Під час виконання проекту було використано наступні бібліотеки та фреймворки:

1) The CImg Library( <http://cimg.eu/index.html> - офіційна сторінка ), з якої використано наступні можливості:

1) Клас CImg для роботи з зображеннями, і його можливості:

1. const T & Operator () для доступу до пікселів.
2. CImg< T > & assign() для конструювання зображення.
3. CImg< T > & resize() для зміни розміру зображення.
4. CImg< T > & normalize() нормалізує зображення.
5. CImg< T > & draw\_rectangle() малює прямокутник на зображенні.
6. CImg< T > & draw\_point() малює точку на зображенні розміром в 1 піксель.
7. CImg< T > & draw\_text() малює заданий текст на зображенні.
8. CImg< T > & draw\_circle() малює круг на зображенні.
9. CImg< T > & draw\_image() малює передане зображення на цьому.
10. CImg< T > & get\_channel() повертає канал зображення.
11. CImg< T > & dilate() розширює зображення.
12. CImg< T > & noise() накладає шум на зображення.
13. CImg< T > & blur() накладає блюр на зображення.
14. Int width() повертає ширину зображення у пікселях.
15. Int heigth() повертає висоту зображення у пікселях.
16. Unsigned long long CImg::tic() повертає поточне значення часу.
17. Unsigned long long CImg::toc() повертає значення часу з останнього виклику CImg::tic()
18. const CImg< T > & display() відображає зображення на заданому дісплеї.

2) Клас CImgDisplay для відображення зображень у вікні та обробки подій і його можливості:

1. bool is\_closed () const повертає true, якщо вікно закрито.
2. bool is\_key () const повертає true, якщо натиснута будь-яка клавіша .
3. bool is\_keyESC () const повертає true, якщо натиснута клавіша ‘ESC’.
4. bool is\_keyR () const повертає true, якщо натиснута клавіша ‘R’.
5. bool is\_keyW () const повертає true, якщо натиснута клавіша ‘W’.
6. bool is\_keyA () const повертає true, якщо натиснута клавіша ‘A’.
7. bool is\_keyS () const повертає true, якщо натиснута клавіша ‘S’.
8. bool is\_keyD () const повертає true, якщо натиснута клавіша ‘D’.
9. CImgDisplay & display () відображає на дисплеї передане зображення.
10. CImgDisplay & close () закриває вікно, за яке відповідає дісплей.
11. CImgDisplay & resize () зводить вікно дісплею до заданого розміру.
12. CImgDisplay & toggle\_fullscreen() розблоковує повноекранний режим.
13. CImgDisplay & flush () очищає всі поточні події.
14. CImgDisplay & wait () чекає задану кількість ms.

2) UT: C++20 μ(micro)/Unit Testing Framework ( <https://boost-ext.github.io/ut/> - Гітхаб )

Використано всі можливості фреймворку для тестування.

**Ретроспектива**

**The CImg Library**

1. За допомогою бібліотеки планувалось реалізувати весь графічний функціонал гри та обробку подій, які викликав користувач.
2. Було обрано цю бібліотеку, оскільки вона дозволяє не тільки працювати з зображеннями, а й виводити їх у вікно, здатне обробляти дії користувача. Також вона є досить простою у використанні.
3. Максимально просто.
4. Достатньо зрозуміло.
5. Оскільки документація бібліотеки представлена на окремому сайті, отримати необхідні знання було не дуже складно. Не зручним виявився пошук необхідних методів в документації, та іноді недостатня інформативність опису методів.
6. Максимально зручно. Надлишковий код писати не довелось.
7. Не був очевидним факт використання об’єкту класу CImg як функтора, щоб дізнатись значення певного кольору в RGB форматі для пікселя на заданих координатах.
8. Повністю зрозуміло.
9. Жодних проблем не виникало.
10. Бібліотека є простою у використанні, для початку роботи не потрібно над важких зусиль. На офіційному сайті є приклади використання деяких класів/методів. Немає надлишкового функціоналу, проте є практично весь потрібний.
11. Серед недоліків необхідно завантажити додаткові плагіни для відкриття готових зображень, які є застарілими, та давно не оновлювались.
12. Частина проекту, яка пов’язана з використанням цієї бібліотеки була б реалізована так само.

**UT: C++20 μ(micro)/Unit Testing Framework**

1. За допомогою бібліотеки планувалось реалізувати Unit tests.
2. Було обрано цю бібліотеку, оскільки вона має весь потрібний для цього проекту функціонал, і є досить простою у використанні.
3. Максимально просто.
4. Не вистачало детальності опису певних можливостей. В цілому зрозуміло.
5. Оскільки документація не завжди була досить детальною, доводилось окремо тестувати певні можливості бібліотеки, для більш точного розуміння процесу.
6. Оскільки бібліотека дозволяє писати функції у вигляді лямбда-виразів, доводилось для кожного тесту додатково писати свою власну функцію, аби збільшити читабельність коду.
7. 7/10
8. Повністю зрозуміло.
9. Жодних проблем не виникало.
10. Бібліотека є простою у використанні, для початку роботи не потрібно над важких зусиль. На Гітхаб-сторінці є прості приклади використання бібліотеки. Є весь необхідний функціонал.
11. Недоліком є недостатня детальність документації.
12. Частина проекту, яка пов’язана з використанням цієї бібліотеки була б реалізована так само.