DeepSpace

Your way to space world

Russia

AN APPLICATION FOR THE ONES WHO WANT TO BE CLOSER TO UNKNOWN

DeepSpace is a mobile app that will allow people to communicate in their very own space community, follow blogs and topics, leave likes and comments on them or create their own articles.

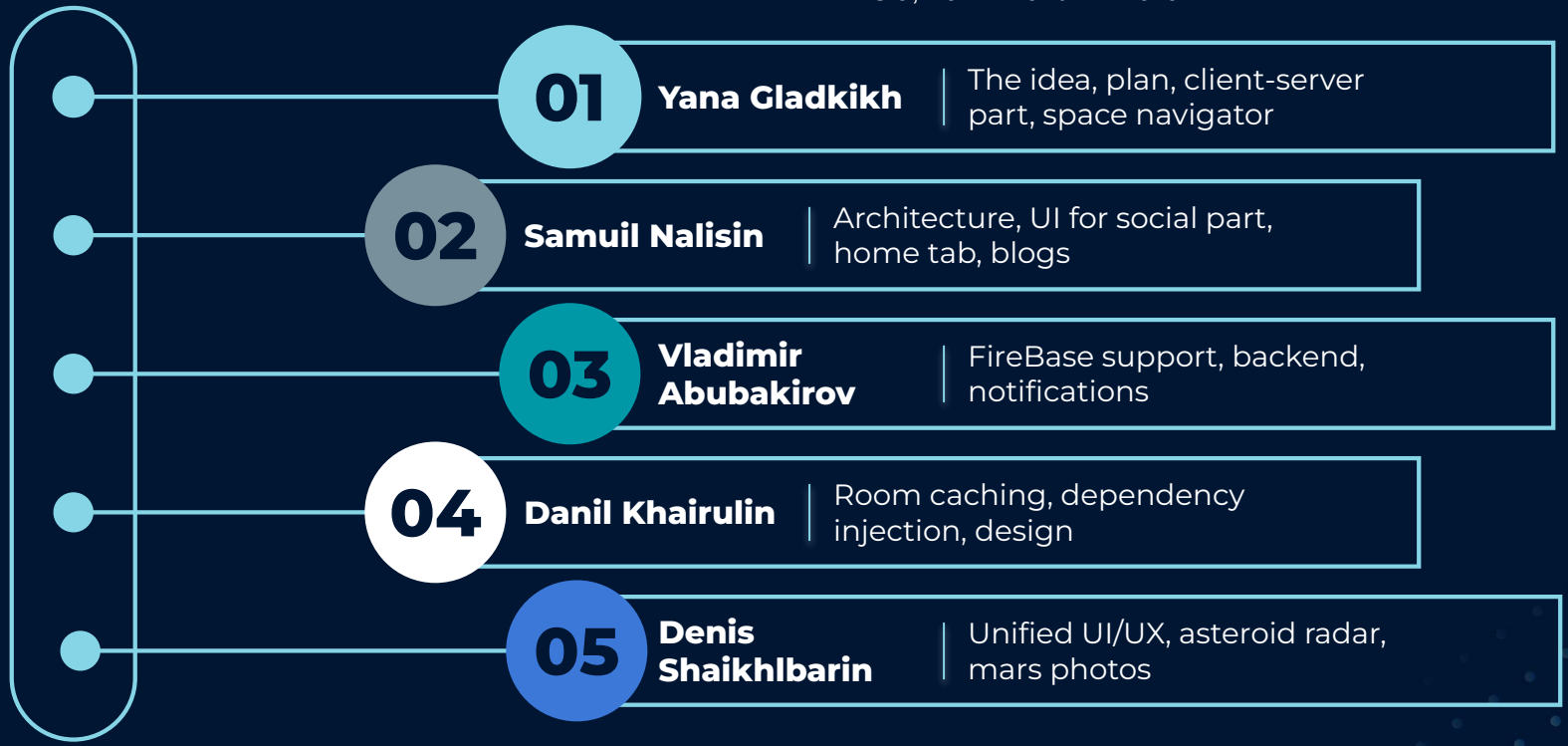
Application provides user features such as the asteroid radar wich allows to track potentially dangerous asteroids, watch photos from Mars, find out the location of planets in real time using the Space Navigator, setup notifications for upcoming SpaceX launches so as not to miss broadcasts and discussions of these events

For what purpose?

