Сьогодні на факультеті розробляється система оцінювання навчальних досягнень студентів та методичне забезпечення самокоригування результатів навчальної діяльності студентів.

Висновки. Аналіз проблеми організації самостійної роботи студентів у вищій школі свідчить, що вона вже не один рік є предметом дослідження як вітчизняних, так і зарубіжних учених. Разом із тим вирішення цієї проблеми в сучасних умовах розбудови вищої школи, внаслідок входження вітчизняної системи освіти у Європейський простір, має сприяти перегляду теоретичних та методичних засад підготовки фахівців у вищих навчальних закладах.

Наш аналіз переконує, що самостійна робота є багатогранною і складається з: творчого сприймання, осмислення лекційного матеріалу під час його конспектування; вивчення навчальної літератури, першоджерел; підготовки до групових занять; закріплення знань: самостійного розв'язання задач або виконання інших індивідуальних домашніх завдань; підготовки до лабораторних робіт і їх виконання; підготовки до методичних занять і їх проведення; участь у гуртках; підготовка до екзаменів і заліків; виконання різних видів практик; написання курсових та дипломних робіт.

Викладацьким колективом факультету доведено, що ефективна підготовка висококваліфікованих фахівців в умовах широкого запровадження кредитно-модульної системи організації навчального процесу у вищих навчальних закладах має великий потенціал і може здійснюватися за умови організації самостійної роботи студентів засобами ІКТ.

ВІДОМОСТІ ПРО АВТОРА

Петренко Світлана Віталіївна – декан фізико-математичного факультету Сумського державного педагогічного університету імені А.С.Макаренка

Коло наукових інтересів: проблеми підвищення якості підготовки студентів фізико-математичного факультету в умовах кредитно-модульної системи навчання.

РОЗРОБКА ДИНАМІЧНИХ ОБ'ЄКТІВ ЗАСОБАМИ WEBGL

Максим РЯБЕЦЬ, Сергій РЯБЕЦЬ

Метою цієї публікації є демонстрація можливостей WebGl на прикладі візуалізації водної поверхні, що може бути корисним при створенні різного роду віртуальних моделей і одним з перспективних напрямків процесу модернізації підготовки студентів технологічних та природничих спеціальностей.

The purpose of this publication is to demonstrate the possibilities WebGl on the example visualization of the water surface, that may be beneficial in creating various kinds of virtual models and one of the most promising areas of the modernization process of training students of technological and natural specialties.

Революційний розвиток інтернет-технологій обумовлює їх впровадження в усі сфери життя сучасного суспільства, створює унікальні можливості для більш всебічного активного й ефективного росту не тільки економіки країни, а й кожного свідомого громадянина. Саме інформаційні технології мають сьогодні той величезний потенціал, який може привести до суттєвих змін в діяльності людської цивілізації.

Зрозуміло, реформування суспільства не можливе без модернізації системи освіти, де крім професійних знань, умінь і навичок в системі підготовки майбутніх фахівців різних спеціальностей важливою складовою повинні бути самостійність і ініціатива, творчий підхід і самоудосконалення. Тому, опанування новими інформаційними технологіями, на нашу думку, сприятиме набуттю таких якостей особистості в першу чергу для студентів технологічних та природничих спеціальностей.

