МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ЗВ'ЯЗКУ ТА ІНФОРМАТИЗАЦІЇ
ОДЕСЬКОЇ НАЦІОНАЛЬНОЇ АКАДЕМІЇ ЗВ'ЯЗКУ ІМ. О. С. ПОПОВА

Десята студентська науково-технічна конференція

ТЕЛЕКОМУНІКАЦІЙНІ, ІНФОРМАЦІЙНІ ТА КОМП'ЮТЕРНІ МЕРЕЖІ І СИСТЕМИ: ТЕПЕРІШНЄ ТА МАЙБУТНЄ

12 вересня 2020 року

Збірка тез

Одеса, 2020

Телекомунікаційні, інформаційні та комп'ютерні мережі і системи: теперішнє та майбутнє: матеріали десятої студентської науково-технічної конференції, м. Одеса, 12 вересня 2020 року – Одеса, ВСП «ФКЗІ ОНАЗ ім. О.С. Попова», 2020 – 32 с.

Дана збірка містить тези матеріалів, що представлені на десятої студентській науково-технічній конференції «**Телекомунікаційні, інформаційні та комп'ютерні мережі і системи: теперішнє та майбутнє»**, що проводиться 12.09.2020 р. у відокремленому структурному підрозділі «Фаховий коледж зв'язку та інформатизації Одеської національної академії зв'язку ім. О.С. Попова».

До збірки включені тези доповідей за секціями:

– автоматизації, телекомунікацій та радіотехніки;
– інформаційних технологій;
– комп'ютерних технологій та захисту інформації.

Робочі мови конференції – українська, англійська.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

ГОЛОВА ОРГКОМІТЕТУ:

Петрусенко С. Ю. – к.п.н., директор коледжу зв'язку та інформатизації ОНАЗ ім. О. С. Попова

ЗАСТУПНИК ГОЛОВИ ОРГКОМІТЕТУ:

Горлінська О. Ю. – заступник директора з навчальної роботи

СЕКРЕТАР ОРГКОМІТЕТУ:

Гуменна О. А. – завідувач навчально-методичного кабінету

ЧЛЕНИ ОРГКОМІТЕТУ:

Ящишина І. Я. – заступник директора з навчально-виховної роботи

Владова Т. С. – завідувач денного відділення, викладач циклової комісії телекомунікацій та радіотехніки;

Кокорєва З. Р. – голова циклової комісії інтелектуальних технологій та зв'язку

Кокорєв О. В. – к.філ.наук, викладач циклової комісії інтелектуальних технологій та зв'язку

Орлва Л. Б. викладач циклової комісії інтелектуальних технологій та зв'язку;

Малюта С. О. – викладач циклової комісії інтелектуальних технологій та зв'язку

Д'яченко А. І. – голова ради студентського самоврядування, студентка групи З-41

ЖУРІ КОНФЕРЕНЦІЇ

ГОЛОВА ЖУРІ:

Каптур В. А. – к.т.н, с.н.с., проректор з наукової роботи ОНАЗ ім. О. С. Попова

ЧЛЕНИ ЖУРІ:

секція автоматизації, телекомунікацій та радіотехніки

Маковеєнко Д. О. – к.т.н., доцент, доцент кафедри телебачення та радіомовлення ОНАЗ ім. О. С. Попова;

Баляр В. Б. – к.т.н., старший викладач кафедри телебачення та радіомовлення ОНАЗ ім. О. С. Попова

секція інформаційних технологій

Кумиш В. Ю. – к.т.н, начальник НДЧ ОНАЗ ім. О. С. Попова

Царьов Р. Ю. – старший викладач кафедри мереж зв'язку ОНАЗ ім. О. С. Попова

секція комп'ютерних технологій та захисту інформації

Кільдішев В. Й, – к.т.н., доцент кафедри кібербезпеки та технічного захисту інформацій ОНАЗ ім. О. С. Попова

Стайкуца С. О. – к.ф.н., доцент кафедри кібербезпеки та технічного захисту інформацій ОНАЗ ім. О. С. Попова

ЗМІСТ

Секція автоматизації, телекомунікацій та радіотехніки

1. Бойштян О. О. Новітні технології в створенні дисплейних пристроїв 6

2. Харламов А. А. Augmented reality as part of life of a modern person 10

3. Васильєва К. Ю., Москаленко Т. Д. Дослідження особливостей поширення електромагнітного поля в середині приміщення 12

4. Червоненко Б. М., Зарейчук І. С. Development and comparative analysis between general purpose data transfer protocols and highly specialized protocols for embedded devices

Секція інформаційних технологій

1. Дріга Б. В. ПРозробка WEB-додатку для обміну повідомленнями з використанням бібліотеки VueJs та програмної платформи Node.js 16

2. Руснак Д. В., Свіщ М. О. Порівняння ігрових рушіїв Unity та Unriol Engine на прикладі розробки гри 18

3. Смірнов С. В. Створення інді гри на ігровому рушії Unity 20

Секція комп'ютерних технологій та захисту інформації

1. Басов С. В. Квантовий криптоаналіз двоключових криптосистем 23

2. Жук О. І. Посилення криптостійкості шифру заснованого на псевдовипадкових генераторах хаосу 24

3. Левченко Д. В., Пузанов К. О. Розроблення нейронної мережі в інтерфейсі Python 27

СЕКЦІЯ АВТОМАТИЗАЦІЇ, ТЕЛЕКОМУНІКАЦІЙ ТА РАДІОТЕХНІКИ

НОВІТНІ ТЕХНОЛОГІЇ В СТВОРЕННІ ДИСПЛЕЙНИХ ПРИСТРОЇВ

Бойштян О. О , студент 4-го курсу, групи Тр-41, галузь знань 17 Електроніка та телекомунікації, спеціальність 172 Телекомунікації та радіотехніка.

Науковий керівник – викладач **Борисенко В. С.**

Фаховий коледж зв'язку та інформатизації ОНАЗ ім. О. С. Попова.

Анотація. *Розглянуто актуальні технології у виробництві дисплеїв LCD, TFT, IPS, OLED, AMOLED, POLED, FAMOLED, Micro LED.*

Мета роботи – порівняння переваг та недоліків дисплеїв з рідкими кристалами (РК) та дисплея з органічним світлодіодом (OLED). AMOLED і IPS - переваги та недоліки. Micro LED порівняно з РК –дисплеєм та OLED.

Технологія відображення поступово, але глибоко сформувала спосіб життя людини, який широко визнаний як невід'ємна частина сучасного світу. На сьогоднішній день рідкокристалічні дисплеї (РК-дисплеї) є технологією, де додатки охоплюють смартфони, планшети, комп'ютерні монітори, телевізори. Однак в останні роки ринок органічних світлодіодних дисплеїв (OLED) швидко зростає і почав кидати виклик РК-дисплеям у всіх програмах. LCD, TFT, IPS, OLED, AMOLED, POLED – це список позначень технологій, які використовують у виробництві дисплеїв. Вибираючи смартфон або дисплей, потрібно враховувати один з найголовніших параметрів - тип дисплея.

