Report: 25/30 (see comments below).

App: 110/120 - Some features finished after sprint close. Also see comments in code on dev. Overall good work.

Repo Usage: 50/50 - although I find it unusual that no PRs were ever declined - this is rare.

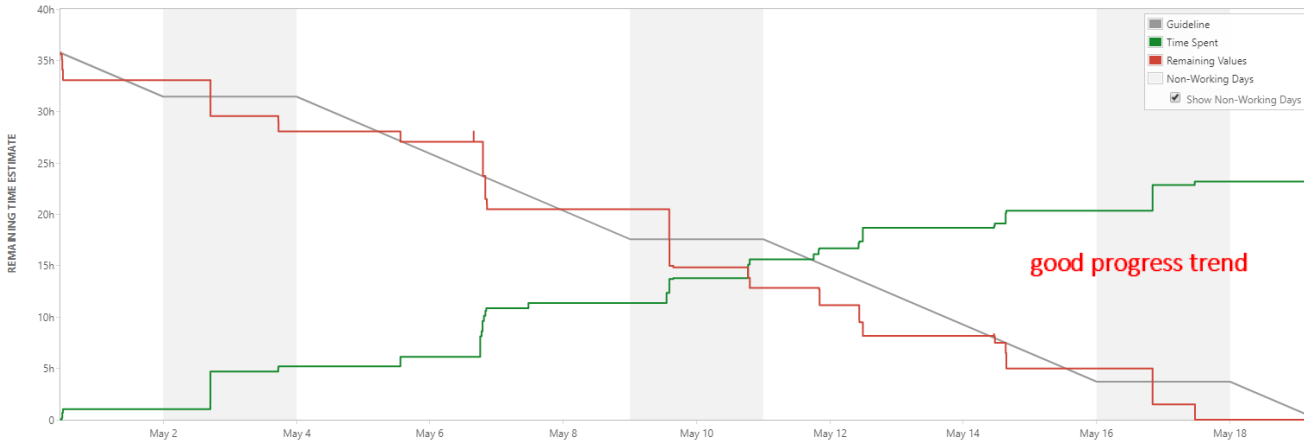
Sprint 3 05/19/20

Part 1: Sprint Review

Burndown chart

View your team's burndown by selecting Burndown Chart (for Sprint 3) from the Reports page of Jira. Use the built-in Windows 10 Snipping Tool - if you never used it, it's available from the Start Menu - just start typing "Snipping Tool" and it should appear. It's use is intuitive. Snip the image of your burndown and paste it below as a full-size image. NOTE: Make sure the burndown image you contains the correct team/sprint name at the upper left (Sprint 3), and that the x and y axes are fully visible.

Sprint 3 Burndown



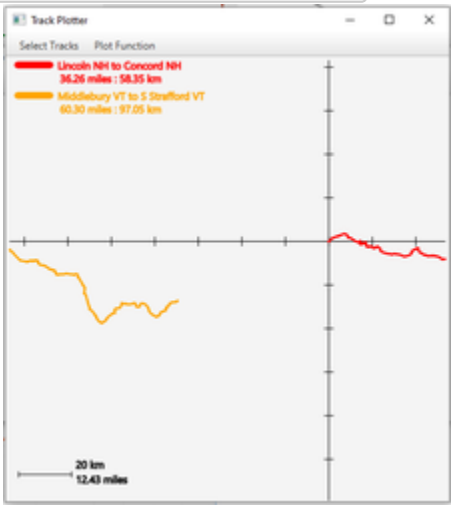
PBI completion status

In this section:

a. Explain which specific PBIs (*Stories, Defects, Internal Improvements, Knowledge Acquisitions*) were completed (**Done** after approved by the Product Owner).

