FOnline Lighting - some considerations by lisac2k

After a while I have installed and launched Fallout 1 and Fallout 2 and noted some differences in approach to lighting between these games and the FOnline engine. However, there are also some similarities between the two (or three) engines.

Similarities

First of all, the darkening of 2d graphics is done very well, use the screenshot below for comparison. Some differences in colours are obvious, because Fallout 1/2 uses a 229 coloured palette for its graphical presentation. This makes transitions between different "dark" and "light" stages (for example, different times of day or night) coarse and abrupt, but the overall transitions are almost the same as those in FOnline.

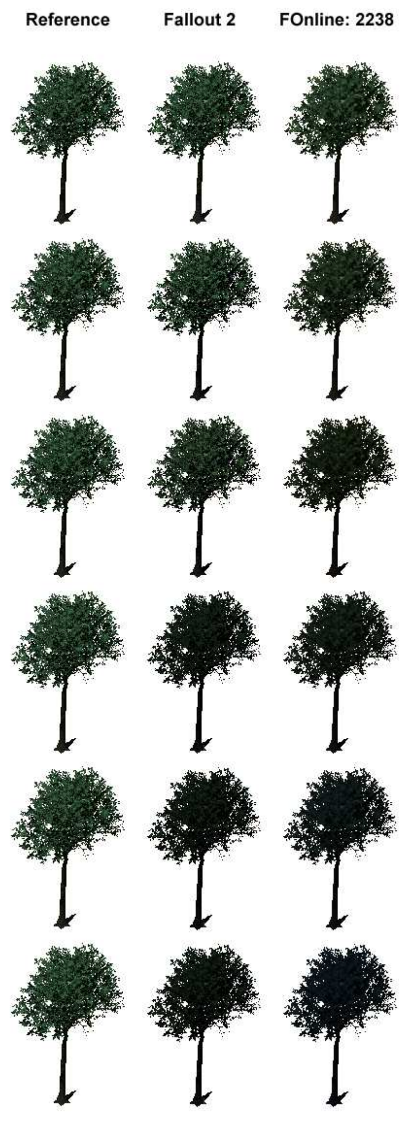


Illustration 1: Darkening of objects by Fallout 2 and FOnline engines compared to the reference (unchanged FRM); top=noon time, bottom=night time

The lightening of the objects for purposes of the day/night transitions works opposite of the darkening and is basically also done very well.

Differences

There are two big differences I want to point out, I hope both can be improved somehow.

The first one: the LIGHTS and the way they are "projected" (rendered) onto the other on-screen graphics, emulating light shafts. See the comparison illustration below:

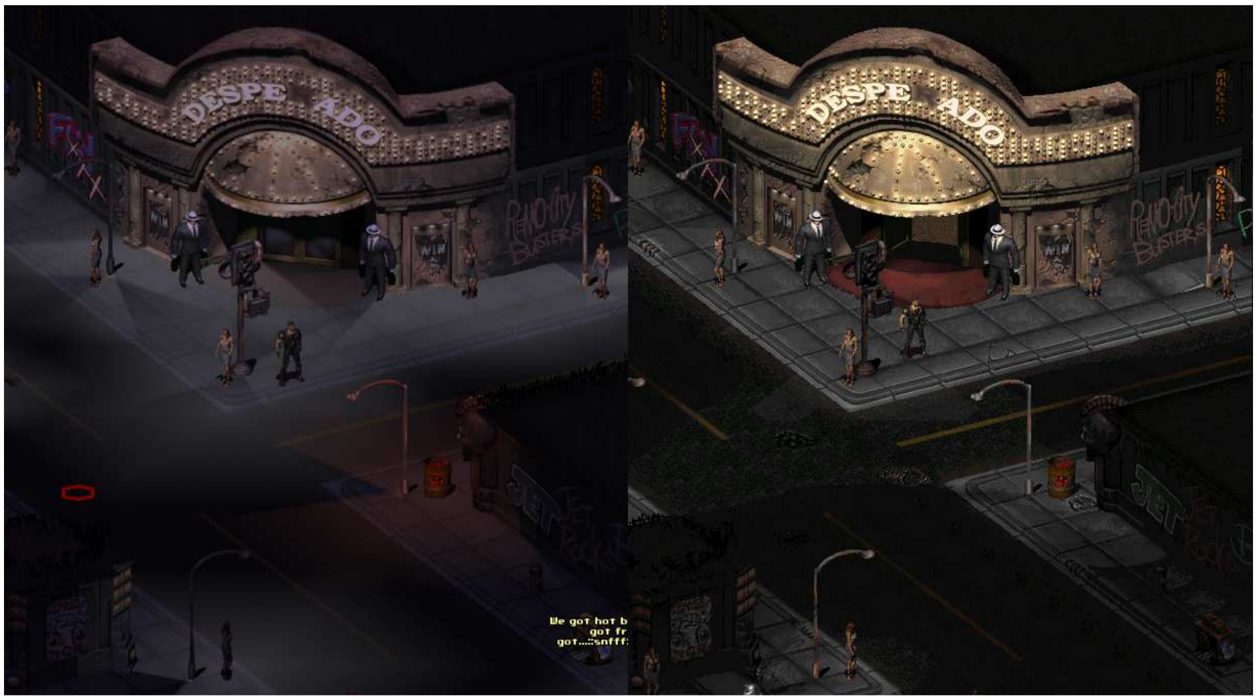


Illustration 2: Comparison of the FOnline lights (left) and Fallout 2 lights (right) taken at the same map/scene

I took the screenshot in New Reno, because there are a lot of lights there, so the comparison looks better than in other areas. What seems to be the problem with **FOnline** lights here?

