**Введение:**

Игра «2048» является примером игры с простым геймплеем и интересным исполнением, которое смогло завоевать огромную популярность среди пользователей. Основная суть данной игры заключается в перемещении плиток с числами на игровом поле, игроку необходимо объединять плитки в одну с более высоким числом до тех пор, пока не будет получена плитка с числом 2048 или пока на поле не закончатся возможные ходы.

Цель данной курсовой работы заключается в создании игры «2048» на языке программирования C++ с использованием библиотеки SFML. В процессе работы будут рассмотрены следующие вопросы: алгоритмы и структуры данных, применяемые в игре, методы реализации игровых механик, работа с

В результате выполнения данного проекта мы получим полностью функциональную игру 2048, которая будет иметь удобный и привлекательный пользовательский интерфейс, а также работать без каких-либо ошибок. Кроме того, мы получим опыт работы с языком программирования С++ и библиотекой SFML, которые могут быть использованы в будущих проектах.

**1 Постановка задачи**

* 1. **Предметная область игры**

Компьютерные игры – это сложные программы, которые служат человеку для обучения или развлечения и рассчитаны на разные возрастные группы. Существует большое количество разнообразных жанров игр, наиболее популярными являются жанры: ролевые игры, симуляторы, логические игры, аркады, стратегии.

Игра «2048» относится к жанру логических игр и направлена на развитие логического мышления. Цель игры - перемещать и объединять плитки с числами на игровом поле, чтобы получить плитку с числом 2048.

Игровое поле представлено в виде сетки плиток, которые могут быть объединены и перемещены. Каждая плитка имеет свое число, которое удваивается при объединении с другой плиткой.

Одним из интересных аспектов игры «2048» является возможность получать более высокие числа, чем «2048», что делает игру еще более захватывающей и вызывающей.

Игра начинается с двух плиток, случайно расположенных на игровом поле, плитки содержат в себе числа 2 или 4. Игроку представляется возможность перемещать плитки в любом направлении - вверх, вниз, вправо или влево - при помощи стрелочек на клавиатуре. При каждом движении на свободной плитке появляется новая плитка с числом 2 или 4, случайно выбираемым игрой.

Если при перемещении плиток оказывается ситуация, что в одну сторону сдвигаются плитки с одинаковым значением, они объединяются в одну плитку, у которой число будет равно сумме чисел двух объединенных плиток. Например, если две плитки с числами 2 объединить, то получится новая плитка с числом 4. Если плитки с числами 4 объединить, то получится новая плитка с числом 8 и так далее.

чем меньше игровое поле, тем больше шанс совершить несколько ошибок так, что на игровом поле не будет возможности слияния плиток и больше не будет свободного места для появления новых плиток.

Победа в данной игре определяется ситуацией, когда на одной из плиток значение становится равным установленному, в большинстве случаев данное значение равно 2048, однако оно может изменятся в зависимости от режима игры.

Для создания игры «2048» будет использоваться язык программирования С++, а также библиотека SFML для создания графического отображения. Также необходимо описать структуры данных и алгоритмы, которые будут использоваться для реализации логики игры и графического отображения игрового поля и плиток. Для реализации игрового поля может быть использован двумерный вектор, элементами которого будут являться целые числа, хранящие числовые значения, которые отображаются на плитках.

**1.2 Обзор существующих аналогов**

"2048 for Windows" - классическая игра, реализованная в виде программы для ПК. Имеет простой и удобный интерфейс, представляет выбор размера игрового поля, победного значения. Интерфейс данной игры представлен на рисунке 1.

