eleed
e-learning
& education

Issue 14
2022

Integration of Gamification and Learning Analytics in Jupyter

Annabell Brocker
a.brocker@informatik.rwth-aachen.de

Sven Judel
judel@informatik.rwth-aachen.de

Ulrik Schroeder
schroeder@informatik.rwth-aachen.de

Learning Technologies Research Group
RWTH Aachen University
52074 Aachen

urn:nbn:de:0009-5-55339

Abstract

Learning programming languages and programming concepts is associated with many problems for novices, which leads to high dropout and failure rates. Reasons often given are a lack of motivation or too little interactive materials. While gamification can be applied to increase motivation, the project Jupyter can be used as an interactive programming environment to enable more interactions with sample code. However, an approach of integration gamification in Jupyter is still missing. This report presents a concept of integrating arbitrary game elements and mechanics into Jupyter. Learning Analytics and Gamification Analytics will be applied afterwards to monitor and evaluate the measures and their impact.

Keywords: e-learning; Gamification; Programming Novices; Jupyter; Gamification Analytics; Learning Analytics

1 Motivation

Learning programming concepts is associated with different challenges for many novices, as for example (Derus & Ali, 2012; Bosse & Gerosa, 2017) :

- Using a computer
- Using programming environments
- Learning the syntax of the programming language
- Understanding basic programming concepts
- Designing a program to solve specific tasks
- Finding errors in own program

As a result of these problems, many novices drop out of programming courses or fail at the end of the course (Bosse & Gerosa, 2017). In addition to the above-mentioned challenges, some of which have not been overcome, reasons to drop out often include missing (interactive) sample programming codes, missing live coding, installation difficulties of a programming environment and a lack of motivation of learners (Bosse & Gerosa, 2017).

Using interactive programming environments, such as provided by the project Jupyter (Project Jupyter, 2021c), enables the use of interactive learning materials. As reported by Baumgartner and Herber (Baumgartner & Herber, 2013), the use of interactive media and materials can offer an added didactical value, on condition that they have been well integrated into the learning concept. In addition, Jupyter offers the possibility of a centrally hosted service, named JupyterHub, through which participants can access and work with the programming environment JupyterLab via a website. Increasing the motivation in non-game-based contexts, such as an university course, can be achieved by integrating game elements (gamification) (Deterding, Dixon, Khaled, & Nacke, 2011).

In the following, gamification and Jupyter are introduced and their benefits for programming novices are highlighted. However, as an integration of gamification in Jupyter is still missing, a concept to fill this gap, together with a road map for the implementation, is presented.

2 Gamification and Jupyter in Programming Courses

Teaching programming within a course setting is a common practice. E. g. on Coursera a high three digit number of Python programming courses can be found (Coursera, 2021). In such courses, many studies report lower failure and dropout rates as a result of introducing gamification (Marín, Frez, Cruz-Lemus, & Genero, 2019).

To prevent the need of installing a programming environment on their own and to provide interactive learning materials, programming novices can be grant access to the interactive programming environment Jupyter via a website.

2.1 Gamification

Gamification is known as “the use of game design elements in non-game contexts.” (Deterding et al., 2011, p. 10). With the help of this principle, the user’s motivation (extrinsic or intrinsic) should be increased and thus lead to a higher learning success (Marín, Frez, Cruz-Lemus, & Genero 2019).

Gamification is already used in many different areas, especially in the field of education and learning (Koivisto & Hamari, 2019). Furthermore, there are already numerous different approaches within programming courses. In general, gamification can have a positive impact on the participants’ engagement in the programming course, the participants’ level of knowledge about programming, as well as on their motivation (Venter, 2020). Nevertheless, in order to achieve these effects, a course with gamification must be designed and realised with a high degree of attention on the composition of the game elements, otherwise the objectives will not be achieved and, conversely, it could have a negative effect (Marín, Frez, Cruz-Lemus, & Genero 2019; Rojas, Chiluiza, & Valcke, 2017).

