

Sprint 2 retrospective

User Story	Task	Task Assigned To	Estimated Effort per Task (in hours)	Actual Effort per Task (in hours)	Done (yes/no)	Notes
#1	Website features	Malina, Paula	16	16	yes	
#2	Identify individual tiles on the map and show coord.	Everyone	8	0	no	There was no tile left for this and it was a low priority
#3	Improve tile retrieval	Andrei	3	1	no	A basic algorithm should be simple to implement, but an efficient one is required (time-wise)
#4	Testing, cleanup, refactoring	Andrei, Malina, Paula	10	10	yes	
#5	Deploy the server and database	Everyone	2	5	no	For now, the application is hosted on heroku and the database is on the EWI project server
#6	CAPTCHA	Georgi	10	15	yes	
#7	Create train set for machine learning	Andrei, Malina, Paula	48	32	yes	Took a long time since lots of tiles had to be checked
#8	Store the tiles in a database	Andrei, Malina, Paula	8	6	yes	

#9	Configure TensorFlow	Boris	10	22	yes	Spent way too much time on figuring out how to make it work
#10	Add object layer for a map	Boris	4	3	no	Still needs to be done

User Stories

1. As a user, I want to be able to check information regarding the classification of the tiles, see the legend of the maps and have an accessible user interface.
2. As a user, I want to be able to see information about a tile when I am zoomed in and click on it.
3. As a developer, I want to be able to retrieve tiles efficiently, even for maps that have over a million.
4. As a developer, I want to be able to have clean code and have it tested thoroughly
5. As a developer, I want to be able to have the database and server hosted by the Faculty of Architecture.
6. As a developer, I want to be able to store in the database the coordinates of the tiles and their corresponding label given by the user at CAPTCHA.
7. As a developer, I want to be able to train the machine learning algorithm and as such, a training set is needed
8. As a developer, I want to be able to retrieve the tiles from the database based on their spatial coordinates.
9. As a developer, I want to be able to have an algorithm with high accuracy which classifies the tiles based on their features.

Main Problems Encountered

Problem 1: Training set

Description: Downloading the tiles took a long time on its own, which gives the reason for an improved tile retrieval algorithm for the future. Furthermore, classifying them based on colour alone couldn't have given perfectly accurate results, so the tiles needed to be manually checked. (this won't be an issue in the future, since the machine learning algorithm will use CAPTCHA submissions)

Reaction: We decided to classify the tiles by colours and then verify them manually just this once, in order to have a dataset for the machine learning algorithm and see how accurate it is. By doing so, we can assume that future datasets generated by people filling in CAPTCHAs will have similar results.

Problem 2: Lack of testing

Description: Code that was written during the first sprint and the first half of this sprint was put to use quite quickly and as such, little to no testing was done. (This was part of the adjustment from the first sprint, but it was pointed out during a weekly meeting with our TA, which is why it is included here)

Reaction: As this is obviously a necessity for the code to be both maintainable and sustainable, we shifted our attention from adding new functionality to making sure that already existing functionality is working as intended.

Problem 3: Code structure

Description: Due to the way in which Django interprets files and packages, we came across an issue known as [the double import trap](#) while writing one of our tests.

Reaction: Since it took a while to figure out that this is indeed the problem we were dealing with and since there is not a straight-forward approach that works for any project structure and in any situation, quite a bit of time and headaches were necessary to solve this.

Problem 4: Lack of documentation

Description: During the midterm presentation, it became clear that we had documented the project too little.

Reaction: We decided to make writing a proper documentation an issue (on GitLab) for the next sprint and to make sure that everyone contributes to it by adding proper descriptions of what their code is doing both within the code and on the project wiki.

Problem 5: Configuring tensorflow

Took way too much time to configure tensorflow and find a proper object detection solution in general. We've tried some simple machine learning solutions but they didn't work. Then we picked up tensorflow. Even though object detection script was working in the end, 2 trained models behaved poorly, that's why we had to find a different solution. [Detecto](#) library was used - it's a wrapper around PyTorch to make object detection up and running easier for the developers. In the end, we trained a church detection model which works pretty well.

Adjustments for the next Sprint Plan

- Continue testing as much as possible
- Write documentation
- Add the remaining labels for the machine learning algorithm

Adjustments from the previous Sprint Plan

- One of the adjustments from the first sprint was to start writing more tests and we made sure to do so
- The other adjustment was to find efficient ways for working with tiles; while we have not yet found an efficient solution for retrieving tiles that are usable (not black/transparent), storing tiles in the database was easy to implement and rather quick too (about 3500 tiles per minute)