unity

unity

Certified

Expert

Gameplay

Programmer

# Цели экзамена

Сертифицированный специалист Unity — программирование игрового процесса

# Значение

Программисты игрового процесса специализируются на реализации игрового дизайна. Они вдыхают в игру жизнь, создавая скрипты к ассетам, сделанным командой художников. Они ответственны за работу над пользовательским интерфейсом, окружением, персонажами и объектами, равно как за оптимизацию и скриптинг, необходимый для работы игровой механики. Они также разрабатывают функционал неигровых персонажей (NPC), которые населяют игру.
Программисты игрового процесса плотно работают с дизайнерами (или сами являются ими), и в своей работе полагаются на дизайн-документ игры (GDD). Эти люди отлично подкованы технически и понимают, что нужно для того, чтобы претворить идеи в жизнь. К этому относится умение использовать компоненты Unity, писать свои компоненты, настраивать префабы и организовать каждую сцену. Все это они делают с учетом технического дизайн-документа (TDD).

## Подходящие должности:

- программист игрового процесса;
- игровой скриптер/инженер;
- дизайнер уровней.

# Требования

Сертификация рекомендуется тем, у кого за плечами уже есть несколько лет работы в этой области и разносторонний практический опыт, например:

- опыт работы в студии разработки компьютерных игр;
- опыт разработки захватывающих игр для мобильных, веб- и других платформ;
- хорошие навыки написания кода на C#;
- знание жизненного цикла компьютерных игр и опыт работы над проектами с нуля, от самого начала до выпуска игры;
- отличное понимание всех аспектов игрового дизайна (уровни, неигровые персонажи, игровые механики);
- опыт работы с сервисами Unity;
- понимание рабочего процесса в области игровой анимации и ассетов, включая создание персонажей и окружения;
- уверенные навыки организации файловой структуры, имен и соблюдения других установленных протоколов;
- умение создавать и использовать библиотеку префабов.

# Ключевые навыки

Сертификация «Специалист Unity — программирование игрового процесса» подтверждает, что кандидаты обладают достаточными навыками для эффективной разработки и оптимизации игры. Успешно прошедшие сертификацию кандидаты обладают навыками в нижеследующих областях.

## Прототипирование

- Создание и оценка прототипов основного игрового процесса для RITE.
- Создание и внедрение объектов GameObjects для работы игровых механик.
- Создание основанной на GDD библиотеки префабов с использованием ассетов, созданных командой художников.
- Организация структуры папок для библиотеки префабов.

## Дизайн уровней

- Создание и разработка интерактивных префабов с помощью компонентов Collider и Rigidbody.
- Создание подходящего физического слоя для скрытия интерактивных префабов.
- Настройка префабов, появляющихся при выполнении программы, и внедрение динамического игрового процесса.
- Настройка и внедрение триггеров событий и заскриптованных событий с помощью компонентов Collider в качестве триггеров.
- Создание и скриптинг особых логических компонентов для объектов GameObjects, служащих для связи машин состояний с взаимодействиями/триггерами на сцене.
- Размещение и настройка эффектов на сцене, включая статическое и генерируемое по ходу выполнения программы размещение.
- Настройка LOD с учетом спецификации платформы.
- Оценка размещения компонентов объектов GameObject и их взаимосвязей с учетом спецификации платформы.
- Оценка актуальности методов стриминга и статической загрузки для сцены.
- Построение игровых уровней с несколькими сценами.
- Настройка видеовставок.

## Дизайн неигровых персонажей (NPC).

- Создание и разработка логики NPC и скриптов искусственного интеллекта (AI).
- Внедрение навигации и поиска пути для NPC с помощью компонентов NavMesh, NavMeshAgent, OffMeshLink и NavMesh-препятствий.
- Настройка типов области NavMesh и их стоимости.
- Настройка триггеров для включения/выключения областей NavMesh.
- Внедрение реакций агентов NavMeshAgent и симуляция толпы.
- Оценка размещения NPC и их взаимосвязей с учетом спецификации платформы.
- Настройка покадрового аудио и эффектов в анимационных клипах.

