Graph Theorist Documentation

Table of Contents

Have Comments, Questions, Suggestions?

The *Help* button at the top of *Graph Theorist's* screen drops down a list of options, one of which *email*. You can use this to send any comments, questions or suggestions you might have.

We *want to hear from you*! And we respond promptly to any such emails!

Did you find a bug? Let us know, and we will fix it as quickly as possible.

Don't like how something works? We'll see what we can do about it.

Is there something you particularly do like? We *would* love to hear about it!

And you can change the email address to that of a friend or colleague as well.

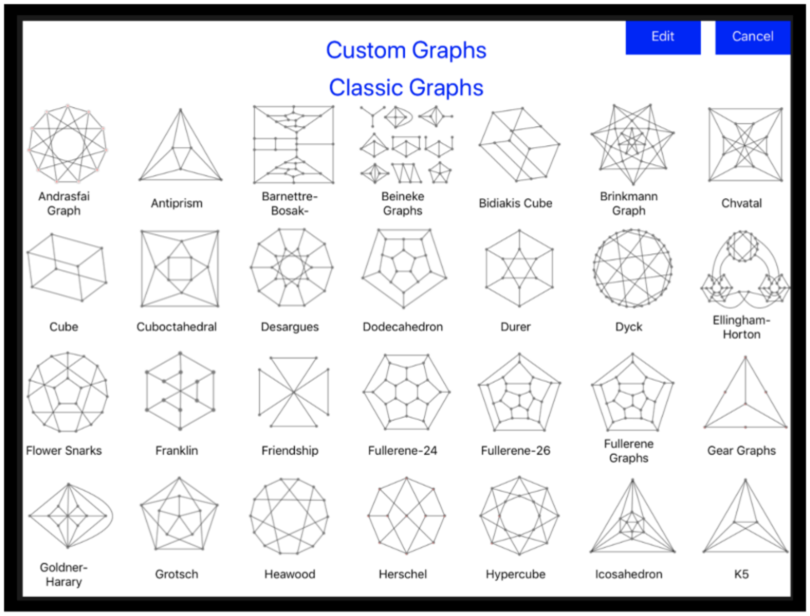
Introduction

Graph Theorist is an app for the Apple iPad and will soon appear as a Macintosh App as well. Its purpose is to allow the user to:

- Conveniently draw, store, and manipulate graphs.
- Print or save graphs as png images.
- Output TikZ code for drawing graphs in LaTeX.
- Perform searches for many graph-theoretic properties.
- Save graphs in a local database

When you power up *Graph Theorist* for the very first time, there is a simple graph on the screen for demonstration purposes. This graph is known as the *Dodecahedron*. It is one of many special graphs stored in the *Graph Theorist* built-in database. You can experiment with this graph (and some of the others in the database as well) to get a feeling for how *Graph Theorist* works. To see the other built-in graphs, tap the *Graph* button in the upper left of the iPad screen.

Note: Don't like to read user manuals? You can see *many* demos of *Graph Theorist* at http://www.graphtheorist.com, and also on YouTube.



Graph Theorist's Built-in Database

Typical Behavior

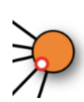
While it is possible to build a graph by just adding vertices and edges directly, you will often take advantage of symmetries as much as possible. Typically, you, load special building blocks from the *Add-ons* popover and manipulate those using various drawing tools such as rotations, scaling, and special constructions and operations. Of course, adding new vertices and edges is part of this process.

The Fundamentals

The graph *canvas* (the area shown below and bounded by a gray box) is where all the action is. It is in here that you create and edit your projects.

You can create a vertex at a specific location by double-tapping at the location.

Double tapping a vertex *selects that vertex* (and you see a tiny circle appear on the bottom left of the vertex). The concept of *selected vertices* plays a key role in *Graph Theorist* as we will discuss soon.



A *selected* vertex has a small circle on its lower left.

The status of the *lock button* (enclosed in red in the image below) is crucial to the behavior of the graph canvas.

Locked – Create Vertices and Edges

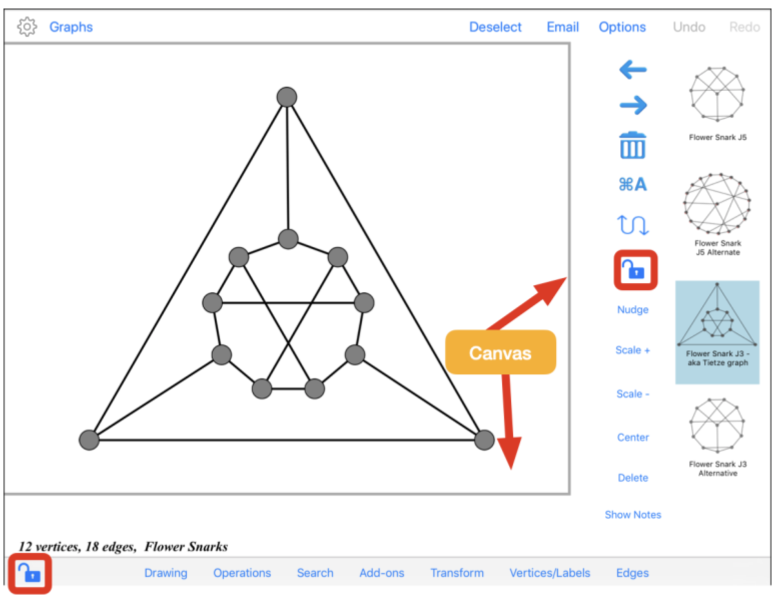
Adding/Deleting Edges

When the lock button is in locked position, then dragging from one vertex to another creates an edge between those vertices. Dragging over an existing edge deletes that edge from the graph.

If, when in lock mode, you press on a vertex for more than a moment, then a popup contextual menu will appear giving you options for that vertex and any currently selected vertices.

Unlocked – Move Vertices Around

In unlocked mode , i.e., when the lock button is open), tapping a vertex and dragging moves the vertex to a new location. Double tapping an empty portion of the display still creates a new vertex and double tapping a vertex selects that vertex.



The lock button determines the behavior of the canvas.

Pro Tip: You can lock/unlock the canvas quickly using the option tap to toggle the lock status.

You can turn this option on by going to *Settings/General/Unlock Canvas With Single Tap* and select YES. You can access *Settings* by tapping the gear icon at the top left of the screen. If you use this option, take care to not tap too closely to a vertex in unlock mode as that may cause that vertex to think you want to drag it, and it will move.

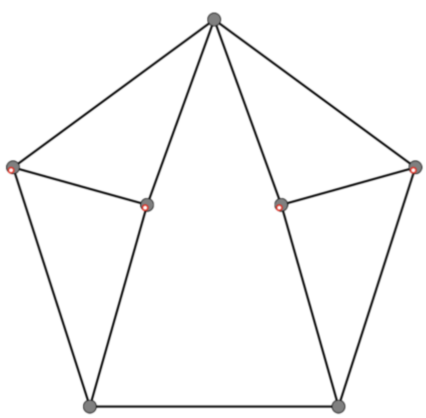
Even better, if you are using an external keyboard, you can toggle the lock status of the canvas by typing the tilde "~" key.

Selected Vertices

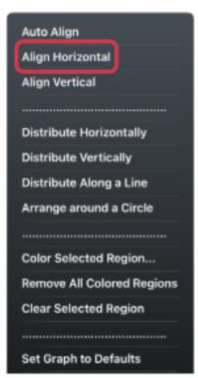
Graph Theorist performs most of its actions on the currently selected vertices.

For example, if you'd like to align some four vertices horizontally, then you would select the vertices and choose *Align Horizontal* from the *Drawing* menu.

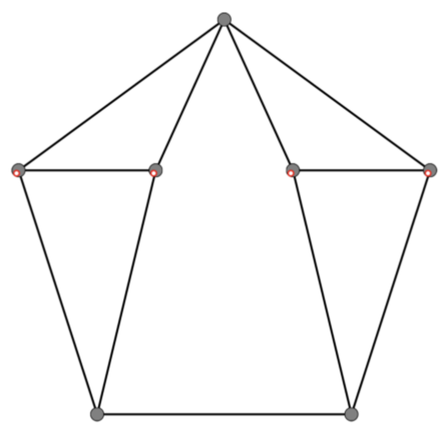
Similarly, if you want to change the color for some vertices to red, then you would select them, open the *Vertices/Labels Inspector*, and then tap the color red button. The actions of the *Vertices/Labels Inspector* window take place on the currently selected vertices.



First select the vertices to align.



Choose action.



The selected vertices align

Graph Theorist performs most of its magic on the selected vertices.

Selecting and Deselecting Vertices

There are two chief ways to select or deselect vertices.

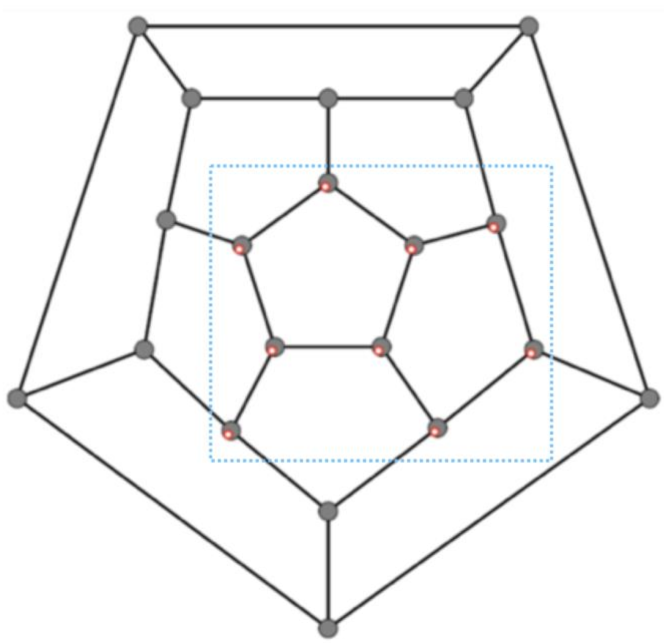
You can double tap a vertex to select it. Or if it is already selected, double tap to deselect it.

You may drag a *selection rectangle* as a shortcut to select a group of contiguous vertices. Place your finger on any empty area of the graph canvas and drag out a rectangle to enclose the vertices you want to select. Drawing a selection rectangle deselects any vertices that are not inside the rectangle.

A selected vertex shows a small circle at the vertex's lower left.

In addition to double tapping and drawing a selection rectangle, you may select all vertices at once by tapping the *Select All* button (or typing 'A' while holding down the command key when using an external keyboard).

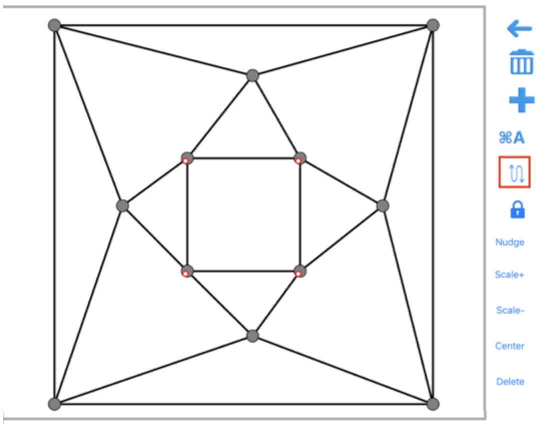
You can *deselect* all vertices by tapping the *Deselect* button at the top of the screen.



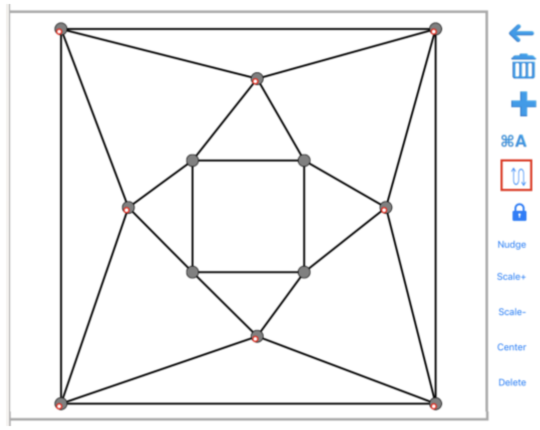
Selecting vertices via a selection rectangle

Pro Tip: As a quick alterative, you can *deselect* all vertices by dragging out a small selection rectangle that does not enclose any vertices.

Also, you can use the *inversion button* (in the red square below) to select all the currently unselected vertices while simultaneously deselecting the currently selected vertices.



Initially there are four selected vertices.



After the pressing the inversion button.

Note: Take care with double taps. A double tap in a blank portion of the display creates a new vertex, while double tapping on a vertex selects (or deselects) it.

When in unlocked mode, double tapping near, but not on, a vertex may be interpreted by the app as an intention to drag it. Thus, it is generally safer to perform vertex creation when in locked mode.

Dragging Vertices

In unlocked mode (i.e., the lock icon is open), you can tap any vertex and move it wherever you like.

Just as you can drag a single vertex around when in unlocked mode, you may move the entire set of selected vertices by dragging any one of them; all the selected vertices move in unison and maintain their position relative to one another.

Note: If you want much greater control of the location of dragged vertices, you can use the *Nudge Tool.* Tap the *Nudge button* along the side or bottom of the display to activate the Nudge Tool.

The Nudge Tool

The numbers at the bottom of the *Nudge Tool* allow you to choose the extent of the nudge in pixels. After selecting the distance that you want the vertices to move, tap one of the arrows to move the selected vertices in the direction of the arrow.

Hide the Nudge Tool by tapping the Nudge button again (which is now in red).

Like most *Graph Theorist* actions, dragging vertices can be undone.
Select undo after a drag action, and the vertices smoothly animate back to their previous position.

Undo All Recent Drag Actions

Many of the actions you perform are dragging vertices. It can be unpleasant to have to undo all of those one at a time, so *Graph Theorist* provides an option to undo all of the most recent drag actions at once. Choose the *Undo Consecutive Drags* from the *Options* menu on the top right of the display, and the application undoes all the drag operations up to the last undoable action that was not a drag.

Note: There is no redo consecutive drag option – after choosing to undo the most recent drag actions the Redo button redoes them one at a time.

Note: Graph Theorist interprets movements of the vertices using the Nudge Tool as dragging actions, and so the menu selection *Undo Consecutive Drags* includes all nudge actions as well.

Modifying Graphs

You can modify the appearance and/or structure of the current graph in many ways. You've already seen how you can add a vertex by double-tapping any blank portion of the screen, and how you can add an edge in locked mode by dragging from one vertex to another. You can reposition a vertex by simply dragging it while in unlocked mode.

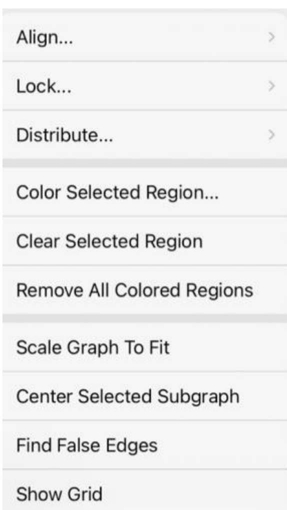
You can delete an edge by dragging over the edge, and you can delete the selected vertices from the graph by tapping the *Delete* button at the right side (or bottom) of the display. Also, you can delete a just-created edge or vertex, by tapping *Undo*.

Additionally, you can perform many special actions on the set of selected vertices by using the popover menus and/or special inspector windows. Use the buttons at the bottom of the display to activate these actions.

Note: Every *modification* that *Graph Theorist* performs on the current graph can be undone, and *Graph Theorist* supports an unlimited amount of undo's.

However, the creation of a new graph, or loading a graph from the local database are not undoable actions, and the original graph cannot be recovered unless it was saved previously. Similarly, as described later, loading a variant is an action that cannot be undone.

