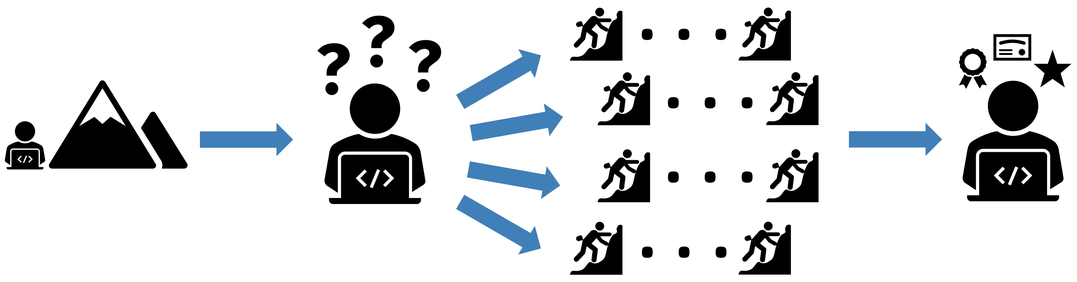
Gamifying JupyterLab to Encourage Continuous Interaction in Programming Education

Annabell Brocker, Sven Judel, Rene Roepke, Nikol Mihailovska, Ulrik Schroeder

Motivation

„It is better to make many small steps in the right direction than to make a great leap forward only to stumble backward.“ [10]



When using virtual learning environments, e.g. for programming education, **Learning Analytics (LA)** can be used to **analyze learners' activities** and gain **insights on how learners interact** with educational materials. LA can also be used to **implement gamification** in order to foster motivation and engagement, e.g. in introductory programming courses where novices perceive learning how to code as hard and high dropout rates can be reported [1,4].

JupyterLab presents an **interactive programming environment** in which learners can work actively with program code when solving assignments, thus, **offering valuable data for LA and gamification**. This way, learners can already be supported during code development, however, **JupyterLab does not support gamification** to engage learners in continuously using the programming environment for practice.

Research Questions:

- ***How can Learning Analytics be applied on game elements in Jupyter?***
- ***What data is necessary to allow the support of continuous interactivity?***

Gamification in JupyterLab

Metrices and Game Elements

In order to **measure the learners' engagement** with learning materials or their activity within the environment overall, different metrics can be used, e.g., overall time spent in the course [8], activity type, start and end time of learning sessions [7] and the frequency and duration of interactions [9].

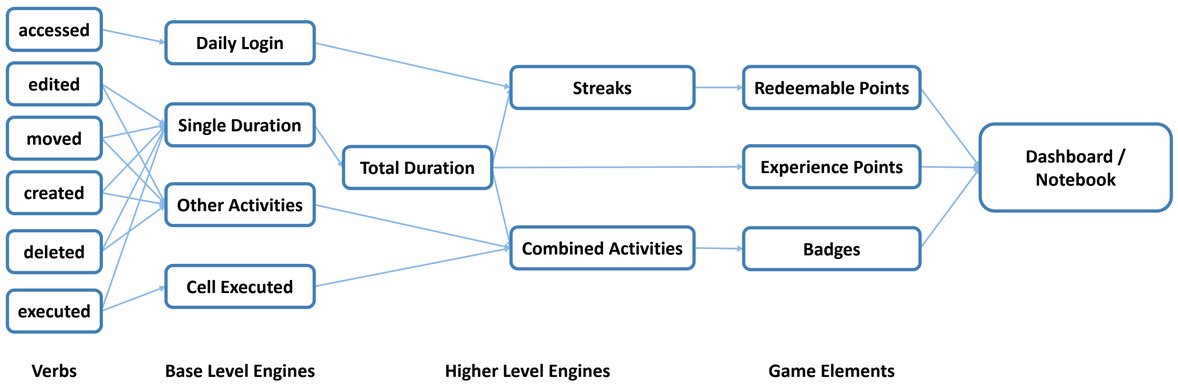
Metric	Game Element	Explanation
Frequency	Redeemable Points	Obtained by streak of consecutive logins. Online currency that can be used to unlock additional material or badges.
Duration	Experience Points	Obtained by each interaction within the JupyterLab environment. Denote the level of progress.
Activities	Badges	Obtained by reaching predefined activity requirements. Visualize the successful completion of tasks or accomplishment of goals.

