Interactive Dashboard Design

Robo^3 game sends a message to a remote server each time a user performs an action of the following: entering a level, exiting a level, completing a level and failing a level. Some additional information related to the number of instructions, cycles and functions used in each solution is also contained. This allow to store data about the user behavior and progress in the game. With this information, we can come up with several ways of showing the data in charts, such as: number of plays, total playtime, number of cycles and/or instructions per level, success probability of each level, among others.

However, in order to determine the best way to display and represent this data, it's necessary to understand the needs and motivations of each of the two different users that will have access to the visualization system. What specific user problems will the system solve? How will the user interact with the system in order to make a decision?

User 1: Student

- In general, the student would like to see his personal progress of the game in the dashboard related to the different levels and the total play time.
- Data visualization should be focused on a simple and straight forward way and should aim to engage and motivate the student.
- Some information like success probability of each level should be avoided since it could discourage the student in the levels in which the interaction becomes more like a trial and error.
- A point system can be implemented (aside from total progress) to motivate the user. This system

User 2: Professor/Tutor

- In the case of the Professor/Tutor, the data visualization should include more advanced functionalities in order to take better decisions to evaluate the students.
- First, a clear overview of the whole group progress should be displayed. Filter controls can be applied to allow the user to select smaller groups or individual students.
- In addition to the basic student dashboard charts, visualization should be able to display data in a wider range of representations. Success of probability can also be shown in relation to other variables like number of instructions, cycles and functions per each solution (scatter plot).

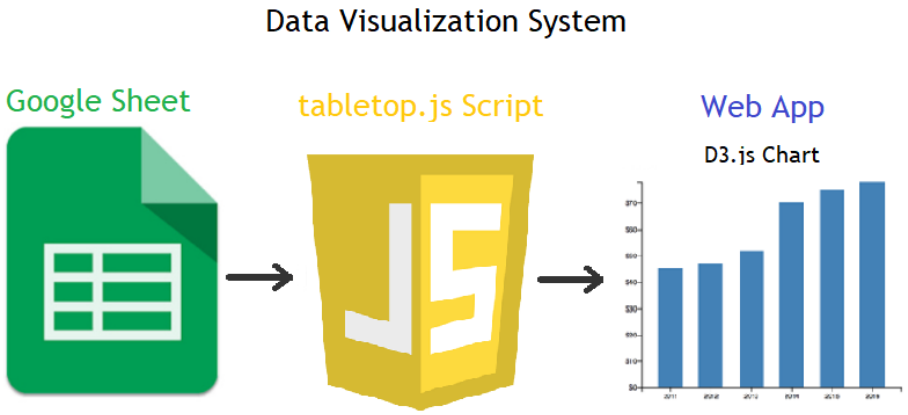
can assign points depending on the efficiency of the solution for each level. This means that the student can complete the game (100%) but still can replay each level in order to achieve the maximum amount of points by finding an optimal solution.

- Filters and zoom controls can be applied to the charts giving the user the possibility to see the data in detail if requested.

