

**CENG 462**  
**GAME DEVELOPMENT**  
**Final Report**

Sina Erdem Özdemir  
21050151019

## Introduction

This report presents a detailed analysis of the project developed using Unity's Data-Oriented Technology Stack (DOTS) and Entity Component System (ECS). The objective of the project is to create an efficient and scalable application using DOTS frameworks and address the challenges encountered during this process.

## Project Objectives

- Understand and utilize Unity's DOTS frameworks.
- Design an application adhering to the ECS paradigm.
- Demonstrate the advantages of DOTS in terms of performance optimization and scalability.

## Process

### 1. Setup and Environment Preparation

### 2. ECS Implementation

#### Entity Structure

- Translation: Position data.
- Rotation: Rotation data.
- Destination: Target position.
- MovementSpeed: Movement speed.

#### Systemler

- MoveToDestinationSystem: Ensures entities move toward their target positions.
- NewDestinationSystem: Assigns new targets to entities.
- PersonCollisionSystem: Handles collision detection and manages the shrinking of "brain".

### 3. Physics and Collision Systems

- Unity Physics Paketi: Used for collision detection and physical interactions.
- Collision Response: Triggers the shrinking of "brain" when collisions between entities are detected.

### 4. Performance Optimizations

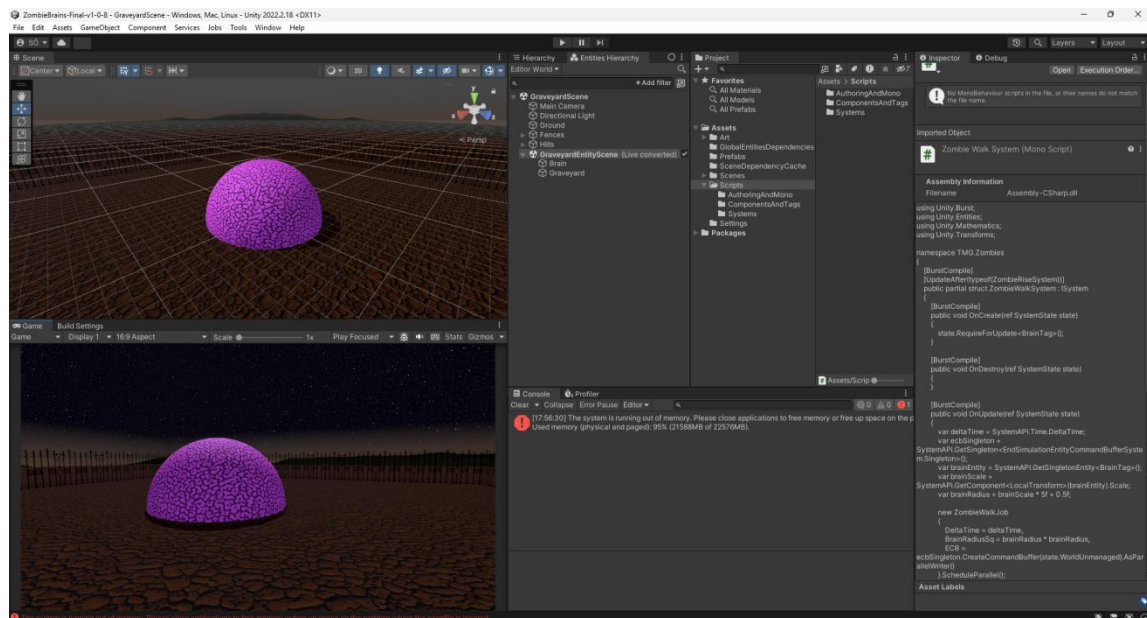
- Burst Compiler: Used to speed up the code.
- Job System: Tasks were processed in parallel by utilizing multiple CPU cores.
- ECS: Ensured organized data storage and more efficient memory usage.

## 5. Game-Level Design and Gamification (Incomplete)

- Creating a mix of Vampire Survivors and Tower Defense game.
- Brain clicker mechanics to generate damage through brain wave animations.
- Zombie health and death animations following damage.
- Enemy variations and score systems based on the level design.

## Challenges

- Steep Learning Curve of DOTS: DOTS and ECS required more technical expertise compared to the traditional Unity model.
- Lack of Support: Since DOTS is not yet fully complete, compatibility issues arose with features like animation and audio.
- Debugging: Debugging was more complex due to parallel processing.
- Memory Issues: Memory problems occurred during multi-system usage.
- Project Completion: Challenges were faced due to time and workforce constraints.



## Conclusion

The project effectively demonstrated positive results in terms of performance and scalability using DOTS and ECS. The strengths of DOTS were particularly observed in managing and moving large numbers of entities.

## Appendices

- GitHub Linki: <https://github.com/SinaErdem/CENG462Final>
- YouTube Video: [https://youtu.be/v2W-\\_U\\_V1t4](https://youtu.be/v2W-_U_V1t4)
- Project Link:  
<https://drive.google.com/file/d/16EjuxzdoiUrF7AVziNoWWPTbr1K7YGx/view?usp=sharing>