

大作业选题三实验报告（3-14组）

陈卓裕 2021012365 zhuoyu-c21@mails.tsinghua.edu.cn

葛宏泽 2020010968 gehz20@mails.tsinghua.edu.cn

侯士博 2021010778 housb21@mails.tsinghua.edu.cn

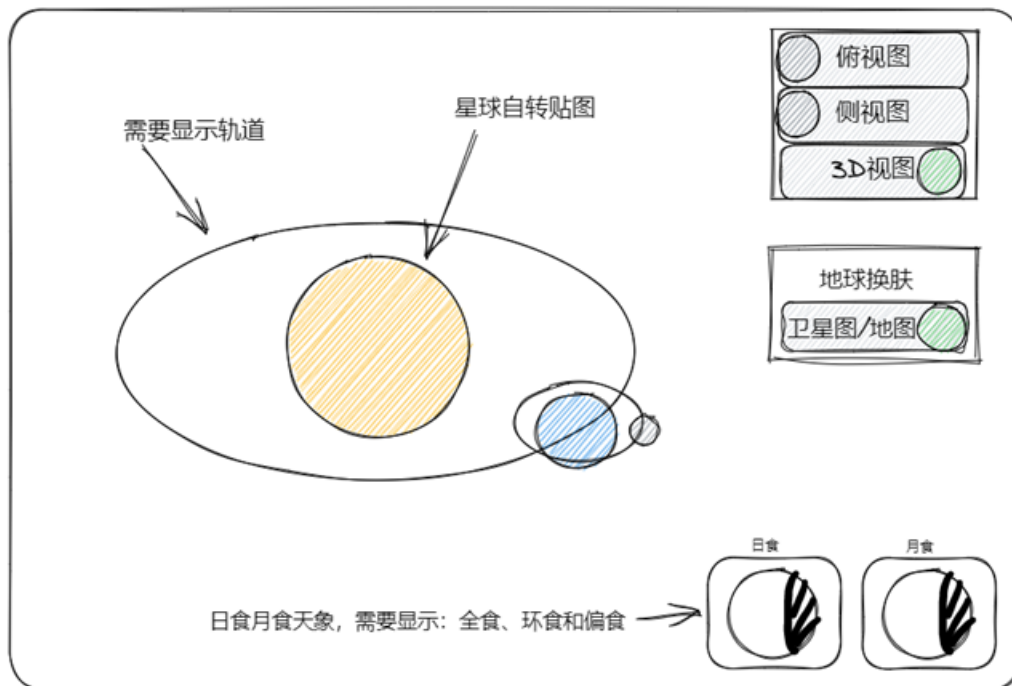
陈怡名 2021010855 c112358myc@gmail.com

<https://github.com/2023Web/orbit-simu>

实现思路

UI设计

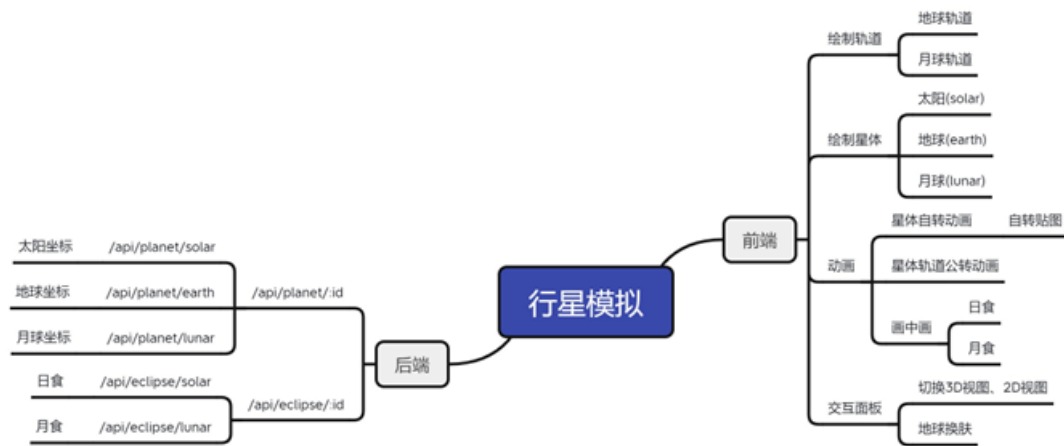
如下图，为所设计的UI：



其中，主要可以分为三个部分。左半侧为星球轨道的显示，右上侧为视图切换与材质切换，右下侧为日食月食显示。

程序框架

如下图，为所设计的程序框架：



Presented with xmind

本项目采用前后端分离进行项目推进。将前端分为绘制轨道、绘制星体、动画制作、交互面板四部分，其中动画分为星体自转、公转以及画中画实现日食月食部分，交互面板有切换视图和地球换肤功能。

