COLLABORATIVE GAMIFICATION DESIGN FOR SCIENTIFIC SOFTWARE

Position Paper

Francisco Queiroz [1,2]
Rejane Spitz [2]

[1] *Tecgraf Institute, PUC-Rio*
[2] *Arts & Design Department, PUC-Rio*

WSSSPE4
4th Workshop on Sustainable Software for Science: Practice and Experiences

University of Manchester, Manchester, UK
12-14 Sep. 2016

SOFTWARE SUSTAINABILITY

SOFTWARE SUSTAINABILITY [1]

- EXTENSIBILITY
- INTEROPERABILITY
- MAINTAINABILITY
- PORTABILITY
- REUSABILITY
- SCALABILITY
- USABILITY

[1] C. Venters, L. Lau, M. Griffiths, V. Holmes, R. Ward, C. Jay, C. Dibsdale, and J. Xu, "The blind men and the elephant: Towards an empirical evaluation framework for software sustainability," *Journal of Open Research Software*, vol. 2, no. 1, 2014.

SOFTWARE SUSTAINABILITY [1]

- USABILITY

[1] C. Venters, L. Lau, M. Griffiths, V. Holmes, R. Ward, C. Jay, C. Dibsdale, and J. Xu, "The blind men and the elephant: Towards an empirical evaluation framework for software sustainability," *Journal of Open Research Software*, vol. 2, no. 1, 2014.

SOFTWARE SUSTAINABILITY

- USABILITY

GAMIFICATION[2]

[2] C. Wolff, "The Case for Teaching 'Tool Science,'" in 2015 IEEE Global Engineering Education Conference, 2015, pp. 932–938.

SOFTWARE SUSTAINABILITY

- USABILITY

GAMIFICATION DESIGN

(AWARE OF)

EXTENSIBILITY
INTEROPERABILITY
MAINTAINABILITY
PORTABILITY
REUSABILITY
SCALABILITY
USABILITY

GAMIFICATION

GAMIFICATION

- **“The use of game design elements in non-game contexts” [3]**
  - Deterding et al.
- ***Gameful* design: designing an activity as a game. (often involves goals, points, etc.)**

[3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, “From game design elements to gamefulness: defining gamification,” in Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, 2011, pp. 9–15.

GAMIFICATION

- **"The use of game design elements in non-game contexts" [3]**
  - Deterding et al.
- ***Gameful* design: designing an activity as a game. (often involves goals, points, etc.)**

- **"Gamification should be composed of cross media references from games to other products" [4]**
  - Popa
- **Transposition of game-like aesthetics and interactivity to other media**

[3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, 2011, pp. 9–15.

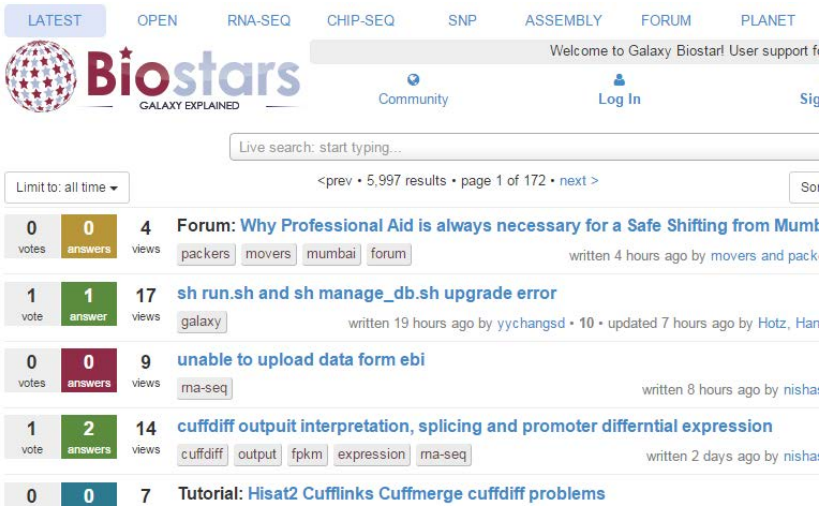
[4] D. M. Popa, "Design Case: Gamification of ERP – A user centered design approach," 2013 [Online]. Available: http://gamification-research.org/wp-content/uploads/2013/03/Popa.pdf

GAMIFICATION | SCIENTIFIC SOFTWARE (and correlated areas)

- Community building and exchange of knowledge.
- Use of gameplay, game aesthetics and/or technology by engineering software.
- Gamification of citizen science and science education.