LCD - плоский дисплей з рідкими кристалами (РК-дисплей). LCD – узагальнене позначення категорії екранів, до якої належить і TFT. TFT ж являє собою панель на основі LCD, але при виготовленні такої панелі використовують транзистори, які відносяться до типу тонкоплівкових. І це єдина її відмінність від інших РК версій. Для створення кольорової картинки в LCD-матриці використовується масив пікселів. За роботу кожного пікселя відповідає три транзистори (червоний, зелений і синій) та конденсатор. Комбінація трьох субпікселів (червоного, синього та зеленого) в кожному пікселі дозволяє формувати будь-який інший відтінок або колір.

TN-TFT та IPS – це технології виготовлення екранів з використанням рідких кристалів і тонкоплівкових транзисторів. В IPS-технології практично такий же принцип, як і в TN-TFT, лише кристали спочатку розташовуються паралельно один до одного в одній площині з електродами. Якщо напруга відсутня, кристали не рухаються. Технологія IPS зробила LCD-дисплеї якіснішими, дозволила покращити кути огляду, відтворення кольорів і підвищити контрастність. Ця технологія

виявилася більш вдалою в порівнянні з TN-TFT, і її продовжують удосконалювати. IPS-технологія використовується в дисплеях, відомих під назвою Retina. Назву Retina запатентовала компанія Apple, але це не є назвою підвиду технології IPS. Різниця між звичайним дисплеєм і дисплеєм Retina: подвоїна кількість пікселів по горизонталі і по вертикалі. Проблеми з мініатюризацією вирішили за допомогою створення спеціальної системи HiDPI.

Перші AMOLED-дисплеї створювали надто яскраву картинку, яка спричиняла спазм очей. Згодом розробники усунули практично усі ці проблеми. Сучасні AMOLED-матриці відтворюють кольори максимально наближені до натуральних, і більше не спричиняють напруження очей. Також у порівнянні з LCD вони тонші, що дозволяє зменшити товщину самого смартфона.

POLED. Після доопрацювання OLED-технології екрани на діодах отримали великий попит. По-перше, вони не використовують заряд батареї, як LCD-дисплеї. По-друге, їх використовують для безрамкових і вигнутих смартфонів. Для встановлення цих дисплеїв у вигнуті смартфони скляну підкладку замінили на пластикову, після чого їх стали називати POLED, де перша буква P означає пластик (plastic). Екрани, виготовлені за технологією POLED, коштують дорожче і тонші за AMOLED-екрани.

AMOLED і IPS - переваги та недоліки. Зараз на ринку змагаються дві провідні технології з виробництва дисплеїв – OLED і LCD, а AMOLED і IPS -це вже їхні вдосконалені варіанти. Серед усіх ЖК-дисплеїв найкращими вважаються IPS. AMOLED і IPS – це абсолютно різні технології з різною орієнтацією на результат.

OLED-дисплей (органічний світлодіод) – це емісійний дисплей, на відміну від РК-дисплея. В даний час зазвичай використовуються багатошарові структури в OLED з різними функціональними матеріалами. Як правило, кожен шар в OLED досить тонкий. Таким чином, OLED є ідеальним кандидатом на гнучкі дисплеї.

FAMOLED (Flexible display) - це екран, який вважається гнучким і здатний змінювати свою форму. Нові дисплеї виготовлені на органічних світлодіодах, а деформуються вони за рахунок використання пластикової полімерної підкладки, а не скла. Чим тонший дисплей, тим він гнучкіший Технологія OLED використовує самопідсвічуванні пікселі для точного налаштування яскравості та якості зображення, а завдяки відсутності задньої підсвітки нові дисплеї надзвичайно легкі, тонкі та гнучкі.

Micro LED також відомий як «LED» або «micro LED», який складається з світлодіодів. Таким чином, OLED також є екраном власної світиться технології. Суть цієї технології полягає в мікросрежденіі і матричному світлодіоді, а також для скорочення рівня світлодіодного блоку до рівня менше 50 мікрон, щоб реалізувати

власну люмінесценцію одиночного точкового приводу. В даний час технологія основного екрану має два види TFT LCD і OLED.

Висновки

У кожної технології є свої плюси і мінуси. Технологія IPS зробила LCD-дисплеї якіснішими, дозволила покращити кути огляду, відтворення кольорів і підвищити контрастність. IPS-технологія використовується в дисплеях, відомих під назвою Retina. Назву Retina запатентовала компанія Apple, але це не є назвою підвиду технології IPS. Зараз на ринку змагаються дві провідні технології з виробництва дисплеїв - OLED і LCD, а AMOLED і IPS - це вже їхні вдосконалені варіанти. OLED-дисплеї є емісійними; їхні переваги : справжній чорний колір, швидкий час відгуку та надто тонкий профіль, що дозволяє зробити гнучкі екрани. Конкуренція на ринку та технічний прогрес, як відомо, не стоїть на місці.

Перелік посілань

1. Рідкокристалічний дисплей та органічний світлодіодний дисплей: сучасний стан та перспективи майбутнього | світло: наука та програми https://uk.mediatekstore.com/liquid-crystal-display

2. Сучасні технології виготовлення дисплеїв// https://all-spares.ua/uk/articles-and-video/modern-technologies-of-manufacturing-displays.html 29.08.2018

3.Томилин М.Г., Невская Г.Е. Дисплеи на жидких кристаллах – СПб: СПбГУ ИТМО, 2010. – 108 с.

4. Дисплеи на органических светодиодах //https://a-golubev.ru/tehnologii/displei-na-organicheskih-svetodiodah-oled.html .

5. LCD, OLED и фундаментальные принципы квантовых точек// http://ru.panasyslcd.com/news/lcd-oled-and-the-fundamental-principles-of-qu-13307005.html

6. 3 популярных виды матриц: IPS, PLS И TN// https://www.moyo.ua/news/3_populyarnyh_vida_matric_ 17.08.2018

7.LCD vs OLED — что же лучше https://www.iguides.ru/main/other/lcd_vs_oled_chto_zhe_luchshe/ Е. Морозов 7 02. 2018

8.В сфері нових технологій. Випуск 25.
- https://ua.gecid.com/ittech/v_oblasti_novyeh_tehnologiyi__vyepusk_25/

9. Что такое Retina дисплеи//https://observer.com.ua/chto-takoe-retina-displei/

10. Как все начиналось: гибкие и складные дисплеи — история появления и выход «в люди»// https://habr.com/ru/post/455268/7.06. 2019 .

11. Что такое микро светодиод?// http://ru.panasyslcd.com/news/what-is-the-microled-screen-technology-for-app-17874924.html

AUGMENTED REALITY AS PART OF LIFE OF A MODERN PERSON

Kharlamov A. A. 4rd year student, group TO-41(graduated), course “172”
Scientific Supervisor – teacher KokorevO. V.

Annotation. *The main goal of this development is to prove that costly hardware for displaying augmented reality technology is inferior in functionality to self-assembled devices.*

Augmented reality (AR) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory and olfactory.

Augmented reality is the technology that expands our physical world, adding layers of digital information onto it. Unlike Virtual Reality (VR), AR does not create the whole artificial environments to replace real with a virtual one. AR appears in direct view of an existing environment and adds sounds, videos, graphics to it.

