Appendix 2: Code

```
<html lang="en" class="no-touch">
<head>
  <meta charset="UTF-8">
  <title>Model</title>
  <meta name="robots" content="noindex">
  <style>
    body{
      font-family: Calibri, Arial;
      display: none;
      background: #1f1e23 url(./4.png);
      color: #eeeeee;
      overflow: hidden;
    }
    .display{
      justify-content: center;
      flex: 2;
      background: #151515;
    }
    .settings_app{
      background: #2f2f2f57;
      width: 100%;
      height: 687px;
      overflow-x: auto;
      /*border: 2px #151515 solid;*/
      color: #eeeeee;
      flex: 1;
    }
    .settings_app::-webkit-scrollbar-track {
      -webkit-box-shadow: inset 0 0 6px rgba(0, 0, 0, 0.3);
      border-radius: 10px;
      background-color: #151515;
    }
    .settings_app::-webkit-scrollbar {
      width: 12px;
      background-color: #151515;/*#F5F5F5;*/
    }
    .settings_app::-webkit-scrollbar-thumb {
      border-radius: 10px;
      -webkit-box-shadow: inset 0 0 6px rgba(0, 0, 0, .3);
      background-color: #555;
    }
    .flex{
      display: flex;
    }
    .message{
      display: list-item;
      text-align: center;
      width: 100%;
      margin-bottom: 8px;
```

```css
}
.message a{
  margin-left: 5px;
  color: #5bbbe0;
}
.trajectory{
  background: #151515;
}
input{
  width: 100%;
  background: #292828b3;
  color: #eeeeee;
  padding: 4px 5px;
}
input[type=color]{
    padding: 0 1px;
    position: relative;
    top: 2px;
    margin-left: 4px;
}
.display{
  width: max-content;
  position: relative;
}
.settings_one{
  padding: 5px 15px;
}
.settings_planet{
  background: #1515152b;
}
.settings_sputnik{
  background: #1515152b;
}
button{
  margin: 5px;
  margin-left: calc(50% - 50px);
  width: 100px;
}
.control_panel_1{
  position: absolute;
  top: 100px;
  left: 15px;
  display: flex;
  flex-direction: column;
  flex-wrap: wrap;
  max-height: 620px;
}
.control_panel_1 button{
  margin: 0;
  width: 34px;
  border-radius: 50%;
  padding: 5px 0px;
  height: 34px;
  text-align: center;
  font-weight: bold;
  font-size: 10px;
  color: #482323;
}
.control_panel_2{
  position: absolute;
  bottom: 10px;
}
.trajectory{
  border: 2px #151515 solid;
}
.planet_name_h1{
  background: #00000066;
  cursor: pointer;
}
.planet_name_h1 p{
  padding: 5px 15px;
  margin: 8px;
  text-align: center;
  font-size: 20px;
  font-family: Calibri, Arial;
}
.hide{
  display: none !important;
}
.bg{
  position: fixed;
  width: 100%;
  height: 100%;
  background: #202327d4;
  top: 0;
  left: 0;
}
.settings_body{
  position: fixed;
  top: 30%;
  left: calc(50% - 215px);
  background: #151515d9;
  padding: 15px;
  margin: auto;
  width: 400px;
}
.planet_color, .star_color, .body_color{
  width: 40px;
}
.settings_app .planet_h1{
  padding-bottom: 0;
}
```

```
.settings_app .planets, .settings_app .sputniks{
  padding: 0 15px;
}

/*Button animation*/
.btn_cont_anim {
  width: 150px;
  height: 40px;
  margin-left: 5px;
  margin-right: 5px;
  margin-top: 0;
  position: relative;
  display: inline-block;
  border-radius: 3px;
}
.line_anim {
  stroke: #009FFD;
  stroke-dasharray: 85 400;
  stroke-dashoffset: -223;
  stroke-width: 6px;
  fill: transparent;
  transition: 1s all ease;
}
.line_anim.line_red {
  stroke: #cc2222;
}
.line_anim.line_green {
  stroke: #207519;
}
.btn_cont {
  margin: -40px 0 0 0;
}
.btn_cont button {
    width: 150px;
    background: none;
    color: white;
    font-weight: 100;
    font-size: 15px;
    text-decoration: none;
    border: none;
    cursor: pointer;
    height: 40px;
    margin: 0;
}
.btn_cont_anim:hover .line_anim {
  stroke: #06D6A0;
  stroke-dasharray: 50 0;
  stroke-width: 3px;
  stroke-dashoffset: 0;
}
.btn_cont_anim:hover .line_anim.line_red {
      stroke: #65271b;
    }
    .btn_cont_anim:hover .line_anim.line_green {
      stroke: #14a001;
    }

    h1{
      position: absolute;
      left: 0px;
      top: -20px;
      z-index: 2;
      font-size: 22px;
    }
    .settings_metric{
      position: absolute;
      right: 0px;
      top: 0;
      z-index: 2;
      font-size: 22px;
    }
  </style>
</head>
<body style="display: block;">
  <div class="line" style="position:relative;">
    <h1>Gravity Simulator</h1>
    <div class="message"></div>
    <div class="settings_metric">
        <select>
          <option value="0">Standard Form</option>
          <option value="1">Scientific Form</option>
          <option value="2">Astronomical Unit</option>
        </select>
    </div>
  </div>
  <div class="line flex">
    <div class="display flex">
      <canvas class="trajectory" width="1366" height="1366" style=""></canvas>
      <div class="control_panel_2">
        <div class="btn_cont_anim">
          <svg height="40" width="150">
            <rect class="line_anim line_green" height="40" width="150"></rect>
            <div class="btn_cont">
              <button class="start">start</button>
            </div>
          </svg>
        </div>
      </div>
    </div>
    <div class="settings_app">
      <div class="settings_one">
        <center>1 real second = <span class="time_compare"></span></center>
      </div>
      <div class="settings_one">
        <label>Time Multiplier(affects accuracy)</label>
        <input class="step" value="60480">
        <input class="step_range"type="range" min="0" max="11" class="size"
value="6"><!--1111111111-->
      </div>
      <div class="settings_one">
        <label>Timer Step(ms)</label>
        <input class="step_t" value="10">
      </div>
      <div class="settings_one">
        <label>Star Mass(kg)</label>
        <input class="star_mass format" value="1 989 000 000 000 000 000 000 000
000 000">
      </div>
      <div class="settings_one">
        <label>Star color</label>
        <input type="color" class="star_color" value="#c3ce1c">
      </div>
      <div class="settings_one planet_h1">
        Planets
      </div>
      <div class="settings_one planets">
        <div class="settings_planet">
          <div class="planet_name_h1">
            <p>Mercury</p>
          </div>
          <div class="planet_content hide">
            <div class="settings_one">
              <label>Planet Name</label>
              <input class="planet_name" value="Mercury">
            </div>
            <div class="settings_one">
              <label>Planet Mass(kg)</label>
              <input class="planet_mass format" value="330 000 000 000 000 000 000
000">
            </div>
            <div class="settings_one">
              <label>Distance from the star(<span
class="metric_dist">km</span>)</label>
              <input class="planet_rad format distance" value="57 900 000">
            </div>
            <div class="settings_one">
              <label>Speed Module(km/s)</label>
              <input class="planet_v" value="47.4">
            </div>
            <div class="settings_one">
              <label>Color</label>
              <input type="color" class="planet_color" value="#b85300">
            </div>
            <div class="settings_one planet_h1">
              Satellites
            </div>
            <div class="settings_one sputniks"></div>
            <div class="settings_one" style="padding: 0px 15px;">
              <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
                <svg height="40" width="150">
                  <rect class="line_anim" height="40" width="150"></rect>
