



DEV DIARY 4

The Map 1

DEV DIARY 4

The Map 1

Hi Folks! Long overdue but here is the new dev diary on our map!

If you've seen the latest spudgun video as of posting then you already know much of what i'm going to showcase here so i'm going to try keeping it brief where it's already been covered. We do have some exciting new things to show you however, especially when it comes to procedurally generating the province centres.

So, let's get right into it.

Below is an example of our test provinces, these won't last long and more of simply a demonstration of various rendering features.



The Map 1

On this example you can also see the beginning of our province UI



This is a raw terrain map where we've experimented with tinting. We're also experimenting with the transitions between terrains. Extremely crude at the moment of course.



The Map 1

This below image demonstrates a very nice feature we have, that being a completely contiguous map with no seam down the date-line.



Graciously we've been permitted by the TGC modding team to use their map for testing purposes and I must say it looks pretty damn good!



The Map 1



Here you can see a side-by-side of the TGC and testing maps



The Map 1

By zooming up on our custom “Qaz” map you can see the level of detail is immense. We’re actually having to split this image up into two and sew them together again as some graphics cards cannot support this large an image



Other than that we’ve been working on a system to procedurally generate province centres on the whole map which should allow modders to just be able to paint in their desired borders as required and the game will worry about the rest. Something similar will be done with port locations, however for both the ports and the centres there will of course be a way to manually define them.

1. Use orwellian's propagative distance method where you take the lowest number of your neighboring cells and add 1 to it. do 4 passes on each province, each time rotating the direction of how provinces are processed by 90 degrees and only update the cell if the distance calculated is smaller than the cell's previous value. after the first pass you can ignore "0" cells for a tiny performance increase if you so desire.

2. Then you take the highest numbered pixel and designate it the "zinc center" (🧪) of the province which we'll need later. In cases where there is a tie for highest numbered pixel (such as a cluster of 2x2 which will be common enough imo) the top leftmost pixel is selected.

3. then we need to calculate the average position of all pixels within the province area which idk how to do and designate that as the "absolute" province center.

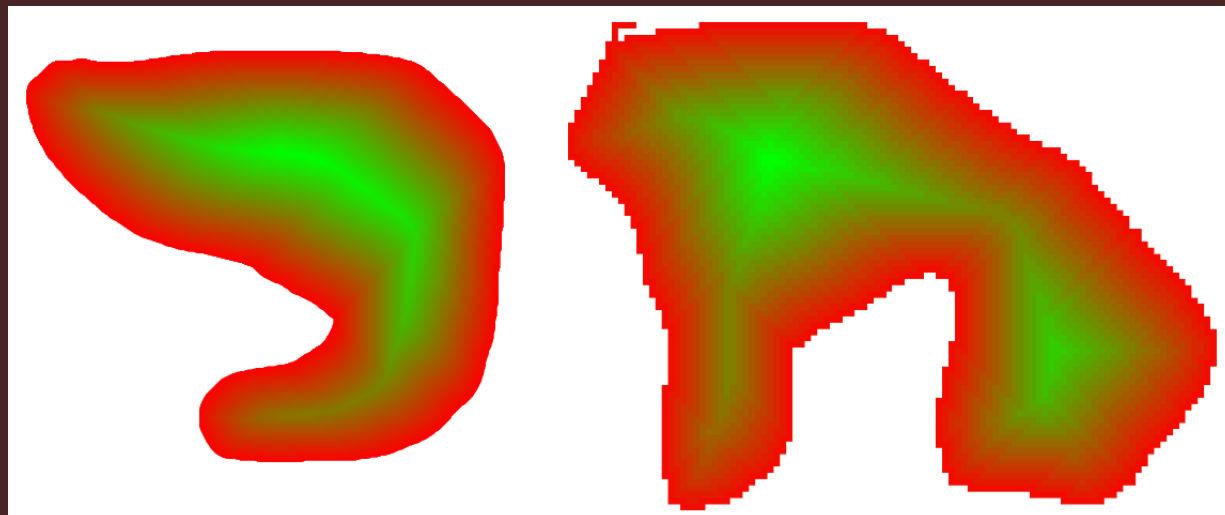
4. Then we take an arbitrary constant as well as the distance value of the zinc-center pixel and we use that to generate a vector in the direction of the absolute center. Because we use the value of the zinc-center pixel it is impossible for the vector to collide with the province borders unless the constant is wrong.

5. At the end of the vector we place the weighted province center which is what we use for the actual province center in game! (edited)



The Map 1

Here's some sneak peeks into the logic we've made (credit to Orwellian!) to calculate the best places to put province centres. (Weird shapes used on purpose for stress testing)



The code used to do this is ofc potentially very useful and can be used for so many other things.

And that's pretty much all we have for you this time folks! These dev diaries will now be following more or less a 5 week or so pattern, but ultimately they're whenever we have stuff to show. I hope you enjoyed both this update and the video spud put together, see you in the next one!