Assignment:
DOTS in Unity

Future Games

Student
Pablo Schmidt

13-09-2024

Used in the game:
-Entities package
-Mathematics package
-ECS

This assignment has been done using DOTS (Data Oriented Technology Stack), in order to learn how to create better performing code, though to run on less powerful hardware.

Before starting the project, I added the following packages: “Entities” and “Mathematics” from Unity packages.

ECS (Entity Component System) design pattern was used. To simplify ECS, it can be said that it consist of 3 main parts:
-Entities
-Components
-Systems

It breaks down game entities into smaller components and systems to handle their behaviors more efficiently, especially in large-scale games.
The logic lives in the systems that define behavior for the entities that contain only the components that those systems care about.

An simplified description of the way the game was programmed:

1) **PlayerSystem.cs** as an ECS system which is responsible for the movement and shooting of the player.

2) **PlayerAuthoring.cs** is a MonoBehaviour script used to bake the player entity into the ECS system during the conversion from GameObjects to entities. ThePlayerAuthoring class sets up the necessary components and data (movement speed, shooting cooldown, bullet prefab) for the player entity.

- **PlayerBaker Class**: This is a "baker" class responsible for converting the PlayerAuthoring MonoBehaviour into an ECS entity during the conversion process. Inside the Bake method, it creates the player entity and adds the PlayerComponent to it.

3) **EnemySpawnerAuthoring.cs:** This MonoBehaviour specifies how often enemies should spawn (spawnCooldown) and provides a list of different enemy types (enemiesSO) (NOTE: This game is intended to be continued to make it a portfolio piece).

EnemySpawnerBaker: During the ECS conversion process, this class "bakes" the GameObject's data into an entity. It creates two components:

- EnemySpawnerComponent: Contains the spawn cooldown for controlling how often enemies spawn.
- EnemyDataContainer: Holds a list of enemy types with their attributes (health, speed, damage, etc.) to be used when spawning enemies.