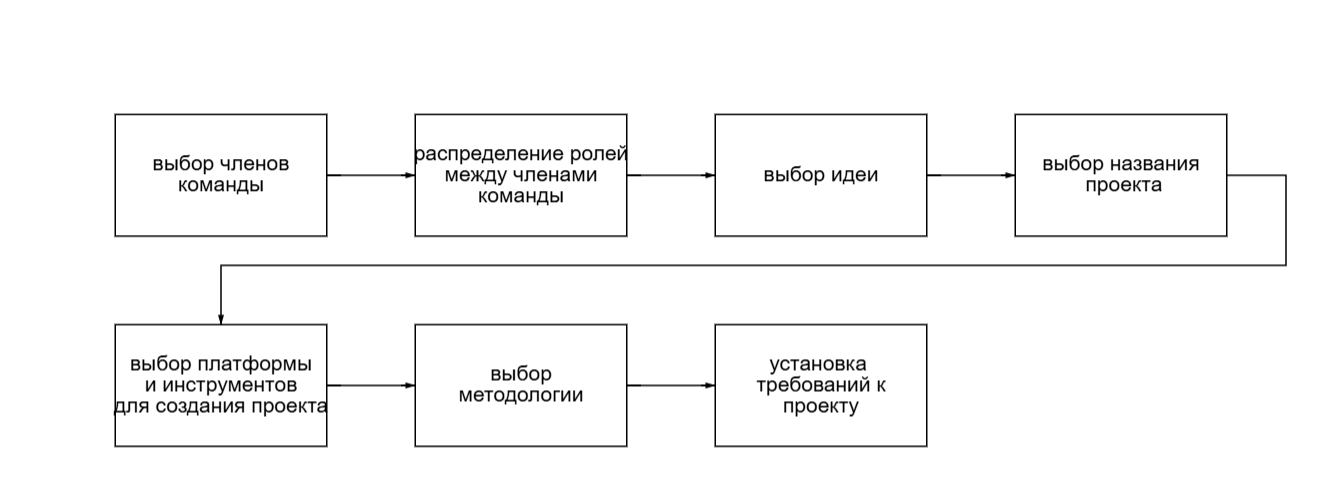
**Пререквизиты к созданию проекта**

****

**Как сделать правильный выбор?**

* **Компетенция и навыки**:

Каждый член команды должен обладать необходимыми техническими и профессиональными навыками для выполнения своих задач. Разнообразие навыков в команде позволяет эффективно решать различные задачи и проблемы

* **Командная работа и взаимодействие:**

Хорошие коммуникативные навыки и способность работать в команде способствует успешной реализации спринтов

* **Самоорганизация и ответственность:**

Члены команды должны быть способными к самореализации и брать на себя ответственность за выполнение задач

* **Мотивация и вовлеченность:**

Важно выбирать командиров, которые заинтересованы в успехе проекта и готовы работать на результат

Команда и роли

|  |  |  |  |
| --- | --- | --- | --- |
| № | Фамилия Имя | Факультет, группа | Роль |
| 1. | Герасимович Андрей | ФИТ, ПИ-7 | Тимлид, разработчик main |
| 2. | Бондаренко | ФИТ, ПИ-7 | Разработчик input |
| 3. | Граховская Каролина | ФИТ, ПИ-7 | тестировщик |
| 4. | Артеменко Вадим | ФИТ, ПИ-7 | Разработчик grid |

**Как выбрать верную методологию?**

Необходимо учитывать несколько факторов, по которым можно и нужно делать выбор методологии.

* **Сколько времени дается на проект**
* **Какие у команды навыки**
* **Требования заказчика**
* **«Фиксированность требований»**

**Какую методологию выбрали мы? Почему?**

Выбор нашей команды пал на методологию Agile, а в частности смесь Agile/Kanban, Agile/Scrum.

Главная причина – простота и гибкость этих методологий. Канбан позволяет визуализировать прогресс и поставить явные цели проекта перед всеми членами команды.

Скрам дает возможность конкретизировать и упростить процесс управления разработкой, разбивая его на части. Также приятным бонусом становится и минимизация требований к документации в связи с гибкостью методологии и постоянными изменениями.