GAMIFICATION | SCIENTIFIC SOFTWARE (and correlated areas)

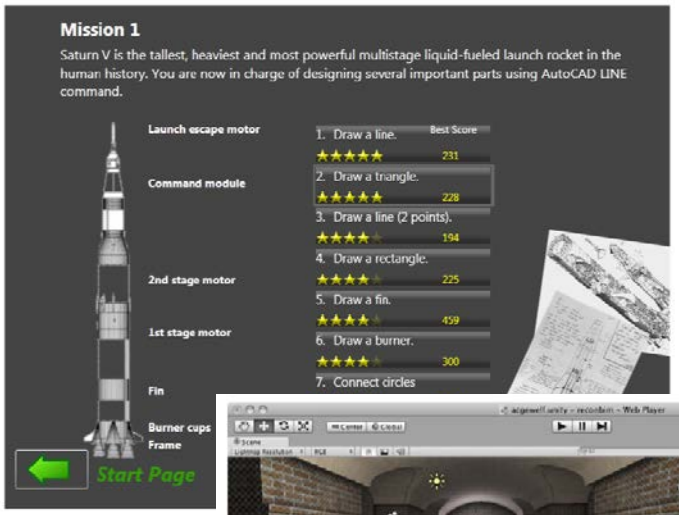
- Community building and exchange of knowledge.
- Use of gameplay, game aesthetics and/or technology by engineering software.
- Gamification of citizen science and science education.



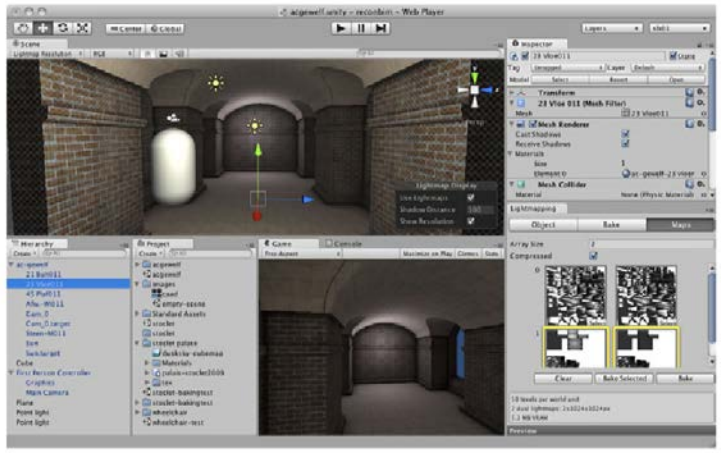
biostar.usegalaxy.org

GAMIFICATION | SCIENTIFIC SOFTWARE (and correlated areas)

- Community building and exchange of knowledge.
- Use of gameplay, game aesthetics and/or technology by engineering software.
- Gamification of citizen science and science education.



[5]



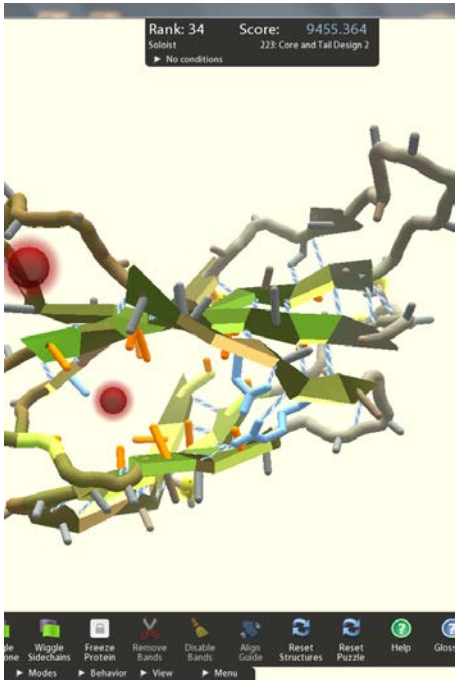
[6]

[5] W. Li, T. Grossman, and G. Fitzmaurice, "Gamicad: a gamified tutorial system for first time autocad users," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 2012, pp. 103–112.

