

On roguelike design

Based on a talk and intermediate at the IRDC 2008, Berlin. By David Ploog

Being a roguelike designer (not coder), I will focus on some points I noted over the time. Since my background consists only of NetHack and Crawl, I just have examples from these two games and I apologise for this narrowmindedness. I incorporated feedback from the IRDC discussions into this article. Thanks go to all who participated and to Markus Maier for proofreading.

I would like to mention that there is nothing absolute about goals or principles stated here. They apply to Crawl rather well, and it might be well possible that for some game, violating some of them is the way to go. I do believe that thinking about this is worthwhile in any case (i.e. which principles you consider important for your game). Also, there are exceptions to everything, even within Crawl.

Before starting, let me mention that the use of personal pronouns in this article is a bit shaky: 'we' means the Crawl devteam, 'I' is used for personal opinions, and 'you' is used for the virtual starting roguelike developer, who happens to be part of the target audience of this.

Major design goals:

- challenging and random gameplay, with skill making a real difference
- meaningful decisions (no no-brainers)
- avoidance of grinding (no scumming)
- (gameplay supporting painless interface and newbie support)

Minor design goals:

- clarity (playability without need for spoilers)
- (internal consistency)
- replayability (using branches, races, playing styles, and gods)
- (proper use of out of depth monsters)

Balance — randomness, challenge, and skill

In Crawl, balance essentially means challenging and random gameplay, with skill making a real difference. All words in this sentence are meaningful: the game is *random*-driven, so has a rather high variance. It can happen that the first few levels contain so much good loot that a good player will be confident about victory. It can happen as well that after twelve levels of misery, not a single shop or a piece of jewellery has cropped up. Both events are really rare, but there are no tries to narrow down variance here. The game is *challenging* in the sense that we would rather err on the side of the monsters, i.e. against the hero. The reason is that if a game is reliably winnable, then there is not margin left for comparing the best players. By having a harder game, this can be avoided (truth to be told, in its current incarnation, Crawl is lacking in challenge) and this is where *skill* enters the picture. Of course, one can ramp up difficulty pretty arbitrarily — our Litmus test against doing this is whether unspoiled players can win the game.

Crusade against no-brainers

Speaking about games in general, wherever there's a no-brainer, that means the development team put a lot of effort into providing a "choice" that's really not an interesting choice at all. And that's a horrible lost opportunity for fun.

An example for a no-brainer is weapon choice in *Crawl*: in most cases the axe is best, but it doesn't really matter because anything will do in the end. A typical example for a no-brainer in *NetHack* is the existence of the Restore Ability spell: it is completely superseded by the presence of unicorn horns (not to mention that there's the potion, too).

The opposite of a no-brainer is the tradeoff. If all choices come with some downside, then it is much easier to balance things out. If choices A, B, and C all address an issue but A is the most straightforward solution and B or C are never taken, then you could beef B and C. But you could also make A more costly (and there are always a number of ways to do so), allowing you much freedom for design.

Anti-grinding

Another basic design principle is avoidance of grinding (also known as scumming). These are activities that have low risk, take a lot of time, and bring some reward. This is bad for a game's design because it encourages players to bore themselves. Even worse, it may be optimal to do so.

The borders between tactical features and scumming are not set in stone. For example, running into corridors might be considered boring gameplay (and be punished), but roguelikes rarely do this. Or the use of stairs to shake off all predators could be considered clever or lacklustre. Likewise with pillar dancing. In all such cases, there is only one instance to lay down the law: the developer(s). If you feel that an action is always tedious and not fun, then go ahead and make it costly or plain impossible.

In the case of *Crawl*, collecting items and selling them to shops was deemed to be way too grinding. The solution was simple: shops don't buy. Many other solutions would have been feasible: for example, shops only accepting rare artefacts, or items to be so rare that each of them has value besides just a fiscal one.

Clarity

Whether spoilers are good or bad largely depends on taste. The crowd coming from puzzles or a roleplaying background will generally love to find out spoiler details. Here, we consider spoiler information to be important but hidden details.

Crawl's policy is this: The joy of discovering something spoily is nice, once. (And disappears before it can start if players feel compelled to use the spoilers right away.) The joy of dealing with ever-changing, unexpected, and challenging strategic and tactical situations that arise out of transparent rules, on the other hand, is nice again and again.

As a concrete example, recent versions of Crawl give away food quality: corpses are colour-coded in a way that shows if they are edible, poisonous etc. Here, a less radical solution could have been to show this information once the player has eaten such a corpse (this is akin to a monster memory). This would have been led to complications in Crawl, as some food properties do not trigger reliably — hence the simple approach.

Replayability

Replayability values of roguelike games are essentially always good, just by their basic properties: randomly generated environment coupled with permanent death and the ability/necessity to learn from mistakes.

For Crawl, I found that two other aspects help a lot: apart from the choice of class and race, there is the choice of god. And (sometimes related to this), there are several, fundamentally different playing styles. It is possible to slowly shift from one playing style to another, and in certain games, this might be even the best thing to do.

In the following, I present some notes about roguelike design coming from my experience. The order is pretty much random.

