It is probably true that any significant body of computer code has bugs in it, where the size of *significant* is depressingly small. DragginMath is no exception. Careful design and coding practices, coupled with deliberate testing and retesting, help to reduce the number of bugs. Some will still get through. This is a software developer's first fact of life.

We believe it is important for software developers to be upfront about bugs, informing users of the problems we know about but have not been able to fix. In some cases, we don't yet know how to fix them. In other cases, we have not been able to schedule the time against other priorities. Some things that ought to be simple… simply aren't.

And then there are the bugs we don't yet know about. It is possible that you will be the first to find a bug in DragginMath. If you do, please tell us about it. Along with telling us *what went wrong*, we especially need to know *how you made it go wrong*. **We will be happy to hear from you.**

This release of DragginMath is a useful and usable product. It does not do *all* of algebra, but it does a lot of it. **You can expect upcoming releases to do more.** For what it does, we have tried to identify, implement, and test all use cases. There are many, and we sometimes discover that we missed one. These are not really bugs, more like unseen opportunities. Of course, we are eager to find these. If you find one before we do, please let us know so we can add it to DragginMath's repertoire.

One of the design challenges of DragginMath is pairing the various possible screen gestures with specific rules of algebra. Some gestures could reasonably mean several things, and we as designers have to decide which meaning to use. *Our* design choice may not always be the one *you* might prefer in some specific case. When this happens, there is another way of approaching the issue that will give you what you want.

DragginMath Bugs

When *redoing* a step that requires you to answer a dialog (for example, enter **a=b**, then drag **a** up onto =), several things will go wrong. We expect to have this fixed in the next release. Meanwhile, just be aware that redoing such steps won't do what you expect. Fortunately, this particular kind of change is uncommon.

When distributing operators over inequalities, DragginMath tries to figure out if the distributed operators reverse the relation. It correctly handles the common cases you might actually encounter in coursework, or in life, for example **-(a<b)** or **⅟ (a<b)**. It correctly handles many cases you aren't likely to encounter in coursework or anywhere else. But there are other cases that are farther afield. Some are quite subtle, and we are still discovering the possibilities. Even after further analysis and development, this feature will probably always carry a warning to review the results of non-trivial distributions over inequalities.

Very rarely, when dragging something around the screen, DragginMath freezes. We don't know why. This happens more to new users than experienced users. We often go for months of testing without seeing this, but some new users are quite good at it for a short time. Then they become experienced users. There is probably some clue in that, but we haven't yet figured out what it is. If you can reliably make this happen, please tell us how you do it. To regain control, tap the top text and start over again. In extremely rare circumstances, you may have to kill DragginMath and restart it: tap twice on your device's home button, find the small DragginMath screen among the choices, and swipe up on it. Then tap the DragginMath icon to start again.

Something that is *not a bug*: DragginMath reduces fractions, roots, and logarithms by building a *factorization cache*, which computes the prime factorizations of whole numbers. This operation takes a long time, even on a computer. Therefore, when DragginMath computes the factorization of a given number, it remembers the result so it doesn’t have to compute it again later. If a math problem needs the factorization of a large number, this cache can want to become quite large, perhaps larger than the available memory of your iPad or iPhone.

What happens when a number is too large? DragginMath terminates suddenly and without warning, *and any recent work will be lost*. This recent work might include your homework assignment. You can then restart the app without any ongoing consequences, except that trying to solve the same problem again will have the same result. Of all the things we *could* do to handle this, sudden termination might be the least obnoxious.

How large is too large? That depends on your device, and also on the memory requirements of other running apps. In other words, we can’t answer that question, because we have no way of knowing in advance. And if you kill or start another app on your device, then try it again, your personal definition of *too large* could change. On our test devices, we start to see this limitation somewhere between 100,000,000 and 1,000,000,000.

It could be different for you.

Once again: this is *not a bug*. It is a *risk*. It is a risk you are *unlikely to encounter* if you work on problems presented by typical algebra coursework, or by life. In other words, unless you try to make this happen, you will probably never see it happen. But if you try to make it happen, you will easily succeed. Is it worth your time to do this? Probably not.

iPad / iPhone / iOS Issues

We physically test on both iPad and iPhone. But there are lots of device models out there, and we don't have access to all of them. Software that works well on our test devices or on device simulators still might do something strange or inconvenient on your device. If you think this is happening to you, please tell us about it.

Our current major concern is the iPad Mini, which DragginMath intentionally thinks of as an iPhone due to its small screen. Some non-Apple code we found on the internet determines if your device is a Mini or not. You're right: it shouldn't be that hard. We have no way of knowing for certain if this code handles all cases correctly, but it passes the tests we are able to do.

When laying out user interface components on the screen, it is important to not cover or conflict with existing Apple components, such as the status bar at the top of most screens, or physical features, such as the so-called "notch" at the top of some newer devices. Once again, it shouldn't be that hard. And yet it is, even when we use code provided by Apple to handle exactly this issue. In this release, it is our firm belief that the initial layout of DragginMath's screen is correct on all devices in all orientations. The *first time* you rotate your device a quarter-turn, the rotated layout may be slightly wrong. The layout after any succeeding rotation will be correct. Yes, that's strange.

The curved display boundaries of some newer iPhones partially obscure some of DragginMath's buttons. We continue to work on this.

When you return from one of DragginMath's full-screen dialogs (the ℹ️ screen, for example), the layout of components at the top of the screen may overlap other components. You can fix this by rotating your device into another orientation, then rotate it back. We continue to work on this.

When holding an iPad or iPhone, the edge of a thumb or finger or part of the hand sometimes wraps around the edge of the device onto the screen's sensitive area. This can render DragginMath comatose. So… *don't do that!* When you correct your part of the problem, DragginMath will correct its part of the problem.

The usual backspace key icon in iPad/iPhone apps is ⌫, but font anomalies on older devices make this work badly in DragginMath keyboard layouts. We use ⬅️ instead. We have no plans to change this.

DragginMath overrides your Dark Mode setting, always running in Light Mode. We have no plans to change this.

When the primary ℹ️ screen or the ☰ screen are displayed, rotating the device may disable the controls within that screen. If you encounter this problem, tap OK to close the screen, then open it again in your preferred orientation. We continue to work on this.

Some ℹ️ screens have links to email and the web. On some devices, if you tap a link and then return to DragginMath, links don't work on that screen after returning to it. On other devices, all links continue to work as expected. These should all work (or not work) the same way, as the same code builds all ℹ️ screens. This demonstrates the amoral nature of software: if something *should* work but *doesn't*, it is only *doesn't* that matters. If you encounter this problem, tap OK to close the screen, then open it again and tap the link you want. We continue to work on this.