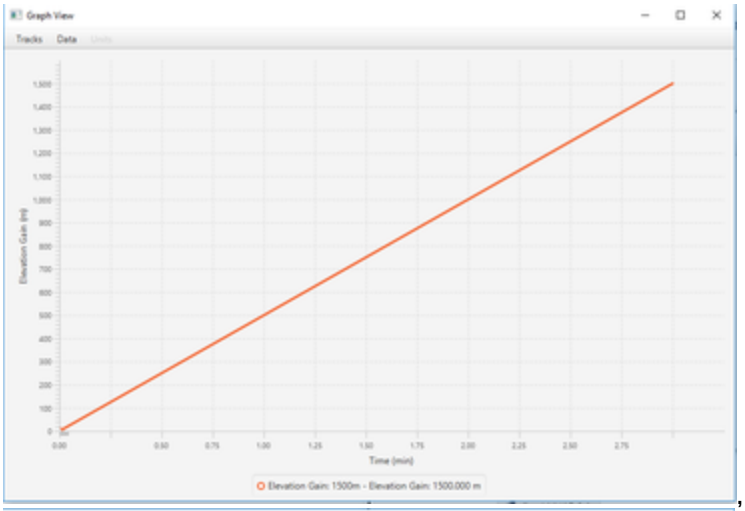
i. **MHA5-92** - 2D Plot improvements

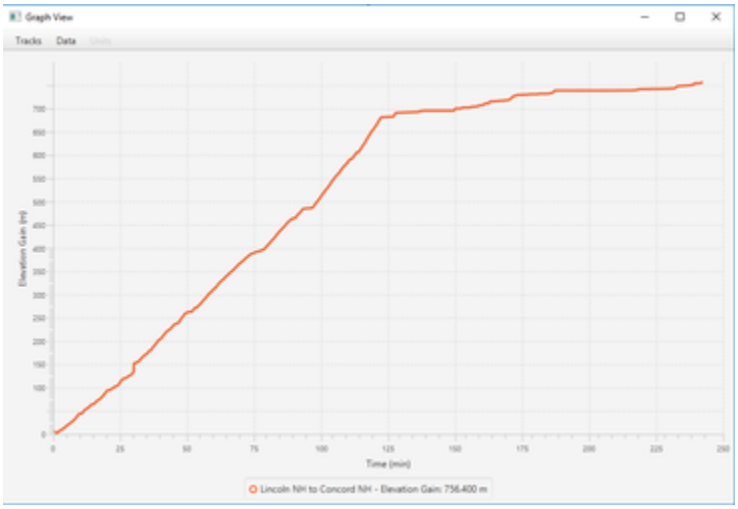
DONE

1. 

ii. **MHA5-13 - 6. Show elevation gain graphically**

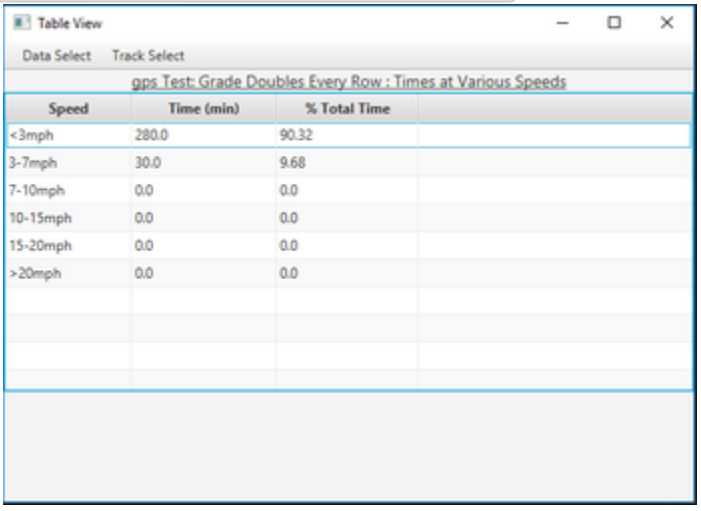
DONE

1. 

