ASSIGNMENT 2 – Implementing a Gantt Chart generator on a command line with a text interface.

COMP10050

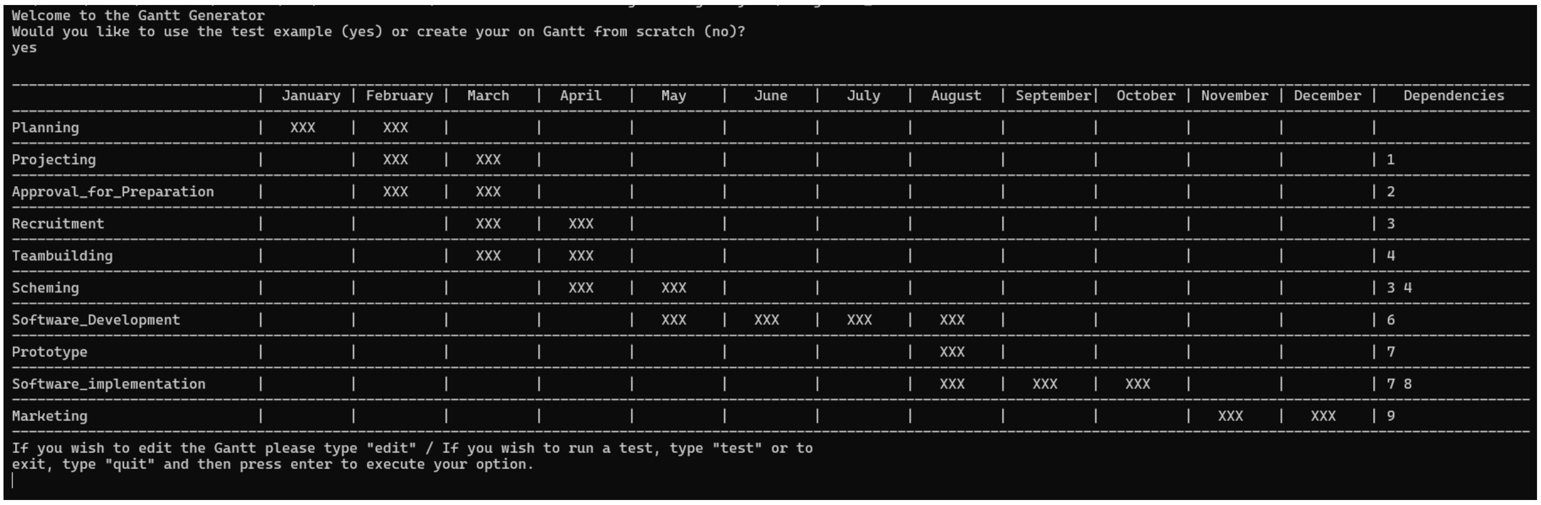
Spring 2023/2024

Group D (Simon Lasak, Darren Lyons)

https://csgitlab.ucd.ie/20515149/group-d-assignment-2.git

This document provides implementation details and contains information about our decisions in the Gantt Chart C source code.

Displaying the Gantt Chart



```
Welcome to the Gantt Generator
Would you like to use the test example (yes) or create your on Gantt from scratch (no)?
yes

------------------------------------------------------------------------------------------------------------------------------------------------------------------
                              | January  | February | March    | April    |   May    |   June   |   July   |  August  | September| October  | November | December |   Dependencies
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Planning                      |   XXX    |   XXX    |          |          |          |          |          |          |          |          |          |          |
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Projecting                    |          |   XXX    |   XXX    |          |          |          |          |          |          |          |          |          | 1
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Approval_for_Preparation      |          |   XXX    |   XXX    |          |          |          |          |          |          |          |          |          | 2
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Recruitment                   |          |          |   XXX    |   XXX    |          |          |          |          |          |          |          |          | 3
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Teambuilding                  |          |          |   XXX    |   XXX    |          |          |          |          |          |          |          |          | 4
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Scheming                      |          |          |          |   XXX    |   XXX    |          |          |          |          |          |          |          | 3 4
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Software_Development          |          |          |          |          |   XXX    |   XXX    |   XXX    |   XXX    |          |          |          |          | 6
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Prototype                     |          |          |          |          |          |          |          |   XXX    |          |          |          |          | 7
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Software_implementation       |          |          |          |          |          |          |          |   XXX    |   XXX    |   XXX    |          |          | 7 8
------------------------------------------------------------------------------------------------------------------------------------------------------------------
Marketing                     |          |          |          |          |          |          |          |          |          |          |   XXX    |   XXX    | 9
------------------------------------------------------------------------------------------------------------------------------------------------------------------
If you wish to edit the Gantt please type "edit" / If you wish to run a test, type "test" or to
exit, type "quit" and then press enter to execute your option.
|
```

We used for loops and width format specifiers to print out the rows of the Gantt Chart.

However, also a “variable” width specifier (*) that calculates how many whitespaces before the | separation line in the task name. Below you can see that printf has 3 arguments. The middle one is the value that the asterisk takes. It is 30 – the length of the name of the task and the vertical line. This way, we can format the output of the chart without needing to know exact lengths of the string beforehand.

```c
printf("\n%s%*c", (tasks[i].name), 30 - strlen(tasks[i].name), '|');
```

Then there was a loop from 0 to 11 that printed out 10 spaces and the | vertical line which were boxes for each month in the task row. One if/else statement was used to determine whether the input of start and end month from user was in corresponding to the number of the box/month. I the box index was in the range of the start-end month of the task „XXX“ was printed in the corresponding box.

Mapping the tasks to struct.

The struct for the task looks like this:

```
struct task{
    char name[30];
    int start_month;
    int end_month;
    int numOfDependencies;
    int dependencies[9];
};
```

Contains all the information needed for one task.

In the main function an array of 10 structs was initialised like this for simple handling and memory efficiency:

```
struct task allTasks[10];
```

Then simple scanf took care of writing the data from user input straight to a corresponding task number.

```
scanf("%d", &allTasks[4].start_month);
```

- Here the number of a month would be mapped straight to a start_month variable in the 5th task

The same was applied to all the variables in the struct.

Edit and change tasks

There are two functions that the program uses to change tasks. First, it must match the name of the task to a corresponding index of the array of tasks and then call the function to edit tasks with that index. The edit function was almost identical to the input. However, the only difference may be that instead of scanf to map the name of the task to the struct we used strcpy because the character array is immutable and must be dealt with differently.

Circular dependency

We have implemented function that recursively goes through the dependency path and prints whether there is a possibility of a circular dependency. It asks for the number of the task to be tested and prints out the path of its dependencies. It keeps track of the tasks already visited and if it lands on a task that has been already in the dependency path it warns the user that there is a possible circular dependency.

ASCII Art

The ASCII art included says GOODBYE to the user. Simple and nice.