StickWRLD User Manual

For use with the StickWRLD XUbuntu OVA distribution of StickWRLD

StickWRLD OVA Version 003
June 2014

Written by Dr. R. Wolfgang Rumpf
StickWRLD by Dr. Will Ray

Table of Contents

Introduction

StickWRLD is a unique visualization tool designed to help users uncover correlations and relationships by visually "browsing" their data. StickWRLD can show correlations and relationships that may otherwise go undetected by more conventional approaches such as a correlation P-test. Since the user can dynamically drive the visualization by changing parameters such as the level of significance (by increasing or decreasing the threshold p-value and residual[1] for the relationships to display) as well as the degree of relationships to display (e.g. "2-node" vs "3-node"), StickWRLD can be used as a hypothesis engine, allowing a user with "expert" or "domain knowledge" to look for specific, expected relationships – or to recognize relationships of interest – and see what other relationships are included within those thresholds.

We will begin by installing the StickWRLD Virtual Machine, a dedicated VirtualBox™ instance which already has StickWRLD installed. Then, we'll walk through a few simple examples to show you the basics of StickWRLD. Finally, we'll show you how to convert and load your own data into StickWRLD for analysis!

REQUIREMENTS

- VirtualBox™ running on Mac OS X, Windows, or linux
- Minimum Core2-Duo CPU @2Ghz or faster
- Minimum 2 Gb RAM or more
- Minimum 5 Gb free hard drive space or more

If you have any questions or concerns, or experience difficulties installing and/or executing StickWRLD, please email us at wolfgang.rumpf@nationwidechildrens.org

Chapter 1 – Installation & Configuration

This chapter describes how to get the StickWRLD VM up and running on your own machine – whether you are running Mac OS X, Windows, or Linux. If you already have StickWRLD installed, you can skip to the next chapter.

Getting Started – StickWRLD OVA

One way StickWRLD is distributed is as an OVA, or *Open Virtualization* Appliance. Essentially an OVA is a prepackaged *Virtual Machine* containing everything you need for a specific application – in this case, the OVA contains an XUbuntu operating system that has StickWRLD and all of the required libraries pre-installed. The *Virtual Machine*, or VM, runs inside of the VirtualBox™ application on your *Host Machine* – the computer that you use to run VirtualBox™. Since the OVA is a full VirtualBox™ VM, it is a fairly large file – almost 3Gb.

While you are copying the OVA to your local drive, you should make sure that you have VirtualBox™ installed. If you are on Mac OS X, VirtualBox™ will be located in your **`/Applications`** folder. For Windows users, VirtualBox™ will be installed in your **`C:\Program Files`** directory. Linux users can type **`virtualbox`** in any terminal window to launch the application.

If you do not already have VirtualBox™ installed, you can learn more about it as well as download it for free from this web site:

http://www.virtualbox.org

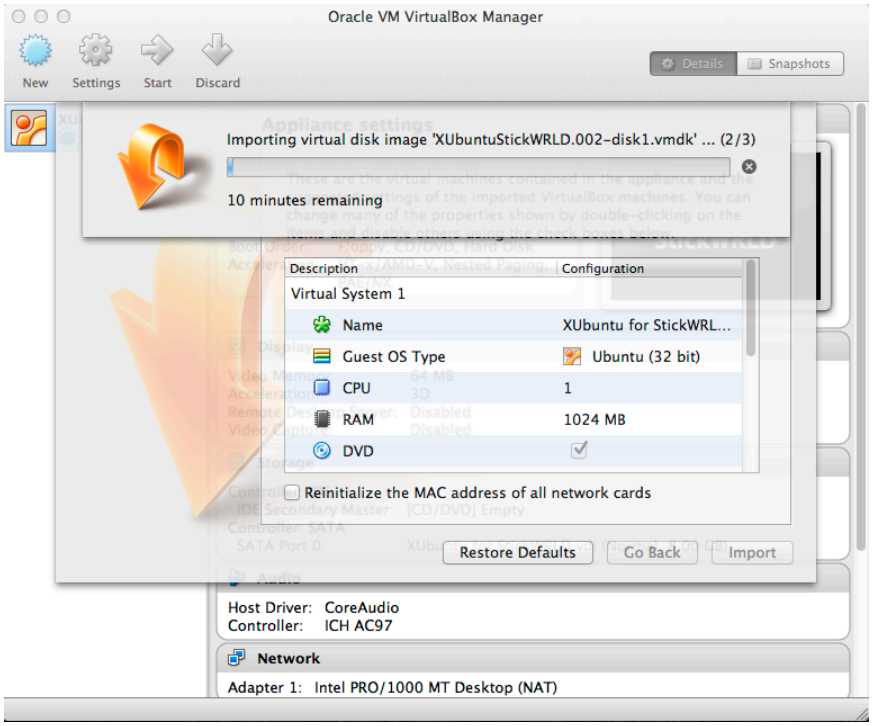
Users of Ubuntu or similar linux distributions can typically install VirtualBox™ directly in *Aptitude*, by typing the following in the terminal:

```
sudo apt-get install virtualbox-qt
```

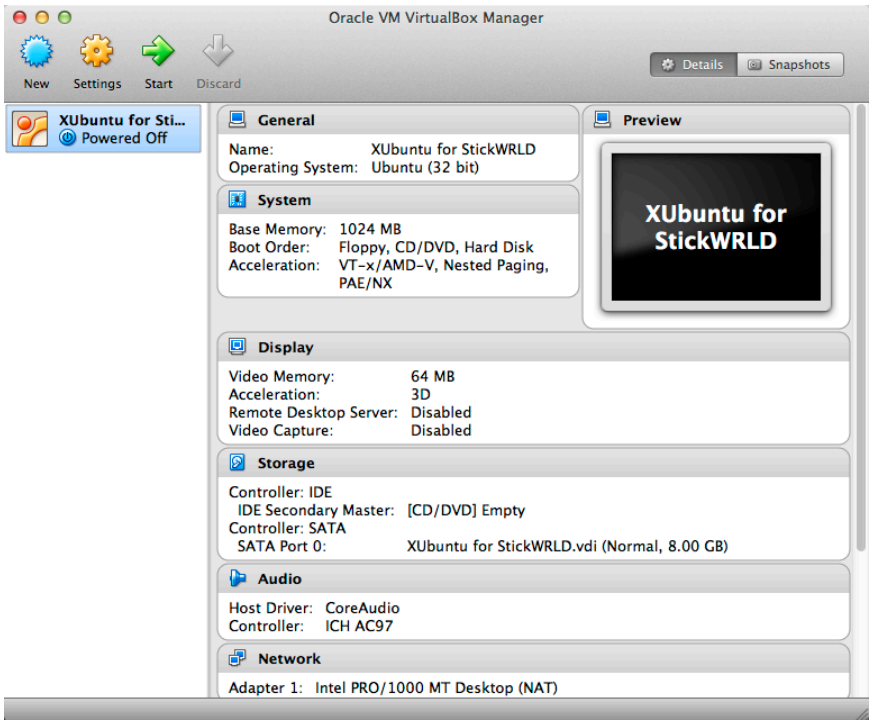
Installation

Once you have download the OVA and made sure that you have VirtualBox™ installed, all you have to do is double-click on the OVA. This will launch VirtualBox™ and start the *Virtual Machine Import* process:

VirtualBox™ will walk you through the process of importing the virtual machine from the OVA. When you see the dialog box above, simply press the "Import" button to continue:



VirtualBox™ will create a new VM and unpack the OVA into it. When finished, you'll see the following screen:

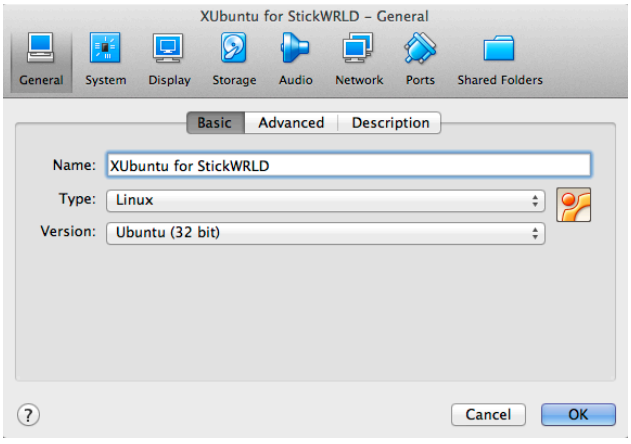


If you like, you can now simply select the **XUbuntu for StickWRLD** VM in the left panel and press "Start" – but for the best experience, you should configure the VM settings to best fit your current hardware. We'll do that in the next section.

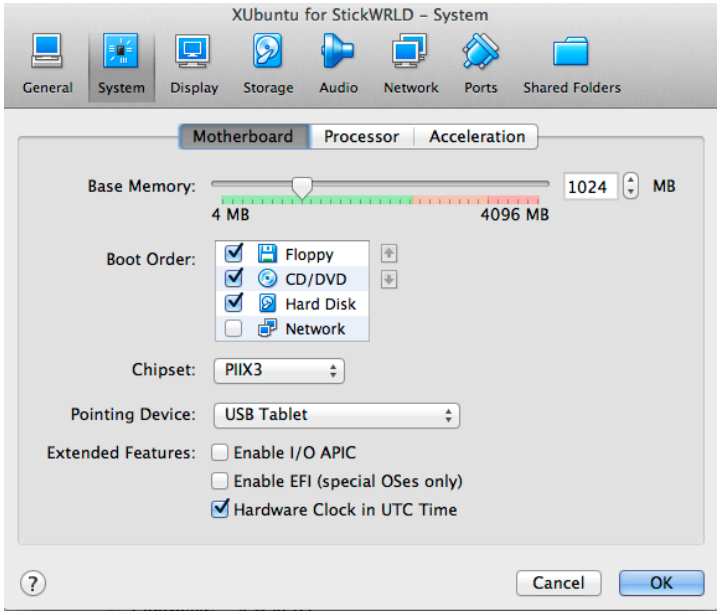
Configuration

Configuring your Virtual Machine involves telling VirtualBox™ how much RAM to let the VM have, as well as how much video memory – both of which will effect the performance of not only the VM, but StickWRLD as well.

With the **XUbuntu for StickWRLD** VM selected in the left panel, press the "Settings" button. This will open the *Settings Manager* dialog:

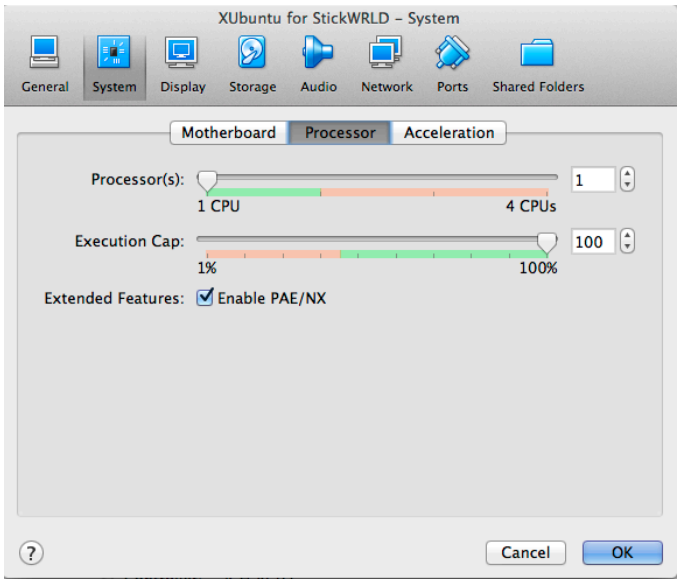


The *General* section displays the basic VM information. To configure the VM's *System* settings, press the "System" button:



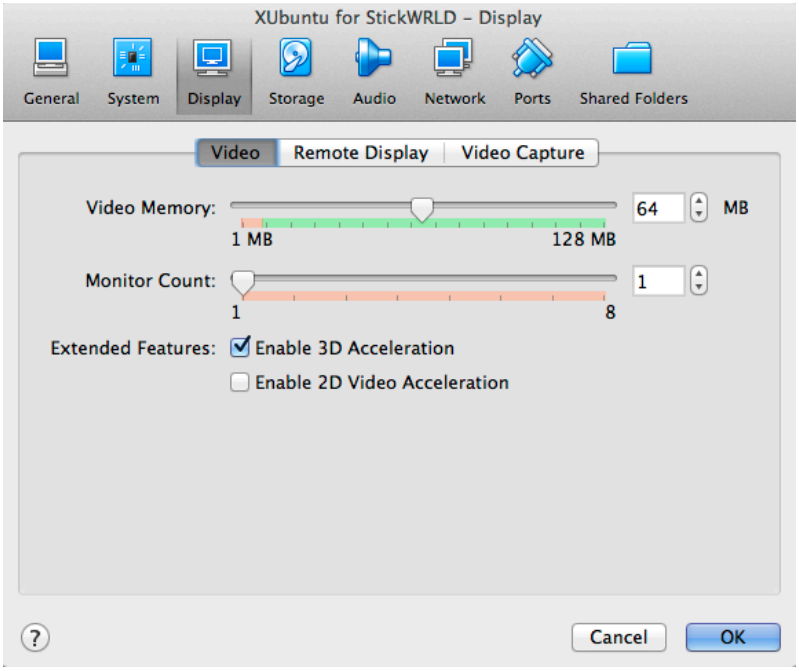
The first thing displayed in the *System* section is the *Motherboard* tab. Here you can tell VirtualBox™ how much RAM to let the VM have. We recommend giving the VM at least 1Gb (1024 Mb) of RAM, but if your system has more than 4 Gb, consider giving the VM more.