- Popularization of space and science
- An opportunity to discuss the latest events and communicate with like-minded people
- A handy tool for setting reminders about upcoming SpaceX launches
- The space navigator allows to point the phone in the right direction in order to find out which planets are located there right now
- Asteroid radar dispels all fake news about the treat approaching Earth
- Quick access to technical specifications of SpaceX ships and lots of interesting information and facts to read

OUR TEAM

We are the team of developers from all over the Russia. We are living in different cities across our country, but it doesn't prevent us from being friends and creating team projects.
Yana Gladkikh - Voronezh, Vladimir Abubakirov - Moscow, Samuil Nalisin - St. Petersburg, Denis Shaikhlbarin - Ufa, Danil Khairulin - Kazan



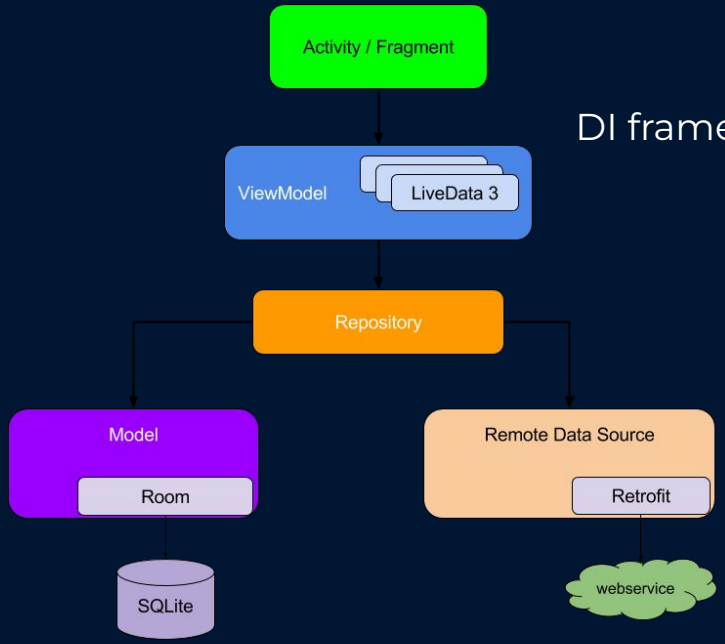
MAIN TOOLS

KOTLIN

Modern language that makes programmers happier

MVVM

Architecture pattern recommended by google



Hilt

DI framework specially for android

FireBase

Easy data storing

OUR TEAM

Yana Gladkikh

Retrofit
Maps Sdk
OpenGL
Sensors
Android Navigation

Samuil Nalisin

Compose
Hilt
RecyclerView
Android lifecycle
Retrofit

Vladimir Abubakirov

FireBase SharedPreferences
FireBase storage
FireBase auth
Sentry

Danil Khairulin

Room db
Hilt
Android navigation
Coroutines

Denis Shaikhlbarin

Material design
Material motion
Animations
Epoxy
Wallpapers
CustomViews
Glide

THE FEATURES

COMMUNITY

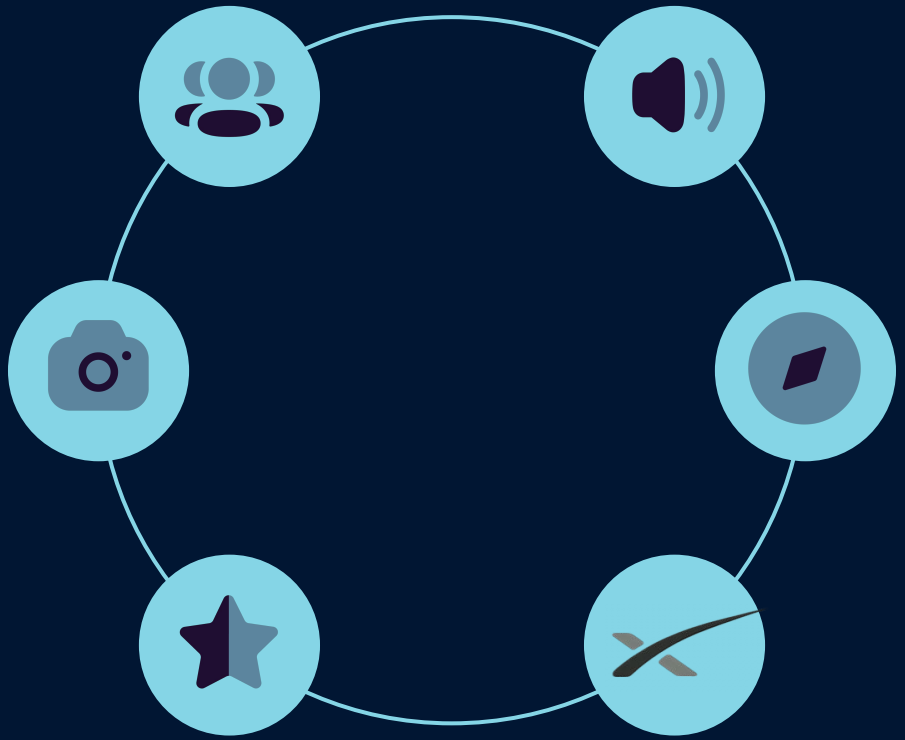
Discuss, share and find new friends

MARS

Take a look at photos taken by Mars Rovers

ASTEROIDS

Can you see some star shower today?



UPCOMING EVENTS

Don’t miss SpaceX launch live!

SPACE NAVIGATOR

Where are the planets now?

