1. Які конкретні задачі планували вирішувати за допомогою цієї бібліотеки?

* SFML

Створення діалогово вікна. Малювання базових геометричних фігур, тексту. Можливість працювати з картинками. Обробка різних подій від користувача.

* NVIDIA CUDA

Можливість запускати код не тільки на головному процесорі, а також на додадкових

девайсах (відеокарта)

2. Чому було обрано саме цю бібліотеку, а не аналоги?

* SFML

Доволі популярна бібліотека. Open-source бібліотека. Низький поріг

входження. Хороша документація.

* NVIDIA CUDA

Низький поріг входження. Треба ознайомитись з принципом роботи

найпопулярніших відеокарт у світі.

3. Наскільки просто та зрозуміло було отримати, встановити, налаштувати та почати використовувати цю бібліотеку?

* SFML

Це залежить від компілятора. На MinGW компіляторі все запрацювало з першого разу. На MSVC були деякі труднощі, які підкорились лише через декілька днів. Але якщо в цілому, дуже легке підключення. Присутня можливість зібрати бібліотеку, завдяки CMakeLists.txt, що дуже полегшує роботу

* NVIDIA CUDA

Оскільки це не зовсім бібліотека, а розширення на с++, то треба було встановити компілятор для цієї штуки. Встановлення проходить легко, як будь-який додадок (далі, далі … встановити). Але прикро, що він йде на зв’язку з MSVC. Це обмежую програміста у виборі та можливому переході на інший інструментарій.

4. Наскільки зрозумілою та корисною була документація бібліотеки?

* SFML

Дуже зрозуміла документація. Та дуже классно, що документація прописана у самих хедерах, що доволі зручно. Не потрібно шукати щось в браузері, просто клікаєшь по тому що тобі потрібно і все. Та ще іноді трапляються приклади коду у хедєрах.

* NVIDIA CUDA

Документацію не читав, тому сказати не можу. Деяку інформацію підкреслив з stackoverflow та деяких відео на YouTube. Мені цього було достатньо, тому необхідності в документації не було.

5. Наскільки було зрозуміло, як саме використовувати бібліотеку, які класи/методи/функції використовувати для вирішення поставлених задач?

* SFML

Доволі все зрозуміло та інтуїтивно розумієш що до чого. Порівняно з Qt це щось :)

6. Наскільки зручно було використовувати бібліотеку, чи не треба було писати багато надлишкового коду?

* SFML

Використовувати зручно. Бібліотека дуже юзер френдлі, тому коду менше бути просто не може :)

7. Наскільки зрозумілою була поведінка класів/методів/функцій з бібліотеки?

* SFML

Поведінка передбачена. Якщо щось було не так, це вже по моїй вині. Про помилки бібліотека повідомляє, тому з цим проблем не має

8. Наскільки зрозумілою була взаємодія між різними класами/методами/функціями цієї бібліотеки, а також взаємодія між бібліотекою та власним кодом?

* SFML

Доволі зрозумілою. Взаємодія між власним кодом та бібліотечним була на високому рівні. Прочитав пару сторінок з книги про SFML, подивився як треба правильно писати та офіційній сторінці на GitHub, і як структурувати код у мене в голові відклалося.

* NVIDIA CUDA

Прикро, що не завжди можна використовувати елементи ООП. Та не має підтримки с++17 та с++20

9. Чи виникали якісь проблеми з використанням бібліотеки? Чи вдалось їх вирішити, як саме?

* SFML

Були проблеми тільки з підключенням до MSVC компілятора.

* NVIDIA CUDA

Проблеми виникали лише через незнання певних ключових слів. Але трошки пошукавши, все стало зрозуміло.

10. Що хорошого можна сказати про цю бібліотеку, які були позитивні аспекти використання бібліотеки?

* SFML

Дуже проста бібліотека для розуміння, та дуже компактна. Немає нічого зайвого, присутній тільки той функціонал, який використовується у більшості випадків. Дуже гарна названі класи та їх методи. Класи не перегружені функціями, тому не потрібно шукати щось одне з 10 можливих варіантів. Кожен клас відповідає за свою ціль і не більше.

* NVIDIA CUDA

По факту, малий функціонал (кажу про деякі ключові слова), але наскільки він потужний.

11. Що поганого можна сказати про цю бібліотеку, які були негативні аспекти використання бібліотеки?

* SFML

Важко сказати. Гарна документація, гарний open-source код, є precompiler binaries. Так як я дуже полюбляю SFML, то щось сказати проти нього мені важко :)

* NVIDIA CUDA

Не можу сказати, оскільки не було достатньо досвіду з цим інструментом

12. Якби довелось вирішувати аналогічну задачу, але вже враховуючи досвід використання в цій лабораторній роботі, що варто було б робити так само, а що змінити? Можливо, використати інші бібліотеки, чи використати інші можливості цієї бібліотеки, чи інакше організувати код, чи ще щось?

* SFML

Одразу прочитати документацію, навіть поверхосно. Подивитися на приклади коду, щоб одразу відклалася в голові структура проекту, та як працювати з цією бібліотекою. І я би замість того, щоб робити свій юзер інтерфейс, підключив би якійсь існуючий, наприклад ImGui. Тому що доволі багато часу витрачається на це, а краще цей час витратити на щось більш суттєве. Та і це ще + одна бібліотека, + ще один інструмент.

* NVIDIA CUDA

З приводу цього розширення, я не можу сказати, краще воно за всі інші, чи ні, оскільки не мав справу з іншими. Рекомендую від самого початку пошукати декілька інструментів для вирішення цієї проблеми, а вже потім робити вибір. Але нічого поганого окрім зв’язки з MSVC сказати не можу.

Додадкові посилання:

1. SFML <https://www.sfml-dev.org/>’
2. NVIDIA CUDA: <https://developer.nvidia.com/cuda-zone>
3. Симуляція диму на Хабр (CUDA)і: <https://habr.com/ru/post/470742/>
4. Симуляція диму на сайті NVIDIA: <http://developer.download.nvidia.com/books/HTML/gpugems/gpugems_ch38.html>
5. Intrinsic Functions (AVX2): <https://www.youtube.com/watch?v=x9Scb5Mku1g&t=1252s&ab_channel=javidx9>
6. Базове розуміння програмування на GPU: <https://www.youtube.com/watch?v=1cHx1baKqq0&ab_channel=SirajRaval>
7. Відео, яке замотивувало 1: <https://www.youtube.com/watch?v=LrEvoKI07Ww&ab_channel=Muzkaw>
8. Відео, яке замотивувало 2: <https://www.youtube.com/watch?v=TYzOSeK_qxM&t=238s&ab_channel=lutrarutra>
9. Відео, яке замотивувало 3: https://www.youtube.com/watch?v=XIvO\_tzBIMw&ab\_channel=Muzkaw