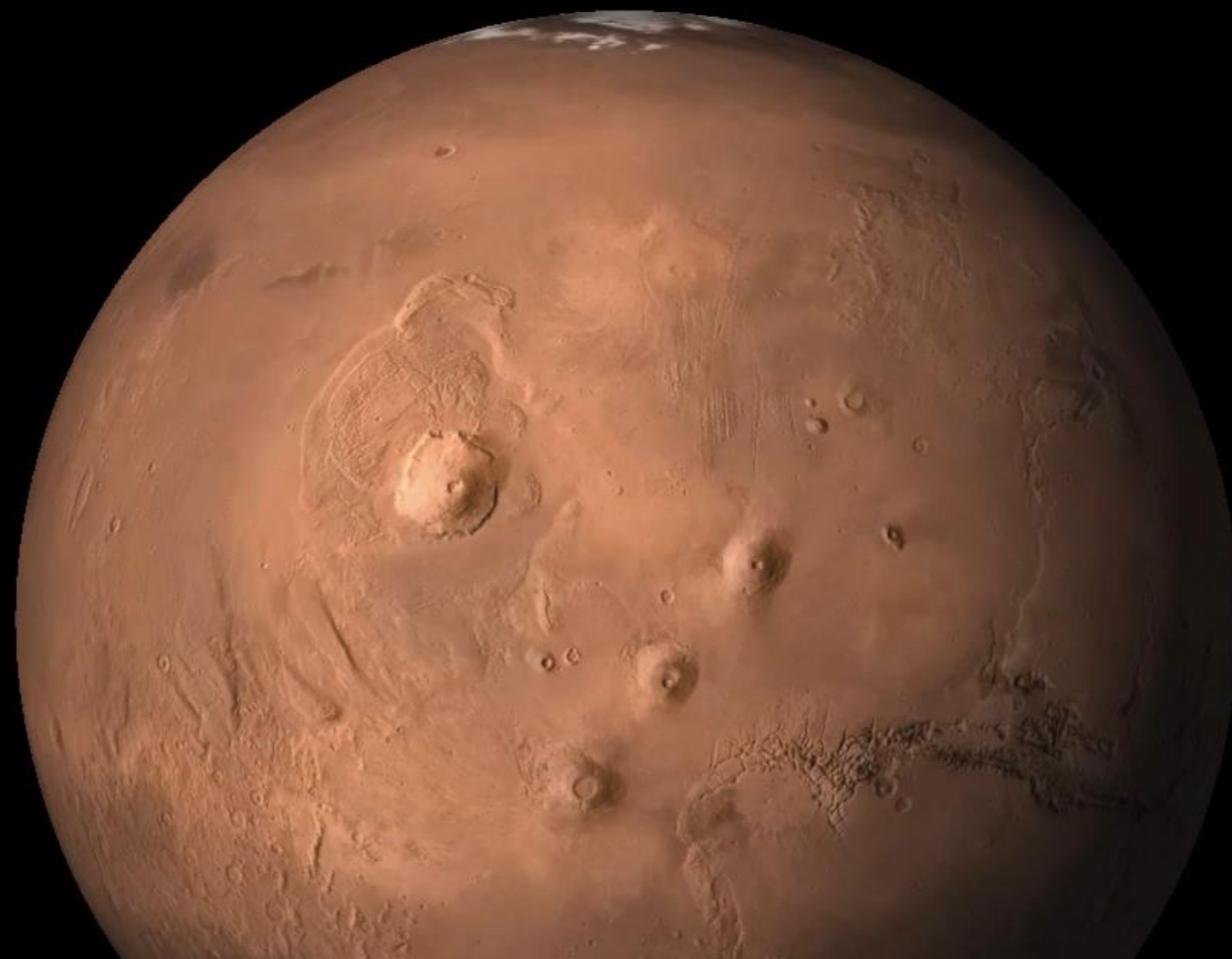


WebGL Engine Programming

Levan Khelashvili

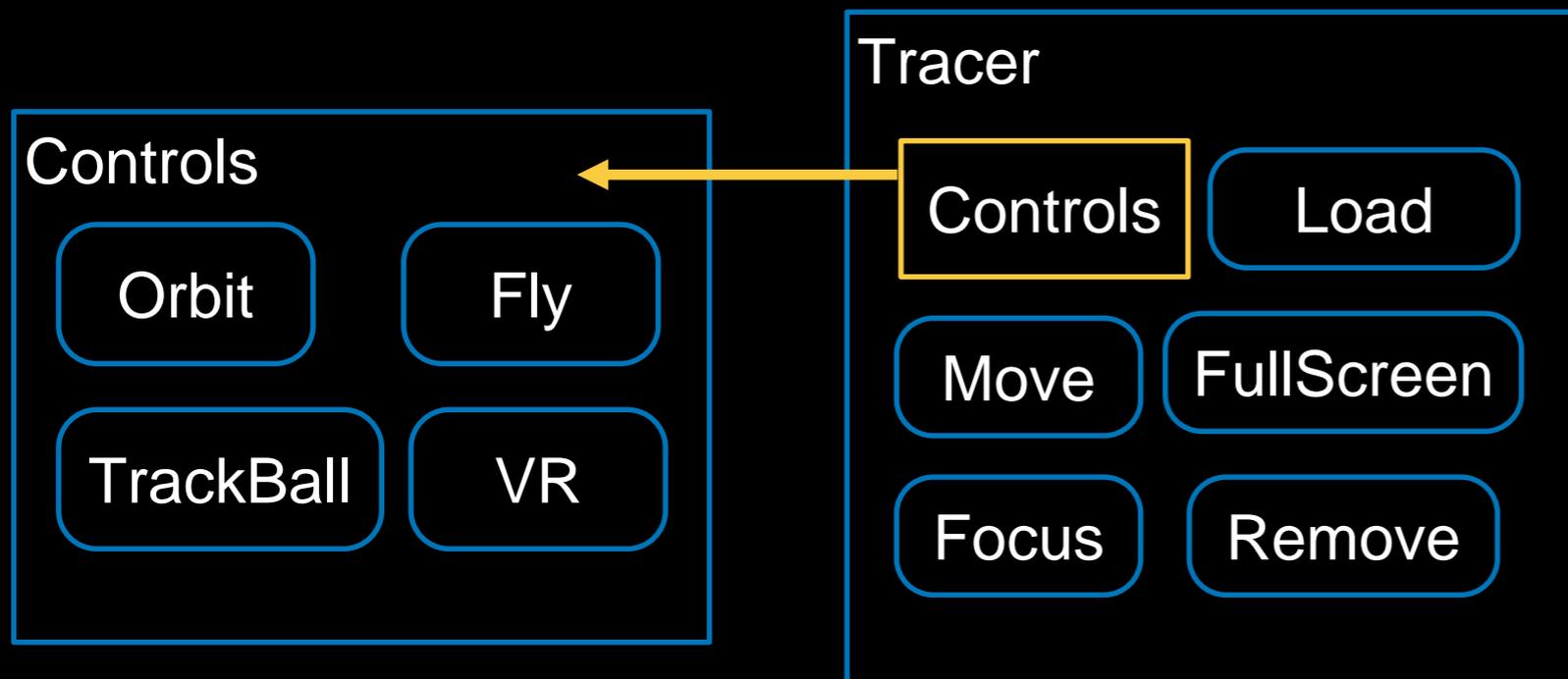


Application System

- We are making Application using JavaScript based on ThreeJS WebGL.
- ThreeJS is rich Web Graphic Library that makes 3D Rendering possible for web browsers.
- ThreeJS contains lots of tools, like physics and lighting.

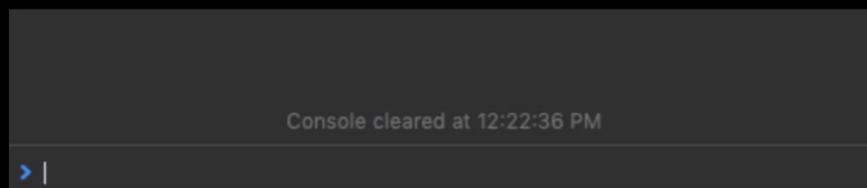


Tracer as Application Core



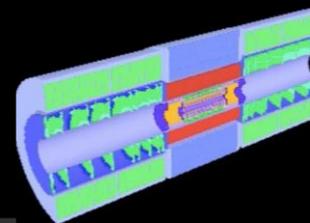
I have made class **Tracer** that acts as Interface for other classes and is Engine that contains all functions that are available in system

Using Class **Tracer** we can make any 3D application as simple as typing what we need. For example, if we want to change geometry's material's opacity we should do this:



Currently we have Atlas **Tracer** as core that contains all features that we have made. We can make

Tracer Tile Cal or
Tracer Sct Inn

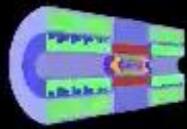


Tracer as Application Core

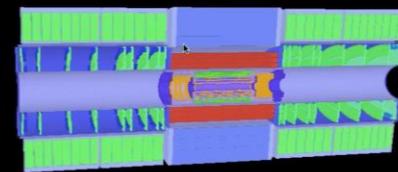
We have made **Application** that can be **Core** of other **3D applications**