The game elements in programming courses, which have been used a lot in the past, are summarised in a literature review from 2020 (Venter, 2020):

- Leaderboards
- Badges
- Points
- Levels

This application of game elements is also known as reward-based gamification (Nicholson, 2015). These elements serve primarily to address the extrinsic motivation of the participants for a short time (Nicholson, 2015; Agapito & Rodrigo, 2018). In order to stabilise the intrinsic motivation in the long term, the use of other game elements is also suitable (Nicholson, 2015; Agapito & Rodrigo, 2018). The use of gamification in programming courses is particularly noticeable in digital courses (Venter, 2020). Digital courses offer the advantage of logging data automatically and evaluating it according to the relevant regulations. However, the use of digital courses at the same time means that the integration of gamification depends on several factors. One of the main issues is:

What data can be logged from the programming environment?

An answer on this question depends on the choice of the programming environment. One possible programming environment is offered by Jupyter.

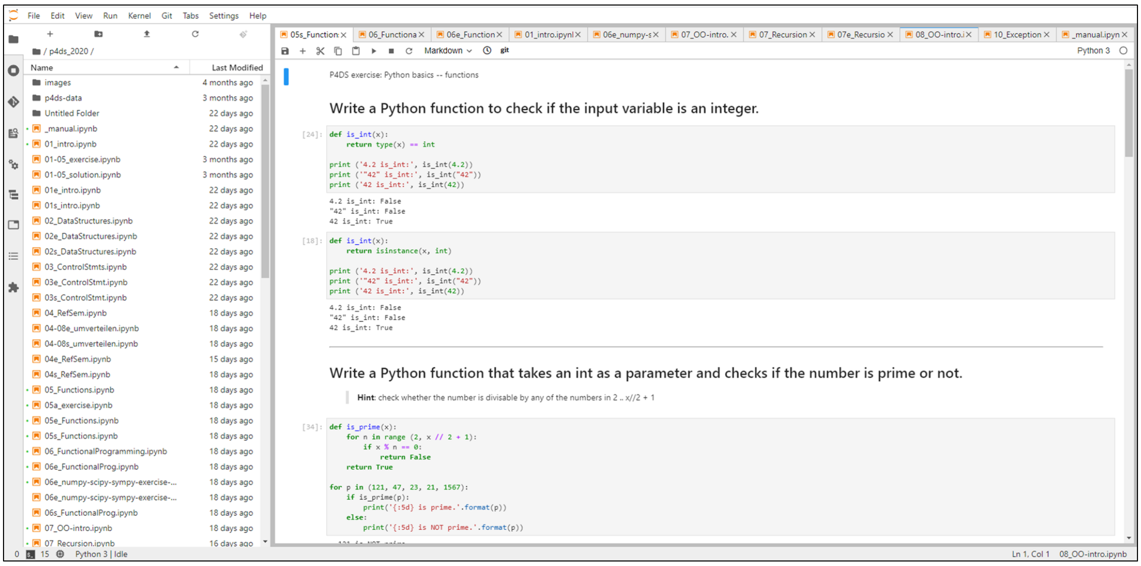
2.2 Jupyter

Jupyter (Project Jupyter, 2021c) offers open-source software for interactive programming environments. More than 40 different programming languages, as for example Python or R, are supported. The software package includes Jupyter Notebooks, JupyterLab as well as JupyterHub.

Jupyter Notebooks (Jupyter Team, 2021) are interactive documents consisting of multiple cells. Those cells can be code cells for direct and interactive coding as well as markdown cells to include text, visualisations or links in between the code. This can help when getting started with programming, as small code fragments can be executed directly and explained in advance or immediate afterwards. These Jupyter Notebooks can be read, edited and executed through the web-based interactive programming environment JupyterLab (Project Jupyter, 2021d). Within this environment, there is also a file browser, a window manager, a terminal emulator as well as an extension manager. An example screenshot is given in Figure 1. On the left, the file browser lists all files and notebooks, that can be opened by double-clicking. On the right, a notebook is opened including executed code cells as well as markdown cells.