                  <div class="btn_cont">
                    <button class="add_sputnik">Add</button>
                  </div>
                </svg>
              </div>
            </div>
            <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
              <svg height="40" width="150">
                <rect class="line_anim line_red" height="40" width="150"></rect>
                <div class="btn_cont">
                  <button class="delete_planet">Delete</button>
                </div>
              </svg>
            </div>
          </div>
        </div>
        <div class="settings_planet">
          <div class="planet_name_h1">
            <p>Venus</p>
          </div>
          <div class="planet_content hide">
            <div class="settings_one">
              <label>Planet Name/label>
              <input class="planet_name" value="Venus">
            </div>
            <div class="settings_one">
              <label>Planet Mass(kg)</label>
              <input class="planet_mass format" value="4 870 000 000 000 000 000
000 000">
            </div>
            <div class="settings_one">
              <label>Distance from the star(<span
class="metric_dist">km</span>)</label>
              <input class="planet_rad format distance" value="108 200 000">
            </div>
            <div class="settings_one">
              <label>Speed Module(km/s)</label>
              <input class="planet_v" value="35.0">
            </div>
            <div class="settings_one">
              <label>Color</label>
```

```
       <input type="color" class="planet_color" value="#ffa200">
      </div>
      <div class="settings_one planet_h1">
       Satellites
      </div>
      <div class="settings_one sputniks"></div>
      <div class="settings_one" style="padding: 0px 15px;">
       <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
        <svg height="40" width="150">
         <rect class="line_anim" height="40" width="150"></rect>
         <div class="btn_cont">
          <button class="add_sputnik">Add</button>
         </div>
        </svg>
       </div>
      </div>
      <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
       <svg height="40" width="150">
        <rect class="line_anim line_red" height="40" width="150"></rect>
        <div class="btn_cont">
         <button class="delete_planet">Delete</button>
        </div>
       </svg>
      </div>
     </div>
    </div>
    <div class="settings_planet">
     <div class="planet_name_h1">
      <p>Earth</p>
     </div>
     <div class="planet_content hide">
      <div class="settings_one">
       <label>Planet Name</label>
       <input class="planet_name" value="Earth">
      </div>
      <div class="settings_one">
       <label>Planet Mass(kg)</label>
       <input class="planet_mass format" value="5 970 000 000 000 000 000
000 000">
      </div>
      <div class="settings_one">
       <label>Distance from the star(<span
class="metric_dist">km</span>)</label>
       <input class="planet_rad format distance" value="149 597 870.7">
      </div>
      <div class="settings_one">
       <label>Speed Module(km/s)</label>
       <input class="planet_v" value="29.8">
      </div>
```

```
            <div class="settings_one">
              <label>Color</label>
              <input type="color" class="planet_color" value="#0091ff">
            </div>
            <div class="settings_one planet_h1">
              Satellites
            </div>
            <div class="settings_one sputniks">
              <div class="settings_sputnik">
                <div class="planet_name_h1">
                  <p>Moon</p>
                </div>
                <div class="planet_content hide">
                  <div class="settings_one">
                    <label>Name of Satellite</label>
                    <input class="planet_name" value="Moon">
                  </div>
                  <div class="settings_one">
                    <label>Satellite Mass(kg)</label>
                    <input class="sputnik_mass format" value="73 600 000 000 000
000 000 000">
                  </div>
                  <div class="settings_one">
                    <label>Distance from the Planet(<span
class="metric_dist">km</span>)</label>
                    <input class="sputnik_rad format distance" value="384 400">
                  </div>
                  <div class="settings_one">
                    <label>Speed Module(km/s)</label>
                    <input class="sputnik_v" value="1.02">
                  </div>
                  <div class="settings_one">
                    <label>Color</label>
                    <input type="color" class="planet_color" value="#888888">
                  </div>
                  <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
                    <svg height="40" width="150">
                      <rect class="line_anim line_red" height="40"
width="150"></rect>
                      <div class="btn_cont">
                        <button class="delete_planet">Delete</button>
                      </div>
                    </svg>
                  </div>
                </div>
              </div>
            </div>
            <div class="settings_one" style="padding: 0px 15px;">
              <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
```

```
                <svg height="40" width="150">
                  <rect class="line_anim" height="40" width="150"></rect>
                  <div class="btn_cont">
                    <button class="add_sputnik">Add</button>
                  </div>
                </svg>
              </div>
            </div>
            <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
              <svg height="40" width="150">
                <rect class="line_anim line_red" height="40" width="150"></rect>
                <div class="btn_cont">
                  <button class="delete_planet">Delete</button>
                </div>
              </svg>
            </div>
          </div>
        </div>
        <div class="settings_planet">
          <div class="planet_name_h1">
            <p>Mars</p>
          </div>
          <div class="planet_content hide">
            <div class="settings_one">
              <label>Planet Name</label>
              <input class="planet_name" value="Mars">
            </div>
            <div class="settings_one">
              <label>Planet Mass(kg)</label>
              <input class="planet_mass format" value="642 000 000 000 000 000 000
000">
            </div>
            <div class="settings_one">
              <label>Distance from the Star(<span
class="metric_dist">km</span>)</label>
              <input class="planet_rad format distance" value="227 900 000">
            </div>
            <div class="settings_one">
              <label>Speed Module(km/s)</label>
              <input class="planet_v" value="24.1">
            </div>
            <div class="settings_one">
              <label>Color</label>
              <input type="color" class="planet_color" value="#FF0000">
            </div>
            <div class="settings_one planet_h1">
              Satellite
            </div>
            <div class="settings_one sputniks"></div>
            <div class="settings_one" style="padding: 0px 15px;">
```

```
       <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
