Asteroid Visualization

Armend Imeri

Lokman Musliu

Yll Deari

Artan Emini

# Abstract

Asteroids are minor planets which travel in the cosmos. Their existence is very well known, but their threat is something which very few people know about. Awareness about them should definitely be risen. They are one of the very few objects in space which have the potential to kill all the living organisms in the world. In this report we describe the project we have built in regard to raising awareness about asteroids.

We start by showing the technologies chosen, and then describing the design of our web application. People will be able to go on this website and very easily be able to visually see all the threat opposing asteroids in a simulation. In the end we conclude on how it is possible to improve this project such as visually simulating the collision of an asteroid with Earth.

# Acknowledgements

It is thanks to our great team that we managed to get this project on the road. The different backgrounds of each of the team members helped develop all aspects of the application including: design, development, modelling, research, mathematics and documentation.

Of course none of this would be possible if Seeu Techpark didn’t provide their excellent services and facilities for us to work on. We were able to easily exchange information with other groups that was beneficial for all parties.

**Contents page**

[Abstract](#h.ul82mtwh841n)

[Acknowledgements](#h.t9niz9vwdlhj)

[Introduction](#h.xany6pwmbwyr)

[Background](#h.w0ybkx7krpks)

[Prototyping](#h.1mqbi1x898jv)

[System Design](#h.12xzoml3wey7)

[Development](#h.885ac0k8wwp1)

[Testing](#h.lvwpw8aanqiy)

[Conclusions and Future Work](#h.mf8bzfuurecj)

[Bibliography](#h.kthvpcwulhdh)

[Appendix](#h.mfxtw3usxne9)

[Low fidelity prototype](#h.iipd0vqew5s)

[High Fidelity Prototype](#h.icahk6tulroa)

# 

# Introduction

Our project is about visualizing asteroid data saved in NASA databases in a three dimensional world. All the movements would be real and you can track how each asteroid moves around our solar system. We want to get any average user to be able to get stuck in our web application exploring the different asteroids their movements, details etc. After looking at some of the other different projects on the space challenge website we decided that this would be the best challenge for our skills. Different info graphics provide plenty of user friendly information using drawings and illustrations.

# Background

There were many different approaches we could have used to get this project going but the space apps challenge limited us to technologies that do not require plugins so we thought that the most suitable would be ThreeJS. Of course the development would be more time consuming but since we all had past experience with the technologies we went with it instead of Unity. We also had a lot of different ways to access the data provided by NASA and because we wanted to have an independent system we downloaded their database which had information for about 600,000 asteroids. We categorized them in the ones that could one day be a potential hazard and used SQL queries to filter them out for our API which would then provide them to the web application.

Another alternative would be to use the NASA API and database and just use POST and GET methods to get different type of data depended on what we need. This can be easily changed but for now we went with the local database because of internet requirements during development which could not be guaranteed.

# Prototyping

After we did our research we started to prototype on the various ways we can implement and visualize this huge amount of data that we had on our disposal. We started from little sketches on paper and then proceed on the whiteboard to discuss broader topics. We did this in coordination with our team member who previously conducted the research, than we had to share all our knowledge to design the equations to resolve this task.

# System Design

The best way to design the system was a three layered solution which would include:

* SQL Database
* WEB API
* Web Application & 3D Models

The temporary database we used is Microsoft SQL Server 2012 in order to save all the asteroids. We then developed a WEB API using the .NET framework in order for the web application to make Ajax calls and get all the latest information directly from the database. In the future we can open our API for other people to use it with ready converted information for drawing as the team already did all the necessary mathematic calculations for this to happen.

The web application was build using html5, bootstrap, jquery, WebGL and the three.js library for 3D modelling.

# Development

We split the development in 3 pieces. The first would be the database and web API. The second would be the mathematics and trying to make sense of astronomical units into our small 3D realm. The third part would be to draw the geometrics, planets and asteroids. All of these levels are equally important to this project.

During development we all helped each other in different aspects and it would not have been possible to get the project on any level without helping each other.

Thanks to the attention we paid through school we were able to get all the necessary formulas using trigonometry to draw the asteroids in the most precise way possible.

First of all we converted all the AU into our units by multiplying them with 14,950. To a real reference point with the units used in the real world.

