

# **Messiah Technology Analysis**

Copyright © 2015 TalonBrave.info

# Table of Contents

Overview.....	3
Rendering.....	4
Character System.....	4
Background Engine.....	5
Gameplay.....	8
Scripting.....	8
Artificial Intelligence.....	8
PlayStation Version.....	9
Notes.....	10

## Overview

Originally intended to be released on Windows, the Dreamcast and PlayStation, Messiah was released on Windows in 2000, on March 31<sup>st</sup> in North America and September 15<sup>th</sup> in other regions<sup>1</sup>.

The game was developed at Shiny Entertainment by Team EGO and published by Interplay. Tommy Tallarico Studios is also credited, as Tommy Tallarico had assisted with the sound design for the game, though the music was produced by Fear Factory, Jasper Kyd and someone else.

Something that might not be well known, is that David Perry wanted to patent Messiah's technology. This wasn't especially well received and apparently, even someone from id Software wasn't very happy about it<sup>2</sup>.

---

1 [https://en.wikipedia.org/wiki/Messiah\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Messiah_(video_game))

2 Messiah Development Diary

# Rendering

The following section of this document covers the rendering capabilities of Messiah's engine.

The game includes support for both 3dfx's Glide API and Microsoft's Direct3D API for rendering through different compiled executables, with the former being the most stable. It's mentioned that there was OpenGL support in development<sup>3</sup>, however it seems this never made it into the published release of the game.

## Character System

A page existed on the Messiah website that gave some really nice in-depth information on the system<sup>4</sup>.

One of the major technical features implemented within Messiah's engine, was a technology dubbed RT-DAT, which stood for Real-time Deformation and Tessellation.

I may be mixing RT-DAT and the Character System up. RT-DAT could be in reference to the vertex deformation you can produce with the environment rather than the system that deals with characters and tessellation specifically. I might be wrong though, because the term was primarily coined so they could more easily patent the technology which I'm quite certainly mainly revolves around the Character System... For now let's just assume they're basically the same thing.

Previously generally referred to as the Character System (see the previously mentioned web-page, which does this), this technology was heavily publicised during the development of the game; it essentially allowed the game to scale itself to your system on the fly by tessellating a characters mesh based on both their distance and the general level of performance of the computer system.

This was also demonstrated at the Game Developers Conference by Michael Persson in 1998<sup>5</sup>, in which he goes on to state that they originally tried a voxel-based solution for ~~an earlier project~~ the cancelled game, Terminus, on the Sega Saturn, however noted that the amount of detail they could put in didn't scale very well when viewed up close. To solve this, they introduced a level of detail system for the voxels, so depending on the distance it would switch between a different set of voxels for varying detail; this was very quickly taking up a lot of their available storage space on the Saturn and they concluded it wasn't a viable solution in the long-term.

Turns out he actually demonstrated this voxel-like solution for the game Terminus at E3 in 1996<sup>6</sup>, sadly it's very unlikely anything of this still exists. An article about the game exists on Unseen64<sup>7</sup> along with a few screenshots and conceptual art, interesting to see some of it got recycled for Messiah, might have to do a piece on this sometime. He also adds that initially for Messiah's technology, he was targeting both the PlayStation and the Sega Saturn.

He goes on to state that after this that they used some sort of shape-based system as a

---

3 <http://web.archive.org/web/19991104111036/http://www.messiah.com/Html/rant/rantsaxs.html>

4 <http://web.archive.org/web/20001010220712/http://messiah.com/Html/technology.html>

5 <http://www.gdcvault.com/play/1016397/Messiah-What-You-May-or>

6 [http://www.gamasutra.com/view/feature/3379/behind\\_the\\_scenes\\_of\\_messiahs\\_.php?print=1](http://www.gamasutra.com/view/feature/3379/behind_the_scenes_of_messiahs_.php?print=1)

7 <http://www.unseen64.net/2010/07/31/terminus-saturn-psx-cancelled/>

reference for the meshes? This sounds rather interesting but it's hard for me to understand. One thing I do understand here is that, these cylindrical shapes that were used as a form of reference, were used in calculating and sampling the lighting in the game, which I understand was then interpolated across the mesh (rather than having to calculate samples for every polygon).

It also seems like normals were computed at run-time, from the cylindrical reference shapes, which was supposedly easy as the overall shape of the mesh was somewhat easily predictable.

He also seems to demonstrate how the data is stored for the mesh during his GDC presentation, which he describes something that's like voxels but otherwise not. Sadly it's impossible to see because they filmed him at the computer rather than filming what he was demoing (typical). To make matters worse, he also demonstrates how the tools work in great detail and we pretty much miss all of this up until about 14 minutes into the video.

