

На правах рукопису

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеська національна академія харчових технологій  
Навчально-науковий інститут холоду,  
кріотехнологій та екоенергетики  
Факультет інформаційних технологій та кібербезпеки

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

**“СТАН, ДОСЯГНЕННЯ І ПЕРСПЕКТИВИ  
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ”**

*Матеріали конференції*



Одеса  
25–26 квітня 2016 р.

**Стан, досягнення і перспективи інформаційних систем і технологій** / Матеріали XVI Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 25–26 квітня 2016 р. - Одеса, Видавництво ОНАХТ, 2016 р. - 176 с.

Збірник включає матеріали доповідей її учасників, які об'єднані по секціях кафедр: комп'ютерної інженерії (КІ), інформаційних технологій та кібербезпеки (ІТтаКБ).

## **ОРГАНІЗАЦІЙНИЙ КОМІТЕТ**

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

Співголови :

**Капрельянець Л.В.** – д.т.н., проф., проректор з наукової роботи та міжнародних зв'язків,

**Косой Б.В.** – д.т.н., проф., в.о. директора ННІХКтаЕ ОНАХТ,

**Котлик С.В.** – к.т.н., доц., декан ФІТта КБ ОНАХТ,

**Волков В.Е.** – д.т.н., доц., директор ННІМАтаКС ОНАХТ,

**Хобін В.А.** – д.т.н., проф., завідувач кафедри автоматизації виробничих процесів ОНАХТ,

**Невлюдов І.Ш.** – д.т.н., проф., завідувач кафедри технології і автоматизації виробництва радіоелектронних і електронно-обчислювальних засобів ХНУРЕ,

**Мельник А.О.** – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,

**Тарасенко В. П.** – д.т.н., проф., завідувач кафедри СПіСКС НТУУ «Київський політехнічний інститут»,

**Жуков І. А.** – д.т.н., проф., директор інституту комп'ютерних технологій Національного авіаційного університету.

### **Члени оргкомітету:**

**Плотніков В. М.** – д.т.н., проф., завідувач кафедри інформаційних технологій та кібербезпеки ОНАХТ.

**Артеменко С.В.** – д.т.н., проф., в.о. завідувача кафедри комп'ютерної інженерії ОНАХТ.

**Князєва Н.О.** – д.т.н., проф. кафедри комп'ютерної інженерії ОНАХТ.

**Грищенко І.В.** – к.т.н., заступник декана ФІТта КБ ОНАХТ.

**Шамрай О.А.** – к.т.н., доц. кафедри ТДтаВЕ ОНАХТ.

Матеріали подано українською, російською та англійською мовами.  
Редактор збірника Шамрай О.А.

2. Гайворонська Г.С. Навчально-Методичний посібник до лабораторних робіт з дисципліни «Системи доступу користувачів. Частина 3. Функціонування технологій сімейства xDSL.» Одеса 2008.
3. Соколов Н.А. Сети абонентського доступу: перспективы развития/Научно-техническое издание/Соколов Н.А. – Электросвязь, 1997.
4. ИНСТУТ [Електронний ресурс]:Режим доступу <http://www.intuit.ru> 25.10.2014.
5. Група компаний ЛВС [Електронний ресурс]:Режим доступу <http://www.lvs.net.ua> 7.11.2014.

## **МАТЕМАТИЧНІ ФУНКЦІЇ В ІГРАХ «THIRD PERSON STEALTH»**

*Клімук Ю. Є., студентка 542 гр., ОНАХТ*

*Науковий керівник: Шестопалов С. В., к.т.н., доцент кафедри КІ, ОНАХТ*

В наші дні ігрова індустрія стала практично окремою областю мистецтва як кіно, театр і т.д. Це стало причиною великої кількості жанрів ігор, і в кожному жанрі є свої унікальні відмінності. Комп'ютерний гравець переміщається з одного віртуального світу в інший, швидко сприймає незнайомі ситуації й адаптується до них. У кожному разі гравцеві дають повну свободу вибору і безліч пропозицій – кожен знайде щось для себе.

На відміну від класичних відеоігор, де запорукою успіху служать швидка реакція і достатній запас боєприпасів, «third person stealth» вимагає від гравця зовсім інших навичок. У ньому для успішного проходження місії необхідно ховатися, непомітно переміщатися і вбивати ворогів без зайвого шуму, а часто і зовсім уникати будь-яких контактів з противником.

Кожен рік на ринку з'являються проекти жанру «third person stealth». Цього року на прилавках магазинів можна знайти такі шедеври, як «Metal Gear Solid: Phantom Pain» та «Assassin's Creed: Syndicate».

При розробці гри в жанрі «third person stealth» є задачі, котрі доцільно вирішувати з використанням математичних функцій.

Припустимо, що в грі є база і на ній є склад. Шпигунові необхідно дізнатися площу, яку займає склад. Так як база охороняється, він не може піти з рулеткою і виміряти. Тому можна дати йому бінокль з можливістю вимірювання розмірів об'єктів.

