

DOTS and DinoHerdSurvival

Một dự án game endless run nhỏ để học về DOTS và UIToolkit

- [Chuẩn bị](#)
- [Tạo Entity đầu tiên với ECS](#)

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...)

Tạo Entity đầu tiên với ECS

DinoTag.cs

Tạo một struct trống kế thừa `IComponentData`. Struct này chỉ có tác dụng đánh dấu entity là Dino nên không cần data.

```
using Unity.Entities;

public struct DinoTag : IComponentData { }
```

MovementData.cs

Tạo một `IComponentData` chứa dữ liệu cần thiết cho việc di chuyển.

```
using Unity.Entities;
using Unity.Mathematics;

public struct MovementData : IComponentData
{
    public float Speed;
    public float3 Direction;
}
```

DinoAuthoring.cs

Authoring đóng vai trò như một bản thiết kế trên gameObject. Khi chạy game quá trình **baking** sẽ đọc thông tin từ file này để bake data từ một gameObject thành một Entity

```
using Unity.Entities;
using UnityEngine;

public class DinoAuthoring : MonoBehaviour
{
    [SerializeField] private float speed; public float Speed => speed;
}

class DinoAuthoringBaker : Baker<DinoAuthoring>
{
    public override void Bake(DinoAuthoring authoring)
    {
        Entity entity = GetEntity(TransformUsageFlags.Dynamic);
        MovementData movementData = new()
```

```

{
    Speed = authoring.Speed,
    Direction = new Unity.Mathematics.float3(0, 0, 1)
};
AddComponent<DinoTag>(entity);
AddComponent(entity, movementData);
}
}

```

Trong Editor:

- Tạo một Sub Scene bằng cách chuột phải trong cửa sổ Hierarchy chuột phải chọn New Sub Scene > Empty Scene
- Tạo một GameObject `_DinoArchetype` thêm component `DinoAuthoring` và thiết lập thông số.
- Kéo `_DinoArchetype` vào sub scene vừa tạo bên trên.

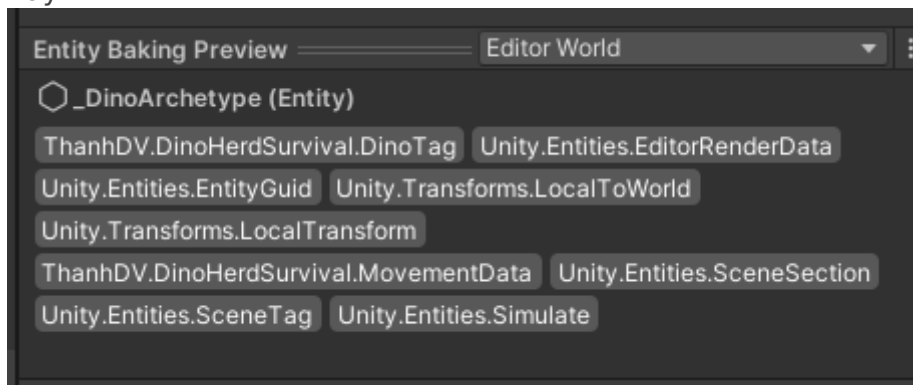
Tạo sub scene

Cũng có thể tạo sub scene bằng cách chọn các gameObject cần đưa vào sub scene sau đó chuột phải chọn New Sub Scene > From Selection

Như thế các gameObject được chọn sẽ tự động được thêm vào sub scene vừa tạo.

Kiểm tra Entity vừa tạo

- Nếu tạo đúng thì trong cửa sổ Inspector phía dưới cùng sẽ có Entity Baking Preview như thế này



- Hoặc cũng có thể mở cửa sổ Entity Hierarchy bằng cách Window > Entities > Hierarchy. Khi chạy game bạn có thể dùng Filter để tìm các Entity có `DinoTag` hoặc `MovementData`. Nếu tìm thấy tức là entity đã được tạo thành công.