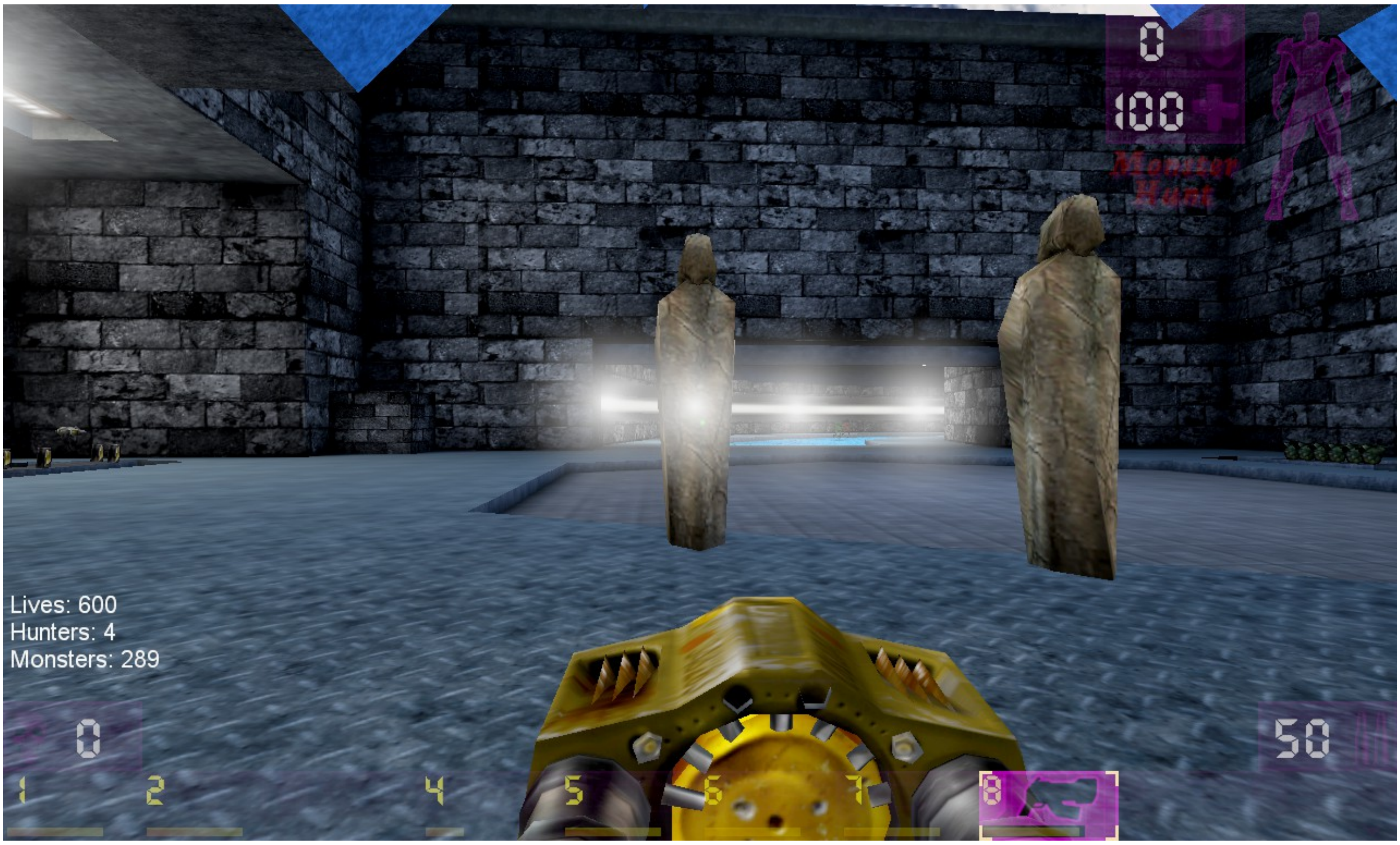
Coroner

- optional mutator for UT'99 -
- v1.0 -

Description:

This is an attempt to mitigate some rendering problem of corona lightning used in various Levels developed in UT. I will sample an edited one for clarification. Corona light it's not visible through movers and brush stuff generally, but it's VERY visible through statues, monsters, and other decorations bugging visibility with an utter FAKE view. Let's see first said screenshot from an edited Level for relevance.



