

Lucrare pentru atestarea competentelor profesionale
la informatică

Astronomie

COMAN MIRCEA GEORGE

Professor Coordonator: Mareş Carmen
COLEGIUL NAȚIONAL "PETRU RAREȘ" | MAI 2017

Cuprins

1. Adresa site-ului
2. Motivul alegerii temei
3. Limbajele de programare și de marcare folosite
 - 3.1. HTML
 - 3.2. CSS 3
 - 3.3. JavaScript
 - 3.4. Python
4. Bibliotecile folosite
 - 4.1. JQuery
 - 4.2. THREE.js
 - 4.3. ReportLabs
 - 4.4. pySVG
 - 4.5. MathJax
5. Cerințe hardware și software
6. Structura și conținutul proiectului
7. Codul sursă
8. Bibliografia

1. Adresa site-ului

<https://astronomytoolkit.ddns.net>

2. Motivul alegerii temei

Am ales ca temă pentru acest proiect **astronomia** având în vedere pasiunea pe care o am pentru această ramură a fizicii.

Aș vrea să ofer tuturor celor care vizitează site-ul meu posibilitatea de a explora universul cu ajutorul mijloacelor moderne de calcul.

De exemplu, simularea sistemului solar oferă o imagine asupra distanțelor relative dintre planete, dar și asupra numărului mare de asteroizi și de comete care orbitează Soarele; generatorul de cadrane solare îi ajută pe utilizatori să înțeleagă modul în care timpul era măsurat în Antichitate și în Evul Mediu și să își construiască singuri un astfel de instrument.

Harta cerului prezintă în timp real bolta cerească, așa cum este văzută de un observator aflat în orice locație de pe Pământ.

3. Limbajele de programare și de marcare folosite

Partea de front-end a paginii web a fost realizată în HTML, CSS și JavaScript, iar partea de back-end a fost realizată în Python.

3.1. HTML

HTML (HyperText Markup Language) este un limbaj de marcare utilizat pentru crearea paginilor web. HTML este o formă de marcare orientată către prezentarea documentelor text pe o singura pagină, utilizând un software de redare specializat, numit *agent utilizator HTML*, cel mai bun exemplu de astfel de software fiind *browserul web*. HTML furnizează mijloacele prin care conținutul unui document poate fi adnotat cu diverse tipuri de metadata și indicații de redare. Indicațiile de redare pot varia de la decorațiuni minore ale textului, cum ar fi specificarea faptului că un anumit cuvânt trebuie subliniat sau că o imagine trebuie introdusă, până la scripturi sofisticate, hărți de imagini și formulare. Metadatale pot include informații despre titlul și autorul documentului, informații structurale despre cum este împărțit documentul în diferite segmente, paragrafe, liste, titluri etc. și informații cruciale care permit ca documentul să poată fi legat de alte documente pentru a forma astfel hiperlink-uri.

3.2. CSS 3

CSS (Cascading Style Sheets) este un standard pentru formatarea elementelor unui document HTML. Stilurile se pot atașa elementelor HTML prin intermediul unor fișiere externe sau în cadrul documentului, prin elementul `<style>` și/sau atributul `style`. CSS se poate utiliza și pentru formatarea elementelor XHTML, XML și SVG.

CSS3 reprezintă un upgrade ce aduce câteva atribute noi și ajută la dezvoltarea noilor concepte în webdesign.

Unele dintre cele mai importante segmente (*module*) noi adăugate acestui standard pentru formatarea elementelor HTML aduc un plus considerabil în dezvoltarea activității webdesign.

Mai jos sunt prezente în listă cele mai importante modulele adăugate în CSS3:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

Deși au apărut unele deficiențe de compatibilitate între browsere, majoritatea proprietăților CSS3 au fost implementate cu succes în variantele browserelor noi.

3.3. JavaScript

JavaScript (JS) este un limbaj de programare orientat obiect bazat pe conceptul prototipurilor. Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul Javascript din aceste pagini fiind rulat de către browser. Limbajul este binecunoscut pentru folosirea sa în construirea siturilor web,

dar este folosit și pentru accesul la obiecte încastate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către Brendan Eich de la Netscape Communications Corporation sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.

În ciuda numelui și a unor similarități în sintaxă, între JavaScript și limbajul Java nu există nicio legătură. Ca și Java, JavaScript are o sintaxă apropiată de cea a limbajului C, dar are mai multe în comun cu limbajul Self decât cu Java.

Până la începutul lui 2005, ultima versiune existentă a fost JavaScript 1.5, care corespunde cu Ediția a 3-a a ECMA-262, ECMAScript, cu alte cuvinte, o ediție standardizată de JavaScript. Versiunile de Mozilla începând cu 1.8 Beta 1 au avut suport pentru E4X, care este o extensie a limbajului care are de a face cu XML, definit în standardul ECMA-357. Versiunea curentă de Mozilla, 1.8.1 (pe care sunt construite Firefox și Thunderbird versiunile 2.0) suportă JavaScript versiunea 1.7.

3.4. Python

Python este un limbaj de programare dinamic multi-paradigmă, creat în 1989 de programatorul olandez Guido van Rossum. Van Rossum este și în ziua de astăzi un lider al comunității de dezvoltatori de software care lucrează la perfecționarea limbajul Python și implementarea de bază a acestuia, CPython, scrisă în C. Python este un limbaj multifuncțional folosit de exemplu de

către companii ca Google sau Yahoo! pentru programarea aplicațiilor *web*, însă există și o serie de aplicații științifice sau de divertisment programate parțial sau în întregime în Python. Popularitatea în creștere, dar și puterea limbajului de programare Python au dus la adoptarea sa ca limbaj principal de dezvoltare de către programatori specializați și chiar și la predarea limbajului în unele medii universitare. Din aceleași motive, multe sisteme bazate pe Unix, inclusiv Linux, BSD și Mac OS X includ din start interpretatorul CPython.