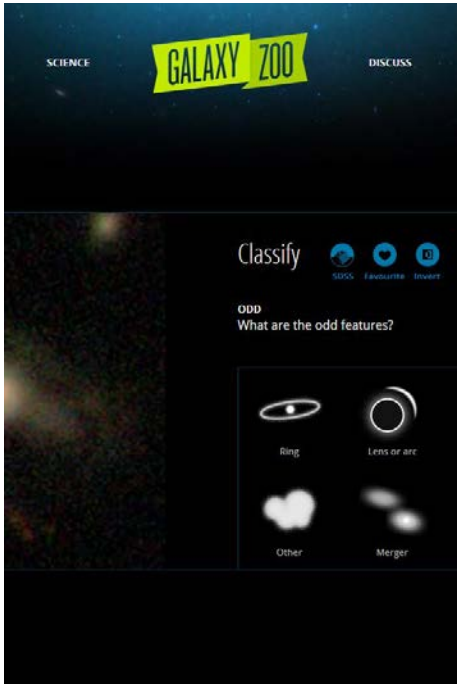
[6] S. Boeykens, "Using 3D Design software, BIM and game engines for architectural historical reconstruction," CAAD Futures, Liege, Belgium, 2011.

GAMIFICATION | SCIENTIFIC SOFTWARE (and correlated areas)

- Community building and exchange of knowledge.
- Use of gameplay, game aesthetics and/or technology by engineering software.
- Gamification of citizen science and science education.



Fold.it[7]



Galaxy Zoo[8]

[7] S. Cooper, A. Treuille, J. Barbero, A. Leaver-Fay, K. Tuite, F. Khatib, A. C. Snyder, M. Beenen, D. Salesin, D. Baker, and others, "The challenge of designing scientific discovery games," in Proceedings of the Fifth international Conference on the Foundations of Digital Games, 2010, pp. 40–47.

[8] B. Simmons, C. Lintott, K. Masters, A. Greenhill, G. Graham, and K. Holmes, "Defining and Measuring Success in Online Citizen Science: A Case Study," Computing in Science \& Engineering, vol. 17, p. 28, 2015.

OUR PREVIOUS EFFORTS

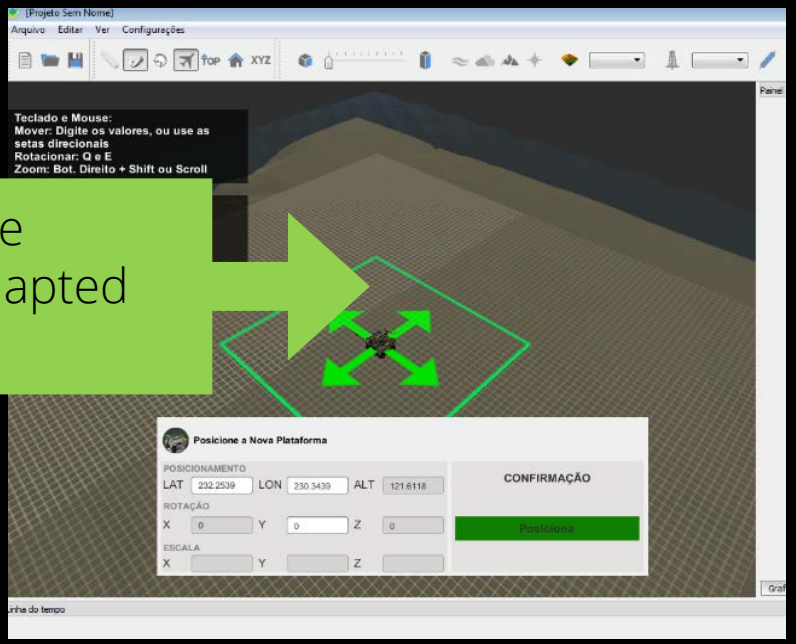
- Action research: Gamified functionalities for a simulation software (oil and gas industry).
- Literature review: Scientific software development, use, user interfaces, and gamification.

