Making Places

Dr Richard A. Bartle
Dept. Computing and Electronic Systems
University of Essex, UK

Abstract

This essay charts the development of virtual worlds from their early beginnings to the complex, multi-million player systems they have become today. It examines how evolutionary changes in the way that the virtual worlds of yore represented space have, over time, led directly to new design features that are now regarded as natural elements of the virtual world paradigm. It also suggests how the new, by returning to the old, can create something refreshing and original to both.

Introduction

The online computer games known variously as virtual worlds, MMORPGs[1] and MMOs[2] attract millions of players[3] across the globe, each investing many hours[4] a week in their hobby for months if not years[5]. Few other forms of entertainment are this compelling, yet they remain little-understood. There are theories[6] as to *why* people play them with such dedication, but little discussion of what they *are*. In the past, they've been described as games, simulations, services and media[7], yet fundamentally they are none of these: virtual worlds are *places*.

As places, virtual worlds have a number of place-like features: they're there all the time; you can visit them; you can do things while you're visiting. The main difference between virtual worlds and other places is that they're not real – they're maintained entirely by computers and exist only in the human imagination. Formally, a virtual world is an automated, shared, persistent environment with and through which people can interact in real time by means of a virtual self; informally, it's an imaginary place able (through the magic of computers) to masquerade as real, such

---

[1] "Massively Multiplayer Online Role-Playing Games".
[2] MMO is a rare example of an acronym (MMORPG) having itself been contracted for being too long and unpronounceable.
[3] On January 12th, 2007, Blizzard Entertainment announced that their virtual world, *World of Warcraft*, had reached 8,000,000 players worldwide.
http://www.blizzard.com/press/070111.shtml
[4] On average, 22 hours a week, or, put another way, 2-3 hours a night and more at weekends.
http://www.nickyee.com/pubs/Yee%20-%20MMORPG%20Demographics%202006.pdf
[5] Some long-standing virtual worlds, my own included, have people who have been playing more or less daily for over a decade.
[6] See in particular:
Bartle, Richard A.: *Virtual Worlds: Why People Play*. In Alexander, Thor (ed.), *Massively Multiplayer Game Development 2*. Charles River Media, Hingham MA, 2005. pp 3-18.
Koster, Raph: *A Theory of Fun for Game Design*. Paraglyph Press, Scottsdale AZ, 2005.
[7] Bartle, Richard A.: *Designing Virtual Worlds*. New Riders, Indianapolis IN, 2003.

that you and other people can go there whenever you want and do things both in it and to it.

The first virtual worlds were text-based. Everything was described in words: the world, its inhabitants, the objects, the players, the events that occurred, the actions that the players undertook – everything! Almost all today's virtual worlds are directly descended from a single, textual primogenitor, *MUD* ("Multi-User Dungeon"), which was written in 1978[8]. *MUD* begat many imitators, one of which, *AberMUD*[9], was released to the nascent Internet and soon afterwards was responsible for three new branches: *TinyMUD*s[10] (which eschewed the game-like aspects of virtual worlds and concentrated instead on the social side of things plus world-building); *DikuMUD*s[11] (which emphasised strong, combat-oriented gameplay); and *LPMUD*s[12] (which were very customisable and somewhere in between). *TinyMUD* led to *LambdaMOO*[13], the textual precursor to today's *Second Life*[14]; *DikuMUD* led to the majority of today's game-style worlds, such as *World of Warcraft*[15].

The main reason for having textual worlds (rather than graphical ones) was because computers in those days were somewhat lacking in the display devices available to them[16]. Although some primitive graphical games were developed at around the same time as *MUD* and were independent of it (most notably *Avatar* on the PLATO system[17]), they never broke free of their hardware and had negligible influence on the future evolution of the genre.

A longer-lived line of graphical virtual worlds began with *Islands of Kesmai*, which was written in 1981[18] (again, independently of *MUD*). This rendered its world in ASCII graphics, with character combinations representing different features of the environment (`[]` was a wall, `~~` was water, `{}` was a tree *etc.*). In the early 1990s, a number of *IOK*-like virtual worlds[19] were written that used bitmap graphics instead of ASCII squares; they met with only limited success, however.