Python pune accentul pe simplitatea codului, iar sintaxa sa le permite dezvoltatorilor să exprime unele idei programatice într-o manieră mai clară și mai concisă decât în alte limbaje de programare ca C. În ceea ce privește paradigma de programare, Python poate servi ca limbaj pentru software de tipul *object-oriented*, dar permite și programarea imperativă, funcțională sau procedurală. Sistemul de tipizare este dinamic iar administrarea memoriei decurge automat prin intermediul unui serviciu „gunoier” (*garbage collector*). Alt avantaj al limbajului este existența unei ample biblioteci standard de metode.

Implementarea de referință a Python este scrisă în C și poartă deci numele de *CPython*. Această implementare este software liber și este administrată de fundația *Python Software Foundation*.

4. Bibliotecile folosite

Pentru acest proiect am folosit JQuery, THREE.js, MathJax, pySVG și ReportLabs.

4.1. JQuery

JQuery este o platformă de dezvoltare JavaScript, concepută pentru a ușura și îmbunătăți procese precum traversarea arborelui DOM în HTML, managementul inter-browser al evenimentelor, animații și cereri tip AJAX. JQuery a fost gândit să fie cât mai mic posibil, disponibil în toate versiunile de browsere importante existente, și să respecte filosofia "Unobtrusive JavaScript". Biblioteca a fost lansată în 2006 de către John Resig. JQuery este utilizată de aplicațiile "Solar System" și "Sky Map" pentru a lua date asincron de pe server.

4.2. THREE.js

THREE.js este o bibliotecă utilizată de aplicațiile "Solar System" și "N-Body Simulator" pentru grafică accelerată de placa video.

4.3. MathJax

MathJax ajută la scrierea ecuațiilor matematice.

4.4. pySVG

pySVG este o bibliotecă pentru Python care generează grafică vectorială sub forma fișierelor de tip .SVG.

4.5. ReportLab

ReportLab este o bibliotecă pentru Python care crează documente de tip .PDF.

5. Cerințe hardware și software

5.1. Cerințe hardware

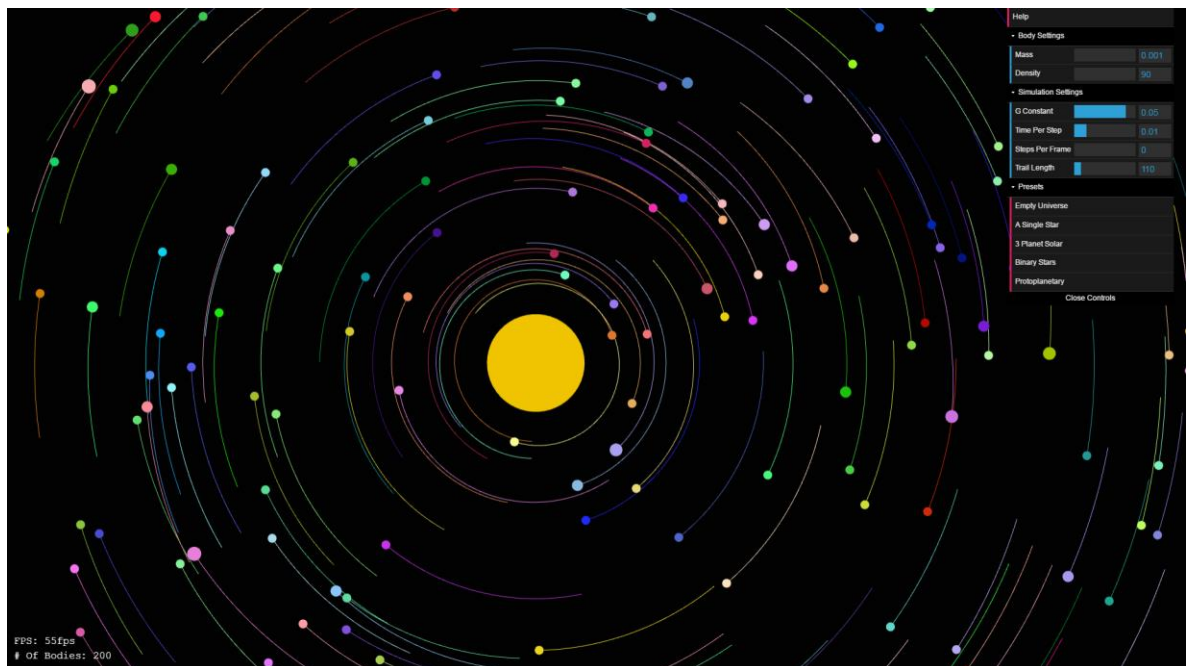
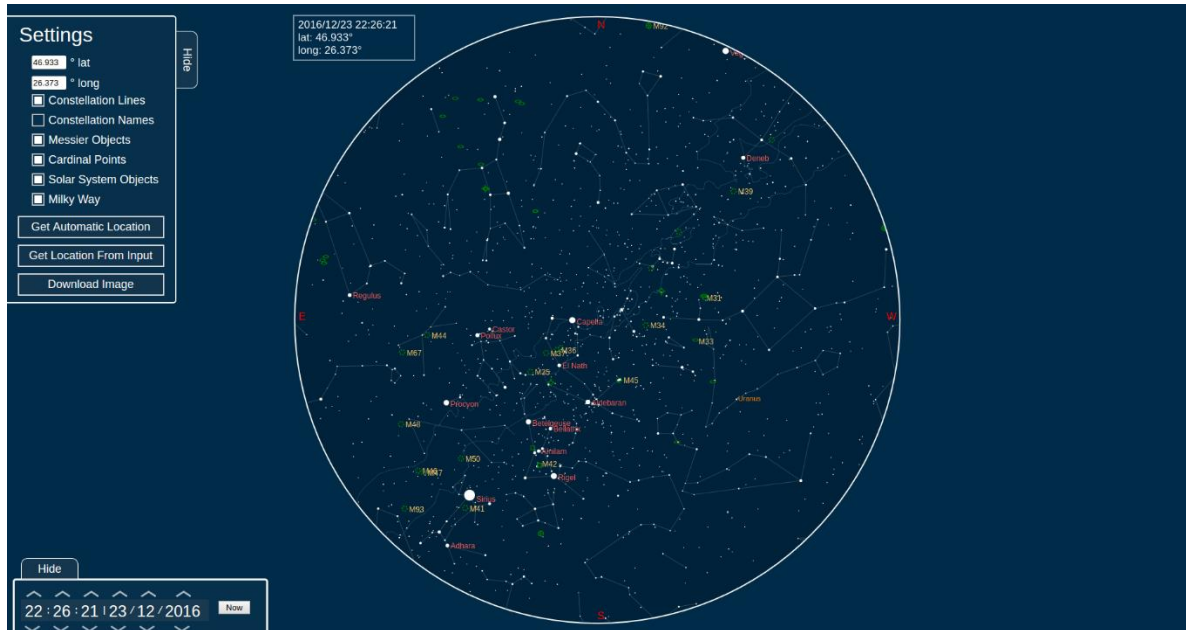
- Placă video compatibilă cu THREE.js
- Memorie RAM: 500MB
-

