

Финальное задание

Выполнили: студенты гр. 5.406-2, Ланетин Сергей Алексеевич, Тихомирова Злата Вячеславна

02/04/2025

Задание

Смоделировать Солнечную систему, используя JavaScript.

Примечание

Отображение программы должно быть в браузере. На выходе должна быть одна HTML-страница (желательно).

План работы

1. Спроектировать и реализовать информационную систему, в которой отображается информация по солнечной системе.
 - a. проектирование элементов системы;
 - b. экранные формы.

Требования

1. Сделать вывод информации, которая выводится при выборе планеты:
 - a. масса;
 - b. окружность;
 - c. газовый состав.
2. Реализовать движение планет по эллиптическим орбитам.
3. Учесть, что все планеты, кроме Венеры, вращаются по часовой стрелке.
4. Учесть, что скорость вращения планет вокруг Солнца высчитывается от наибольшей скорости планет — от скорости Юпитера.
5. Реализовать механизм изменения скорости. Скорость должна изменяться ползунком или +/-.
6. Сделать между Марсом и Юпитером пояс астероидов.
7. Ведение разработки с помощью Git.

Ход работы

Проектирование ИС

Чтобы спроектировать систему, нужно было учесть требования, описанные в плане работы.

При разработке информационной системы мы пользовались Википедией и следующей таблицей:

Масштаб

	<i>Радиус(км)</i>	<i>Расстояние от Солнца (км)</i>	<i>k</i>	<i>Радиус (в масштабе) (мм)</i>	<i>Расстояние от Солнца (в масштабе)(м)</i>
<i>Солнце</i>	700000	0	$2 \cdot 10^{10}$	35	0
<i>Меркурий</i>	2439,5	$58 \cdot 10^6$	$2 \cdot 10^{10}$	0,12197	2,9
<i>Венера</i>	6050	$108 \cdot 10^6$	$2 \cdot 10^{10}$	0,3025	5,4
<i>Земля</i>	6400	$150 \cdot 10^6$	$2 \cdot 10^{10}$	0,32	7,5
<i>Марс</i>	3390	$228 \cdot 10^6$	$2 \cdot 10^{10}$	0,1695	11,4
<i>Юпитер</i>	66854	$778 \cdot 10^6$	$2 \cdot 10^{10}$	3,3432	38,9
<i>Сатурн</i>	58232	$1429 \cdot 10^6$	$2 \cdot 10^{10}$	2,9116	71,45
<i>Уран</i>	25362	$2875 \cdot 10^6$	$2 \cdot 10^{10}$	1,2681	143,75
<i>Нептун</i>	24622	$4500 \cdot 10^6$	$2 \cdot 10^{10}$	1,2311	225

Рис. 1: Таблица с измерениями для планет.

Вывод информации о системе будет в виде абзаца на английском языке.

Форма для записи информации:

<Название планеты>:

Radius: <Радиус планеты>

Mass: <Масса планеты>

Composition: <Газовый состав>

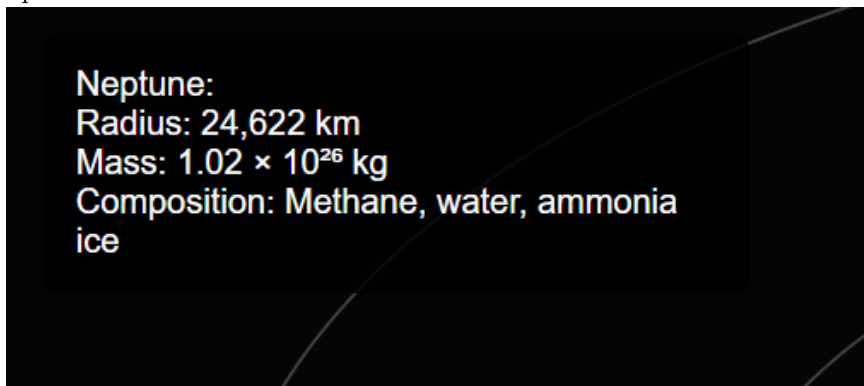


Рис. 2: Пример: информация о Нептуне.