Lastly, Jupyter provides JupyterHub (Project Jupyter team, 2016), a central service that enables institutions to make Jupyter Notebooks available to many users without them having to install a programming environment. Users receive their own account so that they can work with the Notebooks in their own JupyterLab instance.

Jupyter is therefore a suitable approach to prevent many of the problems regarding the technical infrastructure, described in section 1.



Name	Last Modified
images	4 months ago
p4ds-data	3 months ago
Untitled Folder	22 days ago
_manual.ipynb	22 days ago
01_intro.ipynb	22 days ago
01-05_exercise.ipynb	3 months ago
01-05_solution.ipynb	3 months ago
01e_intro.ipynb	22 days ago
01s_intro.ipynb	22 days ago
02_DataStructures.ipynb	22 days ago
02e_DataStructures.ipynb	22 days ago
02s_DataStructures.ipynb	22 days ago
03_ControlStmts.ipynb	22 days ago
03e_ControlStmt.ipynb	22 days ago
03s_ControlStmt.ipynb	22 days ago
04_RefSem.ipynb	18 days ago
04-08e_umverteilen.ipynb	18 days ago
04-08s_umverteilen.ipynb	18 days ago
04e_RefSem.ipynb	15 days ago
04s_RefSem.ipynb	18 days ago
05_Functions.ipynb	18 days ago
05a_exercise.ipynb	18 days ago
05e_Functions.ipynb	18 days ago
05s_Functions.ipynb	18 days ago
06_FunctionalProgramming.ipynb	18 days ago
06e_FunctionalProg.ipynb	18 days ago
06e_numpy-scipy-sympy-exercise-...	18 days ago
06e_numpy-scipy-sympy-exercise-...	18 days ago
06s_FunctionalProg.ipynb	18 days ago
07_OO-intro.ipynb	18 days ago
07_Recursion.ipynb	16 days ago

P4DS exercise: Python basics -- functions

Write a Python function to check if the input variable is an integer.

```
[24]: def is_int(x):
          return type(x) == int

      print ('4.2 is_int:', is_int(4.2))
      print ('"42" is_int:', is_int("42"))
      print ('42 is_int:', is_int(42))
```

```
4.2 is_int: False
"42" is_int: False
42 is_int: True
```

```
[18]: def is_int(x):
          return isinstance(x, int)

      print ('4.2 is_int:', is_int(4.2))
      print ('"42" is_int:', is_int("42"))
      print ('42 is_int:', is_int(42))
```

```
4.2 is_int: False
"42" is_int: False
42 is_int: True
```

Write a Python function that takes an int as a parameter and checks if the number is prime or not.

> **Hint**: check whether the number is divisable by any of the numbers in 2 .. x//2 + 1

```
[34]: def is_prime(x):
          for n in range (2, x // 2 + 1):
              if x % n == 0:
                  return False
          return True

      for p in (121, 47, 23, 21, 1567):
          if is_prime(p):
              print('{:5d} is prime.'.format(p))
          else:
              print('{:5d} is NOT prime.'.format(p))
```

Figure 1: Screenshot of JupyterLab.

Gamification has the potential to increase the learners intrinsic and extrinsic motivation within a learning scenario and Jupyter eases the first steps of getting into programming. However, an integration of gamification into Jupyter, combining the benefits of both, is still missing.

3 Project Proposal

This research project aims at simplifying the introduction into programming by using Jupyter as a web-based programming environment and gamification to increase the motivation of novices. To implement gamification, the interactions and learning activities within JupyterLab and Jupyter Notebooks need to be logged and analyzed. The results serve as the data foundation for integrated game elements and mechanics. Further, these logs should be used for Learning Analytics (LA) and Gamification Analytics (GA) to investigate the effectiveness of the applied gamification and its impact on the overall learning (Heilbrunn, 2019; Conole, Gasevic, Long, & Siemens, 2011).

At RWTH Aachen University, a central and modular LA infrastructure, Excalibur LA, is used to offer an LA service for various platforms of the university (Judel & Schroeder, 2022). It offers a data warehouse, data privacy and security management, a routine to execute arbitrary analyses and a result store to provide the analysis results instead of (re-)calculating them every time they are requested. Therefore, this infrastructure should also be used for this project to reduce the overhead of integrating basic functionalities for gamification, LA and GA in JupyterLab.