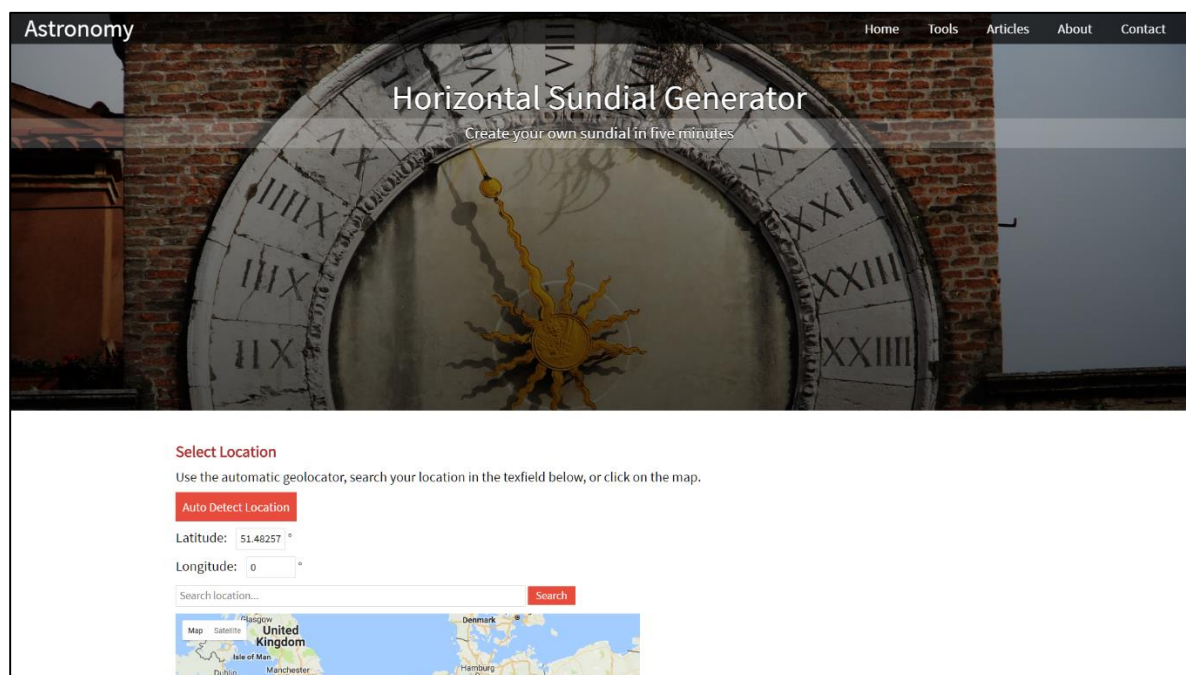
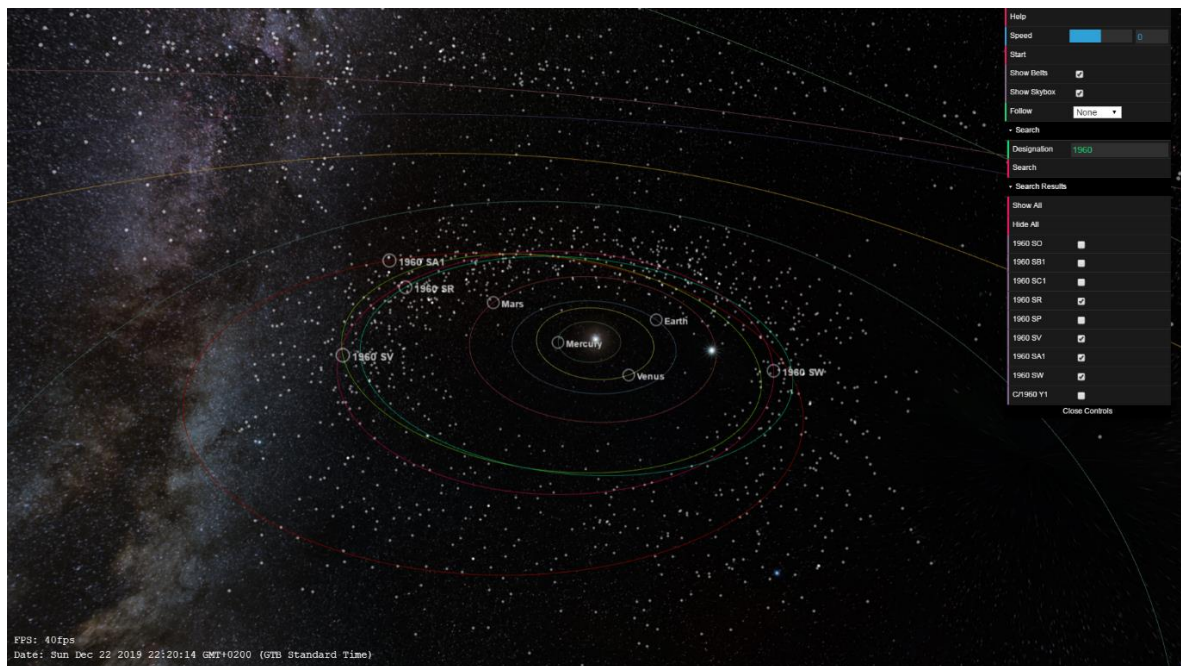
5.2. Cerințe software