A view of the physical real-world environment with superimposed computer-generated images, thus changing the perception of reality, is the AR.

The term itself was coined back in 1990, and one of the first commercial uses were in television and military. With the rise of the Internet and smartphones, AR rolled out its second wave and nowadays is mostly related to the interactive concept. 3D models are directly projected onto physical things or fused together in real-time, various augmented reality apps impact our habits, social life, and the entertainment industry.

AR apps typically connect digital animation to a special ‘marker', or with the help of GPS in phones pinpoint the location. Augmentation is happening in real time and within the context of the environment, for example, overlaying scores to a live feed sport events.

For AR a certain range of data (images, animations, videos, 3D models) may be used and people will see the result in both natural and synthetic light. Also, users are aware of being in the real world which is advanced by computer vision, unlike in VR.

AR can be displayed on various devices: screens, glasses, handheld devices, mobile phones, head-mounted displays.

AR involves the S.L.A.M. technology (simultaneous localization and mapping), depth tracking (briefly, a sensor data calculating the distance to the objects), and the following components:

– Cameras and sensors. Collecting data about user's interactions and sending it for processing. Cameras on devices are scanning the surroundings and with this info, a device locates physical objects and generates 3D models. It may be special duty cameras, like in Microsoft Hololens, or common smartphone cameras to take pictures/videos.

– Processing. AR devices eventually should act like small computers, something modern smartphones already do. In the same manner, they require a CPU, a GPU, flash memory, RAM, Bluetooth/WiFi, a GPS, etc. to be able to measure speed, angle, direction, orientation in space, and so on.

– Projection. This refers to a miniature projector on AR headsets, which takes data from sensors and projects digital content (result of processing) onto a surface to view. In fact, the use of projections in AR has not been fully invented yet to use it in commercial products or services.

– Reflection. Some AR devices have mirrors to assist human eyes to view virtual images. Some have an “array of small curved mirrors” and some have a double-sided mirror to reflect light to a camera and to a user's eye. The goal of such reflection paths is to perform a proper image alignment.

The regular price of augmented reality display devices starts at $ 1,000.

To summarize, I tell youthat in my report I will demonstrate that a self-assembled device will reduce your budget for acquiring such a device by 60 times using easily accessible materials.

References

1.“What is Augmented Reality (AR) and how does it work”. https://thinkmobiles.com/blog/what-is-augmented-reality/

2. “Innovative Examples of Augmented Reality in Action”. https://www.adobe.com/insights/5-realworld-examples-of-augmented-reality-innovation.html

3. “What is augmented reality?” https://www.fi.edu/what-is-augmented-reality

4. “The 10 most innovative virtual and augmented reality companies of 2020”. https://www.fastcompany.com/90457929/virtual-augmented-reality-most-innovative-companies-2020

5. “Augmented Reality”. https://www.interaction-design.org/literature/topics/augmented-reality

6. “VR, AR, MR and What Does Immersion Actually Mean?” https://www.thinkwithgoogle.com/intl/en-cee/insights-trends/industry-perspectives/vr-ar-mr-and-what-does-immersion-actually-

mean/?gclid=Cj0KCQjwu6fzBRC6ARIsAJUwa2TCf9gkRRf2NEDzA5PyDLmXfBnIyEvZkw4PjafcmPNwydJCJUCfW2gaAlfbEALw_wcB

7. “Advancing AR and VR Experiences”. https://www.arm.com/solutions/mobile-computing/ar-vr

8.“VR, AR, MR and What Does Immersion Actually Mean?” https://www.thinkwithgoogle.com/intl/en-cee/insights-trends/industry-perspectives/vr-ar-mr-and-what-does-immersion-actually-mean/?gclid=Cj0KCQjwu6fzBRC6ARIsAJUwa2TCf9gkRRf2NEDzA5PyDLmXfBnIyEvZkw4PjafcmPNwydJCJUCfW2gaAlfbEALw_wcB.

ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ПОШИРЕННЯ ЕЛЕКТРОМАГНІТНОГО ПОЛЯ В СЕРЕДИНІ ПРИМІЩЕННЯ

Васильєва К.Ю. студентка 3-го курсу, група Тр-31, спеціальність 172 «Телекомунікації та радіотехніка».

Москаленко Т.Д.студентка 3-го курсу, група Тр-31, спеціальність 172 «Телекомунікації та радіотехніка».

Науковий керівник–к.т.н., доц**. Рожновський М.В.**

Коледж зв’язку та інформатизації ОНАЗ ім. О.С. Попова

Анотація.*У роботі проведено аналіз залежності зміни величини потужності сигналу на вході приймача мобільної станції від відстані до точки доступу WI-FI.*

Головна мета– порівняльний аналіз виміряної та розрахованої потужності сигналу на вході приймача мобільної станції в мережі радіодоступу WI-FI та визначення величини похибки між виміряними та розрахованими значеннями.

Системи бездротового зв’язку посіли важливу роль в нашому житті, важко уявити людину, яка не користується мобільним телефоном. На сучасному етапі розвитку систем мобільного зв’язку та інших бездротових мереж, таких як Wi-Fi та WiMAX, важливою проблемою залишається, планування мережі, а саме визначення напруженості поля в певних умовах поширення радіохвиль [2,3]. Для розрахунку втрат сигналу при поширені радіохвиль, використовують статистичні моделі [7].

В даній роботі розглянуто поняття електромагнітного поля, радіохвиль, та проблеми їх поширення в бездротових системах передачі даних Wi-Fi [1]. Розглянуто основні математичні моделі, що використовуються для розрахунку напруженості поля та методика аналізу потужності сигналу на вході приймача мобільної станціїх [5,6]. Запропоновано метод вимірювання потужності електромагнітного поля за

допомогою мобільної станції. Базуючись на розглянутій статистичній моделі COST 231-Hata, та, задавшись вихідними даними для двох різних типів приміщень, проведено розрахунок втрат та потужності сигналу на вході приймача мобільної станції, проведено аналіз отриманих результатів. Проведено експериментальні виміри потужності поля в розглянутих умовах розповсюдження[7]. Проведено аналіз отриманих даних та визначено похибку між розрахованими та експериментальними даними. На основі отриманих результатів аналізу дано рекомендації, які можна використовувати при проектуванні мереж бездротової передачі даних Wi-Fi.

Висновки

В даній роботі:

✓ детально проаналізовано основні специфікації стандарту IEEE 802.11 та визначено місце вказаного стандарту в мережах 5G;

✓ детально проаналізовано модель поширення радіохвиль COST-231 Hata;

✓ запропоновано методику проведення як теоретичних, так і експериментальних досліджень з визначення потужності сигналу на вході приймача мобільної станції в мережі Wi-Fi;

✓ запропоновано тип обладнання та програмне забезпечення для проведення досліджень;

✓ проведено порівняльний аналіз розрахованих та виміряних значень потужності сигналу на вході приймача мобільної станції в мережі Wi-Fi;

✓ на основі результатів досліджень сформовано наступні рекомендації:

– модель COST-231 Hata може використовуватись для планування мереж Wi-Fi в діапазоні частот 2,4 ГГц;

– при використані моделі COST-231 Hata для планування мереж Wi-Fi необхідно враховувати, що вказана модель дає похибку в межах від 11,5 до 14,15 %.