The Drawing Menu



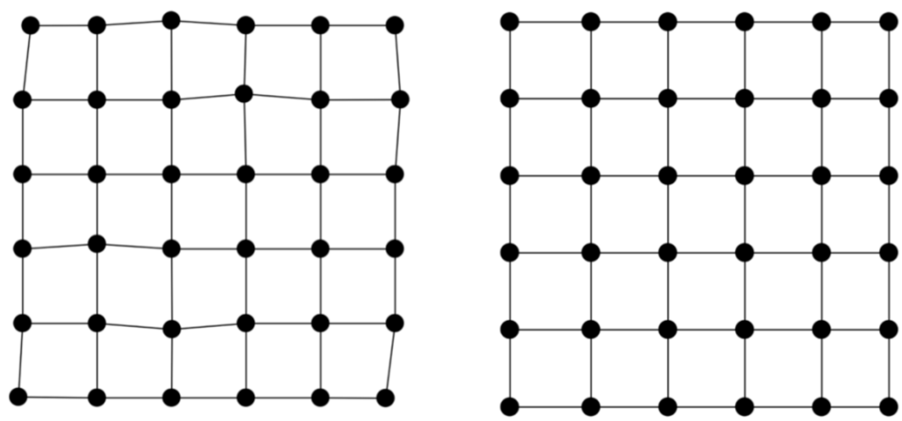
The *Drawing Menu* supports natural and standard actions on graphs. These actions affect the *appearance*, but *not the structure* of the graph.

Here is a discussion of what the various actions in the *Drawing* menu do.

The first option of Align… brings up a submenu of three types of alignment.

Auto-Align Choose this option, and *Graph Theorist* searches for edges that are almost horizontal or almost vertical and aligns them truly horizontal or vertical.

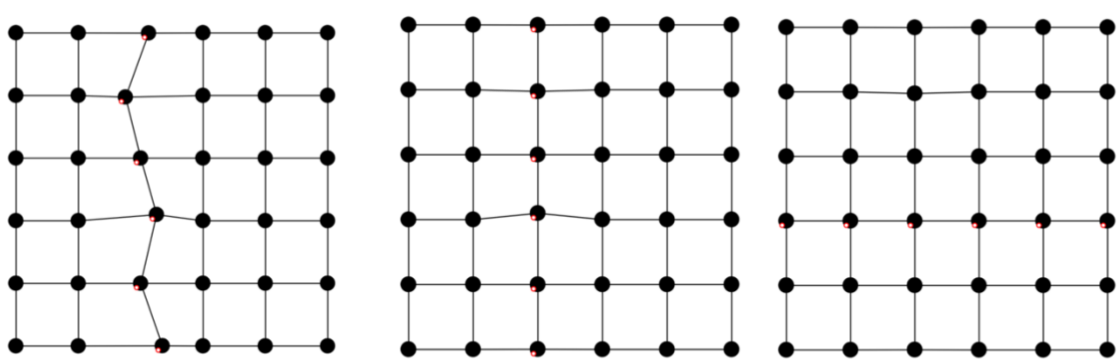
Thus, you can be a *little* sloppy when drawing your graph, and *Graph Theorist* fixes the alignments for you.



Before and after applying auto-align

Align Vertical - the *selected vertices* are all aligned in a vertical line *determined by the initially selected vertex*.

Align Horizontal – the *selected vertices* are all aligned in a horizontal line *determined by the initially selected vertex*; i.e., all selected vertices lie on an imaginary horizontal line extending from the initially selected vertex.



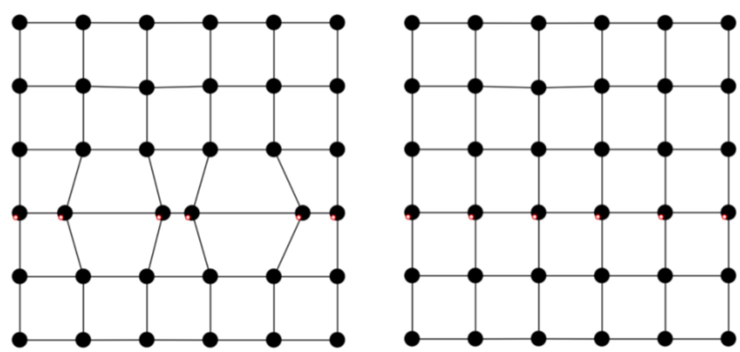
Before and after successively applying Align Vertical and then Align Horizontal

Lock Horizontal – choosing this item prevents dragged vertices from moving away from the horizontal. This places a checkmark next to the *Lock Horizontal* menu item and the option stays in effect until you choose the menu item a second time to deselect it.

Lock Vertical – choosing this item prevents dragged vertices from moving away from the vertices. This places a checkmark next to the *Lock Vertical* menu item and the option stays in effect until you choose the menu item a second time to deselect it or choose lock horizontal.

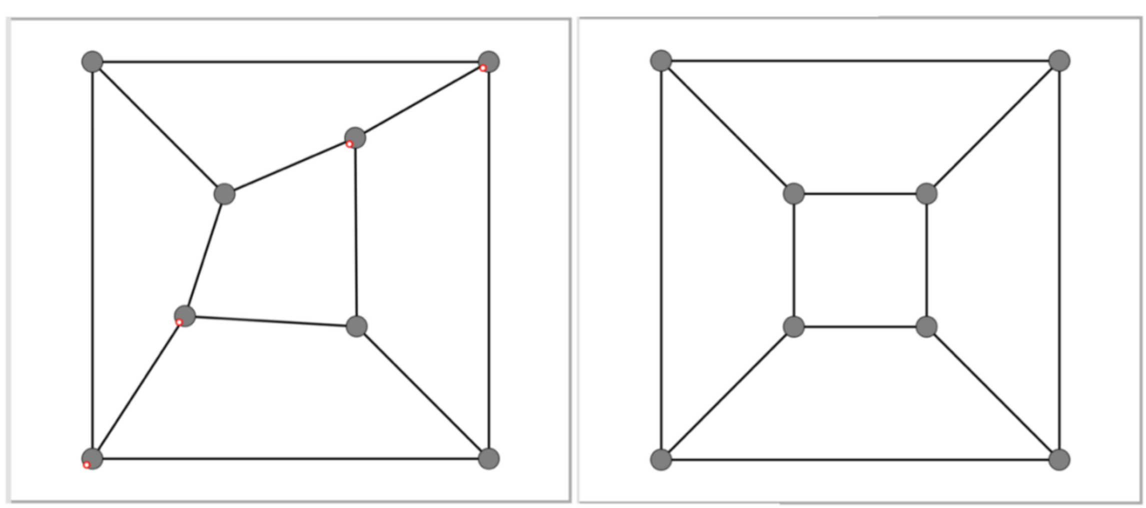
Distribute Vertically – The selected vertices are arranged so that their vertical centers are all the same distance apart. The topmost and bottommost vertices remain in their initial position.

Distribute Horizontally – The selected vertices are arranged so that their horizontal centers are all the same distance apart. The leftmost and rightmost selected vertices remain in their initial position.



Before and after applying Distribute Horizontally

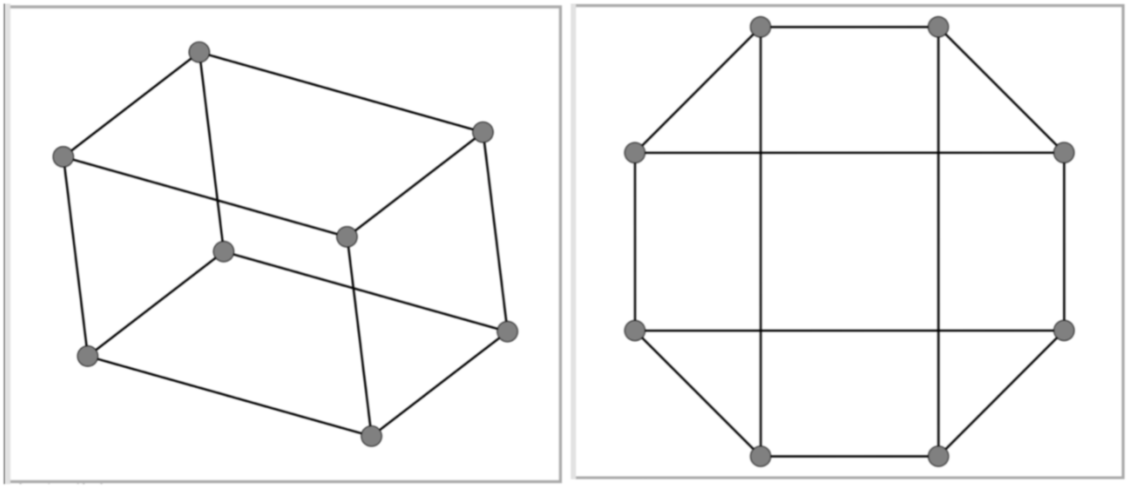
Distribute Along a Line – the *selected vertices* are aligned along the line determined by the first and last selected vertex. In addition, the vertices are distributed evenly along this line.



The Four Selected Vertices Selected from Bottom Left to Top Right Aligned in a Line

Arrange Around a Circle

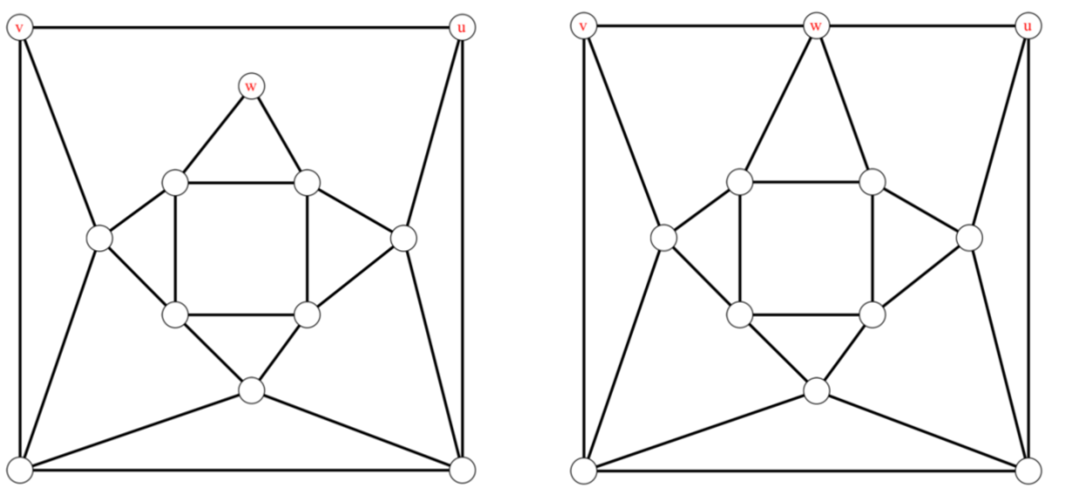
Select the vertices that you'd like to arrange in a circle and *select them precisely in the order you want them to appear*. Choosing this drawing action arranges the selected vertices in a circle. The center of the resulting circle is located at the center of the rectangle that best fits the selected vertices.



Before and After Using the *Arrange in a Circle* Action.
All the vertices in the left-hand side were first selected in order so that they formed a cycle.

Warning: All alignment actions (among others) can produce *false edges*, and you should be mindful of this possibility slipping through the cracks. A false edge forms when a vertex lies on top of an edge between two other vertices.

A *false edge* occurs when the edge between two vertices *v, u* passes through a vertex *w* making it appear that *v* and *u* are not adjacent, but that *v, w* and *u, w* are adjacent.



Aligning *v, w, u* in a line produces false edges – *vw* and *wu* are not edges but *vu* is an edge!

Find False Edges – If you suspect that a drag or other action/operation produced false edges, you can try this option to check for them. If *Graph Theorist* finds any false edges, then it alerts you to that by *temporarily* showing the offending vertices in red and other vertices in gray.

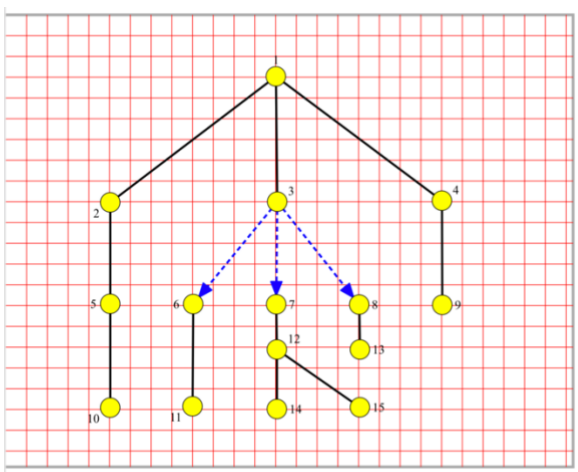
Center Selected Subgraph – *Graph Theorist* finds the smallest rectangle that encloses the set of selected vertices, called the *selection rectangle*, and moves that rectangle (along with the selected vertices) so that its center aligns with the graph canvas center.

Scale Graph to Fit – This scales the current graph so that it fits optimally inside the graph canvas Unlike most actions, this acts on the entire graph ignoring selected vertices.

Warning If the graph contains curved edges, it is possible that some of the edges may fall outside the canvas of the scaled graph.

Show/Hide Grid

This produces (or hides) a grid on the canvas to help you better place vertices.

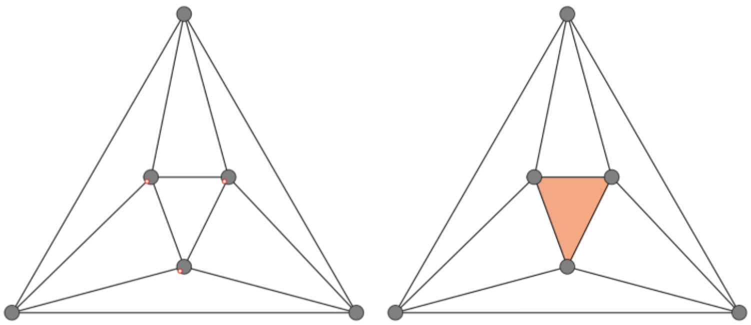


The Grid on the Canvas

Color Regions

A *region* (in the sense of *Graph Theorist*) in a graph is a portion of the display bounded within an induced cycle (i.e., any cycle without interior chords). Note that this is not necessarily a region of a planar graph.

If you select any region in the graph, and then chose *Color Selected Region* from the *Drawing Menu*, you will be given the option to select a color for that region. After selecting a color from the color picker that pops up, tap on the circle-x button at the top right to close the color picker and color the region.



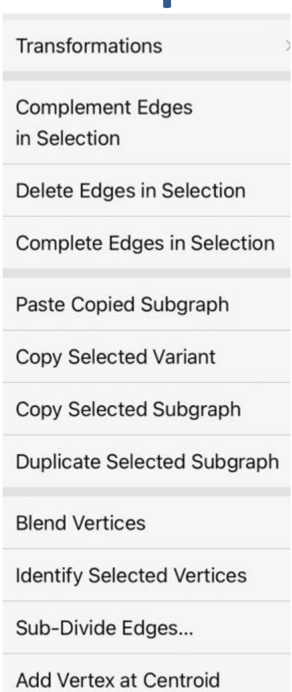
The selected region on the left is filled using the *color regions* menu option.

Choosing *Clear Selected Region* will remove any color associated with that region. Choosing “Remove All Colored Regions” will erase the colors associated with any region.

All these actions on regions can be undone and redone.

Note: You do not have to manually select the vertices of a region in order around the cycle. *Graph Theorist* can determine that the selection is (or is not) a region for you.