Now switch to the *Processor* tab:



Typically you can leave the *Processors* setting at 1 CPU. However, if you are running on a machine with 8 or more cores, you can increase this to 2 or 4 if you like. CAUTION: You should never give a VM more virtual cores than your machine has real cores.

Now we will move from the *System* section to the *Display* section. Press the "Display" button to see the display settings:



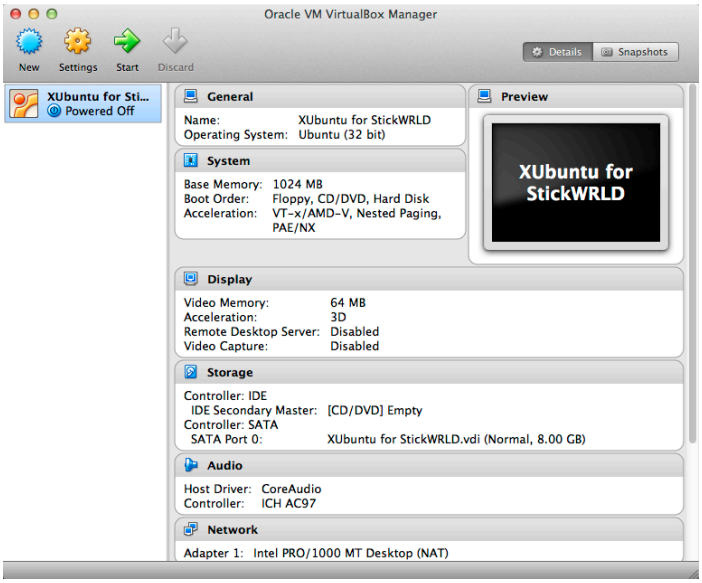
The amount of *Video Memory* will affect the screen size of the VM. If you have sufficient video memory in your host machine, you should increase the video memory to at least 64Mb or more.

There are additional features you can configure, but none of these is essential to running StickWRLD. For now, press "Okay" to return to the main VirtualBox™ control window.

Chapter 2 - Running StickWRLD

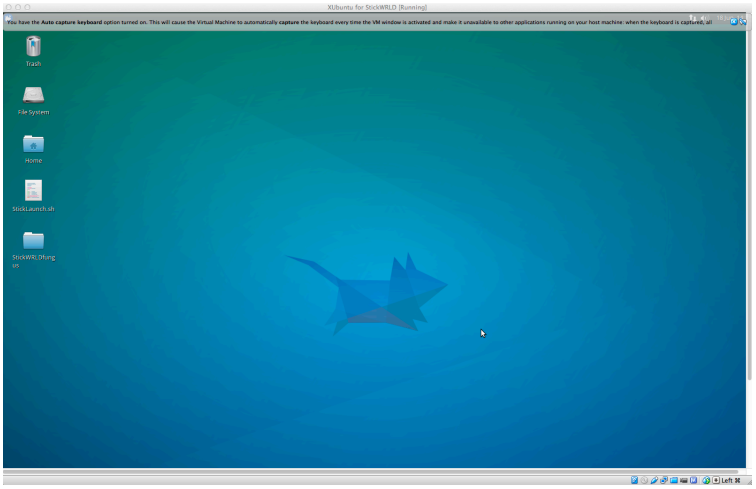
Starting the VM

Now that you have the StickWRLD VM installed and configured, let's start the VM:



With the VM selected in the left panel, press the "Start" button. A new window will open up – the virtual monitor for your virtual machine – and you can watch the VM boot up. You may be told that the VM will load into full screen mode – you can always switch back out using Command-F.

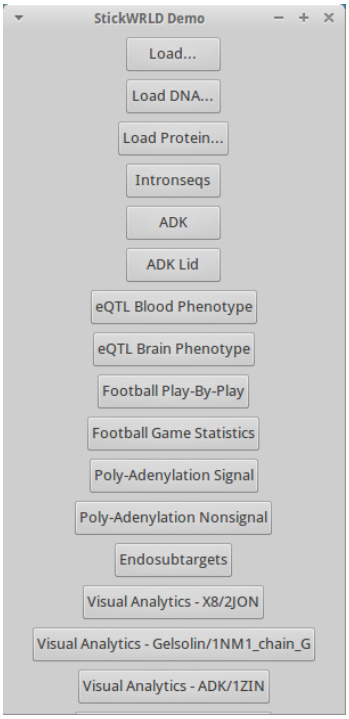
Once the VM has booted you'll see the default XUbuntu desktop – the VM is running!



Launching StickWRLD

At the top left, on the menubar, you will see this icon:

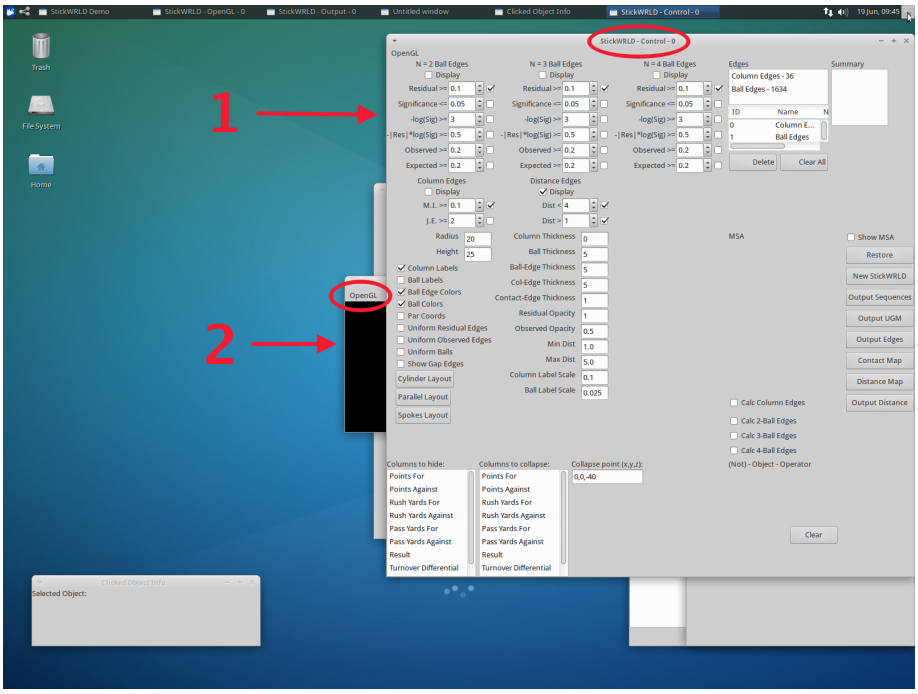
This is the StickWRLD launcher. Click on it to launch StickWRLD. The StickWRLD data loader panel will appear:



This panel lets you load your own custom data (e.g. a matrix or table of variables and values, DNA sequences, Protein sequences, etc.) or any of several example datasets. To begin, let's pick one of the easier example datasets: Football Game Statistics.