Планеты, информацию о которых надо выводить: Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune. Планеты приведены в порядке возрастания удаленности от Солнца.

Чтобы реализовать движение планет по эллиптическим орбитам, нужно было создать планеты и эллиптические орбиты, после чего написать функцию для их движения `animatePlanets`.

Для планет важны следующие данные:

- название планеты;
- скорость планеты (относительно Юпитера);
- размер (задан в относительных единицах, получился как радиус планеты по отношению к радиусу Меркурия).

Для каждой эллиптической кривой сделаем эллипс с параметрами. Например, `gx` — среднее расстояние от планеты до Солнца в млн километров.

```
<ellipse id="mercuryOrbit" rx="58" ry="29" fill="none" stroke="#444" stroke-width="0.5"/>
```

Этот код создаёт эллипс в формате SVG (Scalable Vector Graphics). `id="mercuryOrbit"` присваивает эллипсу уникальный идентификатор `"mercuryOrbit"`, что позволяет обращаться к нему с помощью JavaScript. `rx="58"` и `ry="29"` задают радиусы эллипса по горизонтали и вертикали соответственно, определяя его форму и размер. `fill="none"` указывает, что эллипс не будет закрашен, а `stroke="#444"` и `stroke-width="0.5"` задают цвет обводки (тёмно-серый) и её толщину.

Затем для каждой планеты в соответствие ставится значение `gx` орбиты, скорость планеты по отношению к скорости Юпитера, а также `angle` — начальный угол вращения планеты, `rotation` — переменная-индикатор: если равна 0, то планета вращается по часовой стрелке, если 1, то против.

Например, для Меркурия:

```
mercury: { orbit: 58, speed: 3.65, angle: 0, rotation: 0 }
```

Здесь же мы учитываем требование, что Венера вращается против часовой стрелки.

Перейдем к функции для анимации планет. Функция `animatePlanets` должна анимировать движение планет на экране. Она должна проходить по каждому объекту в массиве `planets`, вычисляя новые координаты каждой планеты на основе её скорости, угла и радиуса орбиты. Затем она должна обновлять атрибут `transform` элемента на странице, отвечающего за визуальное отображение планеты, соответствующим образом изменяя его положение и вращение. Наконец, функция должна использовать `requestAnimationFrame` для организации плавной анимации, вызывая себя повторно для отрисовки следующего кадра, поскольку мы реализуем наше решение на JavaScript.

Далее нужно было реализовать механизм изменения скорости. Скорость должна изменяться ползунком или `+/-`.

Мы написали код, который реализует интерактивное управление скоростью и масштабом элементов на SVG-холсте. Скорость изменяется с помощью ползунка (`speedControl`), значение которого присваивается переменной `speedMultiplier`, влияющей на скорость анимации (не показанную в коде). Масштабирование (зум) осуществляется колесом мыши или жестами на сенсорных устройствах, изменяя `zoomLevel` в пределах от 0.2x до 5x и применяя трансформацию `scale` к группе элементов (`zoomGroup`).

Кроме того, реализовано перетаскивание элементов мышью или пальцем, изменяющее значения `translateX` и `translateY`, также применяемые в трансформации `translate` к `zoomGroup`. Как и говорилось ранее, информация о планетах отображается в блоке `infoBox` по клику на соответствующий элемент.

Чтобы сделать между Марсом и Юпитером пояс астероидов мы написали код, генерирует и анимирует пояса астероидов и отдельные астероиды на SVG-холсте. Сначала создаются эллиптические пояса астероидов между орбитами Марса (на расстоянии 228) и Юпитера (778), с заданным промежутком

(spacing) и количеством астероидов в каждом поясе (asteroidsPerBelt). Каждый астероид получает случайный радиус, непрозрачность, начальный угол и угловую скорость. Затем запускается функция animateAsteroids, которая в цикле requestAnimationFrame обновляет позицию каждого астероида на его эллиптической орбите, используя speedMultiplier для управления скоростью анимации. Позиция вычисляется по формулам для эллипса, с учетом текущего угла, который изменяется с каждой итерацией.