The first graphical worlds to overtake their textual forebears in popularity were rendered in 2½D (*i.e.* the worlds were represented internally in 2D but displayed as if they were 3D). There were two approaches: an isometric viewpoint, as utilised by *Ultima Online*[20], *Nexus: The Kingdom of the Winds*[21] and *Lineage*[22]; a first-person viewpoint, as utilised by *Meridian 69*[23]. Full 3D didn't come until the arrival of

---

[8] Bartle, Richard A. and Trubshaw, Roy: *MUD*. University of Essex, 1978.
[9] Cox, Alan: *AberMUD*. University of Wales, Aberystwyth, 1987.
[10] Aspnes, James: *TinyMUD*. Carnegie Mellon University, 1989.
[11] Nyboe, Katja; Madsen, Tom; Staerfeldt, Hans Henrik; Seifert, Michael and Hammer, Sebastian: *DikuMUD*. Datalogisk Institutved Københavns Universitet, Denmark, 1990.
[12] Pensjö, Lars: *LPMUD*. University of Gothenburg, Sweden, 1989.
[13] Curtis, Pavel: *LambdaMOO*. 1990.
[14] Rosedale, Philip and Ondrejka, Cory: *Second Life*. Linden Labs, 2003.
[15] Pardo, Rob; Kaplan, Jeff and Chilton, Tom: *World of Warcraft*. Blizzard Entertainment, 2004.
[16] *MUD* was first played on teletypes, its text printed out in upper case letters onto paper at 110 baud.
[17] Maggs, Bruce; Shapira, Andrew; Sides, Dave *et al*: *Avatar*. University of Illinois, 1979.
[18] Flinn, Kelton and Taylor, John: *Islands of Kesmai*. University of Virginia, 1981.
[19] *NeverWinter Nights* appeared on AOL in 1991, followed by *Kingdom of Drakkar* on MPG-NET and *Shadows of Yserbius* on The Sierra Network, both in 1992.
Daglow, Dan and Mataga, Cathryn: *NeverWinter Nights*. Stormfront Studios, 1991.
Lineberger, Brad: *The Kingdom of Drakkar*. MPG-net, 1992.
Buiter, Karl: *Shadow of Yserbius*. Sierra On-line, 1992.
[20] Koster, Raph, *et al*: *Ultima Online*. Origin Systems, 1997.
[21] Song, Jake: *Nexus: The Kingdom of the Winds*. Nexon, 1996.
[22] Song, Jake: *Lineage*. NCSoft, 1997.
[23] Sellers, Mike and Schubert, Damion: *Meridian 69*. 3DO, 1996.

EverQuest in 1999[24], and this is where we are today. Almost all of the several hundred commercial virtual worlds currently in development display their content as a 3D scene[25].

The history of virtual worlds is not merely about their visual appearance, of course. They could be described them in terms of their social development, the evolution of their gameplay, or their commercial success. Why, then, did I choose on this occasion to focus on how they look?

Naturally, the designers of virtual worlds have to shape them to the way they will be perceived by the players, but this is not the main reason for offering a display-centric view of their history here. The thing is, the manner in which a virtual world is presented imposes limitations on its underlying representation, and this representation lies at the heart of what the virtual-world-as-place *is*.

Ground Rules

Unlike the architects of real-world constructions, the designers of virtual worlds get to define the physics of their universe. They can remove gravity, make solids permeable to other solids, allow instant acceleration to any velocity, permit two objects to share the same space – they can do pretty well anything they want. If they wish to make it that, when you pick a flower, a snowflake rises from a distant mountain top, they can do so.

On the whole, though, they don't do these things. The reason they don't do them is because such conduct makes the virtual world harder to believe. In real life, people are accustomed to living in a universe that functions automatically and reliably, and have internalised most of its workings. If I ask you to imagine what would happen if I threw a tennis ball at a wall, you'd have no trouble doing so. You'd hardly have to think about it, and would only pay attention were things not to happen as predicted. If the tennis ball melted onto the wall, you'd stop to question why.