Перелік посилань:

1. 4. Гавриленко В.Г. Распространение радиоволн в современных системах мобильной связи/Гавриленко В.Г. – Нижний Новгород, 2003. – 148 с.

2. IEEE 802.11TM WIRELESS LOCAL AREA NETWORKS [електронний ресурс]. Режим доступу: http://www.ieee802.org/11/

3. Современные беспроводные сети: состояние и перспективы развития / [Гепко И.А., Олейник В.Ф., Чайка Ю.Д., Бондаренко А.В.]; под ред. В.Ф. Олейника. — К.: «ЕКМО», 2009. — 672 с.

4. Netmonitor [електронний ресурс]. Режим доступу: https://play.google.com/store/apps/details?id=com.parizene.netmonitor&hl=ru

5. TP-LINK TL-WR340GD [електронний ресурс]. Режим доступу: https://www.technotrade.com.ua/Products/tp-link-tl-wr340gd.php .

6. Tenda W316R [електронний ресурс]. Режим доступу: https://www.dnsshop.ru/product/d520fe0ae3ef3361/marsrutizator-tenda-w316r/

7. Mardeni R. Optimised COST-231 Hata Modelsfor WiMAX Path Loss Predictionin Suburbanand Open Urban Environments / R. Mardeni, T. SivaPriya // ModernAppliedScience. – 2010. – VOL. 4, № 9 – P. 75 – 89.

DEVELOPMENT AND COMPARATIVE ANALYSIS BETWEEN GENERAL PURPOSE DATA TRANSFER PROTOCOLS AND HIGHLY SPECIALIZED PROTOCOLS FOR EMBEDDED DEVICES

Chervonenko Boris, 4th year student, group TR-41, course “172 – Telecommunication and Radio Technology”

Zareychuk Ivan, 2nd year student, group TR-21, course “172 – Telecommunication and Radio Technology”

Scientific Supervisor – teacher **Beldugina Svetlana**

Annotation. *In this study we will explore in-depth details behind data protocols and compare highly specialized data transfer protocols that we developed with general purpose data transfer protocols.*

Data transfer protocol is a system of rules that allow two or more entities of a communications system to transmit information via any kind of variation of a physical quantity. The protocol defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods. Protocols may be implemented by hardware, software, or a combination of both.

The information exchanged between devices through a network or other media is governed by rules and conventions that can be set out in communication protocol specifications. The nature of communication, the actual data exchanged and any state-dependent behaviors, is defined by these specifications. In digital computing systems, the rules can be expressed by algorithms and data structures. Protocols are to communication what algorithms or programming languages are to computations.

To implement a networking protocol, the protocol software modules are interfaced with a framework implemented on the machine's operating system. This framework implements the networking functionality of the operating system. When protocol algorithms are expressed in a portable programming language the protocol software may be made operating system independent. The best-known frameworks are the TCP/IP model and the OSI model.

There are two types of protocols in regards of what they're developed for: Highly Specialized and General Purpose protocols. By developing our own highly specialized protocol, we can compare them in a chart below:

	General-purpose protocols	**Specialized protocols**
Difficulty to implement	Usually very difficult or impossible to implement in resource-restricted systems	Can be shaved down to match resource requirements
Versatility	As stated in the name, general purpose protocols are developed without focusing on a particular use case and usually require additional work to suit the communication requirements	A protocol of such type can be developed to suit the needs of communication in a given system
Hardware requirements	Often require additional, and sometimes costly, hardware to work properly or even work at all	Can often be made to work on any system, no matter how specialized it is
Complexity	Aside from implementation difficulty, protocols of this type are often very complex and require a lot of time to even understand the core principle	Specialized protocols include only what the system it is used in requires and in most times, are far easier to understand and implements
Availability to other people	General-purpose protocols are mostly made for public use and thus have to comply with regulations. When a protocol is public to use and implement, it may pose a risk to the systems in which it is used	Specialized protocols often developed for a specific system and so they can be made private, which improves the security of the aforementioned systems
Difficulty to	General purpose have to include	A specialized protocol includes

develop	most use cases on at least a basic level and comply with regulations, thus they require a lot of work to develop to be redundant enough	only a specific use case and usually can be developed very quickly with the same redundancy

Chart 1.1 – Comparison between highly specialize data transfer protocols and highly specialize protocols.

In summary, protocols are critical for IT data communication. The development of highly specialized protocols is crucial to optimizing communication between devices. They are less complex, and are generally better for the system it is developed for, among other advantages.

REFERENCES:

1.https://en.wikipedia.org/wiki/Communication_protocol

2.https://www.researchgate.net/publication/316169053_Data_Communication_Protocols

3.https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.701.7521&rep=rep1&type=pdf

4.https://www.techopedia.com/definition/25705/communication-protocol

5.https://www.selfgrowth.com/articles/data-communication-protocols-and-functions-of-communication-protocols

6.https://www.lifewire.com/definition-of-protocol-network-817949

7.https://www.lifewire.com/computer-networks-protocols-817374

8.https://www.britannica.com/technology/protocol-computer-science

9.https://courses.cs.washington.edu/courses/csep590a/06au/projects/standards-wars.pdf

СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ ОБМІНУ ПОВІДОМЛЕННЯМИ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ VUEJS ТА ПРОГРАМНОЇ ПЛАТФОРМИ NODE.JS.

Дріга Б. В., студент 4-го курсу, група З-41, спеціальність 121 «Інженерія програмного забезпечення»

Науковий керівник – викладач **Козловська В.П.**, кандидат технічних наук, доцент

Анотація. *Веб-розробка – це один з напрямків програмування, що швидко розвивається. Постійно з'являються нові технології, бібліотеки, фреймворки. Головне завдання доповіді - зацікавити слухача на прикладі створення веб-додатку для обміну миттєвими повідомленнями. Доповідь складається з декількох частин.*

В веб-розробці є 4 частини. Це фронтенд – візуальна частина, бекенд – серверна частина, DevOps – оптимізація процесу розробки, а також тестування – забезпечення якості продукту. Всі ці напрямки тісно пов'язані один з одним. В процесі наукової роботи реалізується месенджер, головною функцією якого є обмін миттєвими повідомленнями, а саме: відправка, отримання повідомлень, оповіщення в реальному часі про отримання повідомлення тощо.

У ролі серверної частини виступає Node.js, який забезпечує API-частину проекту. Так само він працює в зв'язці з веб-сокет частиною. Сервер використовує базу даних MongoDB.

Node.js – платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Засновником платформи є Райан Дал (Ryan Dahl). Якщо раніше Javascript застосовувався для обробки даних в браузері користувача, то Node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

У ролі клієнтської частини програми виступає додаток, зроблений за допомогою фреймворка Vue.js в зв'язці зі сховищем даних - VueX і бібліотекою візуальних компонентів – Vuetify.