Аналогично, функция animatePlanets анимирует планеты, также с учетом speedMultiplier.

Таким образом, экранная форма для системы одна. Поле с информацией о планете появляется, когда мы не неё нажимаем.

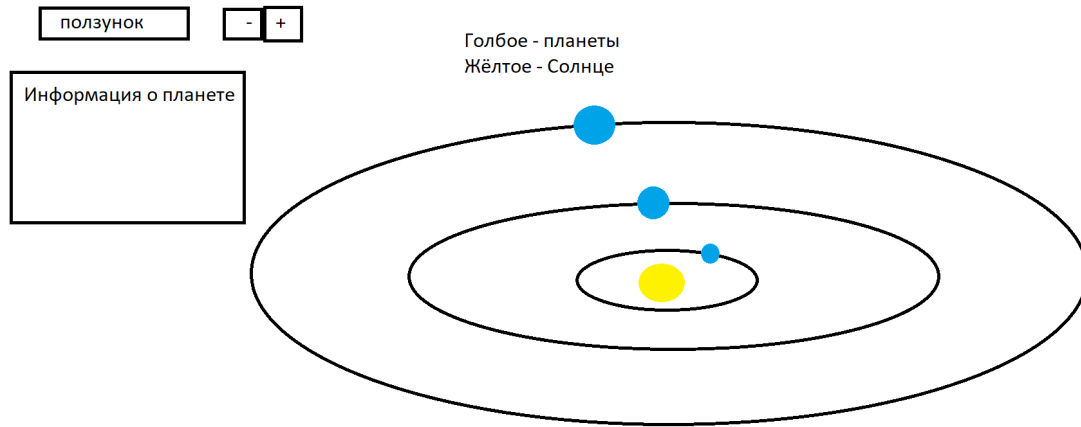


Рис. 3: Экранная форма для системы.

Разработку мы вели с использованием Git. Репозиторий: https://github.com/Dofcop/solar_system/tree/main

Реализация ИС

Ниже приведен скриншот, иллюстрирующий работу программы, написанной для реализации спроектированной системы.

