GAMASUTRA
The Art & Business of Making Games

Interview: The Making Of Dwarf Fortress

By John Harris

There have been few indie gaming success stories as big as *Dwarf Fortress*, an ASCII freeware simulation game in which the player helps to establish and govern a colony of dwarves, as they construct a *Moria* of their own.

The scope of the game defies belief: it contains an extensive world generator, a three-dimensional cellular automata system for simulating fluids, naming languages for all major races, an economics simulation, and even a complete Adventure Mode in which the player can explore abandoned fortresses.

It's so detailed that large web communities have sprung up around the game, both on the developer's forums ar Something Awful, where players trade stories about what happened in their games. Some of these stories have become popular outside the game's community.

Amazingly for a game as rich as *Dwarf Fortress*, it remains the work of just te wo people, programmer Tarn "Toa Adams and his brother Zach. Tarn supports himself primarily with donations from *Dwarf Fortress* enthusiasts. In interview, Tarn talks about the inspiration and origins of the game, and some of the finer points of its constructi

This interview was originally held over a private IRC conversation, and edited into a more traditional interview fo

Games Leading Up To *Dwarf Fortress*

Why don't we start with what gave you the inspiration to create *Dwarf Fortress*?

Tarn Adams: It's really the joining of two different tracks of development. The first has to do with our (my broth Adams and I) "fantasy game." Way back when, like, when I was in elementary school, fifth grade or so, I wrote game called *dragslay*. That was just a D&D game, more or less, with text that scrolled up from the bottom of the

You mean Infocom-style?

TA: Pretty much, yeah. But it didn't even have a map. Just a line through 8 or 10 encounters, then a dragon figl repeat. It was very basic, but it was our first project. Our first fantasy project, rather. We had a ton of other space/samurai/etc. things going on, all BASIC, all very small, so we didn't do so much with it. Then in the summ freshman year in high school, I picked up C and *dragslay* was revisited.

So I was able to make a larger project. *Dragslay* was sort of horrifying. You could kill an enemy and then be atta maggots from the corpse at random. Wounds could fester as well then you'd catch diseases.

In any case, yeah, this time it got a basic world map, and you could depopulate goblin tribes in the wilderness a searched for dragons. Along with several other projects, we worked on that until, say, my sophomore year of co

So the game kept track of goblin populations?

TA: Yeah, it was one of my first save/load experiences outside of score lists. It kept track of their names and kill could yell at you like the enemies do in *Dwarf Fortress*. It was really quite far along, farther along than *DF* adver

in some respects. By yell at you, I mean it would track which players they had killed then they'd proceed to mak you. They also stacked their skulls outside the caves.

Ah, that's a significant departure, I'd say, from the basic fantasy game template, and quite an interes Between-game continuity.

TA: By this time we were operating in spite of the games we were playing, rather than because of them, I think. 1998, 3D FPSs had already been out and popular for five years or so, I guess, and I was hardly playing anything A lot of the games we played, like the *Ultima*s, also kind of got us into thinking about the worlds themselves, rat just playing a game in one.

So I hit a Borland compiler limit, 65K something or other, and moved on to other projects. I didn't see a moneta in my games, and I wasn't into CS as an academic pursuit, so I got serious about math for a few years, and wer school.

The summer before I went to grad school, though, we restarted the fantasy project. This time, it was called *Slav Armok: God of Blood*, named after Armok, the god from *dragslay.* Armok himself was named after "arm_ok", a that counts the number of arms you have left, for inventory purposes. This was a 2D project in a somewhat-isor where you walked around a cave with a bunch of goblins in loincloths. It was entertaining, but short-lived.

I got started on the actual *Slaves to Armok* that I released on Bay 12 around 2000-2004 or so. This was our fan Lots of complex things going on, and of course, a boatload of plans. It was unwieldy, and got even more so whe 3D. This is the first piece of *Dwarf Fortress*.