StickWRLD Example #1: Football Game Statistics

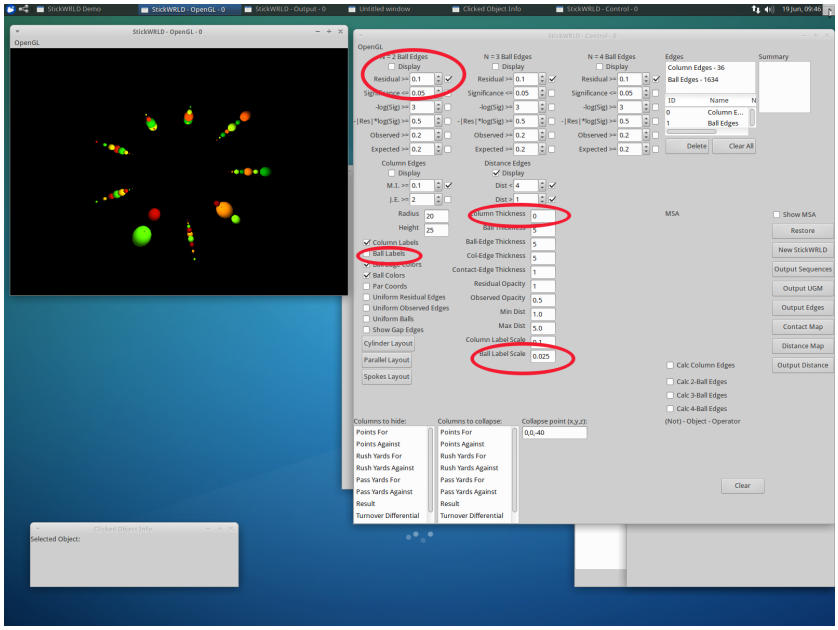
This sample dataset is derived from the openly-available NFL statistics on professional football games, which has over thirty years worth of data on various statistics. Click on the Football Game Statistics button to get started.



When you launch StickWRLD several additional windows will open up. Let's focus on two of these for now – the StickWRLD *Controller* panel (#1 in the above image) and the *OpenGL StickWRLD Viewer* panel (#2 in the above image).

First move the Viewer panel out to the side and click-drag the lower right corner to enlarge the viewer – the larger the viewer panel, the easier it will be to see the details in StickWRLD.

Next, take a look at the items circled in red on the Controller in the image below:



There are four settings that you'll want to modify to make your data easier to view in StickWRLD. Starting at the top, turn on the display of 2-node correlations by checking the box labeled "N=2 Ball Edges". This will show you relevant correlations if there are any. Note that for the default configurations in the Football Statistics demo you won't see any – yet.

Next enable the ball labels by checking the box next to "Ball Labels". This will make sure that the data values represented by the spheres in StickWRLD will be displayed so that you can easily see which values are interesting.

Now set the column thickness to 0.1. This draws a narrow line down through the column representing each variable in the data set, making it easier to visualize the column.

Lastly, set the Label Scale to 0.1. This will make the labels at the top of each column large enough to comfortable read without necessarily bleeding over onto the neighboring columns.

Now, go to the OpenGL viewer and from the OpenGL menu at the top left select "Reset View". This will make sure that all of the settings you have just modified are active.

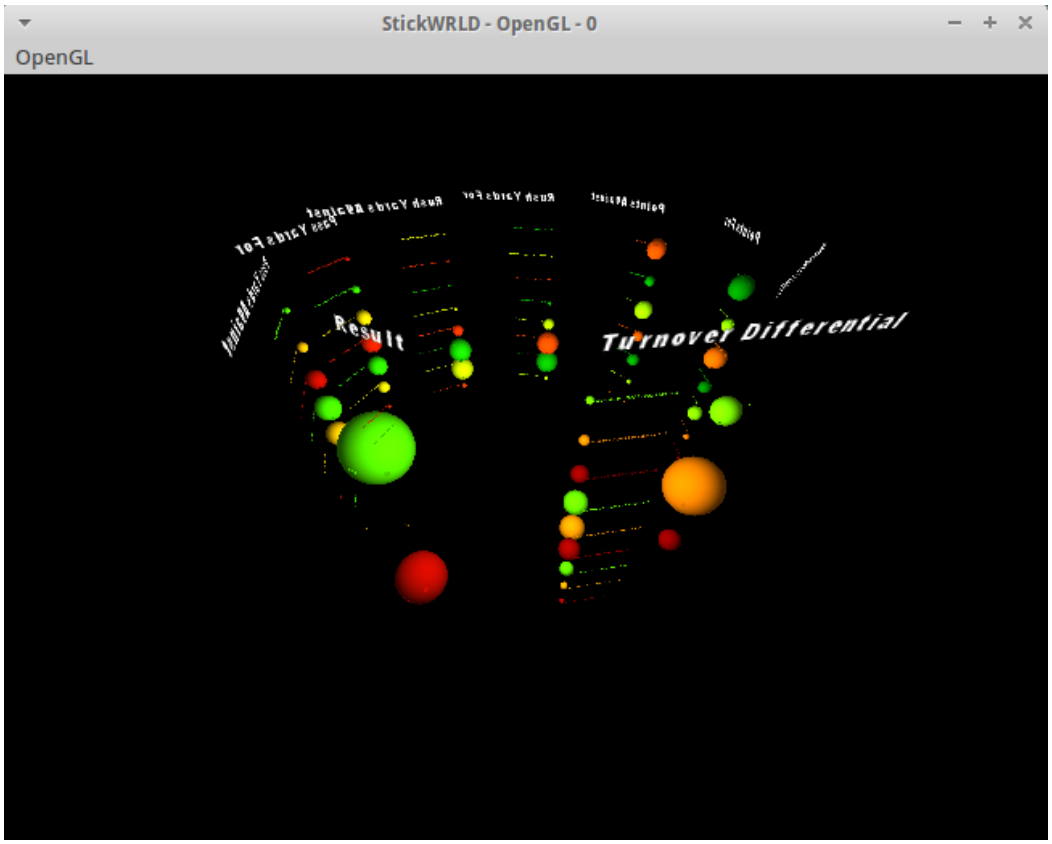
NOTE: if you ever aren't sure whether the changes you've made in the StickWRLD Controller panel have taken effect in the OpenGL view, try the "Reset View" as above.

Now we can walk through the data in StickWRLD!