## Работа с пользовательским интерфейсом и игровой механикой.

- Использование системы анимации для контроллеров аниматоров, машин состояний и скриптов игровых механик.
- Оценка и оптимизация компонентов Collider, Rigidbody и физических материалов для интерактивных объектов GameObjects.
- Внедрение внутриигровых интерфейсов: HUD, мини-карт, радаров, показателей здоровья и других элементов.

## Оптимизация производительности и целевые платформы

- Внедрение загрузки и размещения на игровых уровнях комплектов ассетов.
- Разработка и улучшение схем ввода и контроллеров для разных платформ и/или систем виртуальной реальности (VR).
- Анализ объектов GameObjects и сцен для потоковой оптимизации и оптимизации хранения с учетом спецификации платформы.
- Оптимизация Occlusion Culling на сценах.
- Отладка и тестирование игровых уровней во время выполнения программы.

## Внедрение сервисов Unity: Ads, In-App Purchases, Analytics.

- Внедрение простой и вознаграждаемой рекламы Unity.
- Внедрение встроенных покупок Unity.
- Интеграция Unity Analytics с учетом GDD и TDD.
- Настройка мониторинга событий с использованием Analytics для отслеживания поведения игрока.
- Анализ и оценка существующих уровней; предложение изменений с учетом данных Analytics.
- Использование Unity Performance Reporting для улучшения и оптимизации сборок игры для каждой платформы.

# Темы сертификационного экзамена

**Прототипирование (основной игровой процесс для RITE)**

- Прототипирование основного игрового процесса.
- Конфликты и их решение на стадии прототипа.

**Программирование игровых уровней**

- Настройка физики.
- Рейкастинг.
- Скриптинг префабов, появляющихся во время выполнения программы.
- Логика уровней, модели поведения.
- Наполнение уровня системами частиц и эффектами.
- Оптимизация платформы.
- Загрузка и выгрузка сцен.
- Методы отображения видеовставок.

**Программирование и дизайн NPC**

- Логика NPC и их поведение.
- Навигация и поиск пути.
- Появление и размещение NPC.

**Пользовательский интерфейс**

- Система координат UI и скриптинг.

**Оптимизация производительности и целевые платформы**

- Оптимизация рендеринга.
- Загрузка и настройка комплектов ассетов.
- Отладка игрового процесса.
- Различия платформ и их влияние на игровой процесс.

**Внедрение сервисов Unity**

- Unity Ads.
- Unity In-App Purchasing.
- Unity Analytics.
- Unity Cloud Build.

# Примеры вопросов

## Вопрос 1

Программист создает прототип игры, представляющей собой шутер-сайдскроллер. Главный персонаж — самолет, а враги — разные типы НЛО. Когда игрок уничтожает НЛО, на его месте показывается взрыв. Когда умирает игрок, показывается взрыв, характерный только для него.

Самолет может выпустить на экран до 64 снарядов. Максимальное количество НЛО любого типа на экране равно 128. Максимальное количество снарядов НЛО на экране равно 1024. Максимальное количество взрывов на экране — 128.

Программисту нужно определить, какой тип объектов GameObjects необходимо разместить в редакторе, а какие должны появляться при выполнении программы (т. е. из пула объектов).

**Как это можно сделать?**

**A** Игрок, НЛО, эффект взрыва игрока и эффект взрыва НЛО должны быть размещены в редакторе. Снаряды игрока и НЛО должны появляться при выполнении программы.

**B** Игрок, НЛО, снаряды игрока и снаряды НЛО должны быть размещены в редакторе. Эффект взрыва игрока и эффект взрыва НЛО должны появляться при выполнении программы.

**C** Игрок и НЛО должны быть размещены в редакторе. Снаряды игрока, снаряды НЛО, эффекты взрыва игрока и эффекты взрыва НЛО должны появляться при выполнении программы.

