

WorldViz Vizard Software Platform Specifications Document

WorldViz Vizard Software Specifications



New in Vizard 8:

- Upgraded to Python 3.13
- Continued support for OpenXR devices

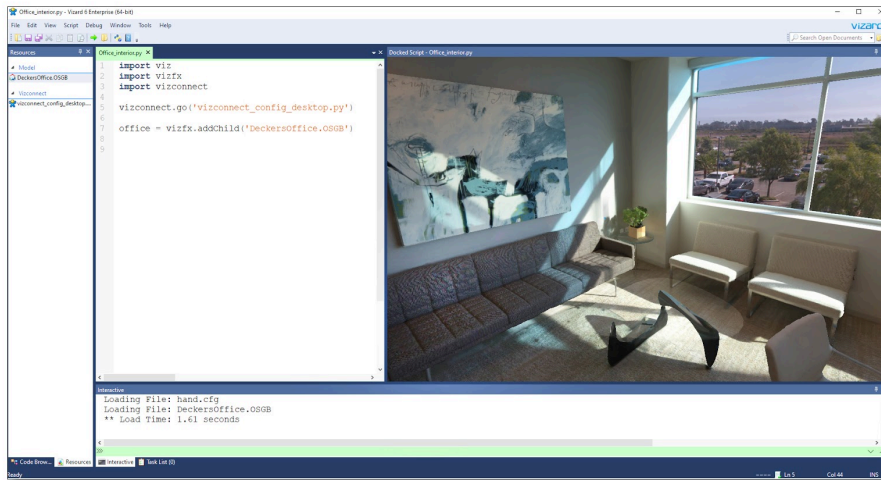
Brief Overview

- Build VR projects rapidly without the need to compile
- Python scripting, including access to open-source community libraries and toolkits (such as scipy, matplotlib, and numpy)
- Enterprise rendering with multi-user, clustering, and multi-channel capabilities
- Physically Based Rendering (PBR) art workflow
- Easy connection to the latest headsets (Meta, HTC, Varjo and more), powerwalls, CAVEs™ and professional grade research hardware such as datagloves, haptic devices mocap, physiological data recorders and scent machines
- Record and output data for scientific research with just a few lines of code with full control over measurements

WorldViz Vizard Virtual Reality Engine – Detailed Feature Review

Core features

Vizard is everything you need to build complete, interactive 3D content. Designed for rapid prototyping, Vizard gets you creating fast, and provides the resources to deploy even the most challenging applications.



Performance Kept Simple

Vizard abstracts the field of 3D computer graphics and places it in your hands through a simple scripting language called Python. You'll quickly see how powerful this interface between technologies is and what it enables you to build very quickly.

- Set up scenes quickly and easily
- Complete large projects quickly
- Import 3D and multimedia resources
- Directly connect to VR hardware
- Populate worlds with built-in avatars
- Leverage the power & simplicity of Python
- Free API to create custom effects
- Control for precise real-time projects
- Reap benefits of more than 20 years old user community