Second, all the asteroid orbits are a certain degree away from the reference point assigned in the Minor Planet Center databases. To move our objects in a three dimensional space the same way, we used the following formulas:

var degreesToRadians = orbitDegrees \* (Math.PI / 180);

asteroidMesh.position.x = Math.cos(degreesToRadians) \* Math.round(furthest \* 14950);

asteroidMesh.position.y = Math.sin(degreesToRadians) \* Math.round(furthest \* 14950);

First of all we converted the degrees of the rotation into radians because that is what WebGL uses after that positioned them accordingly by using the given functions Math.cos and Math.sin. Some rounding was done so that there wouldn’t bee to many decimal points when showing some results in the infographs.

To calculate the speed of the asteroid along it’s orbit we used the following formula:

var rrezja = furthestpoint \* 14950;

// \* 10000 to show it in Kilometers, not in the units we use

var asteroidSpeedPerHour = (2 \* rrezja \* Math.PI \* speed \* 10000) / 365 / 24;

We calculated the speed per year as the Minor Planet Center only tells us how many years an asteroid needs to circle it’s own orbit we then divided it by 365 and 24 to show a km/h speed wich would be more understandable for the everyday user.

“Rrezja” in this case is the radius of the asteroids orbit. And the speed is how many years it takes the asteroid to circle it’s own orbit.

After this we needed to tilt the asteroids orbit and movement to a certain angle defined in the database. As it was hard to tilt the asteroid movement we used an already defined object by ThreeJS called a Quaternion which would give an illusion of another 0 axis to the asteroid which was done as follows:

//Converting Incliniation of asteroid to our units

var angle = degrees \* Math.PI / 180;

//Quaternion to pivot asteroid movement depended on inclination

var quaternion1 = new THREE.Quaternion().setFromAxisAngle(axis, angle);

grouped.rotation.setEulerFromQuaternion(quaternion1);

Everything that we have mentioned was also done for the orbits but since they would be stationary we didn’t use quaternions the is code follows:

//Define Line

var line = (new THREE.Line(asteroidTrajectory, material));

var degreesToRadians = orbitDegrees \* (Math.PI / 180);

//Pivot and position line accordingly

line.position.x = Math.cos(degreesToRadians) \* (furthestpoint - nearestpoint);

line.position.y = Math.sin(degreesToRadians) \* (furthestpoint - nearestpoint);

line.rotation.y = angle;

# Testing

Testing the system was a very tiring development process because of the technologies which were used. Building the solar system in itself was time consuming because of lack in experience with using THREE.js library. We also had to test the queries which were provided by official NASA records. It was challenging because of huge size of their database as they had dumped 600,000 rows of 50 columns on a single SQL, our computers had problems coping with the memory and would run out and freeze. We made our own little software to read the database in parts and insert the queries in our database, we also converted them from MySQL to MS SQL by using some quick find and replace for the unneeded signs which saved us time to configure our API to work with a MySQL database.

# Conclusions and Future Work

All our research and work into asteroids brought us as team on a great journey to study this particular field and try to visualize it with our knowledge as computer science students.

We studied how asteroids move, their position, size, and everything we had on our disposal through NASA’s database of potential hazardous asteroids. All this information needed to be visualized on a website using open source libraries, and give the visitors more information on asteroids and their potential risk. We did this in a very short time that we didn’t expect as a team, and managed to visualize all the data that we had on asteroids and their potential impact.

As young and motivated students we have further development ideas that we would like to implement in the near future for example: giving more information on asteroids using easy to read data and try to simulate their impact, make a smartphone app so mobile users don’t have to use a computer to view our project, implement a wiki-style page for all the potential hazardous asteroids and more information on them. We would do this only for one purpose, to share more knowledge using the current technologies on this particular issue.

# Bibliography

Materials used in this project consist those of the Minor Planet Center. We downloaded their database and used our self-made tools to migrate it to our own MS SQL database.

<http://minorplanetcenter.net/>

# 

# Appendix

## Low fidelity prototype

After we did some brainstorming, drank a dozen cups of coffee and gathered the necessary information from the research we made, we came up with some mockups that would give an overview of how a possible solution might look like.

