**1 Слайд**

Я Овдієнко Андрій Володимирович студент групи ПА-22-2. пройшов навчально компʼютерно-технологічну практику на кафедрі компʼютерних технологій у Дніпровському національному університеті імені Олеся гончара із 12 по 22 Червня. Під керівництвом Сірик Світлани Федорівни. Буду зараз захищати свою практичну роботу

Проходження навчальної комп’ютерно - технологічної практики є невід’ємною складовою частиною під час навчального процесу – оскільки вона являється ефективним засобом для закріплення отриманих під час навчання знань, умінь і навичок, а також практика дозволяє студенту як майбутньому спеціалісту освоїти додаткові знання та отримати базовий досвід роботи з різними середовищами програмування.

**2 Слайд**

При проходженні комп’ютерно – технологічної практики була поставлена задача – вивчити можливості візуального програмування. Також слід зобразити блок-схему алгоритму виводу графіка на екран. Написати програму, яка буде це реалізовувати та підготувати звітність роботи. Індивідуальне завдання – це певна функція, яку слід графічно відобразити.

**3 Слайд**

До постановки задачі були технічні вимоги:

* реалізувати побудову графіків функцій;
* надати оформлення: вісі;
* вміти реалізувати побудову декількох графіків одночасно в одній системі координат. Для цього оберемо простіші графіки для відображення, тобто мені треба відобразити функції: індивідуального завдання; sin(x); cos(x); заданої рандомними числами та функції заданої в текстовому файлі дискретно;
* зробити паралельний переніс графіку відносно вісей абсцис / ординат;
* зробити стиснення / розтягнення графіку функції.

Я обрав мову С++ у середовищі Visual Studio Code. Для роботи із графікою я завантажив додаткову бібліотеку OpenGL, а точніше SFML/Graphics. Кожна людина, коли починає щось робити – вона придумує свої дії – як вона буде це реалізувати. Також справа можна побачити кроки, які надали мені алгоритм реалізації програми:

* побудуова плану роботи, взагалі уявлення що буду робити та приблизний вид графічного інтерфейса
* створення графічного інтерфейсу: в основі сам інтерфейс складається із звичайних прямокутників, які нарисовані один на одному за допомогою типу RactangleShape то відмальовки вінцкцією draw .
* Створення відповідних точок типу CircleShape, нехай буде радіусом 2 пікселя для того, щоб і було видно, і не була лінія занадто широкою, графік буду відображати у вигляді точок, а коло в маленьких розмірах, тобто з малим радіусом – це і є точка.
* Надання створенним точкам значень вирахуваних за певним законом, тобто створення масивів які будуть відповідати зя значення в координатах точки
* Створення можливості переміщення графіків та стискання або розтягування відносно осей, тобто зміна точок так, щоб графік переміщувався або стискався чи розтягувався
* Збереження данних у файлі та отримання даннихх із файла, мається на увазі зберігання діапазону графіка, кількості точок та самих точок, які також можна дістати із файлу.
* Створення інформації про програму, створення певного шрифту або підключення для виводу певної інформації

**4 Слайд**

У сучасному світі графіка є практично скрізь, тому що це візуальне відображення певного закону, який можна аналізувати і на основі цього робити певні висновки.

Наприклад, графік – це також індикатор здоров’я, на основі якого лікарі можуть приймати певні рішення. Може існувати певний фінансовий закон різних комерційних фірм, згідно з яким можна побачити, в який час та чи інша компанія отримує прибуток або збитки. Навіть розрахувати траєкторію ракети і дізнатися, де вона приземлиться або вибухне.

Тобто графіки є невід’ємною частиною життя будь-якої компанії, все порівнюється, відповідно можна будувати графіки функцій, які розкривають показники життя та існування людини. І в певній ситуації робити висновки.

