

# 1. extension methods

```
using UnityEngine;
using System.Collections;

public class Player : MonoBehaviour
{
    void Update ()
    {
        // Set new x position to 5
        transform.SetPositionX(5f);
    }
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

public class Player : MonoBehaviour
{
    void Update ()
    {
        // Set new x position to 5
        transform.SetPositionX(5f);
    }
}
```

wie kent deze  
functie

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....


.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

public class Player : MonoBehaviour
{
    void Update ()
    {
        if(gameObject.HasRigidbody()) {
            // ...
        }
    }
}
```

of deze functie



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....


.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

public class Player : MonoBehaviour
{
    void Update ()
    {
        transform.doALooping("now");
    }
}
```

of deze functie



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

Ooit standaard methods toe willen voegen aan bestaande Classes?

- Bijvoorbeeld aan:
- GameObjects
- Transforms
- Strings
- Etc...

# 1. extension methods

Ooit standaard methods toe willen voegen aan bestaande Classes?

Bijvoorbeeld aan:  
GameObjects  
Transforms  
Strings  
Etc...

dan kan dat met  
extension methods

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

public class Player : MonoBehaviour
{
    void Update ()
    {
        // Set new x position to 5
        transform.SetPositionX(5f);
    }
}
```

met extension methods kan ik bijvoorbeeld elke transform een method geven

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

public class Player : MonoBehaviour
{
    void Update ()
    {
        // Set new x position to 5
        transform.SetPositionX(5f);
    }
}
```

deze method zet  
alleen de x positie

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



# 1. extension methods

```
using UnityEngine;
using System.Collections;

public class Player : MonoBehaviour
{
    void Update ()
    {
        // Set new x position to 5
        transform.SetPositionX(5f);
    }
}
```

Je zou verwachten dat we de Class uitbreiden

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

//Must be in a static class
public static class Extensions
{
    //Function must be static
    //First parameter has "this" in front of type
    public static void SetPositionX(this Transform t, float newX)
    {
        t.position = new Vector3(newX, t.position.y, t.position.z);
    }
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

//Must be in a static class
public static class Extensions
{
    //Function must be static
    //First parameter has "this" in front of type
    public static void SetPositionX(this Transform t, float newX)
    {
        t.position = new Vector3(newX, t.position.y, t.position.z);
    }
}
```

een extension method class  
kan elke naam hebben

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

//Must be in a static class
public static class Extensions
{
    //Function must be static
    //First parameter has to be the type you want to extend
    public static void SetPositionX(this Transform t, float newX)
    {
        t.position = new Vector3(newX, t.position.y, t.position.z);
    }
}
```

maar de methods  
moeten wel static zijn

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

//Must be in a static class
public static class Extensions
{
    //Function must be static
    //First parameter has "this" in front of type
    public static void SetPositionX(this Transform t, float newX)
    {
        t.position = new Vector3(newX, t.position.y, t.position.z);
    }
}
```

dit dwingen we af door de class ook static te maken

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

//Must be in a static class
public static class Extensions
{
    //Function must be static
    //First parameter has "this" in front of type
    public static void SetPositionX(this Transform t, float newX)
    {
        t.position = new Vector3(newX, t.position.y, t.position.z);
    }
}
```

hiermee bepalen  
we voor wie het is

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

//Must be in a static class
public static class Extensions
{
    //Function must be static
    //First parameter has "this" in front of type
    public static void SetPositionX(this Transform t, float newX)
    {
        t.position = new Vector3(newX, t.position.y, t.position.z);
    }
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

//Must be in a static class
public static class Extensions
{
    //Function must be static
    //First parameter has "this" in front of type
    public static void SetPositionX(this Transform t, float newX)
    {
        t.position = new Vector3(newX, t.position.y, t.position.z);
    }
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



# 1. extension methods

```
using UnityEngine;
using System.Collections;

public static class Extensions
{
    public static bool HasRigidbody(this GameObject gobj)
    {
        return (gobj.rigidbody != null);
    }

    public static bool HasAnimation(this GameObject gobj)
    {
        return (gobj.animation != null);
    }
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

```
using UnityEngine;
using System.Collections;

public static class Extensions
{
    public static bool HasRigidbody(this GameObject gobj)
    {
        return (gobj.rigidbody != null);
    }

    public static bool HasAnimation(this GameObject gobj)
    {
        return (gobj.animation != null);
    }
}
```

je mag er gewoon meer  
per Class aanmaken



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

An extension method, is a syntactical shortcut which gives the appearance of "extending" a type by "adding" a method to it

# 1. extension methods

....syntactic sugar  
Gewoon een static method die wordt aangeroepen

# 1. extension methods

Zo kunnen we elke GameObject een GetInterface geven..

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

.....  
.....  
.....  
.....

# 1. extension methods



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods



Player gebruikt GetComponent voor skateboard

.....

.....

.....

.....

.....

.....

.....

.....

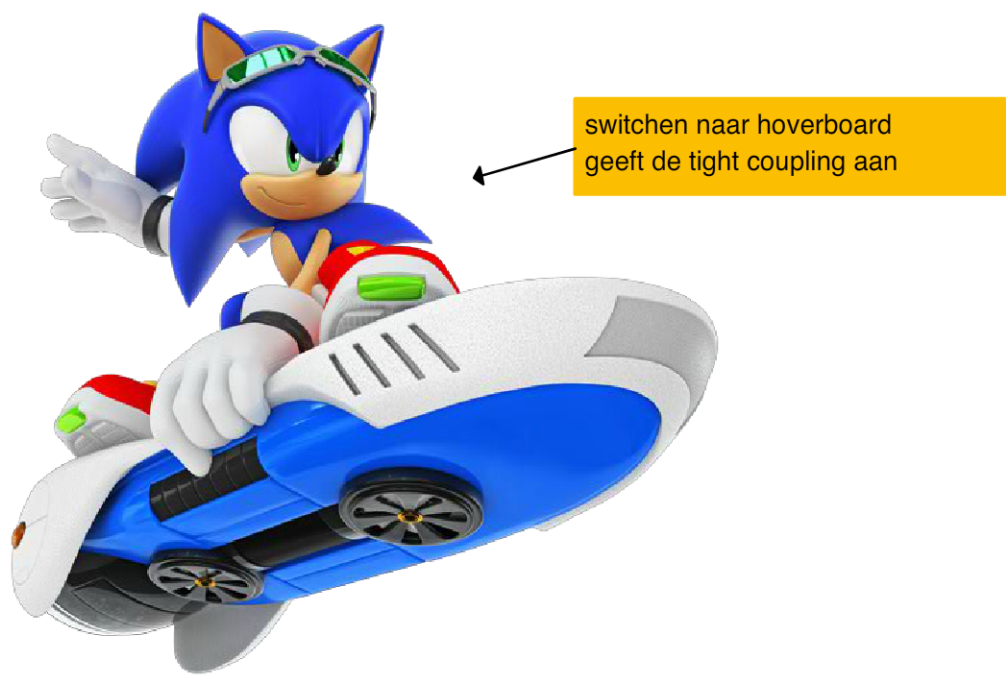
.....

.....

.....

.....

# 1. extension methods



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



# 1. extension methods



GetInterface is een veel lossere koppeling naar een component

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# 1. extension methods

Voorbeeld die jullie mogen gaan testen

Maak:

- een Player component
- skateboard component
- hoverboard component

En zorg ervoor dat ze 'loosely coupled' zijn

Er staat een voorbeeld met wapens in mijn repository