Action Research[9]

- Research, adaptation and prototyping of design elements.
- Lessons Learned include:
  - Criteria for researching games
  - Scientific software needs
  - Difficulties in gameful design
  - Perils of hi-fidelity prototyping
  - Impediments to implementation

New control scheme for 4D navigation (better insight)

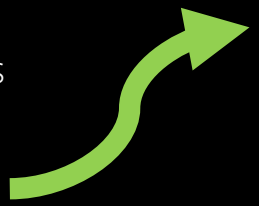
Wiimote-compatible positioning tool (adapted for precision)



[9] F. Queiroz, R. Spitz, P. Elias, R. Pinheiro, T. Azevedo, V. Rodrigues, L. Reis, A. Raposo, "Video games as inspiration for scientific software"

Literature Review[10]

- Investigation on scientific software
  - Development
  - Use
  - User interfaces
  - Gamification
- Elaboration of a Design Lens (guidelines)



THE LENS OF THE LAB

Scientific software should **augment insight, productivity, and knowledge**. It should facilitate and integrate supported stages of scientific work (modeling, simulation and result analysis, and generate output for publication, sharing, or further research.
When designing for scientific software, consider the questions:

- How can the interface **represent the scientific matter**, reinforce the way it works and support the theory behind it? How can it **present and explore complex data** at high levels of precision? How can it prevent and fix errors?

- Is the user interface intuitive, consistent and uncluttered? Is it flexible enough to allow for **incremental expansion and customization**? Is it adequate to the platforms it was designed for, and to other software it should be integrated to?

- **How do scientists work?** How is the work environment, culture, ethics, conventions, current practices and best practices? What do users need and expect? How can design embrace different levels of scientific specialization, computer literacy, and programming skills? How can it **promote and attract collaboration** or community building?

- **How can games inform and inspire** the software aesthetics and interactivity? Which game design elements could provide **structure, goals, feedback, guidance, progression, flow, fun and experimentation?** Would competition and point-based systems motivate or demotivate?

- **Is implementation feasible** regarding scope, planning, timescale, technologies, human resources, and software lifecycle?

[10] F. Queiroz, R. Spitz. "The Lens of the Lab: Design Challenges in Scientific Software"

How to establish good communication, during design stages, between all stakeholders?

- Understand needs
- Gather requirements
- Discuss possibilities

COLLABORATIVE AND OPEN DESIGN

- Can address typical challenges from scientific software
- Employed in many successful development cases
- Compatible with scientific software development culture

COLLABORATIVE AND OPEN DESIGN

- Can address typical challenges from scientific software
- Employed in many successful development cases
- Compatible with scientific software development culture

- **Understanding** specialized users and their work. [14, 15, 16]
- **Empowering** and responding to *actual* user-base.[17]
- **Integrating** disciplines[18], visions[19], schedules [20, 21] and authorship.[22]

[14] C. M. Pancake, "'Improving the Usability of Numerical Software through User-Centered Design"
[15] R. R. Springmeyer, "Applying observations of work activity in designing prototype data analysis tools"
[16] P. De Matos, J. A. Cham, H. Cao, R. Alcántara, F. Rowland, R. Lopez, and C. Steinbeck, "The Enzyme Portal: a case study in applying user-centred design methods in bioinformatics"
[17] D. De Roure and C. Goble, "Software design for empowering scientists"
[18] C. Chen, J. Zhang, and M. S. Vogeley, "Reflections on the Interdisciplinary Collaborative Design of Mapping the Universe"
[19] M. Spencer, "Brittleness and Bureaucracy: Software as a Material for Science"
[20] D. Kelly and S. Smith, "3rd CASCON Workshop on Software Engineering for Science"
[21] G. O. Domik and K. D. Mickus-Miceli, "Software design and development in a scientific environment: lessons learned during the development of STAR, an astrophysical analysis and visualization package"
[22] M. Turk, "Fostering Collaborative Computational Science"

COLLABORATIVE AND OPEN DESIGN

- Can address typical challenges from scientific software
- Employed in many successful development cases
- Compatible with scientific software development culture

- Project Community / **OMERO** software.[23]
- **European Middleware Initiative**: Community-based discussion tools.[24]
- **STAR** software iterative design process.[25]
- **University of Illinois' *hackathon*** for taxonomy software interface.[26]
- **biok**: programmable software co-designed and co-developed with users.[27]

[23] S. Loynton, J. M. Burel, D. Sloan, and C. Macaulay, "The 'Project Community Approach' to Academic Scientific Software Development"

[24] A. Di Meglio, F. Estrella, and M. Riedel, "On realizing the concept study ScienceSoft of the European Middleware Initiative: Open Software for Open Science"

[25] G. O. Domik and K. D. Mickus-Miceli, "Software design and development in a scientific environment: lessons learned during the development of STAR, an astrophysical analysis and visualization package"

[26] A. K. Thomer, M. B. Twidale, J. Guo, and M. J. Yoder, "Co-designing Scientific Software: Hackathons for Participatory Interface Design"