1. The graphics under the lights (and light rays/shafts) look bleached, washed out and in the end there is no contrast (or at least not enough contrast) in these areas. When I say bleaching, I mean deficit of colours, turning vibrant colourful graphics into white/grey variations of those
2. The graphics are re-colored because of the light rays/shafts, which are usually adding blue-ish or other colors to the already bleached graphics

I do not know why lights work this way, but it reminds me of polygons which are drawn over existing graphics and some blending mode is set for them, for example transparency. I tried to make a similar effect in Photoshop (see the Illustration 3 below). First I created a white hexagon which is similar to the light hexagon around players. Then I set the transparency of the polygon to 12% and the blending mode to "normal". The last step was creating light "intensity falloff" near the light edges, so that transition between the light area and dark area is not too coarse (the effect looks just like the lights in the FOnline engine). It was done half-arsed in Photoshop and the quality of the work sucks, but is worth an example for this purpose.

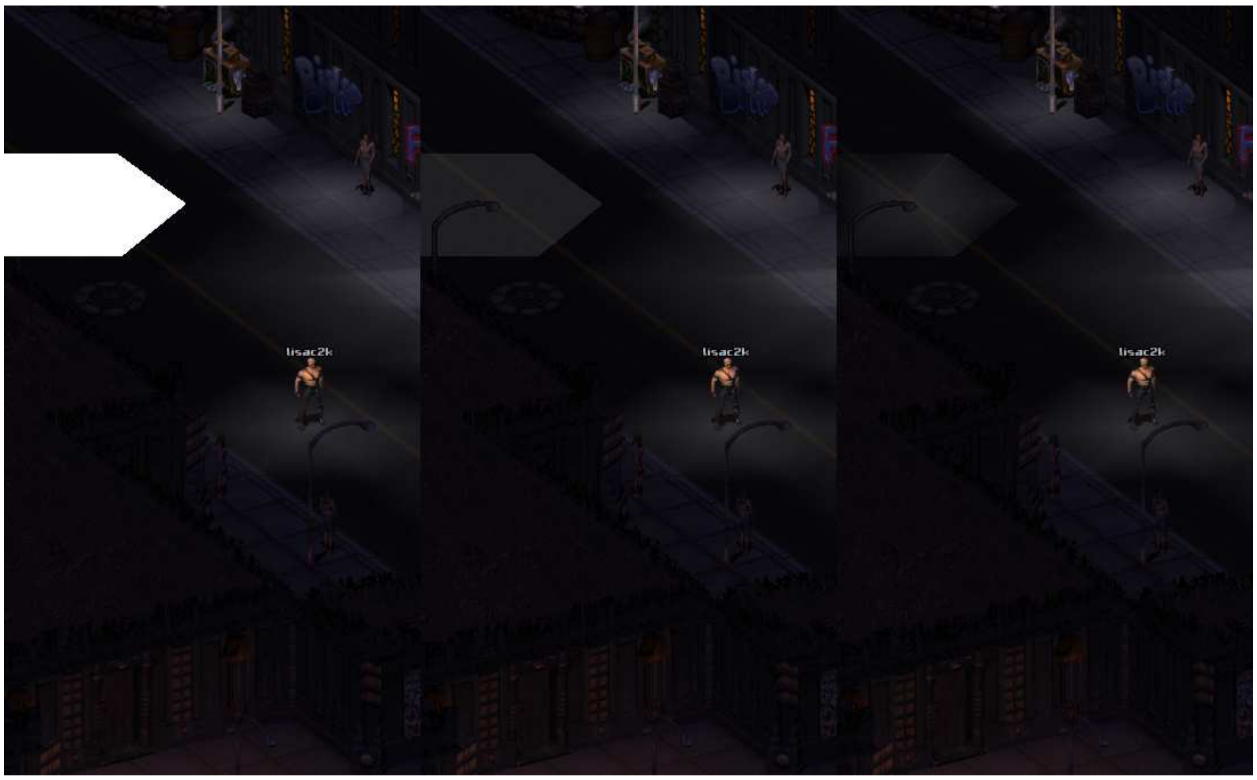


Illustration 3: recreation of the lights and light rays in Photoshop by using polygons with transparency blending mode (transparency=12, blending mode=normal)

It looks to me as FOnline engine first "darkens" graphics in order to represent night (for example). This works very well, as I said under "Similarities". The problem is when such darkened graphics have light on them - in this case the light shaft (geometry?) is overlapping them and it makes the graphics bleached, washed out, i.e. graphics lose their vibrancy and contrast.