Slaves to Armok: God of Blood

Now, you might have seen the various, questionable games littered throughout my site. I would occasionally tak from *Armok* and grad school to write really short projects on weekends or whenever I found time or had an idea them got released, and many others just died before they saw the light of day.

A game called *Mutant Miner* is one of these. It was roughly inspired by *Miner VGA* and Teenage Mutant Ninja Tu mutagen, or something. As in *Miner VGA*, you'd dig out tunnels in the ground under a few buildings, looking for and deal with threats. It's turn-based.

In *Miner VGA* you can find many things in the ground. In *Mutant Miner*, we added green radioactive goop. You c back to one of the buildings at the top, and apply it to yourself to grow extra arms and other mutations that wou you combat threats down below. Which were just, like, these holes that would spit out enemies or blocks of slim you'd encounter in the mountain.

I eventually wanted to put in extra miners though, and since it was turn-based, it started to drag like a battle in Gold Box game. And instead of rewriting the game, I thought, well maybe it should be dwarves instead. And it s real-time, to combat the SSI problem. Now, you'd be digging out minerals in a mountain, combating threats insi

making little workshops.

Then I thought, well, how should the high score list work? We really like to keep records of plays. Not just high s but expansive logs. So we'll often try to think of ways to play with the idea. This time, the idea was to let your a come into the fortress after you lose and find the goblets you've made, and journals it generates.

If your adventurer successfully brought these back to town (after facing threats in the now-abandoned fortress), would get to see the fortress' stats. For instance, if they found a journal that said "This month, we produced 3 si goblets...," they get the entire set of stats on silver goblet production in the score list.

That was the idea. It was supposed to take two months. I started in October 2002.

Creating the game?

TA: Yeah, longer games took a few weeks, most of the other ones up there took 2 days (*ww1medic, corin, Kobo* etc.) so I didn't think it was a bad estimate. And it might not have been, but I called up my brother and we just planning it out. It became obvious that it was really stealing thunder from *Armok* which was right in the middle cycle at this time, so *DF* development was actually stopped that November and I went back to *Armok*, which wa all right, and various other projects.

"Strike the earth!"

TA: By around 2004 though, it was obvious that *Armok* was floundering under its own weight. So I announced o forums that I'd be switching gears to dwarves. I kept adventure mode a secret though.

(Ed note: Adventure mode is a special feature of Dwarf Fortress*, where the player can take an adventurer throu areas of the game's random world in a traditional roguelike adventure. The player can visit old places he's worke Fortress mode.)*

So, you kept going with a variety of different games at the time?

TA: Oh yeah, it's important to break things up. *DF* isn't the only game I want to write.

Did you already have a good forum community at the time?

TA: It was nowhere near as large as it is now, but there were a few dozen hardcore people and many more occa lurkers/posters. Most of them came on board with me working on the dwarves. They knew only that it was a dw kept the RPG part quiet, for fun.

It took longer than I thought, but I finally got it out in August 2006. It was our only fantasy project by that time *Armok* dev had moved over there, so it was *Armok 2* as well. Hence the long-winded name, which is for kicks, n

Ah, hence *Slaves to Armok, God of Blood II: Dwarf Fortress*. A quick question: The game already con great many ideas working in concert. It's really quite amazing. Do you think *Dwarf Fortress* will get t point where it too will flounder under its weight?

TA: Nope. After many project deaths, I think I've finally got a handle on simultaneous programming and design. to keep it up though. There's certainly a lot to learn. And I think the latest z-axis release shows I'm not afraid to entire project if I have to [laughs]. Having it collapse is not really an option at this point. My livelihood is tied up

The z-axis thing, that really was a surprise, yes. But a good one. Before, the world generator seemed determine climate and wildlife and not much else. Now it's an absolutely essential part of the game, determines the entire nature of the fortress to come.

TA: I was going to be starting armies without messing with the z-axis, but there came a point where I felt like I' hemming myself in. I'm only going to have time for one shot at this fantasy project now, so I want to stuff as m as I can.

