

РОЗРОБКА ІГРОВОГО ДОДАТКУ ПІД ОПЕРАЦІЙНУ СИСТЕМУ ANDROID НА UNITY3D

Unity — багатоплатформовий інструмент для розробки дво- та тривимірних додатків та ігор, що працює на операційних системах Windows і OS X. Створені за допомогою Unity застосування працюють під системами Windows, OSX, Linux, Android, WebGL, iOS, BlackBerry, Tizen, XboxOne, Xbox360, PS3, PS4, PSVista, Windows Store, Samsung TV.

Будь-який ігровий двигун надає безліч функціональних можливостей, які застосовуються різних іграх. Реалізована на цьому двигуні гра отримує всі ці функціональні можливості, крім того, додаються її власні ігрові ресурси і код ігрового сценарію. Основні переваги перед іншими передовими інструментами розробки ігор у Unity є: продуктивний візуальний робочий процес і потужна міжплатформенна підтримка.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі. Двигун підтримує такі сценарні мови: C #, JavaScript. 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 (моделі), так і порожні ігрові об'єкти – тобто ті які не мають моделі. Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. Також у них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами). Так, у будь-якого предмета на сцені обов'язково присутній компонент Transform - він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктах з видимою геометрією також за замовчуванням присутній компонент Mesh Renderer, що робить модель видимою.

Графічний двигун Unity використовує спеціальний набір API (Application Programming Interface) функцій, розроблених для простого і ефективного вирішення завдань, пов'язаних з ігровим та відеопрोगрамуванням (DirectX, OpenGL, OpenGL ES). Підтримуються bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), динамічні тіні з використанням shadow maps, render-to-texture та повноекранні ефекти post-processing.

Unity підтримує файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic Substance. В ігровий проект Unity можна імпортувати об'єкти цих програм та робити налаштування за допомогою графічного інтерфейсу.

Скриптова система ігрового рушія зроблена на Mono — вільний відкритий проект з реалізації .NET Framework.

Програмісти можуть використовувати UnityScript (власна скриптова мова, подібна до JavaScript та ECMAScript), C# або Boo (мова програмування, подібна до Python).

У Unity3d існує цілий ряд подій, які виконуються в певному порядку. Наступні функції викликаються при запуску сцени (один раз для кожного об'єкта в сцені):

1. Awake: функція завжди викликається перед будь-якими функціями Start, а також тільки після того, як створено екземпляр prefab (якщо GameObject неактивний під час запуску, функція не викликається, поки об'єкт не стане активним);
2. OnEnable: функція викликається тільки після того, як об'єкт активується;
3. OnLevelWasLoaded: подія настає при завантаженні сцени.

Перед першим викликом Update настає подія Start лише якщо об'єкт активовано.

OnApplicationPaused настає між кадрами, параметр функції вказує на те, чи перебуває додаток в стані паузи.

Для побудови основної логіки гри використовуються функції Update:

1. FixedUpdate. Функція викликається фіксовану кількість раз на кадр незалежно від FPS (частіше чи рідше). Після FixedUpdate відбувається розрахунок фізики гри;
2. Update. Функція викликається одразу ж після обробки вхідних даних з частотою відповідною до FPS;
3. LateUpdate. Подія настає після завершення всіх Update. Зручно використовувати для зміни позиції та повороту камери від 3-го лиця.

Для рисунку GUI викликається функція OnGUI (викликається декілька разів на кадр).

Функції yield використовуються для обробки підпрограм, оскільки додаток Unity3D виконується в одному потоці. OnDisable настає при деактивації об'єкта. OnApplicationQuit викликається при завершенні роботи додатку.

Гра в Unity складається з декількох сцен, кожна з яких містить об'єкти типу GameObject. Об'єкти на сцені складають ієрархічну структуру. GameObject – основний клас для всіх об'єктів на сцені. Кожний GameObject має декілька компонентів (Component), що приєднанні до нього.

Обов'язковий компонентом є transform. Transform містить дані про абсолютну позицію об'єкта на сцені, позицію відносно батька, розмір об'єкта, абсолютний поворот об'єкта в сцені та поворот відносно батька. Компоненти реалізують логіку гри. Для створення власного сценарію необхідно унаслідуватися від класу MonoBehaviour, що є компонентом та може бути приєднаний до ігрового об'єкта на сцені.

Новизна даного проекту полягає в реалізації максимальної кількості технологій оптимізації, при підтримці на високому рівні графіки і UI / UX дизайну.