

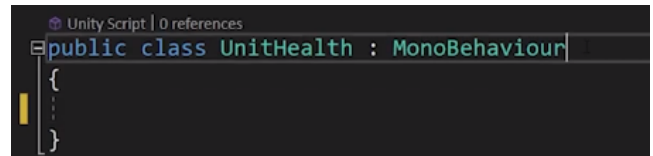
Tutorial 2 - Health - Jagoda Kopec

For my next tutorial, I'm going to try and figure out how to make a health system in unity, something I haven't done before. I have found a video which might prove to be useful.

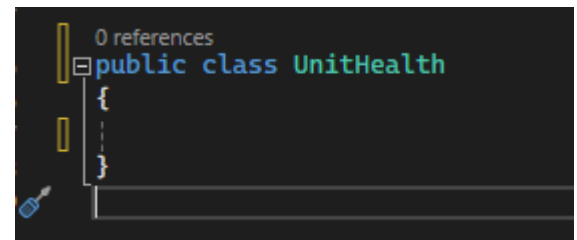
<https://www.youtube.com/watch?v=9i0UGVUKiaE>

As I am watching the tutorial and listening to what the person in the video is saying, I'm going to start and create another Script in my "Scripts" folder in Unity and call it "UnitHealth". Then I will double click on it to open Visual studio and get started.

The person in the video said to not make this a MonoBehaviour script as we won't need anything to do with the start and update methods. So, to make this a non MonoBehaviour script, I just delete the start and update voids and the word MonoBehaviour that comes after UnitHealth.



```
0 references
public class UnitHealth : MonoBehaviour
{
    ...
}
```

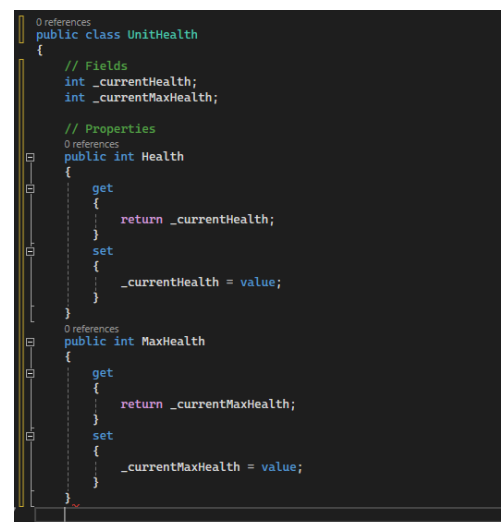


```
0 references
public class UnitHealth
{
    ...
}
```

I will then create an int which is gonna be called `_currentHealth`; . Apparently whenever you create fields inside a class it should always come with an underscore and start with a non capitalised letter! Something to remember for later.

After I've done that I will write underneath it "int `_currentMaxHealth`;" which I assume is there to represent my character's max health.

Continuing on, I will create Properties for the Fields I have created.



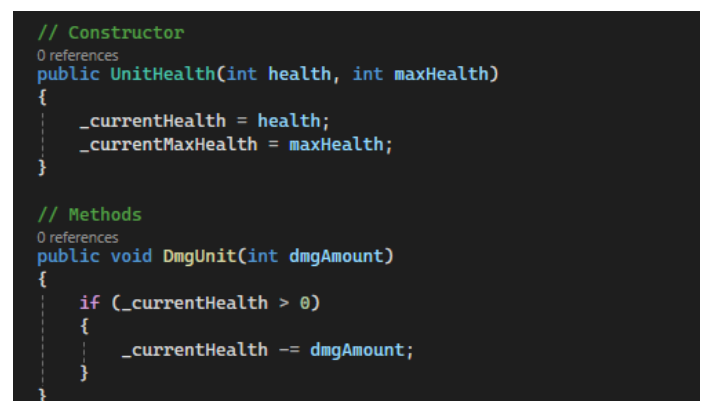
```
0 references
public class UnitHealth
{
    // Fields
    int _currentHealth;
    int _currentMaxHealth;

    // Properties
    0 references
    public int Health
    {
        get
        {
            return _currentHealth;
        }
        set
        {
            _currentHealth = value;
        }
    }

    0 references
    public int MaxHealth
    {
        get
        {
            return _currentMaxHealth;
        }
        set
        {
            _currentMaxHealth = value;
        }
    }
}
```

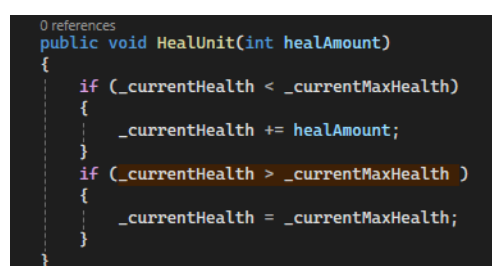
Next, I have made a constructor, which will let me create methods that will actually do things to my players health. For example, my player will be able to take damage or receive health!