We appreciate that attitude, believe me.

TA: It would be almost like facing mortality to refuse a solid idea. I hate having to say something good doesn't f specs.

What's left to answer is why'd we be so into doing a fantasy game. That's probably the same as everybody else: D&D, myths, and of course, the movie Beastmaster. (We like the part where the evil priest is like, "You'll be sac 'The God of AAaa,'" like they didn't even bother thinking of a name, just powering through on the power of their badassedness.) But there were all kinds of things like that. In the movies, books, the arcade, PC, consoles, we v surrounded by that sort of thing.

So along with generic sci-fi, generic fantasy is part of our heritage. This kind of brings us to the stories.

Storytelling as an Idea Source

TA: That started with *Armok*, although there might have been traces of it with C *dragslay*. We had a spiral noteb we decided to write stories about events that happened in the game universe, things that we'd like to have happ brother Zach would write a chapter, then I'd write the next chapter, we'd go back and forth several times. Then over it and decide on some low-hanging fruit to implement.

It was partially inspired by our repeated experiences with plots in video games. We never really wanted to write a lot of them seemed like they could be generated by a computer. So we thought about breaking stories down in elements, and working with those instead. You'd be very hard-pressed to capture really beautiful symbolism or a advanced writing device like that with a random generator, but there are very few game stories where that woul issue.

It's really the same principle as world generation or anything else in the game: finding the key, basic elements, rules that govern them, and then activating those things in the world.

So, kind of a random drama generator?

TA: That's right. Create actors with motivations, and let them go. It's about the same process you'd go through class, or with Dramatica or something. Not to say I've implemented much of this but that's the idea, and it appli aspects of *DF* design.

Now my brother, who isn't programming, has taken the bulk of the story writing role. He has a lot of fun churnir out, and then we look at them. I assume there are more impressive story generators, world generators, body pa etc. I'm just trying to put moderate versions all in the same place. It should give rise to some really awesome st the players themselves.

Actually, I can't point to a better world generator than *DF*'s.

TA: I can't either, but I'm the last person to ask. I've only played a handful of new games in ten years.

The nice thing about *DF* will be that the actors in the story have the whole, random *DF* universe to work with. So they aren't cutting-edge complicated, it should be awesome.

It must be heartening to see the Something Awful guys writing stories about the trials of their fortre

TA: Yeah, Boatmurdered had me laughing my ass off [laughs].

(Ed note: Boatmurdered is the story of a single long Dwarf Fortress *game played in turns by a large number of S Awful forum goers. It became a minor internet sensation some time back, getting as far as the front page of Me*

It's like Zach's stories are being retold, in a way.

TA: Yeah, even for the mechanics we haven't added yet, people just fill in the blanks. It's like that with a lot of g people think they are more complicated than they actually are. I remember having that experience with the grou *Falcon*.

Stories aren't the only way we plan. I think a few people had that idea. But it's certainly a fun way to plan that c

crystallize exactly what we want to accomplish.

The Wounds System & ASCII Graphics

TA: Yeah, we were role playing quite a bit, at least through '93. *Cyberpunk* was influential, as far as the damage goes.

The body-part system.

TA: Yeah, *Cyberpunk* did not have hit points. You'd get shot, brain damaged, in the hospital permanently, roll up character, back into it.

In most RPGs, hit points are something wholly replenishable. In *DF*, if a character loses a limb, he's r getting that back!

TA: Hit points are depressing to me. It's sort of a reflex to just have HP/MP, like a game designer stopped doing

While I think a lot of people who stick with that kind of system do it in order to simplify the game, it seem to be unexamined by many who use it.

TA: You should really question all of the mechanics in the game from the bottom up. I dunno. It really doesn't s conscious decision sometimes.

Of course, somebody with a game with dwarves and elves and goblins should be careful, but I think about that t think it's a bad decision many times, but that's assuming it was a decision. I just don't see it as the case. And th many equally simple things you could do that are more inventive.