        <svg height="40" width="150">
         <rect class="line_anim" height="40" width="150"></rect>
         <div class="btn_cont">
          <button class="add_sputnik">Add</button>
         </div>
        </svg>
       </div>
      </div>
      <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
       <svg height="40" width="150">
        <rect class="line_anim line_red" height="40" width="150"></rect>
        <div class="btn_cont">
         <button class="delete_planet">Delete</button>
        </div>
       </svg>
      </div>
     </div>
    </div>
    <div class="settings_planet">
     <div class="planet_name_h1">
      <p>Jupiter</p>
     </div>
     <div class="planet_content hide">
      <div class="settings_one">
       <label>Planet Name</label>
       <input class="planet_name" value="Jupiter">
      </div>
      <div class="settings_one">
       <label>Planet Mass(kg)</label>
       <input class="planet_mass format" value="1 898 000 000 000 000 000
000 000 000">
      </div>
      <div class="settings_one">
       <label>Distance from the Star(<span
class="metric_dist">km</span>)</label>
       <input class="planet_rad format distance" value="778 600 000">
      </div>
      <div class="settings_one">
       <label>Speed Module(km/s)</label>
       <input class="planet_v" value="13.1">
      </div>
      <div class="settings_one">
       <label>Color</label>
       <input type="color" class="planet_color" value="#a7876c">
      </div>
      <div class="settings_one planet_h1">
       Satellite
      </div>
```

```
            <div class="settings_one sputniks"></div>
            <div class="settings_one" style="padding: 0px 15px;">
              <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
                <svg height="40" width="150">
                  <rect class="line_anim" height="40" width="150"></rect>
                  <div class="btn_cont">
                    <button class="add_sputnik">Add</button>
                  </div>
                </svg>
              </div>
            </div>
            <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
              <svg height="40" width="150">
                <rect class="line_anim line_red" height="40" width="150"></rect>
                <div class="btn_cont">
                  <button class="delete_planet">Delete</button>
                </div>
              </svg>
            </div>
          </div>
        </div>
        <div class="settings_planet">
          <div class="planet_name_h1">
            <p>Saturn</p>
          </div>
          <div class="planet_content hide">
            <div class="settings_one">
              <label>Planet Name</label>
              <input class="planet_name" value="Saturn">
            </div>
            <div class="settings_one">
              <label>Planet Mass(kg)</label>
              <input class="planet_mass format" value="568 000 000 000 000 000 000
000 000">
            </div>
            <div class="settings_one">
              <label>Distance from the Star(<span
class="metric_dist">km</span>)</label>
              <input class="planet_rad format distance" value="1 433 500 000">
            </div>
            <div class="settings_one">
              <label>Speed Module(km/s)</label>
              <input class="planet_v" value="9.7">
            </div>
            <div class="settings_one">
              <label>Color</label>
              <input type="color" class="planet_color" value="#9c8154">
            </div>
            <div class="settings_one planet_h1">
```

```
              Satellite
            </div>
            <div class="settings_one sputniks"></div>
            <div class="settings_one" style="padding: 0px 15px;">
              <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
                <svg height="40" width="150">
                  <rect class="line_anim" height="40" width="150"></rect>
                  <div class="btn_cont">
                    <button class="add_sputnik">Add</button>
                  </div>
                </svg>
              </div>
            </div>
            <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
              <svg height="40" width="150">
                <rect class="line_anim line_red" height="40" width="150"></rect>
                <div class="btn_cont">
                  <button class="delete_planet">Delete</button>
                </div>
              </svg>
            </div>
          </div>
        </div>
        <div class="settings_planet">
          <div class="planet_name_h1">
            <p>Uranus</p>
          </div>
          <div class="planet_content hide">
            <div class="settings_one">
              <label>Planet Name</label>
              <input class="planet_name" value="Uranus">
            </div>
            <div class="settings_one">
              <label>Planet Mass(kg)</label>
              <input class="planet_mass format" value="86 800 000 000 000 000 000
000 000">
            </div>
            <div class="settings_one">
              <label>Distance from the Star(<span
class="metric_dist">km</span>)</label>
              <input class="planet_rad format distance" value="2 872 500 000">
            </div>
            <div class="settings_one">
              <label>Speed module(km/s)</label>
              <input class="planet_v" value="6.8">
            </div>
            <div class="settings_one">
              <label>Color</label>
              <input type="color" class="planet_color" value="#17b0c4">
            </div>
            <div class="settings_one planet_h1">
              Satellite
            </div>
            <div class="settings_one sputniks"></div>
            <div class="settings_one" style="padding: 0px 15px;">
              <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% - 75px);">
                <svg height="40" width="150">
                  <rect class="line_anim" height="40" width="150"></rect>
                  <div class="btn_cont">
                    <button class="add_sputnik">Add</button>
                  </div>
                </svg>
              </div>
            </div>
            <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left: calc(50% - 75px);">
              <svg height="40" width="150">
                <rect class="line_anim line_red" height="40" width="150"></rect>
                <div class="btn_cont">
                  <button class="delete_planet">Delete</button>
                </div>
              </svg>
            </div>
          </div>
        </div>
        <div class="settings_planet">
          <div class="planet_name_h1">
            <p>Neptune</p>
          </div>
          <div class="planet_content hide">
            <div class="settings_one">
              <label>Planet Name</label>
              <input class="planet_name" value="Neptune">
            </div>
            <div class="settings_one">
              <label>Planet Mass(kg)</label>
              <input class="planet_mass format" value="102 000 000 000 000 000 000 000 000">
            </div>
            <div class="settings_one">
              <label>Distance from the Star(<span class="metric_dist">km</span>)</label>
              <input class="planet_rad format distance" value="4 495 100 000">
            </div>
            <div class="settings_one">
              <label>Speed Module(km/s)</label>