**Проект «Игра в жизнь»**

В 1970-е получила известность двухмерная клеточно-автоматичная модель с двумя состояниями клеток, известная как игра «Жизнь». Изобретённая Джоном Конвеем и популяризованная Мартином Гарднером, она использует свои правила: на квадратной решетке каждая клетка имеет 8 соседей; если клетка имеет двух «живых» соседей, она остается в прежнем состоянии. В остальных случаях клетка «умирает». Несмотря на свои простоту, система проявляет огромное разнообразие поведения, колеблясь между хаосом и порядком. Одним из феноменов игры являются глайдеры- сочетание клеток, «движущиеся» по сетке как единое целое и взаимодействующие с другими статичными или подвижными конструкциями. Возможно установить стартовое состояние клеток, при котором глайдеры будут выполнять некоторые вычисления. Впоследствии было доказано, что игра «Жизнь» может полностью эмулировать универсальную машину Тьюринга.

**Назначение:** данный проект предназначен для создания «игры в жизнь».

**Цель:** основная цель игры заключается в моделировании процесса жизни и смерти клеток на двумерной сетке. Каждая клетка может находиться в одном из двух состояний: «живая» или «мертвая»

**Правила игры:**

1. **Рождение:** мертвая клетка, окруженная ровно тремя живыми соседями, становится живой
2. **Сохранение:** живая клетка, имеющая два или три живых соседа, продолжает играть. В противном случае(если у нее меньше двух или больше трех живых соседей) она умирает от одиночества или перенаселения соответственно.

**Модули:**

1. Функция printGrid является основной для визуализации игрового процесса и взаимодействия пользователя с программой. Предназначение printGrid:
2. Очистка экрана (перед каждым выводом она очищает консоль, чтобы сделать ее более наглядной
3. Границы поля (выводит верхнюю и нижнюю границы сетки, чтобы сделать ее более наглядной)
4. Состояние клеток (выводит текущее состояние каждой клетки на поле)
5. Курсор (отмечает положение курсора специальным символом, а также отображает его состояние (занято/свободно))
6. Подсказки для управления (выводит инструкции для пользователя по взаимодействию с игрой)
7. Функция countLiveNeighbors отвечает за подсчет количества живых (занятых) соседних клеток для заданной клетки на игровом поле. Предназначение countLiveNeighbors:
8. Счётчик: (устанавливает начальное значение счетчика живых соседей в 0)
9. Перебор соседей (проходит через все 8 соседних клеток заданной клетки)
10. Обработка границ (использует операцию взятия остатка для того, чтобы обеспечить тороидальную структуру сетки, то есть если соседняя клетка выходит за границу, она "появляется" на противоположной стороне)
11. Подсчёт живых соседей (увеличивает счетчик на 1, если соседняя клетка является живой)
12. Возврат значения (возвращает общее количество живых соседних клеток)
13. Функция updateGrid отвечает за обновление игрового поля в соответствии с правилами игры "Жизнь". Она анализирует текущее состояние поля и создает новое поколение, применяя эти правила к каждой ячейке. Предназначение updateGrid:
14. Создание копии (создает копию текущего игрового поля, чтобы хранить результаты следующего поколения)
15. Перебор клеток (проходит через каждую ячейку игрового поля)
16. Применение правил игры
17. Обновление игрового поля (заменяет содержимое исходного игрового поля новым поколением)
18. Функция initializeGrid настраивает начальное состояние игрового поля (двумерного вектора grid) перед началом основного игрового цикла. Её основная задача — позволить пользователю задать начальное состояние клеток (включить или выключить их) с помощью действий:
19. Перемещение курсора
20. Установка и удаление клеток (клавиша Enter изменяет состояние клетки под курсором. Если клетка была "живой" (true), она станет "мертвой" (false), и наоборот)
21. Завершение ввода (клавиша 'q' завершает процесс ввода и выходит из цикла, таким образом заканчивая работу функции initializeGrid)

Модуль "input" отвечает за бесконечный цикл игры и паузы, включает функцию inputHandler.

inputHandler обеспечивает основной цикл взаимодействия пользователя с игрой, включая управление и визуальную обратную связь. Предназначение inputHandler:

1. Инициализация курсора: задает начальное положение курсора в центре игрового поля.

2. Бесконечный цикл ввода: постоянно ожидает ввода пользователя.

3. Обработка действий пользователя: (включает функцию printGrid)

Модуль "main" включает в себя основную функцию main, включающую в себя все остальные, и таким образом реализует игру "Жизнь".