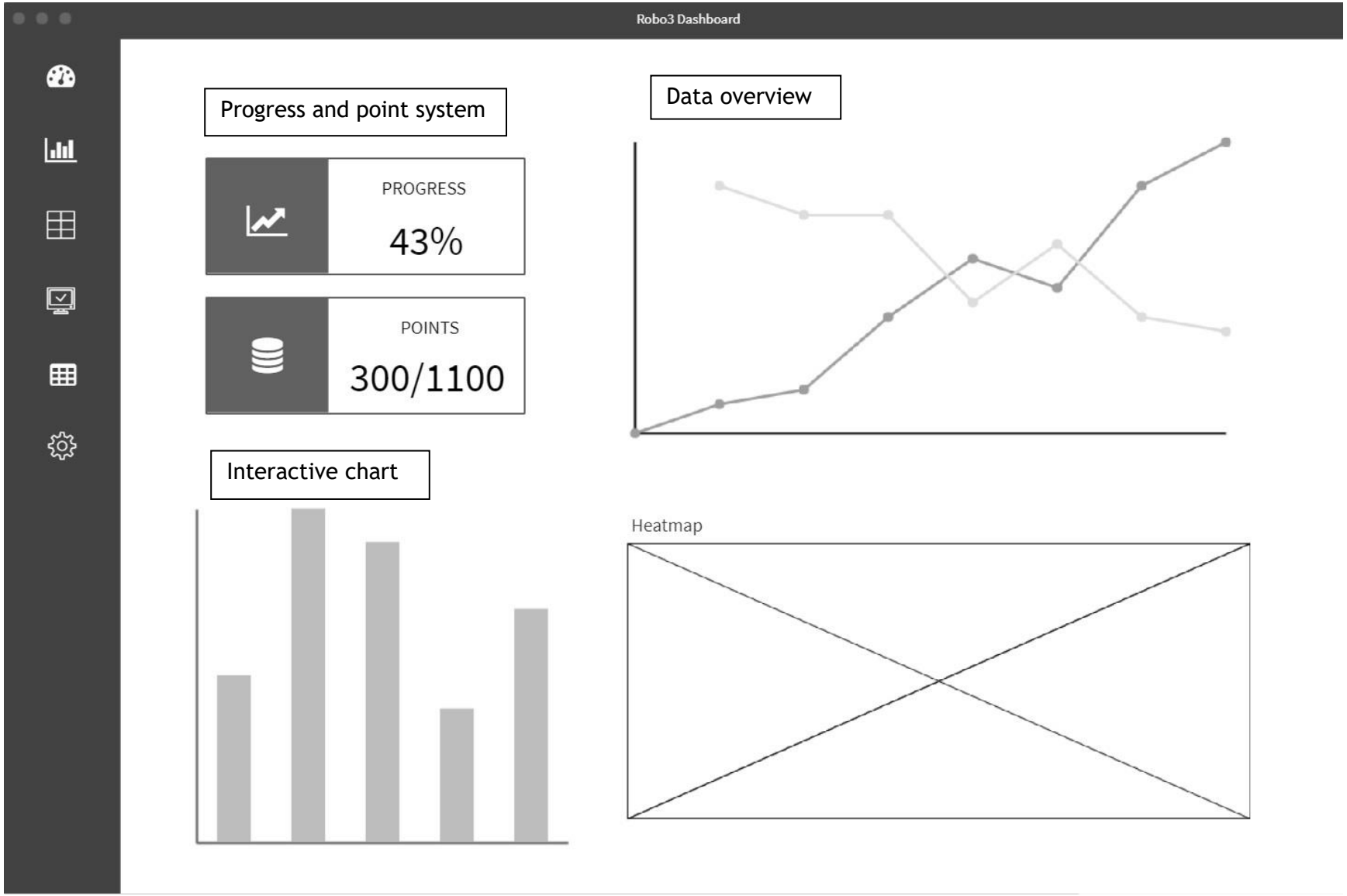
- A heat map for each student can be implemented to give more information about their frequency and constancy. This could give additional information to consider at the moment of evaluation.

How is this built?