              <input class="planet_v" value="5.4">
            </div>
            <div class="settings_one">
```

```
              <label>Color</label>
              <input type="color" class="planet_color" value="#0000FF">
            </div>
            <div class="settings_one planet_h1">
              Satellites
            </div>
            <div class="settings_one sputniks"></div>
            <div class="settings_one" style="padding: 0px 15px;">
              <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
                <svg height="40" width="150">
                  <rect class="line_anim" height="40" width="150"></rect>
                  <div class="btn_cont">
                    <button class="add_sputnik">Add</button>
                  </div>
                </svg>
              </div>
            </div>
            <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
              <svg height="40" width="150">
                <rect class="line_anim line_red" height="40" width="150"></rect>
                <div class="btn_cont">
                  <button class="delete_planet">Delete</button>
                </div>
              </svg>
            </div>
          </div>
        </div>
        <div class="settings_planet">
          <div class="planet_name_h1">
            <p>Pluto</p>
          </div>
          <div class="planet_content hide">
            <div class="settings_one">
              <label>Planet Name</label>
              <input class="planet_name" value="Pluto">
            </div>
            <div class="settings_one">
              <label>Planet Mass(kg)</label>
              <input class="planet_mass format" value="14 600 000 000 000 000 000
000">
            </div>
            <div class="settings_one">
              <label>Distance from the Star(<span
class="metric_dist">km</span>)</label>
              <input class="planet_rad format distance" value="5 906 400 000">
            </div>
            <div class="settings_one">
              <label>Speed Module(km/s)</label>
              <input class="planet_v" value="4.7">
            </div>
            <div class="settings_one">
              <label>Color</label>
              <input type="color" class="planet_color" value="#777777">
            </div>
            <div class="settings_one planet_h1">
              Satellite
            </div>
            <div class="settings_one sputniks"></div>
            <div class="settings_one" style="padding: 0px 15px;">
              <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
                <svg height="40" width="150">
                  <rect class="line_anim" height="40" width="150"></rect>
                  <div class="btn_cont">
                    <button class="add_sputnik">Add</button>
                  </div>
                </svg>
              </div>
            </div>
            <div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">
              <svg height="40" width="150">
                <rect class="line_anim line_red" height="40" width="150"></rect>
                <div class="btn_cont">
                  <button class="delete_planet">Delete</button>
                </div>
              </svg>
            </div>
          </div>
        </div>
      </div>
      <div class="settings_one" style="padding: 0px 15px;">
        <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
          <svg height="40" width="150">
            <rect class="line_anim" height="40" width="150"></rect>
            <div class="btn_cont">
              <button class="add_planet">Add</button>
            </div>
          </svg>
        </div>
        <!--<button class="add_planet">Add</button>-->
      </div>
    </div>
  </div>
  <div class="absolute">
    <div class="control_panel_1">
      <button class="add_body show" data-show="settings_body" style="color:
#eeeeee;background: #3c3834;font-size: 14px;">+</button>
    </div>
    <div class="bg hide"></div>
    <div class="settings_body
    show_me hide">
      <div class="settings_one">
        <label>Body Mass(kg)</label>
        <input class="body_mass format" value="100 000 000 000 000 000 000 000">
      </div>
      <div class="settings_one">
        <label>Speed Module(km/s)</label>
        <input class="body_v format" value="30">
      </div>
      <div class="settings_one">
        <label>Color</label>
        <input type="color" class="body_color" value="#FF0000">
      </div>
      <div class="settings_one">
        <div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50% -
75px);">
          <svg height="40" width="150">
            <rect class="line_anim" height="40" width="150"></rect>
            <div class="btn_cont">
              <button class="add_body_settings">Add</button>
            </div>
          </svg>
        </div>
      </div>
    </div>
  </div>
</body>
<script>
    let map = {};//width and height of the window in km
    map.x = 14000000000;//width of the whole field in km
    let w = document.querySelector('.trajectory').width;//number of dots on the
canvas (width)
    let h = document.querySelector('.trajectory').height;//number of dots ont he
canvas (height)
    map.y = Math.round(map.x / h * w);//width of the whole field in km
    map_dp = 0.25;//coefficient of scaling planets
    map_dm = 1.2;//coefficient of scaling the map
    let start = {};//Displacement to the center of the window
    document.querySelector('.trajectory').setAttribute('style', 'width:
'+(w/2)+'px;height: '+(h/2)+'px;');//If canvas has 2 000 width, 2 px will mean 1 px
    start.x = w/2;//default camera position (width)
    start.y = h/2;//default camera position (height)
    start.n = -1;//which body to center
    let dt = 0;//step fo calculations
    let step_t = 0;//timer step
    let canvas = document.querySelector('.trajectory');
    let ctx = canvas.getContext('2d');//starting canvas in 2d
    let arr_p = [];//empty array for future celestial bodies
    let p = {};//calculations
    let b = {};//calculations
    let arr_b = [];//empty array for asteroids
    let map_d = 0;//coefficient of the enlargement on the map
    let no_right = 0;//message for "Not accurate"
    let number_format = 0;//message for "Not accurate"
    //coloring the planet titles while the page is loading
    document.querySelectorAll('.planet_content').forEach(function(elem, i){
      let val = elem.querySelector('.planet_color').value;
      elem.parentElement.querySelector('.planet_name_h1').setAttribute('style',
'background: ' + val + '66;');
    });
    //Function for calculating time comparing to real life time
    function time_compare(){
      let step = Number(document.querySelector('.step').value);
      let t = Number(document.querySelector('.step_t').value);
      let y = Math.trunc(step * 1000 / t / (60 * 60 * 24 * 365));
      let ost = (step * 1000 / t) % (60 * 60 * 24 * 365);
      let d = Math.trunc(ost / (60 * 60 * 24));
      ost = ost % (60 * 60 * 24);
      let h = Math.trunc(ost / (60 * 60));
      ost = ost % (60 * 60);
      let m = Math.trunc(ost / 60);
      ost = ost % 60;
      let s = ost;
      document.querySelector('.time_compare').innerHTML = (y > 0 ? (y + ' years
') : '') +
                                                        (d > 0 ? (d + ' days ') :
'') +
                                                        (h > 0 ? (h + ' hours ') :
'') +
