

Here  $I$  is the set of all city intersections. Expression (5) consists of variable (first term) and constant components (second term). The variable component is a function of time because a traffic in city is a highly dynamic entity.

Expression (5) can be represented as a record

$$\sum_h W^h(n_{ij}) \rightarrow \min, \quad (6)$$

which is an objective function of the problem: the total weight of all consecutively located along the route lanes should be minimal. The minimization of function (6) is performed by an A-star algorithm, the main code of which for the graph [2].

*Definition 2. The weight of a route (6) is a minimum virtual path traveled by the vehicle from the starting position to the final one. The passage of such route takes a minimum of time (taking into account the same average speed  $V_{ij}$  ( $i, j \in A$ ) car movement in the city).*

Expression (6) is equivalent to the following

$$t^{trip} = \sum_{j=1}^I (t_j^{delay} + L_{ij} / V_{ij}) \rightarrow \min, \quad (7)$$

where  $t^{trip}$  – travel time of vehicle from its starting position to the destination one.

*Definition 3. Equivalence of expressions (6) and (7) gives grounds to talk about the optimization of vehicle travel on selected route just in time. Accordingly, the total time spent by all vehicles involved in traffic will also be optimal (minimal).*

Thus, the algorithm proposed in the study minimizes an actual travel time of the vehicle on the selected route.

1. A multigraph model that reproduces the transport network of a city district, simulates all an actual existing lanes. Each arc of the graph receives a weight that changes synchronously according to changes in traffic.

2. The study developed a working software module [1] that navigates an optimal time routes in graph, and hence in the real transport network. The investigation uses a heuristic A-star algorithm – a powerful computational method of graph theory. This makes it possible to synchronize vehicles flows and therefore urban traffic takes a qualitatively new level.

## VI. LIST OF REFERENCES

1. Nikolyuk, P.K., Prjamukhina, O-M. D, Perepelytsia A.S. (2022). A-Star algorithm. GitHub: [A-Star\\_algorithm/Astar.java at main · npk54/A-Star\\_algorithm \(github.com\)](https://github.com/npk54/A-Star_algorithm)
2. Sewall, J., van den Berg, J., Lin, M. C., & Manocha, D. (2011). Virtualized Traffic: Reconstructing Traffic Flows from Discrete Spatiotemporal Data. IEEE Transactions on Visualization and Computer Graphics, 17(1), 26-37. <https://doi.org/10.1109/tvcg.2010.27>

**УДК 004.9**

*Пархоменко В.В., студент 2*

*курсу спеціальності 122*

*«Комп'ютерні науки»*

*Потапова Н. А., к.е.н., доцент,*

## **ВИКОРИСТАННЯ ТЕХНОЛОГІЇ NANITE В UE5**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Unreal Engine 5 – один з найпопулярніших у світі інструментів для створення 3D-вимірів у реальному часі з фото реалістичними зображеннями. [1].

Nanite – це віртуалізована геометрична система Unreal Engine 5, яка використовує новий внутрішній формат сітки та технологію візуалізації для відтворення деталей масштабу пікселів і великої кількості об'єктів. Але варто враховувати, що ця технологія працює лише над деталями, які можна сприйняти. [2]

Переваги цієї системи та актуальність полягають в тому, що:

- вона досить сильно доповнює геометрію зображення, збільшуючи кількість об'єктів, які обробляються в реальному часі (що було складніше робити раніше на основі UE4);
- в ній відсутні обмежень на кількість полігонів і використання пам'яті для прорисовки геометрії;
- усувається потреба в ручному налаштуванні рівня деталізації для окремих моделей, тому що це виконується автоматично;
- майже повністю відсутні втрати в якості картини;
- стає можливим імпортування високоякісних артів, отриманих в програмах для моделювання, напряду.

Але варто розуміти, що деякі обмеження таки залишаються. Тому основною метою є дослідження системи Nanite, її можливостей та недоліків.

Якщо коротко, то працює система Nanite таким чином: під час імпорту, меші аналізуються та розпадаються ієрархічні кластери, які складаються з груп трикутників і під час рендерингу на різних рівнях деталізації, в залежності від положення камери, кластери швидко змінюються і без жодних проблем з'єднуються з сусідніми кластерами всередині одного і того ж об'єкту.

Якщо ж говорити про практичне застосування, то найкраще система покаже себе з мешами, які містять велику кількість трикутників дуже маленького розміру та які мають велику кількість екземплярів на сцені. Найгірше Nanite використовувати для умовних небесних сфер, в яких трикутники будуть величезними а сама сфера єдиний об'єкт на сцені.

В даний момент, можливості системи обмежені твердотілими мешами, які займають більшу частину сцени для проектів. Nanite підтримує різноманітні динамічні переміщення, повороти та масштабування, але не застосовується деформація меша в цілому, незалежно від того, статичний він чи динамічний.

В більшості випадків, масштабування відбувається разом з розширенням екрану за допомогою дрібно блокового рівня деталізації і відсікання на основі оклюзії. Тобто зазвичай, кількість трикутників, які виводяться на екран, пропорційна кількості пікселів, незалежно від геометричної складності вихідних

даних зі сцени. Є винятки, які можуть порушувати методи Nanite, але це буде говорити лише про те, що на етапі рендерингу, вона буде працювати так само, як і звичні системи.

Таким чином, на даний момент як UE5, так і сама система Nanite ще доробляються, але вже зараз можна побачити, що завдяки Nanite, рендеринг сцен і зображення відбувається значно швидше, але на більш детальних скульптурах можуть вилізати боком спотворення та нерівності геометрії.

#### Список літератури

1. URL: <https://www.unrealengine.com/en-US/unreal-engine-5>.
2. URL: <https://docs.unrealengine.com/5.0/en-US/nanite-virtualized-geometry-in-unreal-engine>.

#### УДК 004.9

*Перепелиця А.С., студент 3 курсу спеціальності 122 «Комп'ютерні науки»*

*Ніколюк П. К., д.ф.-м.н., професор, професор кафедри інформаційних технологій*

### ПРОКЛАДАННЯ ОПТИМАЛЬНОГО МАРШРУТУ ПЕРЕМІЩЕННЯ ТРАНСПОРТНОГО ЗАСОБУ У ТРАНСПОРТНІЙ МЕРЕЖІ МІСТА

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Умовно процес прокладання маршруту для транспортного засобу у транспортній мережі міста можна поділити на 3 етапи:

- Підготовка абстрактної моделі мапи
- Вираховування оптимального маршруту від поточної позиції ТЗ до кінцевої точки
- Перерахунок маршруту з врахуванням змін на дорогах міста

Абстрактно реальну карту міських доріг можна реалізувати у вигляді орієнтованого непланарного зваженого мультиграфа [1,2].