Spells as threats to balance

Spells can hamper balance in a big way. Here, I address spells of auxiliary character, not the standard damage dealing spells. The latter are easily balanced by costing Magic, which is limited. Those spells can be problematic which allow the player to bypass use of resources — in contrast to potions, scrolls, or wands, a spell can be cast in an unlimited fashion, given enough time.

CRAWL: Haste is a very strong effect, and can be invoked via potion, wand, god power and spell. The latter two are troublesome, although the divine haste at least costs piety.

CRAWL: Older versions used to have the Tomb of Dorokhloe spell which would turn all floor next to the player into rock walls. Coupled with controlled blink (another spell), this means life saving in two turns. Exercising conservative judgement about threats and using this combination liberally, players of the b26 version found the game to be of a completely different nature once the Tomb was available. After nerfing the spell twice it was finally removed. We reintroduced it as a card; it is still strong, but limited in use now.

CRAWL: Similar to the Tomb, the Banishment spell (or other sources of banishment) provided life saving by instantaneous escape into the Abyss. While this place is dangerous for many early and midgame visitors, it is a safe haven later on (this is a deficit in itself). Players have been observed to tackle Gehenna:7 again and again, banishing themselves at the slightest hint of something going wrong. Self-banishment via spell got removed without even trying to nerf it.

NETHACK: I believe that means of turning MP (magic power) into HP (health) are problematic. Whereas NetHack's Full Healing potion is a rare resource (barring farming), the Extra Healing spell is not so. Coupled with means of regaining MP, the unbalance should be obvious.

Less is more

While most developers are (naturally and rightly so) concerned with additions to their games, I would like to point out that removing features should always be an option. In my experience, two reasons for removal of features popped up: balance (see broken spells above), and lack of differentiation. If a game provides a variety of some thing (be that races, items, or whatever), but the individual choices matter little, then candidates for the axe should present themselves.

CRAWL: The game boasts a large number of species, but they still lack differentiation. Of course, the aptitudes are different, yet this may be not enough to provide genuinely different playing experiences. For this reason, Hill Dwarf and Elf had to go. (The developer of Steamband did the same with classes.)

NETHACK: Lack of differentiation for classes is an often bemoaned issue of NetHack. (There is a hint at this on the official website.) While all classes will be different throughout the first few levels (and the quests are obviously and well set apart), in the midgame (at least by reaching the Castle, barring challenge games), most characters look rather alike (the Monk may be the notable exception).

CRAWL: Most sources of amnesia (forgetting parts of the map of a level) have been removed, namely forgetfulness spells, traps and mutations. The use of a metagame effect (in principle, a player could draw the levels or make a photo shot) is bad in itself; in this case, there was added aggravation by interface tedium (re-exploration).

Beware the powergaming spiral!

The headline alludes to the following phenomenon: for various reasons (nifty new features, or striving for balance), the most recently added feature is often stronger than the already existing predecessors. For example, a weapon may be found to be too weak compared to some other type of weapon. Needless to say, after adding new weapons, item, and other resources in an unreflected way, the game becomes easier. In order to address this, new and tougher monsters are added. Iterate this cycle a few times, and the items and monsters you started with faded into meaningless. The first serious problem is that falling into this trap may render meaningful choices useless (before the addition of weapon C, players had to choose between weapons A and B, each with their strengths and weaknesses; afterwards, C is the best choice). The other one is running in circles, of course. The most famous example for (intentional?) spiraling might be the Nethack spin-off SLASH'EM.

One reason why new content generally favours the player might be that developers think (too much) in terms of the players. Likewise, player feedback is often centered around the character, and more often than not about shortcomings.

I'd like to point that commercial games often need the spiral as an economic foundation. This is extremely obvious with collectible card games, but also applies to many games that come with sequels or expansion sets.

CRAWL: Like the addition of new items, new levels or branches trigger the spiral: they provide players with more experience (more monsters to kill) and more loot. On the other hand, players and developers love new levels. In Crawl, we will add a new branch (the Shoals) in a way that avoids this issue at least: any given game will either contain branch A (Swamp) or branch B (Shoals). In this way, the expected amount of experience and loot per game stays roughly the same (well, lots of testing and micro-balancing is still needed to make this true).

Fight the option!

Many programs of a free origin (shareware, open source etc.) seem to suffer from infestation by options. Usually, new features indicate progress. However, in order to keep old players happy and/or to provide maximum choice, an option is added to trigger or tweak the new feature. I have heard two reasons for this phenomenon: First, by adding new features in this way, the code will generally be more elegant and flexible (with is a virtue in itself). Second, this style allows to improve the program while avoiding some decisions or discussions about intent or design. Third, options take care of the fear that the addition may prove to be bad or not accepted well by the audience – in which case it allows for easy going back (or ignoring the new parts).

However, I consider this to be misleading: especially in a game, developers should take responsibility. The most basic demand is that the default options work well (i.e. make it so that going back to the old, not as nice state requires option tweaking on the player's side). But even better would be to avoid the option in the first place. If you are concerned about the third point, it might be a good idea to hide your option from the player.