- Drivere pentru placa video
- Browser (de preferat Chrome sau Firefox)

6. Structura și conținutul proiectului

Proiectul este format dintr-o pagină principală, 4 pagini interactive (“Sky Chart”, “Solar System”, “N-Body Simulation” și “Sundial”) și câteva articole. Mai jos sunt câteva screenshot-uri de pe site:





Astronomy
Home
Tools
Articles
About
Contact

Skychart

How I Programmed the Skychart

A sky map or a star chart shows the position of the stars, of the Messier objects and of the planets as they are seen by an observer from a location on Earth on a given time. You can use my sky map generator [here](#)

Algorithms and formulas used

The positions of the stars in equatorial coordinates were taken from the [Yale Bright Star Catalog](#). The equatorial coordinates are converted in horizontal coordinates using the formulas:

$$\tan A = \frac{\sin H}{\cos H \sin \Phi - \tan \delta \cos \Phi}$$

$$\sin a = \sin \delta \sin \Phi + \cos \delta \cos \Phi \cos H$$

where A is the azimuth, a the altitude, H the hour angle, δ the declination and Φ the observer's latitude.

To project the stars on the computer screen I used the stereographic projection. The polar coordinates of a point from the celestial sphere projected

Astronomy
Home
Tools
Articles
About
Contact

Rosetta Spacecraft

Comet Orbiter

Rosetta was a space probe built by the European Space Agency launched on 2 March 2004. Along with Philae, its lander module, Rosetta performed a detailed study of comet 67P/Churyumov–Gerasimenko (67P). During its journey to the comet, the spacecraft flew by Mars and the asteroids 21 Lutetia and 2867 Steins. It was launched as the third cornerstone mission of the ESA's Horizon 2000 programme, after SOHO / Cluster and XMM-Newton. On 6 August 2014, the spacecraft reached the comet and performed a series of manoeuvres to eventually orbit the comet at distances of 30 to 10 kilometres (19 to 6 mi). On 12 November, its lander module Philae performed the first successful landing on a comet, though its battery power ran out two days later. Communications with Philae were briefly restored in June and July 2015, but due to diminishing solar power, Rosetta's communications module with the lander was turned off on 27 July 2016. On 30 September 2016, the Rosetta spacecraft ended its mission by hard-landing on the comet in its Ma'at region. The probe is named after the Rosetta Stone, a stele of Egyptian origin featuring a decree in three scripts. The lander is named after the Philae obelisk, which bears a bilingual Greek and Egyptian hieroglyphic inscription.

Design and construction

The Rosetta bus is a 2.8 × 2.1 × 2.0 m (9.2 × 6.9 × 6.6 ft) central frame and aluminium honeycomb platform. Its total mass is approximately 3,000 kg (6,600 lb), which includes the 100 kg (220 lb) Philae lander and 165 kg (364 lb) of science instruments. The Payload Support Module is mounted on top of the spacecraft and houses the scientific instruments, while the Bus Support Module is on the bottom and contains spacecraft support subsystems.

7. Codul Sursă

Fișierul CSS pentru pagina principală:

```
@import url('https://fonts.googleapis.com/css?family=Source+Sans+Pro');

* {
    margin: 0px;
    padding: 0px;
}

body{
    background-color: #ecf0f1;
    color: #FFF;
    margin: 0px;
    font-family: 'Source Sans Pro', sans-serif;
    min-height: 100%;
}

@keyframes animatedBackground {
    from { background-position: 0 0; }
    to { background-position: 6121px 0; }
}

#container{
    min-height: 100%;
    position: relative;
}

#menu{
    position: absolute !important;
    height: 150px !important;
    background-color: rgba(0, 0, 0, 0) !important;
}

#menu h1{
    text-align: center !important;
    left: 0px !important;
    top: -15px;
    font-size: 60px !important;
    text-shadow:
        -1px -1px 0 #000,
        1px -1px 0 #000,
        -1px 1px 0 #000,
        1px 1px 0 #000 !important;
    font-weight: 900 !important;
}

.dropdown_content{
    position: absolute !important;
    /*top: 152px !important;*/
}
```



```

#links{
    position: absolute !important;
    display: table !important;
    left: 0px !important;
    top: 106px !important;
    width: 100% !important;
    margin-left: 0px !important;
    padding-top: 0px !important;
    padding-bottom: 0px !important;
    background-color: rgba(0, 0, 0, 0.4) !important;
}

/*#bottom_bar{
    top: 300px !important;
}*/

#title{
    position: relative;
    display: block;
    top: 0px;
    width: 100%;
    margin: 0px;
    height: 150px;
    border-style: solid;
    text-align: center;
    border-color: #212121;
    border-width: 0px 0px 2px 0px;
}

#banner{
    background-color: #000;
    background-image: url("../res/starry_night.jpg");

/* background-size: contain;*/
    background-position: 0px 0px;
    background-attachment: fixed;
    animation: animatedBackground 90s linear infinite;

/*background-position: left;*/
    background-repeat: repeat-x;
    background-size: auto 220px;

    z-index: 1;
    position: relative;
    height: 100%;
    width: 100%;
/*
    -webkit-filter: blur(1px);
    -moz-filter: blur(1px);
    -o-filter: blur(1px);
    -ms-filter: blur(1px);
    filter: blur(1px);
*/
}