There are three basic controls you will need to use for this demo. These are:

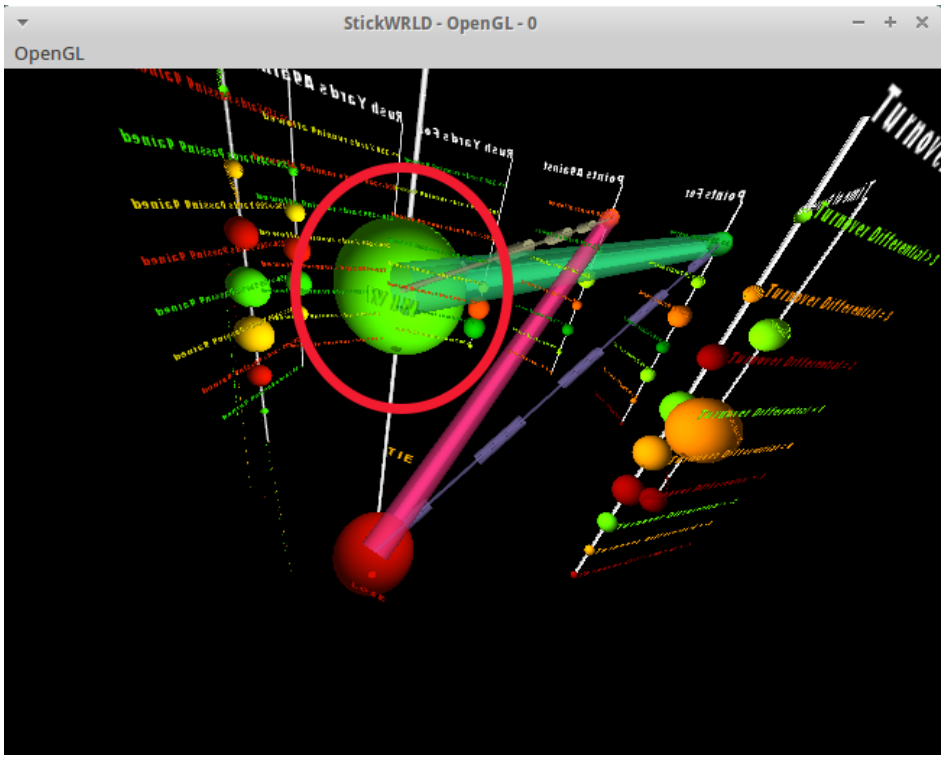
3D View Rotation	**-**	**Left-Mouse Drag**
Zoom 3D view	**-**	**Right-Mouse Drag**
Display Info ("stick")	**-**	**CTRL-Left Click**

Start by using the 3D view Rotation (left mouse button drag) to move the StickWRLD view of the data around. Move it so that you can see the RESULT column towards the front of your view, like this:

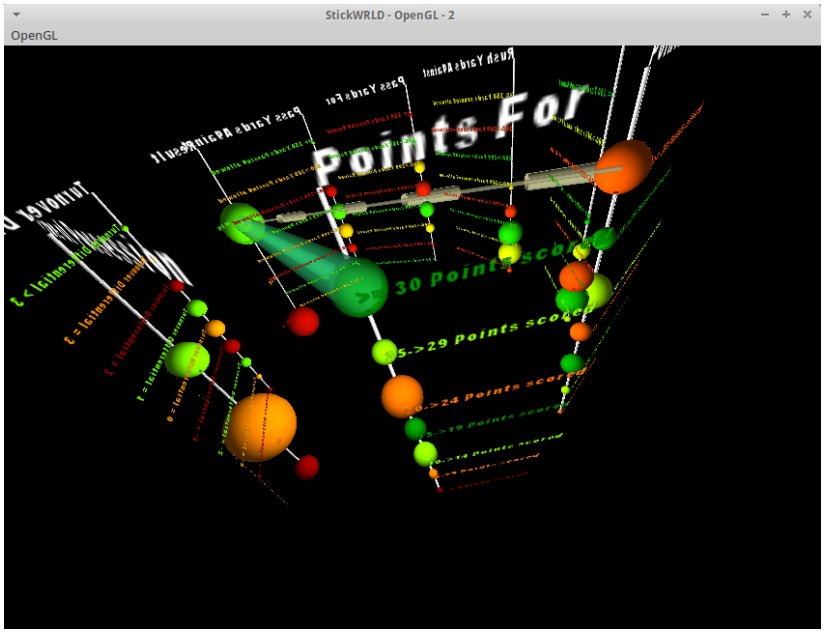


In StickWRLD, each variable in the data set is displayed as a column, where each possible value for that variable is displayed as a sphere within that column. These aren't really rows, since not all columns may have the same number of possible variables. In our football example, for instance, there are three possible results: Win, Lose, and Tie.

With each variable display in a column, the columns are then arranged in a cylinder, as seen above. This will make it easier to show the relationships between them. Let's continue by going back to the Control panel – at the top, under the "N=2 Ball Edges" heading, there is a numerical setting field for "Residual". Using the arrow buttons, click the residual down to 0.085 and watch what happens to the StickWRLD display – you should see four connectors (lines) appear connecting two of the spheres in the "Results" column to spheres in other columns:



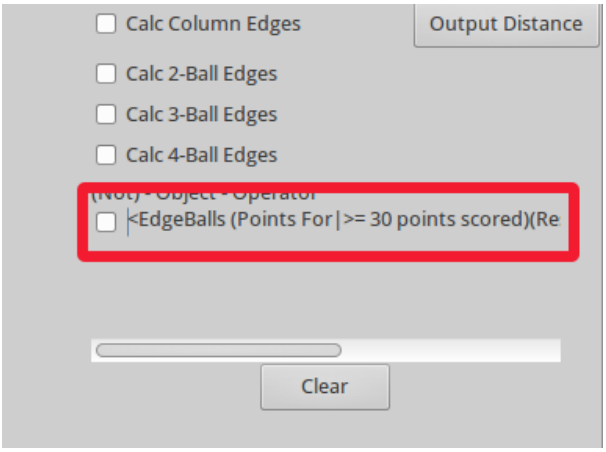
So what just happened? By lowering the *Residual* (*Observed minus Expected)* threshold, four relationships were uncovered in the data. The green sphere circled in the above image represents the value of "Win" in the "Results" column. Notice the thick solid green line leading away from it? That line links the value of "Win" to the value of "Greater than 30 Points Scored" in the "Points For" column – meaning that, based on the available data set, scoring more than 30 points is strongly correlated to winning the game:



Equivalently, notice that the value of "Lose" in the "Results" column is strongly related to the value "Greater than 30 Points Scored" in the "Points Against" column – if the other team scores more than 30 points, chances are you are going to lose.

Each of these relationships has a matching "dashed line" relationship – for example, connecting "Win" to "Greater than 30 Points Scored" in the "Points Against" column – while this sort of reciprocal relationship makes sense for this data set, in other data sets the relationships may be less obvious, and as such the display of negative correlations as well as positive is essential.

