

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Одеська національна академія харчових технологій**  
**Університет Інформатики і прикладних знань, м.Лодзь, Польща**  
**Національний технічний університет України «Київський**  
**політехнічний інститут»**  
**Навчально-науковий інститут комп'ютерних систем і технологій**  
**«Індустрія 4.0» ім. П.М. Платонова**

**XXI Всеукраїнська науково-технічна конференція**  
**молодих вчених, аспірантів та студентів**

**«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ**  
**ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»**

*Матеріали конференції*



Одеса

22-23 квітня 2021 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXI Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 22-23 квітня 2021 р. - Одеса, Видавництво ОНАХТ, 2021 р. – 229 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

## ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

Голова - д.т.н., проф., Єгоров Б.В., ректор ОНАХТ.

### Співголови:

**Поварова Н.М.** – к.т.н., доц., проректор з наукової роботи ОНАХТ,  
**Котлик С.В.** – к.т.н., доц., директор ННІКСіТ "Індустрія 4.0" ОНАХТ,  
**Даріуш Долива**, д.математичн.наук, уповноважений декана факультету Інформатики УІтаПЗ, м.Лодзь, Польща,  
**Ковалюк Т.В.** - к.т.н., доц. кафедри АСОІтаУ НТУУ «Київський політехнічний інститут»

### Члени оргкомітету:

**Плотніков В. М.** – д.т.н., проф., завідувач кафедри ІТтаКБ ОНАХТ,  
**Артеменко С.В.** – д.т.н., проф., завідувач кафедри КІ ОНАХТ,  
**Хобін В.А.** – д.т.н., проф., завідувач кафедри АТПтаРС ОНАХТ,  
**Тарасенко В.П.** – д.т.н., проф., завідувач кафедри СКС НТУУ «Київський політехнічний інститут»,  
**Невлюдов І.Ш.** – д.т.н., проф., завідувач кафедри КІТАМ ХНУРЕ,  
**Мельник А.О.** – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,  
**Жуков І.А.** – д.т.н., проф., завідувач кафедри КСтаМ НАУ.

Матеріали подано українською, російською та англійською мовами.  
Редактор збірника Котлик С.В.

технологій)	
ПЕРСПЕКТИВИ РОЗРОБКИ ТА ВПРОВАДЖЕННЯ WEB-ДОДАТКІВ. <b>ГАФІЯК А.М., ДЯЧЕНКО-БОГУН А.О., ЧЕПІГА Р.В.</b> (Національний університет «Полтавська політехніка імені Юрія Кондратюка»)	212
КІБЕРСПОРТ ЯК ІНВЕСТИЦІЙНО ПРИВАБЛИВА ГАЛУЗЬ ДЛЯ УКРАЇНИ. <b>ЖМАЙ О.В.</b> (Громадська організація «Молодіжна організація “Енектус” при Одеському національному університеті імені І.І.Мечникова)	214
ГЕНЕРАЦІЯ МЕШУ НА ОСНОВІ ІГРОВОГО РУШІЯ <b>UNITY</b> . <b>КУЛАКОВ В.А., ЖУКОВЕЦЬКА С.Л.</b> (Одеська національна академія харчових технологій)	216
РОЗВИТОК КОМП'ЮТЕРНИХ ІГОР ЗА МОТИВАМИ КНИГ. <b>ПИЛИПЕНКО С.А., СІРЕНКО О.І.</b> (Одеська національна академія харчових технологій)	217
АНАЛІЗ ТРИВИМІРНИХ ДИСПЛЕЇВ. <b>РОМАНЮК О.Н., ДЕДА В.П., ХОШАБА О.М.</b> (Вінницький національний технічний університет)	218
РОЗРОБКА ІГРОВОГО ЕКОНОМІЧНОГО СИМУЛЯТОРА. <b>БОДЮЛ О.С., СІРОМЛЯ Д.С.</b> (Одеська національна академія харчових технологій)	220
PLANING THE OPTIONAL WAY OF MODILE WORK. <b>TROFIMENKO M.S.</b> (National aviation university)	222
АНАЛІЗ ШЕЙДЕРІВ. <b>РОМАНЮК О.Н., ІВАХА О.А., ДУДНИК О.О.</b> (Вінницький національний технічний університет)	223
ОГЛЯД СУЧАСНИХ ТЕХНОЛОГІЙ СТВОРЕННЯ ВЕБ-СТОРИНОК ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ. <b>ЧУБАРОВ Є.Є., ЗИБІНА К.В.</b> (Харківський Національний Університет Радіоелектроніки)	225
ВИКОРИСТАННЯ ФРАКТАЛІВ ДЛЯ КОМПАКТНОГО КОДУВАННЯ ПАТЕРНІВ В ГРАФІЧНОМУ ДИЗАЙНІ. <b>ШЕВЧЕНКО В.В., ШЕВЧЕНКО О.В.</b> (Київський національний університет імені Тараса Шевченка)	227