                                                        (m > 0 ? (m + ' minutes
') : '') +
                                                        (s > 0 ? (s + ' seconds
') : '') +
                                                        'in the simulation ' +
                                                        (no_right > 0 ? ('<br><stan
style="color:red;"> Low Accuracy</span>') : '');
    }
    //добавляем функцию выше для полей при изменении которых происходит расчет
    document.querySelector('.step').addEventListener('blur', time_compare);
    document.querySelector('.step_t').addEventListener('blur', time_compare);
    document.querySelector('.step').addEventListener('input', time_compare);
    document.querySelector('.step_t').addEventListener('input', time_compare);
    //Calling the time calculation function while loading the page
    time_compare();
    //Showing whole page after loading the scripts
    document.querySelector('body').setAttribute('style', 'display: block;');

    function NumberFormat(f = 0, z = 5){
      //normal format
      if(f == 0) {
```

```
  document.querySelectorAll('.format').forEach(function(elem, i){
      let val = Number(elem.value.replace( /\s/g, ""));
      if(elem.classList.contains('au')){
          val = elem.dataset.au;
          elem.classList.remove('au');
      };
      //trasnlating into normal format
      let temp = val;
      let n = 0;
      let ost= 0;
      let result = '';
      //counting the decimal places
      for(var i = 0; temp > 1; i++) {
        temp /= 10;
      }
      temp = val;
      let zi = i - z;

      while(temp >= 1){
        ost = temp % 10;
        temp = Math.trunc((temp / 10));
        result = (n >= zi ? ost : 0) + result;
        if(n % 3 == 2){
          result = ' ' + result;
        }
        n++;
      }

      elem.value = result.trim();
  });
  document.querySelectorAll('.metric_dist').forEach(function(elem, i){
      elem.innerText = 'km';
  });
// The E format, example: 2.29E10
} else if(f == 1) {
  document.querySelectorAll('.format').forEach(function(elem, i){
      let val = Number(elem.value.replace( /\s/g, ""));
      if(elem.classList.contains('au')){
          val = elem.dataset.au;
          elem.classList.remove('au');
      };
      //If the number is bigger
      if(val > 10000){//can change
        //translating into 2.29E10 form
        let temp = val;
        let n = 0;
        while(temp >= 10){
          temp = temp / 10;
          n++;
        }
        temp = +temp.toFixed(z);//how many decimals need to be changed
```

```
       temp = temp + 'e' + n;

       elem.value = temp;
      }
    });
    document.querySelectorAll('.metric_dist').forEach(function(elem, i){
      elem.innerText = 'km';
    });
   //translating into au unit
   } else if(f == 2) {
    document.querySelectorAll('.format.distance').forEach(function(elem, i){
      elem.classList.add('au');
      let val = Number(elem.value.replace( /\s/g, ""));
      let temp = String(+(val / 149597870.7).toFixed(z));
      elem.dataset.au = val;
      elem.value = temp.trim();
    });
    console.log(document.querySelectorAll('.metric_dist'));
    document.querySelectorAll('.metric_dist').forEach(function(elem, i){
      elem.innerText = 'au';
    });
   }
  }
  //устанавливанем событие выбора формата отображения
  document.querySelector('.settings_metric select').addEventListener('change',
function(){
    NumberFormat(this.value, 5);
    number_format = this.value;
  });

  //Function for translating km into px for drawing
  function convert(p = {x: 0, y: 0}){
   let result = {};
   if(start.n < 0) {
    result.x = Math.round(start.x + p.x/map.x*w);
    result.y = Math.round(start.y + p.y/map.y*h);
   } else if(start.n >= 0){
    result.x = Math.round(start.x + (p.x - arr_p[start.n].x)/map.x*w);
    result.y = Math.round(start.y + (p.y - arr_p[start.n].y)/map.y*h);
   }
   return result;
  };
  //Fucntion for mapping scrolling in and out the planets depending on their mass
and size
  function mappingPlanet(p){
   let r = Number(p.m/Math.pow(10, 20));
   map_d0 = map_d < 0 ? 0 : map_d;
   let r_arc = 1;
   let mapping = [
    {r: 0, p: 1},
    {r: 100, p: 2 + Math.floor(map_d0*0.25)},
    {r: 10000, p: 3 + Math.floor(map_d0*0.5)},
    {r: 500000, p: 4 + Math.floor(map_d0*1)},
    {r: 15000000, p: 5 + Math.floor(map_d0*2)},
    {r: 100000000, p: 6 + Math.floor(map_d0*4)}
  ];
  for(let i = mapping.length - 1; i >= 0 ; i--){
    if(r >= mapping[i].r){
      r_arc = mapping[i].p;
      break;
    }
  }
  return r_arc;
}
//Function for drawing bodies
function draw(){
  ctx.clearRect(0, 0, w, h);
  //Function for drawing 1 body
  function output(p){
    ctx.fillStyle = p.color;
    let r_arc = mappingPlanet(p);
    let planet = {};
    planet = convert(p);
    ctx.beginPath();
    ctx.arc(planet.x,planet.y, r_arc, 0, Math.PI*2, true);
    ctx.fill();
  }
  //addPanelPlanet();
  arr_p.forEach(function(elem, i){
    output(elem);
  });
}

//Timer function
function timerFunc(){
  ctx.clearRect(0, 0, w, h);
  let r = {};
  dt = Number(document.querySelector('.step').value.replace( /\s/g, ""));
  //Vector, speed and x, y calculations
  for(let i = 0; i < arr_p.length; i++){
    for(let j = 0; j < arr_p.length; j++){
      if(i == j) {
        continue;
      }
      r.x = arr_p[j].x - arr_p[i].x;//x projectile
      r.y = arr_p[j].y - arr_p[i].y;
      r.md = Math.round(Math.sqrt(r.x*r.x + r.y*r.y));
      //In a situation where the distance between the bodies is 0
      if(r.md == 0){
        r.md = 0.0000000001;
      }
      let G = 0.0000000000667; //G constant H*m^2/kg^2
```

```
          //let backG = 14992503748;
          //Devivding by 1000000000 for transfering into km/s^2
          let a = G*Number(arr_p[j].m)/(r.md*r.md);
          arr_p[i].ax = a * r.x / (r.md * 1000000000);
          arr_p[i].ay = a * r.y / (r.md * 1000000000);
          arr_p[i].vx = arr_p[i].vx + arr_p[i].ax * dt;
          arr_p[i].vy = arr_p[i].vy + arr_p[i].ay * dt;
        }
      }
      //Mapping the planets (Location)
      for(let i = 0; i < arr_p.length; i++){
        arr_p[i].x = arr_p[i].x + arr_p[i].vx * dt;
        arr_p[i].y = arr_p[i].y + arr_p[i].vy * dt;
      }
      draw();
    }
    //Function that we call when we click the "start" button.
    function startFunc(no_timer = 0){
      let start = document.querySelector('.start');
      //If the button says start
      if(start.innerText == 'start' || (start.innerText == 'start' && no_timer ==
1)) {
        ctx.clearRect(0, 0, start.x*2, start.y*2);
        //Creating array for planets using the data in the code above
        arr_p = [];
        //Adding sun
        let s = {};
        s.name = 'Sun';
        s.m = Number(document.querySelector('.star_mass').value.replace( /\s/g,
""));
        s.color = document.querySelector('.star_color').value;
        s.r = 0;
        s.v = 0;
        s.x = 0;
        s.y = 0;
        s.vx = 0;
        s.vy = 0;
        s.ax = 0;
        s.ay = 0;
        arr_p.push(s);
        //Adding planets
        document.querySelectorAll('.settings_planet').forEach(function(elem, i){
          let p = {};
          p.name = elem.querySelector('.planet_name').value;
          p.m = Number(elem.querySelector('.planet_mass').value.replace( /\s/g,
""));
          p.r =
Number(!(elem.querySelector('.planet_rad').classList.contains('au')) ?