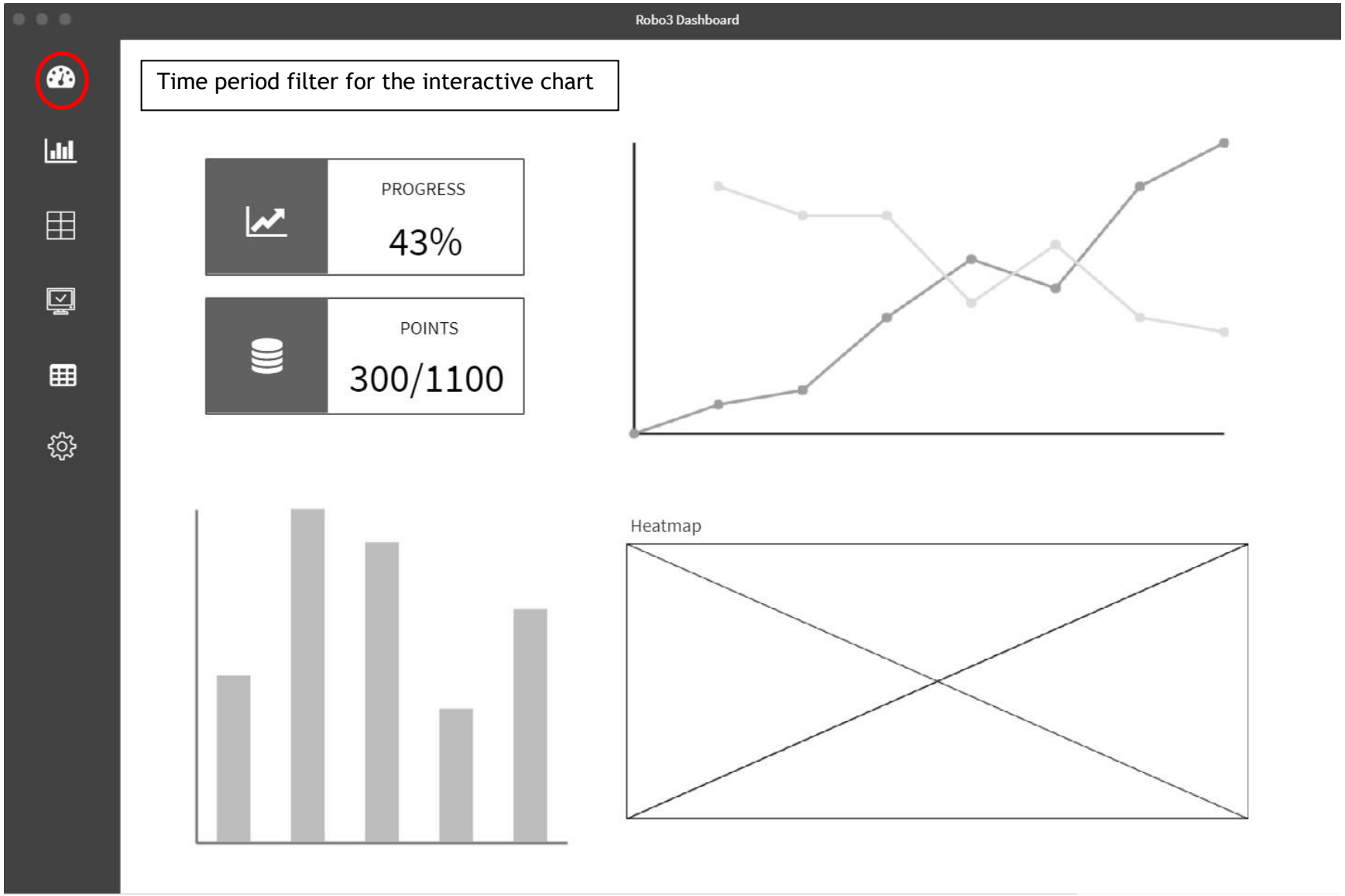
Basically, the data of the game is stored in a google sheet in a predefined format. Then, tabletop.js library is used to fetch the data and get JSON from the google sheet. Finally, the data is processed and represented in several charts using D3.js in a Web App. The Web App is built in such a way that allows scalability and uses a modular design.