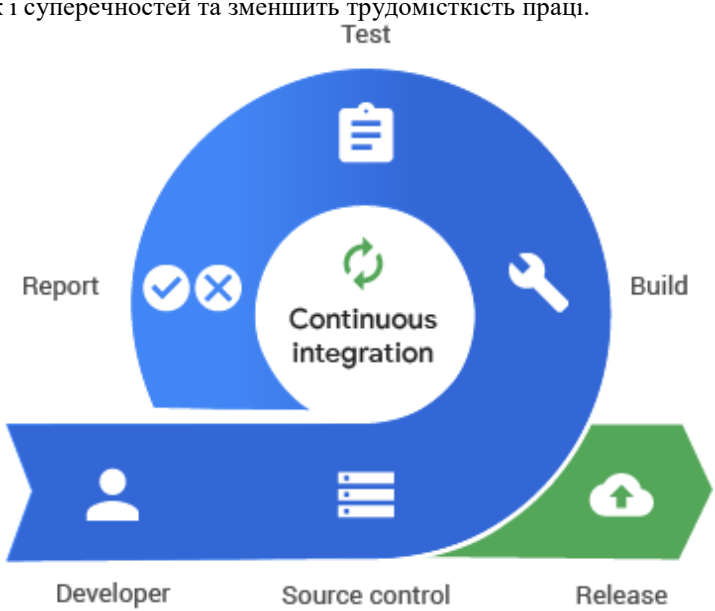
Vue.js (читається як "в'ю", з англ. view) – JavaScript-фреймворк, що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних через реактивне зв'язування даних.

У розділі про DevOps мова йде про такі технології, як Docker, метод постійної інтеграції і доставки оновлень програми, а також про сервіс AWS і про реалізацію CI/CD в сервісі GitLab та задачі DevOps- інженера.

Docker – інструментарій для управління ізольованими Linux-контейнерами. Docker доповнює інструментарій LXC більш високорівневим API, що дозволяє керувати контейнерами на рівні ізоляції окремих процесів. Зокрема, Docker дозволяє, не переймаючись вмістом контейнера, запускати довільні процеси в режимі ізоляції і потім переносити і клонувати сформовані для даних процесів контейнери на інші

сервери, беручи на себе всю роботу зі створення, обслуговування і підтримки контейнерів.

Неперервна інтеграція (англ. Continuous Integration) (рис. 1) – практика розробки програмного забезпечення, яка полягає у виконанні частих автоматизованих складань проекту для якнайшвидшого виявлення та вирішення інтеграційних проблем. У звичайному проекті, де над різними частинами системи розробники працюють незалежно, стадія інтеграції є завершальною. Вона може непередбачувано затримати закінчення робіт. Перехід до неперервної (постійної) інтеграції дозволяє зробити її передбачуваною за рахунок найбільш раннього виявлення та усунення помилок і суперечностей та зменшить трудомісткість праці.



Малюнок 1 - Схема роботи циклу неперервної інтеграції

Висновок. Було проведено аналіз існуючих технологій і вибрано з переліку відповідні. Завдяки проведеній науковій роботі було збудовано веб-додаток для обміну повідомленнями.

Перелік посилань:

1. https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BF%D0%B5%D1%80%D0%B5%D1%80%D0%B2%D0%BD%D0%B0_%D1%96%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D1%96%D1%8F
2. https://uk.wikipedia.org/wiki/Docker
3. https://uk.wikipedia.org/wiki/Vue.js
4. https://uk.wikipedia.org/wiki/Node.js

ПОРІВНЯЛЬНИЙ АНАЛІЗ ІГРОВИХ РУШІЇВ

Руснак Даніїл, Свіщ Максим, студент 2-го курсу, група З-21, спеціальність 121 «Інженерія програмного забезпечення»

Науковий керівник – викладач **Малюта С.О.**

Коледж зв'язку та інформатизації Одеської національної академії зв'язку ім. О. С. Попова

Анотація. *Проводиться аналіз, мета якого – порівняти сучасні ігрові рушії, які використовуються для розробки ігор.*

Зараз дуже активно розвивається розробка комп'ютерних ігор. Для цього використовують ігрові рушії, великі кампанії розроблюють свої ігрові рушії, але в цілях зберігання коштів деякі все ж таки користуються готовими рішеннями, наприклад: Unreal Engine 4 або Unity 3D. Тоді що казати про інді розробників, якщо вони бажають досягти успіхів вони скоріше всього будуть використовувати готовий ігровий рушій. Таких готових рушіїв досить багато : Delta3D, Game Blender, Game Maker, Illusion Engine, SCUMM, CryEngine 2, Eclipse Engine, 4A Engine, Source 2. В цьому місці ми наближуємося до головної мети нашого докладу, ми вирішили порівняти найпопулярніші ігрові рушії Unreal Engine 4 та Unity.

Ігровий рушій - програмний рушій, центральна програмна частина будь-якої відеогри, яка відповідає за всю її технічну сторону, дозволяє полегшити розробку гри шляхом уніфікації та систематизації її внутрішньої структури. Важливим значенням рушія є можливість створення багатоплатформових ігор (сьогодні найчастіше одночасно для ПК, PS4 та Xbox One).

Основну функціональність гри зазвичай забезпечує її рушій, до якого входить рушій рендерингу («візуалізатор»), фізичний рушій, звук, система скриптів, анімація, ігровий штучний інтелект, мережевий код, керування пам'яттю, багатонитевість і граф сцени. Часто на процесі розробки можна заощадити шляхом повторного використання одного рушія гри для створення декількох різних ігор.

Unity — багатоплатформовий інструмент для розробки дво- та тривимірних додатків та ігор, що працює на операційних системах Windows і OS X. Створені за допомогою Unity застосунки працюють під системами Windows, OS X, Android, Apple iOS, Linux, а також на гральних консолях Wii, PlayStation 3 і XBox 360.

Є можливість створювати інтернет-додатки за допомогою спеціального під'єднуваного модуля для браузера Unity, а також за допомогою

експериментальної реалізації в межах модуля Adobe Flash Player. Застосунки, створені за допомогою Unity, підтримують DirectX та OpenGL.

Unreal Engine — ігровий рушій, розроблюваний і підтримуваний компанією Epic Games.

Перша гра, створена на цьому рушію — Unreal, яка з'явилася 1998 року. З тих пір різні версії цього ігрового рушія використали в більш ніж сотні ігор, серед яких Deus Ex, Lineage II, Thief: Deadly Shadows, Postal 2, серіях ігор Brothers in Arms, серія ігор Splinter Cell, Tom Clancy's Rainbow Six, а також у відомих ігрових серіях Unreal і Unreal Tournament від самих Epic Games. Пристосований у першу чергу для шутерів від першої особи, рушій використовувався й при створенні ігор інших жанрів.

Написаний мовою C++, рушій дозволяє створювати ігри для більшості операційних систем і платформ: Microsoft Windows, Linux, Mac OS і Mac OS X, консолей Xbox, Xbox 360, PlayStation 2, PlayStation Portable, PlayStation 3, Wii, Dreamcast і Nintendo GameCube. У грудні Марк Рейн продемонстрував роботу рушія Unreal Engine 3 на iPod Touch і iPhone 3GS. У березні 2010 робота рушія була продемонстрована на комунікаторі Palm Pre, що базується на мобільній платформі webOS.

