01 Jun, 2016

Robert Trzeciak

Program Manager

Northrop Grumman IS

468 Viking Drive

Virginia Beach, VA 23452

757-498-5544 work

757-635-2628 mobile

**Progress Report – 1 May 2016 – May 31 2016**

Contract Number: HSHQDC-06-D-00022

Contract Number 7500097279

Order Number: HSCG23-07-J-TED150

Task Order – Performance Work Statement (PWS) 1.12

Attachments: (1) SAROPS subcontractor financial reports

1. **More debugging. Here’s an example; The sticky shore was working, but slippery was leaking an occasional particle. Because of rounding, where a particle bumped into land was sometimes considered “land” and sometimes considered “water.” Rounding all but prohibits classifying a LatLng as being exactly on an edge. So when there is slippery shore, the particle runs into land, but then must back off enough to be in water. There were times that I wasn’t backing out enough and it was staying stuck on land. Note that we don’t want to back up too much because that might put the particle on land because of some other shoreline (e.g., the intricate and tight passages in the Columbia River or the Mississippi delta). I did get this fixed but the code does seem kind of complicated. I hope I’ve commented it well enough to follow the logic.**
2. **Because of item #1 and others, I re-did my cross product code. Cross products are very useful for determining whether or not a point is on or near an edge. This definitely comes up when deconflicting because one only has to check the points of one exclusion polygon against the edges of another. Exclusion polygons are all ccw so if all 4 points are to the right of an edge, there is no conflict, and the cross product of an edge can be used to determine that. The straightforward way of computing a cross product of an edge is to convert the two LatLngs to Cartesian coordinates and use the “(*i*,*j*,*k*)” trick with determinants. That’s about 3 lines of code and what I did before, but the accuracy was insufficient. Now I use the formulae found in the “aviation formulary” given in the following 2 sites (the second is for the cross product, the first is for various other formulas, including haversine and initial heading):** [**http://williams.best.vwh.net/avform.htm**](http://williams.best.vwh.net/avform.htm) **and** [**http://williams.best.vwh.net/intersect.htm**](http://williams.best.vwh.net/intersect.htm)**. In addition, I re-factored the code so that whenever I need great circle calculations, I go through these routines.**
3. **Buffers seem to be working now. My buffers have been bigger than my ½-trackspacing in the first few test cases, but that shouldn’t matter; I just haven’t had a chance to test them.**
4. **Wrestled with distress times and think I’m well on the way with that; there’s little I have to do since much of that has been pushed off to the gui side; In Jim’s words, the data “should be sanitized by the time it gets to me.” I won’t crash, bad data could force me to give bad numbers back. Right now, we’re just thinking that there won’t be bad data. I am supposed to re-draw a voyage for a particle if its distress time lies outside of the prescribed interval, and give up if that is taking too many re-draws. This is similar to restricting starting points to land. Should be able to knock that out soon, but will wait until other problems are ironed out.**
5. **My ESS+NVG is looking good and has been included in the last two builds.**
6. **I’m off the week of June 11th (So is Judy)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Activity Worked** | **Hours Worked** | **Hourly Cost** | **Total Cost** |
| Kratzke | Coding/Doc/Travel | 173 | 282 | 48786 |
| Vergamini | Coding/Doc/Travel | 0 | 282 |  |
| Stone | Doc | 0 | 223 | 0 |
| L White (Tech Writer) |  | 0 |  | 0 |
|  |  |  |  |  |
| **Totals** |  | 173 |  | 48786 |
|  |  |  |  |  |