



Блок  
Технологии



---

# Тестовое задание на позицию front-end разработчика

2019

Вокруг Земли по своим орбитам вращается несколько спутников (2-4) (бонус: сделать настраиваемым количество спутников в этом диапазоне). У каждого спутника своя фиксированная скорость движения по орбите  $V_s(n)$ , которая задаётся случайным образом при запуске программы. (бонус: сделать настраиваемой скорость каждого спутника и их количество). Кроме спутников вокруг Земли вращается Луна со скоростью  $V_m$  (бонус: сделать настраиваемой скорость движения по орбите у Луны).

Земля всё время работы приложения находится в центре экрана и статична (без имитации собственного вращения).

При старте приложения случайным образом задаётся положение спутников на орбите, положение Луны на орбите, скорости движения спутников и Луны. Запуск ракеты происходит при нажатии на кнопку «Запуск». Ракета начинает движение строго прямолинейно с фиксированной скоростью  $V_r$ .

При столкновении ракеты со спутником или Луной необходимо выводить alert (сообщение) – «Произошло столкновение». После нажатия на «ОК» перезапустить программу.

В случае, если ракета успешно долетела до конца экрана необходимо выводить alert (сообщение) – «Ракета успешно вышла в открытый космос».

При этом  $V_s(n) > V_r > V_m$

Для реализации приложения необходимо обязательно использовать: React JS, Webpack, ES6. UI компоненты взять из приложенного архива, дизайн приложения на следующем слайде (см. рис.1). Готовое решение необходимо разместить на github или codepen.

В случае, если проект будет на github: выкачивание зависимостей проекта, запуск браузера и старт приложения должны происходить по команде `npm run start`.

*Бонус: данные о начальном количестве спутников, скорости  $V_s$ ,  $V_r$ ,  $V_m$  присылать с сервера, поднятом на node js с использованием фреймворка express.*

Спутники - +

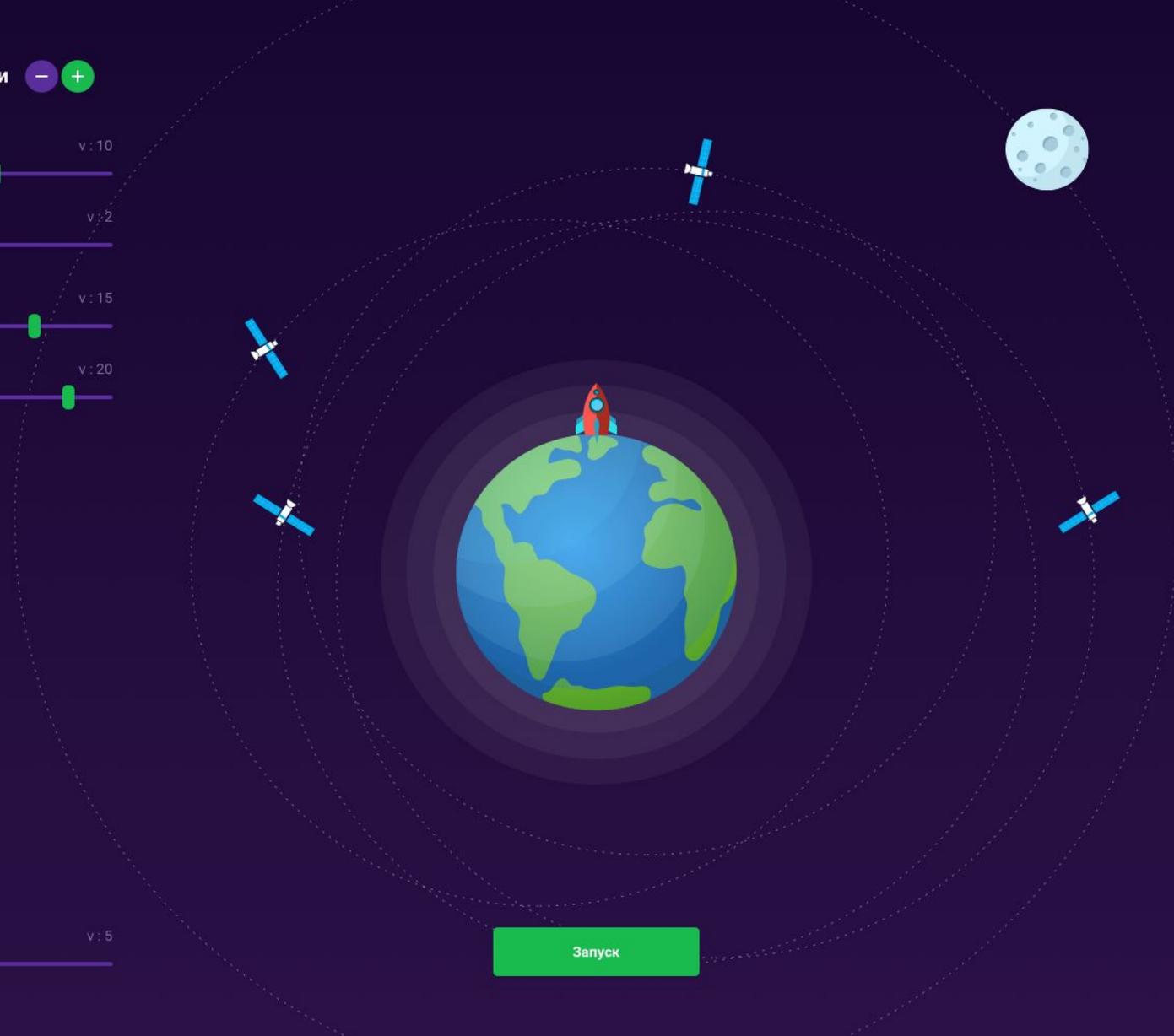
Спутник 1 v: 10

Спутник 2 v: 2

Спутник 3 v: 15

Спутник 4 v: 20

Луна v: 5



Запуск

Рис. 1 - пример дизайна тестового задания