It's worth mentioning that while the bodies could be cut into different sections, certain thresholds could be assigned to them between what's more physically important to the characters appearance, such as their head or torso.

It also seems like you could add singular points of importance to a mesh, so it would try to avoid losing details in certain areas, which meant it would be able to retain the meshes overall silhouette.

The team, while working on Messiah, developed a number of tools to assist with setting meshes up for their Character System. The character meshes were created within 3DStudio MAX, with a large amount of detail put into them, before they were managed by the teams proprietary tools. Using these tools, the models were rigged, bone influences were painted on and then the mesh was cut into separate body parts. Each of these body parts could be tessellated independently from one another, and sections of the mesh could be marked out so that their general shape was retained.

The minimum number of polygons a model can supposedly display was about 30, though with the polygon counter enabled in the game I didn't actually see anything get around this. It's possible though the counter only displays overall; knowing that individual body-parts can change means that this more likely applies for individual sections of the model, rather than altogether.

## **Background Engine**

For the environments in Messiah, there was the Background Engine. Unlike most engines at the time that used BSP trees, like Quake and Half-Life, Messiah instead used a portal rendering system in which levels are split into a collection of different sets that are connected together by portals.

Messiah supposedly uses a space subdivision system instead of a BSP tree, this is likely why the world is split into tiles. This method has been also used for ray-tracing apparently from the 80s to even today.

A portal was essentially a door or window, that defined the link between two sets. If a door was opened for example, the set on the other side would become active or visible. In another case, if the

player was in one set, looking through a portal, then the set linked to the other side of that portal would then become visible.

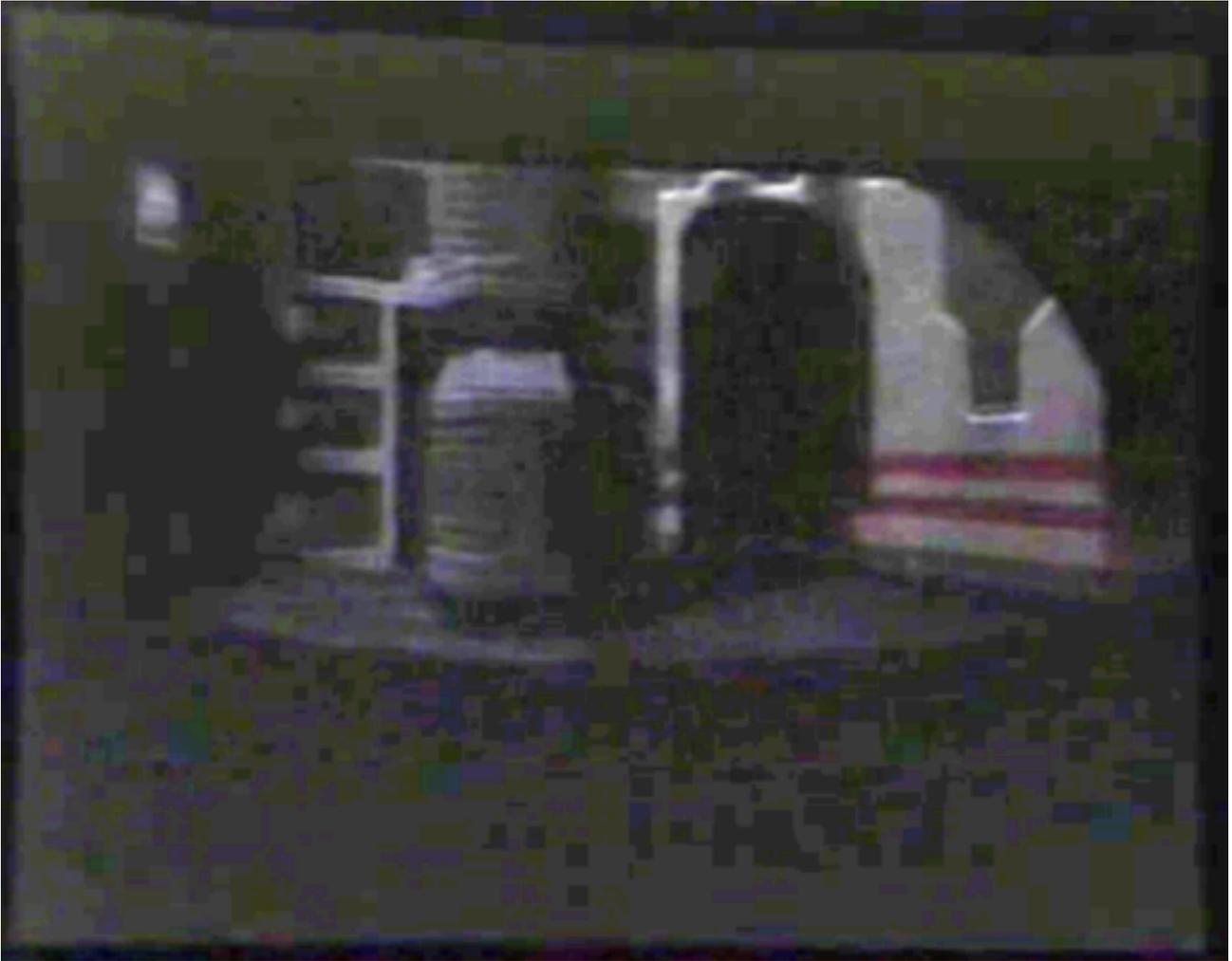
Just to make things a little bit more complicated, the environment was split into separate tiles. Depending on what was visible from one portal to another, these tiles, including whatever would be occupying them, would be culled from the view to save on rendering time.

