Игровые механики:

1. Персонажи: Главный персонаж Игрок: Игрок может двигаться по уровню с помощью клавиш стрелок. Его выносливость уменьшается в зависимости от непрерывности движения или совершения определенных действий, например прорубание с помощью кирки проходов. Список планируется расширять.

Существа (боты): боты (дружественные, враждебные и нейтральные.) Приведения, животные (в том числе подвергшиеся мутации), магические (сказочные) существа. Список планируется расширять

1. Здоровье и выносливость: у игрока есть параметры здоровья, выносливости, жажды и и ясности ума, которые изменяются в зависимости от его действий.
2. Система умений и прокачки: навыки, которые можно улучшать (например, здоровье, выносливость, ясности ума, навык использования кирки и т.д.).
3. Магия или особые способности: магические способностей или навыки, таких как прохождение стен, телепортация или временная неуязвимость. Расходуется ясность ума.
4. Система экипировки: предметы, которые могут быть надеты на персонажа, чтобы улучшать его характеристики. Рюкзаки разных размеров, маскировочные накидки скрывающие игрока от разных существ. Список планируется расширять
5. Уровни и переходы: Игра содержит несколько уровней, которые генерируются с уникальными особенностями (стены, ловушки, предметы, объекты, существа (боты) и т.д.). Генерация случайна, после посещения игроком некоторое время она сохраняется, но при наступлении некоторых условий эти уровни генерируются заново. Существуют различные типы уровней, такие как Обычные, подземелья с лавой, ледяные пещеры, затопленные секции, туманные уровни, магические, и т.д.

Существуют уникальные уровни: Чердак (его состояние никогда не меняется) На нем можно размещать некоторые объекта – Шкаф для хранения вещей, Кровать для сохранения игры, верстак для создания и модификации предметов и инструментов, Компьютер Алдан для общения с игроком и помогающий развивать сюжет. Потайные комнаты (механика разрабатывается). Список планируется расширять.

Переход между уровнями: В игре есть лестницы для подъема и спуска между уровнями.

1. Предметы: В игре присутствуют различные вещи и инструменты, которые можно собирать, носить в рюкзаке и использовать. Вещи – ресурсы, запчасти, еда, вода, аптечки, и т.д. Инструмент – Кирка, нож, фонарь, отмычки, волшебная палочка. Список планируется расширять.
2. Инвентарь и использование предметов: Игрок может выбирать предметы в рюкзаке, использовать их или выбрасывать.
3. Объекты: На уровнях присутствуют различные объекты: активные, с которыми игрок может взаимодействовать – сундуки, мебель, надписи на стенах, потайные двери, ловушки, Пассивные – потерянные вещи, скелеты, развалившаяся мебель. Список планируется расширять
4. Ловушки: На уровнях есть шипы, ловушки с ядом или обвалы которые могут нанести урон игроку. Список планируется расширять.
5. Система крафта: Возможность создавать новые предметы из найденных материалов.
6. Информационная панель: Отображает характеристики игрока, инвентарь и сообщения игры.
7. Сохранение и загрузка игры: Возможность сохранить прогресс и загрузить его позже.
8. Видимость: Карта уровня постепенно открывается по мере перемещения игрока. При переходе с уровня на уровень видимость уже открытых областей сохраняется до момента перезагрузки уровня (игровая механика Выброс)
9. Стартовое меню: Игровое меню позволяет начать новую игру, загрузить сохранение, создать и управлять профилем игрока (поддерживается несколько профилей, каждый со своими сохранениями), изменить настройки или выйти.
10. Сообщения и взаимодействие с объектами: В игре предусмотрены текстовые сообщения для уведомления игрока о событиях, таких как нахождение записок, чтения надписей на стенах, решение загадок или взаимодействие с предметами.
11. Механика сюжета реализованная как комиксы – демонстрация на экране страниц комикса рассказывающих о развитии сюжета. Развитие сюжета возможна по нескольким линиям с несколькими финалами.

Хорошо, я проанализирую техническое задание для создания 2D рогалик-бродилки с видом сверху на базе C# и Unity. Вот ключевые аспекты и рекомендации по реализации:

1. Сюжет и сеттинг:

- Игра основана на оригинальной концепции "Подземелья НИИЧАВО" с элементами научной фантастики и магии.

- Главный герой ищет свою дочь в постоянно меняющемся подземном комплексе.

- Реализация: Использовать систему диалогов Unity для раскрытия сюжета. Создать ScriptableObjects для хранения информации о персонажах и сюжетных элементах.

2. Генерация уровней:

- Процедурная генерация разнообразных уровней (обычные, лавовые, ледяные, затопленные и т.д.).

- Реализация: Использовать алгоритмы процедурной генерации (например, клеточные автоматы или алгоритм BSP) для создания уровней. Применить тайловую систему Unity для отрисовки уровней.

3. Персонаж и управление:

- Движение персонажа, система здоровья, выносливости, жажды и ясности ума.

- Реализация: Создать скрипт PlayerController для управления движением. Использовать систему физики Unity 2D для обработки столкновений.

4. Инвентарь и предметы:

- Система сбора, хранения и использования предметов.

- Реализация: Создать класс Item и систему управления инвентарем. Использовать UI-систему Unity для отображения инвентаря.

5. Система прогрессии:

- Развитие навыков персонажа, система опыта.

- Реализация: Создать ScriptableObjects для навыков и системы прокачки. Использовать PlayerPrefs для сохранения прогресса между сессиями.

6. Боевая система и взаимодействие с NPC:

- Реализация сражений с враждебными существами и взаимодействия с дружественными NPC.

- Реализация: Создать систему AI для NPC, использовать систему NavMesh для навигации NPC.