Tracer has much more than you need to build new 3D application

Controls



Fly



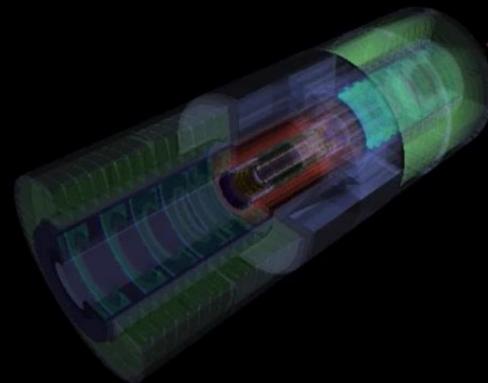
Orbit

What do we need to Build 3-Dimensional Application?

Object Loading



Animations



Cameras
Controls
Animations
Object Loading
Performance monitor
Multiple Modes
Device Detection
Fullscreen mode
Lights

We got it all
and much more...

Tracer is dead, long live the Tracer!

November 4 2018

April 8 2019



4 years of development
4 different programmers
4 different views
30+ releases
1 structure?
NO!

For all this reasons we started Rewriting Tracer in November 4 2018.

During this process we started using GitLab to prevent bugs. After finishing task no we send request on gitlab, everyone tests for bugs and then we merge update.

After 4 month, my 45 tasks and hard work we changed structure of Tracer to Object-Oriented programming structure.

This update was not only restructuring. It was new beginning for Tracer.

1. Performance is greatly improved
2. Added functions that was required for system to work properly
3. Monitoring system was improved.
4. All functions were updated.
5. Now we are using new ThreeJs version
6. Its easier to change, modify and use engine

Years of *structured programming* got us to dead end!

1. Bug fixes were causing more bugs.
2. Because of Structured programming, modifying functions were causing even bugs.
3. Code was not reusable
4. Functions needed update
5. ThreeJS version was getting old and upgrade was impossible because of structure