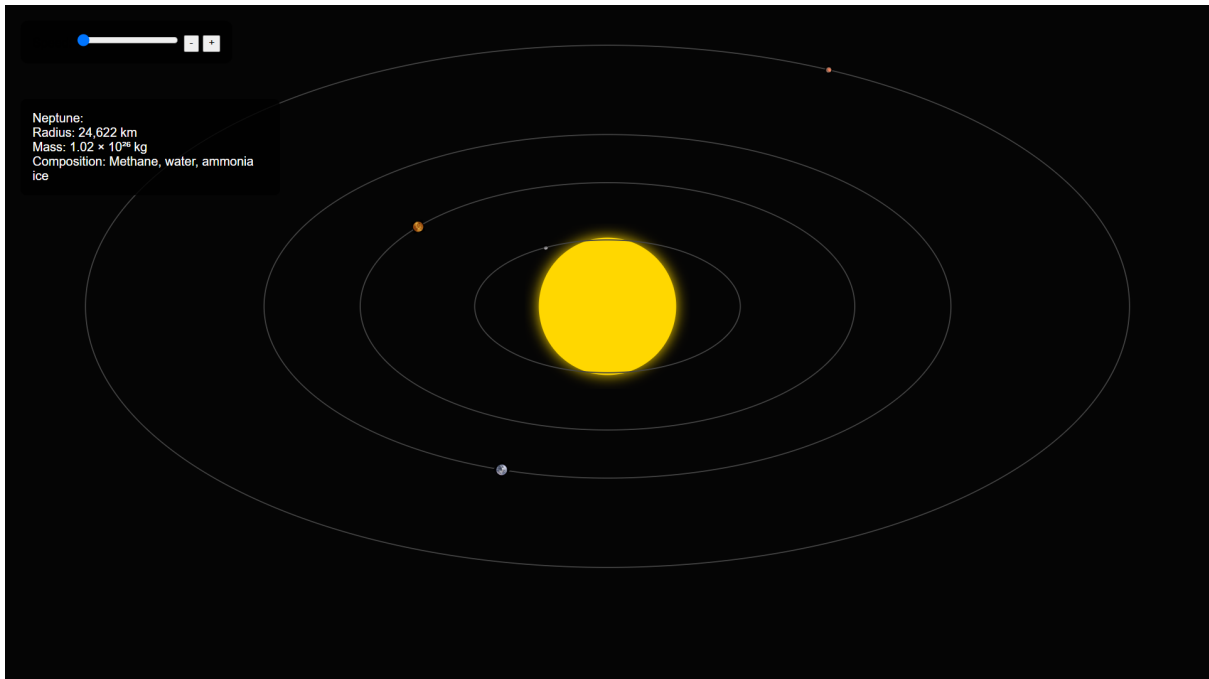


Рис. 4: Скриншот написанной программы.

Источники

1. Википедия. Меркурий. [\(https://ru.wikipedia.org/wiki/РѢРхГГРѣСГГРчРѣ_\(Р»Р»РѣР,,РхСГРѣРѣ\)\)](https://ru.wikipedia.org/wiki/РѢРхГГРѣСГГРчРѣ_(Р»Р»РѣР,,РхСГРѣРѣ)) (дата обращения: 28.03.2025)
2. Википедия. Венера. [\(https://ru.wikipedia.org/wiki/РЎРхР,,РхСГРѣРѣ_\(Р»Р»РѣР,,РхСГРѣРѣ\)\)](https://ru.wikipedia.org/wiki/РЎРхР,,РхСГРѣРѣ_(Р»Р»РѣР,,РхСГРѣРѣ)) (дата обращения: 28.03.2025)
3. Википедия. Земля. [\(https://ru.wikipedia.org/wiki/РЧРхРѣР»Р»CS\)](https://ru.wikipedia.org/wiki/РЧРхРѣР»Р»CS) (дата обращения: 28.03.2025)
4. Википедия. Марс. [\(https://ru.wikipedia.org/wiki/РѢРѣСГГРѣ_\(Р»Р»РѣР,,РхСГРѣРѣ\)\)](https://ru.wikipedia.org/wiki/РѢРѣСГГРѣ_(Р»Р»РѣР,,РхСГРѣРѣ)) (дата обращения: 28.03.2025)
5. Википедия. Юпитер. [\(https://ru.wikipedia.org/wiki/РнР»РчСГРхСГГ_\(Р»Р»РѣР,,РхСГРѣРѣ\)\)](https://ru.wikipedia.org/wiki/РнР»РчСГРхСГГ_(Р»Р»РѣР,,РхСГРѣРѣ)) (дата обращения: 28.03.2025)
6. Википедия. Сатурн. [\(https://ru.wikipedia.org/wiki/РгРѣСГГСГГР,,_\(Р»Р»РѣР,,РхСГРѣРѣ\)\)](https://ru.wikipedia.org/wiki/РгРѣСГГСГГР,,_(Р»Р»РѣР,,РхСГРѣРѣ)) (дата обращения: 28.03.2025)
7. Википедия. Уран. [\(https://ru.wikipedia.org/wiki/РнСГГРѣР,,_\(Р»Р»РѣР,,РхСГРѣРѣ\)\)](https://ru.wikipedia.org/wiki/РнСГГРѣР,,_(Р»Р»РѣР,,РхСГРѣРѣ)) (дата обращения: 28.03.2025)
8. Википедия. Нептун. [\(https://ru.wikipedia.org/wiki/РнРхР»СГГР,,_\(Р»Р»РѣР,,РхСГРѣРѣ\)\)](https://ru.wikipedia.org/wiki/РнРхР»СГГР,,_(Р»Р»РѣР,,РхСГРѣРѣ)) (дата обращения: 28.03.2025)