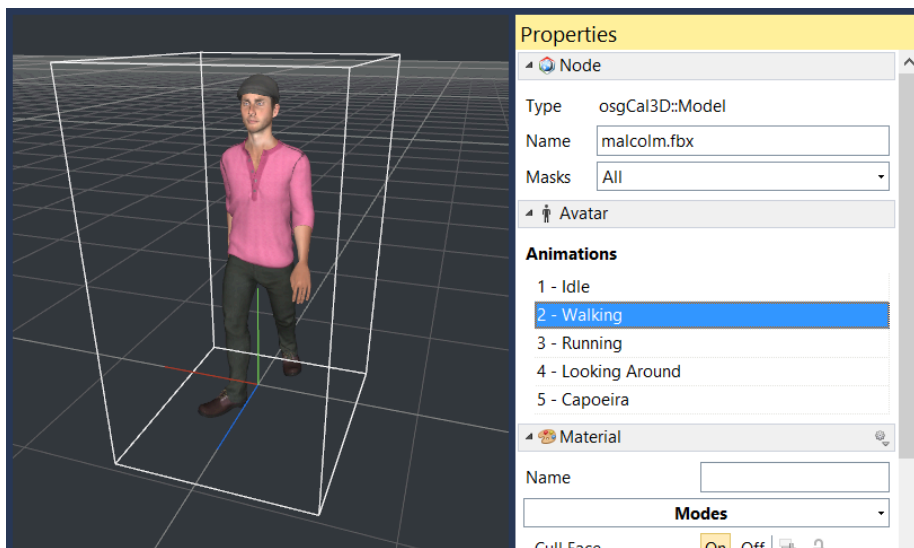
Immersive Virtual Reality

Vizard supports a wide variety of head-mounted displays, LCD shutter glasses, and several other specialty display technologies such as dome projectors and autostereoscopes. Most commercially available tracking devices are supported in Vizard. Vizard also supports numerous other devices such as gloves, haptic displays, force-feedback systems, and simple Microsoft DirectInput™ compatible gamepads and joysticks. There is built-in support for high-quality 3D sound and multi-user networking. glTF, FBX, OSG, VRML and other 3D formats are supported to provide you with immediate access to large 3D databases.

Connect to Hardware

Avatars Included (Avatar support)

With our efficiently designed avatar meshes and animations engine, adding hundreds of humans to your environment is possible via hardware skinning. For custom bodies, custom avatars can be imported into Vizard using common avatar formats such as .fbx, .glTF, or our native format Cal3D. Additionally, sample avatars are included with Vizard.



Solid Workflow

Vizard is based on a create and experience workflow. Python works like an interpreted language, which means you can make a change to your world and then instantly see the effect because you never need to recompile.

Vizard imports industry standard 3D model formats (glTF, FBX, OBJ), our native format (OSGB), point cloud formats (.ply,.dp,.3dc,.asc) and legacy model formats (see below), as well as most standard image formats (JPG, PNG, TIF, TGA, DDS, GIF, BMP, RGB, RGBA, PIC, JP2, PNM) letting you immediately leverage existing content or easily import new.

Collaboration and sharing is simple with Vizard since projects consist of Python-based scripts and your digital media assets. For those familiar with OpenGL, you'll be pleased to see how easily we've given you access to its most powerful features. For those not familiar with OpenGL, you can trust that Vizard's highly efficient render engine and extensive built-in optimizations keep your worlds running fast.

Built in Visual Editor: Inspector



Vizard features a built in visual world editor

- Identify sub-parts of models so that you can get a handle to and manipulate them in your script.
- View polygon and texture information for the model and sub-parts. If the model is negatively affecting performance, this can help you determine the cause.
- Move, rotate, and scale the model and its sub-parts.
- Add, delete, and move nodes within the scene graph.
- Edit model properties such as node names and colors.
- Navigate around the model and copy the 3D position of points on the model and camera information. This is useful for positioning objects and setting up the viewpoint in the Vizard script.

Technical Features

Supported Formats

- **3D model formats:**
 - Basic Geometry
 - Industry Standard Formats: .glTF/.glb, .fbx
 - Native Format: OpenSceneGraph (.OSGB)
 - Legacy Formats: .wrl, .flt, .3ds, .txp, .geo, .bsp, .md2, .ac, .obj, .lwo/lw, .pfb, DirectX .x format
 - Character/Avatar formats: .glf, .fbx, cal3D (native)
 - Point Cloud formats: (.ply, .dp, .3dc, .asc)
- **Raster Image Formats:**
 - JPG, PNG, TIF, TGA, DDS, GIF, BMP
 - RGB, RGBA, PIC, JP2, PNM/PGM/PBM
 - Support for compressed and mip-mapped images provided in .dds format.
- **Audio modes:** mono, stereo, 3D;
 - Supported Formats: .wav, .mp3, .au., .wma, .mid.