UDC 004.3.

## **PLANING THE OPTIONAL WAY OF MODILE WORK**

TROFIMENKO M.S. (*mitrofimow833@gmail.com*)

National aviation university

*The main tasks in planning the optimal path are considered. The shortest path search algorithms are compared. The conclusion is made which algorithm is better to use under certain conditions and tasks.*

Mobile robot motion planning is one of the main tasks of robotic systems. The task of planning the optimal path in the general setting is formulated as follows: on the map of the area it is necessary to determine the route of movement from the starting set of points to the set of endpoints. In this statement, the initial and final are determined in the calculation procedure. The following task options are possible:

- 1) pave the optimal route from the starting point to the end point (start and end points are set);
- 2) pave the optimal route from the set of starting points to a given set of endpoints;
- 3) to build an area of transport accessibility with a given level of costs in relation to many starting points.

Most algorithms for laying optimal routes are based on the method of dynamic planning Bellman - Ford on weighted graphs [1]. For a transport network, the vertices of the graph are the nodes of the transport network, and the arcs correspond to the transitions between the nodes. INgraph theory the problem of the shortest path is to find waybetween the two vertices (or nodes) of the graph so that the sum of the weights of the edges of which it consists is minimal. An example is finding the shortest path between two points on a road map. In this case, the vertices are points, and the edges are roads with weights equal to the time required to overcome this segment.

Consider the main differences of the following algorithms [2, 3]: Dijkstre, Bellman-Ford, search A \*, Floyd-Warshall, Johnson, Algorithm-Lee wave tracing, A \* with iterative deepening, search algorithm for the first best match.

The Bellman-Ford algorithm is a shortest path search algorithm that allows you to determine the path to the vertices, the edges of which can have a negative weight when compared with the Dijkstri algorithm. The algorithm consists of several phases. At each phase, all edges of the graph are considered and the algorithm tries to make a relaxation along each edge of the value.

Dijkstri search algorithm. It should be borne in mind that this algorithm can be applied only to those graphs in which the weights of the edges are not negative, this is its main difference from the Bellman-Ford algorithm. In the process of execution, the algorithm will check each of the vertices of the graph, and find the shortest path to the original vertex. The standard implementation works on a weighted graph - that is, a graph in which each path has a weight, ie the "cost" that will have to be "paid" to go over this edge. On an uneven field, the weight of each edge of the graph is assumed to be the same.

In 2011, another algorithm was introduced - finding a jumping point, which is an improved A \*, it speeds up the search for a path, "jumping" many places that need to be considered. Unlike such algorithms, JPS does not require pre-processing and identification of additional memory costs.

The algorithm IDA \* (A \* with iterative deepening), is a modified version of A \*, uses less memory due to fewer deployed nodes. Works faster A \* in case of successful choice of heuristics.

The Lee-wave tracing algorithm is a search algorithm for the first best match. First of all, we consider nodes, the path from which to the final vertex is probably shorter, that is, due to the heuristics, the same thing is done that A \* does for the Dijkstri algorithm. This algorithm is an extension of Dijkstry's algorithm, the acceleration of work is achieved through heuristics - when considering each individual vertex, the transition is made to the neighboring vertex, the expected

path from which to the desired vertex is the shortest. It is used to find the shortest path on the map in an unfamiliar area.

