HealthController

Set Health variable and add a StartHealth  
StartHealth= Health;  
put this in you Start Function

Because now if the Player dies you can respawn him and give him the Starthealth so he will be after the respawn at Max Health and you don’t need to fill the health up manually l just pass the variable Health = StartHealth and he is full again and can play…

Set the bool variable to true

If the variable is true player can get damage   
call Damage(dmg) function  
after he get damage set the variable to false so he can’t get multiple damage (like 4 arrows hit him and he can’t dodge so he gets damage from all 4 )

Use Invoke(“Functionname”,Seconds you want to wait); [unity will still continue with the code!!!!]  
in the Functionname you set the variable again to true so he can get damage.

You can do the same with the spawn system…

If you want to implement a DOT Field (Damage over time) system   
Than you need a Unity Function called

Void OnTriggerStay2D(Collider2D other(variablename))  
As long the player is inside he will get damage(let’s say it’s fire Field that burns over time as long as he is inside this field)   
this could look like this

float dmg = (Time.deltaTime / 4);

if (other.gameObject.tag == "Fire")

Dying(dmg);

If the Player dies you can enable the controller by   
 playerController.enabled = false;

You can get the playerController when you search in the parent object   
 playerController =transform.parent.gameObject.GetComponent<PlayerController>();

If you use a DOT like a poisen that hits the player and makes constant damage for 5 Seconds.  
you need to set a bool variable (let’s call it dot)

Go to the FixedUpdate()   
all you need here is

float dmg = (Time.deltaTime / 8); How much dmg?

if (dot) and as long as dot is true call the function

Damage(dmg);

Be carefull you need a 2nd script for this where you call in your HealthController the Function DamageOverTime(bool dot) and give it a true variable

All you need to do now is to negate the bool variable and set it to the dot then Invoke the Cooldown function  
 public void damageOverTime(bool variableName)

{

variableName = ! variableName;

this.dot = variableName;

Invoke(MyConst.Cooldown, 5);

}

In the Cooldown Function you set the Dot back to false so the player doesn’t get any further damage.