Additionally, every triangle within the view space of the camera, would be clipped to the camera's view frustum.

From the information on Messiah's website, and from what we can see from the game, Messiah was unique for its time in that content, such as textures, was streamed in from the CD or hard-drive, as the player played through the game. This system worked well, considering the game was designed to appear as just one progressive world rather than individual levels, like most other games at the time.

Each of the levels were created in 3D Studio Max, with certain parts of the geometry specifically named so that the engine could easily identify them. For filling the levels with content, the team used a piece of software they developed called RenEd; this allowed texturing, light placement, pathing for the AI, sound placement and finally, object placement.

For lighting, they apparently used an “adaptive” light-map solution that would adapt its resolution depending on the amount of detail that was required or needed. Honestly I don't understand this 100% on how it would function but the lighting certainly is handled through light-maps in their GDC presentation, but the shadows seem incredibly sharp and there's no ambience when it's baked in from the looks of it. It doesn't seem like this actually survived in the final game, as it seems to use per-vertex lighting information instead, which is likely preprocessed by RenEd after light placement.



*Messiah's light-map at GDC 1998*

According to Messiah's early GDC presentation, for the lighting within the environments, Messiah originally used an adaptive light-map solution. This worked by scaling the resolution of the light-map in certain areas that required more detail, for example, sharp shadows being cast from an object. This however doesn't seem to have survived into the final game, with it instead using pre-processed per-vertex lighting. It's likely that this was done in consideration of the PlayStation's memory limits, however this currently isn't confirmed.

# Gameplay

## Scripting

Messiah featured a C-like scripting language.

All the information regarding this can be found within the game. The scripts seem to be compiled down into a “Messiah.cmp” file, found in the same directory as the executables, though the scripts don't seem to get recompiled in the final release automatically like they do in the earlier test releases. Sax suggested, from our exchange on LinkedIn, that there should be a launch parameter that should enable this but if there is I haven't found it.

Just to confirm the fact that the scripting language was meant to be open to users, this page appears to confirm it<sup>8</sup> and even hints at some possibility that the level editor would be released. This didn't happen though for obvious reasons (everyone got thrown onto the Matrix game once Messiah was shipped.)

## Artificial Intelligence

Early on in Messiah's development, Michael Persson was interested in using neural networks for the AI in the game. This information is from the internal development diary for the game, though I'm not sure if they actually got very far with this.

---

8 <http://web.archive.org/web/20001010220712/http://messiah.com/Html/technology.html>

## PlayStation Version

According to an interview with Mike Persson, by IGN in 1998, Messiah was very much designed from the beginning with a console release in mind. With many technical limitations of the PlayStation taken into account as they were developing the game.

I contacted Michael Persson myself sometime ago, the reason the PlayStation version of Messiah never happened seems to be because, while a prototype for the PlayStation was developed, it wasn't completed before Messiah's release in 2000, by which point the PlayStation 2 was being released and most of the Messiah team was shifted over to work on The Matrix video-game. It's very likely they put off on finishing the PlayStation version because of the release of the Sega Dreamcast, hence why there was talk on potentially doing a Dreamcast version of the game, and news that the PlayStation 2 was being developed.

While a prototype of the PlayStation version of the game was developed, it never actually saw the light of day. By the time of the games release, the PlayStation 2 was on the horizon and members of the team were moved over to work on The Matrix video-game.

## Notes

The following is a collection of additional information I've discovered during this research.