In a virtual world, what happens when a tennis ball is thrown against a wall is up to the designer, but unless there were something special about the wall or the ball they'd probably want it to behave much as it would in reality. If the virtual world's physics are too much at odds with reality's, the result is that everything the player does is strange and new and requires consideration – it's impossible to feel you're *in* the virtual world in such circumstances. Thus, although the virtual world could be limitlessly and capriciously different to the real world, the more that things work the way a user of reality would expect them to, the more *persuasive* the world is. This means that when something does happen that's out-of-the-ordinary (for example magic), it can be correctly identified as such by the players.

So it is that much virtual world design involves attempting to simulate reality to a degree which, while not necessarily perfect, nevertheless is sufficiently below the perceptive radar that it doesn't trigger any sense of dissonance. Luckily, because players actively want to become immersed in virtual worlds, this is actually easier than it sounds. Put another way, you don't have to make people wear full Virtual Reality gear for them to feel they're in a place if it's a place they *want* to feel they're

[24] McQuaid, Brad; Clover, Steve and Trost, Bill: *EverQuest*. Verant Interactive/Sony Online Entertainment, 1999.

[25] Note that this does not mean that they give a stereoscopic view; the world may be 3D, but the computer screens they display on are only 2D.

in; all you have to do is ensure you don't jar their expectations unduly while they're there.

Given, then, that virtual worlds should endeavour to approximate reality for their everyday workings, how can this be implemented?

The real is at a distinct advantage over the virtual in that it works entirely in parallel. It can ray-trace every photon in the universe simultaneously, whereas even the best of today's home computers have a hard time rendering a few shadows in real time. Virtual worlds therefore have to cut corners. As it happens, they have developed three ways to do this, which correspond to the three main display formats:

- Textual worlds have a contiguous arrangement of locations.
- 2½D graphical worlds have a tessellated arrangement of locations.
- 3D graphical worlds have a continuous arrangement of locations.

These formats were not created particularly for virtual worlds, having appeared in innumerable regular computer games[26]. Increased computer graphical capabilities created the pressure to change from text to 2½D to 3D in offline games, and the expectations of the players applied it to online environments, too: virtual worlds with pretty pictures were more likely to attract new players than those which described their worlds in words. However, the different display formats imply different internal representations of the worlds, which impact on what can be described within them. In particular, there are things that textual worlds could implement that 3D graphical worlds can't. This has led to innovative attempts to find other ways of doing similar things, some of which have (as we shall see) been successful enough to become absorbed into the virtual world paradigm.

Shards and Zones

Before examining the consequences on virtual worlds of having different internal representations so as to conform to different display formats, there are two things all formats have in common that need to be mentioned: shards and zones.

Virtual worlds, much as the real world, can only hold so many people. The limiting factor is *content* – that which, if players are thought of as consumers, is what they consume. Content is a hard concept for non-gamers to grasp, but it's the stuff from which players fashion the events that they find fun. A virtual world with lots of things that the players want to do has lots of content; one bereft of desirable activities lacks content. The problem is, a virtual world may have sufficient content for only a certain number of players before they start treading on each other's toes. As an analogy, there are many fun activities at Disneyland but if the park didn't shut its doors when it got full then few people would get to experience those fun activities.

Virtual worlds have an advantage over Disneyland in that they can easily be cloned. If there is only enough content for 5,000 people, a copy of the virtual world can be set up so that the overflow has somewhere to go. This is a long-established practice, with *MUD* being able to crank up a new incarnation of itself if it filled up with players. Most commercial virtual worlds today[27] will open with multiple

---

[26] Graphics superseded text as the *de facto* standard more quickly in offline games, though. Text adventures as a commercial format died almost overnight in the mid-1980s as graphical games supplanted them, but textual virtual worlds thrived for another decade before succumbing to the same fate.

[27] The main exceptions are *EVE Online* and *Second Life*, which each run but one instantiation of their virtual world that can be scaled up as new players arrive.

instantiations running on separate sets of computers. These instantiations have several names, of which the commonest is *server*. This is an ambiguous term, however, so it's usually easier in formal discussions to refer to them as *shards* (which is what *Ultima Online* called them[28]).

Thus, even though it's usual to refer to the cities and continents of virtual worlds as if they were unique, it should be remembered that actually there may be dozens of them. Saying "I've been to the top of the Tower of Althalaxx" is not the same as saying "I've been to the top of the Eiffel Tower"; actually, you've only been to the top of *a* Tower of Althalaxx.