***DF*'s way of doing it seems to work quite well for its purpose. Let's move on to graphics for the mom**

TA: If [*Dwarf Fortress*] had a fantastic interface and graphics tomorrow, it would really be something. I'd lose a people, but it would really be something.

Yeah, it'd be nice if they were there... on the other hand, I think ASCII has worked surprisingly well

TA: I've squeezed almost as much as I can out of the letter E though. It's really getting iffy at this point.

I can go over to something more *Moria*-esque. Sort of a class-based system, like "feline=f," instead of first lette being inline with reality with size = caps. That would get me a little more mileage perhaps, but the problem wou Not that small graphical tiles are any better.

But I don't entertain such thoughts for the time being, since *Armok* died this way. I just tell myself that *DF* grap like a storyboard. Once the storyboard is done, you can make the movie if you want, but not before. And with m specifications, we just keep going with ASCII.

"Losing is Fun!"

How about the "Losing is fun!" motto in the instructions and on the site? Where did that originate?

TA: Ha ha, that was originally just a throw-away line in the manual. But some people like it.

I think it's really an important thing to say, when I think about it.

TA: I put it there to try to create some comfort with the concept of permadeath. Reloading every time you fail w hamstring you in *DF*, especially later on. I've also tried to put out a concept of "playing the world, not the fortres in a sense it's a bit premature, since the best parts of the persistent universe are still in the works.

Right. Permadeath is generally a roguelike concept, of course.

TA: Yeah, I've seen some of those 7DRL contests. It's too bad it takes so much time to undertake a large projec really the only barrier for a lot of people.

Yeah, true. Still some of the 7DRL games are surprisingly good. *ChessRogue* impressed me to no end

TA: I like short projects. I used to have a thing I called a three hour knockoff. I think *Squiggles* might be the on posted.

That reminds me a bit of the Experimental Gameplay Project. To return to "Losing is fun!" for a mom suggestion helps the player to get over losing, so he'll leave behind exploration sites for Adventure n say one of the game's breakthrough ideas is how the fortress gameplay naturally encourages the pla create areas that are challenging to overcome in an Adventure Mode game. An excellent strategy in c becomes a dire challenge in the other.

TA: Yeah, that was the original premise of the smaller game, and it was one of the things I thought would be ve interesting from a metagaming perspective. On the Bay 12 Games forums, they've gotten together and are now "Adventure Park," where they each take a small portion of one world map square and make a for-adventure-mo Then they'll upload it, and anybody can play, wandering through the varied creations. It kind of reminds me of a This kind of sharing and moving between game modes is definitely fertile ground.

Do you think *DF* might become internet-capable in the future, and facilitate this kind of sharing itself

TA: I'd need help with that, since I've never made a multiplayer game. A lot of people are interested in this kind and it's kind of counteracted by my own complete lack of interest in multiplayer games, but something could ha community certainly finds a lot of ways to go multiplayer without my involvement though.

The World Generator

Let's talk about the world generator for a bit. If you wouldn't mind giving us a general overview of th the game goes through? No need to go into great detail.

TA: Sure. The first overall goal of the world generator is to create enough information to produce a basic biome lot of initial attempts at a world generator will start with things like "I need to lay down some forests, and some and some rivers, and some deserts..." and then when you end up with a jungle next to a desert, or a desert nex swamp in an unlikely way, it's difficult to fix.

So the idea is to go down to basic elements. The biomes are not the basics, they arise, at least in *DF*, from seve temperature, rainfall, elevation, drainage. First, it uses midpoint displacement to make an elevation map.

It also makes a temperature map (biased by elevation and latitude) and a rainfall map (which it later biases with orographic precipitation, rain shadows, that sort of thing). The drainage map is just another fractal, with values 100. So we can now query a square and get rainfall, temp, elevation and drainage data.