前端部分侧重于星体运动的显示、日食月食显示与面板的绘制和事件绑定交互。使用CSS实现动画效果，并使用jQuery进行按钮的事件绑定，同时使用ajax进行星体运动数据从后端进行获取。

后端部分侧重于星体运动情况的计算，主要分为两个api。一个是planet，用于显示星体的运动参数，如自转周期、公转周期、星体半径、轨道半径等等。另一个是eclipse，用于显示日食和月食的实际程度。

日食月食动画计算

本部分，对日食月食的实际参数的实现算法，进行相应的说明。

后端计算

在后端代码中，我们定义了两个函数来计算太阳日食和月食的动画持续时间：

1. `calculateSolarEclipseRate(earth, solar, lunar)`：用于计算太阳日食的动画持续时间。
2. `calculateLunarEclipseRate(earth, lunar)`：用于计算月食的动画持续时间。

这两个函数分别接收地球、太阳和月亮的参数，并根据公式计算出太阳和月亮的日食动画持续时间（`eclipse_rate_solar` 和 `eclipse_rate_lunar`）。其中，太阳和月亮的半径、地球的公转半径和角度在函数内部进行计算。

同时，在后端代码中，我们使用了Express框架来创建API端点，用于获取太阳和月亮日食动画持续时间的数据。具体地，我们使用了以下路由：

- `GET` 方式访问 `/api/eclipse/:id`：该端点接收一个参数 `:id`，用于标识是太阳日食还是月食。然后，根据参数选择对应的函数计算日食动画持续时间，并将结果以JSON格式返回给前端。

在处理请求时，为了实时更新动画持续时间，我们使用了 `setInterval` 函数，每100毫秒更新一次动画。同时，我们还设置了 `res.on("finish")` 事件来在请求结束时清除定时器。

前端渲染

前端代码主要使用了JavaScript和CSS来实现日食动画的更新效果。

1. `getEclipseRate(planet)`: 这是一个封装了Ajax请求的函数, 用于从后端获取太阳和月亮日食动画持续时间的数据。它接收一个参数 `planet`, 用于指示是太阳日食还是月食, 然后根据参数发起相应的请求, 并递归调用自身以实现定时更新。获取数据后, 它还会调用 `updateAnimationDuration` 函数更新CSS动画的持续时间。
2. `updateAnimationDuration(eclipseRate, planet)`: 这个函数用于根据获取的日食动画持续时间更新CSS动画的持续时间, 使动画能够实时更新。
3. CSS样式: 在CSS样式中, 我们定义了 `.eclipse` 类用于设置日食图形的基本样式, 以及 `.eclipse-after-solar`、`.eclipse-before-solar`、`.eclipse-after-lunar` 和 `.eclipse-before-lunar` 类来设置不同动画阶段的样式。动画的持续时间通过CSS变量 `--animation-duration-planet` 来控制, 其中 `planet` 会在前端代码中根据日食类型进行替换。