```

```
#gradient{
  z-index: 2;
  position: relative;
  top: -100%;
  width: 100%;
  height: 100%;
  background: -webkit-linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.6));
  background: -o-linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.6));
  background: -moz-linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.6));
  background: linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.6));
```

```
}
```

```
#title h1{
  z-index: 99;
  display: inline-block;
  position: relative;
  vertical-align: middle;
  text-align: center;
  font-size: 60px;
  text-shadow:
    -1px -1px 0 #000,
    1px -1px 0 #000,
    -1px 1px 0 #000,
    1px 1px 0 #000;
  width: 100%;
  height: 100%;
  top: -4.5em;
```

```
}
```

```
#title a{
  display: inline;
  vertical-align: middle;
  position: relative;
  margin-right: 17px;
  font-size: 25px;
  color: #fff;
  padding: 5px;
  float: right;
  text-decoration: none;
  right: 50px;
  top: 1em;
```

```
-webkit-transition: 90ms;
-moz-transition: 90ms;
-o-transition: 90ms;
transition: 90ms;
```

```
}
```

```
#buttonBar{
  z-index: 99999;
  position: relative;
  width: 100%;
```

```

        height: 50px;
        margin-top: -50px;
        top: -100%;
    }

    #buttonBar a{
        display: inline-block;
        float: right;
        position: relative;
        top: 5px;
        right: -3px;
        margin-right: 5px;
    }

    #buttonBar img{
        width: 35px;
    }

    #buttonBar img:hover{
        filter: brightness(80%);
    }

    #pages_container{
        display: table;
        position: relative;
        left: 15% ;
        top: 30px;
        width: 70%;
        padding-bottom: 50px;
    }

    .page{
        display: table-row;
        float: left;
        position: relative;
        top: 0px;
        left: 0px;

        width: calc(21.7vw - 10px); /*container_width * 31% - border_width * 2 * elements/line*/
        height: calc(21.7vw - 10px);
        margin-left: 1%;
        margin-top: 20px;
        padding: 0px;

        text-align: center;
        border-radius: 50%;
        border: 5px solid gray;
        overflow: hidden;

        -webkit-filter: grayscale(100%);
        -moz-filter: grayscale(100%);
        -o-filter: grayscale(100%);
        filter: grayscale(100%);

        -webkit-transition: 200ms;

```



```

-moz-transition: 200ms;
-o-transition: 200ms;
transition: 200ms;
background-size: cover !important;
}

.page h3{

    vertical-align: middle;
    z-index: 0;
    position: relative;
    text-align: center;
    vertical-align: middle;
    top: 100%;
    left: 0%;

    padding-top: 2.5%;
    width: 100%;
    height: 50%;
    color: white;
    vertical-align: middle;

    -webkit-transition: 200ms;
    -moz-transition: 200ms;
    -o-transition: 200ms;
    transition: 200ms;

    font-weight: 500;
    font-size: 23px;
}

/*Animations*/
.page:hover h3{
    background-color: rgba(33, 33, 33, 0.7);
    transform: translate(0,-58%);
}

.page:hover{
    border-color: #212121;
    -webkit-filter: grayscale(0%);
    -moz-filter: grayscale(0%);
    -o-filter: grayscale(0%);
    filter: grayscale(0%);
}

/*Media Queries*/

@media screen and (max-width: 1279px) {
    .page{
        width: calc(33.6vw - 10px);
        height: calc(33.6vw - 10px);
    }
}