The Operations Menu

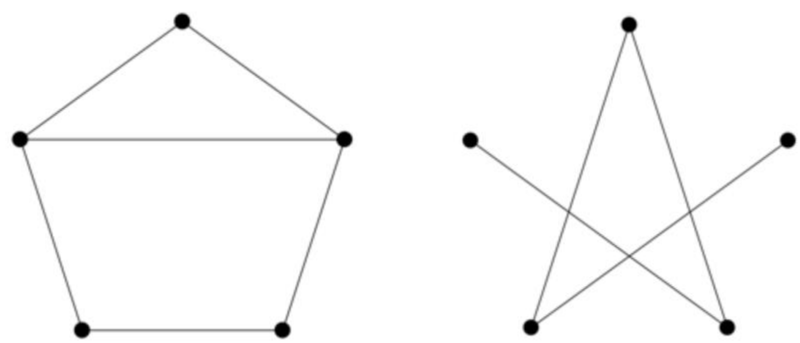


Drawing actions modify how the graph *appears*; in contrast an *operation* on a graph is an action that modifies the *structure* of the graph in some way to produce a new graph.

Graph Theorist allows you to perform many operations on the entire graph or on the subgraph of selected vertices.

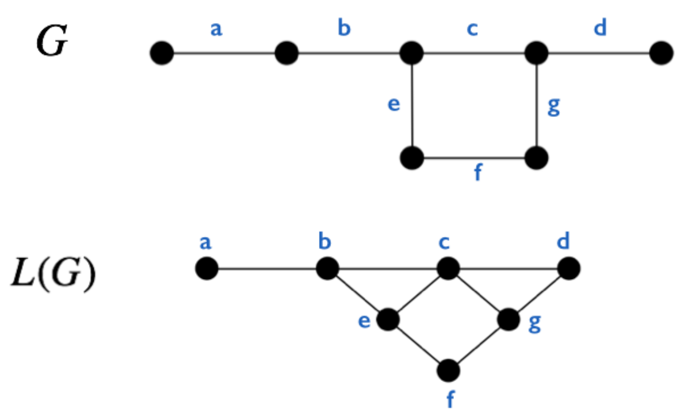
Tapping on the *Operations* button at the bottom of the display brings up a popover menu showing the available operations. There are four operations that are contained in the heading *Transformations* namely *Complement Graph*, *Line Graph, Square Graph*, and *Cube Graph*.

Complement Graph – For the current graph G, this forms the graph *complement* $\overline{G}$. Two vertices are adjacent in the graph complement if and only if they are not adjacent in G.



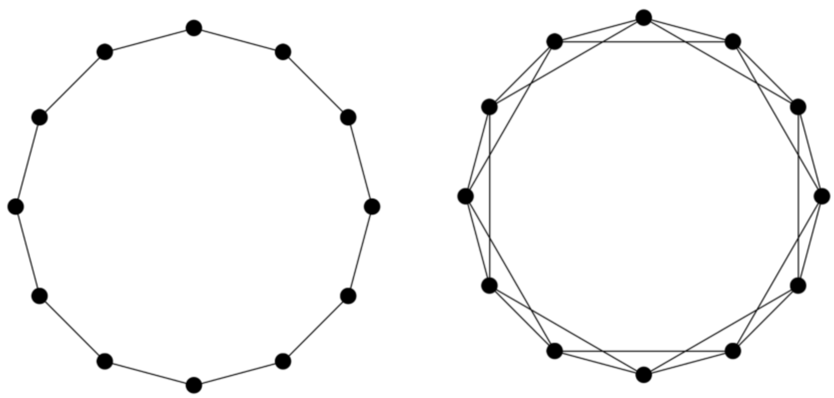
A graph and its complement

Line Graph – This operation replaces the graph by an isomorphic copy of its *Line Graph.* The line graph, $L(G)$, of a graph G has the edges of G as its vertex set and two of the vertices of $L(G)$ are adjacent if their corresponding edges are incident (see example below).



A graph G and its line graph $L(G)$

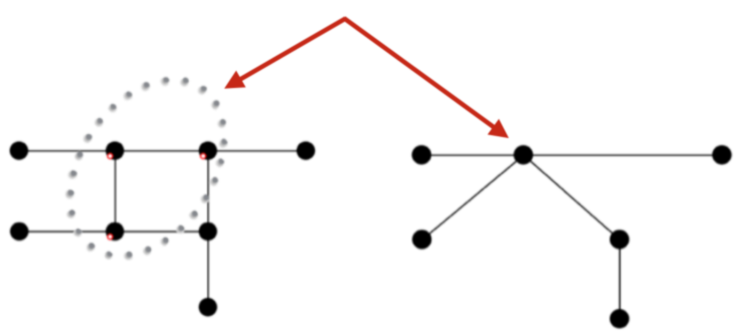
Square Graph – Produces the *square* of the current graph. This amounts to adding an edge between any two vertices that were originally a distance 2 apart (the *distance* between two vertices is the minimum number of edges between the vertices).



A graph and its square (on the right)

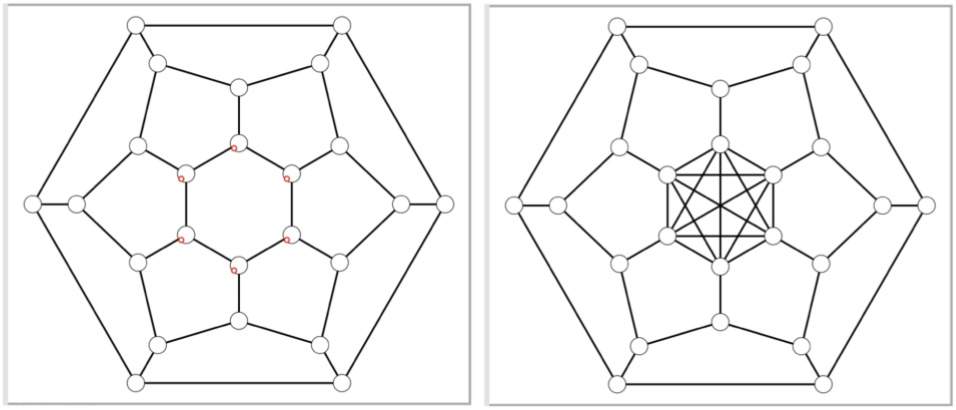
Cube Graph – produces the cube of the current graph. This amounts to adding an edge between any two vertices that were originally a distance 3 apart.

Identify Selected Vertices – Replaces the set of selected vertices by a single new vertex that is adjacent to any vertex that was adjacent to one of the originally selected vertices. The selected vertices animate smoothly towards a common location and become a new vertex where they meet.



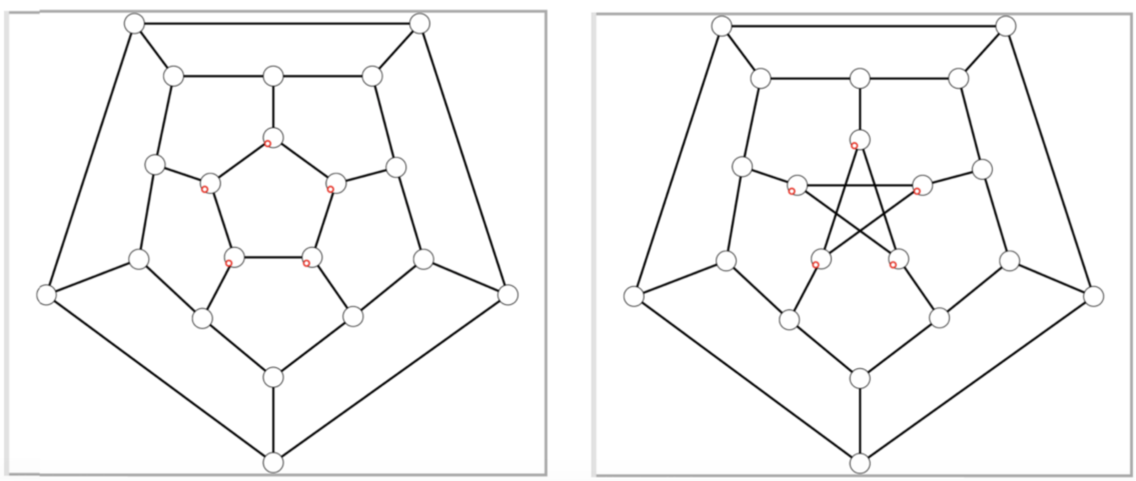
The graph on the right is obtained from the one on the left by *identifying* the three selected vertices into a single vertex.

Complete Selection – Adds all the edges that are missing between any two of the selected vertices.



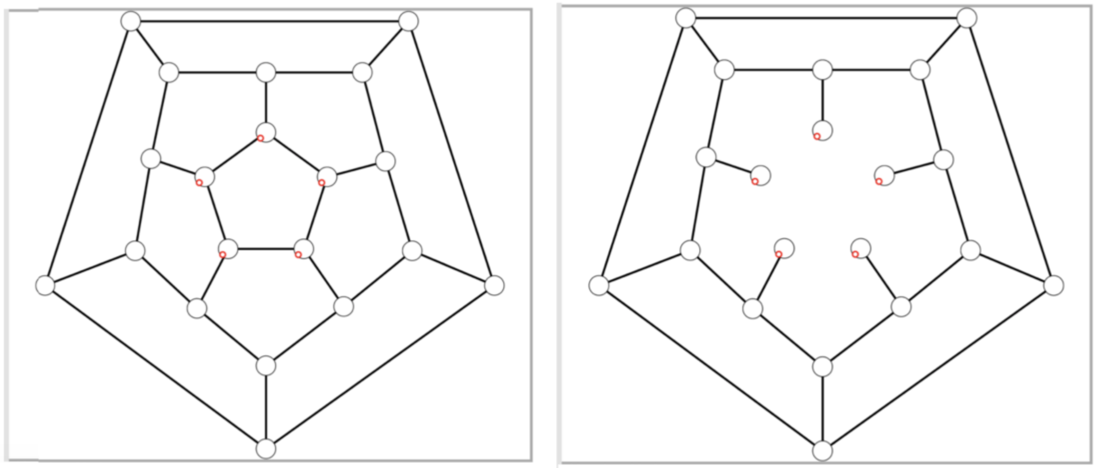
Before and after the *Complete Selected Edges* operation

Complement Selection – All the edges between any two selected vertices are removed and all the edges that were originally missing between the same two selected vertices are added into the graph. In other words, this operation forms the complement of the subgraph determined by the selected vertices.



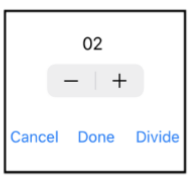
Before and after *the Complement Selection Edges* operation

Delete Selected Edges – Removes al the edges joining any two of the selected vertices.



Before and after the *Delete Selected Edges* operation

Subdivide Selected Edges – Choosing this option brings up the window shown below.



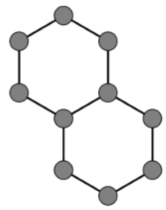
Using the +/- buttons determine the number of times you'd like each edge to be subdivided. Then tap *Divide* to perform the operation or *Cancel* to abort.
For each edge $e = xy$ joining two selected vertices x and y, a new vertex u is added that is adjacent to each of x and y and then the edge $e = xy$ is removed. That's a complicated (but more precise) way of saying, "place a new vertex in the middle of each of the selected edges" (but the original edge is deleted).

Subdividing an Edge

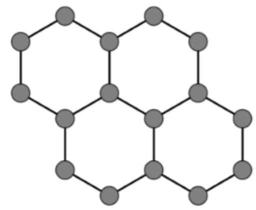
Once the *Subdivide Window* is open, select the edges to subdivide by dragging across them. When you do this a small circle appears in the middle of each selected edge.

Blend Vertices

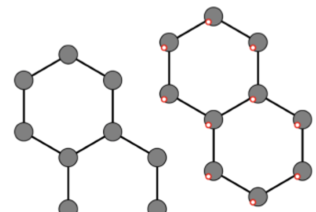
One of the more powerful operations, blending vertices requires some care.
The operation is explained in the following image.



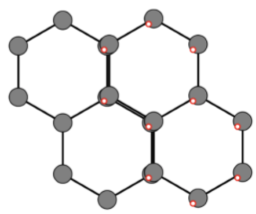
Starting with this Graph



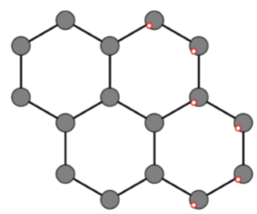
You want to create this Graph



First create a copy of the original graph



Then place appropriate selected vertices over their counterparts

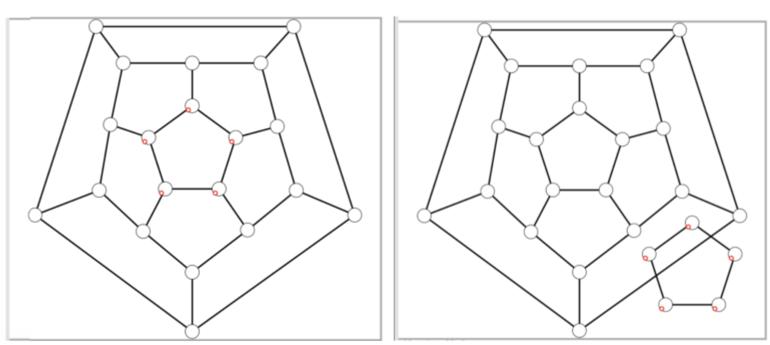


Choose Operations/Blend Vertices

Warning It is important that *only* the vertices to be blended be selected. *Do not* select the vertices below them as well.

Essentially, you use this to replicate designs that are otherwise tricky to produce. It is not necessary that the selected vertices that you want to merge with portions of the original graph lie directly above the originals, the amount of overlap shown in the picture above is plenty.

Duplicate Selected Subgraph– Adds to the current graph an exact duplicate of the subgraph determined by the set of selected vertices. Each of the original selected vertices is deselected and all the vertices in the duplicate are selected.



Before and after the *Duplicate Selected Subgraph* operation
(The resulting 5-cycle was dragged to its position after the operation)

Copy and Paste Selected Subgraph

Choosing *Copy Selected Subgraph* from the *Operations Menu* makes that subgraph available to be pasted anywhere you like. After you copy a selected subgraph, then whenever you tap somewhere on the display and choose *Paste Copied Subgraph*, a copy of that subgraph will be placed at that location. You may choose to paste a subgraph as often as you like and anywhere you like.

Add Vertex at Centroid

If you first select a simple polygon by selecting the vertices in order, then choosing this option will place a new vertex at the centroid of the corresponding polygon.

File Handling

Databases

Graph Theorist includes two distinct databases.

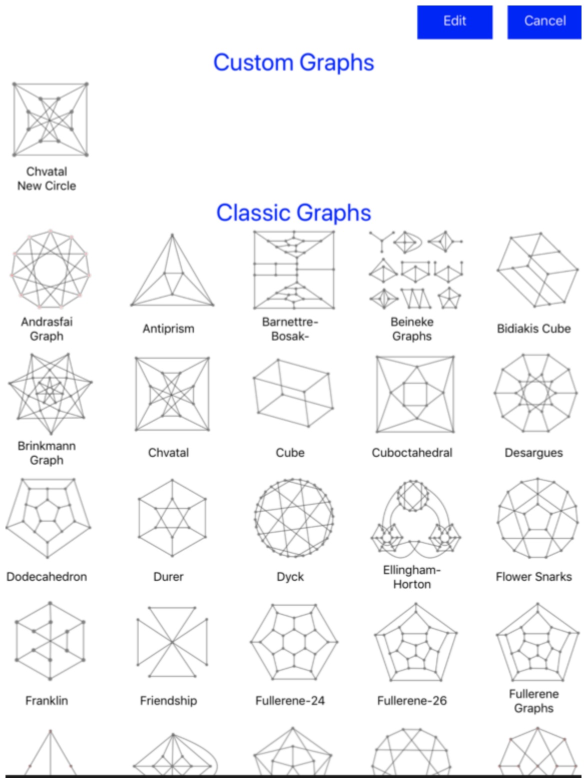
There is the built-in *Classic Graphs* database. And there is the *Custom Graphs* database where you can save and load the graphs you create.