Now, let us see how **Fallout 2** works with lights:

If you look at the Illustration 4, the first thing you will notice is that some graphics are not darkened at all! These graphics are drawn on the screen buffer exactly as they are looking in their respective FRM files. There is no darkening of these tiles, walls, sceneries etc at all! I marked these areas with red 100% limits, and it seems this is where very intense light allows player to see as if it was daytime.

Then there is another area which is a transition between light and dark areas, I marked this area with yellow 50% limits. These graphics are original FRMs, but partially darkened (approx. 50% darkening occurs here).

Last but not least there is the area with no lights at all, that is the rest of the graphics (with no limits created by me). These graphics are 100% darkened, i.e. they are located in a complete dark.

The values for this "darkening intensity" are defined in the Fallout 2 engine and I don't know the exact value, but reverse engineering or experimenting with values could help. Also, it seems that Fallout 1/2 engine lighting works on a hex basis (or on an "object" basis, if such object is piece of wall, door, etc.)

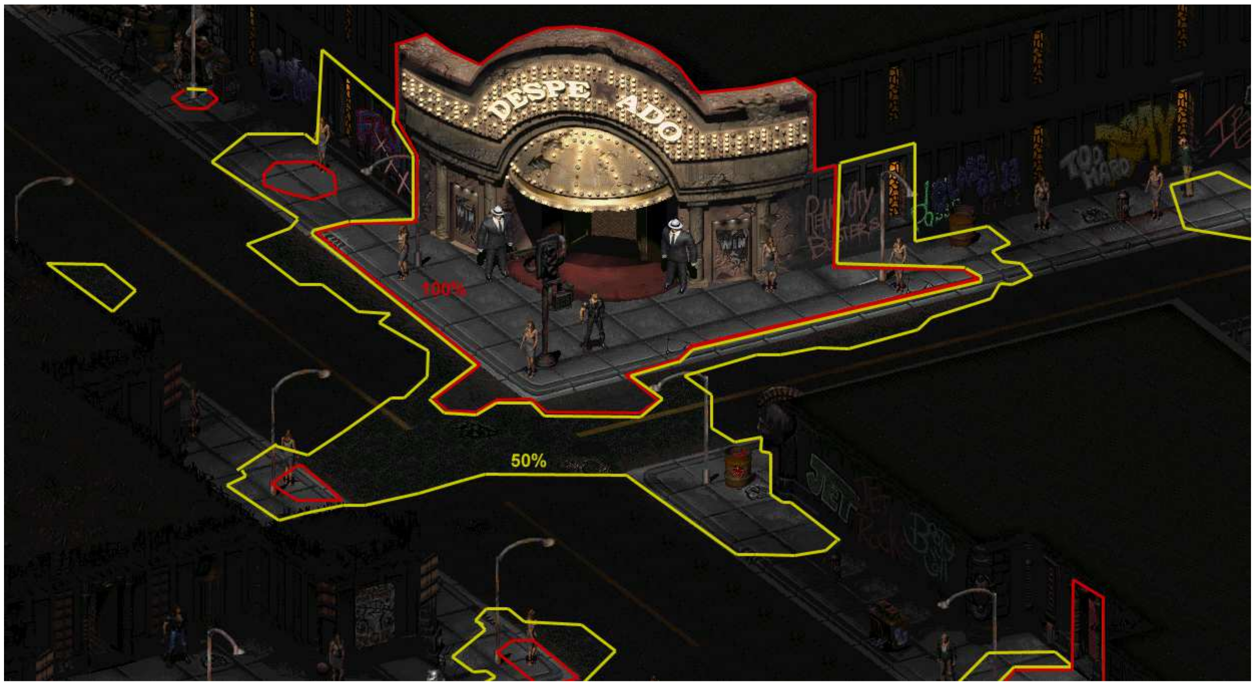


Illustration 4: Lighting system of the Fallout 2 engine, with marked areas with light (red 100%), transition areas between light and dark (yellow 50%), and the rest (which is completely darken by the engine because there are no lights in these areas); there are only these 3 different areas, everything is very simple and optically rather coarse, but very colourful and vivid

So, I thought I could try to **emulate** this lighting system in Photoshop, in order to show you how FOnline could handle the lighting in a more "vivid", more "colorful" way.