,


iii. **MHA5-3 - 15. Display Table of Times at Grades**

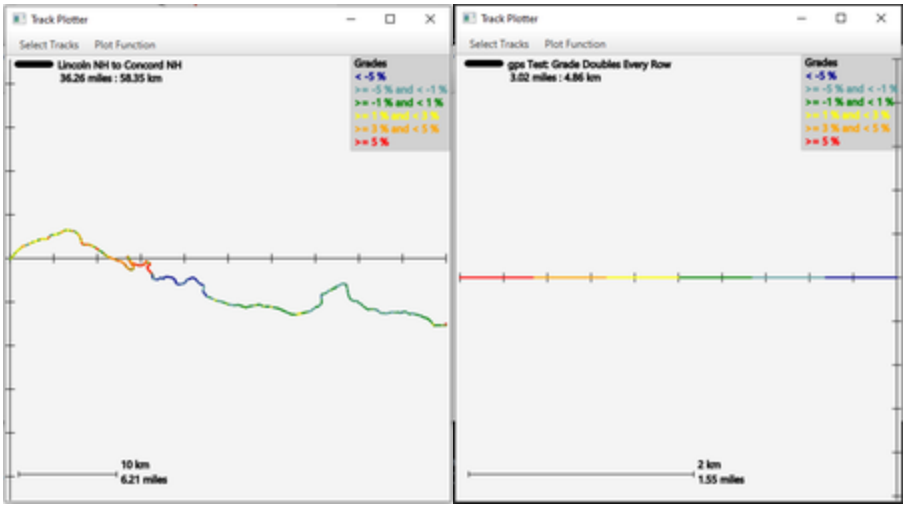
DONE

1. 

Speed	Time (min)	% Total Time
<3mph	280.0	90.32
3-7mph	30.0	9.68
7-10mph	0.0	0.0
10-15mph	0.0	0.0
15-20mph	0.0	0.0
>20mph	0.0	0.0

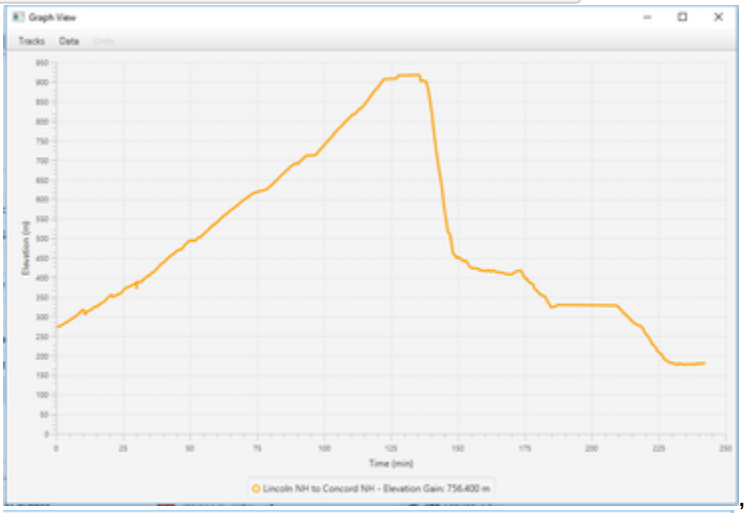
iv. **MHA5-20 - 14. View 2D tracks with colors indicating grade**

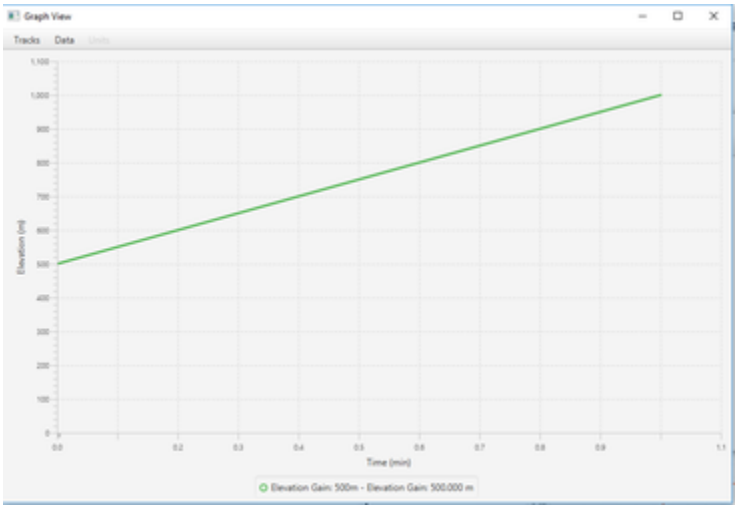
DONE

1. 