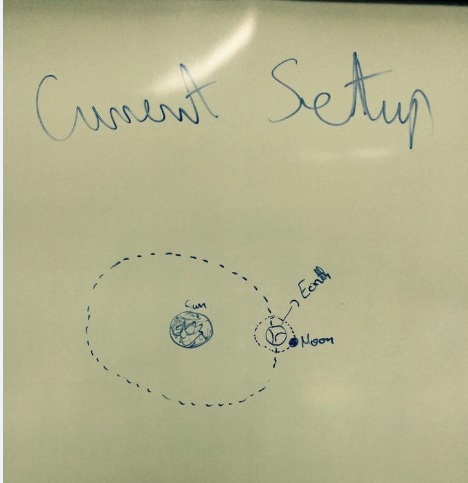


Figure 1.1 - Low fidelity Solar System Prototype

In figure 1.1 we drew a simple solar system with only the Earth as its spinning planet. We also defined the Moon with its own rotation around the Earth and started to implement the next items that were necessary.

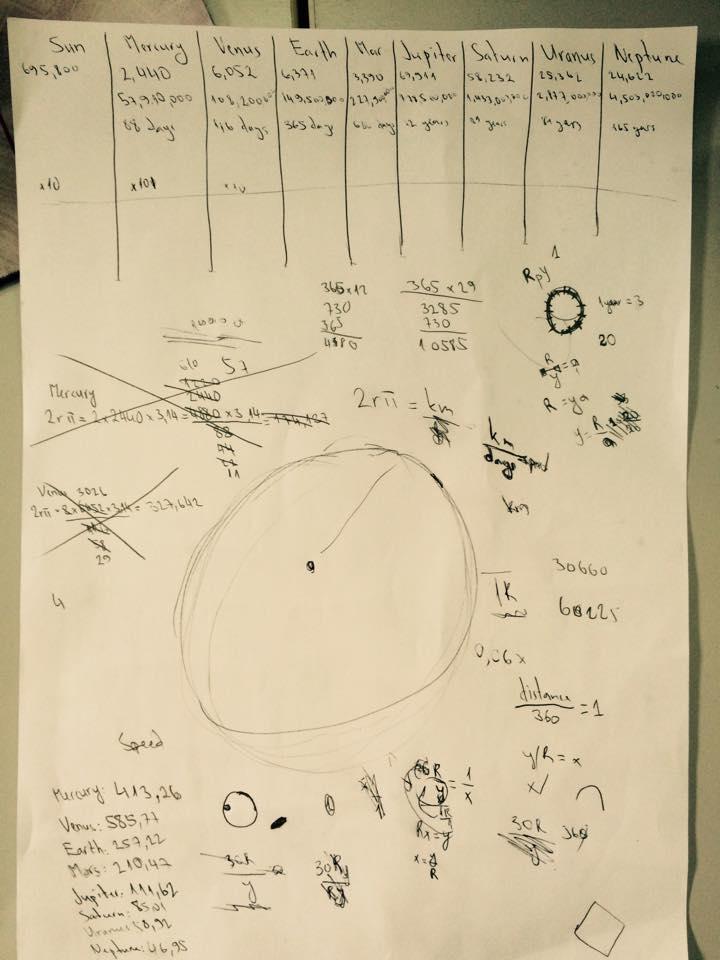


Figure 1.2 - Simple calculation of each planet’s coordinates

In figure 1.2 we gathered information of each planet, their position, distance according to the sun etc. Then we used some particular formulas to calculate the exact coordinates in order to get more precise results.