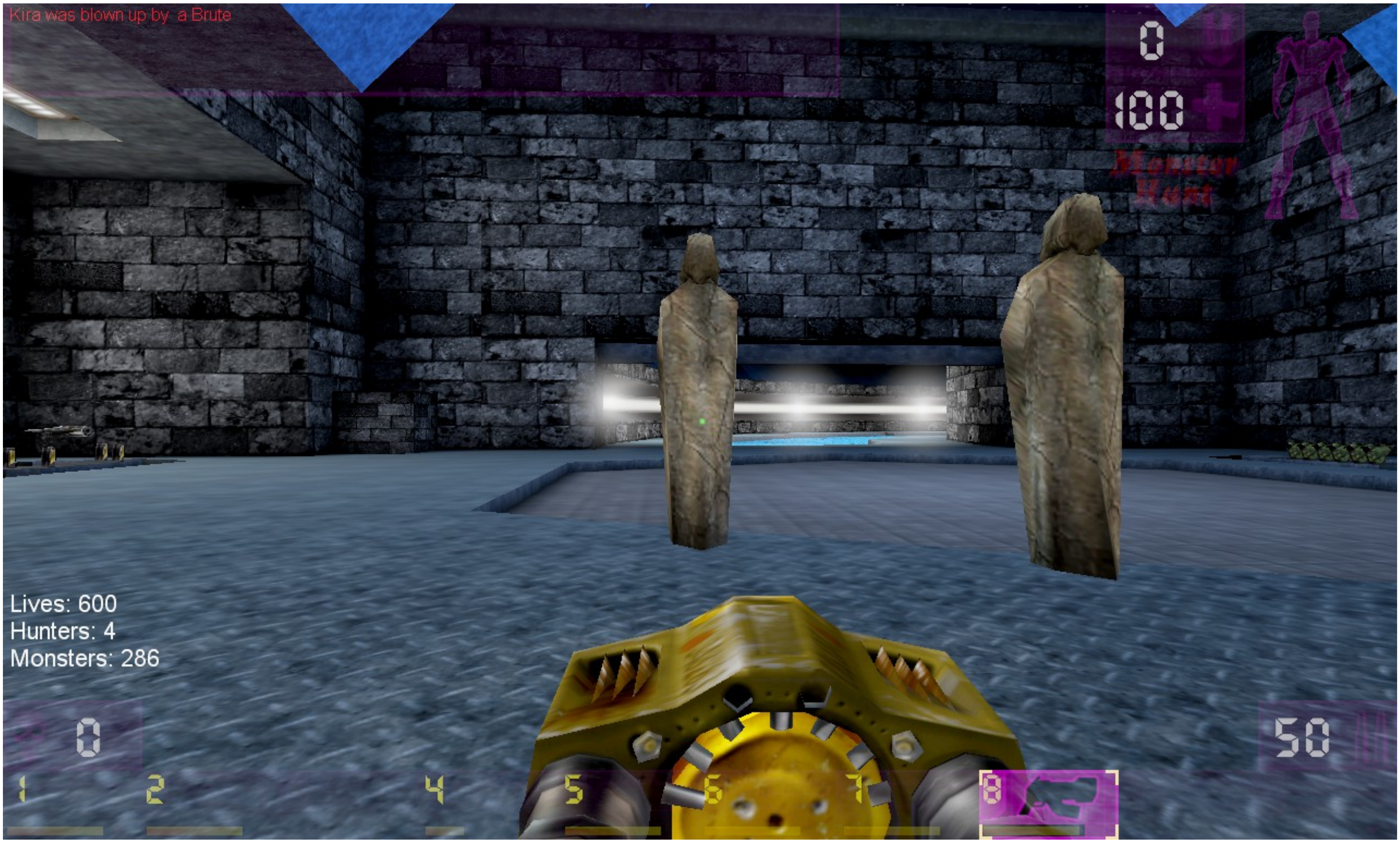
MonkStatue won't occlude this corona - this is D3D9 render but I did not see a big deal neither in OpenGL. In some tactical situation when game do includes nasty monsters, if these creatures are in front of a corona from your sight they bug your visibility making player to turn off coronas as default setup, not the last problem is the DrawScale used. In some maps are really ~~dumb~~ not well set, too big, way too big for no purpose, this stuff do looks good if mapper has skill for doing the right setup.