There's another command you should learn in the StickWRLD viewer – CTRL+Left Click. Go ahead and CTRL+Left Click on one of the correlation lines – for example, the green line connecting "Win" to "Greater than 30 Points Scored". You'll see the bottom right of the Control panel update with some additional information:



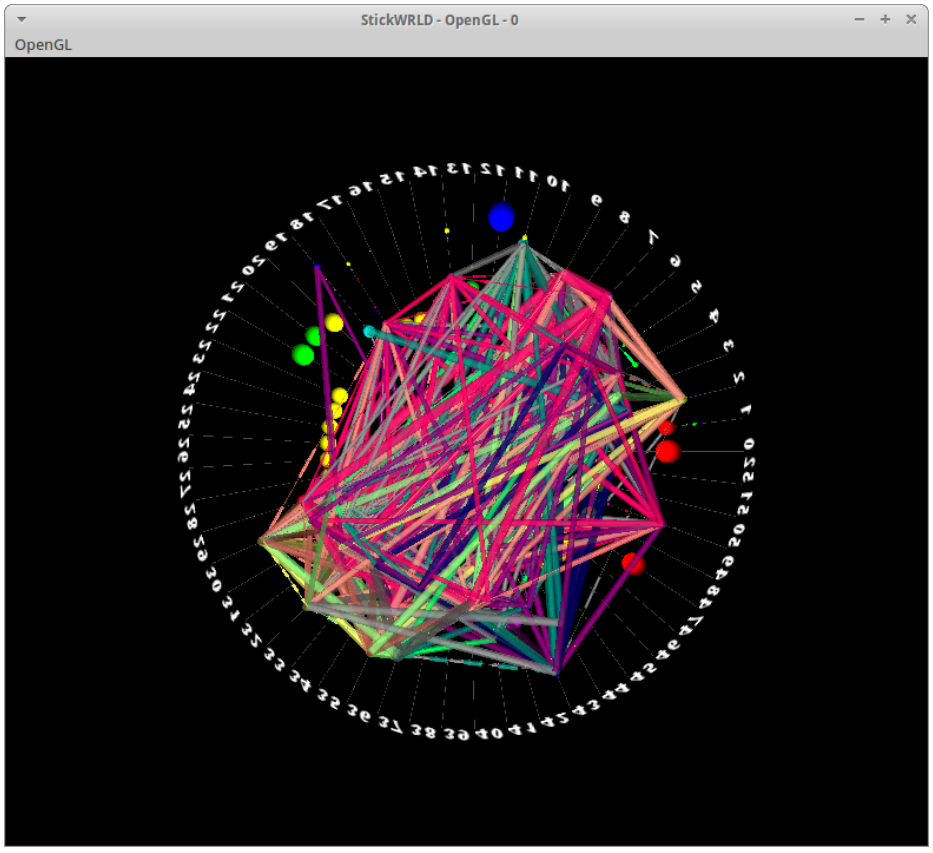
StickWRLD makes it easy for you to see the correlated information – by CTRL+Left clicking on an object you can immediately see what it is – and if it is a correlation line, you can see which values are correlated to one another. This is particularly useful for visualizations with many correlation lines – we'll see an example of one of those later. For now, play a bit with the Football data – start by dialing down the Residual threshold. Once you've dialed it down to 0.05, you should see many different correlations to explore!

StickWRLD Example #2 – ADK Lid

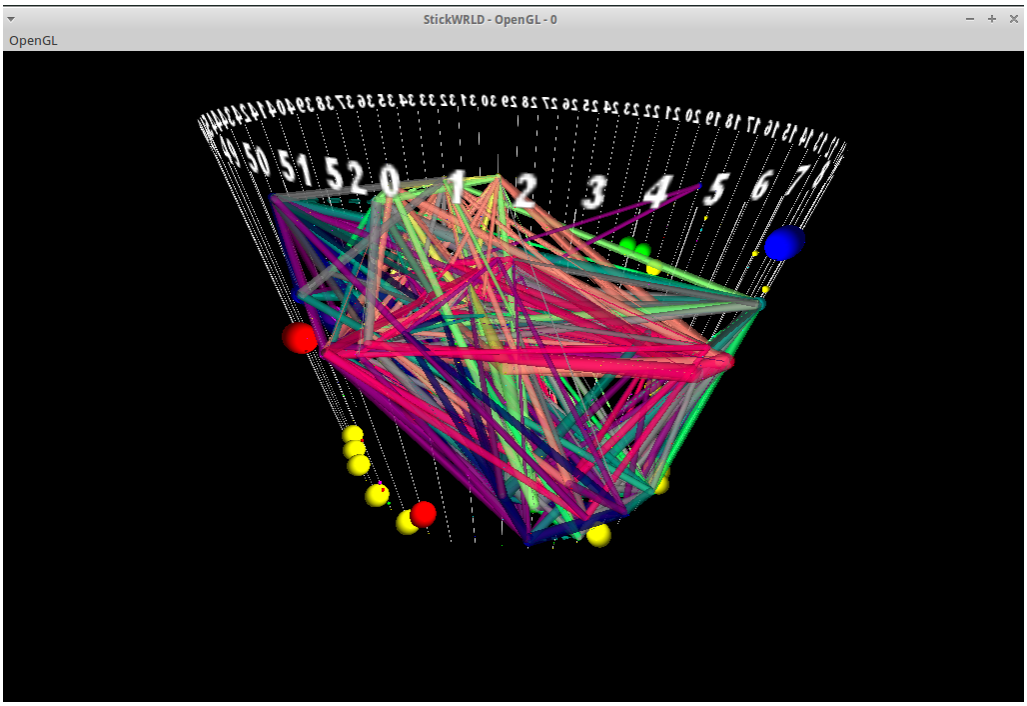
Now that you understand the basic principles of StickWRLD, it's time to move on to something more interesting than football. First close all of the open windows **except** the Data Loader panel. Then, from the data loader, select ADK Lid. This data set is a 53-residue protein sequence alignment for the Lid domain of the adenylate kinase protein. When StickWRLD views this sequence alignment data, each column in StickWRLD corresponds to a column in the sequence alignment – and the values in each column correspond to the Dayhoff single-letter designation for the corresponding amino acids.

As before, you should make sure that the 2 Ball Edge display is turned on, and that column and ball labels are displayed. Additionally, you may find it useful to turn off the "Column Edges" display.