The data warehouse, a Learning Record Store (LRS), only accepts Experience API (xAPI) statements as input logs. Therefore, an extension for JupyterLab to log interactions and send them to the infrastructure as xAPI statements is required. Further, multiple modules to perform analyses on the logged data have to be developed and integrated in the infrastructure. Finally, a data receiver has to be integrated into JupyterLab that retrieves the analysis results and displays them as part of the gamification or within a dashboard as LA and GA indicators as shown within Figure 2.

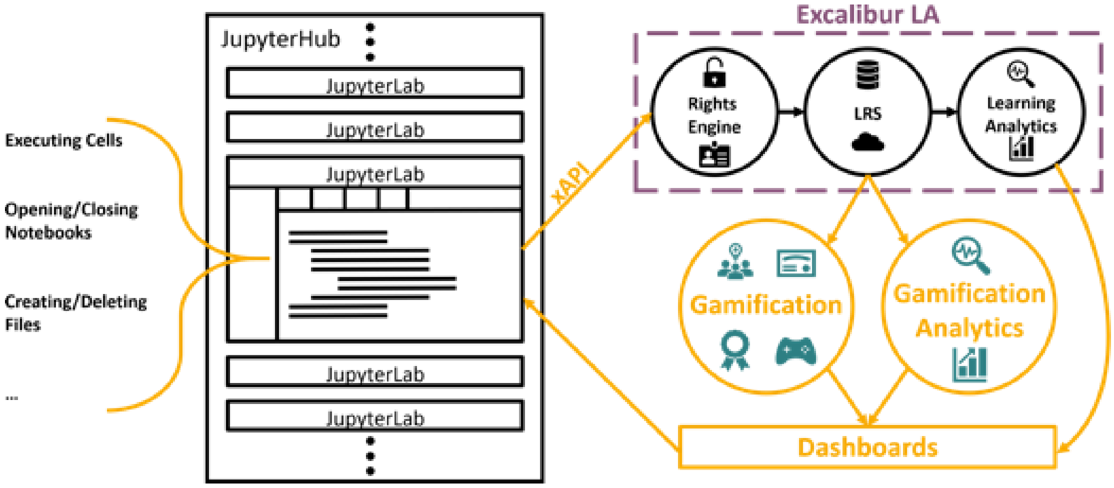


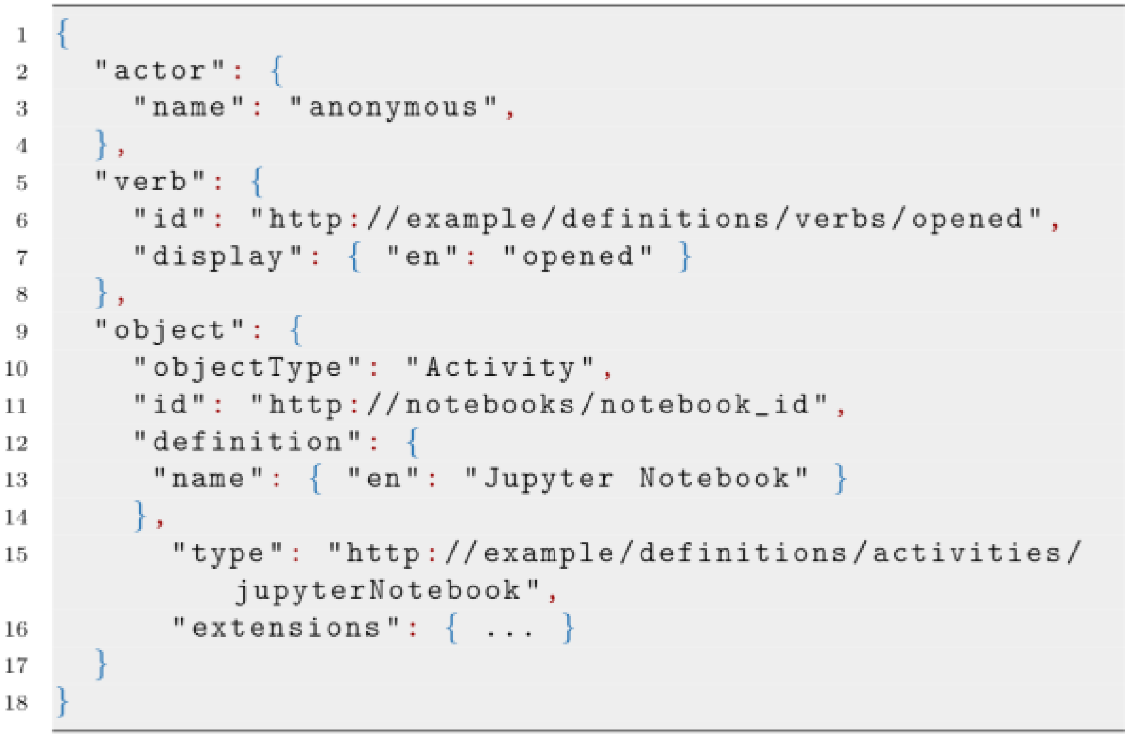
Figure 2: Visual representation for the research project.

