

РОЗРОБКА ДОДАТКІВ НА UNITY3D ПІД ОС ANDROID

Unity — це інструмент для розробки двох- і тривимірних додатків та ігор. Створені за допомогою Unity програми працюють під операційними системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а також на ігрових приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One, та в браузерях за допомогою спеціального модуля Unity (Unity Web Player) і реалізацією технології WebGL. Проект в Unity ділиться на сцени (рівні) - окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як об'єкти, так і порожні ігрові об'єкти - ті, що не мають моделі. Об'єкти містять набори компонентів, з якими взаємодіють скрипти. Так, у будь-якого об'єкта на сцені обов'язково присутній компонент Transform - він зберігає в собі координати місця розташування, повороту і розмірів об'єкта по всіх трьох осях. Unity 3D підтримує систему рівня деталізації (LOD), суть якої полягає в тому, що на далекій відстані від гравця високодеталізовані моделі замінюються на менш деталізовані, і навпаки, а також систему включає в себе оклюзію вибракування, суть якої в тому, що у об'єктів, які не потрапляють в поле зору камери не візуалізується геометрія і колізія, що знижує навантаження на центральний процесор і дозволяє оптимізувати проект.

Двигун гри підтримує безліч популярних форматів, таких як:

.3ds, .max, .obj, .fbx, .dae, .ma, .mb для тривимірних моделей;
.mp3, .ogg, .aiff, .wav, .mod, .it, .sm3 для звукових файлів;
.psd, .jpg, .png, .gif, .bmp, .tga, .tiff, .iff, .pict, .dds для зображень;
.mov, .avi, .asf, .mpg, .mpeg, .mp4 для відеофайлів.
.txt, .htm, .html, .xml, .bytes для тексту

Приклади дій для оптимізації гри.

1. Використовувати якомога менше текстур і матеріалів. Бажано, щоб більшість об'єктів використовували один і той же матеріал. Він може бути використаний на об'єктах двома способами: через `renderer.material` і `renderer.sharedMaterial`. Але в кожному з них має бути `<= 300 Vertex` (саме Vertex, а не трикутників), інакше він не братиме участі в Batch.

2. Об'єднувати статичні об'єкти, що використовують один і той же матеріал, в один загальний Mesh.

Якщо фонова музика одна, то цілком доречно не пакувати трек в пам'яті і читати його після старту сцени.

Ігровий світ:

1. Використовувати тільки одне джерело світла для освітлення, для затемнення окремих частин карти краще використовувати Lightmap.

2. Намагайтеся якомога рідше, або взагалі не використовувати об'єкти з RigidBody.

3. Fog - дає сильне навантаження

4. Шейдери для матеріалів намагаються використовувати з категорії Mobile /

Terrain:

1. Використовувати невеликі розміри террейн ~ 150x150 units

2. Pixel Error - виставити з 0 до максимального

3. Відключати прапорець Draw

4. Не використовувати малювання трави в інструментах Terrain, бо це в рази зменшує кількість кадрів в секунду

Скрипти:

1. Всі розрахунки проводити виключно в Update.2.

2. Не використовувати ініціалізатора new (типу `new Vector3(x, y, z)`, `new Rect()`) в циклах і часто викликаються місцях

3. Використання `CompareTag` для пошуку тега більш ніж удвічі швидше прямої перевірки тега по імені.

4. Конструкція визначення відстані (`this.Transform.position - target.position`).`sqrMagnitude` швидше `Vector3.Distance`

Новизна даного проекту заключається в реалізації максимальної кількості технологій оптимізації, при підтримці на високому рівні графіки і UI / UX дизайну. Так само у використанні нових алгоритмів в шейдерах надаються Unity3d v5 з «коробки». Також новітнє використання двох, на перший погляд, несумісних жанрів як tower defense з додатками елементів RPG, дає нові можливості сюжетного геймплею, завдань, та шляхів розвитку користувачу. Також реалізація невід'ємного елементу RPG, як змагання між користувачами, надає програмному продукту додаткову зацікавленість серед користувачів.

Експорт з Blender в Unity3d. Blender використовує правосторонню систему координат з віссю Z, спрямованою вгору. Таку систему координат використовують майже всі САПР.

Unity3d як майже будь-який інший ігровий движок використовує лівосторонню систему координат. Коли Y спрямований вгору, X - убік, Z- вперед.

Теорія СТВОРЕННЯ 3d моделей в Blender

Для початку треба вивчити структуру папок і файлів нашої 3d моделі в Blender.