- The game has a surface-based sound system. Unclear how this works, and I haven't been able to find anymore information on it other than some screenshots with additional debugging information, which seem to also suggest different reverb types in certain areas. I imagine these have more of an effect with the 3D audio option enabled?
- I'll just uh, leave this here... <https://www.youtube.com/watch?v=QsKuh36TKvw> Evidently the game had some rather strange marketing. This wallpaper made me question my sanity (probably NSFW); <http://files.khinsider.com/vgwallpaper/1280x1024/2083-messiah-018-fmboc.jpg>
- A website<sup>9</sup> seems to describe a very earlier version of the story, however I've no idea where they got it from. The website seems very old, obviously from before the Dreamcast was even given its name. It says Bob was sent down to find the seven seals, very interesting. Also once again mentions that RT-DAT was going to be patented, though seems to state it like it actually happened (which it didn't).
- According to some archives I have, such as a development diary, it seems people in the team were pretty worried about the general response to the game before it was presented to the media.
- It's also mentioned that some company called TSi was where they did the motion capture? Wasn't too sure how they were going to do motion-capture for a baby cherub, which is understandable, so they decided to cast people with dwarfism.
- It's also mentioned that experiments were done towards some multi-player functionality for the game, though it seems assumable this didn't get anywhere.
- Just generally messing around in the game, I found out that barrels can actually be destroyed.
- Turns out you can spawn the male prostitute in the game simply just by using 'specialguy' in the main menu. He makes a really prolonged ear piercing sound effect when possessing him, seems to glide around on his feet a little and generally not quite as much polish as the rest of the characters in the game.
- There's an additional menu command, 'softwarebuys' that seems to give you a pack of grenades. This isn't mentioned anywhere online that I've seen<sup>10</sup>, which seems rather odd.
- Another command which isn't mentioned anywhere is 'voodooextreme', which seems to give you a bazooka.
- Yet another command not mentioned is 'gamespot', which just seems to give you ammo. Not

---

9 <http://www.oocities.org/timessquare/alley/9299/messiah.htm>

10 <http://uk.ign.com/cheats/games/messiah-pc-3117>

sure why video-game publications were given commands that don't appear to do much...

- It also seems like you can get one of the radiation workers back to the start area of the game and to the spot where there's a sealed door you can't normally get past, with a radiation worker moving some radioactive samples over into some sort of ship. It's a little strange because the forcefield opens in an incredibly unique way not seen anywhere else in the game, and if you set the ship, thing, to launch while the worker is loading the sample, he'll look up and then take cover as it crushes him, so the sequence is actually fully interactive and animated. It seems very bizarre that such a small sub-section would be so tedious to even see and this isn't even covered in the official guide for the game.
  - I have attempted to reach this in the game without cheats and it actually seems to be impossible, so getting into this area seems to essentially be broken.
- If you blow up the bits on the side of the gate at the start of the game, you can get through without needing a commander. This makes sense because there wouldn't be any other way through if you failed to bring her with you, but I was actually unaware about it and it's not something the official guide covers.
  - I've attempted to do this without cheats but had no luck. Don't recall seeing any cops with grenades in the start.
- Additionally you can spawn a number of the other Chots that were supposed to appear in the game. As far as I even recall, only the default generic Chot variation is ever used (plus the midgets and behemoth versions but let's ignore those as they're essentially different creatures), so it's interesting to see these other variants. Even more interesting however, is that the fourth variant seems to have its own kind of melee attacks.
- The CIGS droid in the starting area can be destroyed.
- According to an article<sup>11</sup> by IGN, originally the player was only able to possess other characters for a limited amount of time, or rather “[...] just a few minutes.” This otherwise doesn't occur in the final game, since the player is able to possess someone for as long as they like.
- “He uses the umbrella for a shield when he's standing on the ground.” It seems like it's also being implied that at some point, the Chots standard issue weapon, which allowed them to use the umbrella like function, could also operate as a shield. I don't recall this being in the final game, though I do recall them spinning in that fashion but it was while shooting others. It's possible the animation was recycled and the shield functionality was removed.
- Following the above, he also goes on to state that the Riot Cops also had the ability to charge up their laser sight and essentially blind the player. Sounds a little like a flash-bang. This actually makes a little bit of sense, considering that in the final game the Riot Cops, when they have their shields up, they don't actually move and are unable to do anything else. This suggests at some point they might have blinded the player while protecting themselves, and while the player was blinded they would have moved in to beat the player with their

---

11 <http://uk.ign.com/articles/1998/09/04/the-messiah-interview-pt-2>

batons. I can only assume this was removed because it made dealing with them incredibly hard with other cops around.

- It sounds like the droids were intended to have a little bit more presence in the game but otherwise it's difficult to tell from what he says. He says the maintenance droids would “[...] just squeeze people.” It's worth mentioning that in addition to this, both an Offensive and Defensive droid can be spawned within the game, though neither actually do much other than follow you around from the looks of it.
- In the GDC demonstration, it's visibly shown that guards at one point would aim their laser sights at the camera to try and blind you.