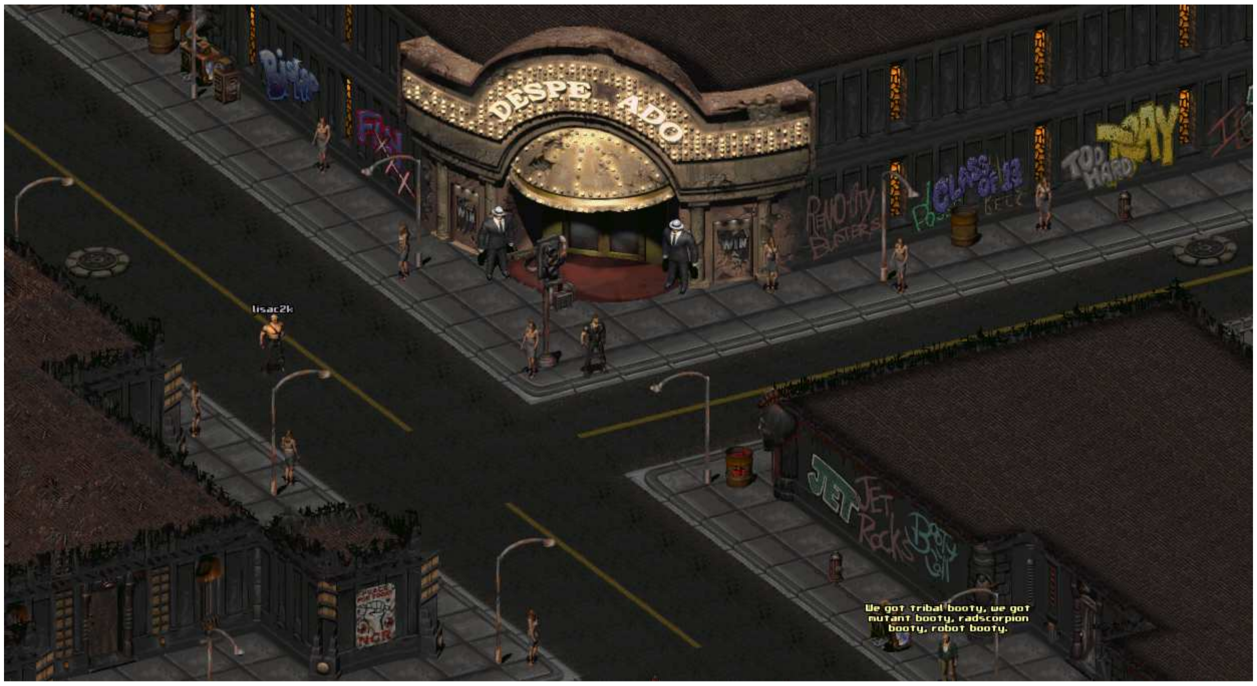
I first took a screenshot of the New Reno streets on noon (12:00), because lights have no impact on graphics at this time (it is daylight, the FRMs are rendered just the way they really are).

Then I needed the area which represents lights on the streets during nighttime. So, I also took one screenshot when difference between light and dark areas is most significant, at this time the visibility of the light rays/shafts is the best (it looks like this happens at 05:00 in the morning).

With a few adjustments and filters in Photoshop I managed to extract the areas with light rays/shafts and decided to organize them in 5 different light intensity areas (let's say 0%, 20%, 40%, 60% and 80% of darkening). There is no 100% darkening, because graphics would be totally black then.

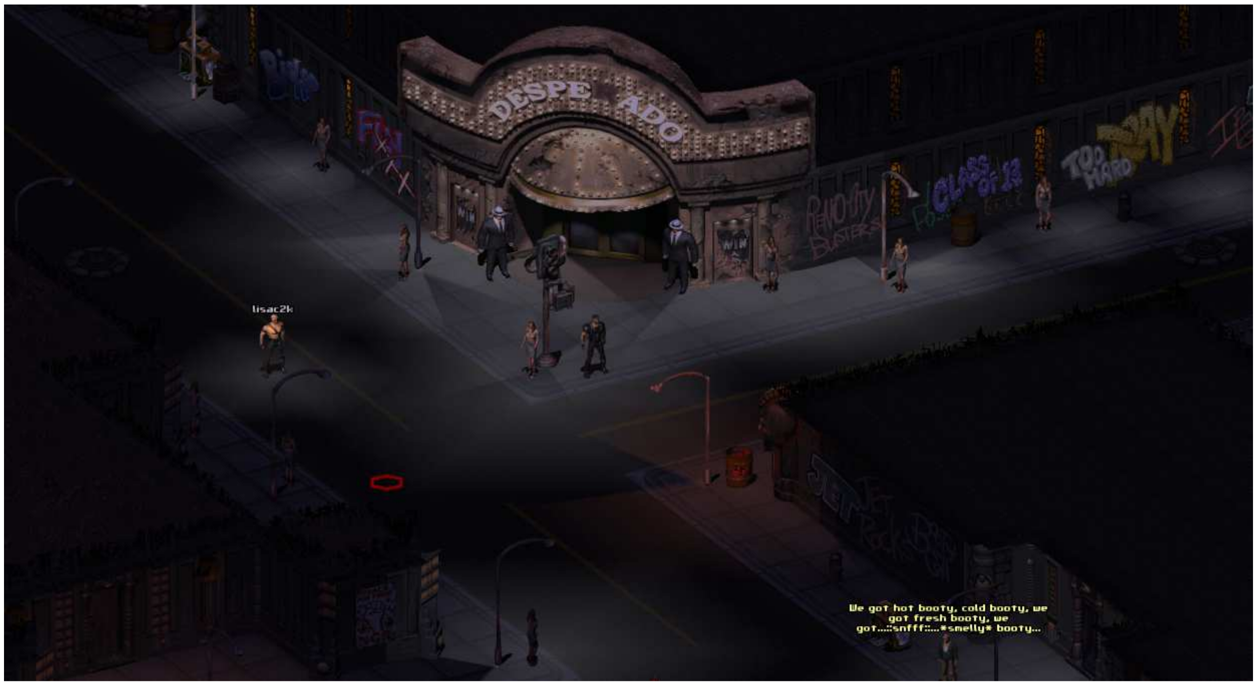
Anyway, this way I had something like an area with lights, which are represented with 5 different light intensity values and rendered in different areas of the New Reno streets. Of course, this was not exact amount of light rays which is in game (and it was not their exact position), but it was quite similar - and it was helpful for testing.