elem.querySelector('.planet_rad').value.replace( /\s/g, "") :
elem.querySelector('.planet_rad').dataset.au.replace( /\s/g, ""));
          p.v = Number(elem.querySelector('.planet_v').value.replace( /\s/g, ""));
          p.color = elem.querySelector('.planet_color').value;
          p.x = 0-p.r;
          p.y = 0;
          p.vx = 0;
          p.vy = p.v;
          p.ax = 0;
          p.ay = 0;
          arr_p.push(p);
          //Adding Satellites
          elem.querySelectorAll('.settings_sputnik').forEach(function(elem, i){
            let sp = {};
            sp.name = elem.querySelector('.planet_name').value;
            sp.m = Number(elem.querySelector('.sputnik_mass').value.replace( /\s/g,
""));
            sp.r =
Number(!(elem.querySelector('.sputnik_rad').classList.contains('au')) ?
elem.querySelector('.sputnik_rad').value.replace( /\s/g, "") :
elem.querySelector('.sputnik_rad').dataset.au.replace( /\s/g, ""));
            sp.v = Number(elem.querySelector('.sputnik_v').value.replace( /\s/g,
""));
            sp.color = elem.querySelector('.planet_color').value;
            sp.x = 0-p.r-sp.r;
            sp.y = 0;
            sp.vx = 0;
            sp.vy = p.v + sp.v;
            sp.ax = 0;
            sp.ay = 0;
            arr_p.push(sp);
          });
        });
        //Adding copy of asteroid array to the array of planets
        let arr_temp = arr_b.map(objects => ({ ...objects }));
        arr_p = arr_p.concat(arr_temp);
        //gathering data from timer
        dt = Number(document.querySelector('.step').value.replace( /\s/g, ""));
        step_t = Number(document.querySelector('.step_t').value.replace( /\s/g,
""));
        //Call the draw function for drawing the planets
        addPanelPlanet();
        draw();

        if(no_timer != 1) {
          //starting timer
          timer = setInterval(timerFunc, step_t);
          //changing the "start" to "stop"
          start.innerText = 'stop';
          //Changing the button color

start.parentElement.parentElement.querySelector('.line_anim').classList.remove('lin
e_green');
```

```
start.parentElement.parentElement.querySelector('.line_anim').classList.add('line_r
ed');
        }
      } else if(start.innerText == 'stop' && no_timer != 1){
        //Stopping timer
        clearTimeout(timer);
        //Changing the "stop" to "start"
        start.innerText = 'start';
        //Changing the color of the button

start.parentElement.parentElement.querySelector('.line_anim').classList.remove('lin
e_red');

start.parentElement.parentElement.querySelector('.line_anim').classList.add('line_g
reen');
      }
    }
    document.querySelector('.start').addEventListener('click', startFunc);
    //Function that we call after creating new satellite or planet
    function addPlanet(this_elem, type = 'planet'){
      //Hiding all the existing planet tabs

this_elem.parentElement.parentElement.parentElement.parentElement.querySelectorAll(
'.planet_content').forEach(function(elem, i){
        elem.classList.add('hide');
      });
      //creating div element
      let planet = document.createElement('div');
      planet.classList.add('settings_'+type);
      //Putting in the code to create new satellite or planet
      if(type == 'planet'){//planets
        planet.innerHTML = '<div class="planet_name_h1">' +
            '<p>New Planet</p>' +
          '</div>' +
          '<div class="planet_content">' +
            '<div class="settings_one">' +
              '<label>Planet Name</label>' +
              '<input class="planet_name" value="New Planet">' +
            '</div>' +
            '<div class="settings_one">' +
            '<label>Planet Mass(kg)</label>' +
            '<input class="planet_mass format" value="5 972 000 000 000 000 000
000">' +
            '</div>' +
            '<div class="settings_one">' +
              '<label>Distance from the Star(<span
class="metric_dist">km</span>)</label>' +
              '<input class="planet_rad format distance" value="120 500 000">' +
            '</div>' +
            '<div class="settings_one">' +
              '<label>Speed Module(km/s)</label>' +
              '<input class="planet_v" value="29.78">' +
            '</div>' +
            '<div class="settings_one">' +
              '<label>Color</label>' +
              '<input type="color" class="planet_color" value="#000000">' +
            '</div>' +
            '<div class="settings_one planet_h1">' +
              'Satellite' +
            '</div>' +
            '<div class="settings_one sputniks"></div>' +
            '<div class="settings_one" style="padding: 0px 15px;">' +
              '<div class="btn_cont_anim" style="margin: 5px;margin-left: calc(50%
- 75px);">' +
                '<svg height="40" width="150">' +
                  '<rect class="line_anim" height="40" width="150"></rect>' +
                  '<div class="btn_cont">' +
                    '<button class="add_sputnik">Add</button>' +
                  '</div>' +
                '</svg>' +
              '</div>' +
            '</div>' +
            '<div class="btn_cont_anim" style="margin-bottom: 5px;margin-left:
calc(50% - 75px);">' +
              '<svg height="40" width="150">' +
                '<rect class="line_anim line_red" height="40" width="150"></rect>'
+
                '<div class="btn_cont">' +
                  '<button class="delete_planet">Delete</button>' +
                '</div>' +
              '</svg>' +
            '</div>' +
          '</div>';
      } else if(type = 'sputnik') {//satellite
        planet.innerHTML = '<div class="planet_name_h1">' +
                  '<p>New Satellite</p>' +
                '</div>' +
                '<div class="planet_content">' +
                  '<div class="settings_one">' +
                    '<label>Satellite Name</label>' +
                    '<input class="planet_name" value="New Satellite">' +
                  '</div>' +
                  '<div class="settings_one">' +
                    '<label>Satellite Mass(kg)</label>' +
                    '<input class="sputnik_mass format" value="73 600 000 000 000
000 000 000">' +
                  '</div>' +
                  '<div class="settings_one">' +
                    '<label>Distance from the Planet(<span
class="metric_dist">km</span>)</label>' +
                    '<input class="sputnik_rad format distance" value="500 400">' +
                  '</div>' +
                  '<div class="settings_one">' +
                    '<label>Speed Module(km/s)</label>' +
                    '<input class="sputnik_v" value="0.82">' +
                  '</div>' +
                  '<div class="settings_one">' +
                    '<label>Color</label>' +
                    '<input type="color" class="planet_color" value="#000000">' +
                  '</div>' +
                  '<div class="btn_cont_anim" style="margin-bottom: 5px;margin-
left: calc(50% - 75px);">' +
                    '<svg height="40" width="150">' +
                      '<rect class="line_anim line_red" height="40"
width="150"></rect>' +
                      '<div class="btn_cont">' +
                        '<button class="delete_planet">Delete</button>' +
                      '</div>' +
                    '</svg>' +
                  '</div>' +
                '</div>';
      }
      //Putting the created element in place

this_elem.parentElement.parentElement.parentElement.parentElement.querySelector('.'