This is where the biome comes from. There's an additional vegetation field so it can alter the amount (from logg example), and there's also a "savagery" and a "good/evil" field. So for instance, if rainfall is >=66/100 and drain than 50, then you have a swamp.

The nice thing about having the fractally-generated basic fields is th biome boundaries all look natural. Part of the trick was to differentia like swamps and forests. There's also a salinity field, to differentiate saltwater marshes, etc. Just lots of basic information, so you don't t shoehorn anything into place. Just let it happen and these fields car potentially be altered.

I guess oceans get high salt as a special step?

TA: Yep, for now, it just sets the salinity to 100 at the oceans (ocea all the low elevation spots), and tapers it off quickly over the land s

You still get some artifacts from the midpoint displacement. It tends vertical and horizontal lines, so the next step is the erosion phase.

It picks out the bases of the mountains (mountains are all squares a given elevation), then it runs temporary river paths out from there, the lowest elevation and digging away at a square if it can't find a lo until it get to the ocean or gets stuck. This is the phase where you s mountain being worn away during world creation. I have it intentior on a mountain at that point so you can watch.

This will generally leave some good channels to the ocean, so it run rivers after this. However, some of them don't make it, so it forces the ocean after that, and bulges out some lakes. Then it loop-erases rivers, and sends out (invisible on the world map) brooks out from t During the loop-erase it also calculates flow amounts and decides w are tributaries, and names the rivers.

This gives us a world with biome data and rivers, but no actual life. So now it actually looks at the general biome each square (what I call a region type) and forms regions, giving them names. So you can have the "Silvery For is taiga in the cold regions and a jungle in the warm region, as long as it is connected.

At this point, it looks at the biomes available in each region and sets up plant and animal populations. This gives canvas for world history.

And from there I imagine it's just looking up the combinations of circumstances and looking them up of random possibilities.

TA: Yeah, I'd like to do more with ranges for animals, so that it's not so scattered. So that many of the northern have certain critters, and many of the western forests have others to kind of set up more of a geographical imag now, it goes strictly by biome type. It also sets up all the seed information around this time. There are some add structures it sets up, for features like cave rivers and so on. So it could set up a non-local feature like a world-w tunnel like this.

It wouldn't be so complicated to have the computer generate the races itself, though the lack of familiarity could jarring. *Armok 1* did this.

Very cool idea, I'd say. But does this mean we won't have dwarves to play in the future?

TA: Nah, I like having a stock universe around. It helps people get into the game, and they can stick with that if

I do like the idea of there being unknown species to discover though.

TA: The intermediate part of this might be the most interesting. Like when an evil wizard creates a randomized r over several games it actually forms towns, starts wars, etc. Then you play one in adventure mode. Lots of poss here.

I've kind of put some of that to the back burner while I work on the basics, like the current Army Arc. But it's re that I could start working on that even now if I wanted to. I've always wanted to be able to do some of this stuf

---

Deeper Details

Anyway, let's move on to what I expect was one of the biggest challenges in developing the game: tl amazing pathfinding.

TA: Is it amazing? It's creaky and slow.

Well it looks amazing from my end, since there's a metric ton of characters all doing it at once.

TA: The dwarves themselves mostly move around with A*, with the regular old street-distance heuristic. The tri that it can't really call A* if they don't know they can get there in advance, or it'll end up flooding the map and k processor.

(Ed. note: see here)

Sometimes in commercial projects the developers do a bit of cheating, and predefine travel paths.

TA: Yeah, that's the hard part. We can't really predefine areas beyond very basic notions because fluids can zip block them off, or they can mine a floor out. All it does now, and this isn't ideal, is keep connected component n if a dwarf is standing in "2", they know they can't get to "3" and don't bother trying. However, they assume they any other "2", and will A* those paths. There are still a few failures, but it's fixable.