v. **MHA5-10 - 5. Implement elevation plot of loaded data**

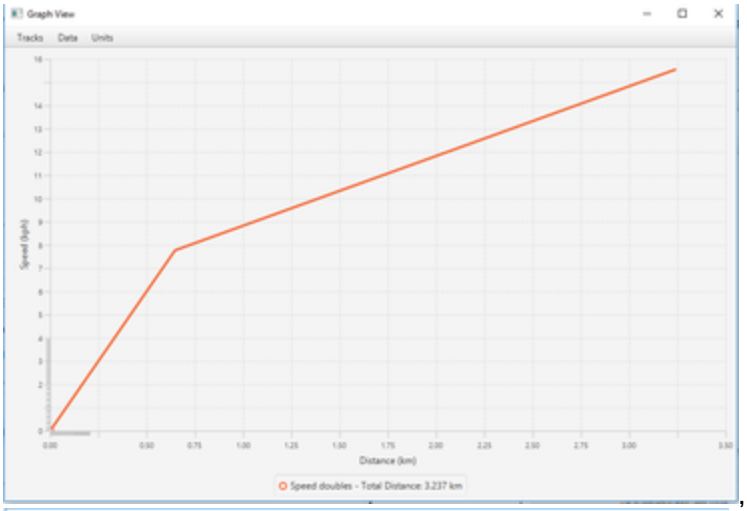
DONE

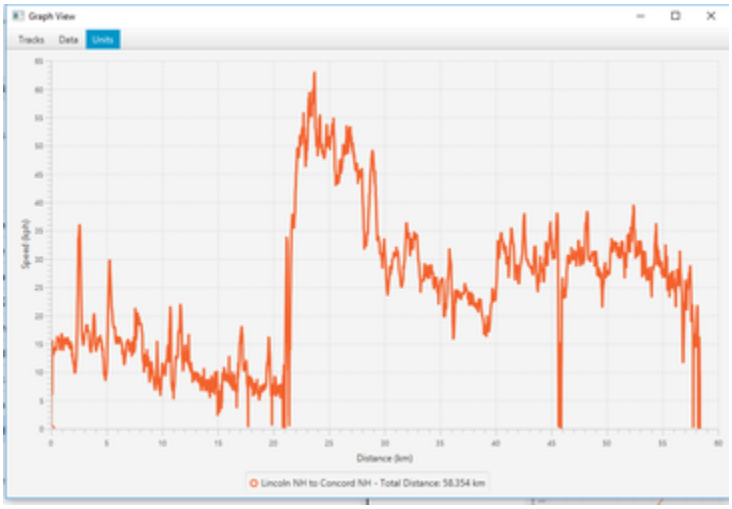
1. 

,


vi. **MHA5-14 - 11. Speed vs time plot**

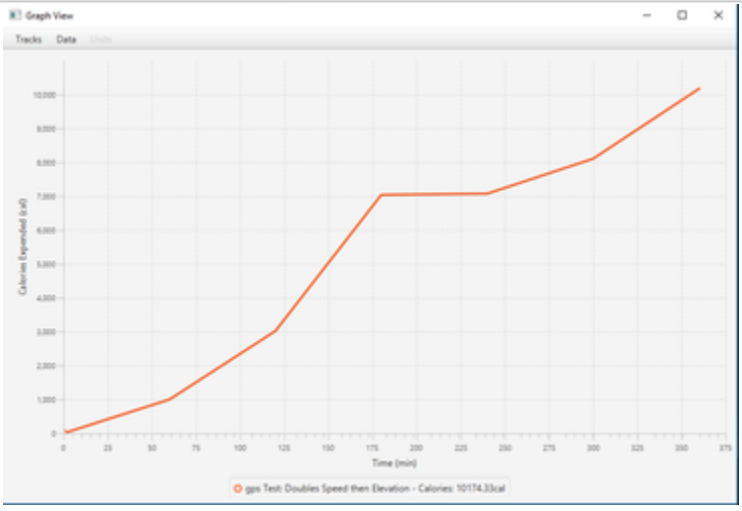
DONE

1. 

,


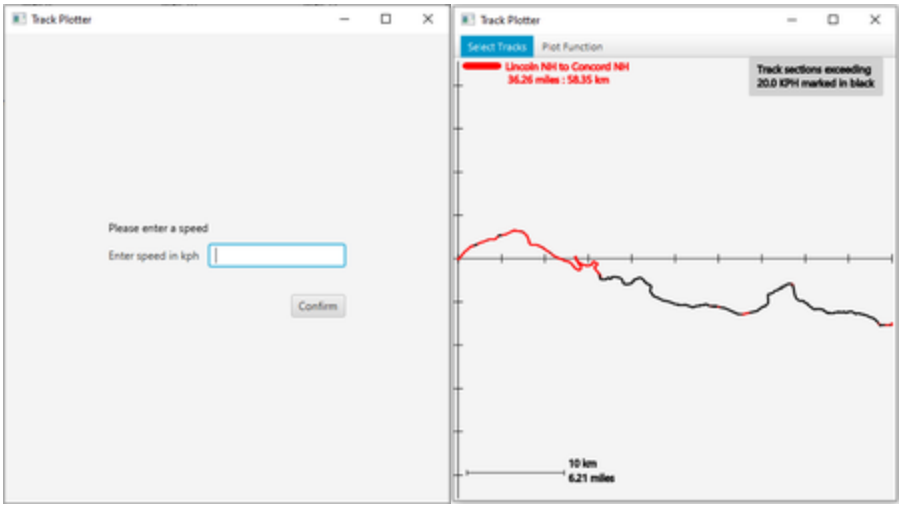
vii. **MHA5-16 - 7. Show calories expended graphically as a function of time**

