**ВСТУП**

Уже давно Інтернет не є чимось новим: ми використовуємо його постійно. За останні роки ми були свідками якісного зростання можливостей ресурсів Мережі, в тому числі і засобів Web-програмування. У наш час стрімкого прогресу просто красиво оформлений текст і картинки на веб-сайті вже нікого не здивують, а стандарт WebGL який побудований на базі OpenGL здивував і не раз, він дозволяє розробникам веб контенту вбудовувати в веб-оглядачі які підтримують HTML5, повноцінну 3D-графіку, не вдаючись до використання плагінів.

В намірах розробників поширити WebGL не тільки в браузерах персональних комп'ютерів, а й у мобільних інтернет-пристроях

Ця технологія дозволяє упроваджувати апаратно-прискорену 3D графіку у веб-сторінки без необхідності використовувати спеціальні плагіни веб-браузера на будь-якій платформі, що підтримує [OpenGL](https://uk.wikipedia.org/wiki/OpenGL) або [OpenGL ES](https://uk.wikipedia.org/wiki/OpenGL_ES). Технічно це буде прив'язкою скриптів [JavaScript](https://uk.wikipedia.org/wiki/JavaScript) до функцій, визначених в бібліотеках OpenGL ES 2.0, реалізовану на рівні браузера.

WebGL є подальшим розвитком експерименту Canvas 3D в [Mozilla](https://uk.wikipedia.org/wiki/Mozilla) і вже представлена у збірках розробників Mozilla Firefox і [WebKit](https://uk.wikipedia.org/wiki/WebKit), а також в попередніх релізах [Google Chrome](https://uk.wikipedia.org/wiki/Google_Chrome) 4.

В листопаді 2009 компанія Khronos Group анонсувала першу чорнову специфікацію WebGL.[[](https://uk.wikipedia.org/wiki/WebGL#cite_note-7)Робота над специфікацією продовжується.

Про підтримку специфікації у своєму браузері [Chrome](https://uk.wikipedia.org/wiki/Google_Chrome), починаючи з версії 9, оголосив [Google](https://uk.wikipedia.org/wiki/Google). Підтримку WebGL у ближчих версіях продуктів оголосили [Mozilla](https://uk.wikipedia.org/wiki/Mozilla) (починаючи з [Firefox](https://uk.wikipedia.org/wiki/Firefox) 4) та [Apple Safari](https://uk.wikipedia.org/wiki/Apple_Safari).

WebGL широко підтримується у сучасних браузерах. Хоча можливість його використання залежить від інших факторів, а саме від [GPU](https://uk.wikipedia.org/wiki/GPU). Офіційний сайт WebGL пропонує просту тестову сторінку для перевірки на сумісність.[[](https://uk.wikipedia.org/wiki/WebGL#cite_note-WebGL_test_page-15)Більш детальна інформація (наприклад, те, який рендер використовує браузер, чи які розширення доступні) надається на сторонніх веб-сайтах

**РОЗДІЛ I. ТЕОРЕТИЧНА ЧАСТИНА**

**1.1 JavaScript**

**JavaScript**(**JS**)— динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту [ECMAScript](https://uk.wikipedia.org/wiki/ECMAScript). Найчастіше використовується як частина [браузера](https://uk.wikipedia.org/wiki/Браузер), що надає можливість коду на стороні [клієнта](https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура) (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, [асинхронно](https://uk.wikipedia.org/wiki/AJAX) обмінюватися даними з [сервером](https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура), змінювати [структуру](https://uk.wikipedia.org/wiki/Об'єктна_модель_документа) та [зовнішній вигляд](https://uk.wikipedia.org/wiki/Веб-дизайн) [веб-сторінки](https://uk.wikipedia.org/wiki/Веб-сторінка). Мова JavaScript також використовується для [програмування](https://uk.wikipedia.org/wiki/Програмування) на стороні сервера (подібно до таких мов програмування, як [Java](https://uk.wikipedia.org/wiki/Java) і [C#](https://uk.wikipedia.org/wiki/C_Sharp)), розробки [ігор](https://uk.wikipedia.org/wiki/Комп'ютерні_ігри), стаціонарних та мобільних додатків, сценаріїв в прикладному [ПЗ](https://uk.wikipedia.org/wiki/ПЗ) (наприклад, в програмах зі складу [Adobe Creative Suite](https://uk.wikipedia.org/wiki/Adobe_Creative_Suite)), всередині [PDF](https://uk.wikipedia.org/wiki/PDF)-документів тощо.