Refence - папка в якій можуть зберігатися допоміжні креслення, малюнки, які допомагають побудувати 3d модель

Текстури - папка в якій повинні зберігатися текстури.

Flags_snow.blend - 3d наша модель в форматі Blender

Тепер для правильного експорту 3d моделі, ми повинні для себе з'ясувати, що Z + це буде напрямок в грі коли об'єкт рухається вперед, (Z-) - напрям коли об'єкт рухається назад. (X + -) це напрям повороту убік - але це в Unity, яка використовує лівосторонню систему координат. А в правосторонній системі координат (Blender, 3ds Max) Y + це буде напрямок коли об'єкт дивиться вперед і (X + -) це напрям убік. Це означає що об'єкт повинен стояти в сцені так, як на малюнку, в іншому випадку чекайте проблем з фізикою. Всі параметри обертання повинні стояти на 0 для подальшої роботи і для правильної генерації Меша (Mesh), треба за шкалою встановити замовчуванням. Для цього ми зробимо такі дії в Blender.

1) Виділимо всі об'єкти в сцені (клавіша A)

2) Об'єкт > Застосувати > Rotate & Scale.

Тепер всі параметри обертання прийняті і встановлені на 0. А параметри масштабування (Scale) встановлені на за

замовчуванням. З'ясувалося що призначати текстуру, краще в самому Unity3d, а не в Blender, в іншому випадку, при кожному експорті FBX файлу у вас буде створюватися папка object.fbm (порожня папка яка ні на що не впливає, крім як створює зайвий об'єкт у вікні проекту.

Експорт з правильною орієнтацією моделі

Тепер можемо робити «Правильний експорт» з правобічної системи координат в лівосторонню систему координат.

- 1) Виділимо всі об'єкти (клавіша A)
- 2) тиснемо послідовність R> X> -90.
- 3) Об'єкт> Застосувати> Повернути.
- 4) тиснемо послідовність R> X> 90.

Обов'язково робіть в даній послідовності. Далі екпортуємо модель в Unity. Файл> Експорт> FBX з наступними настройками:

Швидкий експорт файлів з Blender в Unity3d.

Після експорту моделі:

1) Йдемо в папку з цією моделлю і копіюємо папку Текстура (папку з текстурами) і наш FBX файл.

2) Вставляємо в папку активів Нашого ПРОЕКТ. Це дозволить Unity імпортувати модель прямо з текстурами, які були налаштовані в Blender.

Чому кількість вершин по-різному в Unity відображається і Blender? Як ми бачимо на малюнку кількість вершин (Verts) на об'єкті в Unity3d 28 показує. А кількість вершин в Blender = 8. Пов'язано це з тим, що в Unity3d підраховується кількість вершин самої фігури і кількість вершин в UV раз вертка. При створенні швів, одна грань розділяється на 2 грані, а слідує математики 1 грань складається з 2 вершин, отже з 2-ох граней буде 4 вершини. Для прикладу скористаємося Smart UF Project рядків і встановимо Максимальний граничний кут. Дивимося в Blender і кількість вершин скоротилося з 28 до 24 вершин. Якщо у кого є більш оптимізований спосіб. Пишіть в коментарях, обговоримо. Деякі скажуть що скоротилося всього на 4 вершини, а уявіть якщо таких об'єктів 100 або 200, або кількість вершин в більш деталізованих об'єктах скоротиться на 100-300. На мій погляд не поганий спосіб оптимізації. Як правильно зіставити масштаби моделі в Blender і Unity. Всі одиниці вимірювання, що в Blender Що в Unity за замовчуванням має одиницю Одиниці ВИМІРЮВАННЯ. 1 Одиниці = 1 метр. Тому розробники радять приймати 1 юніт ігрового простору за 1 реальний метр, і краще цього дотримуватися (хоча б щоб з фізикою менше возитися). І так розберемося з одиницями виміру в Blender, в Blender їх можна міняти, але за замовчуванням в Blender використовується Units. Бачимо Dimension параметр (Вимірювання або розмір). Розмір по X, Y, Z = 1,1,1; отже ця фігура куб з розмірами 1x1x1 метр. Екпортуємо в Unity і переходимо на вкладку інспектора. дивимося на параметр Scale Factor, якщо у вас Unity версії 5 менше, то за замовчуванням Scale Factor буде 0.01, виправимо цей параметр на 1, якщо у вас Unity5 то Scale Factor за замовчуванням буде 1.