**Федеральное агентство по образованию   
Федеральное бюджетное образовательное учреждение высшего профессионального образования   
Кемеровский Государственный Университет   
Институт Цифры   
   
  
  
  
Презентация   
на тему: Игра 2D “Leonid”**

**Выполнил:   
 Морозов Матвей Сергеевич  
 Рецензент:  
 Пасютин Александр Сергеевич   
  
  
  
  
  
  
  
  
  
  
  
Кемерово 2024г.**

**Введение в предметную область**

Компьютерные игры появились относительно недавно. И несмотря на это, они колоссально развиваются и приносят ежегодно миллиардные доходы. Быстро развилась компьютерная индустрия: всему поспособствовал большой выбор компьютерных технологий и появление сети Интернет. В результате – обычные развлечения стали не так популярны, как компьютерные. Чтобы поиграть, достаточно иметь компьютер, ноутбук или игровую приставку, а также лицензионную игру или ее копию, скачанную с интернета. Для пользователя не обязательно иметь знания, чтобы выбрать подходящею игру, нужно всего лишь понимать, какой жанр игры более интересен.

В последнее время игры уже перестали быть как просто средство развлечения для свободного время провождения и отвлечения, например есть игры, которые используют в разных учреждениях для обучения, такие игры называются развивающими, для практики симуляторами. Развивающие игры помогают обучать ребенка чему либо, а симуляторы обучают специалистов в разных областях: от пилотов самолета и до самой простой специальности.

Современные игровые движки играют ключевую роль в создании интерактивных приложений и видеоигр. Одной из популярных платформ для разработки является **Godot Engine**. Этот свободный и открытый инструмент позволяет создать игры различных жанров и уровней сложности.

**Описание:** Godot Engine — открытый кроссплатформенный игровой движок для разработки 2D/3D-видеоигр и приложений для ПК, мобильных устройств, веб-платформ. Адаптирован ко всем распространенным операционным системам, включая Linux, macOS, Windows, Android и iOS. Она представляет мощный редактор, систему визуального скриптинга, а также обширную библиотеку ресурсов и инструментов для разработки.   
  
  
  
  
  
  
  
  
  
  
  
  
  
 **2. Актуальность проблемы  
  
Актуальность:**Рынок видеоигр стремительно растёт, и спрос на качественные и доступные инструменты для создания увеличивается. Многие разработчики ищут бесплатные и открытые решения, позволяют легко начать работу без необходимости приобретения дорогостоящих лицензий.   
  
**Проблематика:**Несмотря на популярность коммерческих игровых движков, таких как Unite и Unireal Engine, существует потребность в альтернативах, которые предлагают гибкость, открытость кода и простату использования. Godot Engine отвечает таким требованиям, предоставляя мощные возможности для разработки игр при минимальных затратах.

**3. Цели и задачи работы  
  
Цель:** Исследование возможностей и особенностей Godot Engineдля разработки видеоигр, а также оценка его применимости в реальных проектах.  
 **Задачи:**   
1. Изучение архитектуры и функционала Godot Engine.  
2. Анализ рабочего процесса и инструментария.  
3. Создание демонстрационного проекта для оценки возможностей движка.  
4. Оценка производительности и совместимости с различными платформами.  
5. Подготовка рекомендаций по использованию Godot Engine в разработке игр.

**4. Команда и роли**

Работу выполнил студент 1-го курса Морозов Матвей. **5. Репозиторий и рабочий процесс**

**Репозиторий:**   
Для хранения исходного кода и управления версиями используется система контроля версии Git и хостинга на платформе GitHub.  
Алгоритм реализации проекта – это последовательность шагов и действий, которые необходимо выполнить для создания и разработки игрового проекта на платформе Godot Engine с использованием двумерной графики. Этот алгоритм включает в себя настройку проекта, создание сцены, управление персонажем, логику игры, а также другие необходимые действия, связанные с разработкой проекта в Godot Engine 2D.

**Рабочий процесс:**1. Планирование задач и создание дорожной карты.  
2. Разработка прототипа игры.  
3. Реализация основных механик и интерфейсов.  
4. Тестирование и откладка.  
5. Оптимизация и финальная сборка.  
6. Документирование и подготовка отчетной документации.  
 **Шаги для реализации проекта:**

Категории жанр игры и ее основные механики:

а) Жанр игры : Экшен с элементами выживание

б) Основные механики игры (движение, сбор предметов,

битвы с мобами).

Создать концепт игры:

Разработать идеи для игры и визуализировать ее в виде концепта

(например, описание игровой механики и визуальный стиль).

Определить необходимые ресурсы для создания игры.

Создать дизайн и прототип игры:

Создать дизайн игры, включая интерфейс, меню.

Создать игровые объекты и анимации:

а) Создать игровые объекты, такие как персонажи, враги, оружие и т. д.

б) Создать анимации для игровых объектов.

Написать код игры:

а) Написать код игры на языке программирования **GDScript из Godot.**

б) Реализовать основные механики игры, такие как управление

персонажем, взаимодействие с объектами и т. д.