JavaScript класифікують як [прототипну](https://uk.wikipedia.org/wiki/Прототипне_програмування) (підмножина [об'єктно-орієнтованої](https://uk.wikipedia.org/wiki/Об'єктно-орієнтоване_програмування)), [скриптову мову](https://uk.wikipedia.org/wiki/Скриптова_мова) програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування ([імперативну](https://uk.wikipedia.org/wiki/Імперативне_програмування) та частково[функціональну](https://uk.wikipedia.org/wiki/Функціональне_програмування)) і деякі відповідні архітектурні властивості, зокрема:[динамічна](https://uk.wikipedia.org/wiki/Динамічна_типізація) та слабка типізація автоматичне керування пам'яттю, прототипне [наслідування](https://uk.wikipedia.org/wiki/Наслідування_(програмування)), функції як [об'єкти першого класу](https://uk.wikipedia.org/wiki/Об'єкт_першого_класу).

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну [семантику](https://uk.wikipedia.org/wiki/Семантика_мов_програмування), хоча й мають схожі риси в [стандартних бібліотеках](https://uk.wikipedia.org/w/index.php?title=Стандартна_бібліотека&action=edit&redlink=1) та правилах іменування. [Синтаксис](https://uk.wikipedia.org/wiki/Синтаксис_мови_програмування) обох мов отриманний «у спадок» від мови [С](https://uk.wikipedia.org/wiki/C_(мова_програмування)), але семантика та дизайн JavaScript є результатом впливу мов [Self](https://uk.wikipedia.org/w/index.php?title=Self_(мова_програмування)&action=edit&redlink=1) та [Scheme](https://uk.wikipedia.org/wiki/Scheme).

В [1995](https://uk.wikipedia.org/wiki/1995) році компанія [Netscape](https://uk.wikipedia.org/wiki/Netscape_Communications) поставила завдання вбудувати мову програмування [Scheme](https://uk.wikipedia.org/wiki/Scheme) чи «якусь схожу» в[браузер](https://uk.wikipedia.org/wiki/Браузер) [Netscape](https://uk.wikipedia.org/wiki/Netscape_Navigator). Для цього був запрошений [Брендан Айк](https://uk.wikipedia.org/wiki/Брендан_Айк), [американський](https://uk.wikipedia.org/wiki/Сполучені_Штати_Америки) розробник, що спеціалізувався на[системному програмуванні](https://uk.wikipedia.org/wiki/Системне_програмування). Також, для прискорення розробки, Netscape почали співробітництво з компанією[Sun Microsystems](https://uk.wikipedia.org/wiki/Sun_Microsystems).

З часом, концепція розроблюваної мови програмування була розширена до можливості використання безпосередньо в [HTML](https://uk.wikipedia.org/wiki/HTML)-коді [сторінки](https://uk.wikipedia.org/wiki/Веб-сторінка). Компанії мали на меті створити мову, що могла зв'язати різні частини веб-сайтів: зображень, [Java-аплетів](https://uk.wikipedia.org/wiki/Java-аплет), [об'єктної моделі документа](https://uk.wikipedia.org/wiki/Об'єктна_модель_документа). Ця мова повинна була стати зручною для веб-дизайнерів та некваліфікованих програмістів. Робочою назвою нової мови була Mocha, яка була змінена на LiveScript в перших двох [бета-версіях](https://uk.wikipedia.org/wiki/Бета-версія) браузера Netscape 2.0. А дещо пізніше, користуючись популярністю бренду Java, LiveScript був перейменований на JavaScript і третя бета-версія (2.0B3) [Netscape 2.0](https://uk.wikipedia.org/wiki/Netscape_Navigator) вже вийшла з сучасною назвою. Для цього була придбана відповідна ліцензія у компанії [Sun Microsystems](https://uk.wikipedia.org/wiki/Sun_Microsystems), що володіла [брендом](https://uk.wikipedia.org/wiki/Бренд) [Java](https://uk.wikipedia.org/wiki/Java).

В 1992 році компанією Nombas була розроблена скриптова мова програмування [Cmm](https://uk.wikipedia.org/w/index.php?title=Cmm&action=edit&redlink=1) ([англ](https://uk.wikipedia.org/wiki/Англійська_мова). С-minus-minus, гра слів навколо мови [С++](https://uk.wikipedia.org/wiki/C%2B%2B)), яка пізніше була перейменована на ScriptEase та могла вбудовується в веб-сторінки. Існує хибна думка, що JavaScript був створений під впливом Cmm. Насправді, Брендан Айк ніколи не чув про Cmm до того, як він створив LiveScript. Пізніше, Nombas зупинили розробку Cmm та почали використовувати JavaScript, а згодом брали участь у групі зі[стандартизації](https://uk.wikipedia.org/wiki/Стандартизація) JavaScript.