The analysis for integrating redeemable points, experience points and badges is currently based on 7 different engines which again are based on multiple verbs that corresponds to the users' actions.

For example, when a user with the pseudonym „pseudonymous-abc123“ executes a code cell within a notebook, a human readable xAPI statement is logged in an *actor-verb-object* structure:

<pseudonymous-abc123> <executed> <Code Cell>

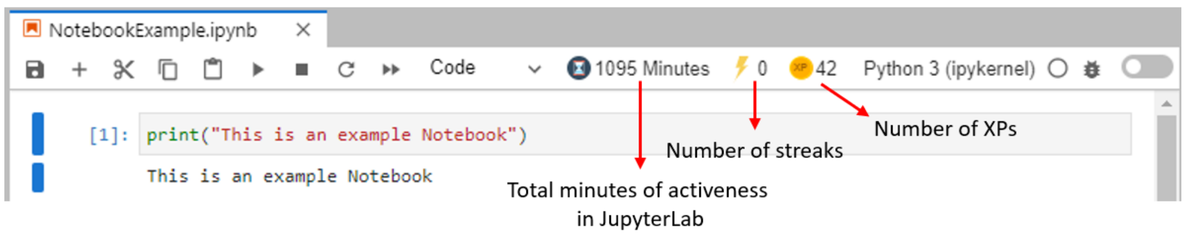
Based on the verbs, the interactions are analyzed by the engines and translated into the game elements:



Visualization in a Jupyter Notebook

Individually selected game elements or analysis results can be made available to participants in each notebook in order to increase direct feedback. Users can see the following:

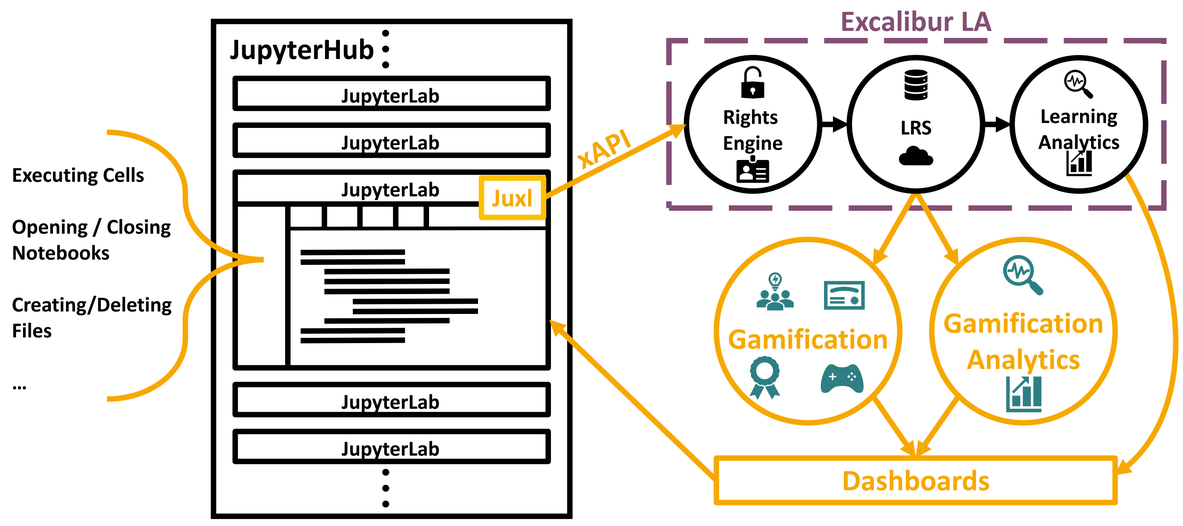
- How long they have been active in JupyterLab?
- How often they have logged consecutively (streaks)?
- How many experience points they have earned so far?



Architecture and Technical Details

The integration of data-driven game elements is based on various events within JupyterLab.