[27] C. Letondal and W. E. Mackay, "Participatory programming and the scope of mutual responsibility: balancing scientific, design and software commitment"

COLLABORATIVE AND OPEN DESIGN

- Can address typical challenges from scientific software
- Employed in many successful development cases
- Compatible with scientific software development culture

- "**Do-it-yourself**" mentality, amateurism, shared knowledge.[28]
- Computational science communities pioneered **'Open'** movement.[29]
- "**Professional End User Developer**".[30]

[28] P. Atkinson, "Orchestral manoeuvres in design"

[29] J. Dongarra, G. H. Golub, E. Grosse, C. Moler, and K. Moore, "Netlib and NA-Net: Building a scientific computing community"

[30] J. Segal, "Some Problems of Professional End User Developers"

OPEN GAMIFICATION DESIGN*

Open access to the conceptualization and planning of gamified functionalities in a collaborative manner, in order to make the design phase as informed as possible.

*Ideally supported by a designer

THE DESIGN BOARD*

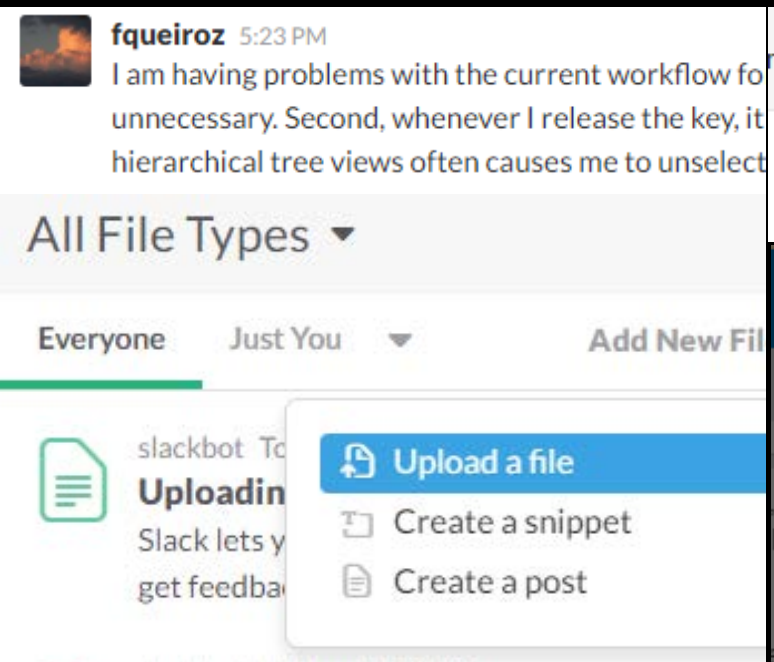
Conversational media used as a tool for collaborative design.

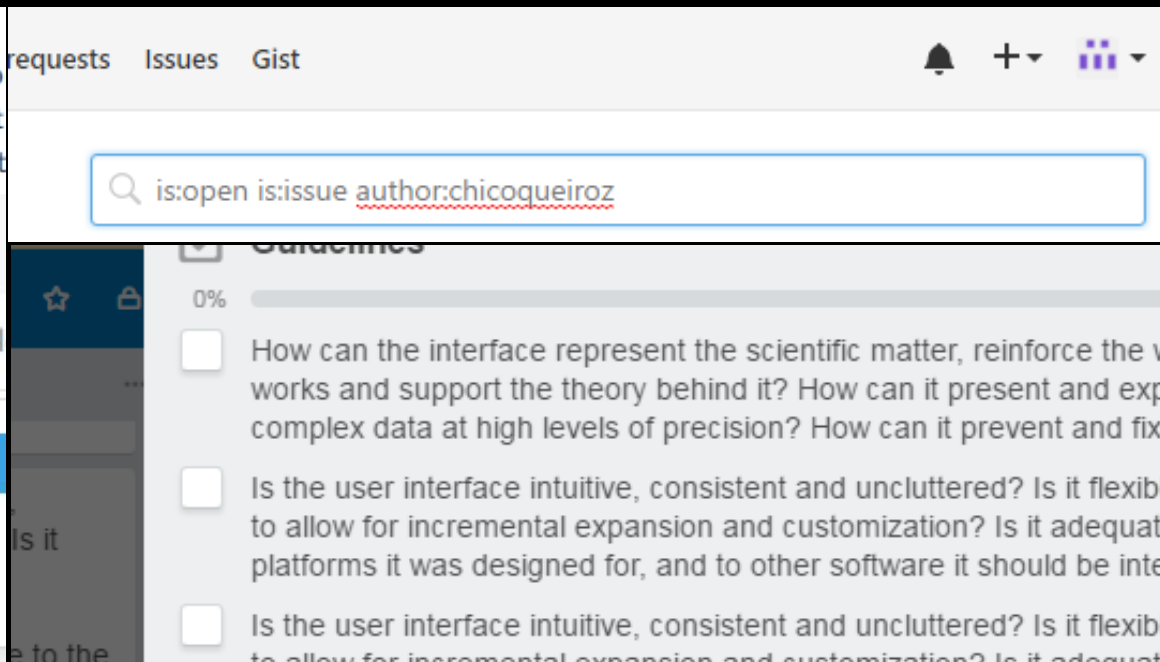
*Ideally used in conjunction with the Lens of the Lab