**5 Слайд**

Для того щоб реалізувати програму було створено графічний інтерфейс

Інтерфейс моєї програми складається з трьох основних блоків: поле, де будуть відображатися графіки, знаходиться зліва, є позначення та нумерація осей, напрямок, підпис; меню для користувача, знаходиться праворуч, де він може задати певні значення інтервалу відображення функції - це відповідні значення A і B, під ними стоїть натуральне число N - кількість точок, з яких буде зроблено графік, нижче можна побачити кнопку, яка свторює графіки за критеріями вижче, нижче ви можете побачити, які функції будуть відображатися і яким кольором, відразу звернемо увагу на те, що з’явиться інше вікно для випадкової функції, де інтервал для псевдовипадкової функції буде встановлено інший, він буде складатися з N точок, але вже з'єднаних лініями, нижче відповідний коефіцієнт зсуву, який працює як для осі абсцис, так і для ординат, щоб зсунути графік, ви повинні вибрати функцію на панель вище, встановити коефіцієнт, він має бути і бути не нульовим, і натисніть кнопку зсуву по певній осі, відразу під кнопками зсуву є коефіцієнт стиснення по осях, який потрібно ввести від 0 до нескінченності, виключивши 0, тому що програма буде видавати повідомлення про дивні дії: додати 0 – це значить нічого не робити, помножити на нуль – прибрати графік, тоді навищо створювали; також є шапка – панель інструментів вгорі де є можливості: дивимося у верхній лівий кут, відкрити інформацію про розробника та програму, а праворуч – відповідні кнопки для збереження та відкриття файлу в якому знаходяться наші дані.

**6 Слайд**

Самі осі можна намалювати рівними лініями по середині висоти(горизонтальну пряму яку назвемо х) та по ширіні(вертикальну пряму яку назвемо у). Стрілки також можна намальювати рівними лініями задавши ним відповідні координати.

Вісь позначена таким чином, що точка перетину осі абсцис і ординат була вже обчислена і є деякою точкою (x0,y0) – центр координат. Тоді маючи певний крок одиничного відрізка і нашу опорну точку можна побудувати певну кількість одиничних відрізків та їх нумерацію, як у додатньому так і для відʼємного напрямку для осі абцис та ординат.

**7 Слайд**

Користувач задає інтервал на якому буде відображатися наша функція це початкова точка А та кінцева В. Тоді можна сказати, якщо у нас є інтервал поділений на n відрізків, із рівномірним кроком H, то в нас буде (n+1) вузлова точка вже із крайовими. Звідси випливає, що при кількості точок N – у нас буде (N-1) рівномірних відрізків, тоді можна знайти крок – різниця між кінцем та початком інтервала поділеного на кількість рівномірних відрізків. Окремим випадком є N=1. Тоді слід брати середнє арифметичне значення нашого проміжка – це і буде наша єдина вузлова точка, яка рахується за певним законом. І не забуваємо про радіус нашої точки, який також слід відняти як і від кожного ікса та і від всіх ігриків.

**8 Слайд**

Слід звернути особливу увагу на те, що наша точко (x0,y0) не має координати (0,0) – бо в компʼютері це верхній лівий край монітора. І рахуватися функція буде наприклад не від y=x; а y=-x; так само і для сінуса буде y=-sin(x), тобто в нас є певна функція f(x), і щоб знайти дійсне значення – слід взяти обернене число до результата обчислення f(x).

Для самого ж переміщення в нас є точки, які будують графік, крок одиничного відрізка, а також коефіцієнт зсуву. Для зсуву нашого графіка по осі х слід додати до кожної точки функції, а саме до ікса наш коефіцієнт множений на відповідний крок. А для зсуву по у слід зробити те саме, але замість операції додавання зробити віднімання, бо у у додатньоу напрямку йде вниз.

**9 Слайд**

У нас є точка (x0,y0) – перетин осей, крок або одиничний відрізок, точки графіка, відповідний коефіцієнт та розмір точки. Реалізація однакова, що для осі абцис, що длля ординат. Слід від кожної точки відняти значення x0 для осі абцис (або y0 якщо розтягуємо або звужуємо по осі ординат) потім тодати радіус нашої точки – 2 пікселя і тоді можна домножати на наш коефіцієнт, за цим робимо обернені дії, тобто додаємо х0 або y0 до кожної точки та віднімаємо радіус 2.

