



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

Институт информационных систем и
технологий

КАФЕДРА ИНФОРМАЦИОННЫХ
СИСТЕМ

Программирование специализированных вычислительных устройств

Отчет по лабораторной работе №4
«Индивидуальный творческий проект»

Выполнил студент гр. ИДБ-21-06
Проверил

Музафаров К.Р.
Лаверычев М.А.

Москва 2022г.

ОГЛАВЛЕНИЕ

Индивидуальное задание	3
Применяемые технологии	3
Описание проекта	3
Результат выполнения задания	5

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Цель: закрепление навыков работы с графическими библиотеками и фреймворками.

Задачи:

1. Согласовать индивидуальное творческое задание с преподавателем.
Индивидуальное задание должно отвечать следующим требованиям:
 - содержать от двух трехмерных объектов, связанных по смыслу;
 - включать элементы анимации, текстуры, освещения.
2. Используя знания, полученные на предыдущих лабораторных занятиях, написать программный код, реализующий согласованное задание.
3. Составить и защитить отчет по работе.

ПРИМЕНЯЕМЫЕ ТЕХНОЛОГИИ

Язык JavaScript и библиотека Three.js, и встроенные в нее функции, так же модели животных, созданные в Blender. Так же был использован Heroku для помещения на домен и выводение на сервер продакшн-кода.

ОПИСАНИЕ ПРОЕКТА

Был проанализирован код из официальной документации Three.js а так же из их GitHub, и взяты и в дальнейшем переработаны некоторые детали для выполнения задачи(пример света, пример обработчика, подключение файлов). Так же был создан каркас проекта.

Затем в Blender были нарисованы анимированные 3D объекты(животные) с сохранением реальных пропорций и размеров. За основу были взяты животные из Sketchfab, те модели, которые были изначально не подходили по стилистическим и техническим причинам.

После этого в каркас проекта и были добавлены файлы gltf. Смотреть Рис.1.

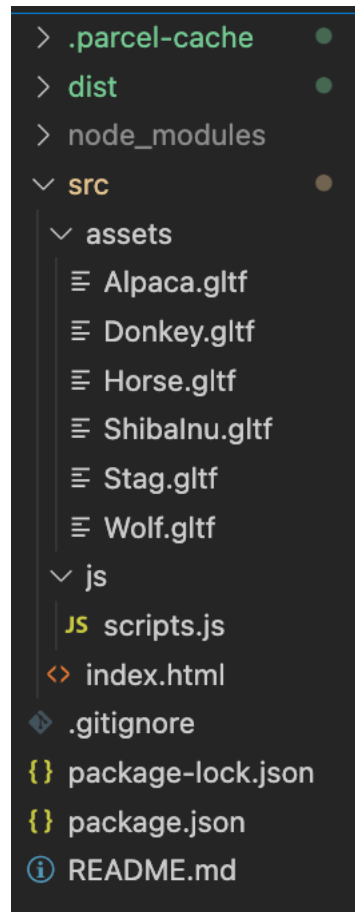


Рис.1. Файлы в каркасе проекта.

Так же по наставлению преподавателя был проанализирован код из Blender, и я выяснил как там задается позиционирование.

```

"accessors" : [
  {
    "bufferView" : 0,
    "componentType" : 5126,
    "count" : 2404,
    "max" : [
      0.6327688694000244,
      5.402582168579102,
      2.2075557708740234
    ],
    "min" : [
      -0.632768988609314,
      0.693108081817627,
      -1.731520652770996
    ],
    "type" : "VEC3"
  },
],

```

```

"matrix": [
  1.3,
  0.0,
  0.0,
  0.0,
  0.0,
  2.220446049250313e-16,
  1.3,
  0.0,
  0.0,
  1.3,
  2.220446049250313e-16,
  0.0,
  0.0,
  -0.85,
  -1.0,
  1.0
],

```

Рис.2,3. Матрицы позиционирования в разных каркасах Blender.