There's a price to pay for that though, on a few levels. First, it's a pain to maintain. If a fluid occupies a square, update. If a fluid flows through a passageway and cuts it in half, it has to reindex one or both sides. There are o to think about handling it, like keeping track of some sort of rectangles or something, and pathing on those, but memory cost is too great.

The memory cost here is large, and it's also a speed burden. There are probably a ton of better ways to do it, it' very hard problem.

And all of this taking place over potentially many Z-levels too.

TA: In a 480x480xZ map. Or even a 768x768xZ at max settings.

And there's another huge downside to it that I have to cope with, and that's the mode of travel. A dwarf walks; does not fly. For a flying creature, the path components are meaningless. So flying creatures are much dumber dwarves, and dwarves are already kinda dumb.

I still stand by the adjective "amazing" though, based on how watching a dwarf make his away arou world, then suddenly disappear because he took an unforeseen shortcut over a hillside, is a cool exp

TA: Yeah, it's nice that they can solve an arbitrary labyrinth of your own design, and they'll pick the best possibl generally. There are some exceptions, like mining where they can mine from multiple sides, and don't try them

At its most complex, running *Dwarf Fortress* is like having an ant farm. You look in, and these little b guys are roaming all around, each with his own agenda. It's fascinating to watch for its own sake.

TA: Somebody modded their dwarves into antmen [laughs]. I even have a few people that went as far as donati they don't even play. They just like to check out the stories.

Fluid dynamics is one of those features that, to look at it, one can't help but wonder, how on earth di that? I think the answer to that question would be of great interest to the authors of falling sand gar

(Ed's note: We're talking about the most recent versions of Dwarf Fortress*, that use a three-dimensioned cellula automation system to manipulate water, much like a multi-plane version of falling sand. In it, each space is a cu can contain from 0 to 7 levels of water. A "1" is the smallest amount of water possible in the game, and a "7" sp of water.)*

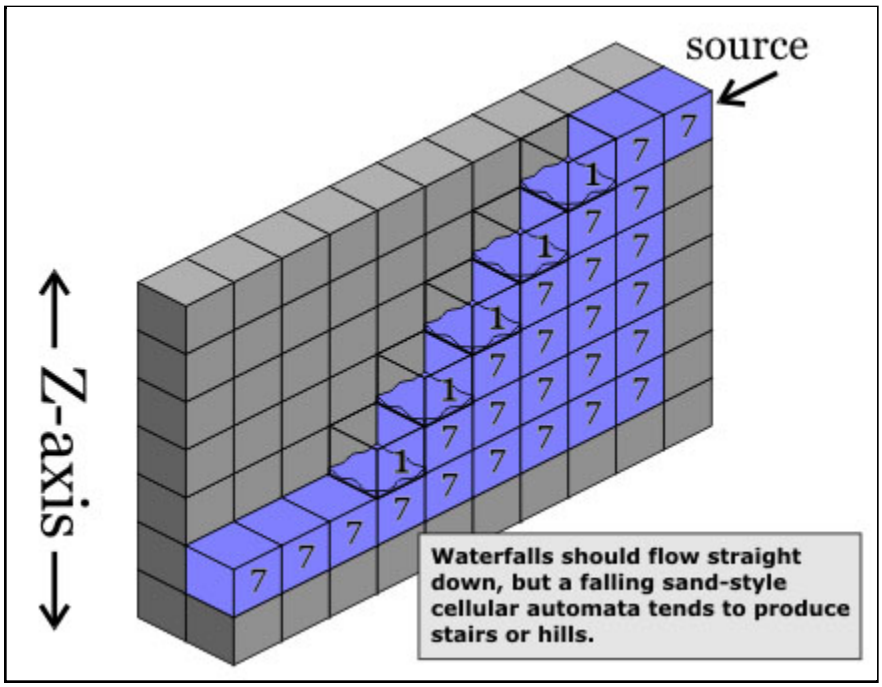
TA: I think the falling sand games are more impressive than my stuff. I've got kind of a bare-bones system. The impressive about it is probably that it runs at all while everything else is going on.