50 functions

We had

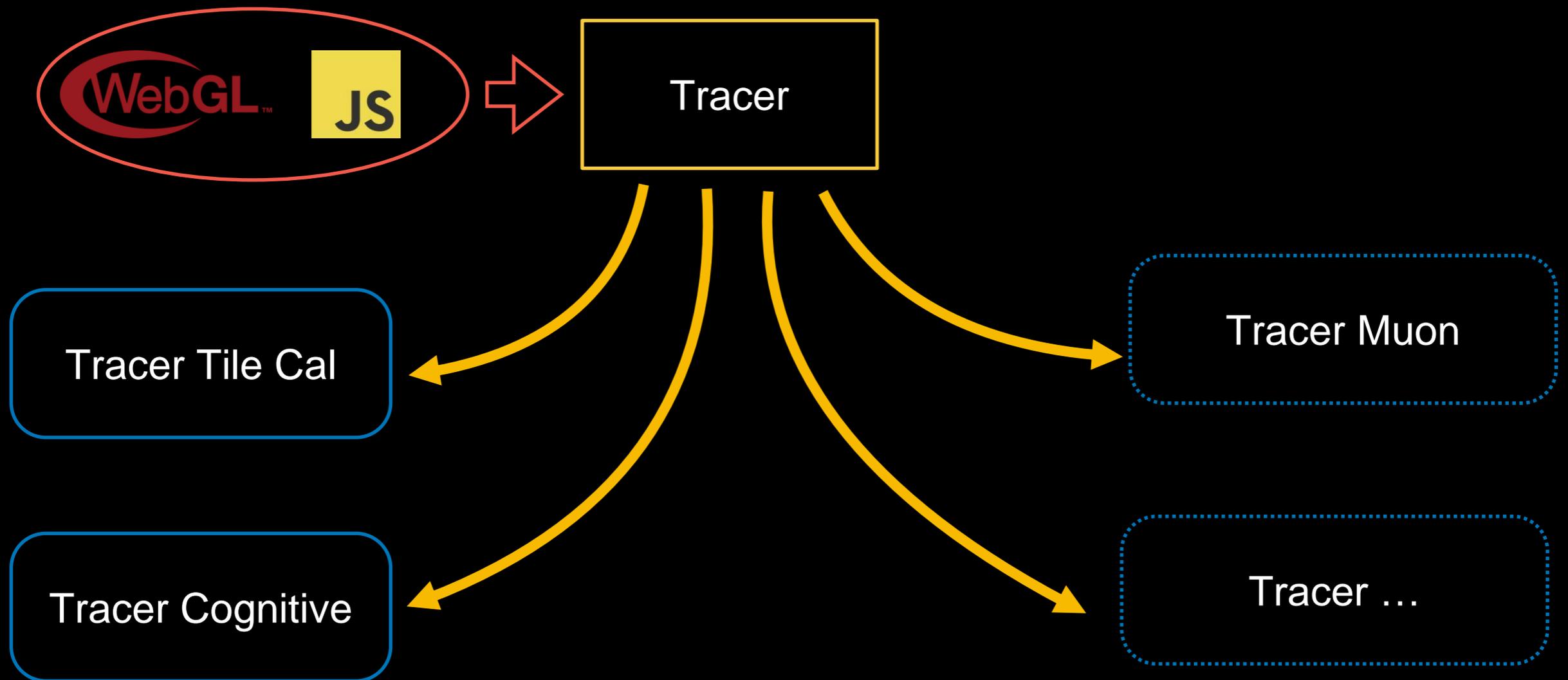
Now we have

1 class
tracer

7 classes

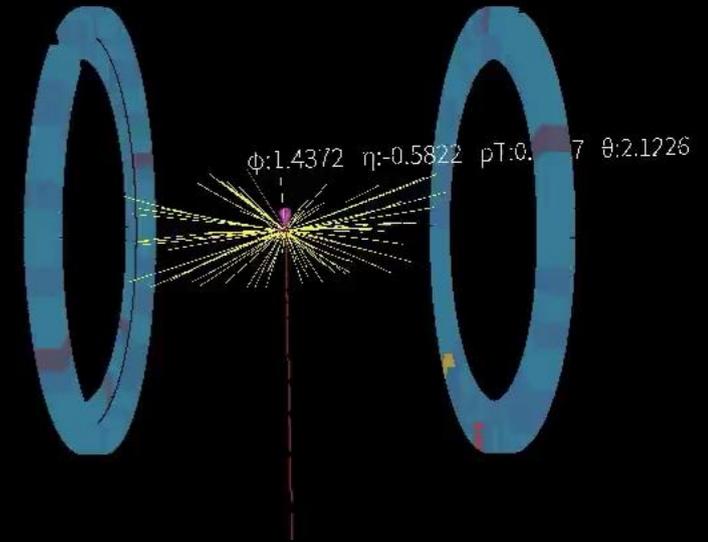
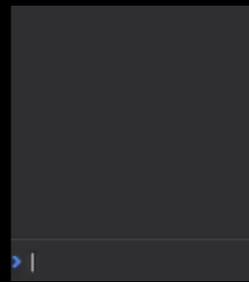
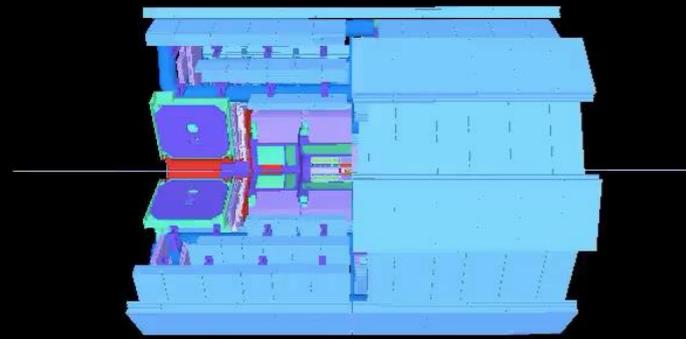
60+ functions

Tracer opens doors to other applications



New rewritten Tracer engine is reusable and can be reused to develop new Tracer applications for any purpose, is it scientific, educational, or visualization.