SPACEX

Learn about the most known private space company

Samuil Nalisin & Vladimir Abubakirov

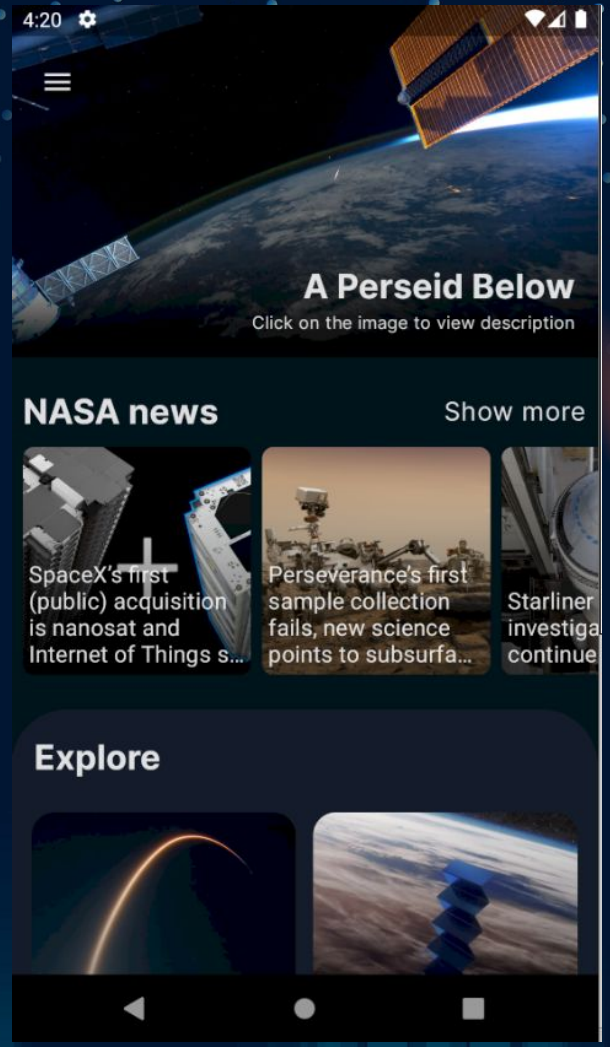
HOME PAGE

IMAGE OF THE DAY FROM NASA

Learn something new everyday

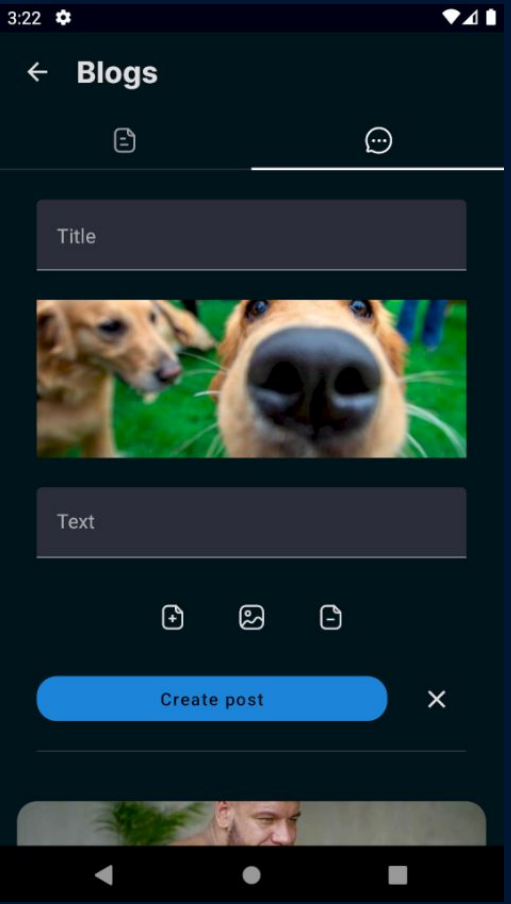
QUICK NAVIGATION AND LATEST NEWS

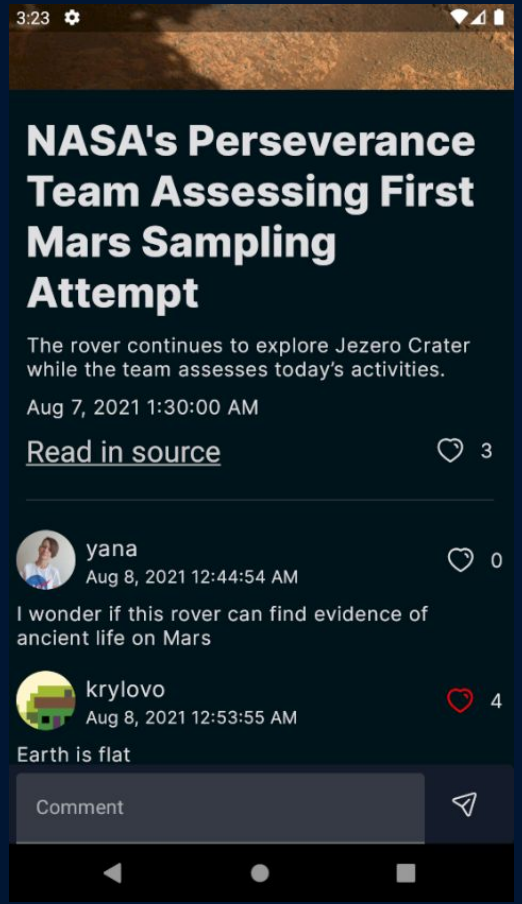
You can use homepage to switch on the other category or have a brief look on latest space news