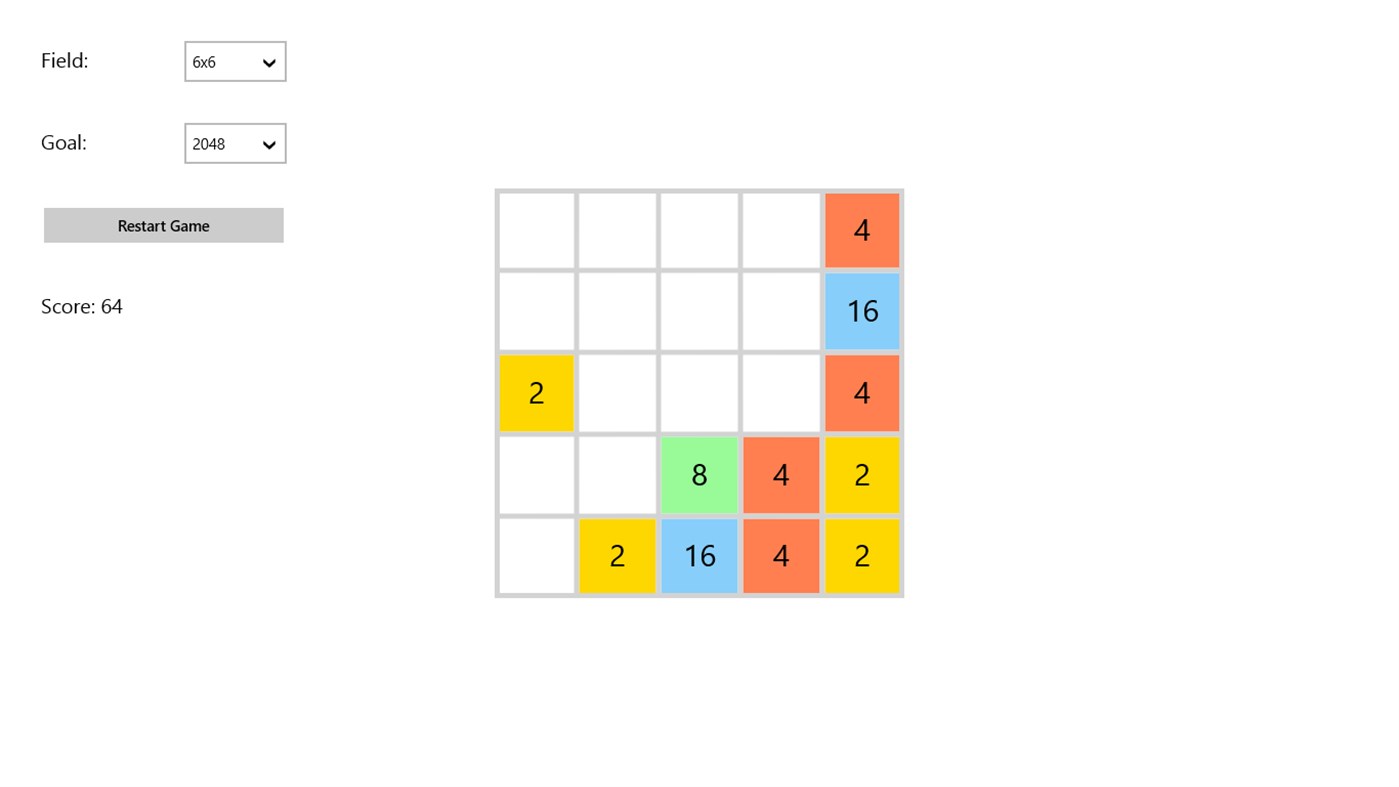


Рисунок 1

Данная игра имеет следующие положительные стороны: выбор размера поля, выбор числа, до получения которого продолжается игровой процесс. Также можно заметить, что все плитки на игровом поле, имеют разную расцветку, в зависимости от их номинала, имеется отображение набранных игроком очков.

Недостатки данной программы заключаются в невозможности сохранения наилучшего результата.

"Игра 512 1024 2048" - игра, которая поддерживается большим количеством операционных систем. На рисунке 2 представлен интерфейс данной программы.

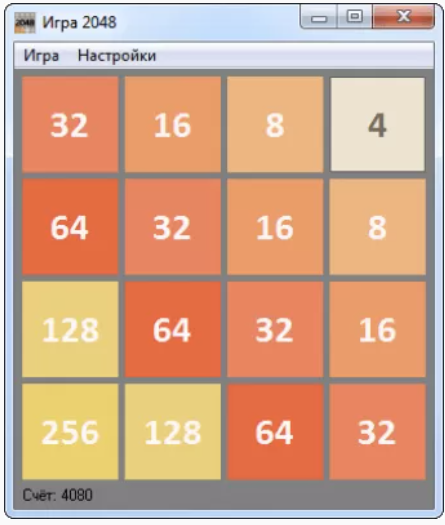


Рисунок 2

Положительные стороны данной игры: данная игра имеет настройки игрового процесса, возможность выбора победного номинала, отображение текущего результата во время игрового процесса.