The reason that the term "server" is ambiguous, by the way, is that formally a server is a single computer, but it's sometimes used as shorthand for a "server cluster". A server cluster is a collection of computers working closely together so as to deliver a particular overall functionality – in this case, running a shard. For most virtual worlds, they share their load on a geographic basis, because players will tend to spend most of their time in a playing session in the same few adjacent areas.

At first, in virtual worlds such as *EverQuest*, the geographic areas each server in a cluster was responsible for were quite large, because of constraints on the client software[29]. The number of textures that the client could hold in memory was limited, therefore when the player moved to an area with a different look (a forest rather than a desert, say), the existing textures had to be replaced by new ones so as to avoid any delay in displaying images. Unfortunately, it could take time to switch to the new textures when an area was entered – up to 40 seconds or so. Transferring responsibility for handling a player from one server to another was also a non-instantaneous task, so this was done concurrently. The areas handled by a single server were called *zones*.

Nowadays, videocard memory can hold many more textures and they can be loaded much quicker. Consequently, servers can be much more nimble in sharing load between them, using dynamic techniques to ensure that areas with a high concentration of players are spread evenly. Zones still live on in the minds of players and (perhaps more importantly) designers, however, and they still share some of the features of earlier versions. In particular, they will each tend to have their own distinct look (and textures), and it is very rare that the denizens of one zone will cross a border into another.

Contiguous Locations

Textual worlds represent space as a set of interlinked nodes. Each node represents an atomic location (commonly called a *room*), which generally conceptualises the smallest meaningful space into which a player's character can fit. The room will have a short description for people who have seen it before and a long description for people who are unfamiliar with it. The long description will usually place the room within a context, noting other locations which are to be regarded as adjacent. These adjacent rooms can then be reached from it using primitive, directional movement commands (typically compass points – north, northeast, east

---

[28] As fictional cover for why there were multiple canonical copies of the *Ultima* universe, it used the wonderful metaphor of a mirror breaking into a myriad of shards, each one able to reflect what the whole reflected.

[29] The *client* is the program which runs on the player's home computer to perform the necessary input/output, in communication with the server.

etc. – plus perhaps thematic "directions" such as out). Typing a direction will move the player's character from the current room to the one pointed at by the appropriate exit link.

A map for a textual world therefore consists of a network of rooms connected by a set of arrows that correspond to movement commands. The rooms are said to be *contiguous*, in that movement between them is instant and there are no spaces between (because if there were, these would also be rooms). They are therefore at a fairly coarse granularity, but it is not one that unduly offends players' sense of locale. People are as capable of thinking themselves to be "in the hallway" as they are to be "at this point" (which, because of what they can see around them, they can straight away determine is in the hallway).

This modelling of the virtual world as a network of nodes has some interesting properties. For example, the arrows on the map need not be bi-directional – it's quite possible to go north from room A to room B only to discover that south from room B leads to room C. Players would, of course, notice this as being counter-intuitive, so it doesn't happen too often; the point is, though, that it's *possible*. Designers can use it to create a series of maze rooms, for example, such as "lost at sea". Rooms can even move, by changing their exit links dynamically.

Another important consequence of having locations arranged as a network is that nodes need not represent rooms of the same size. One node could be a cupboard, another a mountain. This is a wonderful gift to designers, as it allows worlds to be defined in terms of the *importance* of their locations, not their size. It means that designers have greater artistic control over the prevailing atmosphere, and a better ability to alter pace. It also permits rooms to be bigger inside than outside, because there need be no connection between how large the description of a room *says* it is and how many nodes are accessible through it: it's possible to put all of Narnia inside a wardrobe if you so desire.

The final unusual characteristic of a contiguous representation of space within a virtual world is self-reference. A location can link to itself, so that going west from room A takes you back to room A. Furthermore, as rooms are just another kind of object, they can be picked up, carried around and placed inside one another – or, indeed, inside themselves. This allows for great creative freedom on the part of designers, so long as they maintain an overall feeling that the world is consistent (so as not to cause players to disbelieve its fundamental fiction).