Тестирование игры:

Протестировать игру на наличие ошибок и недочетов.   
  
 **6. Календарный план** **Этапы работы:**  
1. Исследование и анализ возможностей Godot Engine- 1 неделя  
2. Создание прототипа игры – 1 неделя  
3. Разработка основных механик и интерфейсов -3 недели.  
4. Тестирование и отладка -2 недели.  
5. Оптимизация и финальная сборка -1 неделя.  
6. Документация и подготовка отчета- 1 неделя.

**7. Инструменты разработки   
  
**  
Логотип игрового движка Godot1. Godot Engine — основной игровой движок.  
2. Pixel studio  
Некоторые отличия редакторов:  
5. Git/GitHub – система контроля версий и репозитории.  
Git — это распределённая система контроля версий, которая позволяет разработчикам отслеживать изменения в файлах и работать над одним проектом совместно с коллегами. Он работает локально на компьютере разработчика и не требует подключения к интернету.   
GitHub — это веб-сервис, основанный на Git. Он предоставляет графический интерфейс и облачное хранилище для проектов, использующих Git. GitHub также предлагает ряд дополнительных функций, таких как управление задачами и баг-трекинг.  
Разница между Git и GitHub в том, что Git — это инструмент, а GitHub — это сервис, который использует этот инструмент.   
Таким образом, Git — это инструмент для управления исходным кодом, а GitHub — это облачный сервис, который использует Git в качестве основы и предоставляет графический интерфейс для работы с Git, а также облачное хранилище для проектов.  
  
**Возможности Godot**

Встроенный функционал «Годо» позволяет разработчику с нуля создать полноценную игру или приложение без использования внешних инструментов.  
Работа с двух- и трехмерной графикой — поддержка эффектов отражения, динамических теней, статичного и динамичного глобального освещения, полноэкранной постобработки (засветки, глубины резкости, гамма-коррекции и т.д.).  
Поддержка реалистичной физики — системы частиц (дыма, тумана, пара, взрывов и т.д.), свойств динамичных и статичных тел, столкновений и разрушений, трассировки лучей и других физических процессов.  
Работа с анимацией — опции создания скелетной анимации, наложения объектов, кат-сцен в реальном времени.  
Сетка навигации (Navigation mesh) — алгоритм нахождения игровым агентом оптимального маршрута в сложном пространстве.  
Поддержка мультимедиа — воспроизведение аудио- и видеофайлов с помощью подключаемых кодеков Theora, OGG Vorbis, WAV.  
AR/VR — встроенный мобильный интерфейс дополненной и виртуальной реальности с использованием 3DOF-датчиков на телефоне.  
Подключение устройств ввода — клавиатуры, мыши, геймпада и сенсорного экрана.  
Процедурная генерация — автоматическое создание внутриигрового контента (окружения, NPC, объектов, оружия и т.д.) с помощью алгоритмов.  
Поддержка языков — «Годо» имеет свой собственный высокоуровневый язык программирования GDScript, также можно использовать С# и C++.

**8. Процесс разработки**

Этап разработки графического оформления игры

Проект разработки игры включал в себя два основных этапа: создание

графического оформления и написание кода игры. Эти этапы были выполнены

параллельно, поскольку написание кода не требовало наличия готовых

анимаций и других элементов графического оформления. Кроме того,

использование стандартных примитивов Godot Engine для проверки работоспособности

кода ускорило процесс разработки.

Важными понятиями в графическом оформлении были тайлы и спрайты.

Тайлы представляют собой небольшие повторяющиеся элементы, которые

применяются в создании крупных изображений, таких как уровней для 2D-игр.

Во-вторых, они являются растровыми изображениями, которые служат для

анимации объектов.

**9. Результат**  
 Создано приложение на базе Godot Engine, демонстрирующее основные возможности движка.

Игра имеет название: Leonid. Leonid – это 2D игра в жанре экшен с элементами выживания, где игроки погружаются в мир, полный опасностей. Задача героя выжить в фантастическом лесу и уничтожить злых созданий в нём обитающих. Волны агрессивных мобов нападают на вас со всех сторон, и ваша единственная надежда – эффективно использовать оружие и навыки, чтобы отбиваться от них. Также вы будете собирать монеты и тратить их на разные призы и бонусы.  
 Проект включает в себя:  
- пространство игрового мира  
-управление персонажами  
-механику взаимодействия с объектами  
- элементы пользовательского интерфейса  
- звуковые эффекты и музыка/

**10. Заключение и выводы**Выводы: Godot Engine оказался мощным и удобным инструментом для разработки видеоигр. Он предлагает широкий спектр возможностей, включая поддержку 2D и 3D -графики, встроенный редактор сценариев и удобные средства для создания пользовательских интерфейсов.  
Рекомедации: Использовать Godot Engine для небольших и средних проектов, особенно для инди-разработчиков и команд с ограниченным бюджетом. Движок хорошо подходит для быстрого прототипирования и создания кросс-платформенных игр.