

Домашнее задание

Общее описание

ВОДА. ЗЕМЛЯ. ОГОНЬ. ВОЗДУХ.
КОГДА-ТО ДАВНО, СТУДЕНТЫ ЖИЛИ В МИРЕ...
НО ВСЁ ИЗМЕНИЛОСЬ, КОГДА ИМ ВЫДАЛИ ДОМАШНЕЕ ЗАДАНИЕ...

Необходимо реализовать 3D игру в жанре action-adventure с элементами rogue-like (<https://ru.wikipedia.org/wiki/Roguelike>) с видом от третьего лица (можно использовать шаблон третьего лица UE5).

High-концепт

Вы в роли члена Гильдии Свободных Продавцов Капусты. В ходе неудачного стечения обстоятельств (как и всегда в этом бизнесе) вы и ваша любимая капусточка разделились в загадочном подземелье, наполненном ловушками. Нужно пройти все комнаты и найти ваше дорогое сокровище.

Игра представляет из себя экшен-адвенчуру с видом от третьего лица. Игровое пространство должно быть случайно сгенерированным и разбитым на комнаты. Внутри комнаты должны располагаться случайно расставленные ловушки (то есть при каждом новом запуске игры/уровня должно формироваться отличающееся от предыдущего варианта пространство). У персонажа игрока должен быть индикатор здоровья, который уменьшается при столкновении с ловушками.

Итоговая цель — добраться до последней комнаты и коснуться объекта-символа окончания игры. После каждой пройденной комнаты перед игроком встаёт выбор из двух следующих комнат, одна из которых более сложная, чем текущая, но в ней можно восполнить здоровье, другая — такая же по сложности, но в ней не будет возможности восполнить здоровье. По ходу прохождения подземелья игрок должен грамотно выбирать в какую из них пойти. Данный менеджмент здоровья станет среднесрочной целью, а успешное завершение комнаты и уклонение от ловушек — краткосрочной.

Референсы

Структурный референс (генерации уровней): [Hades](#).

Геймплейный референс: Lego-игры — самое близкое, что, наверное, подходит под геймплейный референс, к примеру, [Lego Indiana Jones](#). Однако, не ориентируйтесь на этот референс слишком сильно, подойдёт любой на ваш взгляд экшен-адвенчура/3D платформер.

Задачи

Основные задачи:

Моя капуста? МОЯ КАПУСТААА!

Про оценивание пунктов - ниже.

- 1) Персонаж должен уметь корректно передвигается вперед-назад-влево-вправо и прыгать;
- 2) У персонажа должен быть индикатор здоровья на ваше усмотрение (например, шкала, сердечки здоровья, общее освещение уровня);
- 3) С точки зрения игрока, геймплей должен состоять из одного уровня, внутри которого будет от 6 до 10 комнат, которые должны генерироваться по следующим правилам:
 - a) У комнаты должен быть вход и два выхода;
 - b) Вернуться в дверь, из которой персонаж пришел в комнату – нельзя;
 - c) Один из двух выходов должен вести в следующую комнату такой же сложности, как и текущая; второй выход должен вести в комнату большей сложности, содержащий в себе объект, восстанавливающий здоровье (соответственно нужно как-то отметить для игрока тип комнаты);
 - d) Суммарное количество комнат, проходимых персонажем за игру, для каждой новой игровой сессии должно выбираться случайным образом в начале игры и должно находиться в диапазоне [6; 10];
 - e) Если текущая комната является последней, то в ней должен располагаться объект, символизирующий окончание игры, например, тележка с КАПУСТОЙ. Но можно использовать любой другой символ окончания игры на ваше усмотрение;
- 4) В каждой комнате обязательно должны быть ловушки. Также в каждой комнате могут присутствовать баффы и/или дебаффы (в зависимости от вашей реализации);
- 5) Ловушки:
 - a) должны быть представлены двумя типами игровых объектов:
 - i) тип 1: движется по заранее заданной траектории;
 - ii) тип 2: не движется, является статичным;
 - b) оба типа ловушек должны обрабатывать столкновение с персонажем игрока, при котором персонажу наносится урон (количество урона на ваше усмотрение и балансировку);
- 6) Баффы (бусты, бонусы) должны быть представлены несколькими типами игровых объектов, которые располагаются на полу и, при контакте с персонажем игрока, «поглощаются» им. Нужно реализовать:
 - a) Бафф, замедляющий скорость всех движущихся объектов на короткий промежуток времени (промежуток должен быть выведен вами эмпирически при тестировании игры);
 - b) Бафф, дающий игроку возможность проходить сквозь статические ловушки без вреда для здоровья на короткий промежуток времени (промежуток должен быть выведен вами эмпирически при тестировании игры);