You access both of these databases by tapping on the *Graphs* button at the top left of the screen. When you do this, you are taken to a display that looks somewhat like the image below.

You can both save and load the *Custom Graphs* you create yourself.

The *Classic Graphs* however are read only. Nevertheless, there is nothing stopping you from saving a modification of a classic graph into your custom database.

Note that in both databases, graphs are stored in alphabetical order.



Acknowledgement: Many of the drawings of the graphs in the classical database are based on Wikipedia and Wolfram MathWorld. When possible, we acknowledge the author of a particular drawing of a graph.

Creating a New Graph

To create a new graph, choose *New Graph* from the *Options* menu.
If the current graph has unsaved changes, Graph Theorist asks if you want to save the current graph before continuing.

Be careful, because deleting the current graph cannot be undone.

If you choose to continue, the current graph is deleted, and the canvas and the variant list is cleared.

If all you want to do is clear the canvas, but keep the current set of variants, then choose the *Clear Canvas* menu item from the top of the *Options Menu*. This is a much less volatile action. The canvas clears, but all the variants remain. Also, this action *can be undone.*

Warning: If you just want to clear the current canvas, but keep the current variants and such, use the *Clear Canvas* menu item from the *Options Menu*.

The *New Graph* option is destructive and also removes all variants and cannot be undone.

Clear Canvas keeps the current variants and it can be undone.

Saving the Current Graph

You can save the current graph and all its variants to the local database by choosing the action *Save to Local Database* from the *Options* menu. See the next section for a discussion of *variants*.

If you attempt to load a new graph while the current graph has not been saved, then you will be alerted that you may want to save the current graph first.

Loading and Deleting Graphs

To load an existing graph, tap the *Graphs* button on the top left of the display and then simply tap the graph you want to load.

If you'd like to delete one of your custom graphs, then first tap the *Edit* button on the database screen. When you do that the *Edit* button's title becomes *Delete*. Now tap each of the graphs you want to delete, and each selected graph becomes colored blue.

Finally, tap *Delete* and the selected graphs and all their variants are deleted. Be careful, because *this is not an undoable action*.

You cannot delete the built-in classic graphs.

Variants

While working with a graph, you might want to

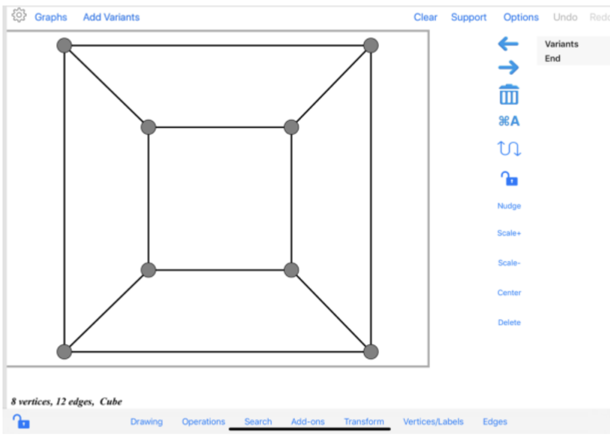
- Save various drawings of the graph,
- Save versions such as its line graph, or
- Save portions of the construction for an easy return to that start.

And more.

Variants are the way to handle each of these situations.

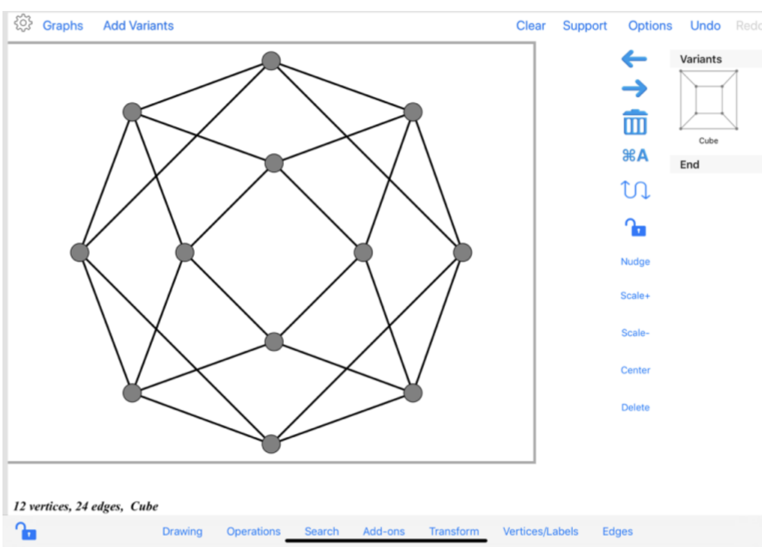
To save the current graph as a variant, simply press the *right-arrow* button and to load a variant to the canvas, first tap the variant to select it (the variant's thumbnail becomes colored blue) and then tap *the left-arrow* button.

Example 1: Creating Different Versions
Suppose you are working with the graph below and decide to create its line graph. You'd like to keep this line graph, but also want to continue to work on the original graph later on.



The solution is to make the current line graph a variant. That's as easy as tapping on the *right-arrow* button in landscape mode or the *up-arrow* button in portrait mode.

You'll be asked to name the variant. If you leave this option blank, the variant name is set to "Untitled".



Now once you are finished working on the line graph, you can save that as a variant as well and then load the original graph to resume work on it.

Example 2: Saving State
Suppose for example that you have constructed a graph up to a point but are concerned about having to undo a bunch of future edits in case you want to return to this state. Then save the current state of the graph as a variant and then you may return to that state at any time by simply loading that variant.

Example 3: Additional Drawings
You'd like to save an alternative drawing of the current main graph before continuing on.

Actions on Variants

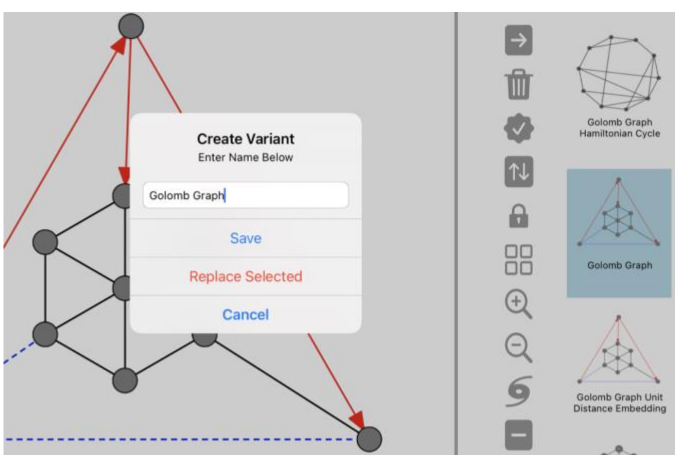
Loading Variants

You load a variant by first tapping on the variant to select it. You see the cell containing the image turn blue. Now tapping on the *Left-Pointing arrow* loads the variant into the canvas.

Warning: Loading a variant cannot be undone and it also flushes all the current undo/redo's. So before loading a variant be sure that you do not want to save the current state of the graph as a variant itself.

Replacing Variants

When tapping the right arrow button to save the current graph as a variant, you get an option to replace the currently selected variant (if there is one). This option appears in red to remind you that the replaced variant cannot be recovered.



Delete Variants

You can delete a variant by first tapping the variant to select it and then tap on the *trash* button. This action cannot be undone.

Reorder Variants

You are not stuck with the default ordering of variants. To move a variant to a new position, press down on the variant image, and when the image appears to pop up, drag it to the desired position. When a variant is moved to a new location it becomes the selected variant.

Add Variants of Saved Graphs

If you choose the *Load Variants* button from the *Options* menu, then you are taken to the graph database and when you select a graph, its variants are appended to the variants list of the current graph.

Caveats and Nuances

Note that the thumbnail image of a variant is made directly from the graph canvas. So, if the graph only takes up a small portion of the canvas then the thumbnail image may look smaller than you might desire. You can use the *Drawing* menu option *Scale Graph to Fit* to make the thumbnail as prominent as possible before creating the variant.

When you save a graph to the local database, all its variants are saved with it.

Saving/Loading to External Files

In addition to saving a graph to the local database, you may choose to save a graph (and all its variants) to a file on your iPad, cloud or service like Dropbox. To save a graph to an external file choose *Export Current Graph* from the *Options* menu and follow the resulting instructions.

And you may choose to load a saved graph as well. Choose *Load External Graph* from the *Options* menu.

Saving/Loading Databases

In addition to saving a single graph to the local database, it's possible to save and load entire database sets. Simply choose *Backup Local Database* from the *Options* menu to save the current local database to a file on your iPad, to a cloud service, or to a service such as Dropbox.

Conversely, you can replace the current local database by choosing *Load Graph Database* from the *Options* menu and then choosing a saved database file. Of course, be aware that this replaces all the graphs in the current database.

Searching

Using the *Search* popover, you can search the current graph for some of the most common graph theoretic entities.

Note: Searching is restricted to graphs having fewer than 64 vertices.

At the beginning of each search, a spinner appears at the bottom right of the display and continues to spin until the search either finishes or you choose to stop it. In addition, in most cases, a percentage-completed value appears to the right of the bounding box in landscape mode and below the box in portrait mode. This indicator is a rough indication of how far along the search is. In some cases, this percentage increases much faster as the search progresses.

As soon as the search begins, the *Clear* button at the top of the display turns into a red *Abort* button; tapping this button at any time curtails the search. *Graph Theorist* disables the remaining top buttons for the duration of the search.

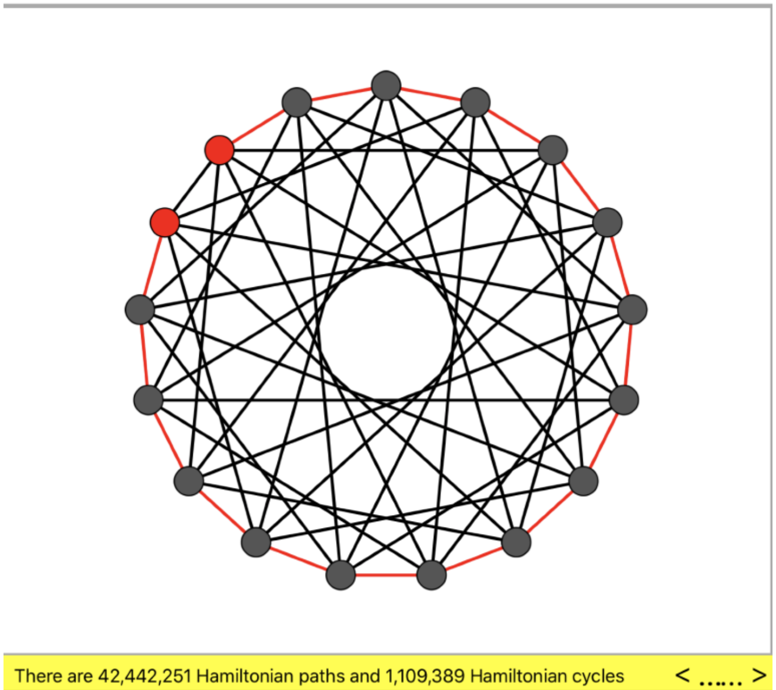
Pressing the red Abort Button stops the search.

Graph Theorist shows the progress of the search and also the final results in the *info-strip* at the bottom of the display (just above the button-bar, see figure below). You can use the black < … > buttons on the far right of the info-strip to navigate back and forth among the search results. The first 1000 search hits are saved and can be viewed using the < … > buttons.

When the result of a search is a set of vertices and/or edges, *Graph Theorist* displays the elements of each found set in red.

Tip: The effective way to use the < … > buttons is to tap slightly inside the < or >.

For example, searching the Andrásfai graph (shown below) for longest paths shows that there are 42,442,251 Hamiltonian paths and you can broiwse through the first 1000 of those



The search shows one of the 42,442,251 Hamiltonian paths (a cycle in this case).
The edges that comprise the path (or cycle) appear in red.
Tap the < ... > buttons to view up to 1000 of the resulting outcomes.

Note: The display of found vertices and/or edges in red is just temporary! As soon as you press the *Deselect* button *twice*, the vertices and edges will again show whatever properties you've set on them.

End of Search

At the end of the search or after you abort the search, the *Abort Button*'s title turns back to *Deselect*, but it remains in red. At this point you can use the < ... > buttons to peruse through the results of the search.

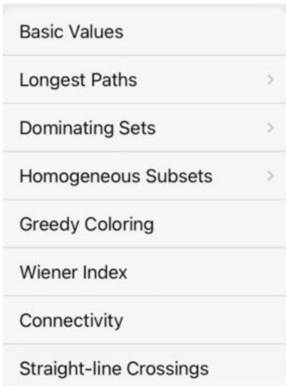
Also, at this point, most buttons and menus are not available. However, the Options menu does have two choices still available. It is possible to copy the image of any of the search results and then paste those images as you see fit. You may also add adornments to the graph (probably based upon the search result you're viewing). For example, you might want to enclose some portion of the graph with a rectangle, label particular vertices, or change the nature of particular vertices before the graph reverts back to its standard form.

After pressing the *Deselect* button once more, its color turns green and all buttons and menu options become available, the current view continues to show the results of the search, but the < ... > buttons disappear and so it is no longer possible to scroll through the other results of the search.

Pressing the green *Deselect* button now removes the results of the search and returns the graph to its initial state.

Here are the things you can search for as of the current version.

The Search Menu

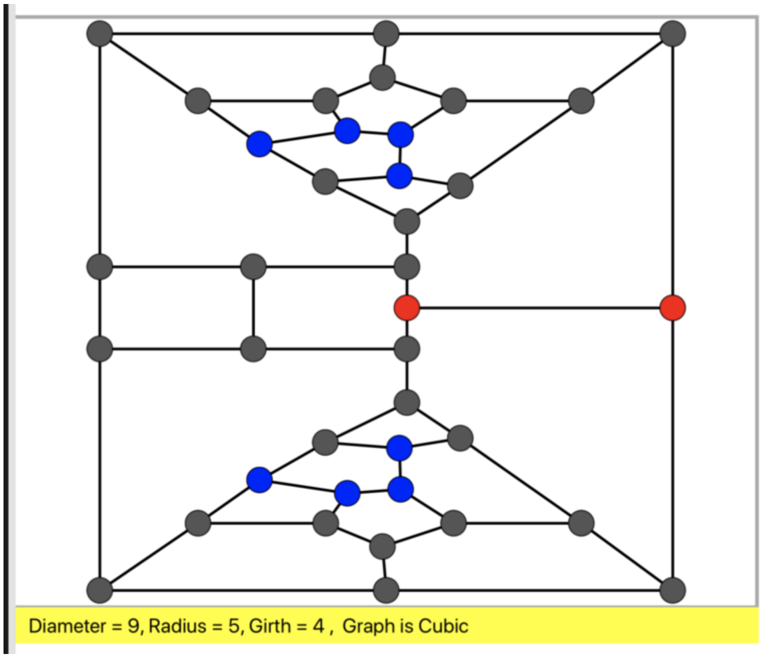


Basics – Diameter, Radius, Girth …

This option determines the radius, diameter, girth, diametrical vertices, and the central vertices, the maximum and minimum degree and displays those values on the bottom info-strip. It also identifies trees, regular graphs, and bipartite graphs.

