

Основы разработки компьютерных игр на Unreal Engine 4 с помощью технологии blueprints.

Summary-шпаргалка по групповому проекту.

Атяпшева Татьяна, 2023, ИТМО

Game Designer:

Зона ответственности: ведение проекта от и до, ведение проектной документации, коммуникация в команде, презентация (лицо) проекта.

- **Пайплайн:** какие этапы у вас были в разработке у какого специалиста, делали ли что-то параллельно или последовательно (опционально: Диаграмма Ганта).
- **Фичи:** продумывание механик (как меняется скорость, цвет; специальные плитки с бафами или дебафами и т.д.), описание в диздоке, обсуждение с коллегами (совместно с Programmer).
- **Дизайн:** продумывание визуальной составляющей, основных концептов, обсуждение с коллегами, описание в диздоке (совместно с 3D-Artist/Level Artist).
- **Управление проектом:** описание задач; обсуждение идей с коллегами; принятые решения, описанные в диздоке; дедлайны и анализ (убирались ли в сроки, или все вовремя и т.д.).
- **Moodboard:** доска настроения, отражающая дизайн и игру в целом; заявить стиль, жанр и сеттинг вашего проекта (совместно с 3D-Artist/Level Artist).
- **Диздок:** ведение дизайнерского документа, состоящего (примерно) из: аудитория игры, moodboard, сториборд (опционально, если придумаете историю и сможете ее привязать к игре), продумываемые фичи с анализом и результатом принятия решения (реализуем, не успели, отложили), на какие платформы будет портироваться проект, планируемые расширения в будущем, программы и инструменты для создания игры, отдельно про команду.
- **Презентация:** подготовка итоговой презентации (в идеале в стиле игры, совместно с 3D-Artist/Level Artist), защита проекта. 15 минут выступление (жестко) + 5 минут на вопросы.

Programmer:

Зона ответственности: ведение проекта в Unreal Engine 4, GitLab, реализация фич, итоговая сборка и отладка финальной версии проекта.

- **Базовый проект:** воссоздание проекта по мастер-классу.
- **GIT:** создание открытого репозитория на команду, консультация команды и помощь при необходимости по работе с системой контроля версий.
- **Реализация фичей:** реализация функциональных фич, которые решили делать при обсуждении с Game Designer.
- **Иерархия проекта:** ведение иерархии проекта: соблюдение нейминга папок, ассетов, blueprints.
- **Фиксинг багов:** исправление ошибок, найденных QA-engineer.
- **Билд:** создание результирующего билда игры и смокинг тест (смокинг тест - это тест на то, что игра в принципе запускается).

3D-Artist:

Зона ответственности: визуальная целостность проекта, корректный импорт и отображение моделей, корректность созданных материалов игры, красивая презентация, соответствие заявленному визуальному стилю.

- **Реализация дизайна:** оформление сцены в выбранном дизайне.
- **Moodboard:** реализация доски настроения совместно с Game Designer.
- **Создание 3D-моделей:** реализация новых 3D-моделей в Blender, создание материалов в Unreal Engine.
- **Поиск и импорт 3D-моделей:** поиск и внедрение подходящих 3D-моделей в игру (бесплатные модели берем из стора Epic Games).
- **Презентация:** подготовка итоговой презентации, ее визуального стиля.

QA-engineer:

Зона ответственности: выпуск проекта без багов, отслеживание соответствия проекта заявленному в дизайне, ведение важного документа для отслеживания развития проекта, финальные баллы команды при нахождении багов преподавателем в финальной версии игры.

- **Поиск функциональных ошибок:** поиск ошибок, связанных с исправной логикой фичей, коллизий, проверка соответствия проекта дизайнерскому документу.
- **Поиск визуальных ошибок:** поиск ошибок, связанных с 3D-моделями, правильным отображением; внимательное изучение сцены, подмечать "выбивающиеся" модели из дизайна.
- **Ведение bug-report:** создание и заполнение документа, в котором описываются найденные ошибки, уровень ошибки, какие подходы использовал, как пытался "сломать" игру, с отметкой времени на тесты. Сколько часов тестировал, что пытался сломать в первую очередь. Что получилось сломать, с какими проблемами и багами столкнулся. Таблица тест кейсов, что проверял по списку. Должен быть один блокирующий репорт (при отсутствии - придумать и верно описать). Проверка нейминга в иерархии проекта (проверка работы Programmer). В одном из разделов описать какие дебажные консольные команды (пример, slomo 4) использовал при тестировании игры.