

## Project follow-up and conception choices.

For this second iteration we had three main tasks. Create an editor view where we could create shapes, create the conversion logic between the visual representation and the TikZ code and finish implementing the import/export functionality. Those tasks represented the amount of work-time that we had originally anticipated coming from all members of the group.

### Shape placement: from canvas to TikZ code

We divided this story in 2 part, in this iteration we needed to allow creation of shape on the canvas and translate them into tikz code,

We had no problem with this story and it should be easy to apply the same logic from TikZ to diagram.

### Conception

We created a java representation of TikZ code. Once a javaFX Shape is added on the canvas, we create the corresponding java object, with the right parameters, and store them into a “canvas” object, which holds all the TikZ java objects.

From there, we only need to print those objects to have the corresponding TikZ code to display.

### The import/export user story

We finished this story and made it work, but, we still feel like it could benefit from a refactor, as some of the MCV concepts are not yet applied.

### Conclusion

In conclusion, we consider that we have reached our goals. However, the refactoring (which should happen in the third iteration, if we follow our schedule) is still required to improve the quality of the code and will make it easier for us in the future. Furthermore, the EditorController is starting to look bloated and would definitely benefit from refactoring into two classes.