**Введение**

1. Цель проекта: автоматизация моделирования игровой симуляции. Наглядное представление каждого поколения симуляции для пользователя.
2. Обоснование необходимости разработки: уменьшение время затратности по сравнению с ручным моделированием симуляции различных конфигураций.

Симуляция будет включать следующие аспекты:  
Создание игрового поля: Двумерная сетка клеток, каждая из которых может быть живой или мёртвой.  
Реализация правил эволюции: Клетка остаётся живой, умирает или оживает в зависимости от количества соседей.  
Интерактивный интерфейс: Пользователь сможет задавать начальные условия (размеры поля, местоположение живых клеток), выбирать язык, останавливать симуляцию.

Для удобства восприятия состояния клеточного автомата будет реализована графическая визуализация симуляции.

Краткий обзор аналогов  
1. Классическая "Игра жизни" Джона Конвея  
Это первая и самая известная реализация клеточного автомата. Её популярность привела к созданию множества вариаций и симуляторов, как на базовых языках программирования, так и с использованием графических библиотек. Игра может реализовываться вручную на листе бумаги, что очень неэффективно, поскольку занимает много времени.  
2. Golly  
Программное обеспечение для работы с клеточными автоматами, поддерживающее различные правила и быстрые алгоритмы симуляции(URL: [Домашняя страница игры Golly Game of Life](https://golly.sourceforge.io/)).   
3. Онлайн-симуляторы  
Существуют многочисленные веб-реализации, позволяющие запускать "Игру жизни" прямо в браузере. Например, на сайтах [playgameoflife.com](https://playgameoflife.com/), [Game of Life](https://mygamelife.ru/) или [conwaylife.com](https://conwaylife.com/).

Источники литературы:

1. Игра «Жизнь» \\ Рувики URL: [Игра «Жизнь» — Рувики: Интернет-энциклопедия](https://ru.ruwiki.ru/wiki/%D0%98%D0%B3%D1%80%D0%B0_%C2%AB%D0%96%D0%B8%D0%B7%D0%BD%D1%8C%C2%BB) (дата обращения 10.01.2025) – Описание правил симуляции, история создания игры и примеры паттернов.
2. Краткий иллюстрированный глоссарий терминов в игре Конвея «Жизнь» \\ Сборник «Жизнь Конвея» URL: [Глоссарий жизни Конвея](http://www.radicaleye.com/lifepage/picgloss/picgloss.html) - Энциклопедия повторяющихся рисунков и терминов.

**Функциональные требования**

1. Функции:

Выбор модификации: однотонная и шахматная   
Принятие на обработку двух целочисленных значений  
Создание окна соответствующего размера исходя из данных, введённых пользователем. Первое число - количество ячеек в ширину, второе - количество ячеек в длину  
Задание состояния каждой ячейки путём взаимодействия с интерактивными кнопками в окне. Чётное количество нажатий - клетка мертва (не закрашена), нечётное - клетка жива (закрашена серым)  
Определение последующего состояния клетки по правилам модификации  
Изменения поля путём изменения состояния клеток по правилам модификации  
Остановка симуляции нажатием определённой клавиши  
Выход из окна нажатием определённой клавиши

Показ правил игры нажатием определённой клавиши

Подсчёт поколения симуляции  
Изменения окна при "окончании симуляции"

Выбор языка внешней спецификации (русский язык или английский язык)

1. Входные данные:

Вариант модификации: данные формата «маскированный ввод». Ввод является обязательным. Источник – пользователь.  
Ширина поля в ячейках: обязательные данные типа int. Значения будут приниматься, если лежат в промежутке [2; 25]. Источник – пользователь. (ошибочные значения приниматься не будут, окно ввода очистится и можно написать новое значение)  
Длина поля в ячейках: обязательные данные типа int. Значения будут приниматься, если лежат в промежутке [2; 25]. Источник – пользователь. (ошибочные значения приниматься не будут, окно ввода очистится и можно написать новое значение)  
Задание состояния ячеек: данные формата «маскированный ввод». Ввод является неограниченным и необязательным. Источник – пользователь.  
В процессе самой симуляции программа может получить от пользователя на вход либо пробел (остановка симуляции, повторное нажатие её возобновление), либо esc (полная остановка программы, сворачивание окна)

1. Выходные данные:   
   Либо бесконечно меняющийся дашборд, либо текстовое сообщение в окне "Симуляция окончена"