+type+'s').appendChild(planet);
      //пересоздаем события скрытия планет и спутников по заголовку
      document.querySelectorAll('.planet_name_h1').forEach(function(elem, i){
        elem.removeEventListener('click', hidePlanetFunc);
      });
      hidePlanet();
      //пересоздаем события переименования планет и спутников по заголовку
      document.querySelectorAll('.planet_name').forEach(function(elem, i){
        elem.removeEventListener('blur', renamePlanetH1);//blur это потеря фокуса
      });
      namePlanet();
      //пересоздаем события удаления планет и спутников по заголовку
      document.querySelectorAll('.delete_planet').forEach(function(elem, i){
        elem.removeEventListener('click', deletePlanetFunc);
      });
      deletePlanet();
      //пересоздаем события добавления спутника
      document.querySelectorAll('.add_sputnik').forEach(function(elem, i){
        elem.removeEventListener('click', addPlanetFunc);
      });
      addSputnik();
      //удаляем события перерисовки карты при изменении радиуса планеты
      document.querySelectorAll('.planet_rad').forEach(function(elem, i){
        elem.removeEventListener('blur', firstDisplayFunc);
      });
      //удаляем события перерисовки карты при изменении радиуса спутника
      document.querySelectorAll('.sputnik_rad').forEach(function(elem, i){
        elem.removeEventListener('blur', firstDisplayFunc);
      });
      //удаляем события перерисовки карты при изменении цвета
      document.querySelectorAll('.planet_color').forEach(function(elem, i){
        elem.removeEventListener('blur', firstDisplayFunc);
      });
      //добавляем события при изменении цвета и радиуса
      firstDisplay();
      //отрисовываем карту
      firstDisplayFunc();
      //пересоздаем события изменения цвета заголовка планеты и кнопок планет слева
      document.querySelectorAll('.planet_color').forEach(function(elem, i){
        elem.removeEventListener('blur', colorPlanetFunc);
      });
      colorPlanet();

      //устанавливаем формат в километрах
      NumberFormat(0, 5);
      //обновляем планеты в меню слева и назначаем события при нажатии
      addPanelPlanet();
      //устанавливаем выбраный формат
      NumberFormat(number_format, 5);
    }
    //ФУНКЦИИ ДЛЯ СОБЫТИЙ И СОБЫТИЯ ПРИ ЗАГРУЗКЕ СТРАНИЦЫ
    //назначаем событие добавления планеты при загрузке страницы
    document.querySelector('.add_planet').addEventListener('click', function(){
      addPlanet(this, 'planet');
    });

    //Функции добавления спуников
    function addPlanetFunc(){
      addPlanet(this, 'sputnik');
    }
    //назначаем событие добавления спутника при загрузке страницы
    function addSputnik(){
      document.querySelectorAll('.add_sputnik').forEach(function(elem, i){
        elem.addEventListener('click', addPlanetFunc);
      });
    }
    addSputnik();

    //Renaming Planets Function
    function renamePlanetH1(){
      this.parentElement.parentElement.parentElement.querySelector('.planet_name_h1
p').innerText = this.value;
      let elem_rename = this;
      document.querySelectorAll('.planet_content').forEach(function(elem, i){
        if(elem_rename.parentElement.parentElement == elem){
          arr_p[i+1].name = elem_rename.value.substring(0,4);
        }
      });
```

```
   //Update planets in the menu on the left set the action when clicked
   addPanelPlanet();
  }
  function namePlanet(){
   document.querySelectorAll('.planet_name').forEach(function(elem, i){
    elem.addEventListener('blur', renamePlanetH1);
   });
  }
  namePlanet();

  //Hiding Planets Function
  function hidePlanetFunc(){
   let hideElem = this.parentElement.querySelector('.planet_content');
   hideElem.classList.toggle('hide');

this.parentElement.parentElement.querySelectorAll('.planet_content').forEach(functi
on(elem, i){
    if(hideElem != elem){
     elem.classList.add('hide');
    }
   });
  }
  function hidePlanet(){
   document.querySelectorAll('.planet_name_h1').forEach(function(elem, i){
    elem.addEventListener('click', hidePlanetFunc);
   });
  }
  hidePlanet();

  //Delete Planet Function
  function deletePlanetFunc(){
   this.parentElement.parentElement.parentElement.parentElement.remove();
   startFunc(1);
   addPanelPlanet();
  }
  function deletePlanet(){
   document.querySelectorAll('.delete_planet').forEach(function(elem, i){
    elem.addEventListener('click', deletePlanetFunc);
   });
  }
  deletePlanet();
  //Change Planet color and location
  function firstDisplayFunc(){
   startFunc(1);
  }
  function firstDisplay(){
   document.querySelectorAll('.planet_rad').forEach(function(elem, i){
    elem.addEventListener('blur', firstDisplayFunc);
   });
   document.querySelectorAll('.sputnik_rad').forEach(function(elem, i){
    elem.addEventListener('blur', firstDisplayFunc);
```

```
      });
      document.querySelectorAll('.planet_color').forEach(function(elem, i){
        elem.addEventListener('blur', firstDisplayFunc);
      });
    }
    document.querySelector('.star_color').addEventListener('blur', 
firstDisplayFunc);
    firstDisplay();
    firstDisplayFunc();
    //Planet Colors
    function colorPlanetFunc(){
      
this.parentElement.parentElement.parentElement.querySelector('.planet_name_h1').set
Attribute('style', 'background: ' + this.value + '66;');
      let elem_recolor = this;
      document.querySelectorAll('.planet_content').forEach(function(elem, i){
        if(elem_recolor.parentElement.parentElement == elem){
          
document.querySelectorAll('.control_panel_1 .planet')[i+1].setAttribute('style', 
'background: '+elem_recolor.value+';');
        }
      });
    }
    function colorPlanet(){
      document.querySelectorAll('.planet_color').forEach(function(elem, i){
        elem.addEventListener('blur', colorPlanetFunc);
      });
    }
    colorPlanet();
    //Universal function for popping out window
    document.querySelectorAll('.show').forEach(function(elem, i){
      elem.addEventListener('click', function(){
        document.querySelector('.bg').classList.remove('hide');
        document.querySelector('.' + elem.dataset.show).classList.remove('hide');
      });
    });
    function hide_window(){
      document.querySelector('.bg').classList.add('hide');