У листопаді [1996](https://uk.wikipedia.org/wiki/1996) року [Netscape](https://uk.wikipedia.org/wiki/Netscape_Communications) заявила, що відправила JavaScript в організацію [Ecma International](https://uk.wikipedia.org/wiki/Ecma_International) для розгляду мови як промислового [стандарту](https://uk.wikipedia.org/wiki/Стандартне_відхилення). В результаті подальшої роботи з'явилась стандартизована мова з назвою [ECMAScript](https://uk.wikipedia.org/wiki/ECMAScript). В червні 1997 року, Ecma International опублікувала першу редакцію специфікації [ECMA-262](https://uk.wikipedia.org/w/index.php?title=ECMA-262&action=edit&redlink=1). Рік по тому, в червні [1998](https://uk.wikipedia.org/wiki/1998) року, щоб адаптувати специфікацію до стандарту ISO/IEC-16262, були внесені деякі зміни і випущена друга редакція. Третя редакція побачила світ в грудні [1999](https://uk.wikipedia.org/wiki/1999) року.

Четверта версія стандарту ECMAScript так і не була закінчена і четверта редакція не вийшла[[10]](https://uk.wikipedia.org/wiki/JavaScript#cite_note-10). Тим не менш, п'ята редакція з'явилася в грудні 2009 року.

На сьогодні, актуальна версія стандарту — 5.1. Вона була випущена в червні [2011](https://uk.wikipedia.org/wiki/2011) року.

JavaScript, наразі, є однією з найпопулярніших мов програмування в [інтернеті](https://uk.wikipedia.org/wiki/Інтернет). Але спочатку багато професіональних програмістів скептично ставилися до мови, цільова аудиторія якої складалася з програмістів-любителів. Поява [AJAX](https://uk.wikipedia.org/wiki/AJAX) змінила ситуацію та повернула увагу професійної спільноти до мови. В результаті, були розроблені та покращені багато практик використання JavaScript (зокрема, [тестування](https://uk.wikipedia.org/wiki/Тестування_програмного_забезпечення) та [налагодження](https://uk.wikipedia.org/wiki/Налагодження_програми)), створені бібліотеки та [фреймворки](https://uk.wikipedia.org/wiki/Фреймворк), поширилося використання JavaScript поза браузером.

JavaScript має низку властивостей [об'єктно-орієнтованої мови](https://uk.wikipedia.org/wiki/Об'єктно-орієнтоване_програмування), але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов [ООП](https://uk.wikipedia.org/wiki/ООП). Крім того, JavaScript має ряд властивостей, притаманних [функціональним мовам](https://uk.wikipedia.org/wiki/Функціональне_програмування), — функції як [об'єкти першого класу](https://uk.wikipedia.org/wiki/Об'єкт_першого_класу), об'єкти як списки, каррінг ([currying](https://uk.wikipedia.org/w/index.php?title=Currying&action=edit&redlink=1)), [анонімні функції](https://uk.wikipedia.org/w/index.php?title=Анонімні_функції&action=edit&redlink=1),[замикання](https://uk.wikipedia.org/wiki/Замикання_(програмування)) (closures) — що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою [Сі](https://uk.wikipedia.org/wiki/C_(мова_програмування)) має такі корінні відмінності:

* об'єкти, з можливістю [інтроспекції](https://uk.wikipedia.org/wiki/Інтроспекція_(програмування)) і динамічної зміни типу через механізм [прототипів](https://uk.wikipedia.org/wiki/Прототипне_програмування)
* функції як об'єкти першого класу
* [обробка винятків](https://uk.wikipedia.org/wiki/Обробка_винятків)
* автоматичне приведення типів
* автоматичне [прибирання сміття](https://uk.wikipedia.org/wiki/Прибирання_сміття)
* анонімні функції

JavaScript містить декілька вбудованих об'єктів: **Global**, **Object**, **Error**, **Function**, **Array**, **String**, **Boolean**, **Number**, **Math**, **Date**, **RegExp**.

Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови [Java](https://uk.wikipedia.org/wiki/Java) (тобто, зрештою, успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї.