CONVERSATIONAL MEDIA

Blogs, Forums, CMSs, Trello, GitHub... any platform that allows participants to:

- Initiate and join **discussions**.
- Publish and access **supportive materia**l (e.g., text documents, images).
- **Search** and/or browse past discussions.
- Access appropriate design **guidelines** (i.e., the Lens of the Lab)





ILLUSTRATIVE SCENARIO

I am having problems categorizing geometry: Having to keep the CTRL key down all the time feels tiring. When I accidently release the key, it makes me unselect everything. Also, dragging names between hierarchical tree views often causes me to unselect everything.

I am having problems categorizing geometry: Having to keep the CTRL key down all the time feels tiring. When I accidently release the key, it makes me unselect everything. Also, dragging names between hierarchical tree views often causes me to unselect everything.

The Lens of the Lab

[...]

- How can games inform and inspire the software's aesthetics and interactivity?

I am having problems categorizing geometry: Having to keep the CTRL key down all the time feels tiring. When I accidently release the key, it makes me unselect everything. Also, dragging names between hierarchical tree views often causes me to unselect everything.

In *Boookworm*, you could select pieces by clicking on them and double-click the last one to send the selected pieces outside the board. We could have a similar mechanic, but also highlighting the folder named after the selected category

I am having problems categorizing geometry: Having to keep the CTRL key down all the time feels tiring. When I accidently release the key, it makes me unselect everything. Also, dragging names between hierarchical tree views often causes me to unselect everything.

In *Boookworm*, you could select pieces by clicking on them and double-click the last one to send the selected pieces outside the board. We could have a similar mechanic, but also highlighting the folder named after the selected category

The Lens of the Lab

[...]

- Which game design elements could provide feedback and progression?

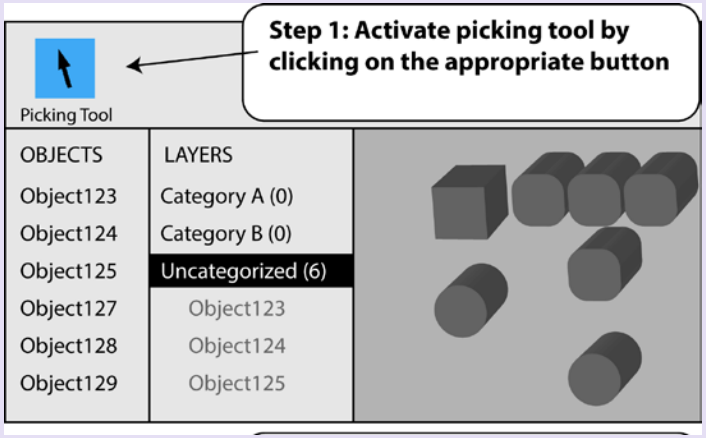
[...]

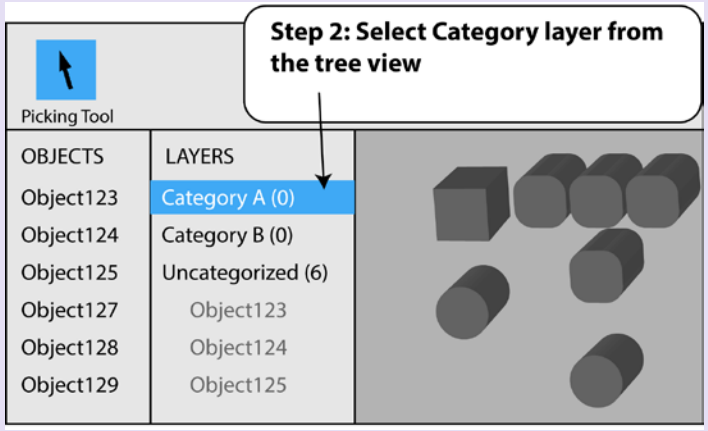
- Is implementation feasible?