Для спрощення портування рушій використовує модульну систему залежних компонентів: підтримує різні системи рендерингу (Direct3D, OpenGL, Pixomatic; раніше підтримувалися Glide API, S3 Metal, PowerVR SGL), відтворення звуку (EAX, OpenAL, DirectSound3D; раніше підтримувалися A3D), засоби голосового відтворення тексту, розпізнавання мовлення (тільки для Xbox360, PlayStation 3, Nintendo Wii і Microsoft Windows, також планувалося для Linux і Mac), модулі для роботи з мережею й підтримка різних пристроїв вводу.

Висновок. Так як вибір середи розробки гри є дуже важливим етапом, саме тому нашою метою є порівняння цих рушіїв на базі розробки гри, цими результатами ми в майбутньому плануємо розмістити в інтернеті, для того щоб початківцям в цьому ремеслі було легше. Це є дуже важливим для початкових розробників, які обирають середу розробки, яка буде простіша для створення задуманого проекту у повному обсязі.

Перелік посилань:

1. https://unity.com/
2. https://www.unrealengine.com/en-US/
3. https://uk.wikipedia.org/wiki/Unreal_Engine

4. https://uk.wikipedia.org/wiki/%D0%86%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9

5. https://uk.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D1%85_%D1%80%D1%83%D1%88%D1%96%D1%97%D0%B2

6. https://uk.wikipedia.org/wiki/Unity_(%D1%80%D1%83%D1%88%D1%96%D0%B9_%D0%B3%D1%80%D0%B8)

СТВОРЕННЯ ІНДІ-ГРИ НА ІГРОВОМУ РУШІЇ UNITY

Смірнов С.В, студент 2-го курсу, групи З-21, спеціальності «Розробка програмного забезпечення».

Науковий керівник – викладач **Орлова Л.Б.**

Коледж зв’язку та інформатизації ОНАЗ ім. О. С. Попова, Україна

Анотація. *Проводиться робота, мета якої – показати процеси та етапи розробки інді-гри на ігровому рушії Unity.*

Ігровий рушій – програмний рушій, центральна програмна частина будь-якої відеогри, яка відповідає за всю її технічну сторону, дозволяє полегшити розробку гри шляхом уніфікації та систематизації її внутрішньої структури. Важливим значенням рушія є можливість створення багатоплатформових ігор.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Рушій підтримує дві сценарні мови: C #, JavaScript (модифікація). Проект в Unity ділиться на сцени (рівні) — окремі файли, що містять свої ігрові простори зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як об'єкти (моделі), так і порожні ігрові об'єкти. Об'єкти, в свою чергу містять набори компонентів, з якими взаємодіють скрипти. Вони мають назву (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), можуть мати тег (мітку) і зображення, на якому об'єкт повинен відображатися. Так, у будь-якого предмета на сцені обов'язково присутній компонент Transform — він зберігає в собі координати місця розташування, повороту і розмірів по всім трьом осям. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою.

Unity підтримує фізику твердих тіл і тканини, фізику типу Ragdoll. У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі

зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів. При імпортуванні текстури в рушій можна згенерувати alpha-канал, mip-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна — буде створено матеріал, з яким буде призначений шейдер, і потім матеріал прикріпиться до моделі. Редактор Unity підтримує написання і редагування шейдерів. Крім того він містить компонент для створення анімації, яку також можна створити попередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розбити на файли.

Як правило, ігровий рушій надає безліч функціональних можливостей, що дозволяють їх задіяти в різних іграх, в які входять моделювання фізичних середовищ, карти нормалей, динамічні тіні і багато іншого. На відміну від багатьох інших рушіїв, у Unity є дві основні переваги: наявність візуальної середовища розробки та міжплатформова підтримка. Перший фактор включає не тільки інструментарій візуального моделювання, а й інтегроване середовище, ланцюжок складання, що направлено на підвищення продуктивності розробників, зокрема, етапів створення прототипів і тестування. Під міжплатформовою підтримкою мається на увазі не тільки місця розгортання (установка на персональному комп'ютері, на мобільному пристрої, консолі і тощо), але і наявність інструментарію розробки (інтегроване середовище може використовуватися під Windows і Mac OS). Третьою перевагою є модульна система компонентів Unity, за допомогою якої відбувається конструювання ігрових об'єктів, коли останні є комбіновані скриньки функціональних елементів. На відміну від механізмів успадкування, об'єкти в Unity створюються за допомогою об'єднання функціональних блоків, а не приміщення в вузли дерева успадкування. Такий підхід полегшує створення прототипів, що актуально при розробці ігор.

Недоліками виступають обмеження візуального редактора при роботі з багатокомпонентними схемами, коли в складних сценах візуальна робота ускладнюється. Другим недоліком називається відсутність підтримки Unity на посилання із зовнішньої бібліотеки, роботу з якими програмістам доводиться налаштовувати самостійно, і це також ускладнює командну роботу. Ще один недолік пов'язаний з використанням шаблонів примірників (англ. Prefabs). З одного боку, ця концепція Unity пропонує гнучкий підхід візуального редагування об'єктів, але з іншого боку, редагування таких шаблонів є складним. Також, WebGL-версія рушія, в силу специфіки своєї архітектури (трансляція коду з C # в C ++ і далі в JavaScript), має ряд невирішених проблем з продуктивністю, споживанням пам'яті і працездатністю на мобільних пристроях.

Висновки

За результатами роботи можна зробити наступні висновки:

1) Unity є одним за найкращих ігрових рушіїв для створення однією людиною, або невеликою командою, інді-ігор.

2) Unity дозволяє створювати ігри для більш ніж для 25 платформ, що робить

3) Ігровий рушій має сучасний рівень графіки, що дозволяє створювати ігри AAA.

Перелік посилань:

1. https://itproger.com/news/69 “Обзор и изучение игрового движка Unity”
2. https://itkeys.org/about-unity/ “О Unity: обзор движка”
3. https://uk.wikipedia.org/wiki/Ігровий_рушій “Ігровий рушій”
4. https:// uk.wikipedia.org/wiki/Unity_(рушій_гри) “ Unity(ігровий рушій)”/

КВАНТОВИЙ КОМП'ЮТЕР ТА КВАНТОВІ РОЗРАХУНКИ

Басов С. В. студент 4-го курсу, групи З-41, спеціальність «Розробка програмного забезпечення»

Науковий керівник – к.т.н., доц., Басов Віктор Євгенійович,

кафедра кібербезпеки та технічного захисту інформації ОНАЗ ім. О. С. Попова.

Анотація. *Розглянуто потенціал використання квантових комп'ютерів для вирішення складних математичних завдань. Розглянуті доступні на даний момент зразки квантових комп'ютерів та історія їх розвитку. Розроблено програму-симулятор квантового комп'ютера.*

Протягом останніх років, було створено спеціалізований квантовий комп'ютер D-Wawe 2000-Q який містить 2048 кубітів, та універсальний квантовий комп'ютер IBM Q System One який містить 20 кубітів [i, ii]. На ґрунті цього, можна припустити, що поява повноцінного універсального квантового комп'ютера є справою недалекого майбутнього, тому сучасним програмістам доцільно вже зараз зайнятися вивченням програмування квантових комп'ютерів, щоб не відстати від технічного прогресу. Для вирішення цього завдання, було створено власний програмний стимулятор квантового комп'ютера.