See illustrations 5, 6 and 7 for more info.



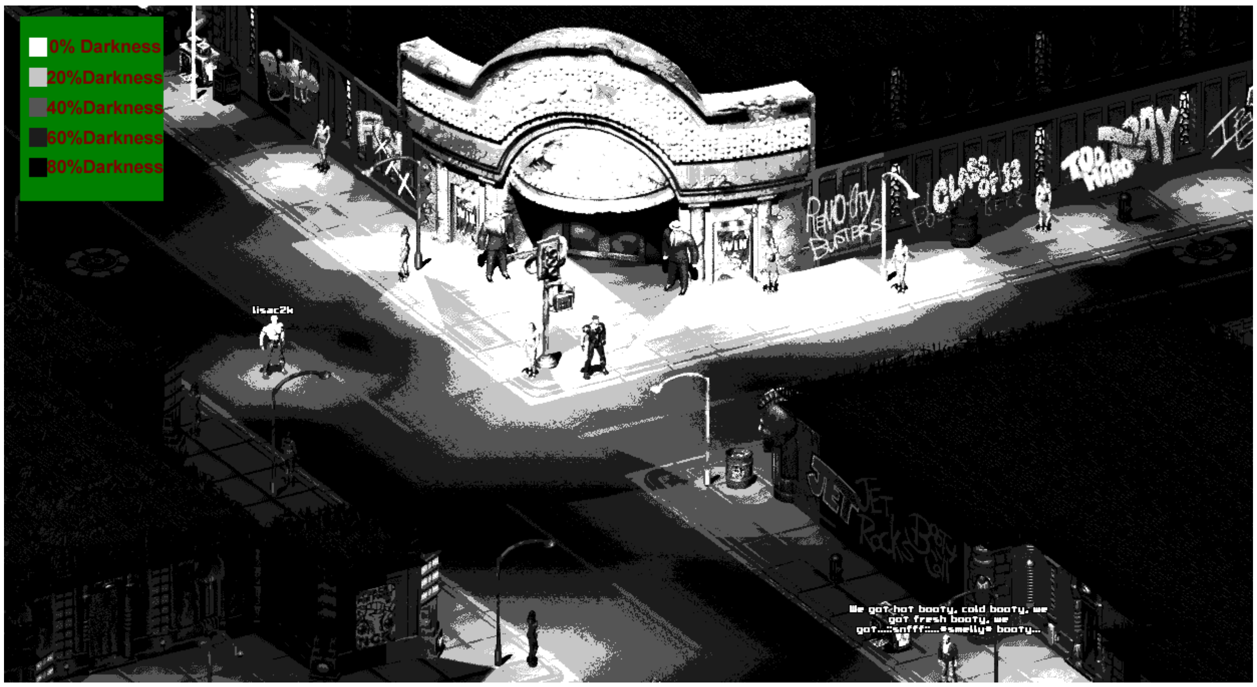
We got tribal booty, we got
mutant booty, radscorpion
booty, robot booty.

Illustration 5: FOnline screenshot at noon (12:00), it seems that the graphics are not changed by the engine (not darkened, nor bleached by light rays/shafts)



We got hot booty, cold booty, we
got fresh booty, we
got...::snfff::...*smelly* booty...

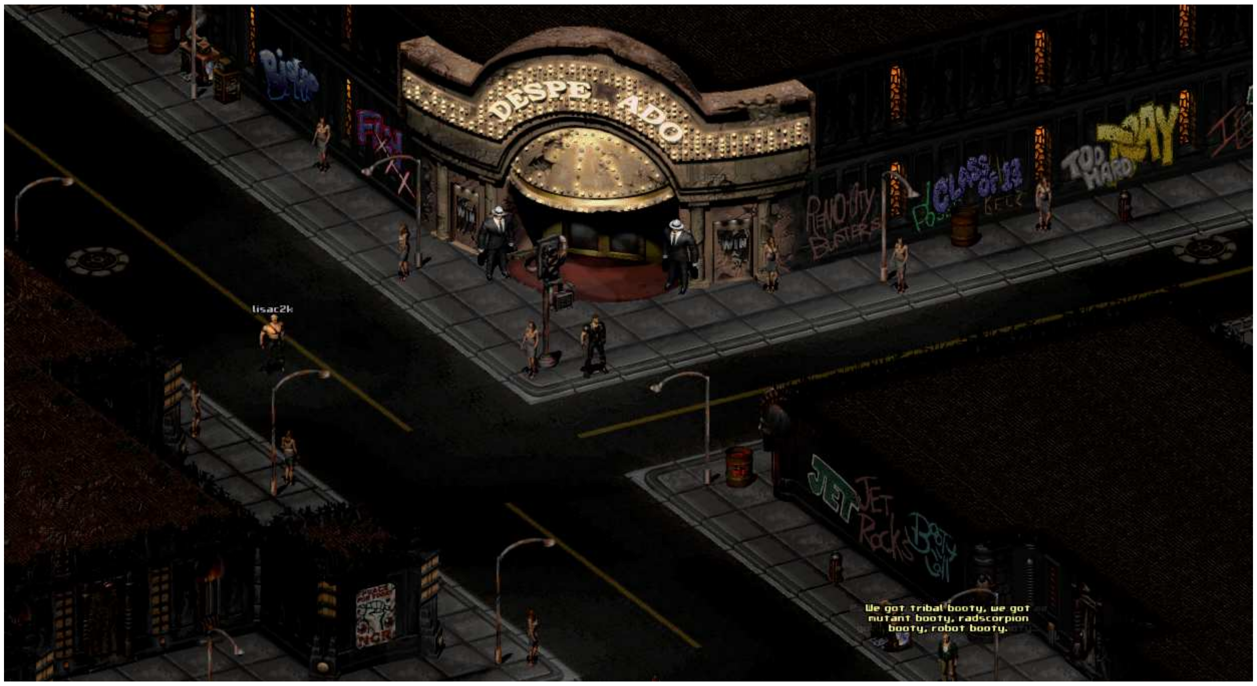
Illustration 6: FOnline screenshot during the greatest difference between darkness and light (05:00), the graphics are changed by the engine lighting algorithm (i.e. darkened in the dark areas and "bleached" by the light sources in the areas with lights)