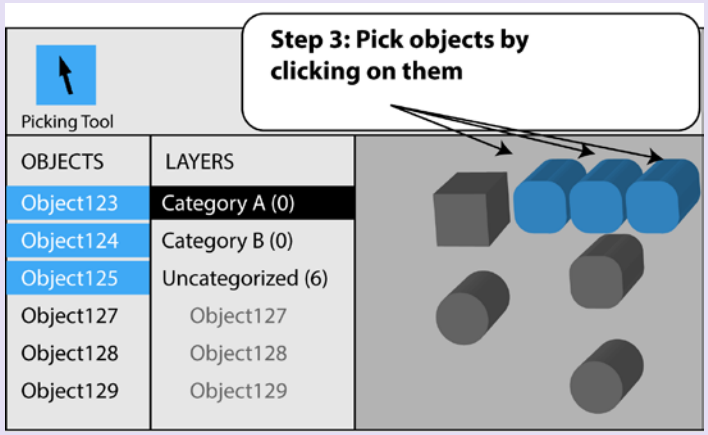
I am having problems categorizing geometry: Having to keep the CTRL key down all the time feels tiring. When I accidently release the key, it makes me unselect everything. Also, dragging names between hierarchical tree views often causes me to unselect everything.

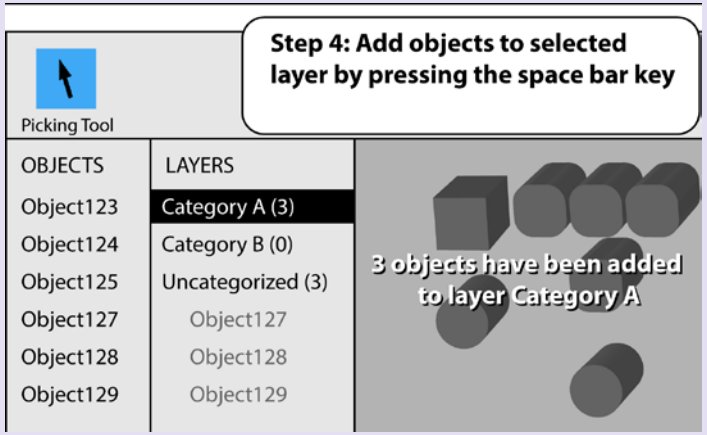
In *Boookworm*, you could select pieces by clicking on them and double-click the last one to send the selected pieces outside the board. We could have a similar mechanic, but also highlighting the folder named after the selected category

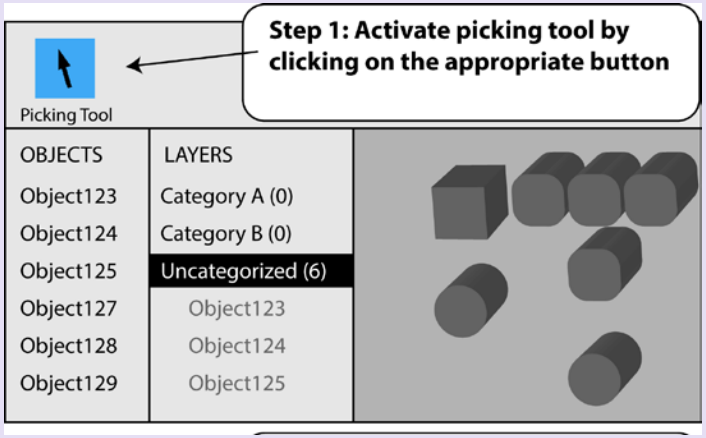
Given our UI development framework, we cannot highlight a layer on the hierarchy tree, but we could display an on-screen message such as “100 objects have been added to Category A layer”.
Also, we could add an object counter beside that layer’s name in the tree view.