Недостатками данной игры также является невозможность сохранить наилучший результат, также в данной игре нет возможности изменения размера игрового поля.

"2048 Game Professional for Windows". Интерфейс данной игры представлен на рисунке 3.

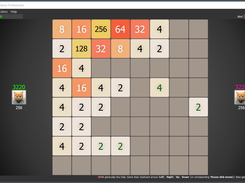


Рисунок 3

Данная игра имеет следующие положительные черты: предоставление выбора различных режимов игры, вариативность выбора размера поля и победного значения, также в ней реализована возможность изменение темы интерфейса игры, реализовано сохранение наилучших результатов.

В качестве недостатков данной игры можно отметить большое количество различной рекламы, а также нет возможности изменения языка интерфейса на русский язык.

Данный жанр игры ограничивает изменение игрового сюжета, однако в играх могут быть дополнительные режимы, возможность соревноваться с другими людьми по сети.

Большое количество аналогов подтверждает тот факт, что игра «2048» пользуется большой популярностью и спросом среди пользователей.

**1.3 Информационная база задачи**

Информационная база (ИБ) ─ это определенным способом организованная совокупность данных, хранимых в памяти вычислительной системы в виде файлов, с помощью которых удовлетворяются информационные потребности управленческих процессов и решаемых задач.

Входные параметры рабочего потока ─ это элементы данных, используемые для выполнения функции. Выходные параметры рабочего потока ─ это элементы данных, которые получаются в результате выполнения действия.

Для создания логической игры используются следующие входные данные: игровое окно, а также текстуры для игрового поля и интерфейса игры.

Выходными данными же является конечный результат, полученный пользователем по завершению выбранного режима.

Основными компонентами игры являются игровое поле, плитки с числовым значением и игровой процесс. Игровой процесс будет реализовываться за счет алгоритмов проверки соседних элементов, а также на перемещении элементов двумерного вектора.

Написание данного курсового проекта является хорошей практикой применения существующих навыков программирования и их развития. Помогает глубоко разобраться в функционале графических библиотек языка C++, а также приобрести необходимый опыт разработки компьюе

Реализация игры «2048» на языке программирования C++ с использованием библиотеки SFML может быть хорошей задачей для студентов, которые хотят изучить язык программирования C++ и научиться использовать графические библиотеки для создания игр. Такой проект может помочь студентам и начинающим разработчикам улучшить свои навыки программирования, практически применить знания и приобрести опыт разработки компьютерных игр.

Выходные данные будут включать полученный результат, количество очков, которое набрал игрок за время игры. Очки начисляются игроку во время игрового процесса за слияние плиток с одинаковым числовым значением.

**1.4 Функциональное назначение**

Функциональным назначением игры «2048» на языке программирования C++ с использованием библиотеки SFML является:

* Игра «2048» развивает различные стороны человека, такие как логическое мышление, аналитические способности, так как игрок анализирует текущую ситуацию на игровом поле, также развивается концентрация и внимания игрока.
* Игра «2048» также развивает стратегическое мышление, так как игроку необходимо продумывать свои действия на несколько шагов вперед, чтобы добиться наилучшего результата.

Игра 2048 подходит для широкого диапазона возрастов и пользователей. Она может быть интересной и полезной для детей и взрослых. Дети могут использовать эту игру для развития своих умственных способностей. Взрослые также могут наслаждаться игрой, чтобы тренировать свой ум и провести свободное время с пользой.

Для реализации игрового процесса будут реализованы след. функции

**2 Проектирование задачи**

**2.1 Алгоритм решения задачи**

Алгоритм – четкая последовательность действий, выполнение которой дает какой-то заранее известный результат.