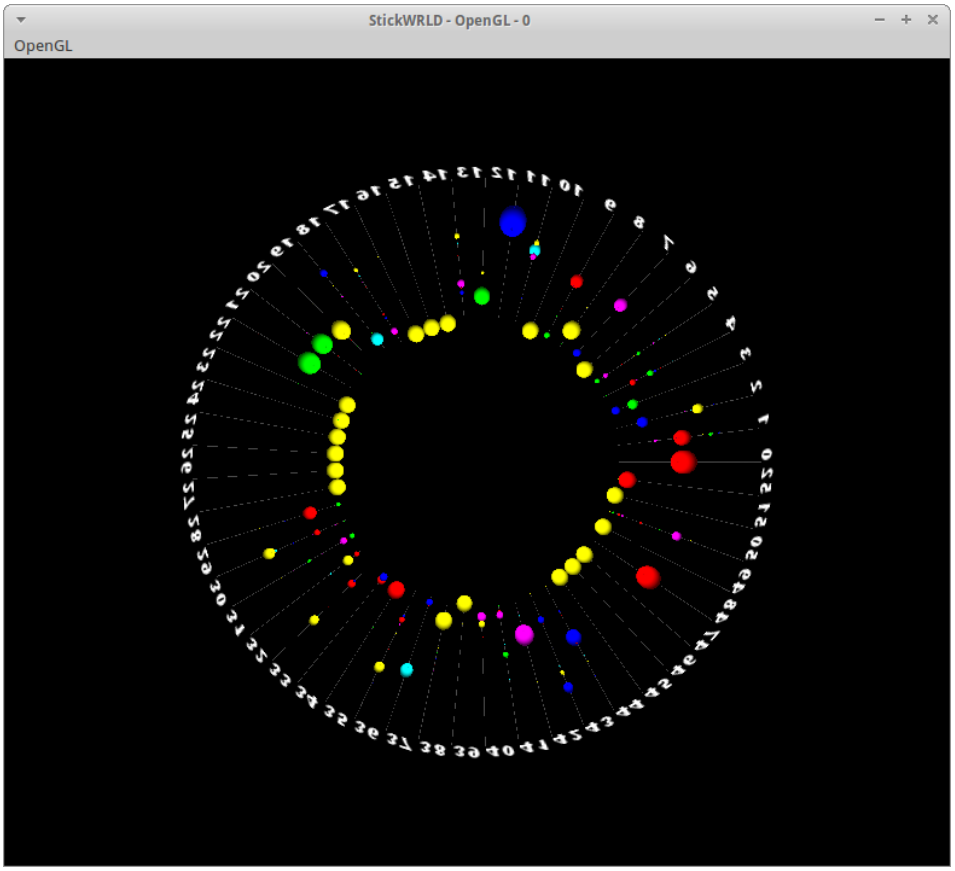
Notice that with the residual set to a default of 0.1 the data set already displays a very large number of connections:



Rotating it to see the side of the "cylinder" lets you see the density from a different angle:

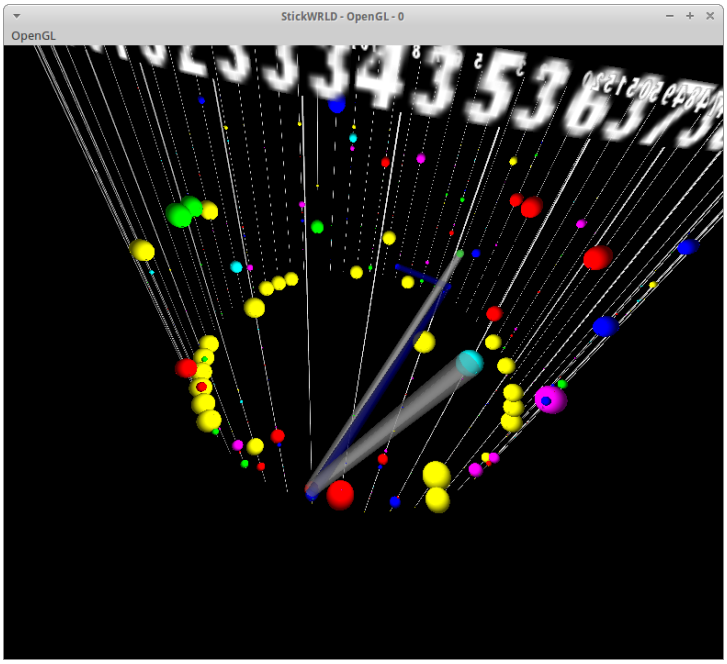


In this instance it may be useful to start by dialing up the residual threshold – changing the residual for the 2 ball edges to 0.245 will remove all of the connections:



Browsing around the data, you can see how StickWRLD can easily let you see residue frequencies – for example, position 22-27, the size of the yellow spheres indicate that the most common residue in all of those positions is alanine.

Dialing the residual down to 0.24 reveals the first set of correlations:



Notice that position 34 now has several correlations – one to a nearby position, 37, where an aspartic acid in position 34 is strongly correlated to a threonine at position 37. More interestingly, position 34 shows two correlations to position 3 - if position 34 is an aspartic acid, then position 3 tends to be a histidine; if position 34 is a cysteine, position 3 tends to be a cysteine. Additionally, a cysteine at position 3 is correlated to a cysteine at position 7 – resulting in a strong correlation of the co-occurance of cysteines at positions 3, 7, and 34.

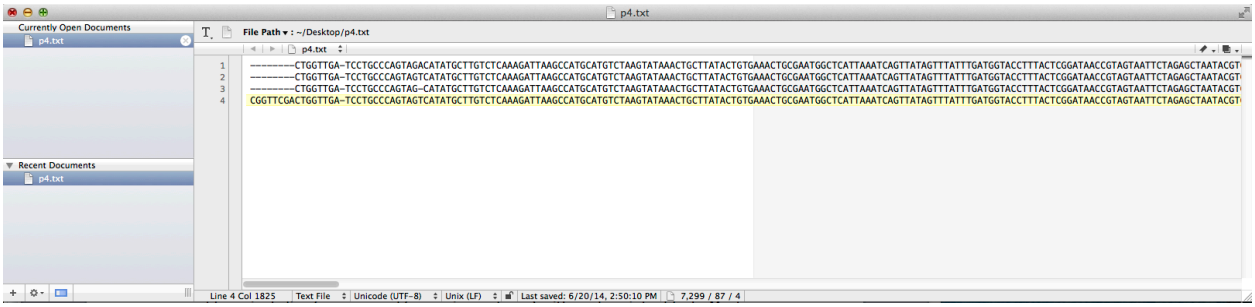
StickWRLD has been used in this fashion to detect the interaction of protein residues in the real world– residues that are not in proximity of one another based on the linear sequence data may in fact be proximal in the protein resulting from the folding of the amino acid chain, and if there are specific characteristics and/or binding requirements, StickWRLD may be used to detect them.

Additional correlations will be revealed as the residual threshold is reduced further – for now, explore on your own. Remember that CTRL+Left Click on a connector will display the correlated values in the Control window!

Chapter 3 – Loading your own data

Sequence Data

StickWRLD was originally designed to work with sequence data, so entering sequence data is fairly straightforward. You do have to tweak your data file first, however – for StickWRLD, the input sequence file (whether DNA or protein) must have each sequence on a single line, with no wrap and/or carriage returns, and no additional text (e.g. identifiers). Most likely you'll have be starting with a Clustal .aln file, and will have to use a tool like TextWrangler on the Mac, for example, to create a new text file with one sequence per line, like this:



Notice that each line is it's own sequence – there is no wrap!

Once you have your data file prepared, run StickWRLD. From the Data Loader Panel, choose the appropriate loader for the type of sequence data you have (DNA or Protein). You will have to tweak the view parameters and residual differently for every data set.

Non-sequence Data

StickWRLD can also be used with multi-axis non-sequence data – for example, clinical data. You'll have to encode the data into a format acceptable by StickWRLD. Let's take a look at some examples.

