

Трофимов Данило, здобувач вищої освіти

КПІ ім. Ігоря Сікорського, Україна

Науковий керівник: Родіонов Павло Юрійович,

доцент кафедри інформатики та програмної інженерії

КПІ ім. Ігоря Сікорського, Україна

ОСОБЛИВОСТІ СТВОРЕННЯ ШЕЙДЕРНИХ ПРОГРАМ ДЛЯ РЕАЛІЗАЦІЇ ОСВІТЛЕННЯ

Анотація. У роботі розглядаються теоретичні засади та практичні аспекти реалізації моделі освітлення Фонга. Запропоновано підхід до реалізації моделі освітлення на базі графічного прикладного інтерфейсу WebGL. Проаналізовано особливості реалізації даної моделі освітлення та запропоновано відповідні шейдерні програми.

КЛЮЧОВІ СЛОВА: модель освітлення, фрагментний шейдер, вершинний шейдер.

Abstract. The paper examines the theoretical foundations and practical aspects of the implementation of the Fong lighting model. An approach to the implementation of the Fong lighting model based on the WebGL graphical application interface is proposed. The peculiarities of the implementation of this lighting model were analyzed from the point of view of the load on the hardware of the information system.

KEY WORDS: fragment shader, vertex shader, lighting model.

Вступ. Розповсюдженою проблемою у сфері комп'ютерної графіки традиційно вважається реалізація освітлення. На даний момент існує множина моделей освітлення, кожна з яких має власні переваги та недоліки. Проте важливою науковою задачею видається систематизувати та узагальнити теоретичні засади реалізації освітлення, а також дослідити методи реалізації конкретної моделі освітлення на базі графічного прикладного інтерфейсу WebGL.

Основна частина. Процеси моделювання освітлення базуються на законах геометричної оптики, таких як закони заломлення, відбиття, прямолінійності тощо. Енергія світла, що падає на поверхню від джерела світла, може бути частково поглинута, відбита або пропущена через поверхню. Кількість енергії, що поглинається, відбивається і пропускається залежить від довжини λ світлової хвилі. Об'єкт можна побачити тільки тоді, коли він відбиває або пропускає світло. Властивості відбитого світла, які визначають колір поверхні, залежать від форми і напрямку джерела світла, від орієнтації поверхні, на яку

падає світло та від властивостей самої поверхні. Цей факт враховується в різних моделях освітлення. Дійсна теорія світла є досить складною з теоретичної та обчислювальної точок зору, тому будуються спрощені моделі освітлення [1].

Модель Фонга аналогічна до моделі Гуро, але, на відміну від нього, тут інтерполюються не значення інтенсивності відбитого світла, а значення вектора зовнішньої нормалі. Тобто, модель Фонга полягає в побудові для кожного пікселя вектора, що відіграє роль нормалі і цей вектор використовується для обчислення інтенсивностей компонент відбитого світла в кожній точці згідно з вибраною моделлю освітлення. Тому зафарбовування на основі

моделі Фонга вимагає значно більше обчислень, оскільки інтерполюються три векторних компоненти, проте одержуються більш реальні графічні зображення [1].

Реалізація освітлення на базі API WebGL передбачає написання вершинних та фрагментних шейдерів. Нижче наведено лістинг вершинного шейдера WebGL-програми, у якій реалізовано модель освітлення.

```
void main() {  
    gl_Position = uProjectionMatrix * uModelViewMatrix *  
    vec4(aVertexPosition, 1.0);  
    vColor = aVertexColor;  
    vNormal = normalize(vec3(uNormalMatrix *  
    vec4(aNormal, 0.0)));  
    vViewDirection = normalize(-vec3(uModelViewMatrix *  
    vec4(aVertexPosition, 1.0)));  
}
```

Нижче запропоновано лістинг фрагментного шейдера для реалізації моделі освітлення.

```
void main() {  
    vec3 ambient = uAmbientColor * vColor;  
    vec3 lightDirection = normalize(uLightPosition -  
    gl_FragCoord.xyz);  
    vec3 normal = normalize(vNormal);  
    vec3 diffuse = uDiffuseColor * vColor * uLightColor;  
    vec3 reflectionDirection = reflect(-lightDirection, normal);  
    vec3 viewDirection = normalize(vViewDirection);  
    float specularFactor = pow(max(dot(reflectionDirection,
```

```
viewDirection), 0.0), uShininess);  
    vec3 specular = uSpecularColor * specularFactor *  
    uLightColor;  
    vec3 finalColor = ambient + diffuse + specular;  
    gl_FragColor = vec4(finalColor, 1.0);  
}
```

Оскільки поверхня має передню та задню грані, кожна сторона має свою нормаль. Наприклад, поверхня, перпендикулярна до осі z, має передню поверхню, яка звернена до позитивного напрямку осі z, і задню поверхню, яка звернена до негативного напрямку осі z. Їх нормалі дорівнюють (0, 0, 1) і (0, 0, -1) відповідно.

У 3D-графіці ці дві грані розрізняються порядком, у якому вказані вершини під час рендерингу поверхні. Коли відбувається рендеринг із зазначенням вершин, передня грань – це та, вершини якої розташовані за годинниковою стрілкою, якщо дивитися вздовж нормалі грані [2].

Висновки. У роботі було систематизовано теоретичні основи щодо реалізації освітлення при створенні графічних сцен. Було запропоновано підходи до написання шейдерних програм для реалізації моделі освітлення, а також наведено приклад таких шейдерів. В якості перспектив подальших наукових досліджень може бути запропоновано створення порівняльного аналізу реалізації моделі освітлення Фонга на базі різних графічних програмних інтерфейсів.

Список інформаційних джерел

1. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
2. Matsuda K. & Lea R. (2013). WebGL programming guide : interactive 3d graphics programming with WebGL. Addison-Wesley.