We got hot booty, cold booty, we got fresh booty, we got....::snfff::....*smelly* booty...

Illustration 7: Light rays/shafts extrapolated from the screenshot at 05:00, grouped in 5 different groups (according to the intensity of darkening of graphics which happens on the screen), from 0% to 80%

At this point I had enough material to try different variations of lighting with Photoshop parameters like brightness, contrast and lightness. Let's see some of the results:



We got tribal booty, we got mutant booty, radscorpion booty, robot booty.

Illustration 8: Lighting emulated by adjusting brightness in 5 steps [-6, -12, -18, -24, -30] for 5 light rays/shafts groups. The brightness interval is [-100,+100], with 0 representing the neutral value.

As you can see in the illustration 8, I managed to keep the vivid and colorful graphics, while the areas with the most intensive lights were darkened just a bit [-6], because street lights are never that strong as the direct sunlight (they are directed lights, not a distant point light affecting a wide area).

Areas with less light were darkened partially [values of -12, -18 or -24], according to the ray intensity (i.e. light group) that covers these areas.

The areas without light are considered the most dark and they were darkened as the last group by decreasing the value of brightness for [-30].

Advantages: Looks much more colorful, there are no "bleaching effects" by light rays/shafts, brightness effect is easy to calculate

Disadvantages: Areas in total dark [-30 brightness] still have too much color. The color should go away with more darkness - because in the dark we see no light and therefore we also see no color.