```

Fișierul CSS pentru o pagină de tip articol:

```
@import url('https://fonts.googleapis.com/css?family=Source+Sans+Pro');
```

```
*:focus {  
    outline: none;  
}
```

```
*{  
    margin: 0px;  
    padding: 0px;  
    font-family: 'Source Sans Pro', sans-serif;  
}
```

```
body{  
    background-color: #FFF;  
    min-height: 100%;  
    min-width: 1140px;  
}
```

```
#main_container{  
    min-height: 100%;  
    position: relative;  
}
```

```
#menu{  
    min-width: 1140px !important;  
}
```

```
#title{  
    position: relative;  
    top: 0px;  
    color: #FFF;  
    width: 100%;  
    height: 90vh;  
    text-shadow:  
        -0.5px -0.5px 0 #000,  
        0.5px -0.5px 0 #000,  
        -0.5px 0.5px 0 #000,  
        0.5px 0.5px 0 #000;  
}
```

```
#banner{  
    position: relative;  
    background-position: center;  
    background-size: cover;  
    width: 100%;  
    height: 100%;  
    background-attachment: fixed;  
}
```

```
#gradient{
  z-index: 2;
  position: relative;
  top: -100%;
  width: 100%;
  height: 100%;
  background: -webkit-linear-gradient(rgba(0, 0, 0, 0.2), rgba(0, 0, 0, 0.7));
/* background: -o-linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.8));
  background: -moz-linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.8));
  background: linear-gradient(rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.8));*/
}
```

```
#title h1{
  position: relative;
  top: -140vh;
  z-index: 3;
  text-align: center;
  vertical-align: middle;
  font-size: 50px;
  font-weight: 500;
}
```

```
#title h2{
  z-index: 3;
  position: relative;
  background-color: rgba(255, 255, 255, 0.2);
  width: 100%;
  top: -140vh;
  font-weight: 500;
  text-align: center;
  vertical-align: middle;
  line-height: 100%;
  padding: 10px 0 10px 0;
}
```

```
#content{
  position: relative;
  top: 0px;
  left: 14%;
  width: 72%;
  padding-top: 30px;
  /*border-style: solid;
  border-color: #9d9d9d;
  border-width: 0 1px 0 1px;
  background-color: #fff;
  padding-bottom: 30px;
}
```

```
#content h5{
  color: rgb(171, 50, 50);
  font-weight: 600;
  font-size: 23px;
```

```

margin-bottom: 10px;
margin-left: 0px;
}

#content h4{
color: rgb(171, 50, 50);
font-weight: 600;
font-size: 26px;
margin-bottom: 10px;
margin-left: 0px;
}

#content h3{
color: rgb(171, 50, 50);
font-weight: 600;
font-size: 29px;
margin-bottom: 10px;
margin-left: 0px;
}

#content h2{
color: rgb(171, 50, 50);
font-weight: 900;
font-size: 32px;
margin-bottom: 0px;
/*margin-left: 35px;*/
}

#content h1{
color: rgb(171, 50, 50);
font-weight: bold;
font-size: 35px;
margin-bottom: 10px;
margin-left: 0px;
}

#content p, li, label, a{
/*text-indent: 50px;*/
text-align: left;
margin-bottom: 10px;
font-size: 20px;
font-weight: normal;
color: black;
}

#content ol{
padding-left: 0 !important;
list-style: inside decimal;
}

#content ul{
padding-left: 0 !important;
list-style: inside disc;
}

```

```

}

input{
  margin-bottom: 10px;
  padding: 5px;
  font-size: 16px;
}

input[type="text"], input[type="number"]{
  border-radius: 1px;
  border: solid 1px #dcdcdc;
  transition: box-shadow 0.3s, border 0.3s;
}

input[type="text"]:focus,
input[type="number"]:focus {
  border: solid 1px #707070;
  box-shadow: 0px 0px 1px 1px #969696;
}

input[type="button"]{
  border-style: none;
  border-width: 1px;
  border-color: #212121;
  padding: 10px;
  background-color: #e74c3c;
  color: #FFF;
  font-size: 18px;
  -webkit-transition: background-color 100ms ease-out 5ms;
}

input[type="button"]:hover, input[type="button"]:active{
  background-color: #CF3A24 !important;
}

label{
  margin-right: 10px;
}

.form{
  margin-left: 0px;
  font-size: 18px;
  font-weight: 300;
}

#bottom_bar{
  display: table;
  position: relative;
  background-color: #212121;
  color: #FFF;
  width: 100%;
  text-align: center;
  height: 1.5em;
}

```

```

}

#bottom_bar p {
    display: table-cell;
    vertical-align: middle;
}

@media only screen and (max-device-width: 480px) {
    body{
        width: 100%;
    }
    #banner{
        height: 60vh;
    }
    #content{
        top: -30vh;
        left: 5%;
        width: 90%;
    }
    #title h1{
        top: -130vh;
    }
    #title h2{
        top: -130vh;
    }
}
}

```

Structura HTML a unui articol:

```

<!DOCTYPE html>
<html>
<head>
    <title>About</title>
    <link rel = "stylesheet" type = "text/css" href = "../basic_html/basic_style.css">
    <script type="text/javascript" src="../lib/jquery-2.1.4.js"></script>

</head>

<body>
<div id = "main_container">
    <?php include("../basic_html/topbar.html") ?>

    <div id = "title">
        <div id = "banner" style = "background-image: url(pic.jpg);">

        </div>
        <div id = "gradient">

        </div>
        <h1>About</h1>
        <h2>Information about the site</h2>
    </div>

```

```

<div id = "content">

    </div>
    <?php include("../basic_html/bottom_bar.html"); ?>
</div>
</body>
</html>

```

Structura HTML a barei de titlu:

```

<style>
#menu{
    min-width: 100%;
    z-index: 4;
    position: fixed;
    display: table;
    width: 100% !important;
    /*height: 50px;*/
    background-color: rgba(0, 0, 0, 0.7);
    top: 0px;
}

#menu h1{
    position: relative;
    color: white;
    display: table-cell;
    vertical-align: middle;
    left: 20px;
    font-size: 35px;
    font-weight: normal;
}

#links{
    position: relative;
    display: table-cell;
    vertical-align: middle;
    margin-left: 200px;
}

#menu h5, .dropdown_link{
    display: inline-block;
    padding: 10px;
    position: relative;
    margin-right: 22px;
    float: right;
    font-weight: normal;
    color: #fff;
    font-size: 20px;
}

```

```

vertical-align: middle;
-webkit-transition: background-color 200ms ease-out 1ms;
-moz-transition: background-color 200ms ease-out 1ms;
-o-transition: background-color 200ms ease-out 1ms;
transition: background-color 200ms ease-out 1ms;
}