I have created two methods, one figuring out that if my player were to take damage, it would have to subtract that number from the player's health, and the second one, doing the opposite. So that if we gain health it would add it to the current health of the player, unless it would be more than player's max health, then it would stop at max health.



```
// Constructor
0 references
public UnitHealth(int health, int maxHealth)
{
    _currentHealth = health;
    _currentMaxHealth = maxHealth;
}

// Methods
0 references
public void DmgUnit(int dmgAmount)
{
    if (_currentHealth > 0)
    {
        _currentHealth -= dmgAmount;
    }
}
```



```
0 references
public void HealUnit(int healAmount)
{
    if (_currentHealth < _currentMaxHealth)
    {
        _currentHealth += healAmount;
    }
    if (_currentHealth > _currentMaxHealth )
    {
        _currentHealth = _currentMaxHealth;
    }
}
```

After I have done all that, I will go back to unity and create a new empty game object in the hierarchy called GameManager, and go to my scripts folder and create a new script called the same thing. After all that is done I will click and drag my GameManager script onto the GameManager object in the hierarchy.

I am going to make this script a singleton, meaning that I will be able to access this script anywhere in any code by just referencing it!

This is the finished health system for the player.

This has set the player's max and starting health to a 100
(public UnitHealth
_playerHealth = new
UnitHealth(100, 100);

```

Unity Script | 1 reference
public class GameManager : MonoBehaviour
{
    3 references
    public static GameManager gameManager { get; private set; }

    public UnitHealth _playerHealth = new UnitHealth(100, 100);

    Unity Message | 0 references
    void Awake()
    {
        if (gameManager != null && gameManager != this)
        {
            Destroy(this);
        }
        else
        {
            gameManager = this;
        }
    }
}

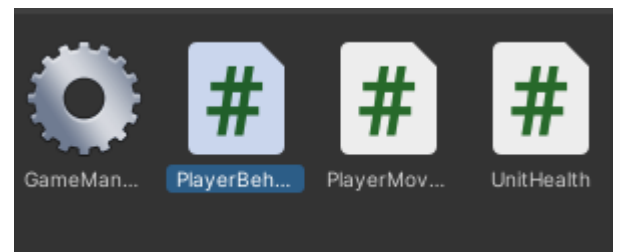
```

Next step is to create a new script called PlayerBehaviour. In this script I will create methods for the player taking damage and healing.

This is where I might have to figure something out a bit on my own as in the video the person shows us how to code the player to take damage when he presses Space and heal when he presses shift. I want something a bit different. I want to take damage when touching spikes and healing when picking up an item or coming up to a well that's on my little map.

I have the health system prepared, and all I have to do, and hopefully it won't be too hard, is to figure out how to make my player take damage from spikes. For now all I know is that it's going to have to be in the Void Update section of the code.

I have made a system where whenever I press space the player takes damage and whenever I press LeftShift I heal.



```

// Start is called before the first frame update
Unity Message | 0 references
void Start()
{
}

// Update is called once per frame
Unity Message | 0 references
void Update()
{
}

0 references
private void PlayerTakeDmg(int dmg)
{
    GameManager.gameManager._playerHealth.DmgUnit(dmg);
}

0 references
private void PlayerHeal(int healing)
{
    GameManager.gameManager._playerHealth.HealUnit(healing);
}
}

```

```
// Update is called once per frame
@ Unity Message | 0 references
void Update()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        PlayerTakeDmg(20);
        Debug.Log(GameManager.gameManager._playerHealth.Health);
    }
    if (Input.GetKeyDown(KeyCode.LeftShift))
    {
        PlayerHeal(10);
        Debug.Log(GameManager.gameManager._playerHealth.Health);
    }
}
```

```
[22:12:23] 80
UnityEngine.Debug:Log (object)
[22:12:25] 60
UnityEngine.Debug:Log (object)
[22:12:26] 70
UnityEngine.Debug:Log (object)
[22:12:27] 80
UnityEngine.Debug:Log (object)
[22:12:27] 90
UnityEngine.Debug:Log (object)
[22:12:28] 100
UnityEngine.Debug:Log (object)
```