PROJECT LOG

Content

Day 1

Plans First thing we will be doing today is planning how we are going to approach this coursework, as there are many different aspects to consider (including this project log) that will require good time management. Today we will be reading over the skeleton code given out with the assignment to understand what parts of the code we need to add to as well as understanding how the initial program works (the output and the status classes' usability is not clear at a first glance for example) and fully grasping the functionality of the ControllerImp class. We will also read up on the logger java class to be able to output the desired results at each level. Hopefully we will have written a couple of methods for the ControllerImp class and created the necessary classes not involved in the skeleton code, using the sample solution given out earlier in the semester.

Achievements After spending approximately 5 hours on the project, we have managed to look through the code and understand most of it. Today we finally grasped how logging works and why it is useful, which will be very helpful in order to add logging to our code (and for usability of program). We also created an Enum class for the possible modes the User could implement, and after reading up on the subject in several websites, we decided to place it inside the ControllerImp class rather than outside as it will be more useful there. General variables that we will be using, such as curTour, have also been created.

Reflection Unfortunately our work pace has not been what was expected, and it has taken us a long time to understand the code provided to be able to get started. There are still a couple of doubts about some methods, like the getOutput() method in the Chunks class and also about how the

program should eventually work together. Hopefully we will resolve these doubts in the next session or at a demonstrator meeting. We are also confused still about how to use location, taking into account that there is no real user input and that there will be given coordinates instead – this seems inflexible and unrealistic, but we will learn to use it. We also need to find time for writing Tests, which will come after writing more of the code, to ensure that they test everything the coursework asks for, but this seems like a far-away activity for now.

Day 2

Plans We have now worked on the assignment for a couple of days distributed across a week, and we find the breaks in between slow us down when we first sit down to work again, as we find ourselves having to read parts again so as to understand them. We have made some progress regarding the creation of the necessary classes such as Location, Waypoint, Leg and Tour with some of their variables, as well as having added some methods to the Create Tour mode of the ControllerImp class. These are finalised, so we will try and do that today.

Achievements Today we worked on filling the classes we have for over five hours (some constructor methods and variables), as well some of the Create Tour methods. We did this by understanding the code and going through the Create Tour block of the ControllerImp class, as this was an easy way of knowing what needed to be where in terms of methods and variables. As a team we managed to understand logging properly and we are now used to adding log messages to the appropriate importance levels with each addition to the code. For example, information about the program's run and actions will be reproduced when the level is set to fine, while actions that would generate errors or warnings will be logged at warning or error.

Reflection Unfortunately, it seems to be taking us much more time than anticipated to complete some tasks, which is particularly worrying taking into account the number of days we have already spent on this assignment and the days we need and have. We have had to keep going back in our code to either add or modify some parts, so there is a big part still left to complete. We also had to 'clean up' our code after adding many comments to understand each part of our work. Overall, I would add that although our progress is slow we now have a very good understanding of the program, which will hopefully make it easier to keep adding to the code.

Another reflection is that working with a partner on this project is particularly helpful, as having the other teammate to discuss our doubts gave solutions faster, and together we managed to solve most of them, placing us on the right track for the rest of the coursework.

Day 3

Plans This might be the last log we complete as it does not look like we are going to be able to finish the project fully. We have worked on the assignment for a total of 8 days which is not ideal, but all we could manage taking into account other courses' work. By now we have nearly finished the Create Tour mode of the ControllerImp class and we will attempt to write Chunks, which will be hard as we have to make sure the final method getOutput() in each stage of the Create mode contains the correct chunks, like CreateHeader for the Create mode. Also, we have already decided to have a Location class as well as a Displacement class, which we were ambiguous about at first.

Achievements Today we finalised the Create Tour mode of the main class, by adding things such as making the current tour blank after it has been added to the list of tours or leg direction (a calculated bearing of type double) and the addLeg method. We also managed to solve some errors in our code such as initialising the waypoint ArrayList, at first we had not done that and so it wasn't working – we finally learnt that if you only declare it as you would an int, you are in fact creating a pointer to the memory location the ArrayList will be stored at, you are not creating an object. This is why when we tried to access it from the ControllerImp class (the addWaypoint method) the program gave errors.

Reflection A particularly problematic part of the code turned out to be the addLeg method, as it is called implicitly, meaning the controller does not choose to add a leg like he can do with a waypoint, and so knowing where to call the leg, at the beginning of a waypoint or at the end, was confusing. Another complication was the checking of a Waypoint being allowed, meaning it is not within another Waypoint's radius – we encountered a problem when considering what would happen in the case of a waypoint

being within a non-adjacent waypoint's radius, and how the situation should be handled. We created the functions isFurtherToThan and isAllowedIn to try and solve this problem.

In the case that this is our last log entry I would like to make the following remarks: I found this assignment interesting as it challenged my knowledge of java, but I found that it was too long for a 10-credit course assignment, and the fact that it coincided with many other deadlines did not help. We will most probably not be able to finish the code, which might mean a 0 in the correctness of code part of the marks as there will be problems with the j-unit tests. Unforeseen problems were a big part of the challenge as we kept having to go back and change the code so it would work properly or look things up so as to be able to understand them and use them correctly.