```

```

#menu h5:hover{
background-color: rgba(205, 205, 205, 0.2);
}

```

```

.dropdown{
float: right;
}

```

```

.dropdown_content{
position: absolute;
top: 45px;
display: none;
background-color: rgba(0, 0, 0, 0.8);
float: left;
width: 240px;
overflow: hidden;
}

```

```

.dropdown:hover .drop_link h5{
background-color: rgba(205, 205, 205, 0.15);
}

```

```

.dropdown_content a, .dropdown_content h5{
width: 100%;
display: block;
float: left !important;
}
.dropdown:hover .dropdown_content {
display: block;
}

```

</style>

```

<div id = "menu">
<h1>Astronomy Tools</h1>
<div id = "links">
<a href = "../contact"><h5>Contact</h5></a>
<a href = "../about"><h5>About</h5></a>

<div class = "dropdown">
<a class = "drop_link" href = "#"><h5>Articles</h5></a>
<div class = "dropdown_content">
<a href = "../skychart_info"><h5>Programming the Sky Map</h5></a>
<a href = "../rosetta"><h5>Rosetta Spacecraft</h5></a>
<a href = "../history"><h5>History Of Astronomy</h5></a>
<a href = "../subfields"><h5>Subfields Of Astronomy</h5></a>
<a href = "../halley"><h5>Halley's Comet</h5></a>
</div>

```



```

</div>

<div class = "dropdown">
  <a class = "drop_link" href = "#"><h5>Tools</h5></a>
  <div class = "dropdown_content">
    <a href = "../skychart"><h5>Sky Chart</h5></a>
    <a href = "../nbody"><h5>N-Body Simulator</h5></a>
    <a href = "../solar_system"><h5>Solar System</h5></a>
    <a href = "../sundial"><h5>Sundial</h5></a>
  </div>
</div>

  <a href = "../"><h5>Home</h5></a>
</div>
</div>

```

Folosirea bibliotecii THREE.js:

```

function init(){
  scene = new THREE.Scene();

  canvasWidth = getBrowserWidth();
  canvasHeight = getBrowserHeight();
  aspect_ratio = canvasWidth / canvasHeight;
  camera = new THREE.PerspectiveCamera(75, aspect_ratio, 0.01, 200);
  camera.lookAt(new THREE.Vector3(0,0,0));
  renderer = new THREE.WebGLRenderer({antialias:true, });

  renderer.setSize(canvasWidth, canvasHeight);
  scene.background = new THREE.Color(0x020202);

  renderer.domElement.style="position:absolute; top:0px; left:0px; margin:0px; width: 100%; height:
    100%;"

  document.getElementById('bellowAbout').appendChild(renderer.domElement);

  camera.position.z = 2;

  controls = new THREE.TrackballControls(camera, renderer.domElement);
  controls.rotateSpeed = 10;
  controls.zoomSpeed = 0.12;
  controls.rotateCamera();
  controls.minDistance = 0.1;
  controls.maxDistance = 90;
  controls.dynamicDampingFactor = 0.3;
}

```

Calculul pozitiilor planetelor pentru “Sky Chart”:

```
function computePlanetCoordinates(vsopData, time){
    var helioCoord = new Object();
    helioCoord.x = computeVSOP(vsopData.x, time);
    helioCoord.y = computeVSOP(vsopData.y, time);
    helioCoord.z = computeVSOP(vsopData.z, time);
    return helioCoord;
}

function computeVSOP(seriesArray, time){
    var result = 0;
    var i = 0;
    for(series of seriesArray){
        result += computeVSOPPolynom(series, time) * Math.pow(time, i);
        i++;
    }
    return result;
}

function computeVSOPPolynom(terms, time){
    var result = 0;
    for(variables of terms){
        if(Math.abs(variables[0]) < VSOP_TERM_LIMIT)
            break;

        result += variables[0] * Math.cos(variables[1] + variables[2] * time);
    }

    return result;
}
```

Folosirea bibliotecii pySVG:

```
def draw_sundial(latitude, longitude, scale, time_zone = 0, time_type = "civil", city_name = ""):
    margin = 5
    relative_scale = scale / 800.0;
    width = scale - 2 * margin;
    height = scale - 2 * margin;
    outter_circle_distance = 35 * relative_scale
    inner_circle_distance = 0
    total_circle_distance = outter_circle_distance + inner_circle_distance

    svg_document = svg()
    shape_builder = ShapeBuilder()
    svg_defs = defs()

    main_circle = shape_builder.createCircle(width / 2.0 + margin, height / 2.0 + margin, width /
    2.0 - outter_circle_distance, strokewidth = 3, stroke = "black")
    #secondary_circle = shape_builder.createCircle(width / 2.0 + margin, height / 2.0 + margin,
    width / 2.0 - total_circle_distance, strokewidth = 3, stroke = "gray")
```

```

outside_circle = shape_builder.createCircle(width / 2.0 + margin, height / 2.0 + margin, width
/ 2.0, strokewidth = 3, stroke = "black")

gnomon_base_x = width / 2.0 + margin
gnomon_base_y = height / 2.0 + margin
gnomon_base_width = get_gnomon_base_width(latitude, scale)
line_length = width / 2.0 - outter_circle_distance
clip_path = clipPath(id="pathRect")
clip_path.addElement(main_circle)
svg_defs.addElement(clip_path)
svg_document.addElement(svg_defs)

style = pysvg.builders.StyleBuilder()
style.setFontFamily("Arial,Helvetica")

style.setFontSize(str(20 * relative_scale) + "px");
for time in range(6 * 4, 18 * 4 + 1):
    civil_time_correction = 0

    if time_type == "civil":
        civil_time_correction = longitude - math.radians(15 * time_zone)

    hour_angle = math.radians(time / 4.0 * 15 - 180) + civil_time_correction
    length = 0;
    color = ""
    start_length = 0;
    stroke_width = 0;

    if time % 4 == 0:
        length = line_length
        color = "rgb(57, 112, 233)"
        start_length = 0
        stroke_width = 3
    elif time % 4 == 2:
        length = line_length #inner_circle_distance
        color = "rgb(89, 135, 237)"
        stroke_width = 2
        start_length = 0 #width / 2.0 - total_circle_distance
    else:
        length = line_length #inner_circle_distance
        color = "rgb(144, 174, 238)"
        start_length = 0 #width / 2.0 - total_circle_distance
        stroke_width = 1

    angle = math.atan(math.tan(hour_angle) * math.sin(latitude))

    if hour_angle < -math.pi / 2.0:
        angle -= math.pi
    elif hour_angle > math.pi / 2.0:
        angle += math.pi

    line_start_x = gnomon_base_x + math.sin(angle) * start_length
    line_start_y = gnomon_base_y - math.cos(angle) * start_length
    line_end_x = line_start_x + math.sin(angle) * length
    line_end_y = line_start_y - math.cos(angle) * length