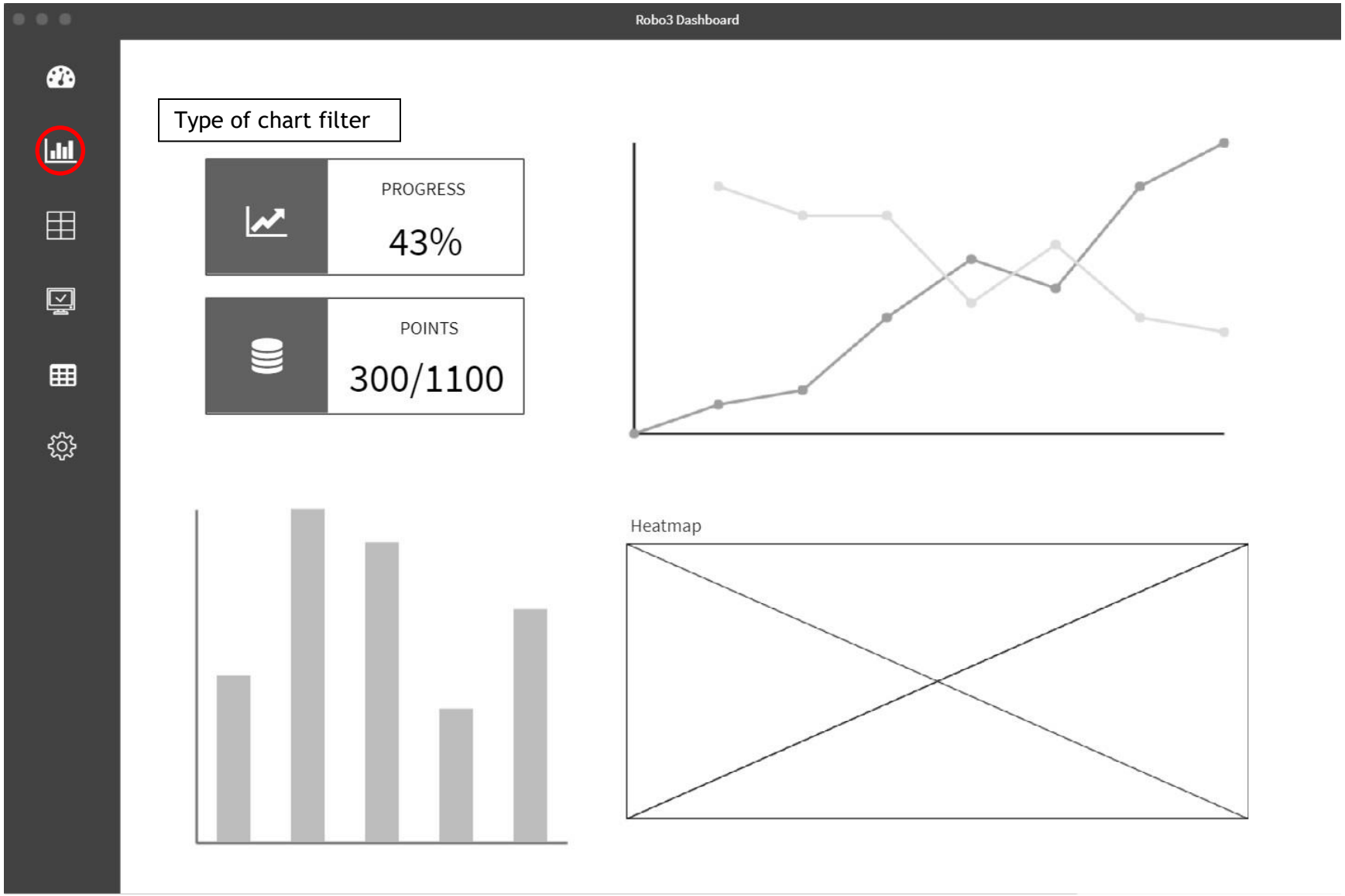
Web App Wireframe



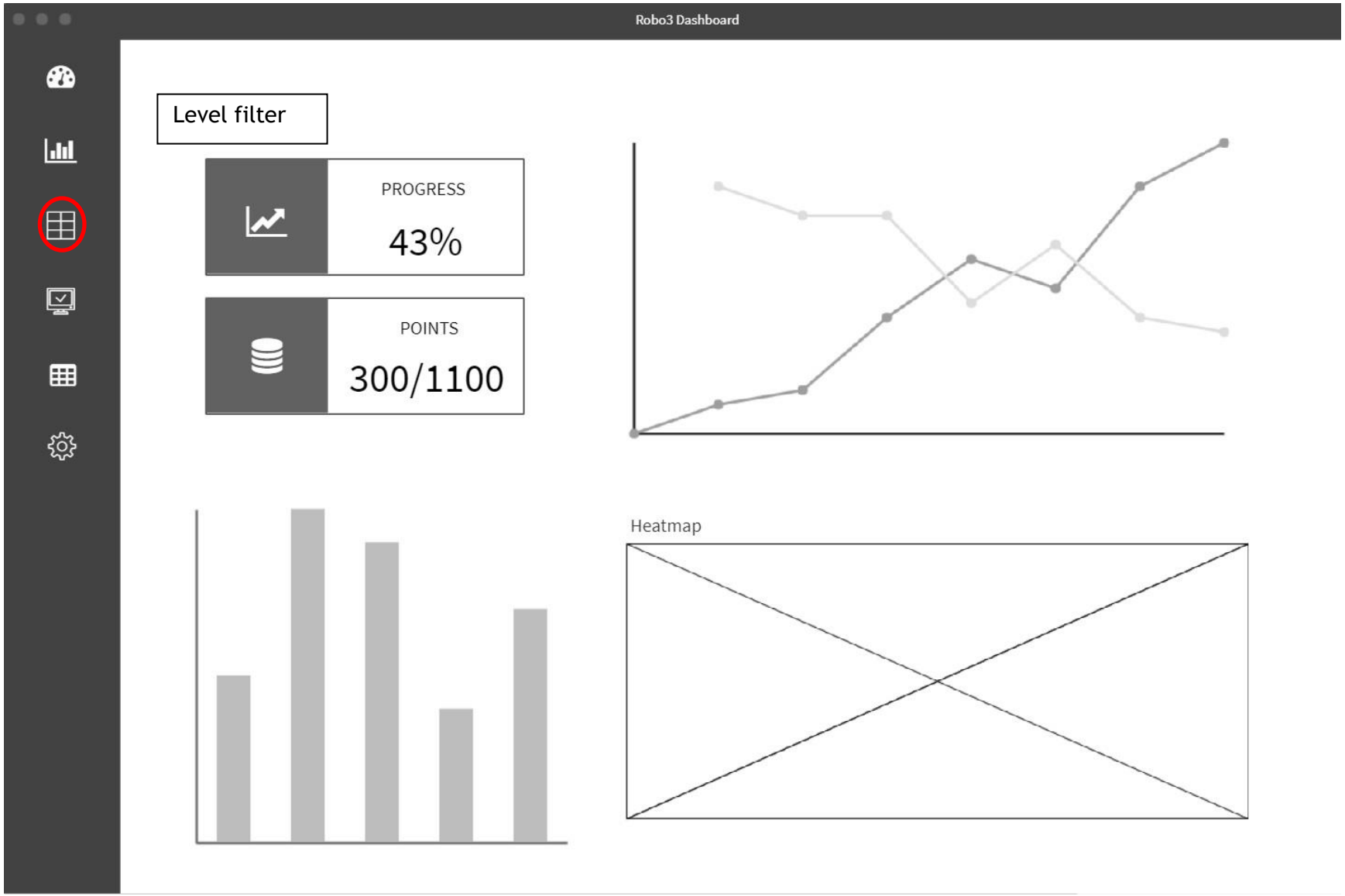
Web App Wireframe



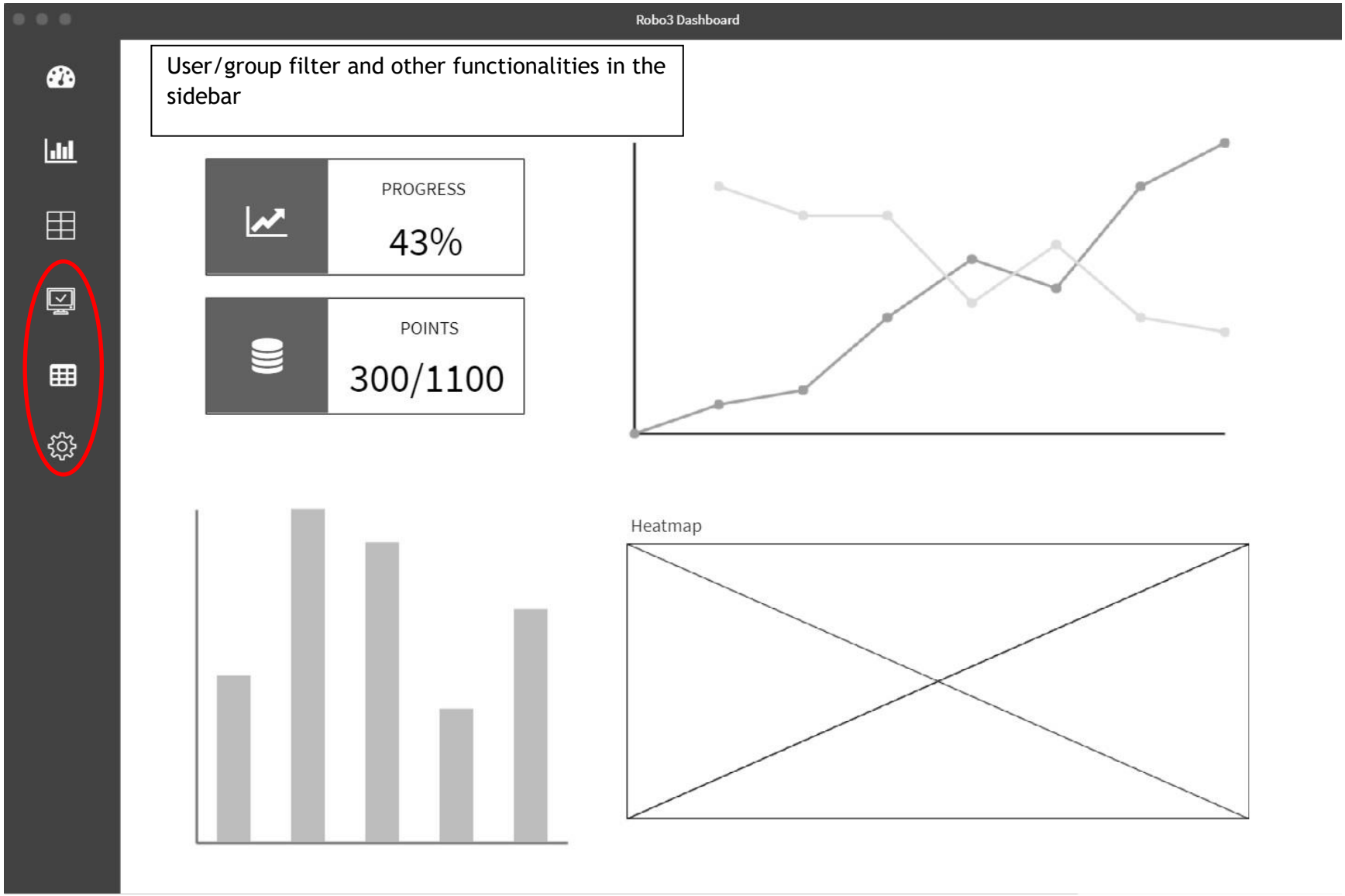
Web App Wireframe



Web App Wireframe