To implement a set of rooms networked in this fashion involves many data records and many more pointers between them. It is far easier to create a simple, two-dimensional array – a grid 100 by 100 would deliver 10,000 rooms immediately. Although some of the early virtual worlds did experiment with this kind of set-up[30], on the whole it did not find favour: the resulting rooms were boring when compared to the linked-node system. It was fine for outer space, but not for planets. Besides, if anyone really wanted to have a few fixed-size rooms in a lattice format, they could implement it using regular nodes anyway; there was no need to use a special data structure. Thus, with the ease-of-implementation issues seemingly outweighed by the loss of flexibility in design that they imply, what could be gained by switching?

The answer is that a grid format gives easy access to a visual representation of the world.

---

[30] The most notable was *MirrorWorld*, which blended it seamlessly into an otherwise network-structured world.
Cordrey, Pip et al: *MirrorWorld*. Input/Output World of Adventure, 1986.

Tessellated Locations

Describing a virtual world's geography in terms of an array of squares makes it possible to render the world graphically as an array of tiles. The major advantages over a network of nodes in this respect are the constant scale[31] and the implicit connection between the squares[32].

Despite its fairly rigid representational structure, a tessellated world can be displayed in a number of ways. At first, with home computer screens unable to deliver anything but quadrilaterals at any speed, the view presented to the player was from an angle directly overhead, usually with north fixed at the top of the screen. As more pixels became available, and computer speeds increased so that it became practical to mask out parts of an image component rather than display a complete block, the overhead angle dropped from 90 to 60 or 45 degrees; later, north was moved 45 degrees clockwise (so as to give a less chunky look). The camera wasn't usually able to change angle beyond that, though, except perhaps in 90 degree rotations (so that objects had to be drawn from at most four angles).

Using this isometric approach, height could now be shown; this meant that hills and mountains no longer had to be suggested by a change in a square's background texture, but were there for all to see from the topography of the surface. When pixel density and available processing power improved to the extent that polygons could be thrown up rapidly on-the-fly, the view changed to first-person 3D, the camera acquired the ability to roam freely, and perspective made an appearance for the first time.

Regardless of their different looks, all these representations have the same underlying structure: a fixed, rectangular array. In theory, a mere rewrite of the rendering engine could have made the ASCII-graphics *Islands of Kesmai* have the appearance of pseudo-3D *Meridian 59*.

To virtual world designers, this tessellated representation presented new challenges. The fixed scale meant that distance could no longer be ignored: whereas two castles could be separated by 20 rooms of "wilderness" in a textual world, they'd need to be at least 200 squares apart in a tessellated world or it would feel as if they were right next door to one another. Furthermore, some of those 200 squares would have to have things in them lest the world seem featureless and uninteresting. To travel large distances would take so long as to frustrate players, so some means of fast transportation would have to be devised (teleporting, ships, portals *etc*.).

Another issue was the very 2D nature of the world model. It might not perhaps be regarded as a great loss to be unable to create a set of rooms that contain each other, but it is rather tiresome for buildings to be restricted to a single storey and for caves and bridges to be impossible. Turning the array from 2D to 3D by the addition of a Z co-ordinate was the theoretically sensible solution, but it greatly increased the amount of memory required[33] and most of the resulting cubes would contain nothing anyway.

The practical solution was to introduce a degree of nodality back into the system. Parts of the location-definition array were given over to self-contained areas

---

[31] One square usually represents a square metre – the floor space a single person occupies.
[32] Movement is merely a matter of incrementing or decrementing X and Y co-ordinates, rather than following pointers.
[33] Keeping the mapping of array space to virtual world space, turning *X* by *Y* squares into *X* by *Y* by *Z* cubes would use *Z* times more memory.

that could not be reached by regular means. Access was gained through particular squares flagged as being *coincident*. As an example, if on the main map you walked onto a square containing a staircase leading upwards, that would teleport you to a sub-map for the floor "above" where you were; if, on that sub-map, you entered the square containing a staircase leading downwards, you'd be taken back to the main map. Although something of a hack, this solution nevertheless worked, the slight break in consistency not deterring players unduly from making such moves.