Сучасні інформаційні технології насамперед пов'язані з інтернетом, де

використовуються відомі веб-браузери: Google Chrome, Firefox, IE, Mozilla, Safari, Android, Opera. Зображення сторінок таких програм подаються у 2D і 3D-графіці, яка використовує Flash технології (Flash - мультимедійна платформа компанії Adobe для створення веб-додатків або мультимедійних презентацій). Ці не безкоштовні технології широко використовуються для створення рекламних банерів, анімації, ігор, а також відтворення на веб-сторінках відео- та аудіозаписів. Новим відкритим функціоналом для створення високопродуктивної двох- і тривимірної графіки у веб-браузерах є технології WebGL та її бібліотеки THREE.js., що дозволяють безпосередньо програмувати графічний процесор комп'ютера і керувати ним [7]. WebGL являє собою технологію, що базується на OpenGL ES 2.0 (відкрита графічна бібліотека з вмістом понад 250 функцій для рисування складних тривимірних сцен з простих примітивів). При цьому для роботи з даною технологією не потрібні сторонні плагіни або бібліотеки. WebGL виник з експериментів над Canvas 3D (canvas - елемент HTML5, призначений для створення растрового двомірного зображення за допомогою мови JavaScript) американського розробника сербського походження Володимира Вукічевіча з компанії Mozilla в 2006 році. Згодом розробники браузерів Opera і Mozilla стали створювати свої реалізації WebGL. А пізніше було організовано робочу групу за участю найбільших розробників браузерів Apple, Google, Mozilla, Opera для роботи над специфікацією технології. І 3 березня 2011 була представлена специфікація WebGL 1.0. [1,5]. Веб-додатки, побудовані з використанням WebGL, є продуктом платформонезалежного програмного інтерфейсу: чи це комп’ютери з ОС Windows, Linux, MacOS, чи смартфони, планшети, ігрові консолі тощо. Вся робота веб-додатків з використанням WebGL заснована на сценарії JavaScript, а деякі елементи його коду – шейдери – можуть виконуватися безпосередньо на графічних процесорах, завдяки чому розробники можуть отримати доступ до додаткових ресурсів комп'ютера, збільшити швидкодію. До переваг застосування WebGL можна віднести використання розробниками стандартних для веб-середовища технологій HTML/CSS/JavaScript та автоматичне управління пам’яттю (не треба виконувати спеціальні дії для її виділення та очищення) [2,9].

Отже, метою нашої роботи була демонстрація можливостей WebGL для розробки графічних об’єктів на прикладі візуалізації водної поверхні як реального об’єкту на веб-сайті.

Моделювання водних поверхонь є складним завданням. Можна виділити такі підзадачі, як моделювання *невеликих водойм* із видимими межами, для яких характерні невеликі зміни поверхні, тобто невеликі коливання, а також інтерференція коливань від декількох сплесків і відбитих коливань від границь; *великі водні поверхні*: розглядаются *невеликі поверхневі коливання*, де поверхня розбивається на трикутники, але досить великого розміру і над поверхнею виробляють невеликі коливання, які відповідають невеликим хвилях; *великі коливання*: це великі хвилі, бризки тощо, де відбуваються значні деформації водних поверхонь, які досить складно фізично описуються, тому великі хвилі практично ніколи не візуалізуються. Бризки можна візуалізувати, наприклад, використовуючи масив частинок (маленьких площин, орієнтованих за нормаллю до спостерігача), для яких встановлюються початкові швидкості, а подальший їх політ відбувається за гравітаційним законом. В нашому випадку, ми обмежились першим випадком, а саме – моделювання невеликих водойм.

При створені нашої моделі ми користувались одним з методів візуалізації води в реальному часі, який ґрунтувався на припущенні про те, що водна поверхня плоска [3,8]. Ілюзія хвиль при цьому створювалася за рахунок рельєфного текстурування з використанням заздалегідь згенерованих карт для нормалей і висот. Такий підхід відмінно підходить для візуалізації невеликих і спокійних поверхонь води, наприклад озер, ставків і калюж. Крім того, плоска поверхня води сильно спрощує візуалізацію відображень, заломлень фізичної моделі, вимагає мінімальної кількості полігонів і дозволяє легко додавати спецефекти, наприклад, відбиття хвилі від об’єктів. Для реалізації вищевказаного підходу застосовуються імітаційні методи, що представляють собою сукупність емпірично підібраних залежностей для досягнення візуальної схожості результату. Їх використання вимагає трудомісткого підбору параметрів і можливе лише для обмеженого кола завдань. Крім отримання геометрії водної поверхні окрему проблему представляє її відображення на екрані. Для цього можна

використовувати методи, які візуалізують ландшафт. Однак, на відміну від незмінного ландшафту, геометрія водної поверхні постійно змінюється, що вимагає перебудовування сіток і також ускладнює процес побудови. В нашій роботі моделювання коливань і хвиль реалізовувалось за допомогою композицій сінусних (або косінусних) функцій, зі встановленими законами для поширення хвилі (наприклад, y=h*Math.Sin(i/n+(time+i)/k), коефіцієнти h,i,n,k – змінні). Це дозволяло моделювати відбиття хвиль від меж, інтерференцію хвиль та ін., адже саме сінусні функції можуть передавати характеристики хвилі, що може бути цілком достатнім для нашого випадку. Відображення текстурних координат залежать від положення точки поверхні та від часу. В моделі, зображеній на рис. 1, закладене таке змінювання швидкості коливань та висоти хвиль. Крім того, користувач має можливість пересуватись по поверхні за допомогою клавіш «стрілок» та «миші».