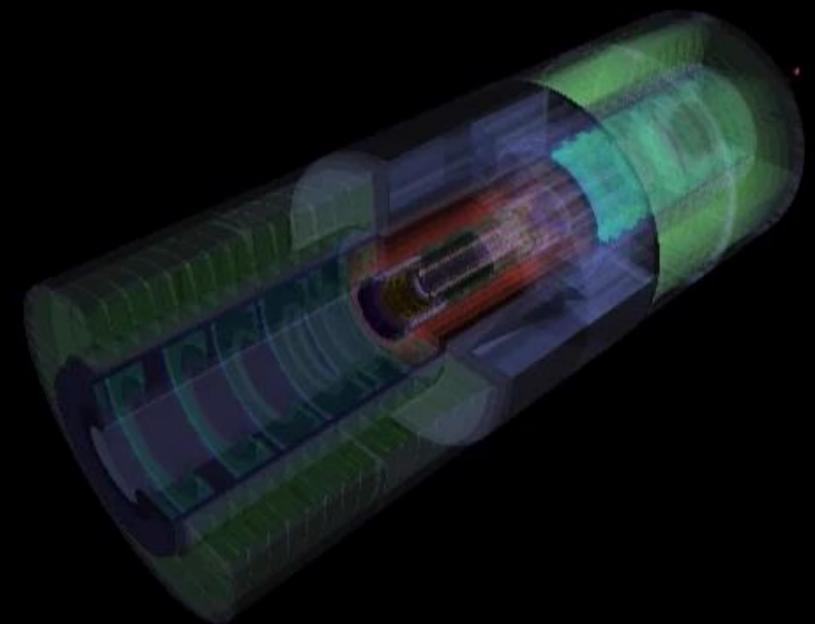
Atlas Tracer



Detector Display + Engine + Event Display

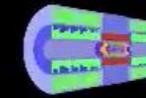
See Most of features
at

tracer.web.cern.ch



Future steps

1 We are currently implementing Fly mode in our engine. It is already complete for Desktop Computers in My Development folder.

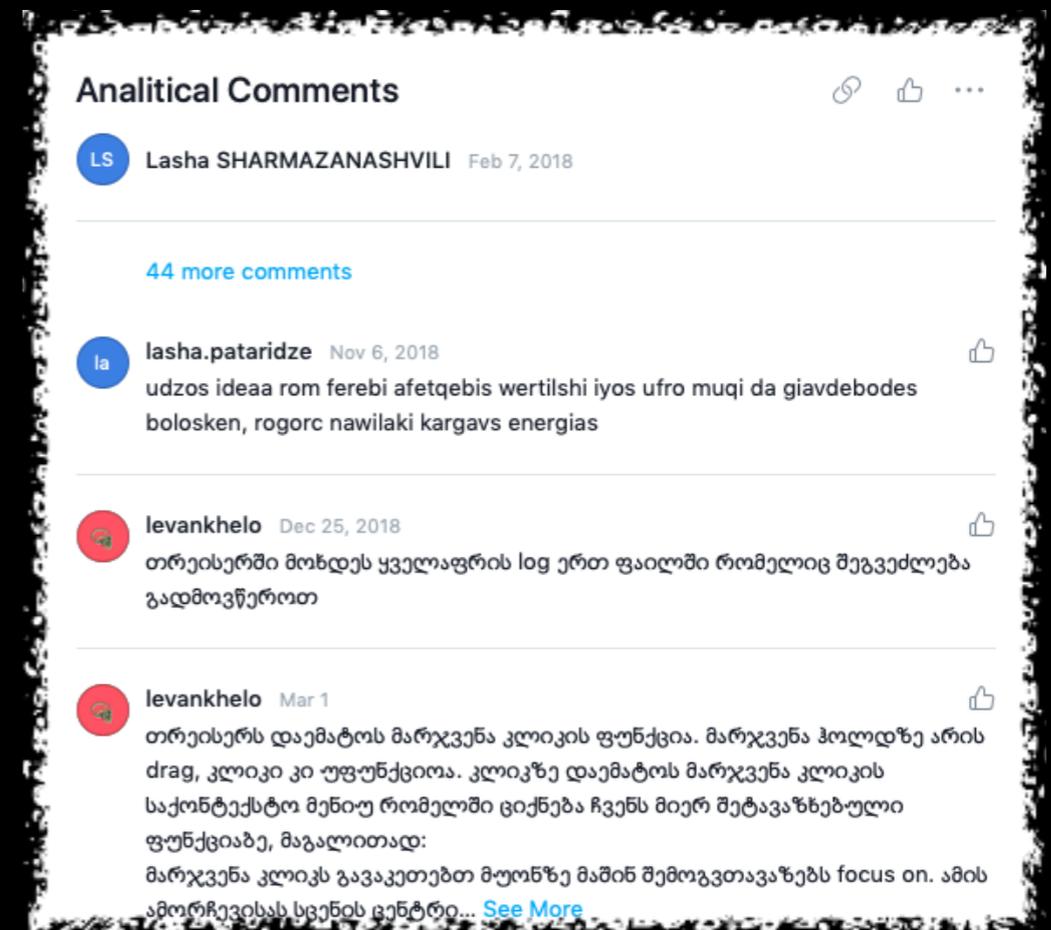


2 In future we are planning to add VR with Google Cardboard support. Fly mode gets us one step closer to this feature



47? We have lots of future plans and ideas about improvement of Engine. As you see by Asana screenshot, we currently have 47 new feature ideas and most of this will be implemented in future releases of **Tracer**

We are working hard to bring new features to **Tracer** and make it more interactive and functionally richer.



Thanks for your attention