**1.2 THREE.js**

Three.js- легка кросбраузерна [бібліотека JavaScript](https://ru.wikipedia.org/wiki/Библиотека_JavaScript), що використовується для створення та відображення анімованої комп'ютерної 3D графіки при розробці [веб-додатків](https://ru.wikipedia.org/wiki/Веб-приложение). Three.js скрипти можуть використовуватися спільно з елементом HTML 5, Canvas, SVG або WebGL. Вихідний код розташований в репозиторії [GitHub](https://ru.wikipedia.org/wiki/GitHub).Three.js дозволяє створювати прискорену на [GPU](https://ru.wikipedia.org/wiki/GPU)3D графіку, використовуючи мову [JavaScript](https://ru.wikipedia.org/wiki/JavaScript)як частина [сайту](https://ru.wikipedia.org/wiki/Сайта)без підключення пропрієтарних плагінів для браузера. Це можливо завдяки використанню технології [WebGL](https://ru.wikipedia.org/wiki/WebGL). Підтримує тривимірні моделі формату [Collada](https://ru.wikipedia.org/wiki/Collada).

## **1.2.1 Особливості THREE.js**

* Рендерер: Canvas, SVG або WebGL
* Сцена: додавання і видалення об'єктів в режимі реального часу; туман
* Камери: перспективна або Ортографічна
* Анімація: каркаси, швидка кінематика, зворотна кінематика, покадрова анімація
* Джерела світла: зовнішній, спрямований, точечений; тіні: кинуті і отримані
* Шейдери: повний доступ до всіх OpenGL шейдерам ( [GLSL](https://ru.wikipedia.org/wiki/GLSL) )
* Об'єкти: мережі, частинки, спрайт, лінії, скелетна анімація і інше
* Геометрія: площину, куб, сфера, тор, 3D текст і інше; модифікатори: тканина, видавлювання
* Завантажники даних: двійковий, зображення, [JSON](https://ru.wikipedia.org/wiki/JSON) і сцена
* Експорт та імпорт: утиліти, що створюють Three.js-сумісні JSON файли з форматів: [Blender](https://ru.wikipedia.org/wiki/Blender) , [openCTM](https://ru.wikipedia.org/w/index.php?title=OpenCTM&action=edit&redlink=1) , [FBX](https://ru.wikipedia.org/wiki/FBX) , [3D Studio Max](https://ru.wikipedia.org/wiki/3D_Studio_Max) , і [Wavefront .obj файл](https://ru.wikipedia.org/w/index.php?title=Wavefront_.obj_файл&action=edit&redlink=1)
* Підтримка: документація по API бібліотеки знаходиться в процесі постійного розширення і доповнення, є публічний форум і велике співтовариство
* Приклади: на офіційному сайті можна знайти більше 150 прикладів роботи зі шрифтами, моделями, текстурами, звуком і іншими елементами сцени

Бібліотека Three.js працює у всіх браузерах, які підтримують технологію WebGL; також може працювати з «чистим» інтерфейсом елемента CANVAS , завдяки чому працює і на багатьох мобільних пристроях. Three.js поширюється під ліцензією [MIT license](https://ru.wikipedia.org/wiki/MIT_license) .

## **1.2.1 Використання** THREE.js

Бібліотека Three.js поставляється в одному JavaScript файлі, який може бути підключений до сторінці в будь-якому місці (рис.1.1).

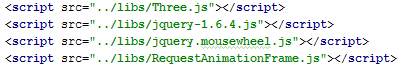


Рис.1.1 підключення бібліотеки на HTML сторінку

## **1.2.2 Критика  THREE.js**

Розробка редактора сцен для Three.js знаходиться на початковій стадії. Таким чином, створення навіть примітивного 3D контенту вимагає написання програмного коду. Як недоліки движка також називаються відсутність своєчасних оновлень документації та уроків.

**1.4 Node.js**

Node.js — платформа з [відкритим](https://uk.wikipedia.org/wiki/Політика_відкритого_коду) [кодом](https://uk.wikipedia.org/wiki/Сирцевий_код) для виконання високопродуктивних мережевих [застосунків](https://uk.wikipedia.org/wiki/Застосунок), написаних мовою [JavaScript](https://uk.wikipedia.org/wiki/JavaScript). Засновником платформи є [Раян Дал](https://uk.wikipedia.org/w/index.php?title=Раян_Дал&action=edit&redlink=1) (Ryan Dahl). Платформа node.js перетворила мову JavaScript, що в основному використовувалась в браузерах на мову загального використання з великою спільнотою розробників.

Node.js характеризується такими властивостями:

* асинхронна [однопотокова](https://uk.wikipedia.org/wiki/Нить) модель виконання запитів
* неблокуючий ввід/вивід
* система модулів [CommonJS](https://uk.wikipedia.org/wiki/CommonJS)
* [рушій JavaScript](https://uk.wikipedia.org/wiki/Рушій_JavaScript) [Google V8](https://uk.wikipedia.org/wiki/Google_V8)

Для управління модулями використовується пакетний менеджер [npm](https://uk.wikipedia.org/w/index.php?title=Npm&action=edit&redlink=1) (node package manager).