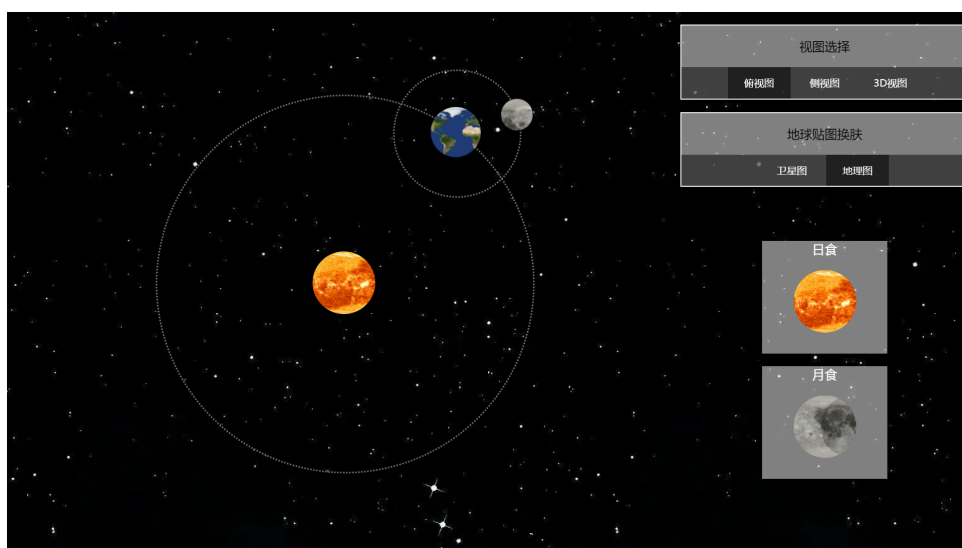
小组分工

- 陈卓裕: 后端星体坐标计算、各个视图的相互转换、地球换肤、交互面板的事件绑定与处理。
- 葛宏泽: 星体运动的贴图、实现星体自转动画、实现星体的椭圆轨道运动。
- 侯士博: 进行数据交互、设计星体的各个轨道参数。
- 陈怡名: 后端日食月食数据处理与计算、前端日食月食的动画渲染与显示。

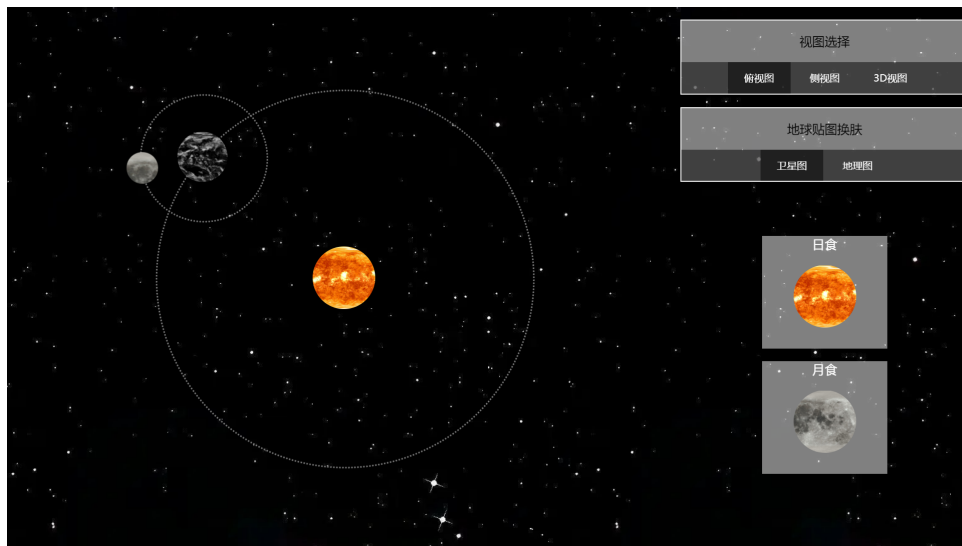
使用说明

首先需要下载并安装 `Node.js` 和 `npm`, 并确保在 `src` 文件夹下安装了 `express` 依赖, 即 `npm install express`。之后可以键入 `node app.js`, 运行并访问网址端口 `http://localhost:8080`, 进行查询。前端代码会自动从后端获取太阳和月亮日食动画持续时间, 并实时更新CSS动画的持续时间, 从而展示出日食的动画效果。