The vertices that comprise the *center* of the current graph (i.e., the vertices with eccentricity equal to the radius) are depicted in red, and diametrical vertices (those that are the end vertices of some diameter) appear in blue.

The < … > buttons do not appear in this case because there is just one result.



The vertices shown in red are the central vertices. Those shown in blue are diametrical.

Maximum Independent Sets

This search finds *all* the maximum independent sets of vertices in the current graph (an *independent set* is a set of vertices no two of which are adjacent).

The bottom info-strip shows the number of maximum independent sets and the <…> buttons on the right of the info-strip allow you to navigate through up to 1000 of the sets.

As the search progresses, *Graph Theorist* consistently displays the most recently found independent set.

If you choose to abort the search before completion, the info-strip displays what it has found so far, and you can use the < … > buttons to scroll through the result found before you aborted the search.

Maximum Cliques

A *clique* is a set of mutually adjacent vertices, i.e., the dual concept to that of an independent set. This search finds *all* the maximum cliques in the current graph.

The bottom info-strip shows the number of maximum cliques, and the <…> buttons on the right of the info-strip allow you to navigate through them.

As the search progresses, *Graph Theorist* consistently displays the most recently found clique.

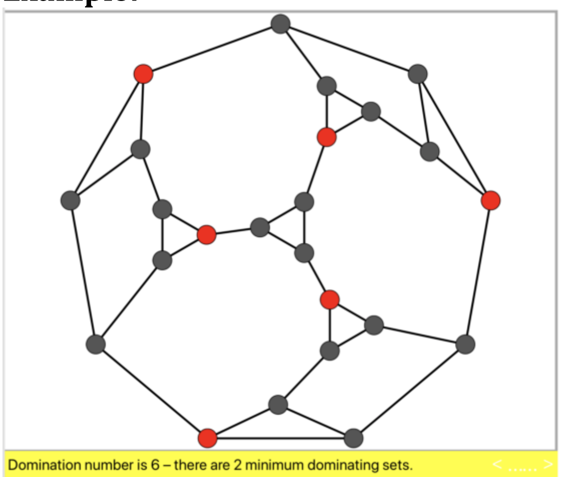
If you choose to abort the search before completion, the info-strip displays what it has found so far, and you can use the < ... > buttons to scroll through up to 1000 of the results found before you aborted the search.

Minimum Dominating Sets

A set *S* of vertices of a graph is a *dominating set of vertices* if every vertex that is not in *S* is adjacent to a vertex inside *S*. The *domination number* of a graph is the smallest number of vertices in a dominating set.

This search option finds all possible dominating sets that have the minimum number of vertices.

Example:



Note that each of the black vertices is adjacent to at least one of the red vertices.
The red vertices represent one of the two minimum dominating sets of cardinality 6.

The bottom info-strip shows the number of minimum dominating sets, and the <…> buttons on the right of the info-strip allow you to navigate through them.
As the search progresses, *Graph Theorist* consistently displays the most recently found minimum dominating set. If you choose to stop the search before completion, the info-strip displays what it has found so far.

Minimum Independent Dominating Sets

An independent dominating set *S* is a dominating set of vertices that is also independent; i.e., every vertex not in *S* is adjacent to something that is in *S* and no two vertices of *S* are adjacent.

Note that in the graph above, no two red vertices are adjacent (so they form an independent set) and every black vertex is adjacent to at least one of the red vertices (so that red vertices form a dominating set).

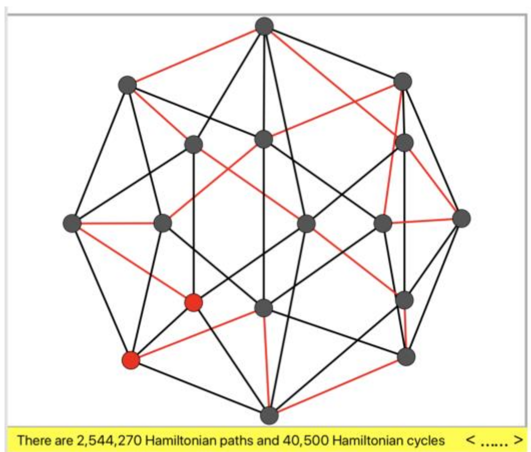
Shortest Path Between Two Vertices

Select any two vertices and then the Search option Distance/Shortest Path will determine the distance between those two vertices and provide an example shortest path. An alert will show if there is no path between the vertices.

Longest Paths

You can search for the paths of greatest length. If there is a path that passes through every vertex, then obviously it is a longest path and such a path is called a *Hamiltonian path*. Similarly, a cycle that passes through all the vertices is called a *Hamiltonian cycle*. See https://en.wikipedia.org/wiki/Hamiltonian_path For more details. The search alerts you whenever it finds a Hamiltonian path or cycle. And if the search runs to completion, up to 1000 of the Hamiltonian paths and cycles are available for display.

As you can see from the figure below, the initial and end vertices of the path appear as red vertices and the edges of the path are red.



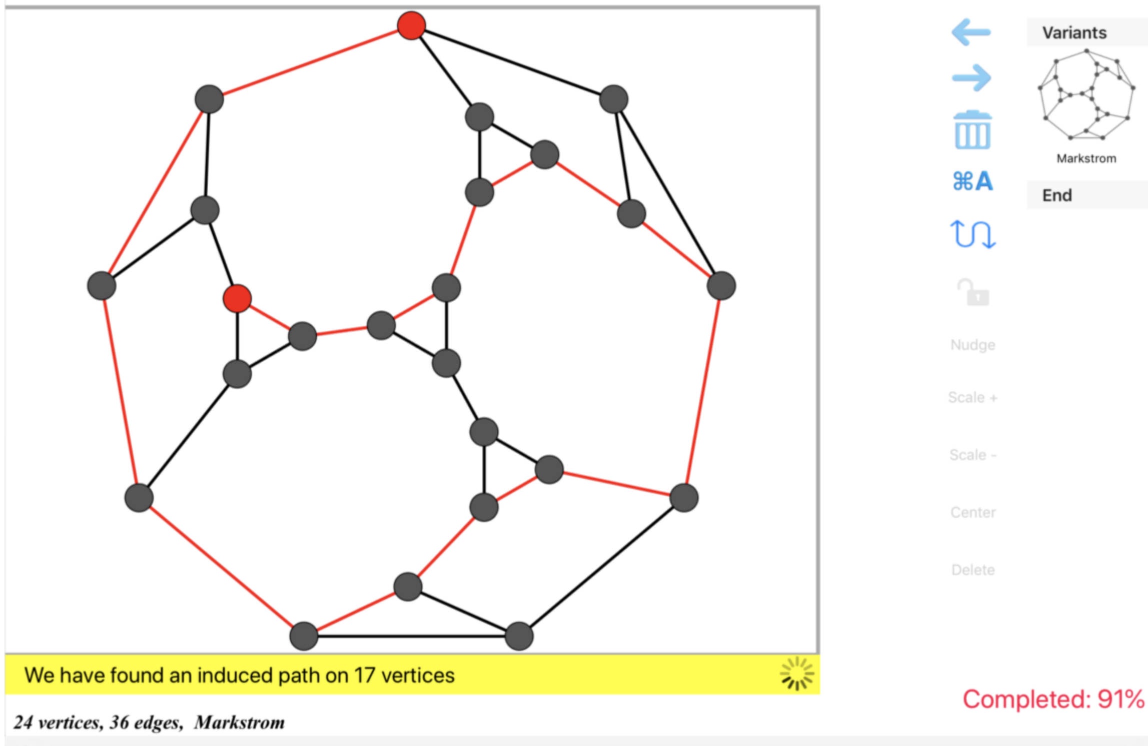
The search shows one of the 2,544,270 Hamiltonian paths in this graph.

If there is a Hamiltonian cycle, then the search displays that as soon as one is found, and it remains on the display until the search terminates or you abort the search. As with all searches, the <….> buttons to the right let you navigate through all the maximum paths (and Hamiltonian cycles should they exist).

> **Note:** Paths are considered undirected here i.e., a path *P* from *a* to *b*, is considered identical to the reversed path from *b* to *a*. However, you may choose to change this in the *General Settings* so that paths are considered directed. If you change this, then the search would report twice as many longest paths as before.

Longest Induced Paths

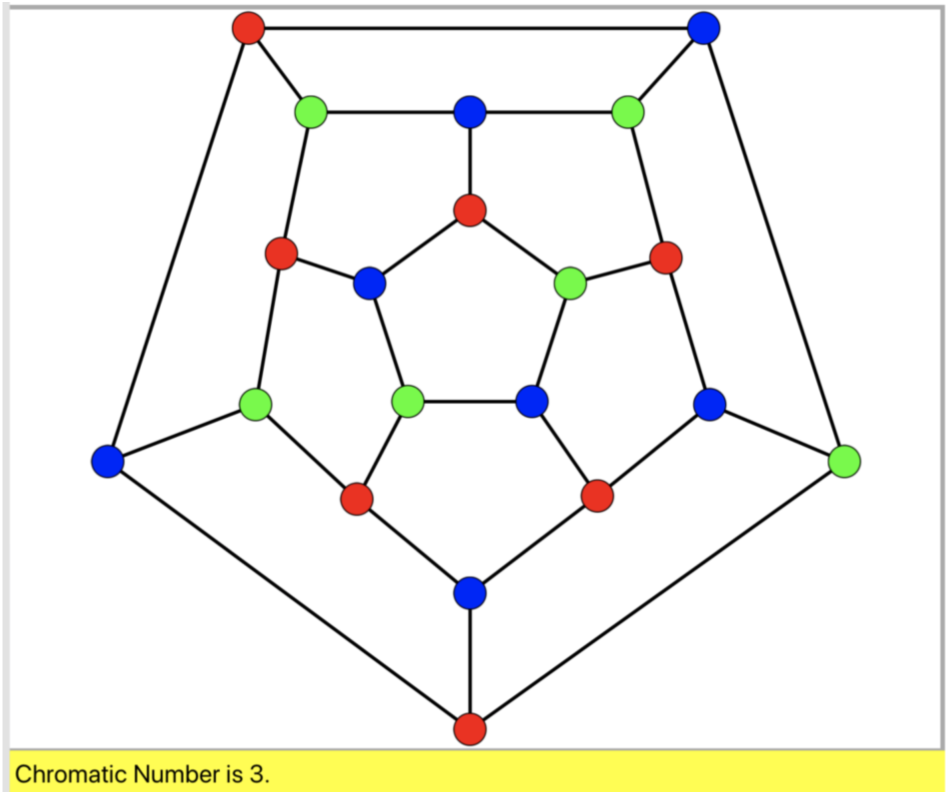
This does essentially what the search for longest paths does, only it looks for *induced* paths. A path is *induced* if none of its vertices are adjacent to vertices of the path other than its neighbors in the path. The image below shows such a search in progress. See https://en.wikipedia.org/wiki/Induced_subgraph for more about induced subgraphs in general.



This search shows the longest induced path it has found so far.
Once the search finishes, you'll discover 216 induced paths containing 17 vertices.

Greedy Coloring

The *chromatic number* of a graph is the minimum number of colors needed to color the vertices in such a way that no two adjacent vertices have the same color. See https://en.wikipedia.org/wiki/Graph_coloring#Chromatic_number for more details. *Graph Theorist* does not generally attempt, at this time, to determine the chromatic number of the graph, but it does estimate this value. *Graph Theorist* (very quickly) performs a million random greedy colorings and presents you with the best result it found. It also provides an actual coloring using that number of colorings. In special circumstances, *Graph Theorist* returns the actual chromatic number.



The Dodecahedron has chromatic number 3.

Connectivity

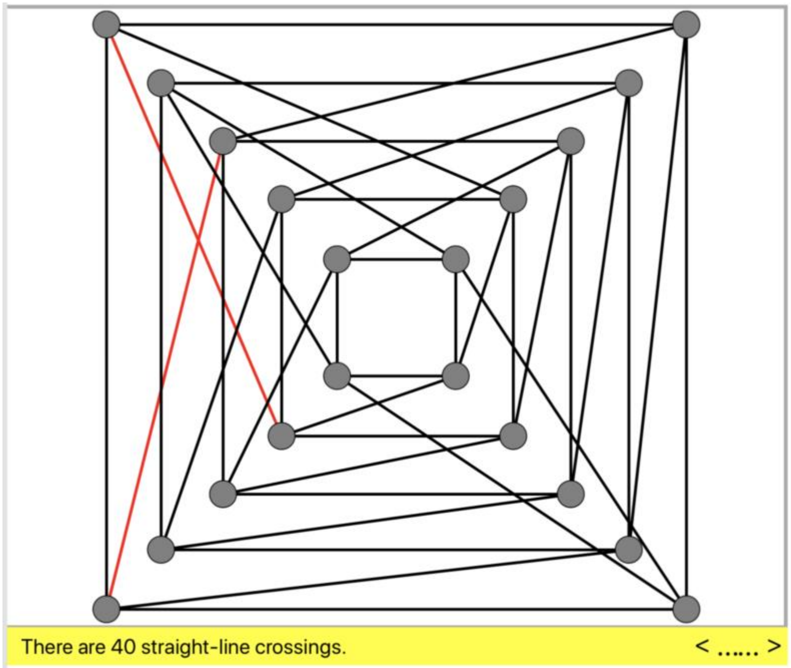
The *connectivity* of a connected graph is the smallest number of vertices that you must remove to disconnect the graph. This search determines the connectivity of the graph. for more information about connected graphs see https://mathworld.wolfram.com/ConnectedGraph.html

Straight Line Crossings

Graph Theorist determines all pairs of edges that cross one another and shows the number of such crossings as well as allowing you to see each such pair using the navigation buttons on the info-strip.

Note: *Graph Theorist* allows you to draw curved edges, however the search ignores any crossing in which at least one edge is curved.

Example: Note that in the graph below there are 40 crossings.



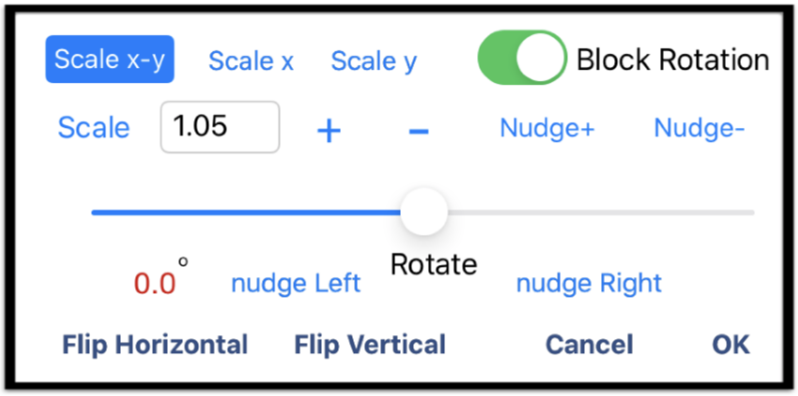
Weiner Index

This is not a real search, but rather a computation. See
https://en.wikipedia.org/wiki/Wiener_index

Choosing this search option on a connected graph produces a simple window that contains the result of this computation.

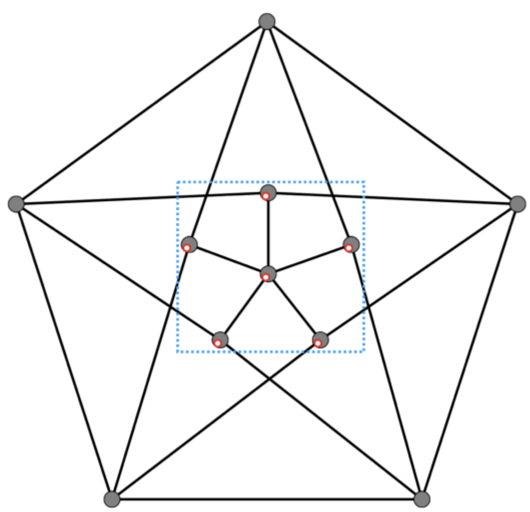
The Transform Inspector

After selecting a collection of vertices, tapping on the *Transform* button brings up the window below. Using the controls in this window, you can scale, rotate, or flip the set of selected vertices in several ways.