3.1 Data Logging

Jupyter provides different options to log user interactions within the platform. As it is a web-based tool, events within the web frontend can be handled with JavaScript (Project Jupyter, 2021a). Further, internal event management is provided such that an extension can react on signals send by some objects (Project Jupyter, 2021b). This enables logging operations within the file system as well as opening or closing notebooks and can become as detailed as tracking single changes in code cells. With this, arbitrary reactions on multiple events are possible. However, an extension that transforms events and user interactions in xAPI statements and sends them to an LRS is currently not present and has to be implemented for this research project.

xAPI (Rustici Software LLC, 2021a, 2021b) is a data standard that enables the expression of online and offline experience in a JSON format that is both human and machine readable. Figure 3 shows an example of an xAPI statement with its three minimum attributes, actor, verb and object, stating who did what. This can be extended by context information, a timestamp or the outcome of an interaction, e. g. to log the content of a code cell, when it is executed and what the kernel returned.

This standard is used by the LA infrastructure to receive the logs from each platform in the same format, easing cross-platform analyses.



```
{
  "actor": {
    "name": "anonymous",
  },
  "verb": {
    "id": "http://example/definitions/verbs/opened",
    "display": { "en": "opened" }
  },
  "object": {
    "objectType": "Activity",
    "id": "http://notebooks/notebook_id",
    "definition": {
     "name": { "en": "Jupyter Notebook" }
    },
      "type": "http://example/definitions/activities/
         jupyterNotebook",
      "extensions": { ... }
  }
}
```

Figure 3: Example for an xAPI statement.

Data privacy aspects, like anonymous default logging with an option to opt-in into pseudonymous or personalized logging, will also be considered during the development of the logging extension. This will include the utilization of privacy and security features provided by the analytics infrastructure.

3.2 Data Analysis

For data analyses, analytics engines need to be developed and integrated into Excalibur LA. More precisely, for each game element to be integrated as well as the corresponding further analyses for GA, a dedicated gamification engine and analytics engine has to be designed and implemented. In the current version of this infrastructure, the data is examined every 24 hours. Such an implementation does not allow real-time investigations to be carried out. However, these are essential for the integration of gamification, as participants should, for example, be shown the achievement of a badge as soon as possible and not during the next day. The infrastructure therefore needs to be expanded to include an investigation that is as close to real-time as possible. In the field of LA and GA, an analysis of the data every 24 is sufficient as no intermediate data but the bigger picture will be considered.

Depending on the type of actor logging (anonymous, pseudonymous, personalized), different analyses can be carried out. The data of anonymous novices can only be used in a general context by the teacher. This is mainly about LA as well as GA, for example to interpret interaction data with the notebook across all users at the top level (How many times was a cell executed?) or to find out whether users interact with game elements.