```

```

line = shape_builder.createLine(line_start_x, line_start_y, line_end_x, line_end_y,
strokewidth=stroke_width, stroke=color)
line.set_clip_path("url(#{s})" % "pathRect")
svg_document.addElement(line)

if time % 4 == 0:
    cos_angle = math.cos(angle)
    sin_angle = math.sin(angle)
    text_x = line_end_x + math.sin(angle) * outter_circle_distance / 4.0
    text_y = line_end_y - math.cos(angle) * outter_circle_distance / 4.0
    hour = hours[(int)(time / 4) - 5]
    #text = int(math.degrees(hour_angle))
    #text = int(math.degrees(angle))
    deg_angle = str(int(math.degrees(angle)))
    text = pysvg.text.text(hour, x = text_x, y = text_y, fill = "rgb(26, 56, 88)")
    text.set_text_anchor("middle")
    text.set_transform("rotate("+ deg_angle +", " + str(text_x) + "," + str(text_y) + ")")
    text.set_style(style.getStyle())
    #text.set_rotate(math.degrees(angle))
    svg_document.addElement(text)

#draw info
city_name_text = pysvg.text.text(city_name, x = width / 2.0, y = height - 0.15 * scale)
city_name_text.set_text_anchor("middle")

svg_document.addElement(outside_circle)
svg_document.addElement(main_circle)
#svg_document.addElement(secondary_circle)

#city_name_text.set_text_size("1em")
#svg_document.addElement(city_name_text)

#draw correction
style.setFontSize(str(15 * relative_scale) + "px")
start_y = 100 * relative_scale;
dist_y = 16 * relative_scale;
dist_x = 50 * relative_scale

for i in range(0, 12):
    for j in range(0, 2):
        text = months[i] + " " + days[j] + ": " + ('{0:+d}'.format(equation_of_time[i][j]))
        x = width / 2.0 - dist_x / 2.0 + j * dist_x
        y = height / 2.0 + start_y + dist_y * (i + 1.5)
        correction_text = pysvg.text.text(text, x = x, y = y)
        if j == 0:
            correction_text.set_text_anchor("end")
        else:
            correction_text.set_text_anchor("start")
        correction_text.set_style(style.getStyle())
        svg_document.addElement(correction_text)

style.setFontSize(str(20 * relative_scale) + "px")
correction_text = pysvg.text.text("Equation Of Time (minutes)", x = width / 2.0, y = height /
2.0 + start_y)
correction_text.set_text_anchor("middle")
correction_text.set_style(style.getStyle())

```

```

svg_document.addElement(correction_text)

arrow_width = 60 * relative_scale
arrow_height = 80 * relative_scale

arrow_x = width - total_circle_distance - margin - 210 * relative_scale
arrow_y = width / 2.0 + 140 * relative_scale

arrow_drawing = arrow(arrow_x, arrow_y, arrow_width, arrow_height, stroke = "#000",
fill="rgb(231,76,60)")

if latitude < 0:
    arrow_x = arrow_x + arrow_width / 2.0
    arrow_y = arrow_y + arrow_height / 2.0
    arrow_drawing.set_transform("rotate(180," + str(arrow_x) + ", " + str(arrow_y) + ")")

svg_document.addElement(arrow_drawing)

svg_document.addElement(shape_builder.createLine(gnomon_base_x, gnomon_base_y,
gnomon_base_x, gnomon_base_y - gnomon_base_width, strokewidth=4,
stroke="rgb(231,76,60)")
return svg_document.getXML()

```

8. Bibliografia

- Jean Meeus - Astronomical Algorithms, Second Edition
- Map Projections - A Working Manual
- Jill Vincent - The mathematics of sundials
- <https://www.caglow.com/info/compute/vsop87>
- [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- <https://en.wikipedia.org/wiki/JavaScript>
- https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- <https://en.wikipedia.org/wiki/JQuery>
- <https://en.wikipedia.org/wiki/Three.js>
- <https://en.wikipedia.org/wiki/HTML>
- [https://en.wikipedia.org/wiki/Rosetta_\(spacecraft\)](https://en.wikipedia.org/wiki/Rosetta_(spacecraft))
- <https://en.wikipedia.org/wiki/Astronomy>
- [https://en.wikipedia.org/wiki/Halley%27s Comet](https://en.wikipedia.org/wiki/Halley%27s_Comet)