It's really not that complicated, just a specialized cellular automata.

TA: Neither is mine, he he. Mine isn't much more than that. Water falls down if it can, over if it can. The pressur interesting, I guess, since that isn't a local operation. There's probably a local model for pressure, but the ones I wouldn't work for me, I think.

Pressure is the thing I'm personally interested in seeing explained.

TA: The main problem was at waterfalls. The water would fall on to the river below, and just start clumping, forr pyramid. This is because of the local behavior. It can't go down, so it goes over.



Ah, I see. So the water is flowing out, but too slowly.

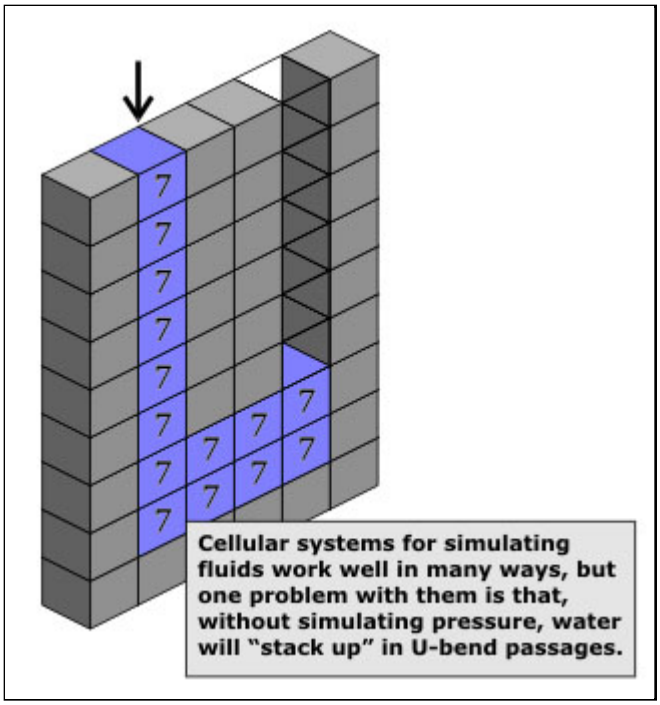
TA: You get a slope that naturally forms. And the hard part is that you really can't have any artifacts at all, it rea just look like a river. Anything unusual is bad, it can't do any pre-calcs based on the map, because the fluid syst work with water the player throws at it.

The water in the falling sand games I've played with does something similar when a great amount of lands in an area, it tends to sort of pile up and takes quite a while to settle down to an even level.

TA: This example was in a canyon or a cave river. If you are out in the open, it'll overflow the banks, which is al The cave river is the case where I started, the water is completely contained, but it still has to look normal. I dic problem, it doesn't do this anymore, but there are probably a ton of solutions. The way I did it, I can also model but it's a bit expensive.

What'd you end up doing?

TA: Let's say you have a U shaped tube, and the left half and bottom are full. This should not be a static situatic



Yes, in the falling sand model, water won't rise up right-hand side.

TA: The nice thing about this is that I'm not dealing with small particles, but large tiles, so I can afford some prett operations. From the local perspective, let's say it is the water at the top left to move. It can't flow anywhere, an flow down, so it checks the square below. If it is marked stops. If it isn't marked "static", it'll start doing a flood-fi downward, through full water spaces, never rising higher own level until it finds an open slot.

If it finds one, it teleports there. Otherwise it marks the below as "static" to avoid a recalc. If the situation ever c (like water is removed) it can just remove the static flag like it would be pricey but it's really pretty fast, and does for very much of the lag on computers having problems. happens at waterfalls.

The recalc elimination aspect helps keep it fast.

TA: Yeah, without the recalc elimination it would be impo the waterfall. Incidentally, I turned this off for magma or so it's all clumpy.

So what happens to, say, a tower of 7 water? I don't know if a tower of water can happen in game.