Data from non-anonymous users, on the other hand, can be analysed much more deeply, so that a kind of individual profile can be created for each user. The gamification engines thereby access the xAPI statement in the LRS to gain new insights, but also the already analysed data in the Result Store to consider old analysis results if needed. All analysis results will be stored in the Result Store (see Figure 4). This prevents the need to run an analysis each time, the results should be further processed or visualized, and therefore reduces the server load, allowing to serve a higher number of clients at the same time. A data retriever can request the results via a REST API that allows only data reading but no manipulations and checks if the retriever is authorized to access the specified analysis results.

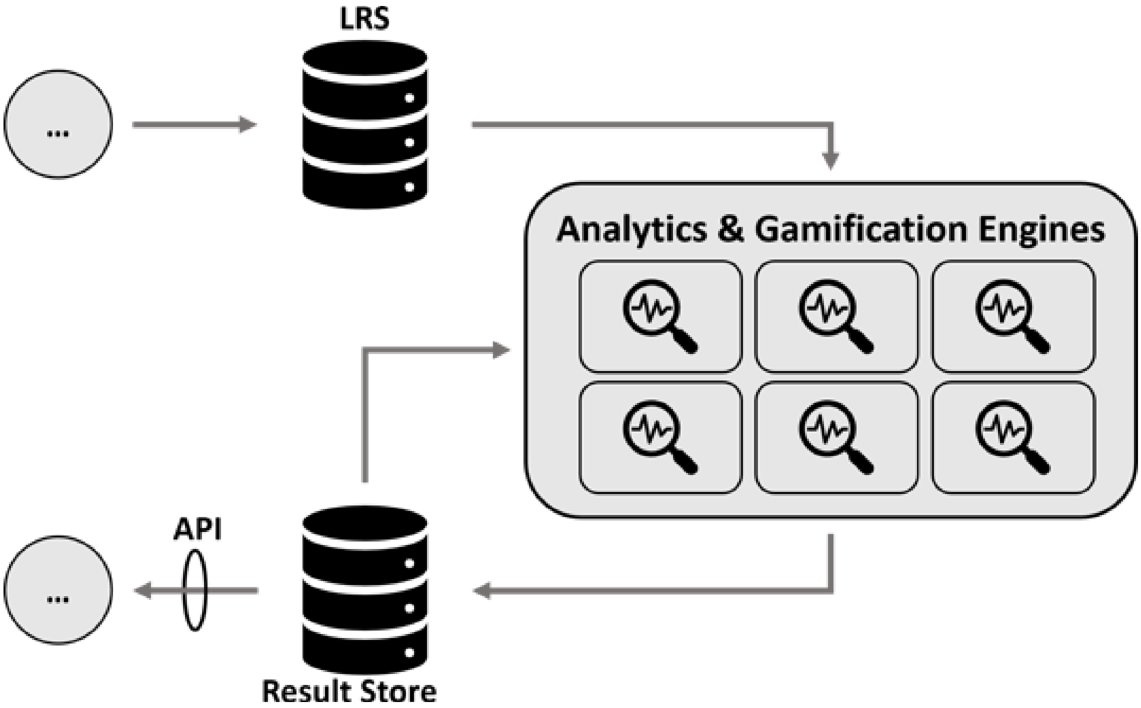


Figure 4: Visualisation of the Data Analysis for integrating gamification, GA and LA into Jupyter.

3.3 Reporting

To report the analyzed data to learners and lecturers, different dashboards and standalone visualizations have to be developed and integrated into Jupyter. A dashboard to give learners an overview of their achieved badges, their level or their position on a leaderboard, is the central reporting unit for learners. But it might be helpful to also include some of the visualizations in other parts of Jupyter. Therefore, single visualization components will be developed in a way that they can be included independently as well as integrated in a dashboard. In contrast, it is planned to display the results of LA and GA solely in a dedicated dashboard. This dashboard should be available to both learners and lecturers but adjusted to their needs. Learners should be able (if they allowed pseudonymous or personalized logging for themselves) to get reports on their performance and, if suitable, a comparison to the course average.