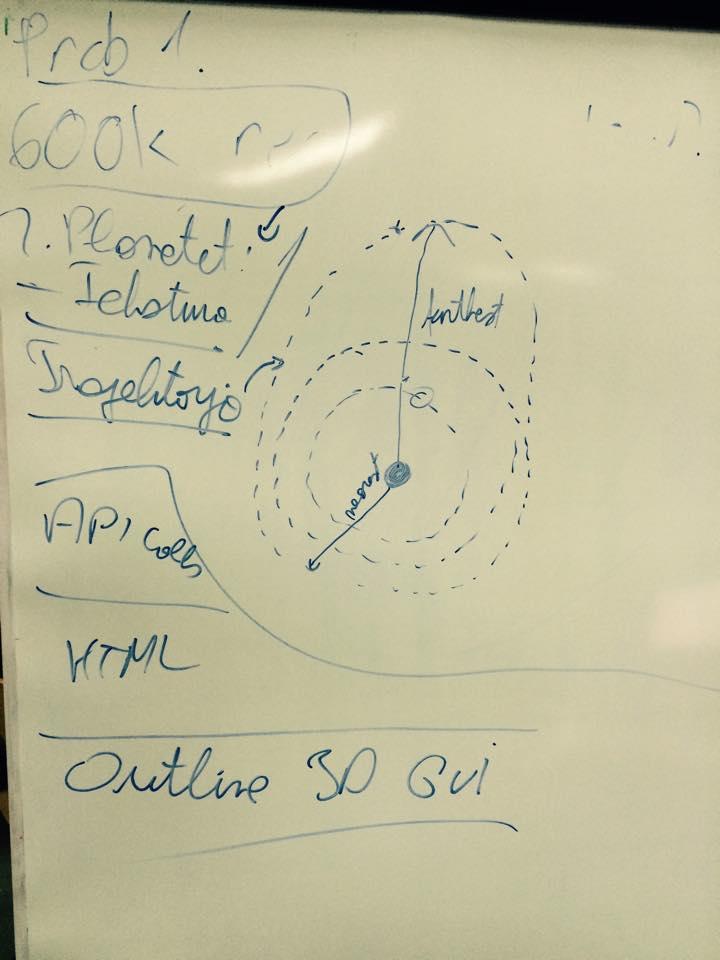


Figure 1.3. The solar system with some information about our system

Figure 1.3 show a similar result as figure 1.1, only in this part we also added the other planetary orbits.

## High Fidelity Prototype

In the high fidelity prototype, we designed the first prototype which contains the necessary elements and their functions along with the asteroid movements.

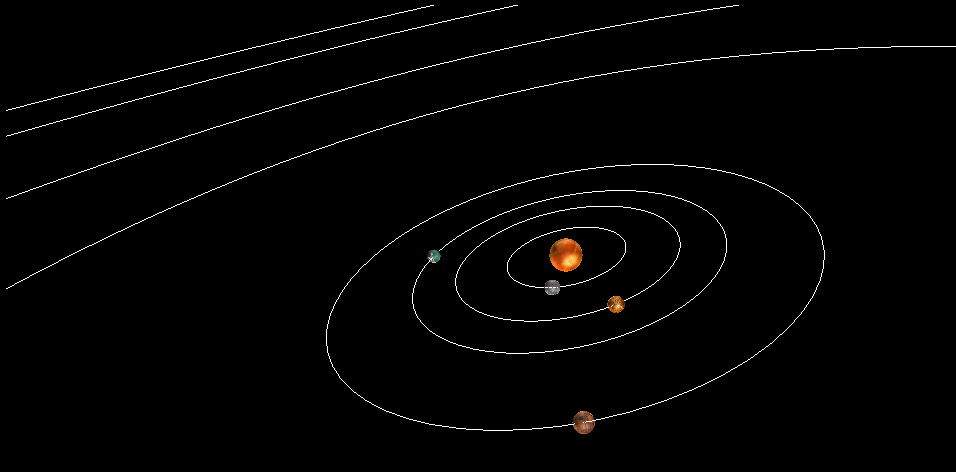


Figure 2.1- The final prototype of a solar system

In figure 2.1. We show the solar system along with the planets and asteroids.

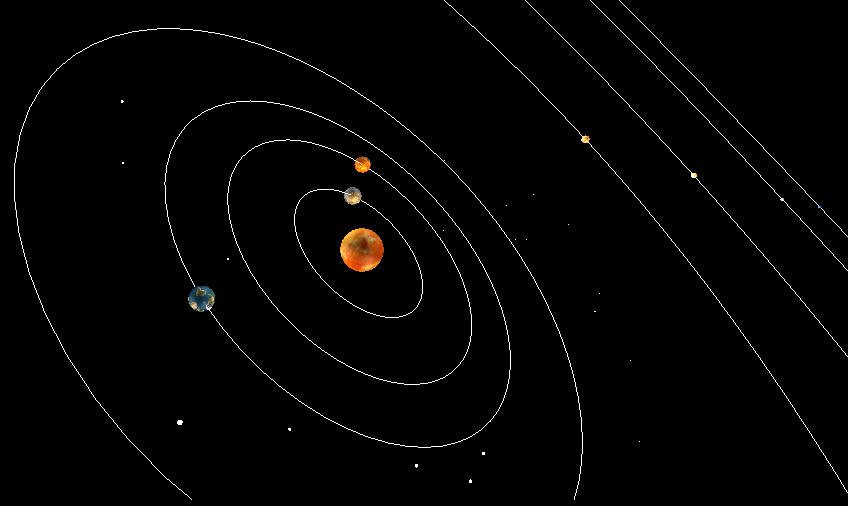


Figure 2.2 a complete overview of the entire solar system and asteroids

In figure 2.2 we described the entire project that shows the complete movement of the asteroids along with the other planets.