Samuil Nalisin & Vladimir Abubakirov

BLOGS & NEWS





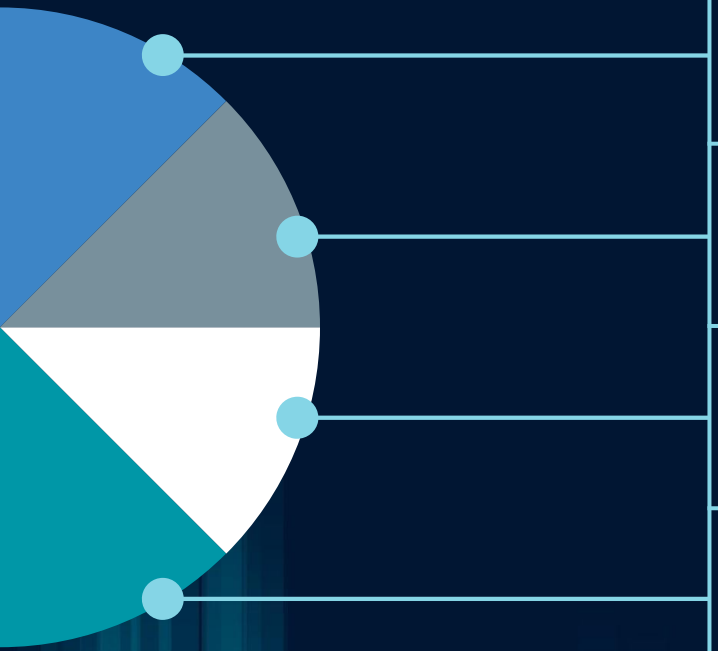
NEWS

Read and leave comments on the latest news

BLOGS

Share your own posts

HOMEPAGE, BLOGS & NEWS TECHNICAL SIDE

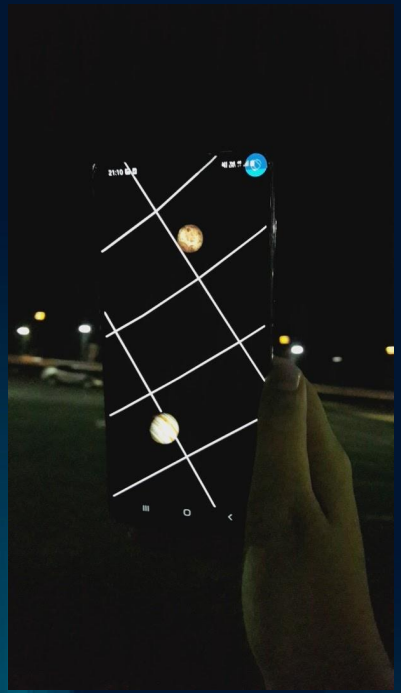


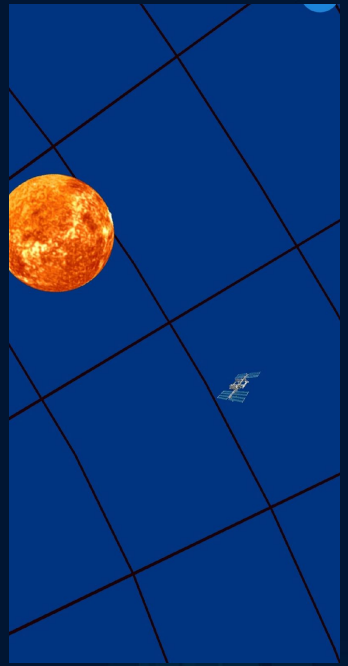
Compose	Only kotlin code, Declarative approach, Simple and optimized animations
FireBase storage	No sql, fast development, easy for integration
Realtime database	Quick and easy setup, flexible data(storage) format
Sentry	Detailed device information, convenient panel with error reports

Yana Gladkikh

SPACE NAVIGATOR

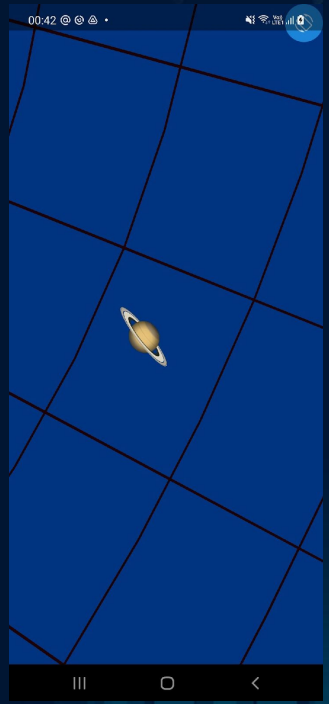
Allow to recognize your location





Point your phone into the right direction

Learn what are those small stars you see in the sky!

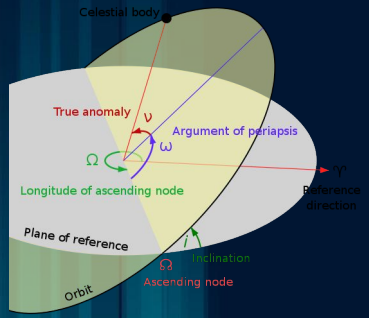


Yana Gladkikh

CALCULATING LOCATION OF PLANETS

In order to determine the location of the planets we've used Orbital Elements (Keplerian Elements)

a(AU) - axis
e – eccentricity
i – inclination
w – argument of periapsis (perihelion)
l – longitude of the ascending node
m – mean anomaly
v – true anomaly
E – eccentric anomaly