Descriptive Data

Descriptive data (e.g. "blue" vs "green" or "true" vs "false") is relatively simple to encode for StickWRLD. However, since StickWRLD currently "expects" data in protein format, you are limited to 20 values for each axis (e.g. for each variable you measure, you can have no more than 20 possible states). Let's take a look at a simple data set:

SUBJECT ID	HAIR COLOR	EYE COLOR	SEX
Subject 1	Brown	Black	F
Subject 2	Blond	Blue	M
Subject 3	Red	Blue	F
Subject 4	Brown	Brown	M

For this data there are three axis, or variables: hair color, eye color, and sex. For hair and eye color, there are three possible values each (brown, blond, and red for hair color; black, blue, and brown for eye color). Sex has two possible values (M or F).

To prepare this data for StickWRLD, first we have to come up with an encoding scheme for each axis. Let's use the following substitutions:

Hair color	**Eye Color**	**Sex**
Brown = A	Black = A	M = A
Blond = R	Blue = R	F = R
Red = N	Brown = N	

Since StickWRLD needs input to be encoded in protein codes, we use the standard single-letter codes for amino acids to represent each different value for the axes. While you must use a unique code for each value within each axis, you can re-use the codes in different axes, as above.

Using the above coding scheme are data now looks like this:

SUBJECT ID	HAIR COLOR	EYE COLOR	SEX
Subject 1	A	A	R
Subject 2	R	R	A
Subject 3	N	R	R
Subject 4	A	N	A

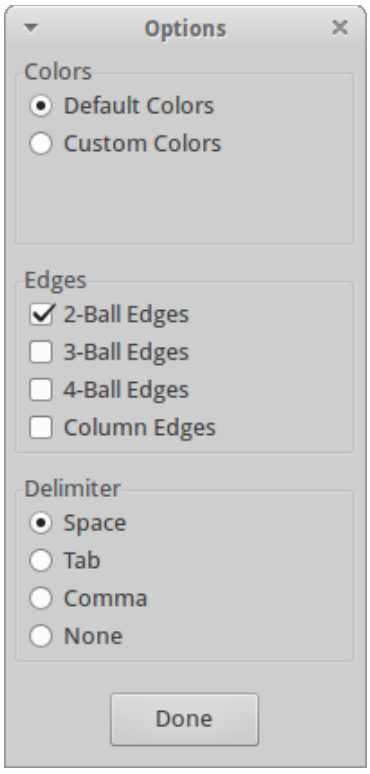
To prepare the data for StickWRLD you now have to create a text file (e.g., *data.txt*) that contains just the data:

```
AAR
RRA
NRR
ANA
```

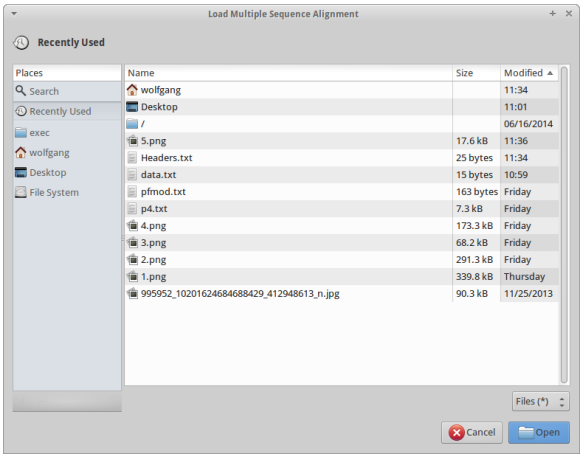
You should also create a text file (*Headers.txt)* that contains the matching variable, or column, headers – one to a line:

```
Hair Color
Eye Color
Sex
```

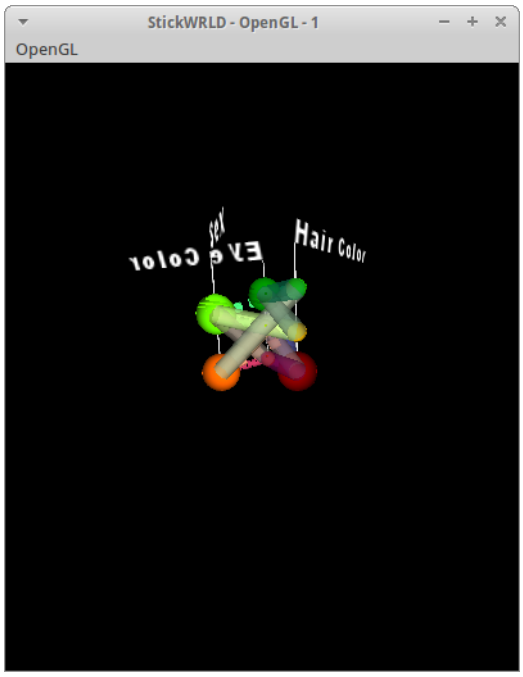
Now let's load this into StickWLRD. From the data panel, choose the "Load…" button. The options dialog appears:



For the delimiter choose "None", since the data is grouped without separators, then press "Done." A file chooser appears:



Select your data file (data.txt) and press “Open”. Another file chooser will appear immediately asking you to choose the *header* file. Select your headers file (Headers.txt) and press “Open” one more time:



As you can see, for a small data set such as this the visual is compact; larger data sets actually are easier to browse.

Numerical (Discrete/Continuous) Data

Numerical data can take the form of discrete or continuous data. In either case, you’ll need to encode the data as we did above for the descriptive data. For discrete data this again becomes a simple matter of substitution encoding, as above. For continuous data, you will have to bin your data into appropriate bins and assign those bins a value. Let’s walk through an example of binning continuous data.

SUBJECT ID	AGE	HEIGHT	WEIGHT
Subject 1	35	5’11”	180
Subject 2	47	6’2”	220
Subject 3	32	5’10”	190
Subject 4	25	5’2”	120

With a data set this small you *could* simply treat the values as discrete data and assign each a unique code. This approach won’t scale to larger datasets, however – so we will bin the data into groups.

You should be aware that there are many different binning strategies that you could use – and different binning strategies will have different results in StickWRLD – so it's important to pick a good binning strategy. In general you should try to balance the number of bins per axis with the number of values per bin – too few or too many of either one and you'll never see any correlations. One approach is "halving" – break the min/max range of each axis in half, then repeat for each half, until you have a good number of bins, each with a good number of values. We recommend that you try different binning strategies to see if different bins result in different correlations.

For the above data one strategy might be the following:

AGE	HEIGHT	WEIGHT
20-30 = A	5'0" – 5'3" = A	- 150 = A
31-40 = R	5'4" – 5'7" = R	151 – 175 = R
41-50 = N	5'8" – 5'11" = N	176 – 200 = N
	6'0" - = D	201 – 225 = D

Now for each subject simply locate the bin that they fall in and encode their value with the proper code:

SUBJECT ID	AGE	HEIGHT	WEIGHT
Subject 1	R	N	N
Subject 2	N	D	D
Subject 3	R	N	D
Subject 4	A	A	A

As before you will need to put the resulting encoded data into a text file.