7. Магическая система:

- Реализация различных магических способностей и их эффектов.

- Реализация: Создать систему частиц для визуальных эффектов магии. Использовать паттерн "Команда" для реализации различных магических эффектов.

8. Крафтинг:

- Система создания предметов из найденных материалов.

- Реализация: Создать ScriptableObjects для рецептов крафтинга. Реализовать UI для системы крафтинга.

9. Сохранение и загрузка:

- Система сохранения прогресса игрока.

- Реализация: Использовать сериализацию данных и PlayerPrefs для сохранения состояния игры.

10. Пользовательский интерфейс:

- Информационные панели, меню, система диалогов.

- Реализация: Использовать Canvas и UI-элементы Unity для создания интерфейса.

11. Звук и музыка:

- Реализация звуковых эффектов и фоновой музыки.

- Реализация: Использовать систему Audio в Unity, создать AudioManager для управления звуками.

12. Оптимизация:

- Обеспечение плавной работы игры на различных устройствах.

- Реализация: Использовать пулинг объектов для оптимизации производительности, применять LOD (Level of Detail) для оптимизации графики.

13. Тестирование:

- Реализация системы для отладки и балансировки игры.

- Реализация: Использовать режим разработчика с дополнительными инструментами отладки.

Для эффективной разработки рекомендуется использовать следующие паттерны проектирования и технологии Unity:

1. Singleton для глобальных менеджеров (GameManager, AudioManager).

2. Observer для системы событий.

3. State для управления состояниями персонажа и NPC.

4. Factory для создания объектов и врагов.

5. ScriptableObjects для хранения данных об игровых объектах, навыках и предметах.

6. Coroutines для управления временными процессами.

7. UnityEvents для создания гибкой системы взаимодействия между объектами.

Этот анализ предоставляет общий обзор реализации игры на основе технического задания. Для более детального планирования потребуется разработка подробного технического дизайн-документа и прототипирование ключевых механик.

### Дорожная карта разработки игры "Подземелья НИИЧАВО" на базе Unity и C#

#### Этап 1: Подготовительный этап (2-3 недели)

1. \*\*Анализ технического задания и планирование проекта\*\*

- Определение основных игровых механик и приоритетов их реализации.

- Разработка схемы классов и системы управления.

- Создание первоначального макета уровней и игровых объектов.

2. \*\*Настройка окружения\*\*

- Настройка проекта в Unity, создание базовой структуры сцены.

- Подготовка скриптов на C# для основных игровых объектов.

- Настройка системы контроля версий (например, Git) для хранения кода.

#### Этап 2: Базовая реализация (4-6 недель)

1. \*\*Создание базовых игровых механик\*\*

- Реализация перемещения игрока и системы управления (камера, управление персонажем).

- Добавление параметров здоровья, выносливости, жажды и ясности ума.

- Настройка базового инвентаря и системы использования предметов.

- Реализация интерфейса пользователя (информационная панель, инвентарь, здоровье и т.д.).

2. \*\*Разработка системы уровней и процедурной генерации\*\*

- Создание базовой системы переходов между уровнями (лестницы, входы/выходы).

- Настройка процедурной генерации уровней (случайная генерация стен, объектов, врагов).

3. \*\*Добавление простых объектов и врагов\*\*

- Реализация базовых объектов (стены, ловушки, предметы).

- Добавление первых врагов (например, мутировавшие существа или магические существа).

- Настройка базового искусственного интеллекта (патрулирование, атака).

#### Этап 3: Расширение функционала (6-8 недель)

1. \*\*Разработка системы прокачки и навыков\*\*

- Введение системы опыта и улучшения характеристик (здоровье, магические способности).

- Добавление магических способностей и уникальных навыков.

2. \*\*Усовершенствование инвентаря и системы предметов\*\*

- Реализация комбинирования предметов и системы крафта.

- Добавление редких и уникальных предметов, которые влияют на геймплей.

- Введение механики деградации предметов и их ремонта.

3. \*\*Работа над аномалиями и артефактами\*\*

- Реализация аномальных зон с искажением пространства и времени.

- Добавление артефактов с уникальными свойствами и возможностью их комбинации.

4. \*\*Система взаимодействия с персонажами и диалогов\*\*

- Настройка системы диалогов и отношений с NPC.

- Введение дружественных и нейтральных персонажей, с которыми можно взаимодействовать.

#### Этап 4: Финализация контента и тестирование (8-10 недель)

1. \*\*Проработка сюжета и создание квестов\*\*

- Добавление основных и побочных квестов.

- Введение временных парадоксов и альтернативных концовок.

2. \*\*Оптимизация и отладка\*\*

- Устранение багов и оптимизация кода.

- Работа над производительностью игры и снижением времени загрузки уровней.

3. \*\*Тестирование\*\*

- Проведение альфа-тестирования и сбор отзывов.

- Исправление найденных проблем и доработка механик.

#### Этап 5: Выпуск и пост-релизная поддержка (4-6 недель)

1. \*\*Подготовка к выпуску\*\*

- Создание маркетинговых материалов (трейлер, сайт, посты).

- Настройка платформ распространения (Steam, GOG и т.д.).

2. \*\*Поддержка после релиза\*\*

- Внесение правок по отзывам пользователей.

- Добавление нового контента и поддержка пользовательских модов.

- Планирование обновлений и сезонных событий.

### Примерные сроки разработки: 6-8 месяцев

Учитывая, что разработкой занимается один человек, важно правильно распределять время и приоритеты. На начальных этапах следует сосредоточиться на основных механиках, постепенно добавляя больше контента и улучшая игру на основе тестирования и отзывов.