# CA2

# Game Manager

The GameManager is responsible for tracking how many zombies are still present in the level. If there are no zombies in the level the manager will load the game complete scene.

## Exercise

* Add a variable to track the number of zombies in the level when the game starts
* The Awake method should be used to count the number of zombies
  + All zombies are already tagged as Zombie
* Add a method called RecordZombieDeath
  + When this method is called you need to subtract one from the variable that has the total number of zombies in the level
  + If the number of zombies remaining in the level is zero or less
    - Load the scene called GameComp lete

# Health Component

Both the Player and Enemies require health tracking. The health component will act as a parent class that can be inherited from later.

## Exercise

* Add a variable to store the current health value
* Add a virtual method called KillCharacter
  + This has no input parameters
  + Child classes will override this method
* Add a method called ApplyDamage. This method accepts a single integer argument called amount.
  + When this method is called, subtract the amount from the health
  + If the health is less than or equal to zero
    - Call the KillCharacter method

# Player Health

Player health is responsible for tracking the player health and handling when the player runs out of health.

## Exercise

* Update the PlayerHealth class to inherit from HealthComponent
* Override the KillCharacter method
  + When called load the scene called GameOver
* Add the OnCollisionEnter2D method
  + If the player collides with an object tagged as Zombie then use the ApplyDamage method to subtract 10 health

# Zombie Health

Zombie health is responsible for tracking the player health and handling when the zombie runs out of health. ZombieHealth performs two additional tasks:

* Instantiating a Splatter prefab when a zombie dies
* Communicating with the GameManager when a zombie dies

## Exercise

* Update the ZombieHealth class to inherit from HealthComponent
* Add a variable for the GameManager
* Add a variable for a GameObject to store the splatter prefab
* When this script starts you need to find the GameObject with the GameManager script attached
  + The GameObject is tagged as GameController
  + You need to get the GameManger from this GameObject
* Override the KillCharacter method
  + Call RecordZombieDeath on the GameManager
  + Instantiate the Splatter prefab at the position of the zombie
  + Destroy the zombie

# Bullet

The Bullet script will be updated to apply damage to a health component when the bullet collides with an object that has a health component.

## Exercise

* Add a variable called Damage (default of 10)
  + This variable will store the amount of damage the bullet will do on collision
* Add a method called ApplyDamageToHealth. This method will accept a single GameObject as a parameter
  + Try to get a health component from the game object we collided with
  + If the game object had a health component then call the ApplyDamage method and use the Damage variable as input.
* Update the OnTriggerEnter and OnCollisionEnter methods to call ApplyDamageToHealth
  + Use collision.gameObject as the input

# Zombie MOvement

This script will make each zombie rotate to face the player. When the player comes within a tracking range the zombie will begin to move towards the player.

## Exercise

* Add a GameObject variable that will hold a reference to the player
* Add a variable for the movementSpeed (default of 100)
* Add a variable for the trackingRange (default of 5)
* Add a variable to hold the RigidBody of the zombie
* On Start
  + Find the game object with the tag of Player. Store the result in the player variable declated earlier
  + Get the RigidBody of the zombie and store it in the variable declared earlier
* In the Update
  + Set the forward direction of the Zombie to always face the player
  + If the distance between the zombie and player is less than the tracking range (see code below)
    - Move the zombie in its forward direction



# Roof Toggle

This script will hide the roof when the player enters a building and show the roof when the player exits.

## Exercise

* Add an OnTriggerEnter method
  + If the roof collides with an object tagged as Player or Vehicle then disable the SpriteRenderer of the roof
* Add an OnTriggerExit method
  + If the roof stops colliding with an object tagged as Player or Vehicle then enable the SpriteRenderer of the roof

# Vehicle Collision

This script will ensure that a zombie immediately dies when hit by a vehicle.

## Exercise

* Add an OnCollisionEnter method
  + If the vehicle collides with an object tagged as Zombie then Get the health component of the zombie and call the KillCharacter method

# Pick Random Sprite

This script will pick a random sprite for a vehicle from an array.

## Exercise

* Add a method called PickRandomSprite
  + Pick a random sprite from the possibleSprites array
  + Get the SpriteRender component of the vehicle and its sprite property to be the randomly picked sprite
  + Call this method at the Start