**D** Игрок и эффект взрыва игрока должны быть размещены в редакторе. НЛО, снаряды НЛО, снаряды игрока и эффекты взрыва НЛО должны появляться при выполнении программы.

# Вопрос 2

Дизайн-документ игры описывает игру от третьего лица с открытым миром. GDD определяет два типа игрового процесса для игрока:

1. Перемещение по миру пешком.
2. Перемещение по миру на мотоцикле.

Относительная высота каждого режима одинакова, но когда игрок на мотоцикле, он может передвигаться значительно быстрее.

Программист определяет несколько областей, которые надо доработать во время перехода с ходьбы на езду и наоборот:

- FOV камеры.
- Дистанции LOD.
- Стриминг уровня.

**Какого метода должен придерживаться программист, чтобы выполнить все задачи, когда игрок передвигается на мотоцикле?**

**A** Значение FOV камеры должно быть уменьшено.
Стриминг уровня должен происходить гораздо быстрее.

**B** Значение FOV камеры должно быть увеличено.
Стриминг уровня должен происходить гораздо медленнее.

**C** Значение дистанций LOD должно быть увеличено.
Стриминг уровня должен происходить гораздо медленнее.

**D** Значение FOV камеры должно быть увеличено.
Стриминг уровня должен происходить гораздо быстрее.

# Вопрос 3

В дизайн-документе описывается игра в большом городе, в котором игрок может ходить и взаимодействовать с кем угодно. Игрок может проходить случайные задания горожан. У игрока может быть только 5 активных заданий за раз. Если игрок будет идти из одного конца города в другой, это займет в среднем 4 часа.

В HUD игры есть мини-карта с видом сверху, на которой отображаются двухмерные значки, указывающие на:

- Местоположение игрока.
- Активные задания.
- Всех граждан в городе.

Мини-карту можно приближать и отдалять. При максимальном приближении она показывает только значок игрока. При максимальном отдалении можно видеть 25% всего города. При максимальном приближении производительность игры заметно падает.

**В чем причина?**

**A** Слишком много вызовов на выгрузку текстур значков граждан из памяти.

**B** Значки активных заданий опрашиваются слишком часто, чтобы понять, сколько заданий активно на данный момент.

**C** Алгоритм обрезания мини-карты слишком долго обрезает остальной город.

**D** HUD слишком сложен и воспроизводится с каждым кадром.

# Вопрос 4

В дизайн-документе описывается мобильная игра с маленьким бюджетом. Для хорошей производительности целые секции игры загружены в комплекты ассетов. Каждая секция независима, поэтому комплекты ассетов можно загружать и выгружать, не опасаясь за другие секции.

Тем не менее, время от времени в игре появляются розовые текстуры.

**В чем может заключаться причина?**

**A** Не работает AssetBundle.mainAsset.

**B** AssetBundle.Unload() вызывается слишком рано.

**C** AssetBundle.LoadAllAssets() вызывается с неправильным типом.

**D** AssetBundle.LoadAssetWithSubAssets() вызывается с неправильной строкой и типом.

# Вопрос 5

В дизайн-документе описывается мобильная 3D-головоломка с более чем 300 уровнями. Для каждого уровня в игре содержится специальное аналитическое событие:

1. levelStarted срабатывает, когда игрок начинает уровень.
2. levelCompleted срабатывает, когда игрок заканчивает уровень.

Команда хочет определить 5 самых сложных уровней в игре. Уровни ограничены по времени.

**Какие дополнительные события нужны для определения 5 самых сложных уровней?**

**A** Добавить levelTime, которое будет передавать количество времени, проведенное на уровне.

**B** Добавить levelRestarted, срабатывающее, когда мобильное приложение закрывается оперативной системой.

**C** Добавить levelFailed, срабатывающее, когда игрок проигрывает или выходит слишком рано.

**D** Добавить levelResumed, срабатывающее, когда мобильное приложение восстанавливается из фонового процесса.

---

Правильные ответы: D, D, C, B, C.