В процессе симуляции текстовое сообщение, содержащее номер поколения

1. Пользовательский интерфейс:

В начальном окне пользователю будет представлено поле с выбор модификации (однотонная и шахматная)

В следующем окне будет представлено 2 поля ввода (ввод ширины и длины)

Над каждым окном будет текстовое сообщение: над первым: «Введите количество ячеек в ширину», над вторым: «Введите количество ячеек в длину»

В случае некорректного ввода под полями ввода появится текстовое сообщение: «Некорректный ввод»

В случае корректного ввода появится новое окно, в котором будет поле ячеек, а также с левой стороны от поля будет инструкция по использованию клавиш быстрого доступа:

«’Пробел’ – остановка симуляции

‘Esc’ – выход из программы

‘Q’ – правила игры»

В случае окончания симуляции перед пользователем появится окно с текстовым сообщением: «Симуляция окончена»

1. Сценарии использования: Пользователь выбирает модификацию, задает размеры поля и стартовую конфигурацию. Запускает симуляцию и наблюдает за эволюцией.
2. Ограниченность функций: пользователю предоставлен ограниченный ввод для избежания некорректной работы программы (в полях для ввода длины и ширины поля принимаются только целочисленные значения в промежутке [2; 25])
3. Зависимости функций: Запуск симуляции невозможен без определения размеров поля.

**Нефункциональные требования**

1. Производительность: Время отклика на запрос пользователя не должно превышать 2 секунды.
2. Надёжность: Система должна обеспечивать доступность не менее 99.9% времени.
3. Безопасность: надёжность вводных данных.
4. Удобство использования: 99.9% пользователей должны успешно выполнить задачу в системе с первой попытки.
5. Сопровождаемость: Требования не предъявляются
6. Масштабируемость: Система должна поддерживать усложнение конфигураций без значительного снижения производительности.
7. Совместимость: ПО должно быть совместимо с персональными компьютерами и ноутбуками на различных операционных системах.
8. Доступность: Система должна быть доступна для использования в течение 24 часов в сутки, 7 дней в неделю.
9. Нормативные требования: Система должна соответствовать стандартам ISO 9241 (интерфейс)
10. Производительность в условиях нагрузки: Производительность должна оставаться стабильной, так как система предусмотрена только для одного пользователя

**Требования к интерфейсу**

1. Типы интерфейсов: Графический пользовательский интерфейс
2. Навигация: По приложению пользователь будет перемещаться с помощью клавиш на клавиатуре и кнопок интерфейса в окне.

’Пробел’ – остановка симуляции

‘Esc’ – выход из программы

‘Q’ – правила игры

В окне ввода количества ячеек при корректном вводе с помощью кнопки интерфейса «Далее» пользователь перемещается в окно поля с ячейками.

1. Внешний вид и оформление: Фоновый цвет всех окон – светло-серый (RGB: 220,220,220).

Все текстовые сообщения имеют шрифт Calibri. Кнопки интерфейса – серый (RGB: 165,165,165), сетка поля ячеек – черный (RGB: 0, 0, 0)

Однотонная модификация:

Все текстовые сообщения и рамки поля ввода, а также ячейки в «живом» состоянии – тёмно-серый (RGB: 71,74,81), ячейки в «мертвом» состоянии - светло-серый (RGB: 220,220,220)

Шахматная модификация:

Все текстовые сообщения и рамки поля ввода, а также ячейки в «живом состоянии» – тёмно-серый (RGB: 71,74,81) у первого типа, светло-серый (RGB: 220,220,220) у второго типа; ячейки в «мертвом» состоянии - серый (RGB: 165,165,165)

В начальном окне два поля ввода расположены в столбик и выровнены по центру.

В основном окне поле ячеек выровнено по центру. Инструкция по использованию клавиш быстрого доступа выровнена по левому краю сверху.

В окне окончания симуляции текстовое сообщение выровнено по центру.

