

BEHIND LETTER RACK 3D

How a word-tile game became a real three-dimensional world



1996 Letter Rack Classic — Delphi 3 / Pascal • 2000 Letter Rack 3D begins • 2004 the 3D world is there



Four years of building the world behind the game.

Created for the love of the game.

From a 2D game to a real world

Letter Rack began in 1996. The original Letter Rack Classic was written in Delphi 3, using Pascal, with simple 2D graphics. Despite its modest appearance, it played a really good game of Scrabble®.

But something was missing.

It wasn't real.

In 2000, I started Letter Rack 3D. Written in Delphi 5 and using OpenGL graphics, it was not simply a graphical facelift. The underlying world, objects, relationships, movement and mathematics had to be built.

Four years later, in 2004, it was there — essentially the Letter Rack 3D that exists today, with minor refinements over the years.

Delphi 3 / Pascal → Delphi 5 / OpenGL → a real 3D word-tile world

When everything is an object

There is a deceptively simple idea at the heart of Letter Rack 3D:

Everything is an object.

Not just the tiles, board and rack, but the cursor, the box and its lid, the dictionary, notebook, pencil, working fob watch and, ultimately, the table itself.

The watch's ticking sound comes from a real Mappin & Webb carriage clock that belonged to my wife. Attention to detail knew no limits — listen when it's quiet, and you'll hear it.

Each object knows where it is and who owns it. A tile can be owned by a rack, the board or the box. Because the tile knows its owner, and the owner knows where it is in the three-dimensional world, the tile has the information needed to construct its own journey.

One tile knows how to do it. So they all know how to do it.

The movement of a tile does not need to be a special animation written for one particular letter and one particular square. The object relationships provide the information from which its journey can be calculated.

Built, rather than borrowed

There is another important part of the story: the engine was not assembled from third-party library functions. The mathematics, movement system, object relationships and the mechanisms that make the world behave as it does were written specifically for Letter Rack 3D.

The reason was not an objection to using libraries. It was a refusal to accept the compromises that came with them.

Every library seemed to have an inadequacy, a limitation or a compromise somewhere that mattered. Something would be almost right, but not quite. A mathematical routine might solve the general problem but not the particular one. A graphics abstraction might make a common task easy while taking away the control needed for this world.

The choice was deliberate: if something needed to work a particular way, it was written that way. If the mathematics needed to behave in a particular way, the mathematics was written. If an object needed to understand its own relationships and use them to construct a path, that behaviour belonged in the object model.

I was prepared to accept anything other than perfection.

That is the engineering philosophy behind Letter Rack 3D. The goal was not to fit the game around the limitations of somebody else's toolkit. The toolkit was built around the game.

OpenGL was the underlying graphics technology, but it was the drawing system — not a pre-built Letter Rack engine. The world model, object architecture, movement calculations, trajectories, rotations and the mathematics connecting them were written for Letter Rack 3D.

That decision cost time. It also helps explain the four-year development period. But when something needed to behave differently, there was no third-party abstraction standing in the way. The machinery belonged to the game.

The mathematics is there to serve the illusion

A tile leaves the rack, travels through the air and settles onto the board. It looks simple because the mathematics is hidden.

Positions, relationships, geometry, rotation and movement all have to work together. Bézier curves shape trajectories; control points influence the path; source and destination come from the actual objects in the world.

Quaternions and other mathematics are used where required to control orientation and rotation. OpenGL provides the graphics layer, but the world model and calculations behind it were written as part of Letter Rack 3D.

The mathematics is there to create the experience, not to announce itself.

Looking inside the engine

The normal game hides the machinery. During development, however, the invisible calculations can be drawn on screen. These debug views expose the trajectories being calculated for moving tiles, along with the points used by the movement system.



Debug view 01 — yellow lines show calculated tile trajectories; the visible points mark positions used by the movement calculations.

The paths are not hand-drawn animation. They are generated from the positions and relationships of objects in the scene. The board, racks, box, notebook, pencil and watch remain visible because this is the real game world with the normally hidden trajectory information switched on.

Many objects, the same rules



Debug view 02 — a busier scene showing many tile trajectories calculated within the same three-dimensional world.

This second view is revealing because the same object model is doing the work repeatedly. Once many tiles have movement paths calculated, the screen becomes a web of trajectories. Each tile has its own state, ownership and destination information, so each can construct its own journey.

The player never needs to see debug mode. It exists as a development tool: a way to inspect and understand the machinery underneath the natural movement seen during play.

The game underneath the world

The 3D world did not replace the game. The game itself was already good. Letter Rack Classic established that in 1996; the project from 2000 to 2004 was about creating a world in which that game could live.

Letter Rack 3D supports competition-oriented word lists, including Collins Scrabble® Words 2024 (CSW24) and the FISE Spanish competition list. These official lexicons contain hundreds of thousands of entries, far more words than most people know and use in everyday life.

That is why the software also provides carefully crafted, much smaller vocabulary lists — around the scale of an everyday vocabulary — so players can enjoy a challenging game using words they are more likely to know and recognise.

The aim is not to require players to memorise huge lists of obscure two-, three- and four-letter words before they can enjoy the game. Players can use the competition lists when they want that experience, or choose the smaller lists for a different kind of challenge. A word can also be introduced on the fly.

Official vocabulary when you want it. Everyday vocabulary when you want that instead.

An engine that could be unfair

At the heart of the game is the lexical engine: the part that understands words.

Every time Letter Rack 3D opens, it plays a complete game in a fraction of a second. When the tiles and components have finished falling into place, the board on screen is the result of that game. It is a real game, played out behind the scenes, and it is different every time.

That speed comes from the way the dictionaries are stored. The word lists are compressed using hashing algorithms, built for extremely fast analysis. From a word list of several hundred thousand words, the engine can find every word that can be played, in a flash.

Which raises an obvious question: is that fair?

Maybe not.

A computer player that always found the best word would not feel like an opponent. It would feel like a machine. So the engine's raw speed is held in check by heuristic processes, designed to give it a human quality of play.

Weighting factors decide when to play a blank. When to use an S. When to hang on to the U because it's got the Q. And many more criteria besides.

It knows every word. It plays like a person.

Why this still matters

No borrowed shortcuts. No library-imposed compromises. The engine was built to make the world behave as required.

The original 2000–2004 architecture has been refined over the years, but its central idea remains: make the game feel as though it exists in a real place.

The dictionary, notebook, pencil, watch, box, lid, rack and table are all part of the same world.

A player can ignore all the mathematics and simply play. But underneath that simplicity is a collection of objects, each with its own position and relationships, cooperating inside a world built from the ground up.

Four years of work to make something that looks effortless.

Four years of work to make something that looks real.

One tile knows. Therefore they all know. That's the magic.

Created for the love of the game.