Тоді для створення віртуального біноклю можемо використати теорему косинусів: квадрат сторони трикутника дорівнює сумі квадратів двох інших його сторін без подвоєного добутку цих сторін і косинуса кута між ними. Загальна формула для всіх трикутників[1]:

$$a^2 = b^2 + c^2 - 2*b*c*\cos(A) \quad (1)$$

Розглянемо інакшу ситуацію: якщо є територія, що охороняється і охоронці розташовані по периметру, є пристрій стеження, який повідомляє охоронцям, якщо на територію пробереться чужий.



Рисунок 1 – Територія, що охороняється

Якщо в цю зону, що охороняється, пробрався ворог, система стеження його засікла і відправила усю охорону його ловити. Але посилати всіх нераціонально. Поки одні добіжуть, інші вже його схоплять – марна трата часу і сил. Тому треба розділити територію на чотири частини. І якщо гравець в зоні другого квадранту (А) то треба посилати групу 1, якщо в зоні першого квадранту (Б) – то 2, в третьому (Г) – то 3, в четвертому (В) – то 4. Для того, щоб дізнатися в якій зоні знаходиться ворог, можна скористатися знаками тангенса або котангенс: в зонах А і В вони від’ємні, а в зонах Б і Г – додатні.

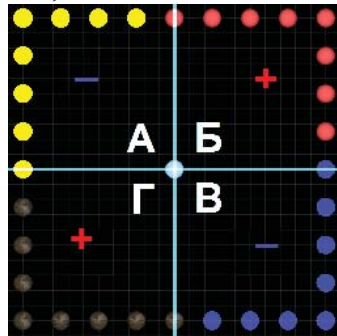


Рисунок 2 – Приклад знаків тангенса і котангенса

А щоб не плутати зони А з В і Б з Г, скористаємося знаками косинуса: А, Б – додатні, В, Г – від’ємні.

Може виникнути і така ситуація: якщо є гравець, який крадеться по території секретної бази і через системи датчиків, розташованих на всій території, охоронець знає точне місце розташування гравця. Щоб схопити гравця, охоронцеві необхідно до нього добігти. Щоб охоронець побіг по найкоротшому шляху, треба повернути охоронця в бік гравця. Таку проблему може вирішити знаками синуса.

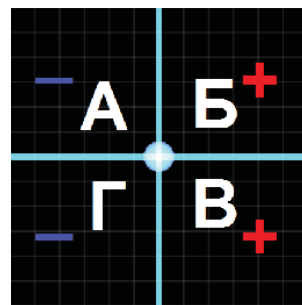


Рисунок 3 – Знаки синуса

Модуль чи абсолютне значення може допомогти, коли розраховується швидкість. Якщо об’єкт рухається вперед, то все нормально. Але можуть виникнути проблеми з пересуванням назад. У нього виходить від’ємна швидкість.

Щоб позбутися від мінуса, але не змінювати напрямки і знак, можна брати абсолютне значення.

### Список літератури

1. Математика Unity3D. 2013 р. - <http://poqxert.ru/page/matematika-unity3d>

## АНАЛИЗ АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ

*Коломиец А. І.*

Вопрос о планировании нагрузки следует решать ещё на ранней стадии развития любого веб-проекта. Первоначально проблемы недостаточной производительности сервера в связи ростом нагрузок можно решать путем наращивания мощности сервера, или же оптимизацией используемых алгоритмов, программных кодов и так далее. Но рано или поздно наступает момент, когда и эти меры оказываются недостаточными.

Приходится прибегать к кластеризации: несколько серверов объединяются в кластер; нагрузка между ними распределяется при помощи комплекса специальных методов, называемых балансировкой. Помимо решения проблемы высоких нагрузок кластеризация помогает также обеспечить резервирование серверов друг на друга [1].

Эффективность кластеризации напрямую зависит от того, как распределяется (балансируется) нагрузка между элементами кластера. Балансировка нагрузки может осуществляться при помощи как аппаратных, так и программных инструментов.

Очень желательно также, чтобы алгоритм балансировки обладал следующими свойствами [2]: **предсказуемость** (эффективность алгоритма для решения поставленных задач); **равномерная загрузка ресурсов системы**; **масштабируемость** (сохранение работоспособности при увеличении нагрузки).

Целью данной работы является анализ основных алгоритмов балансировки нагрузки с указанием их достоинств и недостатков.

Эффективность алгоритмов балансировки нагрузки определяется несколькими показателями, которые представлены ниже [1-2]:

**Пропускная способность** для оценки общего количества задач, которые успешно завершены.

**Затраты** указывают на стоимость процессов, участвующих в решении задачи, перераспределении процессов.

**Отказоустойчивость** измеряет способность алгоритма равномерно выполнять балансировку нагрузки в случае любого сбоя.

**Время миграции** - общее время перехода задачи от одного узла или ресурса к другому.

**Время отклика.** - интервал времени между отправкой запроса и получением ответа.

**Использование ресурсов** надлежащее использование всех ресурсов системы.

**Масштабируемость** - равномерная балансировка нагрузки в системе в соот-