- The instruction locates axon #6 in the purple indicated cell (this cell is the instruction itself)
- It puts the green neurotransmitter in the cell at the end of the located axon.

People who know the PDP11 computer will see similarity with the "program counter autoincrement" addressing mode of the PDP11.

Subroutines and Stack

The picture shows a MAIN program (cells A-F) that calls the subroutine (R-W) twice. In the picture, the instructions with GREEN connections represent the 'normal' instructions in the program, they have no real function here. *The shown call sequence is available as an example "call_demo". Use single-step to try it.*

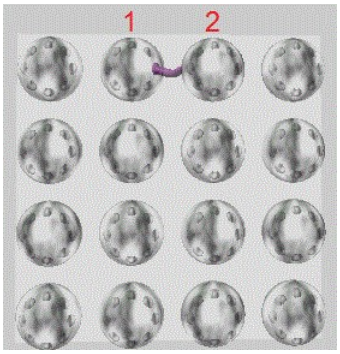


When MAIN is started, it executes A and then encounters the "Call" instruction B. How does it work ?

- B is an AX instruction that creates an axon from "Yellow connection #2" to the place where the Purple neurotransmitter is, that is to B itself. The picture shows this axon from the stack to cell B.
- Connection #3 of B is not connected to the next instruction C, but to the first subroutine instruction R. So the next executed instruction is the first subroutine instruction R.
- Instruction R increments the stack pointer to an empty position by moving the yellow neurotransmitter one cell higher. Then, the subroutine code (cells S-T) is executed. In cell U, the stackpointer is decremented to the original position. *If it is not needed to have the subroutine re-entrant, R and U can be omitted.*
- Instruction V and W implement the RETURN function. V puts a Orange neurotransmitter in the cell at "Yellow connection #2", so into cell B. Cell B has an axon from connection #4 to its next instruction, C.
- W puts the the Purple neurotransmitter (the program counter) to the cell at "Orange connection #4", so at cell C. So the next executed instruction will be C, the subroutine call is completed.
- After executing C and D in the main program, instruction E is another call to the same subroutine, it is handled in the same way.
- *The stack cells use #1 and #3 for double list linking, and #2 for storing the return link. The other connections in a stack cell can be used to save locations of other neurotransmitters during a call.*
- *The instructions U, V, W can be in common for all subroutines.*

Chapter 13: Multiple axons

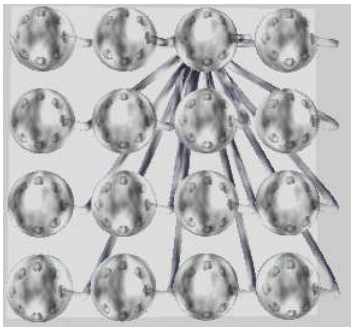
There are a few menu functions that create (or act on) several axons at the same time.



Create an axon in a group and select this axon (in the above example: from connection 2 at cell 1 to cell 2).

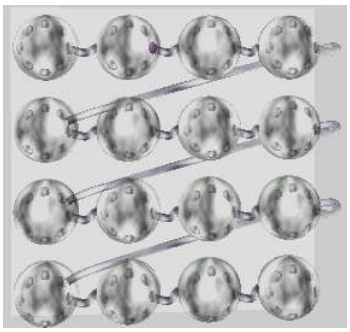
If there is an axon selected, you can do the following in the "ThisGroup"menu:

- copyAxons: Within the group, at each cell, an axon will be placed, at the same connection number and with the same destination as the selected axon. Result:

  Note that the destination cell can be in another group.

- makeChain: Within the group, at each cell, an axon will be placed, at the same connection number as the selected axon, to the same neighbour. At the end of the row, the connection will wrap to the next row. (It is required that the selected axon has the left or right neighbour as destination). A linked list can be created in this way.

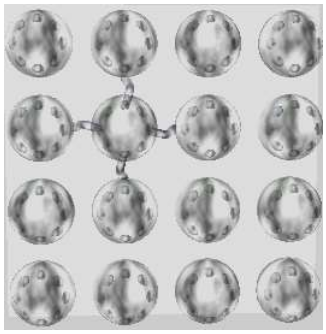
  If connection 3 is used, this can be used to chain instructions together.
  If this command is done for both (left and right) neighbours, a double linked list will be created, this can be used to create a stack.

- removeAxons: Within the group, at each cell, the axon that is at the same connection number will be removed. It can be used to undo the "copyAxons" or "makeChain" commands.

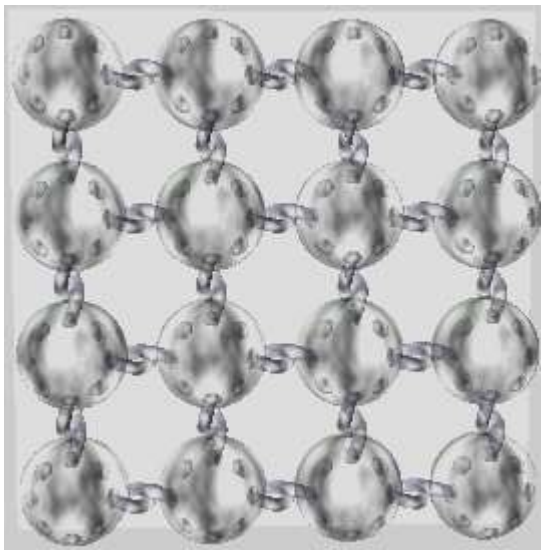
Grid structure

Several applications can benefit from a grid structure (two dimensional array). There is a special command to create this:

At first, create an array of cells, and create axons from one of the cells to its four neighbours:



Then select the cell that you just connected to its neighbours, and select the "makeGrid" command in the ThisGroup menu. This will be the result:



Chapter 14: Cell position

A cell has 3 coordinates:

1. X (from left to right)
2. Y (from bottom of the screen to the top)
3. Z (from nearby to far away)

Newly created cells will always be at Z=0, and at a free available X position. After selecting a cell, there are two options in the "ThisGroup" menu that can be enabled to allow dragging cells (with right mouse button) in X- and Y direction:

- moveCell, will enable individual dragging for all cells in the same group,
- moveGroup, will enable dragging all cells of this group at the same time.

When cells are dragged, their axons will stay connected to the same cell as before.

For projects that are not very big, the space in X and Y direction may be sufficient.

For bigger projects, groups of cells can be moved to the background (in Z direction) with the "zpos" slider in the "ThisGroup" menu. There are about 10 fixed Z positions that can be reached with this slider, so the project can have 10 layers of cells. Cells that are moved to the background can no longer be dragged to another X or Y position.

The background cells will, in many cases, not be very visible because there are other cells in front of them. You can temporary bring a group to the front (Z=0) with the "foreground" checkbox in the group menu. Now you can see all cells of this group and move them around in X- and Y direction. When you are ready, unchecking this checkbox brings the group back to the same, previously selected Z position in the background. It is practical to keep the Z=0 area free for this kind of operations.

At the moment, there is no check if different cells occupy the same position.

Appendix: known bugs

- mouse pointer mismatch when active window area has changed