**2.2 Логическое моделирование**

Диаграмма деятельности (Activity Diagram) - это графическое представление последовательности действий и поведения системы или процесса. Она используется для моделирования и визуализации потока управления или деятельности, отображая различные состояния, действия и переходы между ними.

Цели диаграммы деятельности:

1. Понять логику и последовательность выполнения процесса или операции.
2. Идентифицировать возможные проблемы, узкие места или улучшения в процессе.
3. Согласовать представление о деятельности или процессе между заинтересованными сторонами.

Основные элементы диаграммы деятельности:

1. Действия (Actions): Представляют собой конкретные шаги или операции, которые выполняются в процессе.
2. Решения (Decisions): Позволяют ветвиться потоку управления, основываясь на определенных условиях.
3. Параллельность (Parallelism): Показывает параллельное выполнение нескольких действий или процессов.
4. Соединители (Connectors): Указывают направление и последовательность потока управления.
5. Начало и конец (Start and End): Обозначают начало и завершение диаграммы.

Диаграмма вариантов использования (Use Case Diagram) - это графическое представление функциональных требований к системе, иллюстрирующее, как различные актеры (пользователи или внешние системы) взаимодействуют с системой для достижения определенных целей.

Цели диаграммы вариантов использования:

1. Понять функциональные требования к системе.
2. Определить взаимодействие между актерами и системой.
3. Согласовать представление о функциональности системы между заинтересованными сторонами.

Основные элементы диаграммы вариантов использования:

1. Актеры (Actors): представляют роли, которые взаимодействуют с системой. Актеры могут быть пользователями, другими системами или внешними компонентами.
2. Варианты использования (Use Cases): Описывают конкретные функциональные возможности или сценарии использования системы. Каждый вариант использования описывает, как система и акторы взаимодействуют для достижения определенной цели.
3. Отношения (Relationships): Указывают связи и взаимодействия между акторами и вариантами использования. Наиболее распространенными отношениями являются отношение включения (include) и обобщения (extend).

Упомянуть приложения

**2.3 Выбор и обоснование инструментов разработки**

В качестве языка программирования для создания игры используется язык программирования C++, основной характеристикой которого является возможность ручного управления использования памяти, что позволяет сделать код более эффективным. Для создания графического отображения игры на экране пользователя используется библиотека SFML (Simple and Fast Multimedia Library), которая обеспечивает высокую производительность и простоту в использовании.

Язык программирования C++ является мощным и широко используемым языком, который обладает рядом особенностей, положительных и отрицательных сторон.

Основные особенности языка программирования C++:

1. Мультипарадигменность: C++ поддерживает несколько парадигм программирования, включая процедурное, объектно-ориентированное и обобщенное программирование. Это позволяет разработчикам выбирать подход, наиболее подходящий для конкретной задачи.
2. Эффективность: C++ предоставляет близкое к машинному коду управление памятью и низкоуровневые операции, позволяющие разработчикам создавать быстрые и эффективные программы. Он предоставляет возможность непосредственного доступа к памяти и оптимизации, что особенно важно для систем с ограниченными ресурсами или высоких требованиях к производительности.
3. Объектно-ориентированное программирование: C++ поддерживает концепции объектно-ориентированного программирования, такие как инкапсуляция, наследование и полиморфизм. Это позволяет создавать модульные и повторно используемые кодовые блоки, улучшает структурирование программ и облегчает сопровождение.
4. Богатая стандартная библиотека: C++ предлагает обширную стандартную библиотеку, включающую контейнеры данных, алгоритмы, потоки ввода-вывода, работу с файлами и другие функциональные возможности. Это позволяет разработчикам использовать готовые инструменты и повышает производительность разработки.
5. Переносимость: C++ является кросс-платформенным языком программирования, что означает, что программы, написанные на C++, могут быть скомпилированы и запущены на различных операционных системах и аппаратных платформах.