Lecturers should be provided with the data about the course average and, if allowed by the learners, some more details about single learners to be able to reflect their teaching, the integrated game elements and mechanics and to provide feedback and support to the learners. Further, the dashboards have to be designed to support the users data literacy skills to ease the interpretation of the data visualizations and support the decision making based on them.

4 Road Map

The development of this research project involves individual steps that have to be designed, realised and evaluated in several iterations.

In the first step, which has already been done, the open source data logger prototype Juxl (JupyterLab xAPI logging interface) was developed (Brocker, Judel, Schroeder, & Soeltzer, 2022), which logs selected interaction data and learning activities of the participants in JupyterLab and sends them to a LRS, where the data is initially stored. This prototype will be evaluated and improved within Python programming courses without concrete gamification and LA approaches, held at RWTH Aachen University within the next two semesters.

The next step is to develop a concept that identifies the game elements to be integrated. This should be based on a literature review, as described in subsection 2.1, as well as a comparison of online platforms that already implement gamification. Here, it is necessary to consider which data the respective game elements require and whether these can be logged within JupyterLab or whether it is allowed to log them regarding data privacy and ethical issues. The game elements identified in this process, together with the associated goals, are to be examined by potential target users in a theoretical manner. The focus here is to select the game elements that are most interesting for the users before the implementation phase.

Afterwards, the selected game elements are developed one at a time in an iterative process. Each iteration includes, if necessary, the extension of Juxl's logging capability and the realisation of new gamification engines and the adjustment of already existing ones. Furthermore, appropriate dashboard components have to be developed to visualise the game elements and make them accessible to the user.

Finally, the effects of the game elements will be examined with the help of Gamification Analytics and Learning Analytics. Similar to the realisation of the game elements, the data logging, analysis modules and dashboard must be expanded. A user study will be conducted in order to be able to record and evaluate the personal opinions of the participants. These results will then be taken into account during the preparation of the next iteration.

5 Conclusion

The aim of this research project is to provide novices with an easy introduction to programming. For this, the approaches of gamification, Gamification Analytics and Learning Analytics will be used in Jupyter using the Python programming language as an example. The next steps include to run analyses within a programming course at RWTH Aachen University in the winter term 2022/2023. This course will be extended by gamification elements such that Learning Analytics and Gamification Analytics can be run in parallel. The concept of this course and the analyses are currently under development but will be finished at the end of summer term 2022. As learning activities can also take place outside of Jupyter, for example within a learning management system, the relevant data is saved in the form of xAPI, as this data standard allows cross-platform analyses. In the long term, this should make cross-platform analyses possible, which in turn will lead to a larger data basis for the use of gamification and the resulting Gamification Analytics and Learning Analytics and should continuously simplify the entry into programming.

References

Agapito, J. L.; Rodrigo, M. T.: Identifying meaningful gamificationbased elements beneficial to novice programmers. In: Proceedings of the 26th International Conference on Computers in Education (ICCE). 2018, pp. 619-624. https://archium.ateneo.edu/discs-faculty-pubs/53/ (last check 2022-07-12)

Baumgartner, P.; Herber, E.: Höhere Lernqualität durch interaktive Medien? - Eine kritische Reflexion. In: Erziehung & Unterricht. 2013, pp. 327-335. https://peter.baumgartner.name/publikationen/liste-abstracts/abstracts-2013/hoehere-lernqualitaet-durch-interaktive-medien/ (last check 2022-07-12)

Bosse, Y.; Gerosa, M.: Why is programming so difficult to learn?: Patterns of difficulties related to programming learning mid-stage. ACM SIGSOFT Software Engineering Notes, 2017, 41, pp. 1-6. https://doi.org/10.1145/3011286.3011301 (last check 2022-07-12)

Brocker, A.; Judel, S.; Schroeder, U.; Soeltzer, Y.: Juxl: JupyterLab xAPI Logging Interface. In: 2022 International Conference on Advanced Learning Technologies (ICALT), 2022, pp. 158-160.