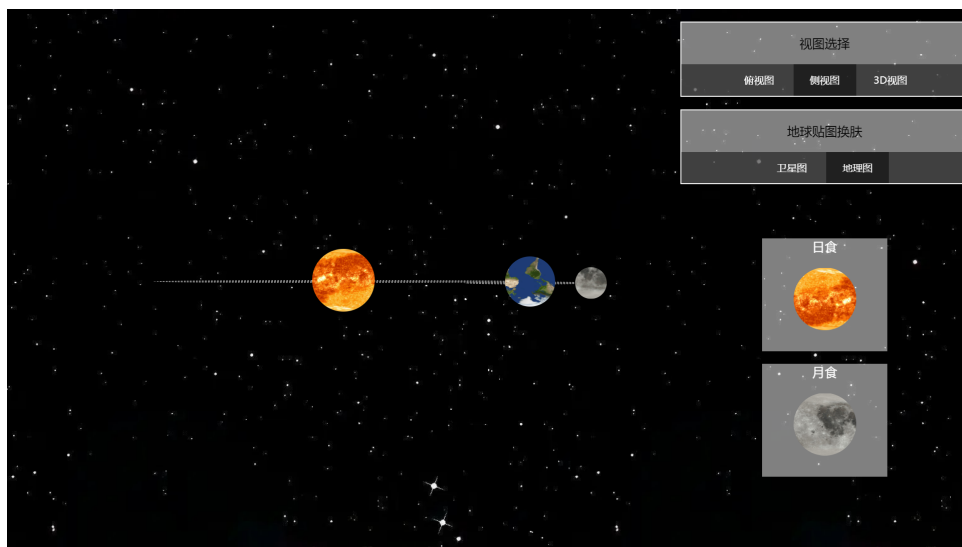
进入主界面后, 默认显示俯视图、地理图状态, 效果如下:



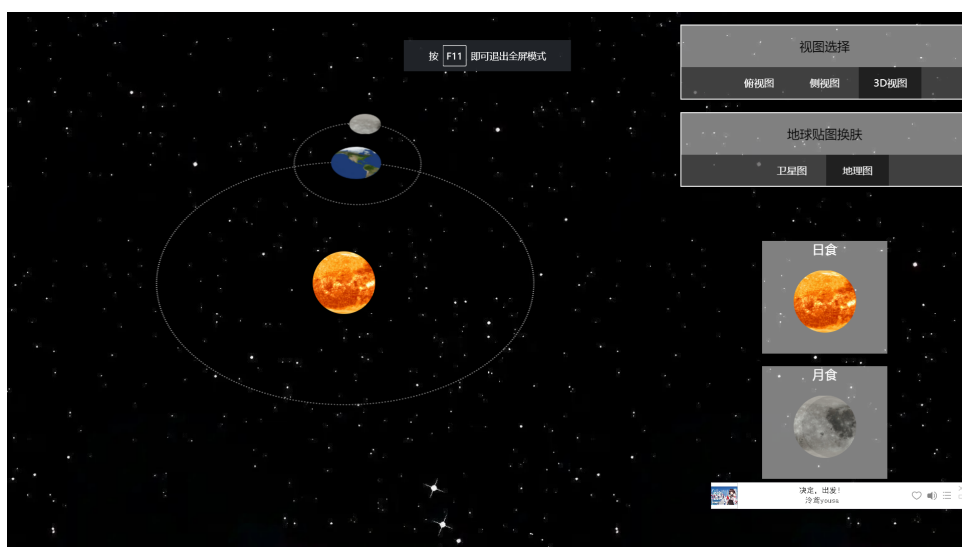
通过选择地理图或卫星图实现地球换肤。点击“卫星图”, 地球材质会切换为卫星图材质:



通过点击右侧栏的侧视图和3D视图可转换相应视角，俯视图在上文已经展示，侧视图用于显示各个星体之间的距离关系与轨道长度关系，因此动画只需要实现自转效果：

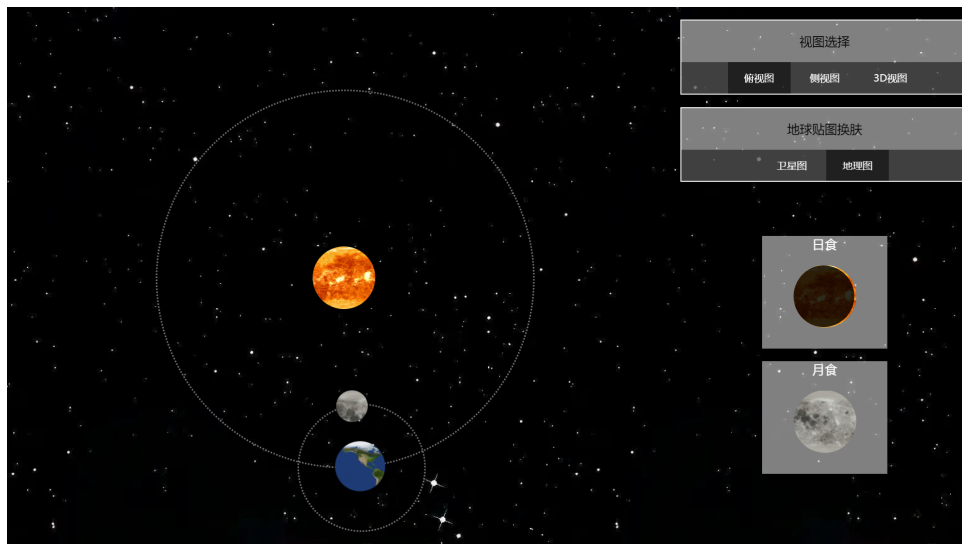


对于3D视图，其效果为：

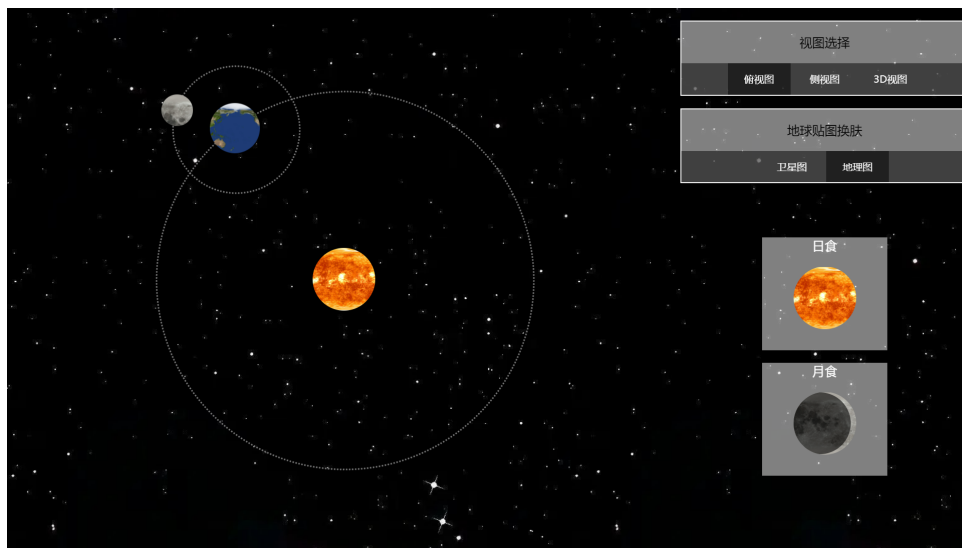


右下角会分别显示日食与月食效果，当日地月共线时，会分别显示不同的状态。并且随着时间的推移，会显示偏食等现象。

当月球夹在太阳和地球中间时，发生日食现象：



当地球加载太阳和月球中间时，发生月食现象：



遇到的问题与解决

- 地球自转容易卡顿。
 - 解决：设置关键帧 `0%{transform: translateX(0);}`;`100%{transform: translateX(-50%)}`；同时设置两张地球贴图以防止卡顿。
- CSS中设置星球半径等需要硬编码。
 - 解决：使用 `:root` 中的 `style`，将星球半径等参数作用与该类中，并使用 `var, calc` 等语法糖进行数据的访问与计算。

参考资料

- CSS实现图片无限循环无缝滚动：<https://blog.csdn.net/k464746/article/details/109290972>
- transform-style: preserve-3d [In CSS]：<https://blog.csdn.net/Cristiano2000/article/details/122388440>
- CSS3 transform-style 属性：<https://www.runoob.com/cssref/css3-pr-transform-style.html>
- css中的 `:root`：<https://blog.csdn.net/Wancc123/article/details/127227765>
- jQuery 参考手册 - Ajax：https://www.w3school.com.cn/jquery/jquery_ref_ajax.asp