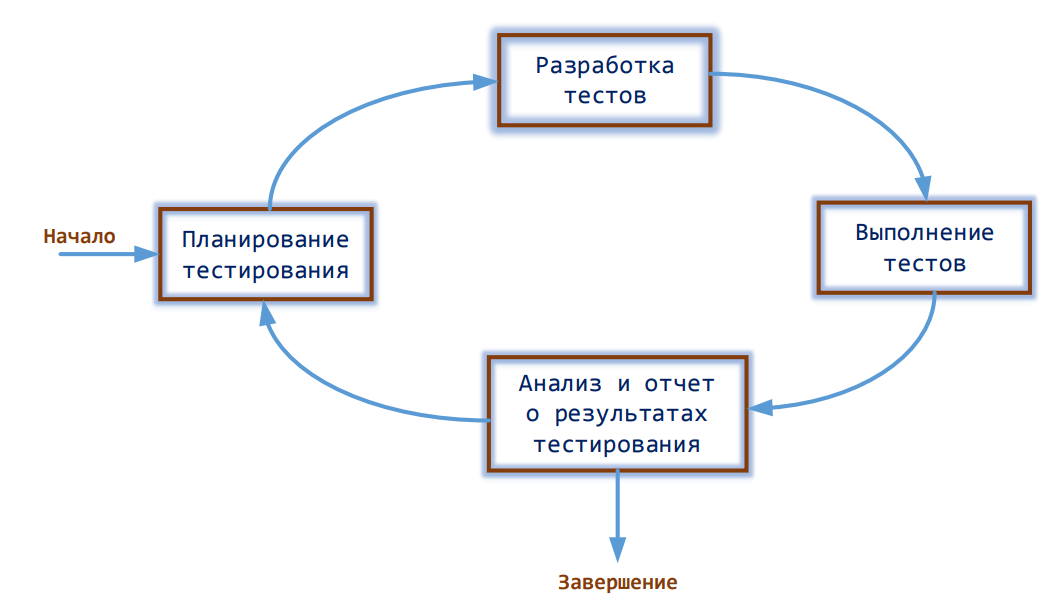
main создает изначальное игровое поле и инициализирует его (вызов initializeGrid), запускает поток для обработки ввода (с помощью inputHandler) и содержит основной игровой цикл, который обновляет и отображает состояние игры.

|  |  |  |
| --- | --- | --- |
| ID | пункт | Пожелание заказчика |
| 1 | Геймплей и механики | Хочется видеть интересные механики, которые будут вовлекать игроков. Они должны быть интуитивно понятными, но при этом предоставлять возможности для глубокого освоения |
| 2 | Уровень сложности | Нам нужно учесть разные уровни навыков игроков. Игра должна быть доступна для новичков, но достаточно сложной для опытных пользователей, чтобы они могли испытать увлекательные вызовы |
| 3 | Обратная связь | Я хотел бы, чтобы в процессе разработки мы собрали обратную связь от тестировщика и игроков |

Диаграмма вариантов использования

Нечем заняться, хочу поиграть

Тестирование функций



|  |  |  |
| --- | --- | --- |
| действие | Ожидаемый результат | Результат теста |
| Вызвать «ptintGrid» с заранрее определенной сеткой «grid» и курсором в позиции (0.0) | На экране отобразиться сетка с символами ‘O’, ‘X’, ‘+’, ‘#’ в заданных местах, а курсор будет на позиции (0,0) | Сетка корректно отображается, курсор на нужной позиции |
| Вызвать «countLiveNeighbors» для ячейки с известным количеством живых соседей | Возврат правильного количества живых соседей | Функция возвращает ожидаемое значение |
| Вызвать «updateGrid» после изменения состояния ячеек в сетке | Сетка будет применена игра «игра в жизнь», состояние клеток обновляется согласно правилам | Состояние клеток будет изменено корректно |
| Запустить «initialzeGrid» и взаимодействовать с программой, устанавливая и удаляя клетки | Пользаватель может установить любую клетку, и ее состояние должно меняться при нажатии клавиши Enter | Состояние клеток изменяется по командам пользователя, программа корректно отображает текущую сетку |
| Запустить «inputHandler» и вводит команды для паузы и перемещения курсора | Программа должна корректно обрабатывать команды для установки/удаления клеток и переключения между режимами паузы и продолжения |  |