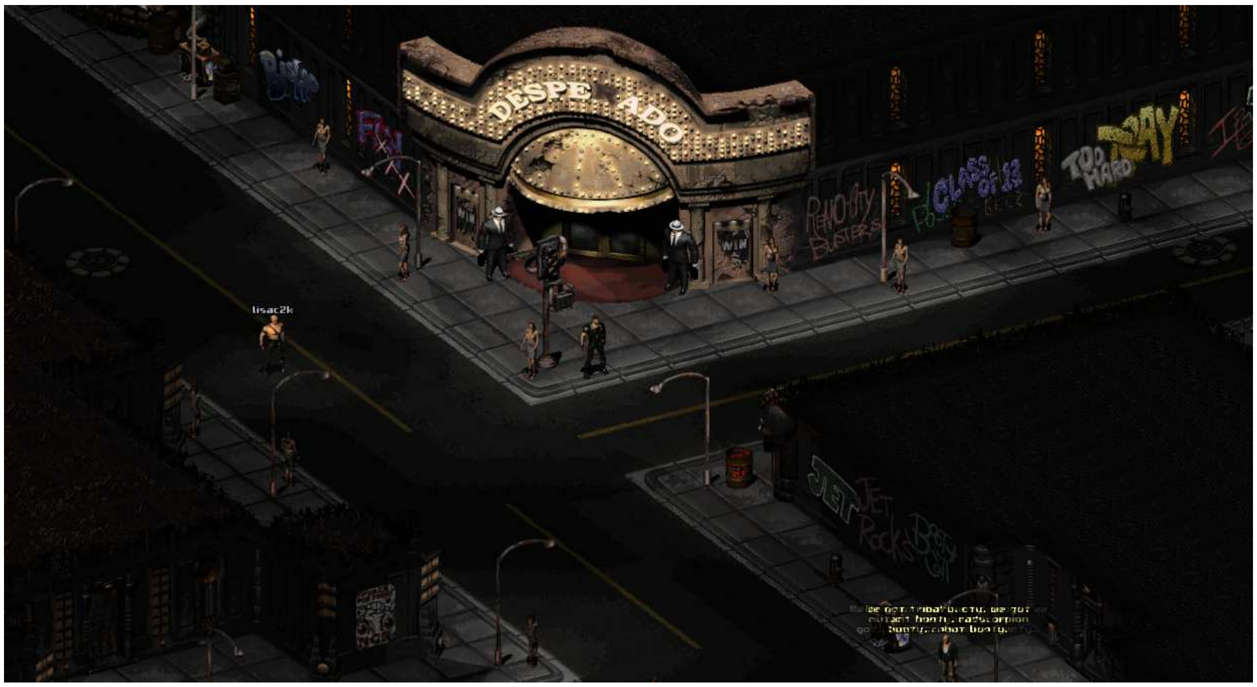


Illustration 9: Lighting emulated by adjusting lightness in 5 steps [0, -20, -40, -60, -80] for 5 light rays group. The lightness interval is [-100,+100], with 0 representing the neutral value.

Now, the illustration 9 was done by adjusting lightness (not brightness). The lightness not only decreases or increases the amount of light, but it also de-saturates pixels (it takes color away, there is less color on the scene).
The areas with strongest light were not changed [value 0], for the areas with less light there was some change [values -20, -40 or -60], and the light values for the areas in complete dark were decreased for [-80] in total.

Advantage: areas under light rays/shafts look good, there is enough color. One more good thing is that the areas in the dark have less color than in the previous method (by adjusting brightness). So, the dark areas now really look dark and with less color, just like our eyes really perceive.

Disadvantage: For adjusting lightness the conversion must be done RGB >> HSL (or HSV) color system, then comes the calculation, and then the conversion back from HSL (or HSV) >> RGB. It's not much of a hassle, but worth mentioning.

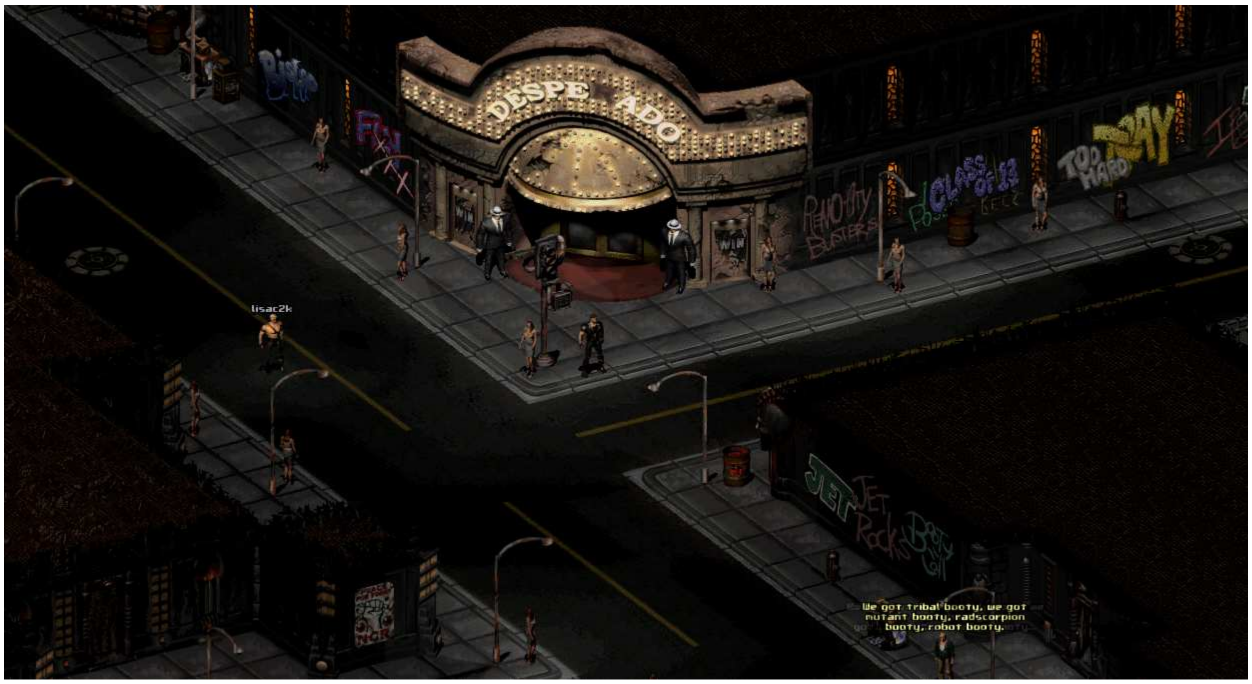

DESPE ADO
lisac2k
We got tribal booty, we got mutant booty, radscorpion booty, robot booty.

Illustration 10: Lighting emulated by adjusting brightness and contrast together in 5 steps [-6, -12, -18, -24, -30]. The result are (again) 5 light rays/shafts groups. The brightness and contrast interval is located between [-100,+100], with 0 representing the neutral value.

The last option was to change brightness and contrast together, because the parameter/effect contrast is also affecting pixels in a way of de-saturating and darkening (if contrast <0).

Everything was done the same way as with illustration 8 (brightness), the only difference was that the contrast was also adjusted together with the brightness effect. For the result see illustration 10.

