

Chuẩn bị

Unity

Dự án được thực hiện trên phiên bản Unity **2022.3.59f1 LTS** vì phiên bản này hiện tại (08/2025) đang hỗ trợ rất tốt các thư viện tracking, quảng cáo.

Các thư viện cho DOTS

Trong Unity cần cài thêm các thư viện cho phục vụ cho DOTS

- **Entities:** Đây là thư viện cốt lõi cung cấp nền tảng cho kiến trúc ECS.
- **Entities Graphics:** Thư viện phục vụ cho việc hiển thị các entities.
- **Burst:** Trình biên dịch hiệu suất cao, giúp tối ưu hóa C# với từng nền tảng.

Khái niệm cơ bản về trong DOTS

- **Entity:** Entity có thể hiểu là mã định danh. Nó không chứa dữ liệu hay hành vi. Nó chỉ báo cho hệ thống biết là "có cái gì đó đang tồn tại"
- **Component:** Không giống như component thường dùng trong lập trình OOP với Game Object. Component ở đây chứa dữ liệu thô không chứa các logic xử lý.
- **System:** Đây là nơi cho các logic và hành vi (behaviour). System sẽ tìm kiếm tất cả các Entity nào có một bộ Component nhất định để thực thi logic với các dữ liệu trong các component đó.
- **Baking:** là quá trình chuyển đổi dữ liệu từ GameObject sang ECS. Đây là cầu nối giữa thiết kế trực quan và ECS hiệu năng cao.
 - Baking được bắt đầu khi dev nhấn nút Play trên Editor.
 - Với product, baking được xử lý ngay trong lúc build game. Bản build không cần xử lý bake lại nữa.
- **SubScene:** Là một scene phụ được đặt bên trong scene chính chứa các authoring (gameObject). Các gameObject trong subscene sẽ được bake thành các entity.

Lưu ý

- Trong DOTS vẫn cần phải sử dụng đến **MonoBehaviour**. MonoBehaviour sẽ không còn là nơi chứa logic game mà sẽ đảm nhiệm việc:

- **Cầu nối để thiết kế và cấu hình.** Unity Editor được thiết kế xoay quanh các gameObject. Dev không thể kéo một Entities lên scene cũng không thể thấy component trong Inspector.
 - **Xử lý các hệ thống cũ của Unity** (ví dụ như `Input.GetKey()`, UGUI, Animation...)
-

Revision #3

Created 2025-09-13 11:48:11 UTC by ThanhDV

Updated 2025-09-14 07:55:37 UTC by ThanhDV