

Содержание

Вступление	3
1 Аналитический раздел	4
2 Конструкторский раздел	5
2.1 Общая архитектура приложения	5
3 Технологический раздел	6
3.1 Выбор языка программирования	6
3.2 Выбор вспомогательных библиотек	6
3.3 Выбор базы данных	6
Заключение	7

Вступление

Повторение является неотъемлемой частью природы, проявляющейся в общих явлениях, таких как поверхностные узоры (например, стены, ткани, полы), геометрические структуры (например, галька, ветви), динамические движения (например, турбулентность жидкости, ходовые циклы, движение толпы) и деятельность человека (Например, рисование, жестикуляция, моделирование). Повторение было важным предметом изучения инженерных и научных дисциплин из-за повсеместности. Главная задача заключается в разработке общих и эффективных методов и простых в использовании интерфейсах для различных явлений и областей приложений. Этот алгоритм представляет собой интерактивную системы для анализа и синтеза повторений эскизов художника. Будет создана интерактивная система цифровой живописи для автодополнения утомительных повторений, таких как штриховки и пунктиры, сохраняющая при этом новаторские вариации и сохраняющая естественные потоки. В отличие от предыдущих работ, посвященных статическим и конечным штрихам, система анализирует рабочий процесс динамического и промежуточного рисования, который позволяет системе понять, как штрихи рисуются в прошлом, чтобы обеспечить высококачественные подсказки, ориентированные на контекст. Пользователи могут рисовать с системой как обычно, в то время как система автоматически предоставляет и обновляет предложения интерактивно без каких-либо дополнительных действий. Пользователи могут игнорировать или принимать эти предложения, аналогичные функциям автозаполнения в интегрированных средах разработки программирования, тем самым сохраняя полный контроль над процессом рисования.

1 Аналитический раздел

[illegible]

2 Конструкторский раздел

2.1 Общая архитектура приложения



Рисунок 2.1 — Общая архитектура приложения

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

3 Технологический раздел

3.1 Выбор языка программирования

Для реализации был выбран язык C++. Данный язык был обоснован следующими причинами: Причины:

- а) The quick brown fox jumps over the lazy dog.
- б) Компилируемый язык со статической типизацией.
- в) Сочетание высокоуровневых и низкоуровневых средств.
- г) Реализация ООП.
- д) Наличие удобной стандартной библиотеки шаблонов

3.2 Выбор вспомогательных библиотек

Для реализации программы была выбрана библиотека Qt.

- а) The quick brown fox jumps over the lazy dog.
- б) The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

3.3 Выбор базы данных

- а) Быстродействие. The quick brown fox jumps over the lazy dog.
- б) Лицензия. The quick brown fox jumps over the lazy dog.
- в) Переносимость. The quick brown fox jumps over the lazy dog.

Заклучение

The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.The quick brown fox jumps over the lazy dog.