Advantage: colorful, vivid colors in light areas; less color than brightness method (ill. 8), but more colors than the lightness method (ill. 9) in the dark areas.

Disadvantage: Photoshop algorithm for changing contrast is still unknown to me

Apart from the current lighting technique, there is also the **second big difference** between the FOnline and the Fallout engines**:** the day/night transitions.

In Fallout 2, the transition [day >> night] lasts 2 hours, e.g. 18:00 to 20:00. It works the same other way, for [night >> day] the transition is from 06:00 to 08:00. Of course, this is not always the same time during a year, because in the Fallout 2 scripts there is a specific code which shifts the transitions during summer/winter/spring/autumn. E.g. in June the night will fall from 20:00 to 22:00, and in September from 18:00 to 20:00 (IIRC).

One way or another, the nighttime lasts e.g. from 20 to 06, and it is a complete night (no change in light and darkness). Respectively, from 08 to 18 there is the daytime (with no change in light and darkness as well).

In FOnline, [day >> night] transition begins at 17:00 and each 2 hours the light and darkness is changed, until 05:00 (?). It seems that 05:00 is the darkest moment in the game. The other transition, [night >> day], begins at 05:00 and changes every hour until 17:00. This probably simulates the sun and the moon, as well as their movement, but maybe this could be improved along the way too.

However, this is not critical issue, so I will not bother anyone with this (for now), but it would be good to re-evaluate the way it works, for purposes of getting better lighting in the engine altogether.

So, what now?

I was hoping someone will reconsider improving the lighting system for FOnline, because I believe it will make game visuals better and attract even more potential users to create something using its SDK.

I'll try to provide a few ideas and possibilities what to do next (maybe you already know this better than me, but I will write it anyway). For the sake of simplicity I will use the term SPRITE for each piece of graphics we see on screen (and it is not UI), i.e. characters, tiles, walls, scenery etc.

It seems to me that the first and foremost fix for the current lighting system should be the proper regulation of the darkening (see ill. 3). By default, the Fallout sprites are made/rendered using light conditions which almost correspond to the noon daytime. They are rendered with one distant point light on the scene, which reminds of the sun positioned at 10:00 or 11:00 o'clock.

This basically means that the sprites should not be any "more lit" than they are. Currently, the sprites are darkened (for the sake of the nighttime simulation) and then additionally "covered" with inept light rays/shafts which ruin their visual appeal.

I think there should be a kind of per-hex lighting for each hex, respectively for each object on a hex, developed and implemented for our branch of the FOnline engine. This is maybe already implemented, but I can't tell it due to my limited coding abilities and no access to the source code.

Anyway, each hex (or object on a hex) could only be darkened, and not "lit up" by the shafts at all. **If something is located in a well lit area, then don't darken it, leave it be as it is** (remember, original sprites were rendered under well lit conditions).

For example, the light intensity can be a simple variable/integer within the limits [0,10], where 0==0% light and 10==100% of the light. So, a piece of wall under strong light (value 10) would be rendered as it is, while another one under dim light (for example value 5, i.e. there 50% of light in that area) would be darkened for 5 steps/units. Translating it into my experiments, this could be analogue to the Photoshop effect where the brightness and the contrast values were both reduced for [-15] (see illustration 10).

Again, I am not sure how the lighting system works, only pointing out current flaws and possible improvements in a pseudo manner. For me, the FOnline engine applies darkness well, so there's no need to change a lot of code to implement this, I guess (since we only need the code for darkening).

Further improvements could maybe include shaders (?) Using shaders the lighting could be represented on per-pixel basis, allowing for smoother transitions between areas and even better visuals. Not sure if anyone would be up to such a task, but if so - then this person could maybe salvage some of the existing code from the SweetFX project (http://sweetfx.thelazy.net, the actual repository is available on https://github.com/terrasque/sweetfxui) and adapt it to FOnline.

Color lights

Using the shaders approach could also easily solve the issue of the colored lights, because each pixel (i.e. its RGB values) can be changed along the way in order to simulate more of one colour (and less of others). For example for red light something like [R+32,G-32,B-32] could work well.

I am not sure how colored lights could be implemented without shaders. Probably by adding some color effect to darkening process, so when sprites are darkened, they receive more of one color, and at the same time less of other colours. Of course, "bleaching" should be avoided at all costs.

Some more thoughts

I do not know how any change in the current lighting system would affect the Z-ordering of the sprites, or rendering of 3D models etc. If anyone needs some more experiments, I can help. Just let me know what are you interested in and we'll arrange something.

I just hope we can achieve a bit better engine/SDK for the best of all of us (and other users).

lisac2k

Appendix 1: List of resources for this document:

Illustrations 1-10 (full resolution):
http://i.cubeupload.com/hRApOr.png
http://i.cubeupload.com/5MG2qn.png
http://i.cubeupload.com/FH1CL1.png
http://i.cubeupload.com/Kbd06e.png
http://i.cubeupload.com/YsMlvK.png
http://i.cubeupload.com/NCctWV.png
http://i.cubeupload.com/KYIwz9.png
http://i.cubeupload.com/JfLJn6.png
http://i.cubeupload.com/3b8orz.png
http://i.cubeupload.com/9fBg77.png