TA: Yeah, you could do it. Pour water into a tube walled off by floodgates in a stack. Then pull a lever and open once. There would be intervening floors I guess, on the sides, but it's close.

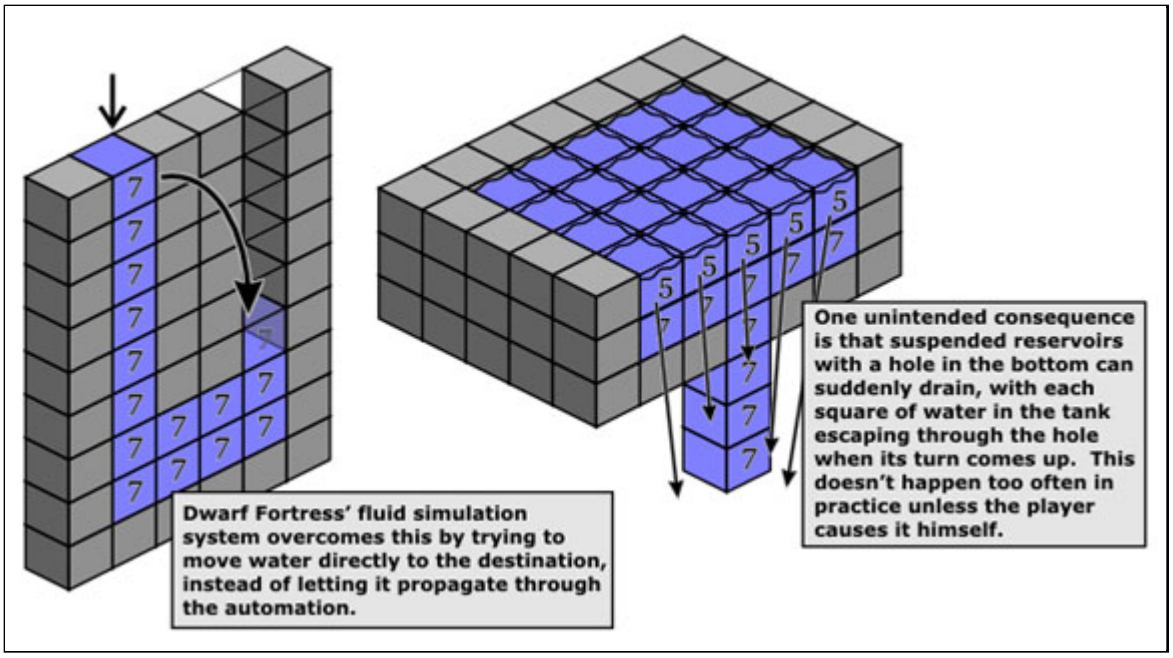
For the tower stack, if it started updating at the bottom, then it would do what happened in the first case, and tl above would just be able to fall down, then repeat. So you get a nice blob spreading out. If you start at the top, square below isn't static so it goes down, then it goes whatever direction, down, to the side, whatever, and puts once it gets to an open spot. So your 7s would teleport and the end result is a blob, but it has moved a bit faste

Some people will purposely open a water reservoir that is more than one deep. The water shoots out fast becaus entire second layer suddenly searches for an opening. You can get super fast flooding that way. Somebody had trap where they'd drop water on the gobs, and it happened to be outside in the arctic so it would immediately fr the gobs would die.

It also tracks which direction the water flows in a few ways so it can push objects and show flow animations.

So then, how does this cause water to flow up the right-hand side of that U pipe?

TA: When a water is doing a search for an outlet, it is allowed to go anywhere up to one below its original Z leve up the pipe.



I don't let it go the last step, or you'd end up with a three-quarter split that doesn't know how to stop. I have a improve that, but that involves expanding the static flag to a byte size counter.

And it respects physics, because it can't rise above its original height!

TA: Yeah, I was happy with it. It's crude, but it works, like the rest of the game. It would be nice to have a local those are hard to come by.

Return to the full version of this article