## **1.4.1 Огляд Node.js**

Node.js призначений для відокремленого виконання високопродуктивних мережних [застосунків](https://uk.wikipedia.org/wiki/Застосунок) на мові JavaScript. Функції платформи не обмежені створенням серверних скриптів для [веб](https://uk.wikipedia.org/wiki/Веб), платформа може використовуватися і для створення звичайних клієнтських і серверних мережевих програм. Для забезпечення виконання JavaScript-коду використовується розроблений компанією [Google](https://uk.wikipedia.org/wiki/Google) [рушій](https://uk.wikipedia.org/wiki/Рушій_JavaScript) [V8](https://uk.wikipedia.org/wiki/V8_(рушій_JavaScript)).

Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, засновану на обробці подій в неблокуючому режимі і визначенні обробників зворотніх викликів (callback). В якості способів мультиплексування з'єднань підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується [бібліотека](https://uk.wikipedia.org/wiki/Бібліотека_програм) libuv, для створення пулу [потоків](https://uk.wikipedia.org/wiki/Нить) (thread pool) задіяна бібліотека libeio, для виконання [DNS](https://uk.wikipedia.org/wiki/DNS)-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що спричиняють блокування, виконуються всередині пула потоків і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

**РОЗДІЛ II. ПРАКТИЧНА ЧАСТИНА**

**2.4 Керівництво користувача**

Заходимо в WebStorm ,відкриваємо папку з проектом, знаходимо файл index.html ПКМ на ньому “Run”(рис. 2.5)

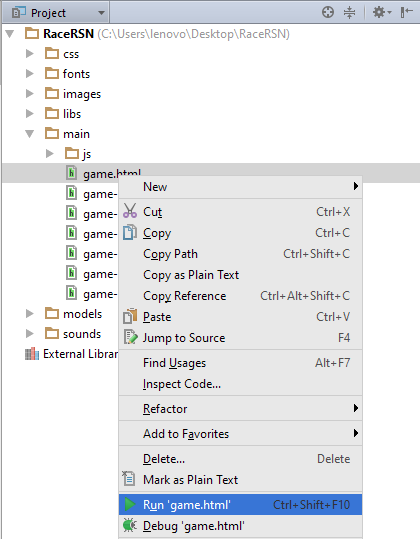


Рис. 2.5 Запуск

**2.4.2 Початковий екран**

Після відкриття index.html користувач потрапляє в браузер в якому WebGL рендерить Minecraft.(рис. 2.6)



Рис. 2.6 Початковий екран (вид від першої особи)

**2.4.3 Зміна виду**

Для зміни виду використовуйте комбінацію клавіш CTRL+R (рис. 2.7)

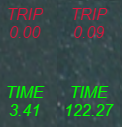


Рис. 2.7 Вид від третьої особи

**2.4.4 Переміщення персонажа**

Користуючись клавішами: W,A,S,D Ви можете рухатись по сцені, SPACE - стрибок.



**2.4.5 Вибір текстури блока**

Для вибору текстури натисніть клавішу ESC , виберіть користуючись мишкою потрібну текстуру в лівому верхньому куті (рис. 2.8)



Рис. 2.8 Вибір текстури

**2.4.6 Створення блока**

Для створення блока затисніть клавішу CTRL + ЛКМ (рис. 2.9)



Рис. 2.9 Створення блока

**2.4.7 Видалення блока**

Користуючись червоною крапкою в центрі монітора наведіться на потрібний блок, натисніть ПКМ або ЛКМ (рис. 2.10)