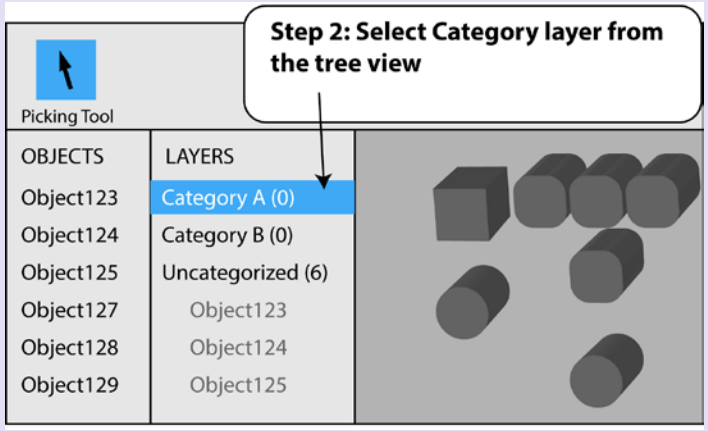

Step 1: Activate picking tool by clicking on the appropriate button
Picking Tool
OBJECTS
LAYERS
Object123
Category A (0)
Object124
Category B (0)
Object125
Uncategorized (6)
Object127
Object123
Object128
Object124
Object129
Object125

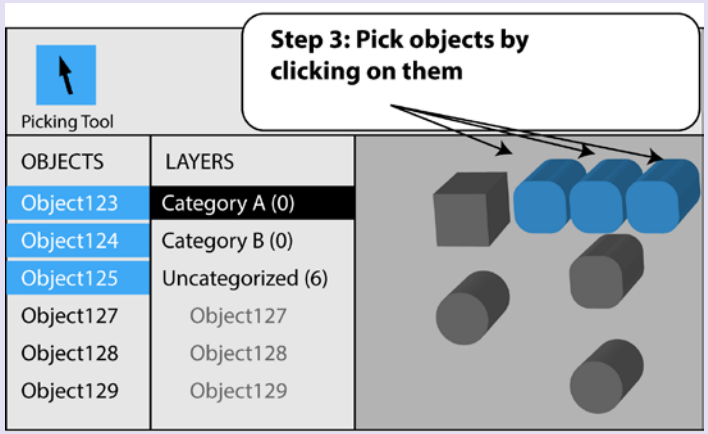

Step 2: Select Category layer from the tree view
Picking Tool
OBJECTS
LAYERS
Object123
Category A (0)
Object124
Category B (0)
Object125
Uncategorized (6)
Object127
Object123
Object128
Object124
Object129
Object125

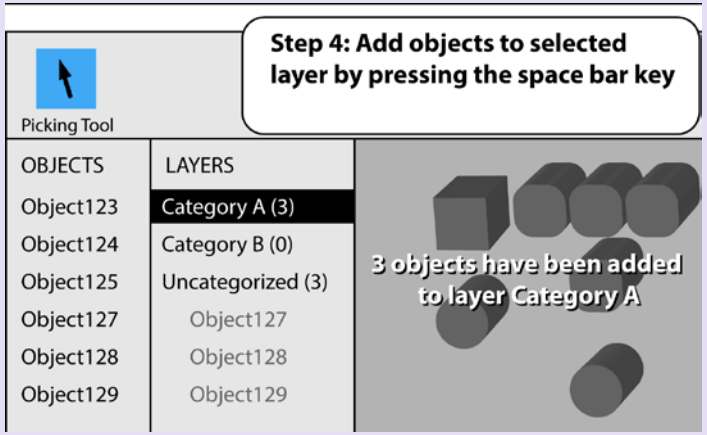

Step 3: Pick objects by clicking on them
Picking Tool
OBJECTS
LAYERS
Object123
Category A (0)
Object124
Category B (0)
Object125
Uncategorized (6)
Object127
Object127
Object128
Object128
Object129
Object129


Step 4: Add objects to selected layer by pressing the space bar key
Picking Tool
OBJECTS
LAYERS
Object123
Category A (3)
Object124
Category B (0)
Object125
Uncategorized (3)
Object127
Object127
Object128
Object128
Object129
Object129
3 objects have been added to layer Category A









OK

OK, let’s add it to the backlog.

POTENTIAL GAINS

- Better usability
- Better compliance /user acceptance
- Less time between design and implementation

NEXT STEPS

- Find opportunities for testing the design board and the Lens of the Lab.
- Further research on scientific software gamification

Thank you !

Special thanks:

- The Gordon and Betty Moore Foundation
- National Center for Supercomputing Applications
- University of Illinois
- WSSSPE4 organizing committee.

chico@puc-rio.br