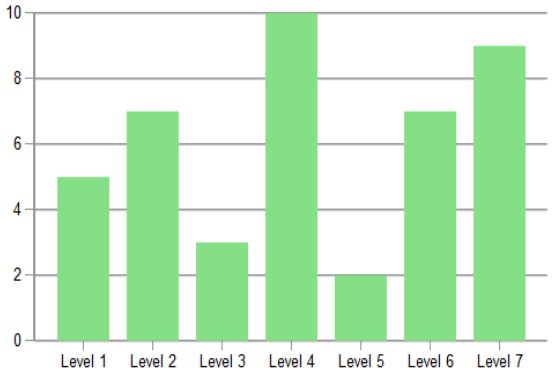


Web App Wireframe

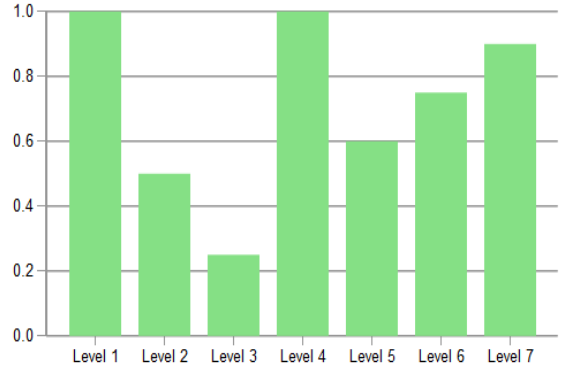


Types of chart

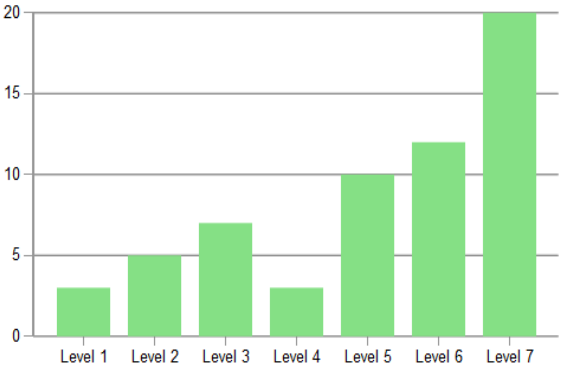
Number of plays