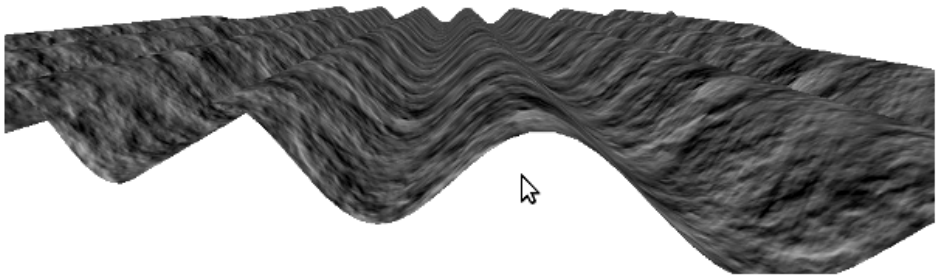

Рис.1. Моделювання коливань і хвиль засобами WebGL та її бібліотеки three.js.

Створення нашої моделі поверхні води вимагало використання з бібліотеки THREE.js. 4 основних складових, таких як *камера* (camera), *контролери* для камери (controls), *сцена* (scene), *відображення* (renderer). *Камера* та її *контролери* дають можливість користувачеві рухатись навколо об'єкта та встановити точку з якої користувач буде його спостерігати. *Сцена* дає можливість встановити що саме потрібно відобразити (туман, текстура, об'єкт тощо). *Відображення* відповідає за появу того чи іншого об'єкту на екрані. Потім використовувалась складова для створення площини, з якою можна здійснювати відповідні маніпуляції. Зокрема, встановлюється розмір поверхні, правильні нормалі граней для згладжування, матриці в яких зберігають вершини моделі, які динамічно додаються у відповідності з викликаною бібліотечною функцією. Пройшовши циклами за вершинами моделі, з'являється можливість встановлювати хвилі різної висоти та швидкості за допомогою відповідних параметрів (коефіцієнтів). Для кілець на воді використовувалися такі ж функції та додатково були створені функції для розрахунків початкових точок вершин кілець, їхнього постійного оновлення та циклічності. Далі, при наявності кілець на поверхні води включалась функція їхнього затухання (зменшення попередніх параметрів), аж до стану спокою поверхні на границях (рис.2).

На рис.2 показана реалізована модель коливання на хвилі від падаючого згори об'єкта, що імітується натисненням правої клавіші «миші» в довільній точці на площині води.

Таким чином, розроблені моделі можуть бути використані як невеликі симулятори водних поверхонь при демонстраціях коливальних процесів для студентів природничих спеціальностей, в курсах інженерної графіки та технічного дизайну при підготовці студентів спеціальності Технологічна освіта тощо. Удосконалення результатів роботи ми пов'язуємо зі створенням більш точних фізико-математичних описів майбутніх віртуальних об'єктів, створенням нових методів та алгоритмів, які дозволили б отримати візуально схоже зображення і задовольняли б поставленим вимогам за швидкодією. Це наступний крок у використанні нових технологій WebGL при створенні комп'ютерних браузерних ігор, САПР, систем віртуальної реальності, візуалізації в наукових дослідженнях.