1. Capturing the events for the corresponding game elements
2. Converting events into xAPI statement
3. Storing xAPI statement within an LRS using Juxl [2]
4. Design and implement corresponding analytics/gamification engines within Excalibur LA [6]
5. Return the results via a dashboard or directly in a notebook



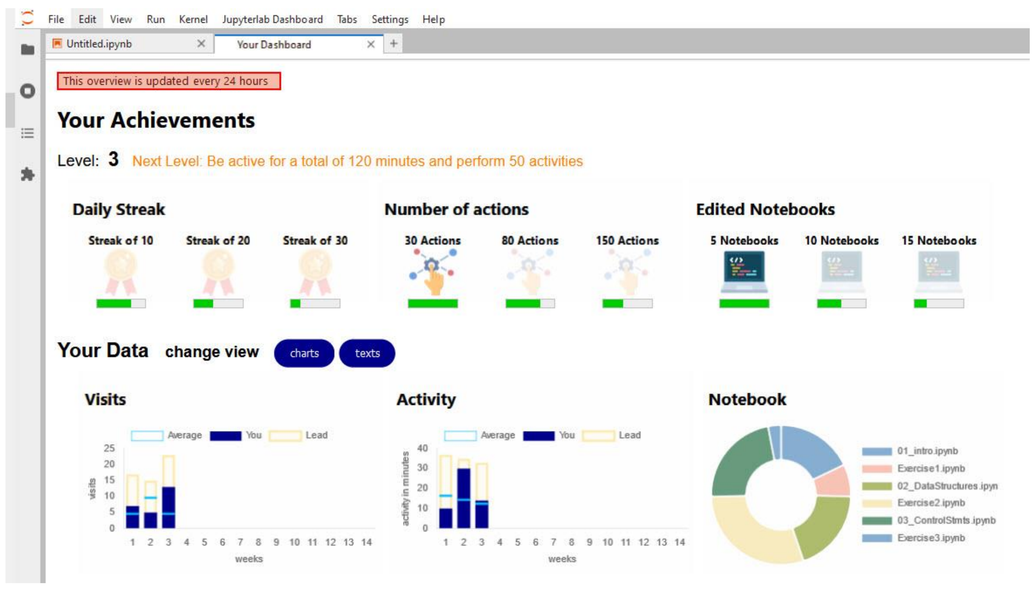
Future Work

A dashboard for JupyterLab:

- Serves to permanently display the learner's current progress
- Provides more detailed information
- Combines the data-driven game elements with further LA visualizations to constantly keep learners' interest up [3, 5]
- Should be individually adjustable

Direct analyses and immediate feedback:

- Inform learners about their progress in a timely manner for selected research and game elements
- E.g. for experience points and badges



References

1. Bosse, Y., Gerosa, M.A.: Why is programming so difficult to learn?: Patterns of difficulties related to programming learning mid-stage. ACM SIGSOFT Software Engineering Notes 41, 1–6 (2017)
2. Brocker, A., Judel, S., Schroeder, U.: Juxl: JupyterLab xAPI Logging Interface. In: Int. Conf. on Advanced Learning Technologies. pp. 158–160 (2022)
3. Cassano, F. et al.: Gamification and Learning Analytics to Improve Engagement in University Courses. In: Methodologies and Intelligent Systems for Technology Enhanced Learning. p. 156–163 (2018)
4. Derus, S.R.M., Ali, A.Z.M.: Difficulties In Learning Programming: Views Of Students. In: Int. Conf. on Current Issues in Education. pp. 74–79 (2012)
5. de Freitas, S. et al.: How to Use Gamified Dashboards and Learning Analytics for Providing Immediate Student Feedback and Performance Tracking in Higher Education. In: Int. Conf. on World Wide Web Companion. pp. 429–434 (2017)
6. Judel, S., Schroeder, U.: EXCALIBUR LA - An Extendable and Scalable Infrastructure Build for Learning Analytics. In: Int. Conf. on Advanced Learning Technologies. pp. 155–157 (2022)
7. Moubayed, A. et al.: Relationship Between Student Engagement and Performance in E-Learning Environment Using Association Rules. World Engineering Education Conference pp. 1–6 (2018)
8. Rodgers, T.: Student engagement in the e-learning process and the impact on their grades. Cyber Society and Education 1(2), 143 – 156 (2008)
9. Growth Engineering: Measuring user engagement on your learning management system (2016), https://www.growthengineering.co.uk/measuring-user-engagement-on-lms/, last accessed 08.11.2022
10. Sachar, L., https://www.goodreads.com/quotes/40904-it-is-better-to-take-many-small-steps-in-the, last accessed 27.10.2022

Annabell Brocker, M.Sc. RWTH
a.brocker@cs.rwth-aachen.de
+49 241 80 21955

Learning Technologies Research Group
Ahornstraße 55, 52074 Aachen
learntech.rwth-aachen.de