

## Other tools: `xfig`

`xfig` is a menu-driven tool that allows a user to interactively create and manipulate figures. Features include:

- ☞ Drawing lines, ellipses, splines, polygons, rectangles, arcs, and arrows.
- ☞ Entering text and arrows.
- ☞ Components can be scaled, moved, copied,



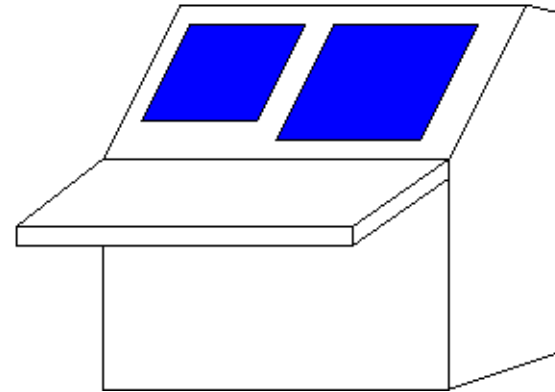
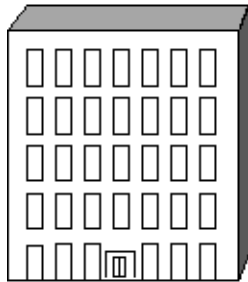
deleted, flipped, rotated, and aggregated into larger components.

- ☞ A variety of line styles are supported.
- ☞ Libraries of icons are also supported.
- ☞ Items can also be floodfilled with colors or patterns.

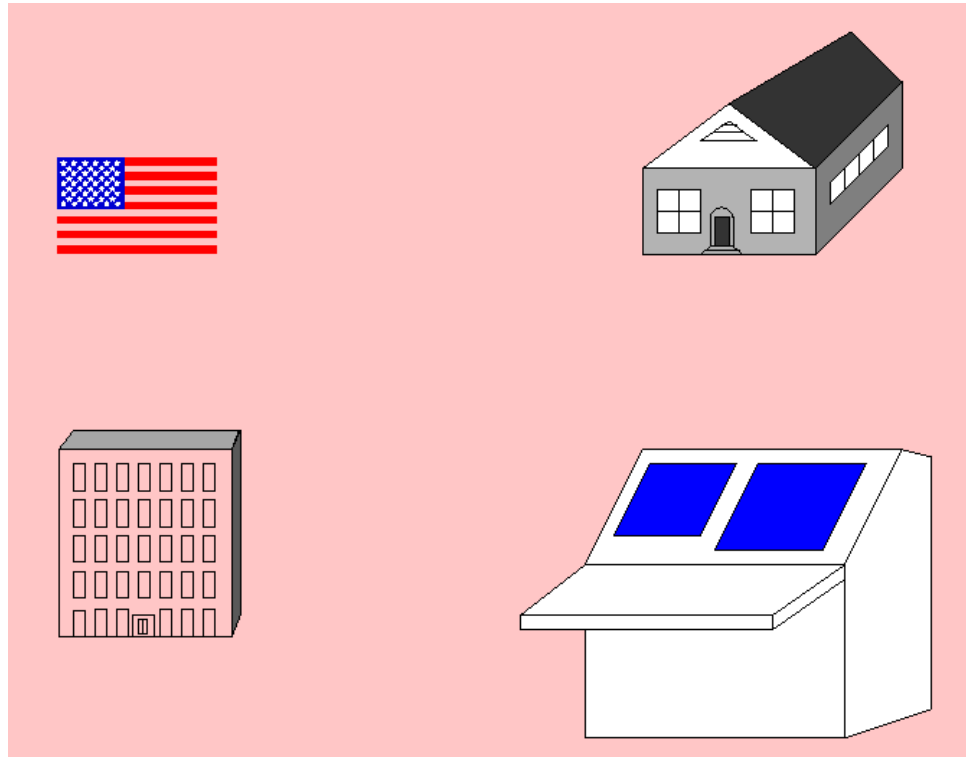


# xfig **example**





# xfig example



## Other `xfig` capabilities

- ☞ Can export into different formats (default is fig format, but in this slide presentation, the fig files were exported as png files), including  $\text{\LaTeX}$  picture format, MetaPost, MetaFont, gif, encapsulated PostScript, Portable Document Format, png, and jpeg.
- ☞ Can use a grid to control placement (“snap to grid”.)
- ☞ Can change the characteristics of existing objects.



☞ Can perform group operations on aggregations of objects.



## xv, gimp, krita **and** inkscape

There are a number of programs to display or manipulate images. The program `xv` is one of the oldest; it has steadily gained features over the years.

Another is the `gimp`, which has as its strongest point manipulation, although many people have criticized its interface.

Recently `krita` has become quite popular. Like `gimp`, it also has its strongest manipulation of images.





A different kind of program is `inkscape`, which while it can take in an image graphic, its strong point is creating scalable vector graphics (SVG).



# spell **and** ispell

The `spell` utility will check a file for spelling problems. It is usually just a script pointing to `aspell/ispell` running in batch mode.

The `aspell` program is a replacement from GNU for `ispell`. Its default mode is interactive. `aspell` is very featureful, and interfaces well with emacs.



# Printing control with `lpr/lprm/lpq`

☞ `lpr` – The traditional BSD method of queuing print items to printers. Some popular options are:

```
-#NUM          a number of copies
-PQUEUE        specify a print queue by name
-p            run a formatter over the file before its printed so that p
```

☞ `lpq [-PQUEUE]` – Lets you look at the print jobs for a given queue `QUEUE`. It gives a job number for each that is useful for deleting items with `lprm`.



👉 `lprm [-PQUEUE] [-]` – Lets you remove items from a print queue. You can either specify job numbers (determined from `lpq`), or with just “-”, which removes all of your items from a queue.



# pr

`pr` is a common formatter for print jobs that does various tasks, such as placing header/footer information such as page numbers and doublespacing.

## Common options:

```
-W NUM      set page width to NUM
-l NUM      set page length to NUM
-h HEADER   specify header rather than the default, which is the filename
-d          doublespace output
-COLUMN     multicolumn output: print with COLUMN number of columns
-w NUM      set page width to NUM for multiple column output
```



# a2ps

The program `a2ps` converts text files to PostScript. It allows you to do things such as printing multiple virtual pages on a single page.

For example:

```
a2ps --print-anyway yes -5 -o termcap.ps /etc/termcap
```

will reformat the `/etc/termcap` file to five pages per sheet.



# Common options for `a2ps`

- `-r`            landscape mode
- `-f #`            use font size #
- `-o OUT`        write output to file name OUT rather than  
printing to `''lpr''`
- `--columns N`    N columns per page
- `-#`            prints # pages per sheet of paper