The General Idea

Graph Theorist bases most its transformations on the *Selection Rectangle*, which is the smallest rectangle that encloses all the selected vertices. This rectangle is not visible on the screen, but the transformations are actually applied to it and the selected vertices go along for the ride.



The doted blue box bounds the selected vertices.
All transformations are applied to this box.
This box is invisible to the user.

Scaling

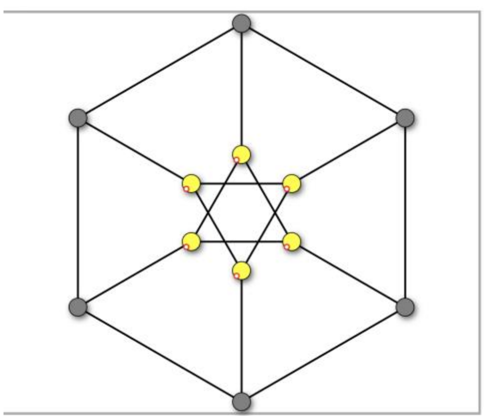
You can choose to scale the set of selected vertices in three ways:

- Both vertically and horizontally (Scale *x*-*y*).
- Horizontally only (Scale *x*).
- Vertically Only (Scale *y*).

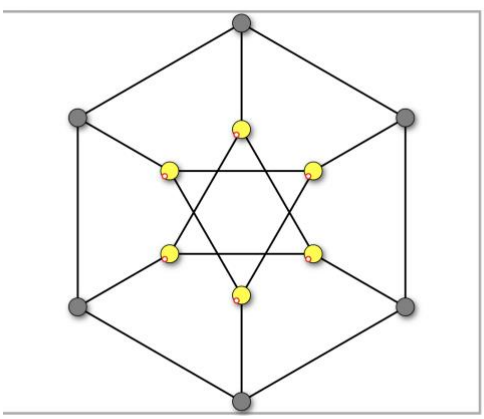
First click on the scale type on the top row, and then enter the desired scaling constant – as a convenience, you can modify the current scaling value up/down by .05 by tapping wither the + or – buttons next to the scaling value.

Then after, selecting the scale value, tap the Scale button to actually perform the scaling.

In each case, the Selection Rectangle is altered in one or both dimensions to produce the scaling effect.



Before



After a scale of 1.3 in both dimensions

If the scale is not exactly what you want, you can modify it slightly by tapping the *nudge+* button (respectively the *nudge–* button) to slightly increase (respectively decrease) the scale.

Rotation

You may rotate the set of selected vertices by dragging the bottom slider to the right for positive rotation and to the left for negative rotation. The rotation is measured in degrees and the number of degrees in the rotation is shown in red just below the slider and to the right. Degrees are measured to tenths of a degree.

Two Types of Rotation

There are actually two types of rotation. The first, a *block rotation*, rotates the selection rectangle and the associated vertices go along for the ride. The second, a *vortex rotation* rotates all the selected vertices *about the last selected vertex*. The next two sections describe these two types of rotation in more detail.

Tip: You can change the vertex around which the others rotate by simply unselecting the vertex you want and then reselecting it – this makes it the last selected vertex.

Block Rotation

With this option, the vertices are actually rotated about the center of the selection rectangle (as shown above, the rotation is actually applied to the selection rectangle itself and the selected vertices are carried along with it).

It may be difficult to get the rotation exactly as you want it by dragging the slider. However, if you get reasonably close to the degree of rotation that you want, you can tap the *nudge right/left* buttons to obtain the accuracy that, up to half a degree, you desire.
You can see the number of degrees in the rotation in red on the left of the transformation inspector.

Vortex Rotation

In this case the selected vertices are rotated about the last selected vertex rather than about the center of the bounding rectangle.

As with the block rotation, it may be difficult to get the rotation exactly as you want it by dragging the slider. However, if you get reasonably close to the degree of rotation that you want, you can tap the *nudge right/left* buttons to obtain the accuracy you desire.

Flipping

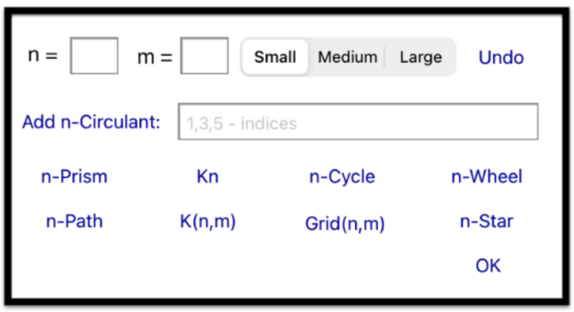
You can also transform the set of selected vertices by flipping them vertically or horizontally.

The vertical flip is about the horizontal line through the selection rectangle's center, and correspondingly, a horizontal flip is about the vertical line through the selection rectangle's center.

You can get a better idea of how the transforms work by playing with them . After performing whatever transforms you like, tap *OK* to keep the effect or *Cancel* to undo all current transforms, and revert to the original graph. *All Transforms can be undone after the Transform Window closes.*

Extra Add-ons

Tapping the *Add-ons* button at the bottom of the display brings up the dialog box below:



The Add-ons Box

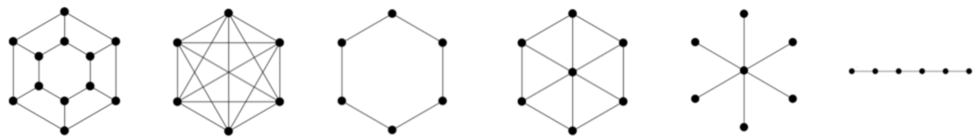
This provides you with the option to add many special graph types to the current graph. When you append any such graph, the vertices appear as selected vertices so that you may immediately modify them as you see fit, for example by applying an appropriate transform or setting vertex properties with the *Vertex Inspector*.

In addition, because it is likely that you want to drag the new subgraph to a special location, adding an add-on also automatically unlocks the graph.

For each choice of graph, you may choose to have the graph load as a *Small* copy in the case where you plan on associating it as part of a larger graph – after dragging to where you want it, you can then scale the graph up to an appropriate size,.. You may want to choose *Large* if it is unlikely to overlap existing parts of the current graph. You may choose *Medium* for a size in-between these two options. If you like you may execute an undo from the Add-Ons Box.

The special graphs that you can load are:

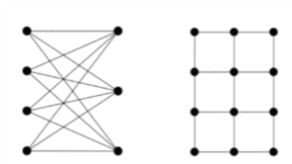
n-Prism	Kn	n-Cycle	n-Wheel
n-Path	K(n,m)	Grid(n,m)	n-Star



6-prism **K6** **6-cycle** **6-wheel** **6-star** **6-path**

Example: For *n* = 6

To load any of the graphs above, enter 6 into the "*n* =" text field and tap the appropriate button



K(4,3) Grid(4,3)

To load either of the graphs above, enter 4 into the "*n* =" text field, enter 3 into the "*m* =" text field and tap the appropriate button.

Circulants

The last kind of graph that you can add using the save/load dialog box is a *circulant*.

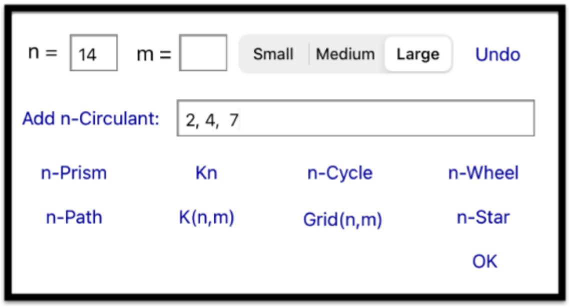
What is an n-Circulant?

An *n-circulant* with indices $a_1, a_2, ¼, a_k$ consists of *n* independent vertices $\{v_1, v_2, ¼, v_n\}$ (i.e., no two of the vertices are adjacent) arranged around a circle and for each index a_j, add all the edges from each vertex v_i to the vertex v_{i+a_j} where the subscripts are reduced modulo *n*. So in the example below, for instance, vertex #2 is adjacent to vertices #4, 6, 9 (i.e., 2+2, 2+4, and 2+7); and vertex #11 is adjacent to vertices #13, 1, 4 (because 11 + 4 = 15 which is 1 modulo 7, and 11 + 7 = 18 which is 4 modulo 7).

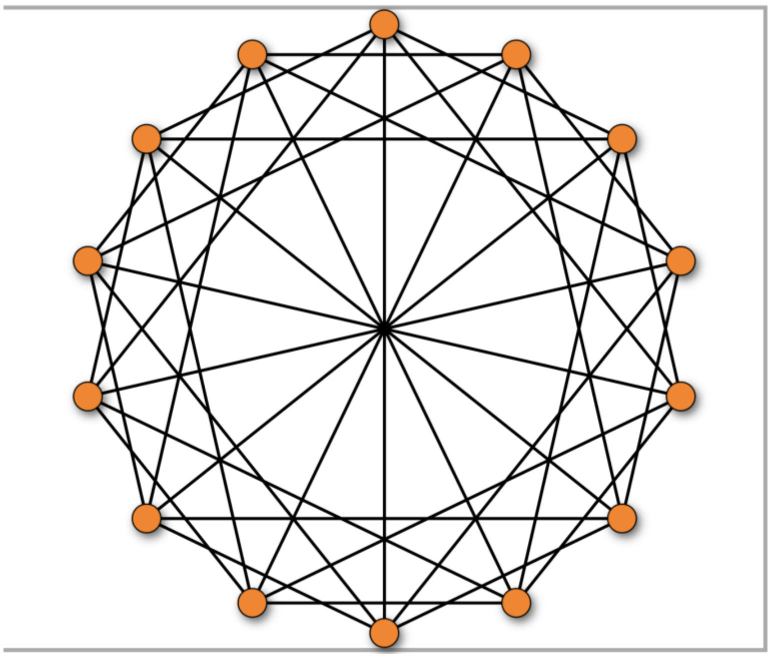
For more information on circulants see:
http://mathworld.wolfram.com/CirculantGraph.html.

To load the 14-circulant below, enter 14 into the "*n* = " text box and enter "2, 4, 7" into the indices text box as shown below. Then *tap* the *Add n-Circulant* button.

Example:



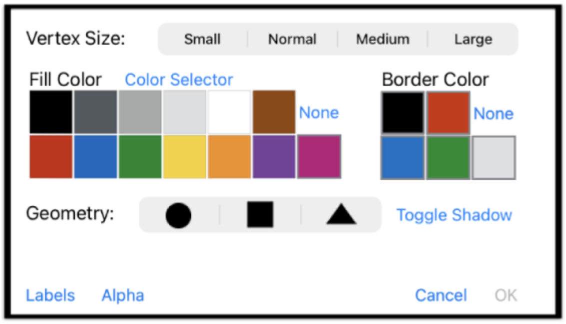
How to add a 14-circulant with indices 2,4,7



Circulant with n = 14, and indices 2, 4, 7.

The Vertex/Label Inspector

Tapping the *Vertex/Labels* button at the bottom of the display brings up the *Vertex Inspector*. The actions in the Vertex Inspector apply only to the currently selected vertices. However, you can select and deselect vertices although the inspector is open. See xxx for a video tutorial on using the *Vertex Inspector*.



The Vertex Inspector

When the *Vertex Inspector* is open you can use it to change the color, size, and type of the selected vertices as well as set or modify the labels for the vertices. To adjust these properties, tap the appropriate control in the inspector's window and the effect takes place at once on all the selected vertices.

The Inspector Window Can Be Made Transparent

You can drag the inspector window around the display. Even so, it's possible that the window could block your view of the vertices you're editing. If this happens, then tap the *Alpha* button the view becomes transparent; tap *Alpha* again to restore the view to opaque.

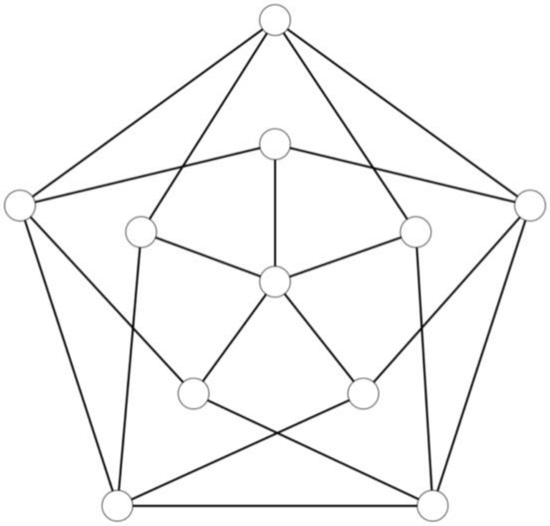
Vertex Properties

It's worth a moment's discussion to explain some of the vertex properties.
The size of a vertex is the radius of its circumscribed circle.

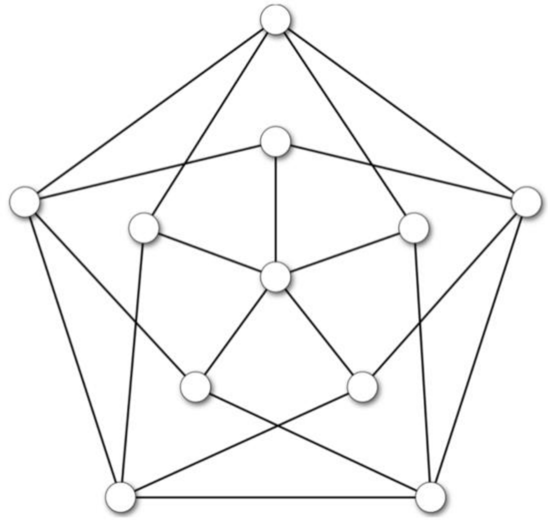
- There are four possible sizes, *small, normal, medium*, and *large*.
- You may also choose from three shapes: circle, square, and triangle.
- You may choose from among 13 fill colors – or choose *none at all* to make the vertex transparent.
- You may tap *the Color Selector* button to bring up the *color selection window*.
- You may choose to have a border color as well as a fill color.

Toggle Shadow

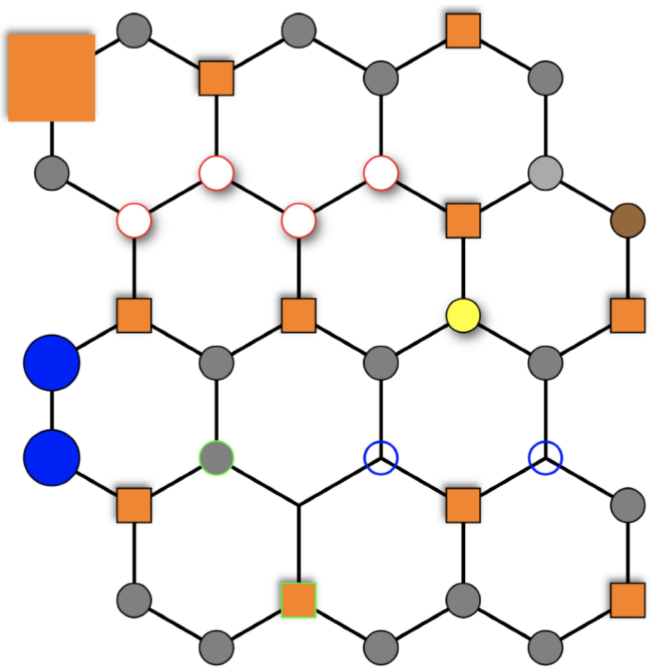
Tapping the Toggle Shadow button shows/hides a shadow effect on the selected vertices. The shadow effect creates a somewhat 3D look to the affected vertices.