- 7) Дебаффы должны быть представлены одним типом игрового объекта. Они должны располагаться на полу и при контакте с персонажем игрока «поглощаться» им. Нужно реализовать:
- а) Дебафф, замедляющий персонажа игрока на короткий промежуток времени (промежуток должен быть выведен вами эмпирически при тестировании игры);
- 8) В игре также нужно реализовать объект, который случайным образом превратится в бафф или дебафф после его «поглощения» персонажем игрока, то есть игрок заранее не должен знать, что ему выпадет;
- 9) Как было отмечено, у комнат должно быть два типа сложности. Сложность комнаты должна выстраиваться на ваше усмотрение с помощью следующих параметров:
- а) скорость движущихся ловушек;
 - б) отсутствие/наличие баффов и дебаффов;
 - в) количество ловушек, баффов и дебаффов;
 - г) время действия баффов и дебаффов.
- Собственные параметры реализации разных уровней сложности не возбраняются;
- 10) В более сложных комнатах должен всегда располагаться игровой объект (например, в виде сердечка или кочана капусты, лежащего на земле), который будет восполнять N здоровья игроку (значение N подбирается на ваше усмотрение).
- 11) Должна быть предусмотрена удобная настройка всех игровых параметров инструментами UE5 (например, продолжительность и эффект того или иного баффа, урон ловушек, восстановление здоровья из пункта 10...). Т. е. вместо того, чтобы задавать значение константы внутри кода, следует создать таблицы с параметрами, лежащими отдельным файлом и при старте игры подтягивающимися внутрь. Соответственно, не должно быть различных типов акторов, которые отличаются лишь параметрами, но не логикой.

Дополнительные задачи:

Хватит терпеть! Продавцу капусты пора научиться давать сдачи!

- 12) Реализация примитивной боевой системы, с помощью которой персонаж сможет одним или несколькими ударами/выстрелами уничтожить ловушку или предмет интерьера, если он будет предусмотрен вашей игрой (тип оружия не принципиален);
- 13) Введение типов прочности для объектов, чтобы персонаж мог уничтожать не все объекты в комнате, а наименее прочные (например, объекты из условного «дерева» подлежат уничтожению, объекты из «металла» разрушить нельзя)

Добавление собственных плюшек (системы покорения стихий, баффов, дебаффов, ловушек, предметов мебели, меню паузы, возможность перезапуска игры и т.д.) сверх поставленных задач (и не идущих им вразрез) может положительно повлиять на итоговую оценку. Креатив приветствуется, хоть и не обязателен для получения максимальной оценки :)

Требования к документации

Без неё не пропустят на таможне...

Документация должна содержать:

1. Вашу Фамилию и Имя;
2. Подробное текстовое описание алгоритмов генерации комнат и их содержимого;
3. Описание принципа изменения уровня сложности в вашей игре;
4. Описание всех значимых реализованных сверх условия дополнений;
5. Ссылки на готовые проекты, из которых были взяты части кода с пояснениями (что именно взяли, почему, что изменили).

Форма представления результата

Главное, чтобы вся капуста доехала...

На проверку нужно отправить следующее:

1. Текстовый файл README, содержащий проектную документацию вашей игры
2. Ссылка на репозиторий (Git):
 - a. Репозиторий должен содержать правильные «.gitignore» и «.gitattributes» файлы (<https://github.com/MOZGIII/ue5-gitignore>);
 - b. Не должно быть лишних папок и файлов (папки DerivedDataCash, Intermediate, Saved).

Рекомендация: при работе с GitHub советуем использовать GitHub Desktop или SmartGit, потому что часть аналогов (Git Bash, Git Kraken) не всегда может загрузить проект из-за большого объема данных.

Оценивание

Зная все правила безопасности, можно стать успешным продавцом капусты.

В этом домашнем задании **необходимо** использовать Gameplay Framework (<https://docs.unrealengine.com/5.0/en-US/gameplay-framework-in-unreal-engine/>)!

Если реализованная логика не находится в нужных классах — **вычитаются баллы**

Проекты, которые не собираются, не компилируются или не запускаются, оцениваются неудовлетворительно — **максимальный итог 1 балла**

Нельзя достичь конечной цели игры (т.е. из-за несбалансированной генерации объектов нельзя пройти комнаты) — **максимальный итог 3 балла**

Оценка от 4 до 6 будет рассчитываться по следующим критериям:

- Реализованы пункты 1-5, 9-10 в разделе «Основные задачи»;
- Представлена релевантная проектная документация;
- В проекте присутствует корректная декомпозиция;
- Код содержит необходимые комментарии.

Для получения **оценки от 7 до 8** необходимо:

- Соблюсти все требования, предъявляемые к работе на оценку 4-6;
- Реализовать баффы и дебаффы (пункты 6-8 в разделе «Основные задачи»).
- Реализовать настройку параметров через таблицы (пункт 11 в разделе «Основные задачи»).

Для получения **оценки от 9 до 10** баллов необходимо:

- Соблюсти все требования, предъявляемые к работе на оценку 7-8;
- Внедрить боевую систему (пункты 12-13 в разделе «Дополнительные задачи»)

Сдача

Крайний срок сдачи работы — **13 ноября в 23:59**.



Да придёт с Вами Сила.