DONE

1. 

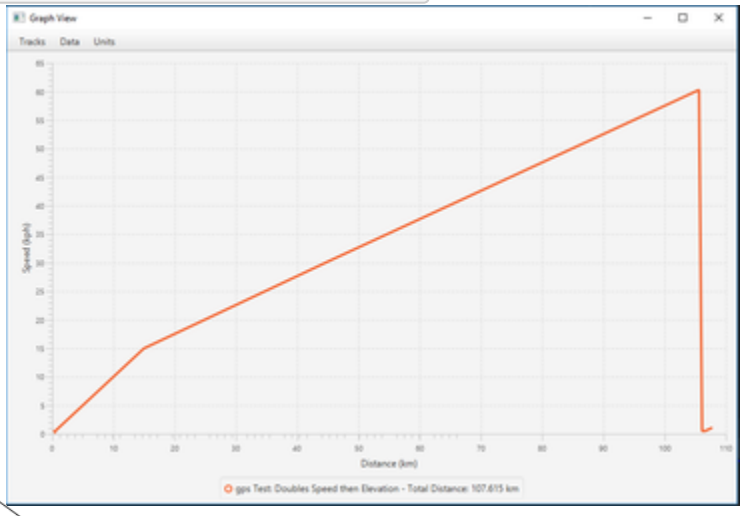
viii. **MHA5-11 - 18. Graphically View Tracks with Instantaneous Speed Exceeding a Value**

DONE

1. 

ix. **MHA5-17 - 12. Speed vs Distance plot**

DONE

1. 

-5 Actually: 10 and 13 were not done by the 5/19 due date - you fixed the remaining problems after the "official" sprint close. You can see this in the "Sprint Report" version of the burndown, where those PBIs were still not validated when I closed your sprint.

x. **If the PBI is a User Story (Feature), insert a screen shot of that PBI that illustrates the feature when run against the test file(s) you created for that PBI.**
You do not have to insert screen shots for non-feature PBIs.

b. Discuss which specific PBIs were **not** finished. Explain the reason(s) that these Issues were not completed.
   i. We finished all PBIs

c. List the errors or needed work in your application that will be characterized as *Defects* for a subsequent sprint.
   i. We did not find any errors. All features have been validated by the Product Owner.

Part 2: Sprint Retrospective

Think about how the past sprints have gone. Consider that you will likely be working in a team in the future - either in upcoming courses or in your post-academic career.

What improvements would you make in future sprints to make your process more efficient and your work product better? List three things you would change.

- Getting a better sense of how much work a task will take to complete is something that would make our process more efficient.
  - During sprint 2, we underestimated how much time our tasks would take, leaving us with PBIs unfinished. In sprint 3, we made the opposite problem based on our experience of sprint 2 and overestimated how much our tasks would take. We estimated 35 hours of work for this sprint but completed it in ~25 hours of work.
  - Knowing how much our team can handle would improve our work efficiency and made our product at the end of sprint better.

comes with experience

- More involvement in pull requests.
  - One of the tasks that Matt had to do was improvements on a task from a previous sprint. (Adding improvements to the plotter)
  - We did a better job of pointing out defects in the pull requests this sprint, but it could probably still be improved.
- Cleaner/more organized code to allow for easier integration of newer features
  - When the graph was initially created, all of it's functionality within the controller, and then when the next sprint involved adding a feature to the graph, all of the functionality got moved to a new class for more organization and easier readability.

- Initially creating this class would have saved time during the next sprint.

To me, this sounds like you're saying "more upfront design" work would help.