CRAWL: Of course, many options are just needed, for example choice of character set or allowing to remap colours. But to give an illustration of option inflation in Crawl, take autopickup. This is governed the options `autopickup` (what item types to pick up), `autopickup_exceptions`, `default_autopickup` (whether this is on or not at start), `autopickup_no_burden`, `pickup_thrown`, `pickup_dropped`, and `chunks_autopickup`. Of these, `default_autopickup` should not be needed at all — there is an in-game toggle for this. The last four should be made default or, if this feels too hasty, be turned into in-game toggles as well (which might be particularly appropriate for `autopickup_no_burden`). Then again, going through options and sorting this out is stressing out (and feels less productive than doing 'real' coding or design), which is why new options keep coming.

Transparency?

Discussions with other developers indicated that a flexible approach to player-developer communication should be pursued. For a new roguelike, player feedback is largely irrelevant. Well, bug reports and interface ideas are useful. But most gameplay ideas will just mean players (unconsciously) trying to get the developer to code up their games. Thus, feature proposals should be taken

with a grain of salt until the game has a defined scope and personality.

On the other hand, it will be useful to open up the source at some stage. Many players will be coders and provide bugfixes or patches. Again, one might not want to do this too early. In any case, open source will be a good way for expanding the developer team or for handing over the game (instead of abandoning it).

Finally, for the reverse direction of communication, it is preferable to watch players without talking to them. This is easiest done if server play works but can be arranged with hotseat just as well. It is highly instructive to watch both new players (gives lots of ideas on how to simplify getting into the game) as well as experts (nobody is quicker at finding exploits).

No matter how useful player feedback is, it should not take away from developer responsibility. Even good ideas might not fit the planned design; light-heartedly dismissing them (or just picking cherries) must be possible. On the other hand, completely disregarding players would be a feeble move: many players will produce a wealth of ideas, among some gems wait to be found.

In Crawl, we've got patches for large features (Vampires, monster list etc.) Gameplay ideas by players can also be found everywhere. Even if it looks like more work (sorting out the feedback and lots of communication for design; checking and tweaking the code for patches), it is worth it: the game is better than it would have been without this input.

What makes my game?

It is an old debate how to approach the creation of a new roguelike: by making an engine, or starting with a game, even if limited. Many developers seem to derive more satisfaction from the latter process. Here, I just want to mention that it will be useful if you know what your game is about, at least in vague terms. Not so much in theme (this is somewhat arbitrary in roguelikes and can be changed rather quickly) but in terms of gameplay and differentiation among roguelikes. I think it is more useful to say 'My game should address shortcomings x, y, z of classic [foo] in this way.' rather than 'I'll try to code [foo] again.' Even better would be some unique idea or vision. Many 7-day roguelikes have been started as experiments to stay away from the trodden paths.

Here is Crawl's statement: *This game aims to be a tactical fantasy-themed dungeon crawl. Strategy should be a concern, and we try for exquisite gameplay and interface. However, don't expect plots or quests.* Note how generic it is. But it still tells you something about its place in the roguelike world.

Gameplay > Interface > Realism

This should be a given, but let me point out an interesting addon. When trying to improve the interface, I found this process to trigger gameplay changes as well; often, by making things simpler.

I give some examples from Crawl: There used to be brown stairs, which nobody really understood. When we finally sorted out this interface mess,

the new brown stairs (now called escape hatches) are easily understood. Now we begin to see that they are used rarely (because normal stairs are strictly superior). Thus one can start wondering about their gameplay value? For example, hatches might disappear upon use, but no one would follow upon taking one.

There are a number blood gods that have to be appeased by kills and sometimes also by sacrifices of corpses. In former versions, it was necessary to pray before kills. This was changed; now any eligible kill is accepted automatically. This started simply as an interface improvement: having to pray all the time is pretty tedious and there is no choice involved. But by this change, the prayer effect was free and could be used for a different effect. (Not yet done, but an option.) Also, this change was the start of some thinking about how to separate the gods better, notably in terms of getting piety. And just because of wanting to avoid pressing 'p' all the time.

Another is targeting: Crawl has targeting in any directions, not just the eight cardinal ones. For this, the targeting interface needs much care. When we started showing the actual beam path (a pure interface feature), questions popped up? Should we indicate the range of spells? That range is highly variable — but why? In the end, we may redo spell ranges in order to use that for balancing spells better. (Range is little used for that end by now.)

Random remarks

When thinking about large new features (god, branch but not spell or weapon), I found it useful to look for a strong and cool theme. Only after that I consider how to possibly realise this with the game mechanics. And when contemplating which of several potential implementations to favour, I found that simplicity generally wins. One reason is that simple mechanics are easier to grasp for players; another is that they are much more robust in terms of balance.

Let me close by encouraging weird ideas. There is so much standard stuff around — try something extravagant!

Contents

Balance — randomness, challenge, and skill	1
Crusade against no-brainers	2
Anti-grinding	2
Clarity	2
Replayability	3
Spells as threats to balance	3
Less is more	4
Beware the powergaming spiral!	4
Fight the option!	5
Transparency?	5
What makes my game?	6
Gameplay > Interface > Realism	6
Random remarks	7