2016-04-02
DNr. 16/5

Chris Jacobsen (datahead)

Additional Meeting Topic Suggestions

Abstract

[A few comments by datahead on several different points on the discussion schedule of the 2nd General Discussion.
— note by Quintus]

Contents

1 Bugtracker Software (Topic 4)

I don't have an opinion about the best bug tracker software now, but I do not think the amount of RAM used should be a major factor unless we believe the RAM usage will cause impacting system issues in the foreseeable future. Yes, it is good for software to be lean and mean, but the features usually take precedence unless there are noticeable system issues.

Our major needs should be met by the software without resulting in a lot of workarounds. In particular, task dependencies would be very useful, rather than simply having tickets reference each other. Usually there are so many ticket cross references in GitHub that it is difficult to weed this out of the discussion.

2 Translation Software (Topic 5)

In my opinion, some sort of web based translation software would benefit the team. Getting the strings in front of a larger number of people through a user interface requiring no native apps be installed and requiring no understanding of a translation text format helps get more people translating in more languages for us. Given how hard it can be to find continual translators for some languages, this should not be underestimated.

weblate.org is better than no translation site, but someone has to do the work to set things up if we were to use it. transifex.com already had the set up work done for us by the Super Tux team. If we do not use it because it is not fully open source, is anyone willing to do the research work for weblate.org?

3 Life energy system (Topic 11)

Having a life energy system would be an alternative to the power up system currently employed in Secret Chronicles. We need some other model than having Alex shift to a smaller size when hit if we want to stop appearing like a Mario clone. Furthermore, it will be much easier to moderate difficulty. For example, we can have a large Larry boss that spawns a series of smaller Larries and goes through different modes of attack. This would tend to result in too difficult of a boss in the current game system (without some sort of auto replenishing power up).

The life system could be represented by a vertical (or horizontal bar). I think a solid color might work best, but it could also work with a series of smaller bars like many classic NES games used. The amount displayed on this bar is determined by the calculated percentage of Alex's current life energy divided by the maximum energy. Using a bar rather than a circular pie graph will prevent us from looking too much like Super Mario 64.

Power ups that change Alex's state, such as the fire berry, could remain with Alex until he loses a certain percentage of life energy. In this way, Alex would not lose a power up with a single hit but could still lose one by taking damage. Life energy regained would also restore the energy needed to keep a power up. If we were to use the current game system, in which Alex loses a power up with one hit, it would feel a bit like tag, which does not match our new life system very well.

Pits could be changed to inflict a certain amount of damage and then warp Alex back to a point. As Quintus has pointed out, if we use this model, we could have the level designers insert respawn points next to pits. Levels would have to be designed to be functional after player respawning, but this sort of issue is not that unusual in level design.

In the past we had discussed having temporary spells and permanent ”power ups”. This could work, though I think we should have some sort of power up(s) that can be brought from level to level and that certainly are not timed.

4 GUI system to use (CEGUI / SFGUI) (Topic 12)

Quintus has already started coding CEGUI changes, so we would need a really good reason to back out now to SFGUI. Given that he has spent the most time with this and is finding upgrading CEGUI to be a feasible option, I think we should take his advice unless he comes to another conclusion.

Also, given that CEGUI has a special, graphical editor, I think it could allow some of our artists to get involved in GUI designs, which would let us take more advantage of their design skills and offload a little work from the programmers. I think we should do some research in making this tool available/known to the non programmers. Yes, many programmers have excellent GUI design skills, but getting more people involved here will benefit us.

5 See also

See my story principles document [DNr. 16/2 — note by Quintus] for suggestions on all other topics.