No Shadow



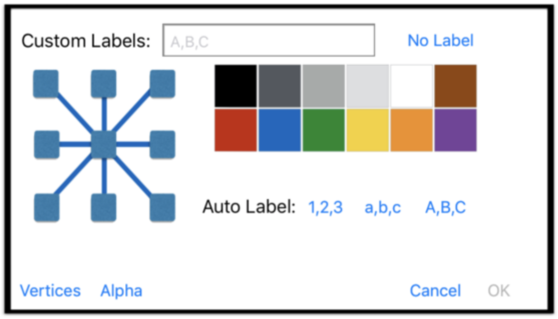
Shadow



Illustrating several vertex options for the same graph

Vertex Labels

The Vertex inspector also lets you modify, add, or edit the labels for the vertices. Tap the *Labels* button to switch to this option.



The Labels option in the Vertex Inspector

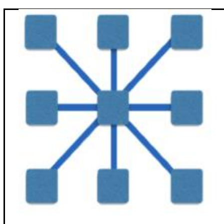
By default, vertex labels do not appear, and if you choose to make them visible then they can be automatically assigned numerical values, or alphabetical values (determined by the current settings) *based upon the order in which they were created*. You can set the color of labels by tapping the appropriate color swatch in the inspector window.

Warning: Be careful to not select white as the vertex color and choose an external label position – you will wonder why no labels appear.

Also, labels only appear for the selected vertices. So, if you want to automatically label all the vertices of your graph, make sure you select all vertices first.

Positioning Labels

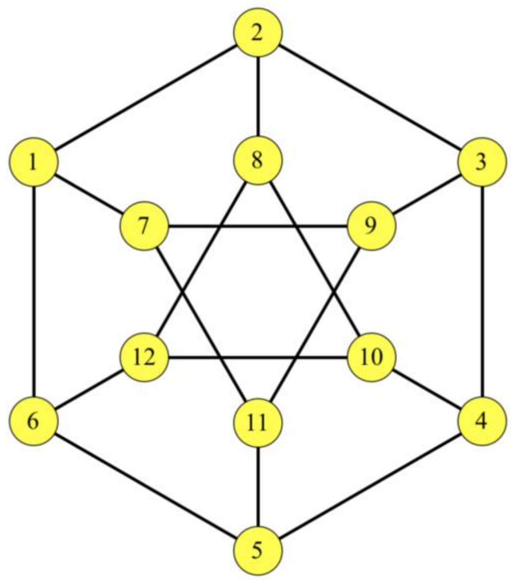
The label view of the Inspector Window has a special tool for positioning labels.



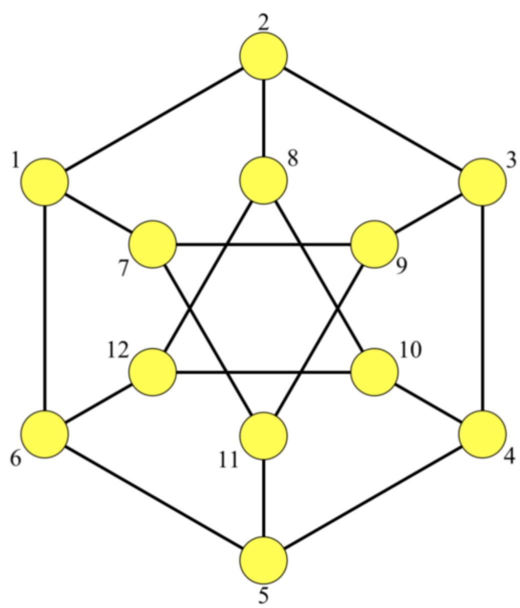
This tool gives you a great deal of flexibility in setting the position of the labels relative to the corresponding vertex.

Tap the center square to set the label as center, the top right square to set the label just above and to the right of the vertex, etc.

To see how this works, choose any graph from the custom database (i.e., by tapping the *Graph* button at the top left of the display), tap the *Select All* button so that you can modify all the vertices at once, choose *Auto Label 123*, and then in the labels display tap the center button. Make sure the label color contrasts with the color of the vertices or you won't see anything!



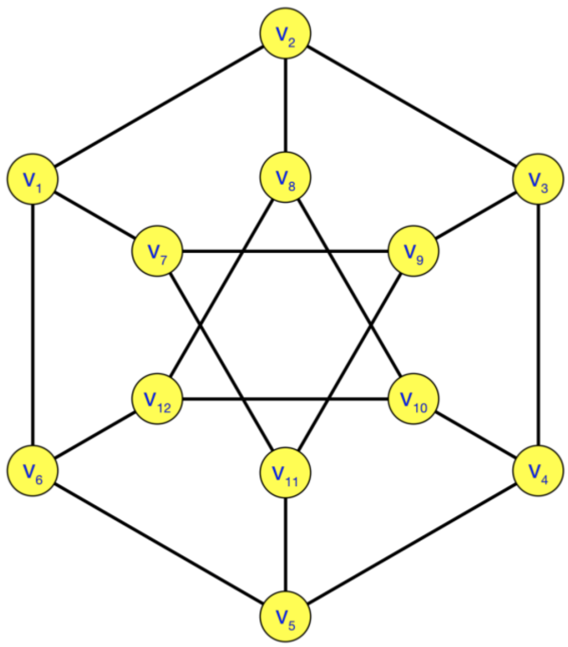
A graph with the default labels in the center of each vertex



The same graph above, with the labels outside the vertices

Auto-Subscripted/Superscripted Labels

If in the label Inspector you select the subscript or superscript option at the bottom of the screen, then rather than seeing the selected vertices labeled automatically with numbers, the result is subscripted labels such as what you see in the graph below.

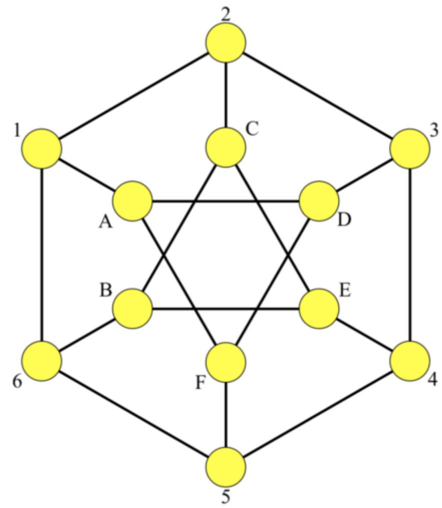


Auto Labels with Subscripts

Warning: Various operations on a graph can result in the vertex labels being off by a bit. For example, identifying vertices will have this effect. Graph Theorist does not attempt to automatically adjust labels since it does not know what you might intend for them to be. However, you may select all the vertices and automatically relabel them if you like.

Custom Labels

You can easily assign custom labels to your vertices. First select the vertices you want to modify – and *the order of selection matters*! Suppose, for example, that you want to label the inside six vertices of the graph above as *A, B, C, D, E, F*. Then select the vertex for label *A*, first, then the vertex for label *B*, etc. Then choose the *Vertex Inspector* and enter the labels separated by commas and press enter.



The same graph as before with custom labels for the interior six vertices

The Transparent Property of the Vertex Inspector

Sometimes the *Vertex Inspector* covers up the vertices that you want to modify. *Graph Theorist* has you covered – by uncovering the inspector.

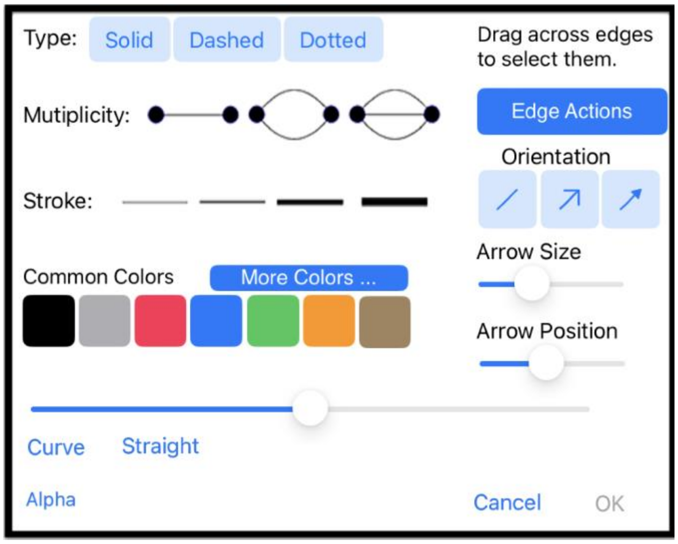
Tap the *alpha* button in the *Vertex Inspector* window and the view becomes partially transparent, allowing you to see the effects of your actions. Tap *alpha* again, and the view becomes again opaque.

The Edge Inspector

You can modify the edges of a graph as well as the vertices. To select an edge, simply select the two vertices that it joins. If you select a collection of vertices, then any actions in the *Edge Inspector* apply to *all the edges that join any two of them*.

The Basics

Bring up the *Edge Inspector* by tapping on the *Edge Inspector* button at the bottom of the display.



The Edge Inspector Window

From the *Edge Inspector*, you can change an edge's:

- Color (any of 12 colors) – tap any color in the color array and the currently selected edges are colored appropriately. Or choose the option *More Colors* to choose from among many more color options.
- Stroke (i.e., thickness). There are four levels of thickness.
- Type (either solid, dashed, or dotted).
- Orientation (either none, open arrow, or closed arrow).
  Note that the orientation of an edge is based upon how it was drawn.
  If the edge was drawn from *A* to *B* then the edge is oriented from *A* to *B*.
- Multiplicity (single, dual-edge, or tri-edge).
- You may also choose to subdivide selected edges.

Selecting Edges

To select edges, you must first open the Edge Inspector. You select edges by dragging across them while the Edge Inspector is active. As you drag across an edge, a green line and a small red dot appears on each selected edge. To deselect a selected edge, drag across it again.

Curved Edges

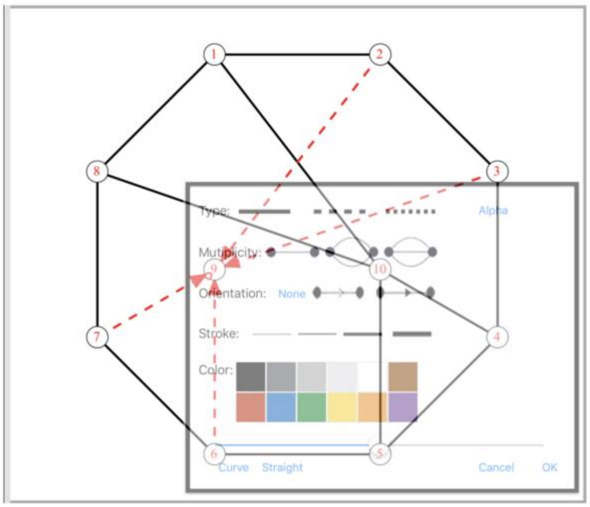
You can also choose to make the edge curved and adjust the curvature. To curve a selected edge (or group of selected edges), tap the *Curve* button, and adjust the slider as you see fit.

Note: Sometimes it may be difficult to select an edge because it is too close to other edges or because it is curved. To address this issue, Graph Theorist provides another option to select the vertices of the edge. That is, if you select exactly two adjacent vertices, then the Edge Inspector includes their edge as a selected edge.

The Transparent Property of the Edge Inspector

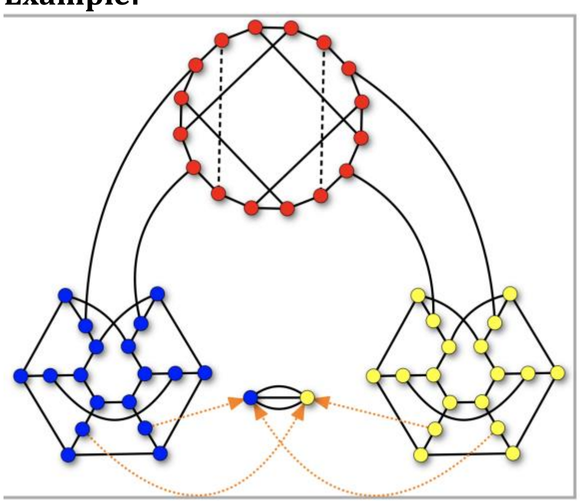
Sometimes the edge inspector covers up the edges that you want to modify. *Graph Theorist* has you covered – by uncovering the inspector.

Tap the *alpha* button in the *Edge Inspector* window and the view becomes partially transparent, allowing you to see the effects of your actions.



Tapping *Alpha* makes the Inspector partially transparent.

Example:



An illustration of *some* of the edge formatting options

Note: When you drag a vertex, or collection of vertices, all the edge properties such as curvature and orientation adjust appropriately.

Known Issues

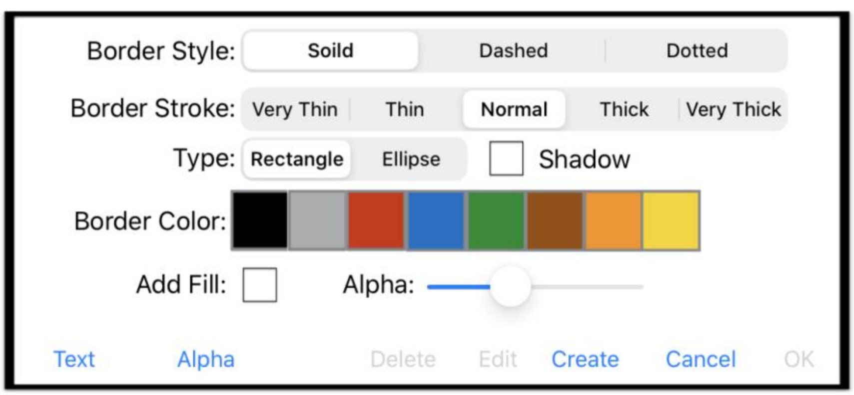
Currently, edge properties are limited in a few ways.

- Edges cannot have labels (but adornments can be used as edge labels).
- Multiple edges cannot have an orientation.
- Loops are not allowed.

Adornments

Many graph depictions have external items such as rectangles, ellipses, and special labels. *Graph Theorist* provides access to many such options, and many more will come in future versions. These options are called *Adornments*.

To add an adornment to a graph, choose the *Adornments…* item from the *Options* drop-down menu. This results in the following window called the *Adornments Inspector.*

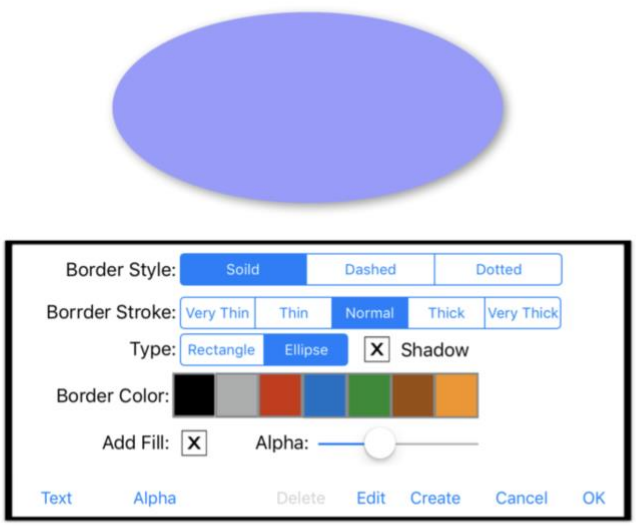


Geometric Adornments

To create a shape (i.e., rectangle or ellipse) adornment, tap the *create* button. You can then (or even before tapping *create*, if you prefer) choose the type (rectangle or ellipse), the border color, the thickness of the border, and the style. If you create a filled shape, then you can also add a level of transparency by dragging the Alpha Slider until you get the level you want.

