Практическая работа

[**Виртуальный колледж 1**](#_yvde9mrx8qoj)

[graph.js 1](#_30mcsrx971of)

[data.js 2](#_jmwh54c8q16g)

[game.js 3](#_ybdgt38c4sc6)

[update 5](#_od78oldxikl8)

[move 5](#_8hsmjp8jsuxs)

[loadImage 5](#_lpsdte66hg75)

[createButtonList 5](#_a4rn6l1yy91j)

# Виртуальный колледж

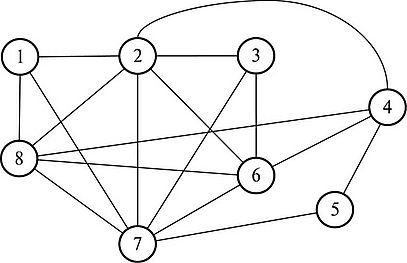
С помощью команды git clone склонируйте репозиторий с кодом([ссылка на репозиторий](https://github.com/zayarniy/virtual_college.git))

В репозитории две папки.

В одной папке лежит программа для создание сетки с фотографиями.

В другой папке игровой движок, позволяющий перемещаться по графу.

Графом называется математический объект определяющий связь между объектами.



## graph.js

Работа с графом реализована в скрипте graph.js

| class Graph {  constructor() {  this.adjacencyList = {};  }   addVertex(vertex) {  if (!this.adjacencyList[vertex]) {  this.adjacencyList[vertex] = [];  }  }   addEdge(v1, v2) {  if (this.adjacencyList[v1] && this.adjacencyList[v2]) {  this.adjacencyList[v1].push(v2); // добавляем v2 в список смежности v1  this.adjacencyList[v2].push(v1); // добавляем v1 в список смежности v2 (если граф неориентированный)  }  }   getNeighbors(vertex) {  return this.adjacencyList[vertex] || [];  } } |
| --- |

Этот скрипт позволяет добавлять вершины графа(addVertex), добавлять связи между вершинами(addEdge(v1, v2)) и получать соседей вершины (getNeighbors(vertex).

## data.js

Сам граф создается в модуле data.js

| const graph = new Graph();   graph.addVertex('00001'); graph.addVertex('00002'); graph.addVertex('00003'); graph.addVertex('00004'); graph.addVertex('00005'); graph.addVertex('00006'); graph.addVertex('00007'); graph.addVertex('00008'); graph.addVertex('00009'); graph.addVertex('000010'); graph.addVertex('000011'); graph.addVertex('000012'); graph.addVertex('000013'); graph.addVertex('000014'); graph.addVertex('000015'); graph.addVertex('000016');   graph.addEdge('00001', '00002'); graph.addEdge('00002', '00003'); graph.addEdge('00003', '00004'); graph.addEdge('00004', '00005'); graph.addEdge('00005', '00006'); graph.addEdge('00006', '00007'); graph.addEdge('00007', '00008'); graph.addEdge('00008', '00009'); graph.addEdge('00009', '00010'); graph.addEdge('00011', '00012'); graph.addEdge('00012', '00013'); graph.addEdge('00014', '00015'); graph.addEdge('00015', '00016');  console.log(graph.adjacencyList); |
| --- |

## game.js

Логика игры реализована в скрипте game.js

| const pathImages='../plan\_numerable/images/' let pos='00001'; let elementPosInfo=document.getElementById('posInfo'); let elementInputPos=document.getElementById('inputPos'); let elementImage=document.getElementById('image'); let elementListButtons=document.getElementById('listButtons')  function update() {  let s='Вы находитесь в позиции:'+pos+"<br>"+"Можно перейти в "+graph.getNeighbors(pos);  elementPosInfo.innerHTML=s;  loadImage(pos);  createButtonList(graph.getNeighbors(pos),elementListButtons); }  function move(newPos) {  if (newPos===undefined){  if (elementInputPos.value==='') return;  pos=elementInputPos.value;  }  else{  pos=newPos;  }  update();  }  function loadImage(imageName) {  elementImage.src=pathImages+String(imageName).padStart(5,'0')+'.jpg'; }  update();  function createButtonList(buttonLabels, targetElement) {  // Создаем элемент <ul> для списка кнопок  const ul = document.createElement('ul');  elementListButtons.innerHTML='';  // Проходим по массиву строк  buttonLabels.forEach(label => {  // Создаем элемент <li> для каждой кнопки  const li = document.createElement('li');   // Создаем кнопку  const button = document.createElement('button');  button.innerText = label;   // Добавление функциональности на кнопку (например, выводим alert)  button.addEventListener('click', () => {  move(label)  //alert(`Вы нажали на кнопку: ${label}`);  });   // Добавляем кнопку в <li>  li.appendChild(button);   // Добавляем <li> в <ul>  ul.appendChild(li);  });   // Добавляем <ul> в целевой элемент  targetElement.appendChild(ul); } |
| --- |

### update

Функция update реализует обновление данных на странице (получает соседей вершины, загружает картинку, обновляет кнопки для перемещения между вершинами)

### move

Функция move производит переход от одной вершины в другую (если это возможно) и вызывает обновление данных (update)

### loadImage

Функция loadImage - обновляет изображение на странице

### createButtonList

Функция creatButtonList - создает список кнопок с вершинами