

Here are some common questions and answers about the I3D tool kit:

Q: Can I use I3D with Pascal? (or COBAL, or Prolog, or Visual Basic...)

A: I3D is designed to be used with C or C++. If you'd like to use I3D with another language, we suggest that you learn C instead, it is the industry standard for graphics after all.

Q: Why doesn't I3D run as fast as Game_X?

A: I3D is a flexible 3D raycasting engine that gives similar performance to early 3D games. It allows many features that other engines do not. This flexibility comes at a small cost in raw speed.

Be sure that you are comparing I3D's performance at the same size window, same features, etc. You will find that I3D's performance on 486/33 or better machines to be comparable to most other raycasting engines.

Q: Why does I3D run slowly under Windows?

A: Once again, compare the SAME SIZE window and you will see that I3D only runs about 15% slower under windows than DOS. (The Windows demo shows a 320x200 window.) This performance loss is due to the slowness of Window's blitting. Some high performance VGA cards may improve this.

You can also substitute the new VFW 1.1 DrawDIB routines for StretchDIBits, for about a 10% improvement in speed. You'll need the VFW 1.1 developer kit to do this.

Q: Why doesn't I3D have a .DLL?

A: The I3D engine is designed to work closely with the main C program. For instance, many data structures are passed via pointers. This would not be very easy to make work with a generic DLL. It is possible to wrap the I3D .LIB into a .DLL, we just didn't think it was worth the trouble.

Q: Why isn't I3D completely written in hand-tuned assembly code?

A: Basically hand tuned assembly code is great for the inner-loops. A bit of I3D (16-bit version) is in-line assembly code. Most of I3D is straight ANSI C, which has helped to save the author's sanity.

Q: What is raycasting?

A: Raycasting is a simplified way of doing a limited 3D environment. It lends itself to texture-mapped interior worlds quite well. It does not involve 3D polygons, Z-buffers, or Phong-shading. Within these limitations, raycasting can be quite fast - much faster than most texture-mapped polygon engines.