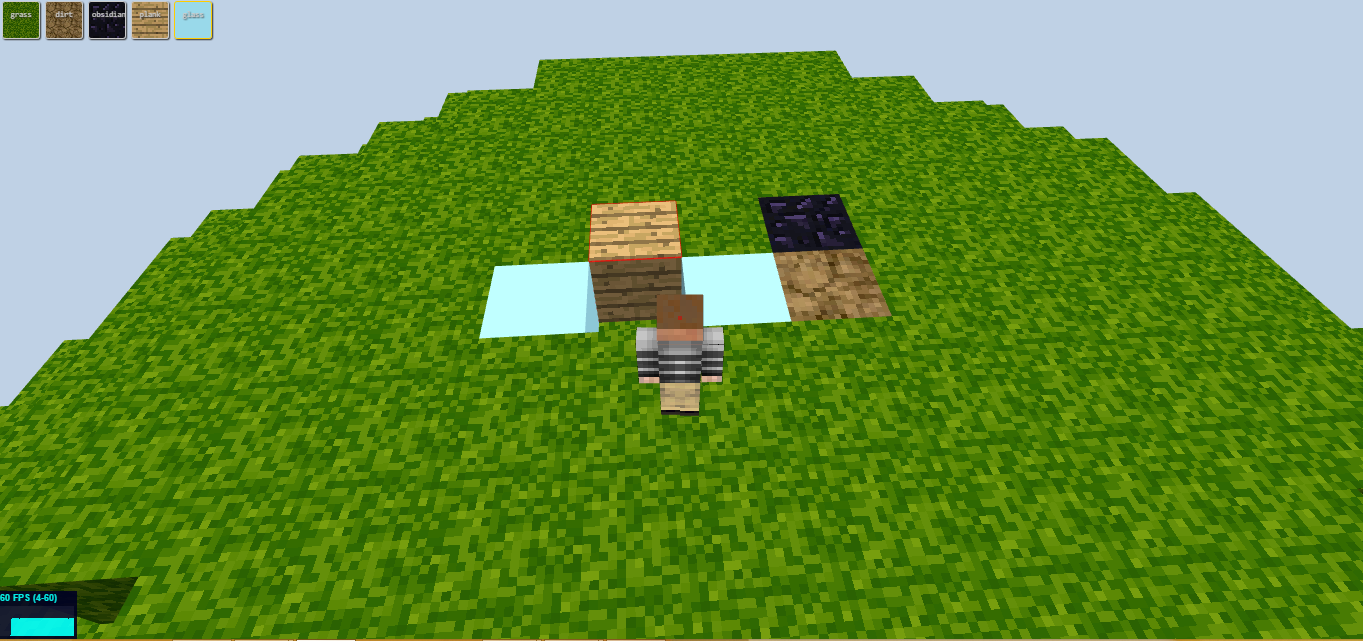


Рис. 2.10 Видалення блока

**Висновки**

Під час виконання цієї курсовї роботи я навчився стоворювати 3D моделі в браузері за допомогою мови програмування JavaScript та бібліотеки THREE.js.

Знайшов зручну для себе бібліотеку VOXEL.js ,за допомогою якої і був побудований даний програмний продукт.

Навчився використовувати контроль версій GIT ,компілятор

JavaScript «Browserify», додавати текстури 3D об'єктам, створювати свої власні текстури.

Розібрався як вибирати потрібну текстуру блока прямо в сцені гри.

Після завершення даної сцени гри, я дістав важливий досвід для себе, який я точно використаю в майбутньому.

**Список використаної літератури**

1. [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) <https://ru.wikipedia.org/wiki/JavaScript>
2. [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) <https://learn.javascript.ru/>
3. [Git](https://uk.wikipedia.org/wiki/Git) <https://uk.wikipedia.org/wiki/Git>
4. [Three.js](https://ru.wikipedia.org/wiki/Three.js) <https://ru.wikipedia.org/wiki/Three.js>
5. [Node.js](https://uk.wikipedia.org/wiki/Node.js) <https://uk.wikipedia.org/wiki/Node.js>
6. [Browserify](https://en.wikipedia.org/wiki/Browserify) <https://en.wikipedia.org/wiki/Browserify>
7. [Three.js](https://ru.wikipedia.org/wiki/Three.js) <http://threejs.org/>
8. Voxel.js <http://voxeljs.com/>
9. Node.js <http://nodejs.ru/>
10. NPM <https://www.npmjs.com/>
11. Browserify <http://browserify.org/>
12. HTML <http://htmlbook.ru/>
13. GIT <https://git-scm.com/>
14. WebGL <https://ru.wikipedia.org/wiki/WebGL>

**Додаток 1 лістинг програми**

**var *createGame* = *require*('voxel-engine')**

**var *highlight*** = ***require***(**'voxel-highlight'**)

**var *player*** = ***require***(**'voxel-player'**)

**var *voxel*** = ***require***(**'voxel'**)

**var *extend*** = ***require***(**'extend'**)

**var *fly*** = ***require***(**'voxel-fly'**)

**var *walk*** = ***require***(**'voxel-walk'**)

**var *debris*** = ***require***(**'voxel-debris'**)

**var *toolbar*** = ***require***(**'toolbar'**)

module.exports = **function** (opts, setup) {

setup = setup || *defaultSetup*

**var** defaults = {

*// generate: voxel.generator['Valley'],*

**chunkDistance**: 2,

**worldOrigin**: [0, 0, 0],

**controls**: {**discreteFire**: **true**}

}

opts = ***extend***({}, defaults, opts || {})

