

додаткові дослідження, щоб з'ясувати вплив AR-технологій на користувачів та організації.

На основі розглянутих проблем можна зробити висновок, що хоча розширена реальність відкриває безліч можливостей, у цій галузі також існують виклики, які потребують уваги та досліджень. Важливо зайнятися вирішенням цих проблем, спільно працюючи між галузями та дисциплінами, щоб максимально використовувати потенціал AR та створювати ефективні та безпечні застосунки.

Список використаних джерел

1. Tyagi A. Multimedia and Sensory Input for Augmented, Mixed, and Virtual Reality, Engineering Science Reference / Amit Kumar Tyagi. – 2021. – С. 112-143.

ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ: РЕКУРСІЯ ТА ОПТИМІЗАЦІЯ КІНЦЕВИХ ВИКЛИКІВ.

Сікан А.О.,

здобувач освіти бакалаврського рівня

освітньої програми Середня освіта (Інформатика)

Житомирський державний університет імені Івана Франка

Науковий керівник – доцент, кандидат педагогічних наук Житомирського державного університету імені Івана Франка, Кривонос Олександр Миколайович.

Ви коли-небудь чули про Ханойську вежу? Це відома головоломка, коли у вас є стопка дисків різного розміру та 3 стрижні, які ви повинні перемістити всі диски з одного стартового стрижня на інший. Хоча ця проблема є важкою для більшості людей, коли вони вперше про неї дізнаються, її можна легко вирішити за допомогою рекурсії[1].

Рекурсія є широко використовуваною технікою в програмуванні. Це один із варіантів, який можна використовувати для вирішення проблем «розділяй і володарюй». За визначенням, рекурсія – це метод розв'язання проблеми шляхом її поділу на менші частини. У програмуванні функція є рекурсивною, коли вона викликає саму себе прямо чи опосередковано[1].

Найбільша перевага рекурсивного програмування перед ітеративним програмуванням полягає, в тому що воно забезпечує чистий і простий спосіб написання коду. Оскільки проблема розбивається на менші частини

шляхом виклику тієї самої функції з меншою підмножиною вхідних даних, виникає менше побічних ефектів, і програміст може швидко зрозуміти основну логіку.

Хоча кожна рекурсивна функція може бути записана у формі ітераційної функції, рекурсія потребує більше місця, оскільки вона продовжує додавати функцію в пам'ять стека, доки не буде досягнуто базовий варіант. Рекурсивна функція також потребує більше часу для завершення через накладні витрати на виклики функції та повернення.

Існує 2 типи рекурсії: головна рекурсія та хвостова рекурсія. Тепер розглянемо один простий і популярний приклад: знаходження факторіала числа.

Головна рекурсія

```
func factorial(_x: Int) -> Int {  
  if x == 0 { return 1 }  
  return x* factorial (x-1)}
```

Хвостова рекурсія

```
func factorial(_x: Int) -> Int {  
  func factTail(_x: Int, _ accum: Int) -> Int {  
    if x == 0 { return accum }  
    return factTail(x-1, x*accum)}  
  return factTail(x, 1)}
```

Обидві функції повертають однаковий результат, але з різною реалізацією. Уявіть, що ми хочемо обчислити факторіал числа 6. Визначення факторіала числа – це створення всіх цілих від 1 до цього числа.

$$x! = 1 * 2 * \dots * (x-1) * x$$

Отже, якщо ми хочемо обчислити $x!$, ми обчислюємо $(x-1)!$ і помножити на x . Як бачите, ми використовуємо той самий метод як у головній, так і хвостовій рекурсії[2]. Однак відмінністю є час (порядок) рекурсивного виклику.

Для головної рекурсії ми викликаємо рекурсивну функцію перед множенням.

*повернути x *факторіал $(x-1)$*

У цьому рядку система має обчислити факторіал $x-1$ перед множенням на x і повертає кінцевий результат. Хвостова рекурсія, з іншого боку, обчислює перед викликом сама себе.

*повертає factTail(x-1,x*accum)*

Функція factTail множить x на поточний накопичувач перед входом у рекурсивний виклик.

Хоча більшість часу ми вважаємо що краще використовувати головну рекурсію, оскільки вона простіша та зрозуміліша, хвостова рекурсія може дати нам перевагу в пам'яті. Ось чому ми маємо термін, який називається **оптимізацією кінцевих викликів (TCO)**.

Розберемо розрахунок у конкретному випадку. Наприклад, ми хочемо обчислити факторіал 3.

Головна рекурсія

(fact 3)
(* 3 (fact 2))
(* 3 (* 2 (fact 1)))
(* 3 (* 2(*1 (fact 0))))
(* 3 (* 2 (*1 1)))
(* 3 (* 2 1))
(* 3 2)
6

Для кожного кроку створюється одна нова локальна змінна, яка додається до стеку. Якщо це не реалізовано, пам'ять може експоненціально збільшуватися та призвести до переповнення стеку

Хвостова рекурсія

(fact 3)
(fact 3 1)
(fact 2 3)
(fact 1 6)
(fact 0 6)
6

У цьому прикладі в кожній точці пам'ять стеку потребує лише дбати про 2 змінні (x і accum), що призводить до складності лінійного простору. Для деяких мов програмування ця оптимізація може допомогти значно знизити вартість обробки.

Список використаних джерел

1. Що таке рекурсія та рекурсивний метод? Режим доступу: URL: <https://training.epam.ua/News/Items/419?lang=ua> (дата звернення 17.02.2023)

2. Рекурсивні функції та процедури. Режим доступу: URL: <https://sites.google.com/site/liseyoap/rekursivni-funkciie-ta-proceduri> (дата звернення 17.02.2023)

РОЗРОБКА ДИЗАЙНУ САЙТУ ІНТЕРНЕТ МАГАЗИНУ

Сіциліцина Антоніна Юрївна
здобувачка вищої освіти першого (бакалаврського) рівня
спеціальності 122 Комп'ютерні науки
Мелітопольський державний педагогічний університет
імені Богдана Хмельницького

Анотація. Розглянуто огляд розробки дизайну сайту інтернет магазину за допомогою таких програмних засобів як Adobe Photoshop, Adobe Illustrator та Figma. Вказані переваги та недоліки відповідних засобів.

Design development of the internet store site. Abstract. An overview of the development of the design of the website of the online store using such software tools as Adobe Photoshop, Adobe Illustrator and Figma is considered. The advantages and disadvantages of the respective means are indicated.

Зручність використання та корисність, а не візуальний дизайн, визначають успіх чи провал веб-сайту. Оскільки відвідувач сторінки — це єдина особа, яка клацає мишкою і, отже, вирішує все, дизайн, орієнтований на користувача, став стандартним підходом до успішного та орієнтованого на прибуток веб-дизайну. Зрештою, якщо користувачі не можуть користуватися функцією, вона може не існувати [1].

Розробка дизайну сайту інтернет-магазину є надзвичайно важливим етапом в процесі створення та успішної експлуатації електронної комерції. Ми виділили декілька причин, чому розробка дизайну сайту інтернет-магазину є настільки важливою.

Дизайн веб-сайту інтернет-магазину відіграє роль першого враження, яке отримує відвідувач. Візуально привабливий та професійно оформлений дизайн здатний привернути увагу, створити позитивне враження та зацікавити потенційних покупців. Хороший дизайн допомагає створити зручне та інтуїтивно зрозуміле користувацьке середовище. Навігація, розташування елементів, візуальні індикації та інші аспекти дизайну впливають на зручність використання сайту і забезпечують позитивний