Tech Notes and Operation:

Mutator having assets reacting in client, because client is victim of a ~~bad ass~~ bad corona not the server, it will need to be known by player which means adding it into ServerPackages as primary need. Next setup doesn't really have priority rules, it should start even delayed (if my delayer mutator tool is used for preventing initial server-load). Mutator will start working after two seconds and will bring a notifier in Level which will have a native replication so here it's not used any new replication declared. All operations done by the actor notifier Coron bugger has hard-coded stuff and will react when it feels loaded in clients - logs are available in both sides - server/client. In dedicated servers won't do anything, it's sitting there for future new comers in order to trigger internal ~~hack~~ tweak to said fresh player when a copy of it will spawn in player's Level consuming very poor net resources when no one new is coming. Whatever is done to a light will involve only default Light class letting people with creativity to develop their stuff without to mess their work. Majority of maps have default Light class set as corona light. Measure is taken for not messing up TriggerLight class(es) (there are two sh!te in UT, one of them being useless), because it will debate these nasty lights using TAG. Tag is changed for Coronas in order to recognize them for activating/deactivating them depending on a trace with supposed Player Eye offset (to figure this well). When a trace returns an obstruction between light and Player, corona is turned off - it works a bit slow, this is UT render not me. All happens in a state code iterating through tagged lights in CLIENT so it won't lag server at all. As you might guess or not, we can speak about these obstructions as being not that accurate as long as actors are not always well configured by mappers - probably ~~various mental problems or~~ ignorance or whatever...

Considering that some maps have actors miss-configured in collision, this thing won't predict cylinder size in order to block light as should, but it's better than nothing... consider this as a cave-at which will need another special consideration if possible.

When this tool is operational, another instance of the game shown above will look like this, see below:



Kira was blown up by a Brute

Lives: 600
Hunters: 4
Monsters: 286

In image we can see that we do have not only adjusted these buggers, but even turning them off "occluding them" if an actor is located between player and a corona light. I think this second image it's way more relevant and more "real" than first image bleeding eyes - excuse me, but I have an age and I don't like any "eye's ninja challenge" right now, you'll see what I mean - I'm talking with you, teenagers, who think you are Godz but the time will show other stuff for you, just be patient...

If you ask why this tool works on purpose for clients, the answer is: because I did not see any bCorona value replicated from server to client in time for lightning stuff, by turning off a corona in server, client will never know anything about this, and static stuff won't be changed this way, yes this is a poor replication making some mapping stuff limited as well and requiring special coding. Some of these poorly replicated properties or NONE replicated properties are having their advantages. In this case tweaking task goes to player letting server to run its own tasks rather than taking care of player's specific tasks, client might control it's own in-game-image as should.

As another purpose this is not intended to show how to crawl player's Level, this is for adjusting ~~craps~~ miss-configured actors from Level in order to gain the maximum of quality as possible rather than the same ~~bullshit~~ junks since forever.

Setup notes:

Mutator might be saying:

- My name in mutator's chain is:

```
Coroner.Coroner
```

- My will in INI configuration is:

```
ServerPackages=Coroner
```

- Get rid of me if you don't need me.

The boorish coding techs:

Coders might correct whatever stuff from here if they have better deals, but it has lines than cannot be compiled by using normal UT environment - some coders know what is about. If an actor is bStatic intended to not be changed in run-time, making it flexible/dynamic it's not that easy, and then source-code can be created by decompiling this tool using Editor as long as new coders won't know what to do in case of changes wanted and then I won't bug them with a source-code that cannot be compiled. Dependencies used by this package are Core and Engine file, nothing more, nothing less. It will support UZ compression without to crash client after downloading it from a redirect like other tiny things are doing when UZ file it's bigger than decompressed file - lol EPIC.

Tests have been operated in UT using XC_Engine - this mutator is not hunting XC_Engine as a need, it should be operational even without XC_Engine, but for me this engine extension is a "stock", we were crying for nasty critical bugs, now they are solved, I don't have reasons for a return back into dark so I did not do any test in plain original UT.

Archive summary:

Coroner.u - mutator file;
Coroner.int - helper for loading mutator in list - Off-Line usage on main purpose;
Coroner.u.uz - compressed file for redirects;
Coroner.u.lzma - compressed file for newer redirects - XC Player as target;
Coroner_read_me.pdf - Heh, this is exactly the file which you are reading.