Большое количество положительных сторон данного языка программирования также влечет за собой некоторые недостатки, такие как

1. Сложность: C++ является достаточно сложным языком программирования, требующим от разработчика глубокого понимания его особенностей и возможностей. Это может создавать более крутой кривую обучения и затруднять начинающим программистам освоение языка.
2. Уязвимость к ошибкам: из-за своей гибкости и возможности непосредственного доступа к памяти, C++ может быть более подвержен ошибкам и уязвимостям в коде, таким как неконтролируемое переполнение буфера или утечки памяти. Неправильное использование указателей и ссылок может привести к ошибкам времени выполнения и нестабильной работе программы.
3. Большой объем кода: использование C++ может потребовать большего объема кода по сравнению с некоторыми другими языками программирования. Некоторые операции, такие как управление памятью, требуют дополнительных строк кода, что может привести к более длинному и сложному коду.
4. Отсутствие автоматической сборки мусора: C++ не предоставляет встроенного механизма автоматической сборки мусора, что означает, что разработчику необходимо самостоятельно управлять выделением и освобождением памяти. Это требует от программиста более внимательного контроля и может приводить к ошибкам, связанным с утечками памяти или некорректным освобождением ресурсов.
5. Большой набор возможностей и функций: в C++ существует множество функций и возможностей, что может привести к перегрузке и сложности выбора наиболее подходящего подхода или функционала. Это может затруднить разработку и понимание кода, особенно для менее опытных разработчиков.

В качестве интегрируемой среды разработки используется Microsoft Visual Studio 2022. Microsoft Visual Studio (VS) - это интегрированная среда разработки (Integrated Development Environment, IDE), предоставляемая компанией Microsoft. Она предназначена для разработки различных типов приложений, включая приложения для Windows, веб-приложения, мобильные приложения, игры и другие.

Плюсы Visual Studio:

1. Обширный функционал: Visual Studio обладает множеством функций и инструментов, которые позволяют разработчикам создавать и отлаживать приложения различных типов и для разных платформ. Он предлагает интегрированную отладку, автоматическое завершение кода, инструменты для анализа кода, поддержку версий, дизайнеры пользовательского интерфейса и многое другое.
2. Широкая поддержка языков программирования: Visual Studio поддерживает множество языков программирования, включая C++, C#, Visual Basic, F#, JavaScript, Python и многие другие. Это дает разработчикам возможность выбирать наиболее подходящий язык для своего проекта.
3. Интеграция с экосистемой Microsoft: Visual Studio хорошо интегрирован с другими продуктами и сервисами Microsoft, такими как Azure, SQL Server, Team Foundation Server и Microsoft 365. Это облегчает разработку и развертывание приложений на платформе Microsoft.
4. Расширяемость: VS предлагает возможность расширения через дополнения, которые позволяют разработчикам добавлять дополнительные функциональные возможности и инструменты в среду разработки. Существует обширный каталог расширений, которые могут помочь улучшить процесс разработки и повысить производительность.

Минусы Visual Studio:

1. Ресурсоемкость: Visual Studio является мощным и комплексным инструментом, который требует значительных ресурсов компьютера для его работы. Это может привести к замедлению работы на медленных компьютерах или тех, где доступны ограниченные ресурсы.
2. Сложность и крутая кривая обучения: из-за своего богатого функционала, Visual Studio может быть сложным для начинающих программистов. Он требует времени и усилий, чтобы полностью овладеть всеми его возможностями и использовать их эффективно.
3. Платность: в зависимости от используемой версии, Visual Studio может быть платным продуктом. Более расширенные и профессиональные версии IDE могут иметь высокую стоимость, особенно для некоммерческих или небольших проектов. Однако Microsoft также предлагает бесплатные версии, такие как Visual Studio Community, которые предоставляют широкий функционал для индивидуальных разработчиков и малых команд.
4. Ограниченность платформой: Visual Studio является продуктом, ориентированным на разработку приложений для платформы Microsoft. В результате, если вам требуется создание приложений для других платформ, таких как macOS или Linux, вам может потребоваться использовать другие инструменты разработки.
5. Размер установочных файлов: полноценная установка Visual Studio может занимать значительное пространство на жестком диске, особенно учитывая многочисленные компоненты и пакеты, включенные в IDE. Это может быть проблематично для устройств с ограниченным местом на диске.