



GAMBY Programming Frequently Asked Questions

This document is a work-in-progress, and will be expanded as more questions are (frequently) asked.

Why are there three different modes (`GambyTextMode`, `GambyBlockMode`, and `GambyGraphicsMode`)?

In short: *memory*. The fancier the graphics, the more RAM it needs: `GambyTextMode` uses the least, `GambyGraphicsMode` uses the most, and `GambyBlockMode` is somewhere in between.

In more detail: Due to some technical details of the way the LCD module (and ones like it) work, the screen updates in eight 8-pixel-high stripes (called *pages* in the datasheet); to set a single pixel, you also have to set the other 7 in its stripe. Communication with the LCD module is one-way, so there is no built-in way to tell what's already there. In order to keep those other 7 pixels from changing, drawing must be done in an *offscreen buffer*, a sort of working copy of the screen's contents; the buffer (where you *can* access individual pixels) is then copied to the screen. This buffer uses memory. `GambyGraphicsMode` uses the most; `GambyTextMode`, which draws an entire stripe at a time, doesn't need an offscreen buffer and therefore uses the least RAM. `GambyBlockMode` was created as a compromise; it uses 1/4 as much RAM as `GambyGraphicsMode`, but is sufficient to do a game like Tetris™.

I'm using `GambyGraphicsMode` and nothing I draw is showing up... why not?

In `GambyGraphicsMode`, all drawing is done to the *offscreen buffer* -- a sort of working copy of the screen's contents. The changes made to the offscreen buffer are drawn to the screen when the [update\(\)](#) function is called.

In `GambyGraphicsMode`, how often should I be calling `update()`?

It depends. Anything you want be sure changes at the same time should be done between calls to `update()`; when you finally do call `update()`, all your changes will appear at once. Redrawing a large portion of the screen can take slightly more time, however, so if you need the redraw to happen faster you may want to try calling `update()` more often. Generally speaking, though, start by using `update()` less often, then add more as needed.

What's the difference between `drawBlock()` and `setBlock()` (`GambyBlockMode`)?

[drawBlock\(\)](#) draws a block to the screen immediately. [setBlock\(\)](#) modifies the offscreen buffer (a sort of working copy of the screen's contents); the changes don't get shown until [update\(\)](#) is called. You can use `setBlock()` to make multiple changes that you want to be sure happen at exactly the same time. Using `setBlock()` and `update()` can also be faster than repeatedly using `drawBlock()`, at least when changing a large number of blocks at one time. For setting just a few blocks, `drawBlock()` will generally be faster.