One of the consequences of a switch from text to graphics was the increased general granularity of the geography. Textual worlds tended to have broad-sweep locations, their details filled in by the imagination of the player (and therefore personal to every individual). In a graphical world, the details are to be sensed, not imagined; this means everyone sees the same thing, but the results aren't necessarily as glorious as they might be for the imaginative person.[34] On the whole, though, players liked being able to see more of the world at once, and it certainly had more persuasive impact than text when first encountered.

Although the granularity became finer than what was commonly available in a node-based world, it was far from ideal. Anything large and curved was a problem (paths, rivers and circular buildings showed their right angles), and there was an unresolved issue of what to do if people were standing in the way (blocking a doorway, for example). Textual worlds didn't have these questions to answer, but, by making the environment more visually concrete, graphical worlds did.

As with a nodal representation of a world, a grid-based representation is still comprised of contiguous locations; what's more, they *feel* contiguous, because everything conforms to the squares. When you move, you move from the centre of one square to the centre of another. Walls, roads and coastlines follow the boundaries of squares. Trees, furniture and rocks neatly occupy squares and are arranged in tidy, equidistant patterns. People, animals and monsters follow right-angled lines of movement. You *know* you're in a world made of squares, which is a little disappointing as it isn't quite how the real world is…

What if the world were not made of squares, but were made of polygons?

Continuous Locations

In today's virtual worlds, the one-to-one mapping between the structure of the virtual space and the software data structure that models it has gone. It has been replaced by one in which sets of polygons represent surfaces, whether those surfaces are of the ground, of buildings, of denizens or of objects. Instead of creating a set of nodes or squares and saying that each node/square corresponds to a particular location, a location is instead a mere point in a 3D co-ordinate system. Most of the world is empty, so does not need to be represented explicitly in a data structure; lists of objects within visual range are held instead, and rendered in terms of the polygons that make up their surfaces.

This approach uses surfaces because that's all that a video card needs to know in order to display an object's image. Objects are, in fact, hollow inside. Indeed,

---

[34] There are other consequences, such as the loss of functionality inherent in having to use a mouse to issue commands rather than typing them exactly (does clicking on a sword then clicking on a bag mean you want to put the sword in the bag or you want to hit the bag with the sword?). These don't affect the spaciality of the virtual world, though, so are not discussed further here.

sometimes glitches in the world model or incomplete collision-detection by the camera can mean you get to see what's inside a creature or below the planet surface.

In a true 3D world, the representation finally goes from contiguous to continuous. Strictly speaking, however, because computers store information using discrete bits, even their "real numbers" are not actually continuous; nevertheless, the level of granularity is so fine that to players it *feels* continuous. You can inch half a step to the left, and your position as it is stored by the computer will actually change.

Although 3D overcomes the main problem of 2½D in that it allows for multi-storey buildings *etc.*, its principal advantage is that it looks less blocky and more persuasive; this is the main reason why developers prefer it, rather than its representational capabilities[35]. This isn't to say that the virtual world is full of smooth surfaces and wavy lines – it's not. On close inspection, wheels might be octagonal, for example. Also, although the world does not store everything as squares, that doesn't mean the software tools used to create it don't do so: it's not unusual to see paths stretching off into the distance following zig-zag lines rather than bending smoothly. Nevertheless, the situation is far superior to that pertaining in tile-based worlds, which are locked into having square-edged terrain features whatever they do.

That said, 3D does share some of 2½D's limitations. The scale is necessarily still fixed, so the time taken to travel to places and the need for content between those places are still issues. Another problem is how to handle crowds: a textual world can allow a thousand players in a single location and they still won't feel crushed together, but a graphical world has to deal with occupancy. Tessellated worlds can space people out easily enough by permitting only one character per square, but this does make a crowd an impassable object; 3D worlds can put bounding cylinders or spheres on characters, but with similar results – made worse by the fact that players can still see gaps between avatars and figure they should be able to elbow their way through. Because of this, the trend is to remove collision-detection for characters in 3D virtual worlds: you can move your avatar completely through the space occupied by another player character or non-player character. This is somewhat fiction-busting, but players have come to accept it as a fact of (virtual) life.