You can drag the figure to wherever you like and stretch or compress it using the red handles on any side of the figure.

After tapping create, the create button changes to *Done*.



An Ellipse with a blue fill and shadow

An adornment is initially placed in the upper left of the canvas.

Important: The figure will not be finalized until you tap the *Done* button.
Note that the *OK* button is dimmed until you tap *Done.*

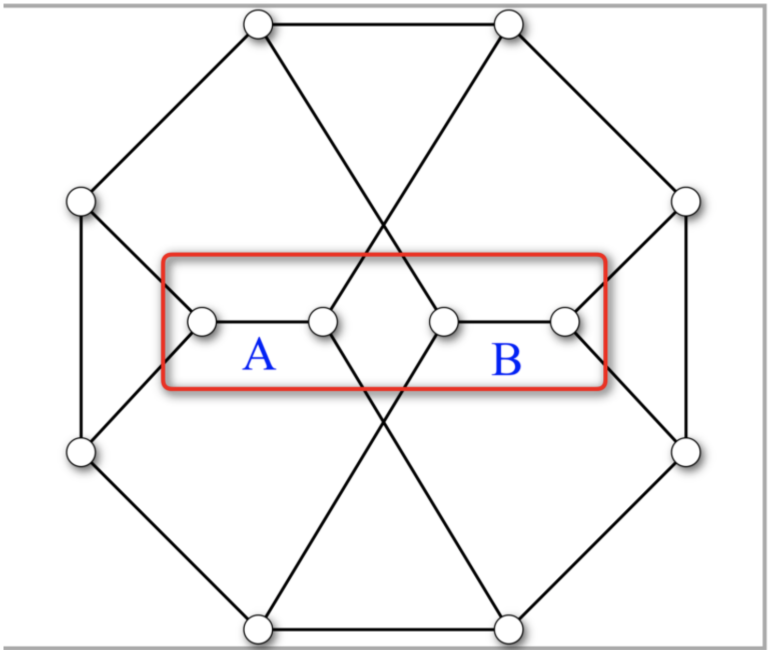
Adjusting Adornment Properties

After *Graph Theorist* creates the figure for you, you can still modify its properties and most importantly you can stretch or compress it as you see fit.
Use the red handles to scale the figure as you like.

You probably don't want the figure to remain in its original position. You can tap anywhere inside the figure and drag it to wherever you like.

But be warned that the figure is not be finalized until you tap the *Done* button. Note that the *OK* button is dimmed until you tap *Done*.

You can create more adornments in any single session. But remember that **you must tap *Done* each time before the adornment is permanently added to the canvas**.



A Rectangle Adornment with two Text Adornments

Text Adornments

Tap the *Text* button at the bottom left of the *Adornments Inspector* and the following view greets you. Note that the *Text* button changes to *Geometry* allowing you to return to the Geometry adornment view.

In this case, tap inside the *Text* box and type whatever label you'd like to create. Choose a size, a color, and choose if you'd like the text italic, bold, both, or neither. Then tap create and drag the label to wherever you'd like to place it.

You may use Unicode characters (including emoticons) if you like.

Just as with geometry adornments, **you must tap** *Done* before the adornment is officially created. You can create as many adornments of either type as you like in one session.

After creating a text adornment, you can move it anywhere you like.
Also, you can rotate the adornment using the indicated slider.

Note: If you have rotated a text adornment, then it may be necessary to tap the adornment near its center in order to move it.

With the few caveats mentioned below, all attributes of the text adornments are carried over to corresponding TikZ code.

First, although you can use Unicode characters in a text adornment, these will not appear in TikZ. Also, there may be slight differences in the location of text adornments in TikZ and you may want to modify their locations inside the TikZ code.

Editing Adornments

Adornments are not set in stone. Tapping on the *Edit* button allows you to edit the currently existing adornments.

Once you tap the Edit button, you can tap any existing adornment, and both move it about, scale it, or modify its attributes.

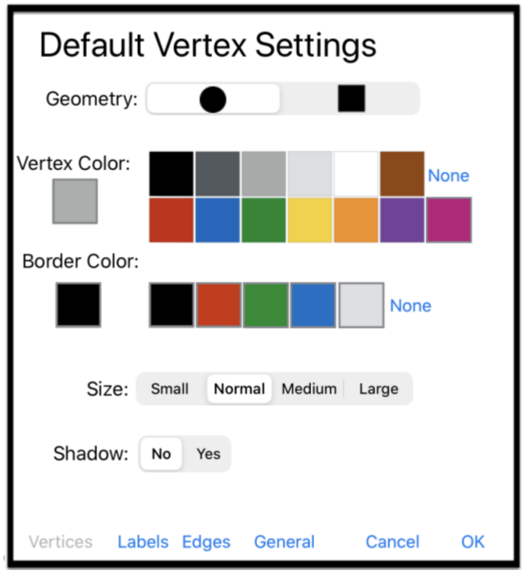
Once you tap Edit, you can edit as much as you like, but you cannot create any new adornments without tapping *OK* (or *Cancel*), and then reopening the *Adornment Inspector*.

While in edit mode, tapping on an adornment, sets the *Adornment Inspector's* properties to reflect those of the selected adornment.

Note that if you are currently editing in the Geometry view of the Adornment Inspector, then tapping on a Text adornment opens the Text view with all the adornment's properties selected (and conversely)/

Default Settings

Tapping on the gear icon on the upper left of the screen, brings up the *Default Settings Inspector.*

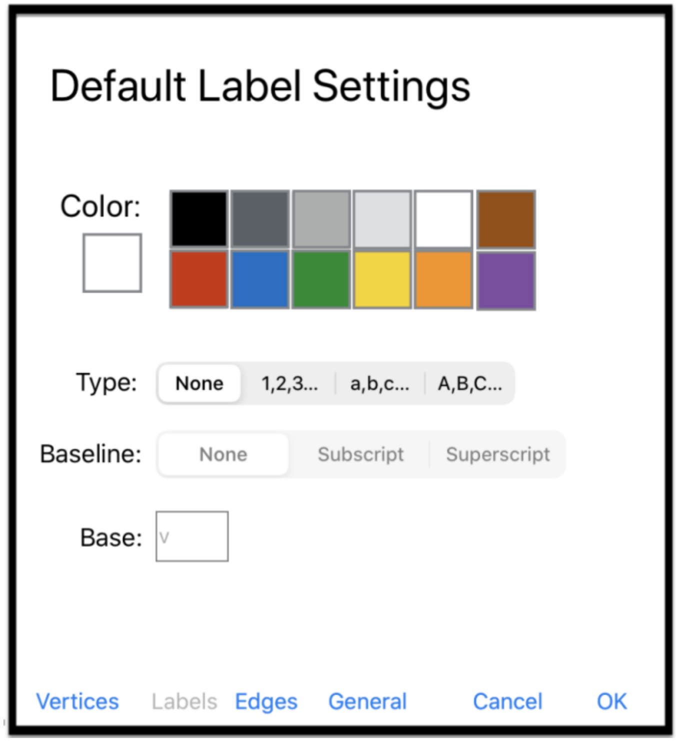


The Vertex Portion of the Settings Window

These settings determine the default properties for new vertices, edges, and labels. Thus, if you select red, and normal size for vertices, then all vertices created after that default to normal size, red vertices. Of course, you can always override these defaults in the Vertex Inspector.

Also, graphs loaded from the classical database as well as variants from the classical database load with properties determined by the global settings.

However, graphs loaded from the Custom database (and all user-created variants) load with the properties set at the time they were created.

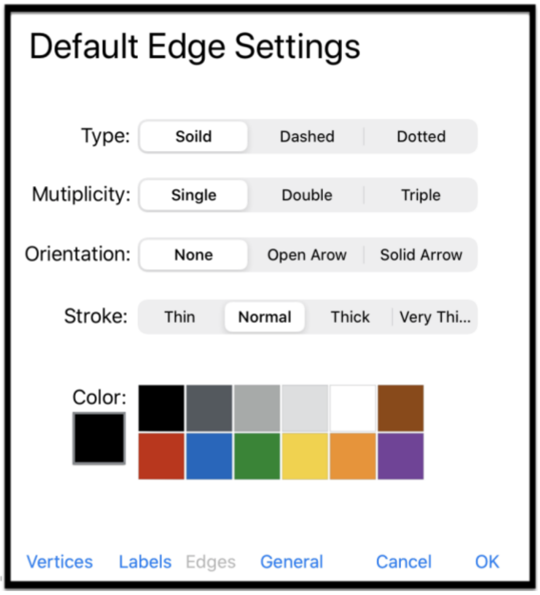


The Labels Portion of the Settings Window

Here you can choose the default color, and type for labels. Also, here you can choose the default base value for subscripts and superscripts. If you do nothing the default value is *v*, otherwise it is whatever you enter in the Base field.

Note that there is no default setting for the position of a label – this automatically defaults to the center because there is no way to know what other position of a label would be appropriate.

For this reason, when choosing "Set Graph to Defaults" from the Drawing menu, all label properties are set to the default values unless a label has a non-central value set. In this case, the position of a label is left unchanged.

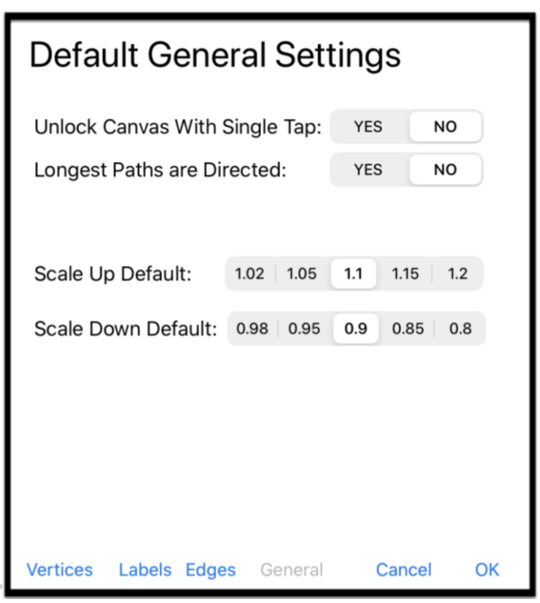


The Edge Portion of the General Settings Window

You can choose the default values of edge type, multiplicity, color, and orientation. For instance, if you choose red color and solid arrow orientation, then every edge you draw has these properties. Of course, you can always override these defaults in the Edge Inspector.

General Default Settings

There are other optional settings available as well. These add convenience and customization to *Graph Theorist*.



The General Settings Portion

You can choose the scale up and scale down default settings. The default scaling up value is 1.1, and the default scaling down value so 0.9. These default values are used by the scale up/down buttons on the side of the canvas in landscape mode (and on the bottom in portrait mode).

You may choose to determine if searches for longest paths (and induced paths) are directed or undirected. If you choose Paths are Directed, then a path from vertex *v* to vertex *u* is considered different than the same path running from *u* to *v*. Paths are undirected by default.

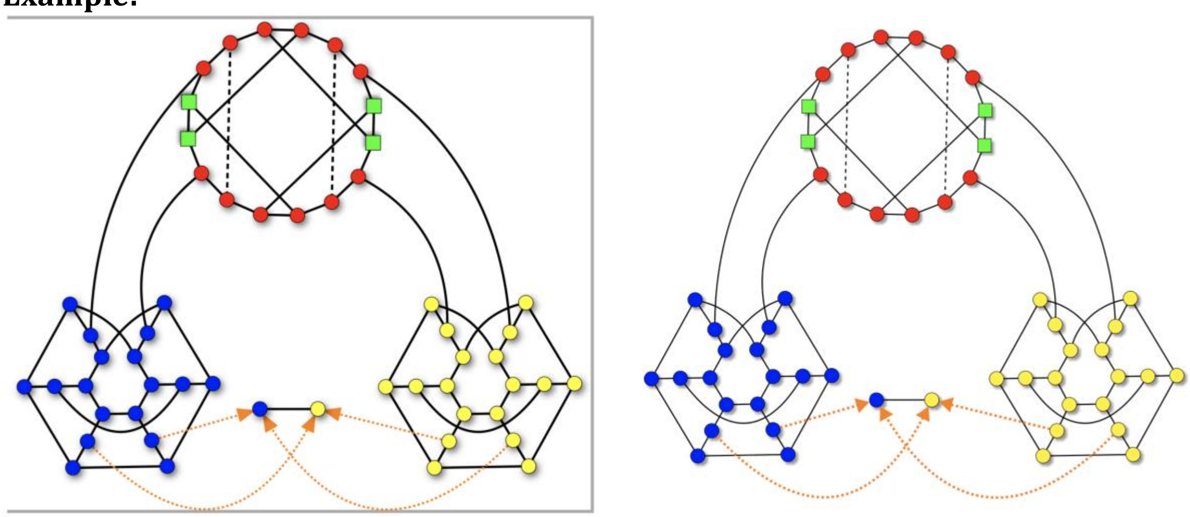
Also, you may choose to have single taps on the canvas unlock the canvas. This can be a great convenience in many circumstances so that you don't have to tap the lock button when you're doing something in the middle of the canvas. This option is off by default. But when you turn it on then a single tap anywhere inside the canvas (but not on a vertex) toggles the state of the lock icon.

TikZ Output and LaTeX

Perhaps you'd like to include an image of your graph in a LaTeX document. One of the most popular ways of doing this is with a TikZ file.

Graph Theorist is cable of producing a TikZ file that very closely replicates any graph you've drawn. As of this version of *Graph Theorist*, the only drawing feature that does not have a counterpart in the resulting TikZ file are the shadow effects on geometric adornments.

Example:



A graph and its equivalent in LaTeX on the right (from TeXShop on a Macintosh

By choosing the Export TikZ item in the Options Menu, you can save the appropriate TikZ image code embedded in a document. You can access this TikZ file from your desktop computer so that you can use it with your TeX editing software. There are also TeX programs for the iPad, and you may load saved files directly in them as well.

Although *Graph Theorist* generally does a fine job of mimicking the appearance of the graph in TikZ code, you can tweak the TikZ code to suit your purposes in case things do not look exactly as you want them.

Copy TikZ Code

Rather than save the TikZ code to an external file, you may alternatively choose to copy just the TikZ Picture code and then paste it into a TeX program on the iPad.

Choose the *Copy TikZ Code* option from the *Options* menu. In this case it may be necessary to add appropriate libraries to the LaTeX document.

Note: There are a few caveats.
- Shaded rectangles and ellipses show as normal rectangles and ellipses in TikZ.
- Fully transparent vertices in Graph Theorist show as white vertices in TikZ.
- There may be some (usually small) degree of distinction between the two formats.
- Unicode characters display fine on the iPad but may not translate to the TikZ file.

You can of course modify the TikZ code as you see fit, perhaps adding features that are not currently in *Graph Theorist*.

Support and Email

To use email simply tap the *Support* button from the top and enter the email address for the recipient.

The default email address is to *Graph Theorist*'s support – please use it whenever you have a question, issue, or comment – we want to hear from you, and *we respond promptly*!

Or you can use the email option to email anyone you want and include an image of a graph if you like.

Graph Images

Sharing Graph Images

Choose *Share Graph Image* from the *Options* menu and follow the resulting window messages to share the image of the current graph via Messages, Email, Facebook, or Twitter.

This option also allows you to save the image of the current graph to Photos.