*// setup the game and add some trees*

**var** game = ***createGame***(opts)

**var** container = opts.container || ***document***.**body**

***window***.**game** = game *// for debugging*

game.appendTo(container)

**if** (game.notCapable()) **return** game

**var** createPlayer = ***player***(game)

*// create the player from a minecraft skin file and tell the*

*// game to use it as the main player*

**var** avatar = createPlayer(opts.playerSkin || **'textures/shama.png'**)

avatar.possess()

avatar.yaw.position.**set**(2, 14, 4)

setup(game, avatar)

**return** game

}

**function** *defaultSetup*(game, avatar) {

**var** makeFly = ***fly***(game)

**var** target = game.**controls**.target()

game.**flyer** = makeFly(target)

*// highlight blocks when you look at them, hold <Ctrl> for block placement*

**var** blockPosPlace, blockPosErase

**var** hl = game.**highlighter** = ***highlight***(game, {**color**: 0xff0000})

hl.on(**'highlight'**, **function** (voxelPos) {

blockPosErase = voxelPos

})

hl.on(**'remove'**, **function** (voxelPos) {

blockPosErase = **null**

})

hl.on(**'highlight-adjacent'**, **function** (voxelPos) {

blockPosPlace = voxelPos

})

hl.on(**'remove-adjacent'**, **function** (voxelPos) {

blockPosPlace = **null**

})

*// toggle between first and third person modes*

***window***.addEventListener(**'keydown'**, **function** (ev) {

**if** (ev.**keyCode** === **'R'**.charCodeAt(0)) avatar.toggle()

})

*//selector*

**var** blockSelector = ***toolbar***({**el**: **'#tools'**});

**var** currentMaterial = 1;

blockSelector.on(**'select'**, **function** (material) {

**switch** (material) {

**case 'grass'**:

currentMaterial = 1;

**break**;

**case 'dirt'**:

currentMaterial = 3;

**break**;

**case 'obsidian'**:

currentMaterial = 4;

**break**;

**case 'plank'**:

currentMaterial = 5;

**break**;

**case 'glass'**:

currentMaterial = 6;

**break**;

}

});

game.on(**'fire'**, **function** (target, state) {

**var** position = blockPosPlace

**if** (position) {

game.createBlock(position, currentMaterial)

}

**else** {

position = blockPosErase

**if** (position) game.setBlock(position, 0)

}

})

game.on(**'tick'**, **function** () {

***walk***.render(target.**playerSkin**)

**var** vx = **Math**.abs(target.**velocity**.**x**)

**var** vz = **Math**.abs(target.**velocity**.**z**)

**if** (vx > 0.001 || vz > 0.001) ***walk***.stopWalking()

**else *walk***.startWalking()

})

}

**var *createGame*** = ***require***(**'voxel-hello-world'**);

**var *game*** = ***createGame***({

**materials**: [

**'grass'**,

[**'grass'**, **'dirt'**, **'grass\_dirt'**],

**'dirt'**,

**'obsidian'**,

**'plank'**,

**'glass'**

],

**texturePath**: **'./textures/'**,

});