Квантові комп'ютери, базуючись на абсолютно відмінних, від класичних цифрових комп'ютерів, фізичних принципах, дають надзвичайно великий приріст швидкості розрахунку деяких алгоритмів [iii]. Одним з цих принципів є квантовий паралелізм, що проявляється у здатності системи кубітів знаходитися в усіх своїх можливих станах одночасно, що дозволяє провести серію однотипних розрахунків не послідовно, а одночасно, і кількість таких розрахунків обмежено лише кількістю кубітів, що ми задіяли у нашій системі.

На жаль, квантові комп'ютери дозволяють прискорити далеко не всі класичні алгоритми, тому структура квантового комп'ютера, в наш час уявляється у такому вигляді, що до звичайного класичного комп'ютеру підключається «квантовий сопроцесор», та класичний комп'ютер забезпечує підготовку даних та налаштування для квантової частини, а також отримання результатів розрахунків по завершенні квантових обчислень. А на додаток до всього, квантові алгоритми видають вірний результат з певною вірогідністю, що вимагає також перевіряти ці дані, тобто квантовий розрахунок для надійності треба виконувати декілька разів.

Складність розуміння фізичних принципів, а також необхідність створення складних програм, здатних до виконання квантовими комп'ютерами, спричинило появу квантових мов програмування, таких як Q# та QCL, створених на основі існуючих високорівневих мов програмування [iv]. Вони містять у собі як класичні оператори, так і команди, які відповідають за побудову відповідної квантової схеми. Саме завдяки створенню цих мов, програмісти здатні програмувати квантові комп'ютери, не вдаючись у фізичні принципи їх функціонування.

Основною проблемою сучасних квантових комп'ютерів є надзвичайна складність цих машин у створенні та експлуатації. Відсутність необхідних технологій, є основною причиною що пройшло більше 30 років між зародженням ідеї, та створенням першого універсального квантового комп'ютера. Зокрема, це стосується і цифрових комп'ютерів, які використовуються для контролю квантового процесора, та аналізують отриманні у наслідок розрахунків дані.

Створення у 1994 році Пітером Шором однойменного алгоритму[v], створило прецедент, коли двоключові криптосистеми можуть стати не ефективними у продовж найближчого часу. Саме це, доволі практичне застосування квантового комп'ютера, стало причиною надзвичайного інтересу не тільки наукової спільноти, але й спецслужб розвинених держав, що й забезпечило значне фінансування проектів розробки квантових комп'ютерів.

Висновок. У процесі роботи було створено програмну модель алгоритму RSA для чисел, що не перевищують 2^{32}, стимулятор квантового комп'ютера та модель атаки на алгоритм RSA за алгоритмом Шора, з використанням стимулятора квантового комп'ютера.

У результаті виконаного дослідження одержані наступні результати:

- створена діюча модель алгоритму RSA;
- створений стимулятор квантового комп'ютера;
- реалізована атака на RSA алгоритмом Шора за допомогою стимулятора квантового комп'ютера;
- отримані програмні інструменти дозволять у найближчому майбутньому моделювати інші дослідження в квантових обчислюваних системах.

Перелік посилань

1. Квантовый компьютер D-Wave 2000Q[Электронный ресурс]. – Режим доступа: http://information-technology.ru/news/6263-kvantovyj-kompyuter-d-wave-2000q-za-15-mln-dollarov.

2. IBM Q System One – первый квантовый компьютер для коммерческого использования [Электронный ресурс]. – Режим доступа: https://ru.gecid.com/news/ibm_q_system_one_pervyj_kvantovyj_kompyuter/

3. Quantum Algorithm Zoo [Электронный ресурс]. – Режим доступа: http://quantumalgorithmzoo.org/

4. Квантовые вычисления и функциональное программирование // Душкин Р. В. – 2014 – сс 111.

5. Algorithms for Quantum Computation: Discrete Logarithms and Factoring /Shor P//. Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on — 1994. — P. 124–134.

Посилення криптостійкості шифру заснованого на псевдовипадкових генераторах хаосу

Жук О.І., студент 4-го курсу, групи К-41, спеціальність «Комп’ютерні науки та інформаційні технології».

Науковий керівник – д.т.н., доц., **Корчинський В.В.** кафедра кібербезпеки і технічного захисту інформації ОНАЗ ім. О. С. Попова

Анотація. *Метою даної наукової роботи є посилення криптостійкості шифру заснованого на псевдовипадкових генераторах хаосу.*

Криптографія є одним з найбільш потужних засобів забезпечення конфіденційності і контролю цілісності інформації. Багато в чому вона займає центральне місце серед програмно-технічних регуляторів безпеки. Наприклад, для портативних комп'ютерів, які вкрай важко фізично захистити. Таким чином криптографія дозволяє гарантувати конфіденційність інформації навіть у випадку її крадіжки. Для цього криптографія використовує різні розвинені методи шифрування.

Тому, шифрування є невід’ємною частиною передавання даних. Якщо шифр не є надійний, то перехоплені зловмисником дані можуть бути легко дешифровані. Важливою характеристикою шифру є криптостійкість яка визначає його стійкість до дешифрування. Зазвичай ця характеристика визначається інтервалом часу, необхідним для реалізації процесу дешифрування.

В роботі запропоновано метод поточного шифрування який реалізується на основі гамми послідовностей з підвищеною криптографічною стійкістю. Для цієї задачі запропоновано застосування програмних генераторів динамічного хаосу.

Процес формування гамми послідовності полягає в наступному:

1) Обирається номери датчиків.

2) Задаються початкові параметри датчиків.

3) Генерується число в інтервалі від нуля до одиниці, якщо x[i] дорівнює або більше 0.5 то приймається логічне значення імпульсу одиниця, якщо x[i] менше 0.5 то приймається значення логічного імпульсу нуль.

4) Формується послідовність x[i], y[i], x[i+1], y[i+1], … , x[n], y[n], де x[i] та y[i] двійкові послідовності першого і другого генераторів; n це номер послідовностей. За умови що довжина інформаційної послідовності m менше 2n забезпечується абсолютна надійність шифрування.

Висновок. Запропоновано метод підвищення криптостійкості гамми послідовності який заснований на невизначеності застосування двох або більше генераторів хаосу.

Перелік посилань:

1. Опис працювання шифру: https://en.wikipedia.org/wiki/XOR_cipher
2. Процес шифрування: https://en.wikipedia.org/wiki/Encryption
3. Додаткова інформація по алгоритмам шифрування: Панасенко С. П. Алгоритми шифрування. Спеціальний довідник.- Санкт-Петербург 2009 рік.

