

```

public static Vector2 xy(this Vector3 v, Vector2 other) { v.x = (float)other.x();v.y = (float)other.y(); return v.
0 references
public static Vector2 yy(this Vector3 v) { return new Vector2(v.y, v.y); }
1 reference
public static Vector2 zy(this Vector3 v) { return new Vector2(v.z, v.y); }
0 references
public static Vector2 zy(this Vector3 v, Vector2 other) { v.z = (float)other.x();v.y = (float)other.y(); return v.
1 reference
public static Vector2 xz(this Vector3 v) { return new Vector2(v.x, v.z); }
0 references
public static Vector2 xz(this Vector3 v, Vector2 other) { v.x = (float)other.x();v.z = (float)other.y(); return v.
1 reference
public static Vector2 yz(this Vector3 v) { return new Vector2(v.y, v.z); }
0 references
public static Vector2 yz(this Vector3 v, Vector2 other) { v.y = (float)other.x();v.z = (float)other.y(); return v.
0 references
public static Vector2 zz(this Vector3 v) { return new Vector2(v.z, v.z); }

```

```

// Vector3 with 3 components.

```

```

0 references
public static Vector3 xxx(this Vector3 v) { return new Vector3(v.x, v.x, v.x); }
0 references
public static Vector3 yxx(this Vector3 v) { return new Vector3(v.y, v.x, v.x); }
0 references
public static Vector3 xyy(this Vector3 v) { return new Vector3(v.x, v.y, v.y); }

```

```

0 references
public static Vector3 yyy(this Vector3 v) { return new Vector3(v.y, v.y, v.y); }

```

```

0 references
public static Vector3 xxy(this Vector3 v) { return new Vector3(v.x, v.y, v.x); }

```

```

0 references
public static Vector3 yxy(this Vector3 v) { return new Vector3(v.y, v.y, v.x); }

```

```

0 references
public static Vector3 yxz(this Vector3 v, Vector3 other) { v.z = (float)other.x();v.y = (float)other.y();v.x = (fl
0 references
public static Vector3 xzx(this Vector3 v) { return new Vector3(v.x, v.z, v.x); }

```

```

0 references
public static Vector3 yzx(this Vector3 v) { return new Vector3(v.y, v.z, v.x); }

```

Vector Swizzle

Vector Swizzle – Documentation

What is Vector swizzle ?

Vector swizzle is a set of C# extensions for Unity which aims to provide some syntactic sugars to use Unity Vectors.

It's a feature we often find in shader languages called "Swizzle" too.

You can find more info in the GLSL documentation here:

[https://www.khronos.org/opengl/wiki/Data_Type_\(GLSL\)#Swizzling](https://www.khronos.org/opengl/wiki/Data_Type_(GLSL)#Swizzling)

The Vector swizzle plugin contains the following:

- Added "Vector4Int" type for coherence.
- xyzw / rgba / stpq masks without mixing symbols from different masks.
- Implemented the same way as the GLSL implementation.
- Allows for both get and set.
- Each "unit" field also wrapped in an extension method for typing consistence (v.x == v.x())
- Works on Vector2, Vector3, Vector4, Vector2Int, Vector3Int, Vector4Int, Color & Color32

Vector Swizzle – Documentation

How to use it?

Vector Swizzle is a set of extension methods so the first step to use it is simply to add the following code in the top of your C# script.

```
using VectorSwizzle;
```

Once done, you will immediately get access to every function the extension provides.

Features

- Get and Set extensions methods
- Compliant with GLSL Swizzle implementation
- 3 masks (xyzw, rgba, stpq)
- Vector4Int added

```
gameObject.transform.position.xy(playerTransform.position.xy());  
  
// Is equivalent to  
gameObject.transform.position = new Vector3(playerTransform.position.x,  
                                             playerTransform.position.y,  
                                             gameObject.transform.position.z);
```

For more code samples you can check the “Assets/Plugins/VectorSwizzle/Samples” folder which contain more code samples.

Vector Swizzle – Documentation

Support

In case of problem or issue, don't hesitate to reach through this email:
smairedev.support@outlook.com.