Scene Editing



- **Visual Scene Editor:** Lets you quickly view assets, examine and modify graph structures, see stats on polygon and texture usage and preview animations, create optimization helpers, and modify materials

Hardware Connectivity

- **Visual configuration interface:** Easy plug and play connections to most common virtual reality devices via VizConnect.
- **Headsets:** Support for a wide variety of common HMDs, including offerings by Meta, HTC and Varjo
- **Trackers:** Support for a wide variety of common tracking devices (see [this list](#) of supported vendors).
- **Support for mouse, multiple gamepad/joystick/wheel** (any USB DirectInput compatible device), full keyboard.
- **Extensive connectivity to research hardware** such as data gloves, EKG devices and scent machines. See [here](#) for a complete list of support hardware.
- **Mocap Streaming** stream mocap data in real time into a virtual environment

Multi-User Capabilities

- Connect with multiple users using a networked connection or frame synchronized clustering

Engine

- **Exe publishing:** can share and demonstrate your applications from any Windows PC.
- **Python Based Programming:** Python is Vizard's core medium for programming logic. This means you get a modern, object-oriented language that has an enormous and growing user community backing it ensuring that your code investment is secure and portable. Existing libraries are available for hundreds of different specialty tasks.
- **OpenGL based render core:** support for most OpenGL features, including recent extensions such as vertex buffer objects (VBO) and vertex/fragment programs.
- **OpenSceneGraph** provides an internal scene graph structure to your project, allowing you to reap the performance and programming benefits of this logical structure. Since OpenSceneGraph is an open architecture, you can write your own performance modules to expand on the already rich feature set in Vizard should the need arise.
- **Microsoft DirectX:** audio and video capabilities are enabled by DirectX to provide highly optimized multimedia features such as video textures and spatialized sound.
- **Multi-user network based support via UDP** which provides extremely easy-to-use front end for network-enabling any Vizard application.
- **Numerous advanced display modes** are built into Vizard, including multi-pipe rendering, quad buffering, line interlaced stereo, and real-time render-to-texture image warping.



- **Pre-programmed interactive nodes** that minimize development time for new projects includes gravity, collisions, translation and rotation engines, path trajectories, and servoed eye point control.
- **Extensive support for 2D and 3D text**, including TrueType fonts and techniques for reliably aligning your text.
- **Character animation support:** bone base skinned deformation, morph targets, motion blending, arbitrary prop attaching, and LOD.
- **Collision detection capabilities** between either the viewpoint and any node on the scene graph or between any two arbitrary mesh nodes on the scene graph.
- **Sophisticated matrix operators** are available that allow you to directly operate in homogeneous coordinates without having to write a single line of matrix math; used to transform objects (vertices or textures coordinates), viewpoints, tracker coordinates, or arbitrary data.
- **Optimization features:** LODs, occluders, billboards
- **Background Node:** Supports cube maps, rectilinear panoramas, dome, dome mirrored, plane
- **GUI Creation Libraries** - Lets you create HUD style UI elements such as text, push buttons and slider bars

Rendering

- **Sophisticated material shading techniques:**
 - Physically based rendering (PBR) capabilities
 - Cubic environment maps, pixel shaders, bump mapping, reflections, refraction, shadows, multi-pass texturing, particles, transparency, projective texture mapping, and non-photorealistic rendering filters.
- **Multiple monitor/viewports configurations** that are user definable. 16:9 aspect ratio or any other custom configuration is possible; zoom-in/out effects.
- User definable linear and exponential fog.
- **Render to texture.**
- **Maximum texture size** depends only on hardware.
- **Non- power-of-two static and video textures** through the ARB texture rectangle allowing texture effects without resizing penalty.
- **Polygon offset feature** used for rendering polygons which overlay underlying geometry; this helps avoid z-buffer collisions.
- **Unlimited number of cameras** that can be used.
- **Dynamic lighting** (up to eight lights, types: spot, omni and directional)
- **Embed HTML display** right in your graphics window for instant display of text and graphics.

For more information and to try Vizard 8 click [here](#)