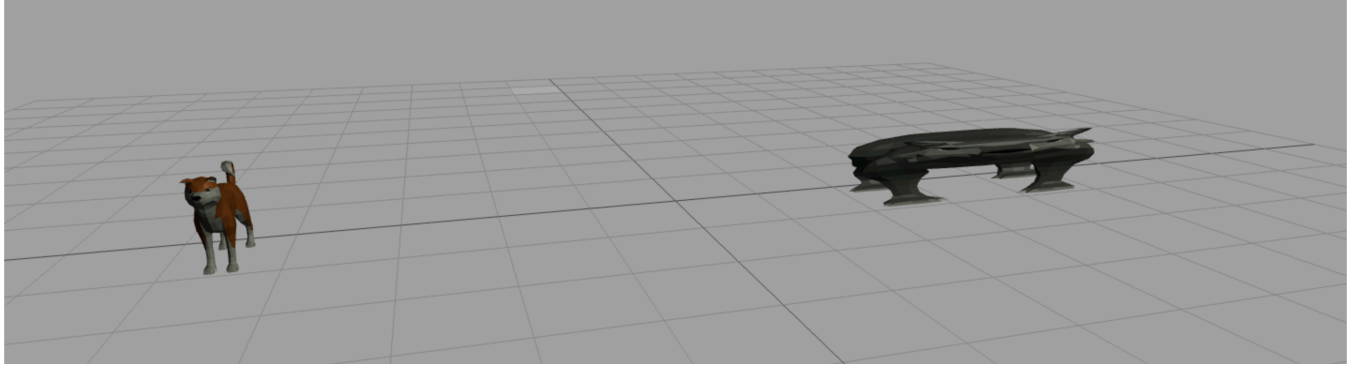


Рис.4. Пример некорректного позиционирования.

Импортированы необходимые библиотеки:

```
import * as THREE from 'three';
import {OrbitControls} from 'three/examples/jsm/controls/OrbitControls.js';
import {GLTFLoader} from 'three/examples/jsm/loaders/GLTFLoader.js';
import * as SkeletonUtils from 'three/examples/jsm/utils/SkeletonUtils.js';
```

Three - это основа проекта.

OrbitControls - для управления сценой.

GLTFLoader - для отображения моделей из Blender.

SkeletonUtils - для корректного отображения скелетной анимации.

Позиционирование камеры:

```
const camera = new THREE.PerspectiveCamera(
  45,
  window.innerWidth / window.innerHeight,
  0.1,
  1000
);
const orbit = new OrbitControls(camera, renderer.domElement);
camera.position.set(10, 6, 10);
```

Позиционирование света:

```
const ambientLight = new THREE.AmbientLight(0x333333);
scene.add(ambientLight);

const directionalLight = new THREE.DirectionalLight(0xFFFFFF, 1);
scene.add(directionalLight);
directionalLight.position.set(3, 3, 3);
```

Далее был написан код, который распределяет животных по точкам, выбранным пользователем. Запуск ведется через parcel.

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ЗАДАНИЯ

Можно самому посмотреть результат не устанавливая код.

На домене:

<https://animatanimas.herokuapp.com>

Так же код предоставлен в открытый доступ на GitHub:

<https://github.com/Karim90403/Anime-Forest>

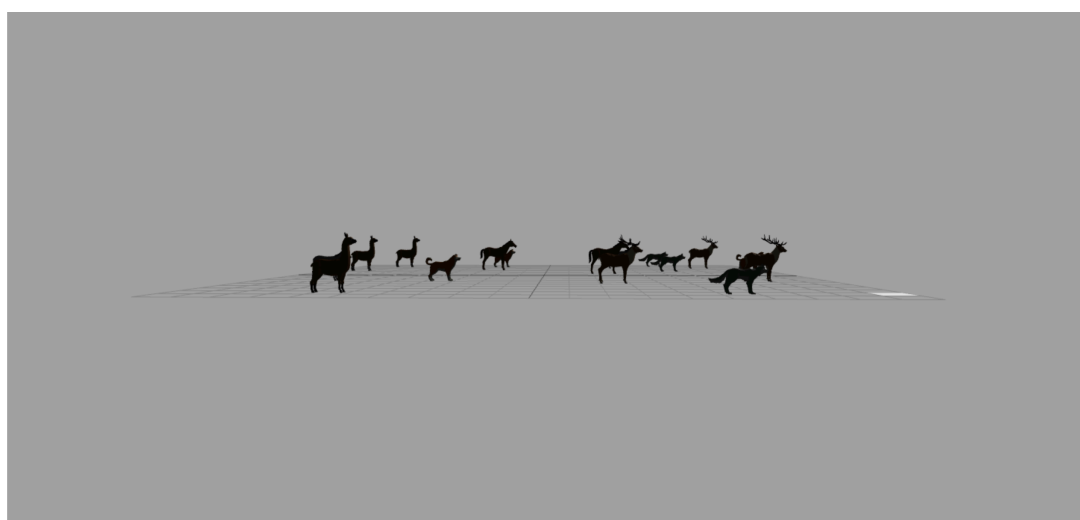
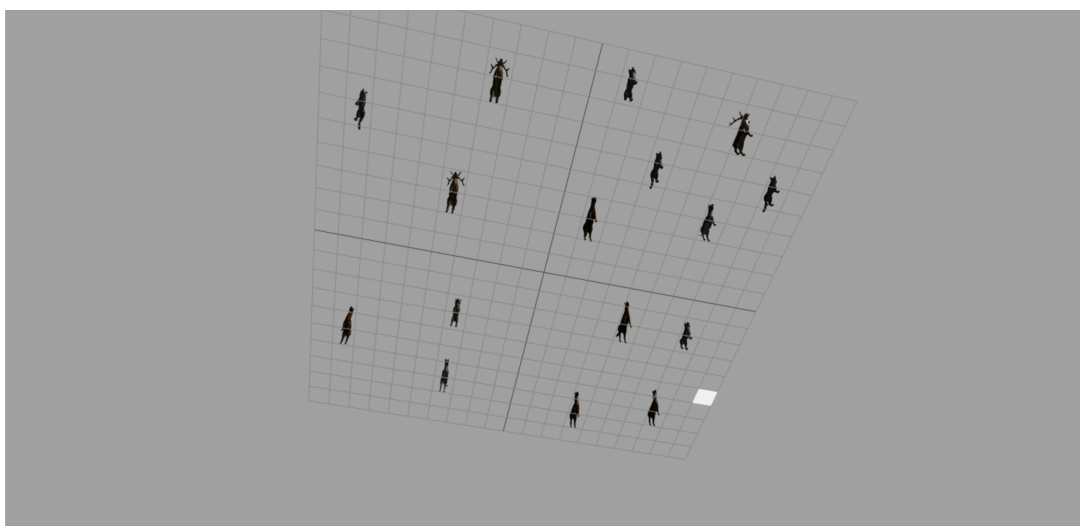
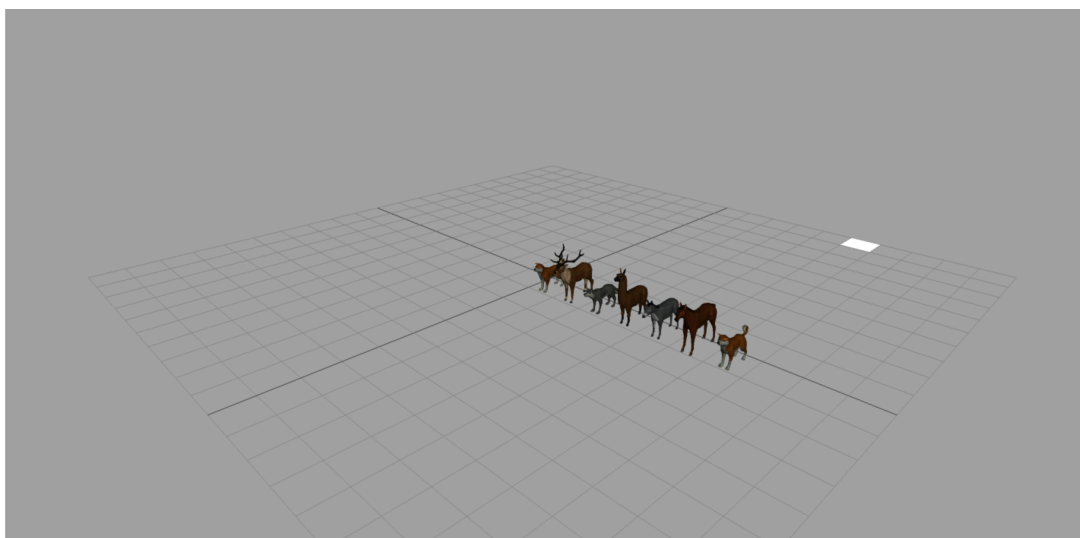


Рис.5-7. Результат выполнения кода