<**html**>

<**head**>

<**title**>Minecraft</**title**>

<**meta http-equiv="cache-control" content="max-age=0"** />

<**meta http-equiv="cache-control" content="no-cache"** />

<**meta http-equiv="expires" content="0"** />

<**meta http-equiv="expires" content="Tue, 01 Jan 1980 1:00:00 GMT"** />

<**meta http-equiv="pragma" content="no-cache"** />

<**link rel="stylesheet" href="style.css"**>

<**link rel="shortcut icon" href="favicon.ico"** />

</**head**>

<**body**>

<**div id="crosshair"**></**div**>

<**nav class="bar-tab" id="tools"**>

<**ul class="tab-inner"**>

<**li class="tab-item active"**>

<**div class="tab-label"**>grass</**div**>

</**li**>

<**li class="tab-item"**>

<**div class="tab-label"**>dirt</**div**>

</**li**>

<**li class="tab-item"**>

<**div class="tab-label"**>obsidian</**div**>

</**li**>

<**li class="tab-item"**>

<**div class="tab-label"**>plank</**div**>

</**li**>

<**li class="tab-item"**>

<**div class="tab-label"**>glass</**div**>

</**li**>

</**ul**>

</**nav**>

<**script type="text/javascript" src="build.js"**></**script**>

</**body**>

</**html**>

{

**"name"**: **"voxelgame"**,

**"version"**: **"1.0.0"**,

**"description"**: **""**,

**"main"**: **"index.js"**,

**"scripts"**: {

**"test"**: **"echo \"Error: no test specified\"** **&& exit 1"**

},

**"author"**: **""**,

**"license"**: **"ISC"**

}

**body** {

**color**: **#eee**;

**font-family**: **Monospace**;

**font-size**: 12**px**;

**background-color**: **#f0f0f0**;

**margin**: 0**px**;

**overflow**: **hidden**;

}

**#container** {

**-moz-user-select**: **none**;

**-webkit-user-select**: **none**;

**user-select**: **none**;

}

**#crosshair** {

**position**: **fixed**;

**top**: 50%;

**left**: 50%;

**margin**: -2**px** 0 0 -2**px**;

**width**: 4**px**;

**height**: 4**px**;

**background-color**: **#d00**;

**opacity**: 0.5;

}

**#instruct** {

**font-size**: 80%;

**background-color**: **rgba**(0, 0, 0, 0.3);

**color**: **#eee**;

**text-align**: **right**;

**position**: **fixed**;

**bottom**: 10**px**;

**left**: 0**px**;

**width**: 100%;

**padding**: 10**px**;

**-webkit-box-sizing**: **border-box**;

**-moz-box-sizing**: **border-box**;

**box-sizing**: **border-box**;

}

*/\* toolbar \*/*

**nav**.**bar-tab** {

**position**: **fixed**;

**top**: 5**px**;

**left**: 5**px**;

**margin**: 0**px**;

**padding**: 0**px**;

}

**nav**.**bar-tab ul** {

**margin**: 0;

**padding**: 0;

**list-style**: **none**;

}

**nav**.**bar-tab li** {

**border**: 1**px solid #ccc**;

**-webkit-border-radius**: 3**px**;

**-moz-border-radius**: 3**px**;

**border-radius**: 3**px**;

**padding**: 3**px**;

**margin**: 0 5**px** 0 0;

**list-style**: **none**;

**float**: **left**;

**width**: 30**px**;

**height**: 30**px**;

**font-size**: 80%;

**color**: **#ccc**;

**text-shadow**: 1**px** 1**px** 1**px #333**;

**box-shadow**: 1**px** 1**px** 1**px #333**;

**overflow**: **hidden**;

}

**nav**.**bar-tab li**.**active** {

**border**: 1**px solid #ffcc00**;

}

**nav**.**bar-tab a** {

**color**: **#ccc**;

**text-decoration**: **none**;

}

**nav**.**bar-tab li** .**tab-label** {

**padding-top**: 5**px**;

**font-size**: 80%;

**text-align**: **center**;

}

**nav**.**bar-tab li**:**nth-child**(1) { **background-image**: **url**(**'textures/grass.png'**); }

**nav**.**bar-tab li**:**nth-child**(2) { **background-image**: **url**(**'textures/dirt.png'**); }

**nav**.**bar-tab li**:**nth-child**(3) { **background-image**: **url**(**'textures/obsidian.png'**); }

**nav**.**bar-tab li**:**nth-child**(4) { **background-image**: **url**(**'textures/plank.png'**); }

**nav**.**bar-tab li**:**nth-child**(5) { **background-image**: **url**(**'textures/glass.png'**); }

*/\* Share \*/*

.**voxel-share** {

**position**: **fixed**;

**top**: 50%;

**left**: 50%;

**margin-left**: -250**px**;

**margin-top**: -150**px**;

**width**: 500**px**;

**background-color**: **#eee**;

**padding**: 2**px** 2**px** 2**px** 2**px**;

**-moz-box-shadow**: 0 0 5**px #333**;

**-webkit-box-shadow**: 0 0 5**px #333**;

**box-shadow**: 0 0 5**px #333**;

**-webkit-border-top-left-radius**: 3**px**;

**-webkit-border-top-right-radius**: 3**px**;

**-moz-border-radius-topleft**: 3**px**;

**-moz-border-radius-topright**: 3**px**;

**border-top-left-radius**: 3**px**;

**border-top-right-radius**: 3**px**;

}

.**voxel-share img** {

**width**: 100%;

}

.**voxel-share textarea** {

**box-sizing**: **border-box**;

**margin**: 3**px** 0**px**;

**padding**: 5**px**;

**width**: 75%;

**font-family**: **Arial**, **sans-serif**;

**font-size**: 12**px**;

**background-color**: **#ddd**;

**float**: **left**;

}

.**voxel-share button** {

**border**: 1**px solid #333**;

**margin**: 5**px**;

**background-color**: **#ddd**;

**width**: 20%;

**float**: **right**;

**-webkit-border-radius**: 3**px**;

**-moz-border-radius**: 3**px**;

**border-radius**: 3**px**;

}