Коефіцієнт має бути додатнім. Якщо він буде від 0 виключено до 1 виключено, то функція буде звужуватися, якщо ж коефіцієнт 1 – нічого не зміниться, а у випадку, коли коефіцієнт більше одиниці – графік розтягується відносно певної осі.

**10 Слайд**

Для перевірки індивідуального завдання я ввів проміжок графіка від мінус 12 до 12 з кількістью точок тисяча, щоб вже графік виглядав як суцільна лінія, відповідно сгенерував та відтворив саме функцію індивідуального завдання. Тим самим я побудував індивідуальну функцію у своїй програмі. І дійсного доказу, що графік побудован правильно – я використав онлайн середовище WOLFRAM CLOUD – та ж програма Mathematica, якщо її завантажити, але безкоштовна, побудував там графік функції індивідуального завдання на проміжку [-12;12]. Тоді можна побачити, що графіки функцій однакові, і це значить, що моя функція і програма працють правильно.

**11 Слайд**

Дані зберігаються у форматі “.txt”, звісно і відкриваються через цей формат. Від користувача потрібно лише ввести назву файле без всякого розширення, а також можна ввести повний шлях до файлу з назву. Також коли вводимо назву файла, то знизу буде кнопка “Save” або “Open” – в залежності від того ми хочему зберегти файл чи відкрити його. І коли ми натиснимо на ці кнопки, то в нас є 4 варінта розвитку подій, 2 для кожної кнопки. Для “Save” – файл збережено, якщо він зберігся та його повна назва, яку ввів користувач і вже з розширенням “.txt”, якщо по тій чи іншій причині файл не створився або не відкрився, то вивідиться повідомлення, що файл не збережено. Те ж саме для “Open” – якщо файл відкрився та зчитав дані – вивідиться повідомлення про те що файл відкрився та його назва із закінченням “.txt”, якщо ж такого файла не було, то він створиться, але вивідиться повідомлення, що файл не був відкритий.

**12 Слайд**

Під час стажування з комп’ютерних технологій я навчився візуальному програмуванню на C++ за допомогою графічної бібліотеки SFML. Тепер я можу створити програму з різними кольорами та стилями. Я знаю, як створювати різні діаграми розсіювання з певними інтервалами та відтінками кольорів.

Робота складалася з побудови плану дій, створення графічного інтерфейсу, створення точок і надання їм координат для відображення, розрахованих на основі проміжку та кількості точок, які надав користувач, обробки паралельного переносу, стиснення, працювання з файлами збереження/відкриття даних, створення інформації про програму. Я зробив звітність: оформив щоденник практики, проміжні результати були – демонстрував код програми, створив презентацію, звіт, доповідь до презентації - опанував усі ці моменти.

Роботи була великою, тому додано окремі програмні модулі, які складаються з: заголовного файлу, сурс файлу. Додаток А відповідає за основний програмний код. У Додатку Б зберігаються вікна, які відкриваються для введення імені файлу для збереження/відкриття даних. Додаток В містить вікна помилок, коли, наприклад, не введено межу функції або кількість точок, або відсутні коефіцієнти. Додаток Г відповідає за можливість вводити лише числа в полях для інтервалів, кількість точок і коефіцієнтів, як правило, з можливістю видалення. Додаток Ґ містить базовий клас функцій, тобто батьківський клас. Додаток Д — це похідний клас від класу в Додатку Ґ, який відповідає за обробку та надання значень x і y для індивідуальної функції. Додатки Е, Є, Ж роблять те саме, що й додаток Д, але для функцій: cos(x), sin(x), випадкова функція. Додаток З відповідає за додаткові вікна: встановлення випадкових інтервалів розкладу, налаштування вікон інформації про програму та користувача, збереження даних у файл та їх отримання. Додаток И містить функцію, яка видаляє всі нулі з кінця рядка, якщо срока має крапку. Додаток І відповідає за кнопки для відображення графікі. Додаток Ї відповідає за кнопки переміщення та ступення стиснення. Додаток Й – скрипт запуску.