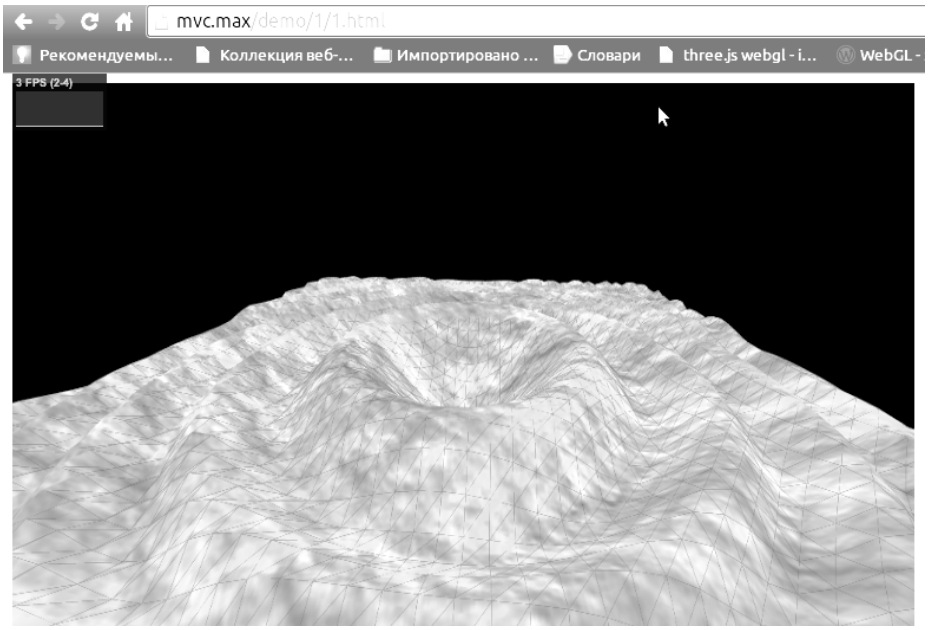

Рис. 2. Моделювання коливань кілець води засобами WebGL

Отже, з появою таких технологій як WebGL стало можливим вбудовування 3D графіки прямо у зміст Web-сторінки. Швидше за все в недалекому майбутньому користувач зможе працювати з тривимірною графікою, грати в тривимірні ігри, читати тривимірний текст - і все це прямо у вікні браузера [4; 6].

Величезним плюсом використання технології WebGL є висока швидкість роботи, за рахунок того, що весь вміст сайту завантажується спочатку і нічого не довантажується в процесі використання. Це зручно, і дозволяє швидко працювати з додатком. Швидкість роботи цього додатка залежить тільки від характеристик відеокарти, тому що додаток в браузері фактично працює як звичайний графічний додаток і використовує ресурс відеокарти, встановленої на комп'ютері, а його відображення відбувається в браузері.

БІБЛІОГРАФІЯ

1. Parisi T. WebGL: Up and Running. - USA: O'Reilly Media, 2012.- 230 с.
2. Инструменты веб-разработчика: Three.JS [Електронний ресурс]. – Режим доступу: http://blog.divkit.ru/instrumenty-veb-razrabotchika-threejs.html (03.06.2012).
3. Создание реалистичной поверхности воды с использованием GLSL [Електронний ресурс]. – Режим доступу: http://www.gamedev.ru/code/ /articles//?id=4361&page=2 (09.06.2009).
4. Изучаем WebGL [Електрон. ресурс] /2011. – Режим доступу: http://metanit. com/web/webgl/1.1.php.
5. Чистый WebGL 101 [Електрон. ресурс]/Часть 1: Введение /2012. – Режим доступу: http://css-live.ru/articles/chistyj-webgl-101-chast-1-vvedenie.html.
6. Learning WebGL [Електронний ресурс]. – Режим доступу: http:// //learningwebgl.com/blog/?page_id=1217 (12.07.2009).
7. Web-based Graphics Library [Електрон. ресурс] / JavaScript /2012. – Режим доступу: http://ru.wikipedia.org/wiki/WebGL.
8. WebGL and simulations on GPU notes / [Електронний ресурс]. – Режим доступу: http://www.ibiblio.org/e-notes/webgl/gpu/contents.htm (23.10.2013).
9. WebGL [Електронний ресурс]/2010. – Режим доступу: http://russian-webgl. blogspot.com/.

ВІДОМОСТІ ПРО АВТОРІВ

Рябець Максим Сергійович – магістрант кафедри прикладної математики, статистики та економіки Кіровоградського державного педагогічного університету імені Володимира Винниченка, кандидат технічних наук.

Рябець Сергій Іванович - доцент кафедри теорії і методики технологічної підготовки, охорони праці та безпеки життєдіяльності Кіровоградського державного педагогічного університету імені Володимира Винниченка, кандидат технічних наук.

Коло наукових інтересів: проблеми технологічної освіти, web програмування, хмарні технології.