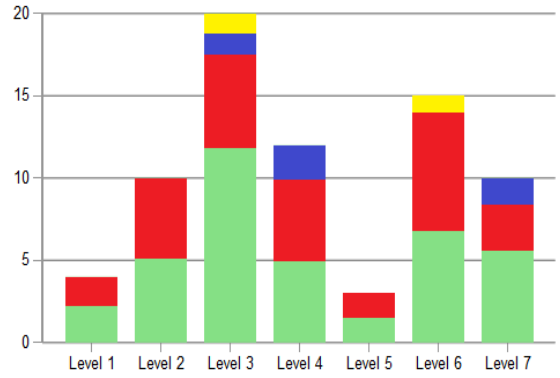
Success probability



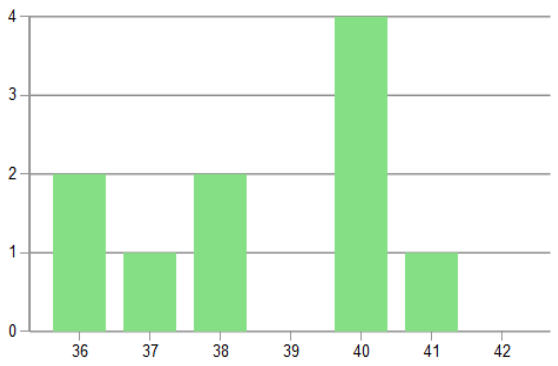
Total playtime (minutes)



Instructions count



Cycles (Reverse)



Success Probability/Instructions count