As a result of the analysis, it was concluded that when developing a mobile robot for movement and orientation in confined spaces, the IDA \* search algorithm has advantages, and for the development of combat mobile robot - it is better to use the Lee Algorithm - wave tracing.

#### **List of literature sources**

1. K. K. Tomas, R. L. Rivest, and K. F. Shtajn, "Algoritmi: pobudova i analiz," M.: «Vilyams», pp. 296-391, Aug. 2006.

2 Y.G. Martynenko, "Upravlenie dvizheniem mobilnykh kolesnykh robotov," pp. 1590-1594, Sept. 2005.

УДК 004.925

### **АНАЛІЗ ШЕЙДЕРІВ**

РОМАНЮК О. Н., ІВАХА О. А., ДУДНИК О.О.  
Вінницький національний технічний університет

*Комп'ютерна графіка широко використовується в різних галузях діяльності людини, оскільки забезпечує реалістичне відтворення предметів і процесів [1]. При формуванні графічних зображень широко використовуються шейдери.*

Шейдером [1-7] називається програма для візуального визначення поверхні об'єкта. Це може бути опис розсіювання та поглинання світла, накладення текстури.

**Шейдер** – це комп'ютерна програма, яку виконують процесори відеокарти (GPU). Відеокарта обробляє потік даних, які мають точку входу та точку виходу, код команд, які виконують обробку. Під «поток даних» мається на увазі інформація про координати вершин полігонів і текстурних координат, які передаються в програму шейдера.

Існує три типи шейдерів [2-7]: вершинні (вертексні), геометричні та фрагментні (піксельні). Основна перевага використання шейдерів – їх гнучкість, що спрощує і здешевлює цикл розробки програми, і при тому що підвищує складність і достовірність сцен, що візуалізуються.

**Вертексний (вершинний) шейдер** – це найбільш усталений і найпоширеніший тип серед 3D-шейдерів. Метою таких шейдерів є перетворення 3D-положення кожної вершини у віртуальному просторі до 2D-координати, при якій вона відображається на екрані.

Вершинний процесор може виконувати різні види операцій з графікою: маніпулювання координатами положення кольору або текстури, перетворення вершин і нормалі. Вершинні шейдери забезпечують чіткий контроль над деталями положення, руху, освітлення та кольору.

**Піксельний шейдер**, також відомий як фрагментний, обчислює колір, текстурні координати, глибину та інші параметри кожного фрагмента зображення. Піксельний шейдер використовується на кінцевому етапі графічного конвеєра для формування фрагмента зображення [1].

**Геометричний шейдер** – це шейдер, який здатний обробляти та генерувати нові графічні примітиви, такі як точки, лінії та трикутники. Крім того, геометричний шейдер здатний генерувати примітиви «на льоту», не залучаючи при цьому центрального процесора. Вперше почав використовуватися на відеокартах Nvidia серії 8 [8].

На даний момент існує близько шести різних мов програмування шейдерів, і всі вони так чи інакше транслюють в одні і ті ж машинні коди, які виконуються на GPU [6].

**Шейдерні мови програмування** є C-подібними, але в той же час більш гнучкі в роботі зі змінними. У цілому, мова для шейдерів націлена під конкретне завдання, а саме: вираховування з плаваючою комою, матрицями чи векторами.

Максимальна ефективність і швидкість шейдерної програми залежить від ряду умов :

**XXI Всеукраїнська науково-технічна конференція  
молодих вчених, аспірантів та студентів**

**«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ  
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»**

Одеса

22-23 квітня 2021 р.

Збірник включає доповіді учасників конференції. Тези доповідей публікуються у вигляді, в якому вони були подані авторами.

Відповідальність за зміст і форму подачі матеріалу несуть автори статей.

**Редакційна колегія:** Котлик С.В., Корнієнко Ю.К.

**Комп'ютерний набір і верстка:** Соколова О.П.

**Відповідальний за випуск:** Котлик С.В.