Data provided by NASA official site

	a AU, AU/Cy	e rad, rad/Cy	I deg, deg/Cy	L deg, deg/Cy	long.peri. deg, deg/Cy	long.node. deg, deg/Cy
Mercury	0.38709927	0.20563593	7.00497902	252.25032350	77.45779628	48.33076593
	0.00000037	0.00001906	-0.00594749	149472.67411175	0.16047689	-0.12534081
Venus	0.72333566	0.00677672	3.39467605	181.97909950	131.60246718	76.67984255
	0.00000390	-0.00004107	-0.00078890	58517.81538729	0.00268329	-0.27769418
EM Bary	1.00000261	0.01671123	-0.00001531	100.46457166	102.93768193	0.0
	0.00000562	-0.00004392	-0.01294668	35999.37244981	0.32327364	0.0
Mars	1.52371034	0.09339410	1.84969142	-4.55343205	-23.94362959	49.55953891
	0.00001847	0.00007882	-0.00813131	19140.30268499	0.44441088	-0.29257343
Jupiter	5.20288700	0.04838624	1.30439695	34.39644051	14.72847983	100.47390909
	-0.00011607	-0.00013253	-0.00183714	3034.74612775	0.21252668	0.20469106
Saturn	9.53667594	0.05386179	2.48599187	49.95424423	92.59887831	113.66242448
	-0.00125060	-0.00050991	0.00193609	1222.49362201	-0.41897216	-0.28867794
Uranus	19.18916464	0.04725744	0.77263783	313.23810451	170.95427630	74.01692503
	-0.00196176	-0.00004397	-0.00242939	428.48202785	0.40805281	0.04240589
Neptune	30.06992276	0.00859048	1.77004347	-55.12002969	44.96476227	131.78422574
	0.00026291	0.00005105	0.00035372	218.45945325	-0.32241464	-0.00508664
Pluto	39.48211675	0.24882730	17.14001206	238.92903833	224.06891629	110.30393684
	-0.00031596	0.00005170	0.00004818	145.20780515	-0.04062942	-0.01183482

Yana Gladkikh

CALCULATING LOCATION OF PLANETS

With the help of orbital elements, it is possible to calculate the heliocentric coordinates of the planets. Knowing these coordinates, you can find Ra and Dec (Right ascension and declination), and knowing them, you can finally calculate the geocentric coordinates and place the planets on the grid

First of all we need to calculate the true anomaly. According to NASA materials, it can be done by solving the Kepler equation $E - e \sin E = M$

So, by solving the equation, we will find an eccentric anomaly. To do this we need to decompose the series :