Such a solution has a side-effect, though. If multiple people can occupy the same physical space, then they can access the same content associated with that space. Textual worlds could fairly easily put alternative content nearby, but this is not possible for graphical worlds as the constraints of visually simulating reality mean there simply isn't room for it. The problem is exacerbated by the higher player numbers enjoyed by graphical worlds. Basically, you may get 50 people wanting to access an area where there's only content enough for 5.

The traditional approach in textual worlds was *laissez-faire*. If 50 players want to storm through the content, let them; if they don't, well, assuming the existence of enough alternative content in the virtual world as a whole, they will soon learn to go to different dungeons instead of all hitting up the same one. This solution works up to a point, but it runs the not entirely insignificant risk that players will find their alternative content in an alternative virtual world…

Another possibility is to allow only one group access to an area at a time. Have a bolt on the door, so that whoever gets in first can stop other people from following; use magic or some other fiction to allow that player's allies in. Put a timer on it, so no one group can hog it indefinitely. This is great if you're the lucky group

---

[35] It should be noted that 3D requires much more artwork than isometric 2½D, because the level of detail is greater. This makes such worlds rather more expensive to create than 2½D, which in turn is far more expensive than text.

that gets in, but frustrating if you're left waiting in line until the previous group has completed.

The modern solution is to switch to an *instance*. This takes the coincident squares idea of 2½D worlds and converts it to that of coincident *planes*. You step through such a plane, and you're transported to a self-contained mini-world beyond. The idea is extended, however, by allowing multiple copies of the mini-world to exist. One group of players will go through into their own, private instantiation of the area – an instantiation that will disappear when they leave but which can never be accessed by other players. The next group of players will be taken to *their* own replica of the area instead. Thus, instances can be regarded as sharded zones of a virtual world.

Instances allow players to access popular content without spoiling the experience for each other, and they mean the content can be fine-tuned for groups of particular sizes. Their disadvantage is that nothing that goes on within an instance can have any effect on the virtual world beyond, in the same way that nothing that goes on in one shard can affect what goes on in another. It can do so at an individual level (allowing you to go somewhere you couldn't before, for example), but any global effects are likely to be short-lived. Nevertheless, on the whole players approve of the idea of instances and expect them, probably to the point that even if a new virtual world had so much readily-accessible content that it didn't need to limit access they would still be surprised if it didn't have instances.

This means that a virtual world featuring instances could be mapped in terms of a directed graph. The non-instanced part of the world is a set of inter-linked zones, and the instances connect to those[36]. This suggests that although the virtual world is continuous at the visual level, at a more conceptual level it's contiguous – just as were its textual forebears.

Thus, things have come full circle: a contiguous representation has become a continuous representation, which has become broken down into groups of contiguous (but grainier) locations. The difference is, this experience has brought something new with it: the potential for multiple copies of those contiguous locations.

Summary

The way that a virtual world represents space is strongly related to how that space is to be displayed. The closer to reality is the appearance, the more constrained is the representation. Over time, virtual world designers have developed techniques to address some of these constraints, leading to solutions that are now considered to be part and parcel of what a virtual world is.

These solutions bear an uncanny resemblance to the older, more flexible representational structures that were present in textual virtual worlds. Once designers realise this, they will be able to perform some of the tricks they could do when virtual worlds were rendered as words, not as pictures.

Better yet, they can try out new tricks! Although textual worlds *could* have sharded their nodes, they didn't – their designers didn't give it any thought. Instances, which were introduced to solve a problem that textual worlds didn't have, now make sharding a node fairly easy. Overlapping, coincident worlds become possible, in which players who made one decision will be taken to one version of a zone, but players who made another decision are taken to a different one. From the point of view of the individual player, the world is consistent; from a god's-eye-view, it's

[36] And, sometimes, to each other. Instances can have instances within them.

anything but – a multidimensional layering of differently-phased zones, each one personal to every player but at heart fundamentally the same. In this scenario, actions *could* have global consequences, but only for those involved – your group might cause the volcano to explode, exposing a strange netherworld of troglodytes and their demonic slaves, but to the players who haven't unleashed the necessary magics it's still just a mountain. This is a startling opportunity for designers to do something really *new* with virtual worlds for once.

What began as a partial solution to the representational problems imposed by the higher look-and-feel expectations of players has thus led to an improvement to the original representation itself.