      document.querySelectorAll('.show_me').forEach(function(elem, i){
        elem.classList.add('hide');
      });
    }
    document.querySelector('.bg').addEventListener('click', hide_window);

    //Add Asteroid
    //Проосходит при нажании на кнопку добавить в появившемся окне
    document.querySelector('.add_body_settings').addEventListener('click', 
function(){
      //скрываем окно
      hide_window();
      b = {};
```

```
   //выводим подсказку и кнопку отмены
   document.querySelector('.message').innerHTML = 'State the asteroid location
<a href="" class="cancel">Cancel</a>';
   //отключаем кнопку добавления астеройдов
   document.querySelector('.add_body').disabled = true;
   //считываем и задаем масса, скорость и цвет
   b.m = Number(document.querySelector('.body_mass').value.replace( /\s/g, ""));
   b.v = Number(document.querySelector('.body_v').value.replace( /\s/g, ""));
   b.color = document.querySelector('.body_color').value;
   //Function after first click, getting the position and rendering the asteroid
at that positionx
   function firstClickNewBody(elem){
    //Finding asteroid position
    b.vx1 = elem.offsetX;
    b.vy1 = elem.offsetY;
    b.x = (elem.target.width/elem.target.clientWidth*elem.offsetX - start.x) *
(map.x / w);
    b.y = (elem.target.height/elem.target.clientHeight*elem.offsetY - start.y)
* (map.y / h);
    let r_arc = mappingPlanet(b);
    //drawing asteroid
    ctx.fillStyle = b.color;
    ctx.beginPath();
    ctx.arc(elem.target.width/elem.target.clientWidth*elem.offsetX,
elem.target.height/elem.target.clientHeight*elem.offsetY,r_arc,0,Math.PI*2,true);
    ctx.fill();
    //выводим сообщение и кнопку отмены
    document.querySelector('.message').innerHTML = 'State the direction <a
href="" class="cancel">Cancel</a>';
    //отменяем событие первого клика и назначаем событие второго клика
    document.querySelector('.trajectory').removeEventListener('click',
firstClickNewBody);
    document.querySelector('.trajectory').addEventListener('click',
secondClickNewBody);
    //result of clicking cancel on first action
    document.querySelector('.cancel').addEventListener('click', function(e){
     document.querySelector('.message').innerHTML = '';
     document.querySelector('.add_body').disabled = false;
     document.querySelector('.trajectory').removeEventListener('click',
secondClickNewBody);
     e.preventDefault();
    });
   }
   //second mouse click on canvas = asteroid velocity direction
   function secondClickNewBody(elem2){
    b.vx2 = elem2.offsetX;
    b.vy2 = elem2.offsetY;
    let v = Math.sqrt(Math.pow(b.vx2 - b.vx1, 2) + Math.pow(b.vy2 - b.vy1, 2));
//The distance formula between 2 dots
    let vx = b.vx2 - b.vx1;
    let vy = b.vy2 - b.vy1;
```

```
        sin = vy/v;
        cos = vx/v;
        b.vx = b.v * cos;
        b.vy = b.v * sin;
        b.name = 'Asteroid';
        delete b.vx1;
        delete b.vx2;
        delete b.vy1;
        delete b.vy2;
        //Deleting the second click action and deleting the message
        document.querySelector('.trajectory').removeEventListener('click',
secondClickNewBody);
        document.querySelector('.message').innerHTML = '';
        //turning the button of the asteroid on
        document.querySelector('.add_body').disabled = false;
        console.log(b);
        //adding asteroid to the asteroid array
        arr_b.push(b);
        //Creating copy of the asteroid and adding it to the celestial body list
        let b_temp = { ...b };
        arr_p.push(b_temp);
        //rereshing the planet menu on the left side of the screen
        addPanelPlanet();
      }
      //clicking action result
      document.querySelector('.trajectory').addEventListener('click',
firstClickNewBody);
      document.querySelector('.cancel').addEventListener('click', function(e){
        document.querySelector('.message').innerHTML = '';
        document.querySelector('.add_body').disabled = false;
        document.querySelector('.trajectory').removeEventListener('click',
firstClickNewBody);
        e.preventDefault();
      });
    });
    //scrolling mouse wheel = canvas result
    document.querySelector('.trajectory').addEventListener('wheel',
function(event){
        if(event.deltaY < 0) {
          map.x /= map_dm;
          map.y /= map_dm;
          map_d += map_dp;
          firstDisplayFunc();
        } else if(event.deltaY > 0) {
          map.x *= map_dm;
          map.y *= map_dm;
          map_d -= map_dp;
          firstDisplayFunc();
        }
        event.preventDefault();
    }, false);
    //Adding planets to the panel on the left
    function addPanelPlanet(){
      document.querySelectorAll('.control_panel_1 .planet').forEach(function(elem,
i){
        elem.parentElement.removeChild(elem);
      });
      arr_p.forEach(function(elem, i){
        let planet = document.createElement('button');
        planet.classList.add('planet');
        planet.setAttribute('data-n', i);
        planet.setAttribute('style', 'background: '+elem.color+';');
        planet.innerText = elem.name.substring(0,4);
        document.querySelector('.control_panel_1').appendChild(planet);
      });
      //Adding the action of clicking the planet on panel on the left
      document.querySelectorAll('.control_panel_1 .planet').forEach(function(elem,
i){
        elem.addEventListener('click', function(){
          //Showing which planet is rendered/displayed in the center
          start.n = this.getAttribute('data-n');
          //Drawing the changes
          startFunc(1);
        });
      });
    }
    addPanelPlanet();
    //Block start & + buttons untill Asteroid creation is done or Cancelled
    document.querySelector('.step_range').addEventListener('input', function(){
      let val = this.value;
      //timer 10мс s     m    h   d    week  month  70d    1/2year  1y      2y
5y      10y
      let arr_range = [0.01, 0.6, 36, 864, 6048, 25920, 60480, 157248, 315360,
630720, 1576800, 3153600];
      document.querySelector('.step').value = arr_range[val];
      //"not accurate" Message
      if(val >= 7){
        no_right = 1;
      } else {
        no_right = 0;
      }
      time_compare();
    });
</script>
</html>
```