Solution of Kepler's Equation, $M = E - e^* \sin E$

Given the mean anomaly, M, and the eccentricity, e^*, both in degrees, start with

$$E_0 = M + e^* \sin M \qquad (8\text{-}36)$$

and iterate the following three equations, with $n = 0, 1, 2, \ldots$, until $|\Delta E| \leq tol$ (noting that e^* is in degrees; e is in radians):

$$\Delta M = M - (E_n - e^* \sin E_n) \;\; ; \;\; \Delta E = \Delta M / (1 - e \cos E_n) \;\; ; \;\; E_{n+1} = E_n + \Delta E. \qquad (8\text{-}37)$$

For the approximate formulae in this present context, $tol = 10^{-6} degrees$ is sufficient.

The true anomaly is expressed from the eccentric according to the formula:

$$\nu = 2 \arctan\left(\sqrt{\frac{1+e}{1-e}} \tan \frac{E}{2} \right)$$

Yana Gladkikh

CALCULATING LOCATION OF PLANETS

To calculate the heliocentric coordinates, it is still necessary to find the radius vector

Radius from true anomaly [edit]

The radius (distance between the focus of attraction and the orbiting body) is related to the true anomaly by the formula

$$r = a\,\frac{1 - e^2}{1 + e\cos\nu}$$

where *a* is the orbit's semi-major axis.

Great, now when all values have been found, we will calculate the coordinates according to the formula:

```
X = r * [cos(o) * cos(v + p - o) - sin(o) * sin(v + p - o) *
cos(i)]
Y = r * [sin(o) * cos(v + p - o) + cos(o) * sin(v + p - o) *
cos(i)]
Z = r * [sin(v + p - o) * sin(i)]

r is radius vector
v is true anomaly
o is longitude of ascending node
p is longitude of perihelion
i is inclination of plane of orbit
```

Finally, calculate the geocentric coordinates

If you have the geocentric right ascension α_M, declination δ_M, and distance d_M, then you can get the rectangular coordinates from:

$$X_M = d_M \cos(\alpha_M)\cos(\delta_M)$$

$$Y_M = d_M \sin(\alpha_M)\cos(\delta_M)$$

$$Z_M = d_M \sin(\delta_M)$$

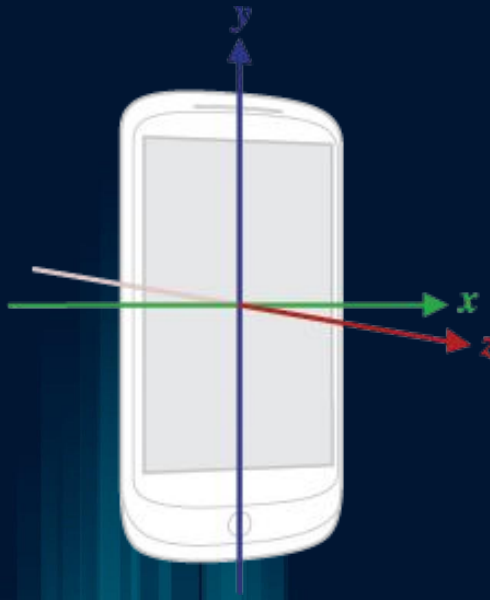
Next let's find Ra and Dec

Ra: $\operatorname{atan2}\dfrac{Y_h}{X_h}$ Dec: $\operatorname{atan2}\dfrac{Z_h}{\sqrt{X_h^2 + Y_h^2}}$