ПОРІВНЯЛЬНИЙ АНАЛІЗ VPN СЕРВІСІВ

Бєлоциця Ю.О., студент 4-го курсу, група К-41, спеціальність 5.05010101 «Обслуговування програмних систем та комплексів»

Науковий керівник – викладач **Орлова Л.Б.**

Коледж зв’язку та інформатизації ОНАЗ ім. О. С. Попова

Анотація. *У мережі Інтернет нас чекає багато небезпек, як наприклад: підміна DNS-серверу. Завдяки чому зловмисник може перехопити наші особисті дані, а далі продати їх. Завдяки VPN, ми можемо зашифровувати увесь трафік та передавати його через DNS сервер VPN провайдеру. Але не всі VPN-сервери безпечні, у цій роботі було проаналізовано основні VPN протоколи ринку.*

Зараз на ринку є чотири основних протоколу тунелювання(ПТ): OpenVPN, PPTP, L2TP, IPsec IKEv2. Було проаналізовано алгоритми шифрування, порти з'єднання, автентифікацію даних, тощо.

РОЗРОБЛЕННЯ НЕЙРОННОЇ МЕРЕЖІ В ІНТЕРФЕЙСІ PYTHON

Левченко Д. В., Пузанов К. О. студенти 3-го курсу, групи К-31, спеціальності «Комп'ютерні науки».

Науковий керівник – **Васіліу Є. В.**

Коледж зв'язку та інформатизації ОНАЗ ім. О. С. Попова, Україна

Анотація. *Розглянуто принцип роботи нейронних мереж. Зображено процес розробки нейронної мережі на базі мови програмування Python за допомогою бібліотек tflearn, Gym та TensorFlow.*

Сьогодні нейронні мережі використовують як альтернативу всім існуючим алгоритмам для машинного перекладу, розпізнавання мови та музики, обробки зображень, визначення об'єктів на фото та відео. Глибоке навчання (Deep learning) – метод машинного навчання, заснований, у першу чергу, на нейронних мережах, хоча можна застосовувати й інші методи. У сучасній реальності практично у всьому, що стосується Deep Learning, використовують нейронні мережі.

Успіх глибокого навчання безпосередньо залежить від потужності техніки. На момент появи нейронних мереж потужності комп'ютерів були низькими, через що і самі мережі були досить слабкими. Саме тому в той час неможливо було створити велику кількість шарів нейронних мереж, а саме від кількості шарів залежать можливості мережі. Але з появою GPU і TPU все змінилося.

В основі будь-якої програми нейронної мережі лежить система нейронів та синапсів, як у людському мозку. Загальна назва цієї системи – модель нейронної системи. Класифікують три основних види нейронних шарів – вхідний (input layer), прихований (hidden layer), вихідний (output layer). Нейрони оперують числами в діапазоні [-1;1], якщо обчислювальне число не належить даному діапазону застосовують процес нормалізації – приведення значення числа до вказаного діапазону.

Для розробки нейронної мережі, використовуючи інтерфейс Python, необхідно використовувати бібліотеку, яка здатна працювати з нейронними мережами, наприклад: Keras, Detectron, TensorFlow, PyTorch, tflearn. Кожна з наведених бібліотек дозволяє працювати з нейронними мережами на будь-якій платформі: macOS, Windows або Linux.

Використовуючи кожен з цих фреймворків треба дотримуватися загального алгоритму створення нейронної мережі:

1. Імпорт потрібних бібліотек для роботи;
2. Присвоєння параметрам оптимального значення;
3. Збір інформації для навчання;
4. Створення моделі нейронної мережі;

5. Навчання моделі;
6. Отримання результатів.

Детальне пояснення наведеного алгоритму створення нейронної мережі:

1. Імпорт бібліотек:

Імпортуємо бібліотеки для коректної роботи нейронної мережі;

random – потрібен для отримання випадкових значень на етапі збору інформації;

numpy – потрібен для збереження даних дій кращих випадкових симуляцій у масиви;

tflearn – потрібен для створення нейронної мережі (її моделі, нейронних шарів і т.д.);

gym – потрібен для створення середовища CartPole, у якому ми тестуємо нейронну мережу;

2. Присвоєння параметрам оптимального значення:

```
LR = 1e-3
goal_steps = 500
score_requirement = 50
initial_games = 10000
```

Рисунок 1 – Параметри

На даному етапі ми задаємо параметри, значення яких буде впливати на результат;

LR(learning rate) – Швидкість навчання мережі;

goal_steps – Кількість кадрів в одній симуляції;

score_requirement – Якщо мережа на етапі збору інформації набирає задану кількість очок (кадрів), то вона запам’ятає усі дії поточної симуляції у масив.

Initial_games – кількість симуляцій, які будуть ініціалізовані на етапі збору інформації;

3. Збір інформації для навчання:

На даному етапі нейронна мережа збирає інформацію. У прикладі нейронна мережа проводить тестування с випадковими вхідними даними та вибирає найкращі з отриманих результатів.

4. Створення моделі нейронної мережі:

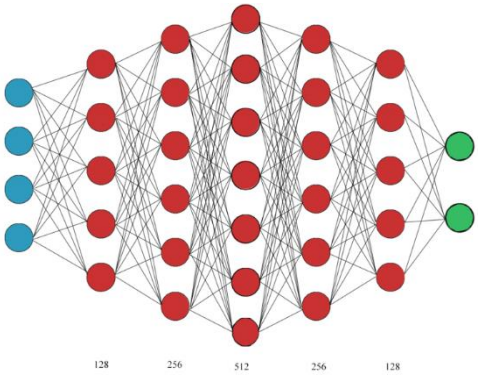


Рисунок 2 – модель нейронної мережі

На етапі створення моделі нейронної мережі ми ініціалізуємо нейронні шари (задаємо функцію оптимізації, функцію активації нейрона і спосіб обчислення функції помилки);

5. Навчання моделі:

При навчанні програма пропускає дані, отримані з етапу збору інформації, через модель задану кількість епох;

6. Отримання результатів:

На даному етапі ми розглядаємо отримані результати після навчання.

Висновок. Роль сучасних інформаційних і комунікаційних технологій в удосконаленні та модернізації комп'ютерного простору залишається важливою протягом багатьох років. У представленій роботі проведено аналіз основних понять і термінів нейронної мережі та розглянуто приклад створення нейронної мережі в інтерфейсі Python.

Перелік посилань:

1. http://www.immsp.kiev.ua/postgraduate/Biblioteka_trudy/ShtuchnNejron MeregNester2004.pdf“ Штучні нейронні мережі: Обчислення”.

2. Neural Networks and Deep Learning, Michael Nielsen / Dec 2019.

3. https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/A Step by Step Backpropagation Example.

4. http://stevenmiller888.github.io/mind-how-to-build-a-neural-network/Mind: How to Build a Neural Network (Part One).

5. https://neurohive.io/ru/osnovy-data-science/jepoha-razmer-batcha-iteracija/Эпоха, батч, итерация — в чем различия?

6. https://neuronus.com/theory/nn/951-etapy-zhiznennogo-tsikla-nejronnoj-seti.html. Этапы жизненного цикла нейронной сети

Технічний редактор – *Горлінська О. Ю.,* заступник директора з НР
Комп'ютерне макетування – *Орлова Л. Б., викладач*

Здано в набір 01.09.2020 р. Підписано до друку 07.03.2020 р.
Тираж 15 прим. обсяг 0,1 друк. арк.
Віддруковано в коледжі зв'язку та інформатизації ОНАЗ ім. О.С. Попова
© ВСП «ФКЗІ ОНАЗ ім. О. С. Попова», 2020.