Date: 5.22.2000

Maintenance Guide

Corrective Maintenance

The corrective maintenance report will be broken down by module or class.

Object

The object class has no known bugs. The ProjectToPlane function can be a bit confusing due to the nature of the mathematics behind it, but is fully functional.

Camera

The camera is currently very stable. There is a problem with the functions that return the up direction of the camera. It should be implemented correctly before release, but it is currently not. This means that the camera will not Roll properly.

Texture

The Texture class currently has problems loading textures that aren't square in dimensions. It is currently unknown why this is a problem.

Animation

There are currently no known bugs as of the writing of this document.

KeyFrame

There are currently no known bugs as of the writing of this document.

BillBoard

There is currently only one "bug", but it could also be seen as a feature. When the BillBoard's are oriented with respect to the camera at render time, they don't orient with respect to pitch. It is unknown at this time if this is actually a problem or how it's supposed to be as no BillBoard programs have been viewed with varying pitch by the developers.

Light

All light types except for the Spot Light are functioning perfectly. The Spot light will hopefully be fixed before the actual delivery to the client, but at this time it does not function.

Vertex

There are currently no known bugs as of the writing of this document.

PW3D

There are a variety of small problems with the renderer at the time of this writing.

Compatibility:
PW3D was tested on a Pentium III 550 with a Diamond Viper V770 TNT2 video card. It currently only functions under Windows 95/98 with DirectX 6.0 or above and OpenGL drivers installed.
There are some problems with Blitting bitmaps on a 16-bit video display. This problem is being looked into and should be fixed before release.

Glitches:
Text output with PW3D is glitchy and flickery. The reason for this problem has to do with double-buffering under windows and a solution is unknown at this time.

RenderAsShadows() causes an effect known as Shadow Acne. This problem will hopefully be solved before release by using a Stencil Buffer while drawing shadws so that each part of the shadow is only drawn once.

SetDisplay() only works when called after the window is first created. It cannot be called to change the display on the fly.

Cleanup Issues:
PW3D cleans up it's own dynamically allocated variables fairly well, but it does not exit the program gracefully. This is due to a lack of debug messages in the pw3d.cpp file when something dies. For example, currently DirectInput can crash the program on incompatible systems without ever telling why. This will hopefully be fixed by release.

Perfective Maintenance

The perfective maintenance report will be broken down by module or class.

Object

The object class should really incorporate reflections as well as reflection mapping for an added coolness factor. The team played around with implementing these effects but had to scrap them due to time constraints.

In order to save memory, the object class should be modified to compute the rotations/translations/scaling at runtime rather than precompute them after calling the Transform() function. Currently, the object contains two copies of it's vertices which is very memory intensive.

Camera

The camera would be greatly improved by implementing a waypoint system with interpolation so as to produce automated fly-bys.

Texture

The texture class should handle reading multiple types of image files as well as a more robust file i/o for .BMP files. A nice added option would be to retrieve a small section of a bitmap to be used as a texture. This way, all textures could be stored in one big bitmap.

Animation

Animation would be aided by a bone system for linking the mesh together. This is much to complex for the project and therefore wasn't in the requirements, but it is a future goal.

KeyFrame

The keyframe needs no further additions in the current engine paradigm.

BillBoard

The BillBoard should probably respond to lights in the future. Currently it does not due to the nature of a BillBoard. BillBoards change orientation with respect to the viewer and therefore should not be lit up like a regular object which does not.

Light

Adding specularity to the light sources as well as a more robust and functional lighting model would improve the visual quality of the engine. The team would have tried to do more in this area, but it turns out optics is a fairly complex subject to achieve mastery in.

Vertex

The Vertex needs no further additions in the current engine paradigm.

PW3D

PW3D should be implemented in both OpenGL and Direct3D for added compatibility.

It requires more robust error checking before any commercial products are produced with it. It is fully capable of producing a quality puzzle game or even a simple adventure game, but its capabilities fall short of a true commercial 3D engine.

PW3D should also allow the programmer to change resolution on the fly as well as Alt-Tab out of the program with no adverse affects.

PW3D's biggest problems rely in the area of testing and compatibility. The TerraForm3D team does not have access to the variety of systems required to rigorously test the code.

Adaptive Maintenance

Newer video cards are coming out with Transformation and Lighting enabled on the video card. Currently PW3D performs all of its own Transformation and Lighting routines. To adapt to the upcoming 3D technology, PW3D should allow OpenGL or DirectX to perform all of its transformation and lighting routines.