CALCULATING LOCATION OF PLANETS

Now we know the location of the planets and it remains only to place them on the grid.
The grid is created with 3D graphics using OpenGL. Two layers are used: lines and planets.

But it is not enough just to create a grid, it also must respond to the movement of the phone, move in accordance with where the user decided to direct it. We used phone sensors to make this into life



It's possible to talk about how all this works for a long time, so I'll try to tell you in general terms.
In three-dimensional space, the device has 3 axes. In the code, these axes are represented as vectors. With the help of sensors, you can get a value such as the ROTATION VECTOR, which returns the position of the device in space as an angle relative to the axis. Using this vector we can get the rotation matrix using the getRotationMatrixFromVector function of the SensorManager library class

To determine where the phone is directed, it is necessary to calculate the transformation matrix. We have a matrix of phone coordinates and a "magnetic" matrix with magnetic coordinates. Multiplying them, we get the desired matrix. Finally we need to calculate the direction vectors of the phone and in the activity we will transfer them to the thread responsible for rendering the map

Danil Khairulin

CACHING

Most modern applications use a local database to store or cache data received from a remote source. That’s why our project uses the Room library to work with sql queries

The advantages of caching are the increasing speed of data loading and the ability to load previously viewed content without a network connection

How caching works?

- When the screen is shown to user, the first step is to download data from the local database
- A request is sent to the server
- After the server's response, the data on the screen and the local database are being updated

DEPENDENCY INJECTION

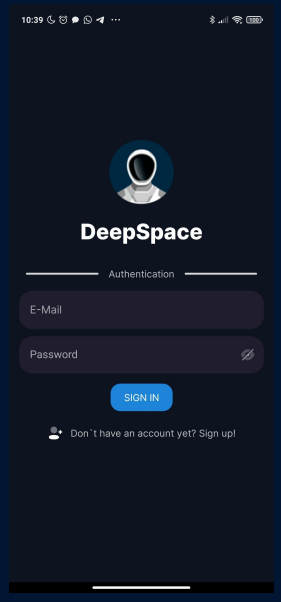
Dependency injection (DI) is a method widely used in programming and recommended by Google for creating android applications. By following the principles of DI, we make the application architecture cleaner and more readable for those who ever will join us or review our code.

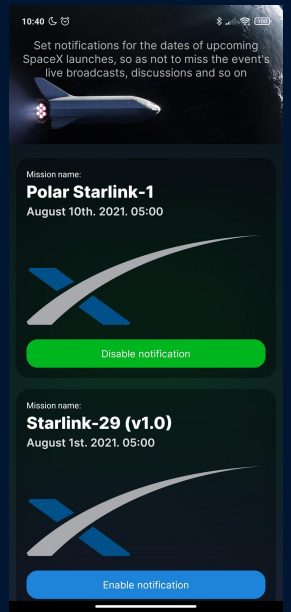
Implementing dependency injection provides the following pros:

- The ability to reuse the code
- Easy to refactor
- Flexible for testing

APPLICATION UI

In order to improve the user experience while using the application, it was necessary to create a uniform appearance. As a result, we've developed a semi-transparent interface design with the addition of colored shadows from objects on a dark background, as a result of which associations with interstellar clouds may appear. It allows you to immerse yourself into the application with greater interest. Particularly there is a single design for the process of "loading" pages.





