Enemy Prefab ReadMe Documentation

1. Overview

The Enemy Prefab represents all enemy types in the game, including movement, detection, chasing, and attack behavior. It uses several scripts that work together to make the enemy aware of the player, move toward them, and apply damage when appropriate. This document explains how the prefab works and how other developers can reuse or extend it.

2. Purpose of the Prefab

This prefab is designed to answer the question:
"How can we create enemies that automatically detect, chase, and damage the player without rewriting code for every scene or enemy type?"

The prefab allows designers and programmers to easily place enemies into any level with full functionality already built in.

3. Scripts Included in the Prefab

EnemyDamage.cs
Handles applying damage to the player on collision using a trigger collider.
(see file: /mnt/data/EnemyDamageScript.cs)

Playerawarness.cs
This is the main enemy AI script. It detects the player, chases them, flips the sprite when turning, and applies damage when the player is within attack range. It also includes the dynamic binding speed system.
(see file: /mnt/data/EnemyMoveScript.cs)

PlayerAwarenessController.cs
Provides distance checking, direction information, and awareness radius logic.
(see file: /mnt/data/PlayerAwarenessController.cs)

EnemyAttackToCharacterHurt.cs
Handles damage based on distance, using an attack radius and cooldown timer.
(see file: /mnt/data/EnemyAttackToCharacterHurt.cs)

These scripts together create a complete enemy system that works in any level.

4. Required Components on the Prefab

The following Unity components must be attached to the enemy game object:

- Rigidbody2D
- Collider (BoxCollider2D or CircleCollider2D)
- Trigger Collider for collision-based attacks
- Animator (if using animations)

Ensure the Player object has the “Player” tag for detection to work.

5. How to Use the Enemy Prefab

Adding to Scene:
Drag the prefab from the Prefabs/Enemies folder into any Unity scene.

Connecting to Player:
The enemy detects the player using the Player tag, so make sure the player object has this tag assigned.

Adjusting Stats:
Developers can change settings in the Inspector, such as:

- detection radius
- attack radius
- attack cooldown
- enemy speed
- damage amount
- player layer mask

These values allow the prefab to be reused for different enemy types.

6. Gizmos and Debugging

Several scripts draw Gizmos in the Scene view:

- Yellow circle shows detection radius
- Red circle shows attack radius

These help developers understand how far an enemy can see and attack.

7. Who Uses This Documentation

This ReadMe is meant for Unity developers, teammates working on new levels, and anyone maintaining or expanding the project. It answers the main setup questions and helps new users understand how to configure and reuse the enemy system.

Additional questions someone might ask after reading this include:

- How do I control the enemy’s movement speed?
- How do I add new attacks or abilities?
- How do I create a new enemy using the same scripts?
- How can I make enemies spawn during gameplay?
- How do I connect animations to attacks?

8. Version Information

Game Version: 6000.1.17f1
Prefab Version: Enemy Prefab v1.0
Author: Anjal Karki