1. Ввод данных: Значения пользователь сможет вводить с помощью полей ввода, а задание состояния ячеек и выбор языка программы осуществляется через интерактивные кнопки на интерфейсе. Некоторые функции доступны при нажатии определённых клавиш на клавиатуре.
2. Обратная связь пользователю: В случае некорректных данных при вводе пользователю будет показано сообщение: «Некорректный ввод. Попробуйте снова». В случае смерти всех ячеек симуляции пользователю будет показано окно с сообщением: «Симуляция окончена»
3. Адаптивность и отзывчивость: Адаптивная верстка для разных расширений экрана ноутбуков или персональных компьютеров. Окно будет подстраиваться под вьюпорт устройства и будет занимать всё видимое пространство на экране (кроме панели задач)\*
4. Управление ошибками: В случае некорректных данных при вводе пользователю будет показано сообщение: «Некорректный ввод. Попробуйте снова».
5. Визуализация данных: визуализация выходных данных осуществляется через дашборды (показ каждого поколения симуляции).
6. Совместимость с устройствами и браузерами: ПО совместимо с персональными компьютерами и ноутбуками.
7. Языковая поддержка: 2 доступных языка (русский язык, английский язык)
8. Безопасность интерфейса: надёжность ввода данных.

**Архитектура системы (UML-диаграммы)**

1. Описание компонентов системы:
2. Основная программа
3. Пользовательский интерфейс
4. Внешняя спецификация
5. Модуль ввода
6. Структура компонентов
7. Основная программа – управление правилами и состоянием клеточного поля, то есть выполнение симуляции;
8. Пользовательский интерфейс – взаимодействует с пользователем и вызывает соответствующие функции основной программы;
9. Внешняя спецификация – устанавливает правила, по которым пользователь будет осуществлять ввод данных;
10. Модуль ввода – обеспечивает передачу данных между пользователем и основной программой.
11. Взаимодействие компонентов

Через пользовательский интерфейс с помощью внешней спецификации пользователь вносит данные (размеры поля, положение живых клеток), которые обрабатываются модулем ввода и передаются в основную программу.

1. Выбор технологий

Использование языка программирования С++ для создания программного обеспечения и библиотеки SDL2 для пользовательского интерфейса.

1. Масштабируемость и производительность

Использование битовых полей

1. Безопасность и обработка ошибок

В случае некорректных данных при вводе пользователю будет показано сообщение: «Некорректный ввод. Попробуйте снова».

**Требования к базам данных**

1. Требования к базам данных не предъявляется по причине их отсутствия.

**Тестирование**

1. Типы тестирования:

Модульное тестирование: Проверка отдельных модулей (функций, классов) на корректность их работы.

Интеграционное тестирование: Проверка взаимодействия между разными модулями.

Системное тестирование: Проверка всей системы в её целостности.

Приемочное тестирование: Тестирование системы с точки зрения конечного пользователя.

Производительностное тестирование: Определение производительности и эффективности системы при различных нагрузках.

Безопасность: Проверка системы на наличие уязвимостей и эффективность мер безопасности.

1. Автоматизированное тестирование: Использование Appium для автоматизации тестов пользовательского интерфейса прикладного ПО.
2. Тестовая документация: Тестовые сценарии, проверочные таблицы, отчеты о выполнении тестов.
3. Тестирование с использованием данных: Тестирование формы ввода данных с использованием различных комбинаций значений. В том числе проверка корректной работы ограниченности ввода.
4. Тестирование производительности: Измерение времени ответа системы приразличных конфигурациях, в том числе при максимальном размере поля. Также сравнение времени, требующегося для различных входных данных.
5. Тестирование совместимости: Тестирование прикладного ПО на различных операционных системах.
6. Тестирование безопасности: Тестирование надёжности ввода данных пользователем.
7. Тестирование масштабируемости: Измерение производительности системы при увеличении нагрузки.
8. Тестирование обновлений и изменений: Тестирование совместимости после обновления библиотек или изменений в коде.

**Требования к документации**

1. Типы документации: Техническая документация, пользовательская документация, документация по тестированию, административная документация.
2. Структура документации: Введение, описание архитектуры, руководство пользователя, спецификация API для технической документации.
3. Формат и стиль: Использование Markdown для технической документации, четкий и понятный стиль написания.
4. Язык документации: Русский язык для пользовательской документации, английский язык для технической документации для международных проектов.
5. Требования к технической документации: Описание архитектуры системы, спецификации API, инструкции по установке и обслуживанию.
6. Требования к пользовательской документации: Четкие шаги по использованию продукта, часто задаваемые вопросы (FAQ), иллюстрации и скриншоты.
7. Обновление документации: Обязательное обновление документации после каждого значимого изменения в функциональности или архитектуре.
8. Согласование документации: Проведение обзоров и согласований документации с командой разработки и заказчиком.
9. Хранение и доступность: Документация будет храниться на открытом Яндекс Диске.