

Task ID	Description	Complexity (S/M/L)	Justification
1	Code reading for comprehension	L	This is the task everyone spent the most time on. Since we wanted to get a detailed understanding of the system we felt this was the only way of achieving this.
2	Individual Class diagrams	L	Creation of two class diagrams according to "Individual Milestone 1" to better understand the system.
3	Debugging the system	M	We thought it would contribute a lot to our comprehension if we could run the program using a debugger. We also spent a lot of time trying to make this happen.
4	Final Class diagram	M	In the beginning, some of us created a class diagram as a part of the individual milestone, and we decided to create one common class diagram to cover the important functionality and combine all thoughts from the first diagrams. This required us to get on the same page comparing all the existing diagrams and agree on the scope of the new diagram. The new class diagram is a big help for the system comprehension.
5	Setup libSMCE and smce-gd on our machines	L	Each member of the group encountered multiple different errors and obstacles of varying complexity before managing to compile/launch the projects.
6	Class documentation	M	We have documented most of the classes of the libsmce library ourselves. We selected the most important ones based on our understanding of the flow of the system. Based on this information we also continued to further specify the classes using the common class diagram.
7	Brainstorming/ Discussions/ Sharing gained knowledge	M	Most of our understanding of the system came from multiple working sessions where we read and discussed code together.
8	Creating the presentation slides	S	Presentation slides for the first milestone to better present the used techniques and first requirements.
9	Testing out the program/trying to compile different	S	We thought it would be nice to see the system in action and that it might help our comprehension. Also making sure we can

	robots		compile robots confirmed that the program was actually working.
10	Doxygen documentation	S	In the beginning we generated the doxygen documentation while also using GraphViz. It helped with understanding the scale of the system, but further code inspection was needed.
11	Flow diagram	S	We created a flow diagram based on how libSMCE is used in the example program stduart.

Name	Task ID	Contribution in %
Max	1	20%
	3	100%
	5	20%
	6	20%
	7	20%
	9	60%
Tim	1	20%
	2	33%
	5	20%
	6	20%
	7	15%
	8	100%
	10	25%
Pontus	1	20%
	5	20%
	6	20%
	7	25%
	9	20%
	10	25%

Marek	1	20%
	2	33%
	4	100%
	5	20%
	6	20%
	7	20%
	10	25%
Lukas	1	20%
	2	33%
	5	20%
	6	20%
	7	20%
	9	20%
	10	25%
	11	100%