Denis Shaikhlbarin

ASTEROID RADAR & MARS ROVERS

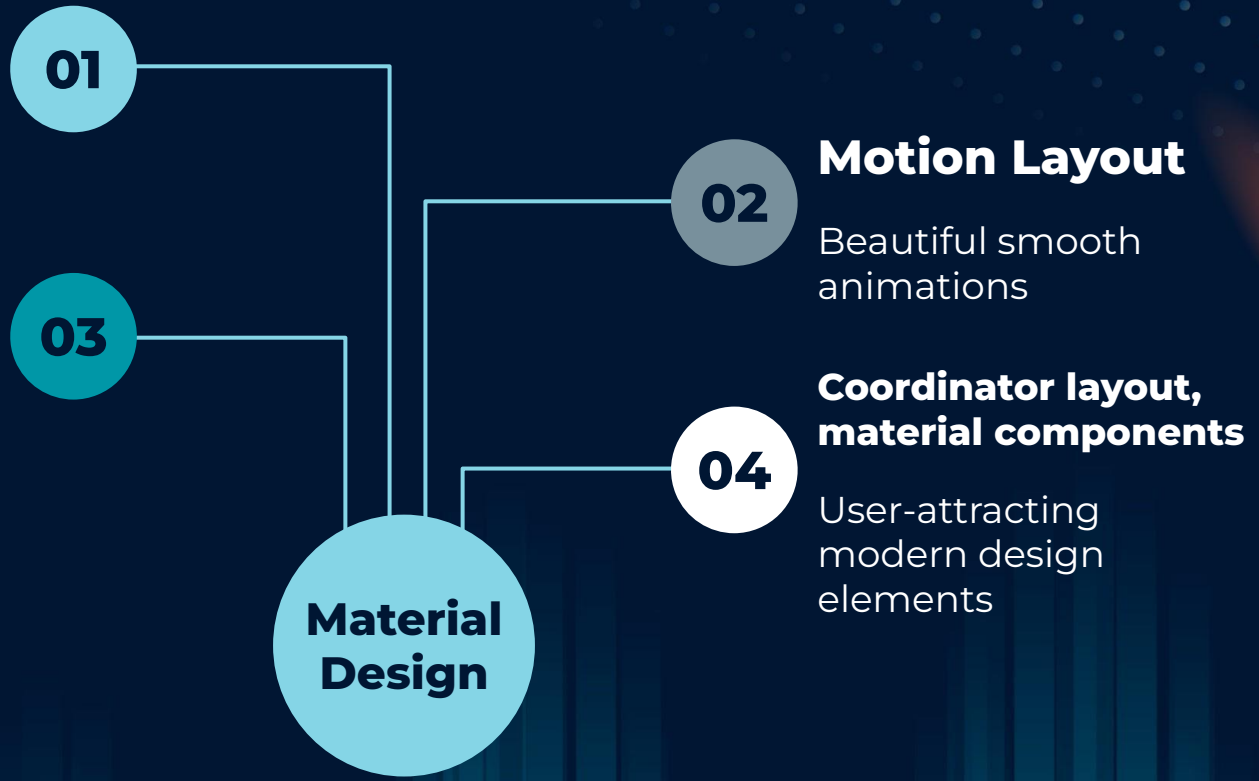
ASTEROID RADAR

Are there any asteroids near to Earth ?

MARS ROVERS

What would you see if you stood on a Mars?

UI TOOLS STACK



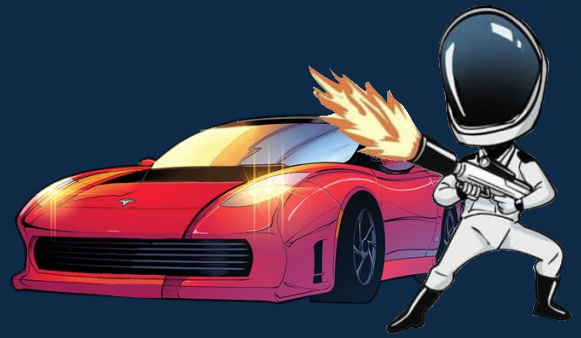
Conclusion

In conclusion, we would like to say that Deep Space is one of the best applications for people who are interested in space subject matter or for those who are just about to start diving into this huge community.
Its main purpose is to increase people's interest in space, give some basic knowledge about this sphere so that they may become the ones who create future of the humanity

Our team managed to create a stable working application. We've gone through thousands of bugs, solved plenty of merge conflicts. Also, we were able to discover something new by working in a team.

Development prospects:

- Adaptation to different languages
- More space objects on the grid such as stars
- Our own server and site (already in process)
- More technical information
- Information about NASA missions
- Notifications for other space events, not only SpaceX



You can check our open source code via the links:

https://github.com/BrightOS/DeepSpace and https://github.com/DeepSpaceTeam