Conole, G.; Gasevic, D.; Long, P.; Siemens, G.: LAK '11 : proceedings of the 1st International Conference on Learning Analytics and Knowledge, February 27-March 1, 2011, Banff, Alberta, Canada.

Coursera: Beliebte Python-Kurse – Python online lernen — Coursera. 2021. https://de.coursera.org/search?query=python . (last check 2022-07-12)

Derus, S. R. M.; Ali, A. M. Z.: Difficulties in learning programming: Views of students. In: 1st International Conference on Current Issues in Education (ICCIE), 2012, pp. 74-79.

Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L. E.: From Game Design Elementsto Gamefulness: Defining “Gamification“. In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek), 2011, pp. 9-15. https://doi.org/10.1145/2181037.2181040 (last check 2022-07-12)

Heilbrunn, B.: Gamification Analytics: Support for Monitoring and Adapting Gamification Designs. In: Gamification - Using Game Elements in Serious Contexts. Springer, heidelberg u.a, 2017, pp. 31-47. https://link.springer.com/book/10.1007/978-3-319-45557-0#toc (last check 2022-07-12)

Judel, S.; Schroeder, U.: EXCALIBUR LA - An Extendable and Scalable Infrastructure Build for Learning Analytics. In: 2022 International Conference on Advanced Learning Technologies (ICALT), 2022, pp. 155-157.

Jupyter Team: The jupyter notebook. 2021. https://jupyter-notebook.readthedocs.io/en/stable/notebook.html . (last check 2022-07-12)

Koivisto, J.; Hamari, J.: The rise of motivational information systems: A review of gamification research. In: International Journal of Information Management, 2019, 45, pp. 191-210. https://doi.org/10.1016/j.ijinfomgt.2018.10.013 (last check 2022-07-12)

Marín, B.; Frez, J.; Cruz-Lemus, J. A.; Genero, M.: An Empirical Investigation on the Benefits of Gamification in Programming Courses. In: ACM Transactions on Computing Education (TOCE), 2019, 19, pp. 1 - 22. https://doi.org/10.1145/3231709 (last check 2022-07-12)

Nicholson, S.: A RECIPE for Meaningful Gamification. In: Reiners, T.; Wood, L. (Eds.): Gamification in Education and Business. Springer, 2015, pp. 1-20. DOI: 10.1007/978-3-319-10208-5 (last check 2022-07-12)

Project Jupyter. (2021a): Design Patterns — JupyterLab 4.0.0a15 documentation. https://jupyterlab.readthedocs.io/en/latest/developer/patterns.html#dom-events . 2021. (last check 2022-07-12)

Project Jupyter. (2021b): Design Patterns — JupyterLab 4.0.0a15 documentation. https://jupyterlab.readthedocs.io/en/latest/developer/patterns.html#signals . 2021. (last check 2022-07-12)

Project Jupyter. (2021c): Jupyter. https://jupyter.org/ . 2021. (last check 2022-07-12)

Project Jupyter. (2021d): The jupyterlab interface. https://jupyterlab.readthedocs.io/en/latest/user/interface.html . 2021. (last check 2022-07-12)

Project Jupyter team. (2016): Jupyterhub. https://jupyterhub.readthedocs.io/en/stable/ . 2016. (last check 2022-07-12)

Rojas, M. E. O.; Chiluiza, K.; Valcke, M.: Gamification in computer programming: effects on learning, engagement, self-efficacy and intrinsic motivation. In: 11th European Conference on Games Based Learning (ECGBL), 2017, pp. 507-514.

Rustici Software LLC. (2021a): What is the Experience API? 2021. https://xapi.com/overview/ . (last check 2022-07-12)

Rustici Software LLC. (2021b): xAPI Statements: The anatomy of an xAPI statement. 2021. https://xapi.com/statements-101/ . (last check 2022-07-12)

Venter, M. I.: Gamification in STEM programming courses: State of the art. In: 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 859-866. DOI: 10.1109/EDUCON45650.2020.9125395 (last check 2022-07-12)