Direction Indicator Manual

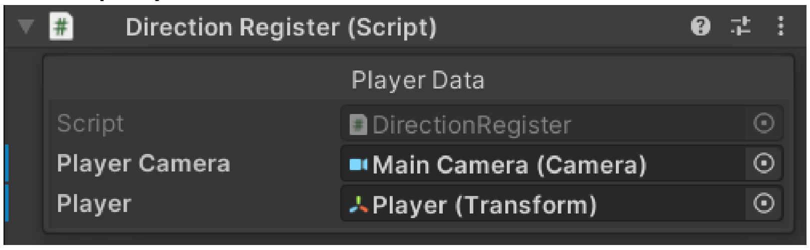
Direction Indicator - A simple and versatile system that can be used to show direction to a target and can be applied to both 2D and 3D projects. The most important advantage is the ability to quickly implement this system in your project.

Quick setup

1. Open folder: Direction Indicator/Prefabs/… .
2. Place "Direction Register" prefab on the scene.
3. In script *DirectionRegister* add to the Player Camera field the camera that is the main player camera and in the Player field of the player.
4. Add a "Direction Register" call to your script (For example, you can place the Waypoint script on an object in your scene).
5. Press Play.

How to use

Place "Direction Register" prefab on the scene. Add to the Player Camera field the camera that is the main player camera and in the Player field of the player.

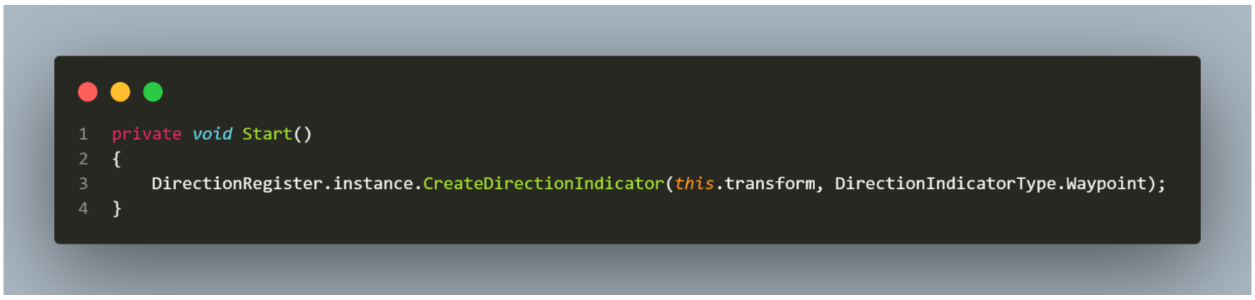


Direction Register (Script)

Player Data

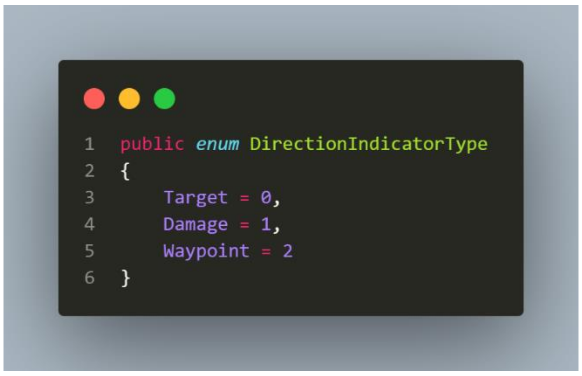
Script	DirectionRegister
Player Camera	Main Camera (Camera)
Player	Player (Transform)

For the indicator to appear, you need to call the *CreateDirectionIndicator* method.



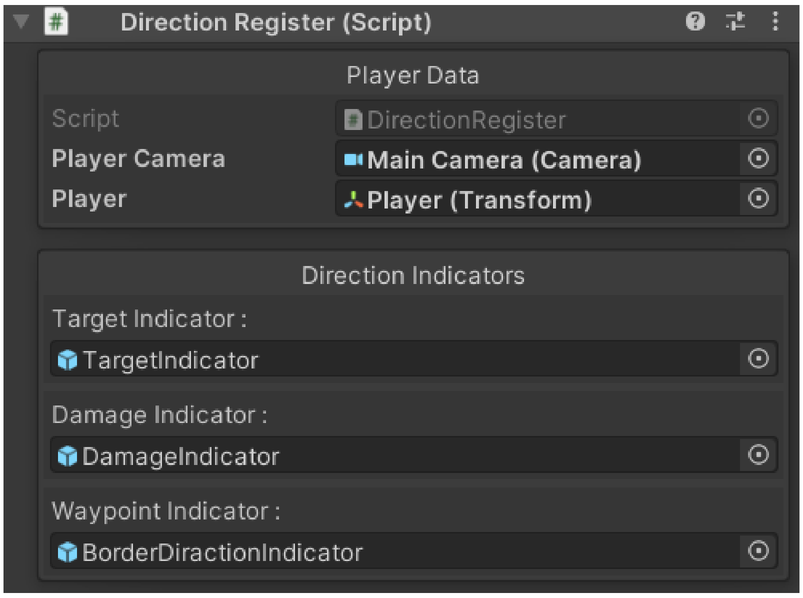
```
private void Start()
{
    DirectionRegister.instance.CreateDirectionIndicator(this.transform, DirectionIndicatorType.Waypoint);
}
```

You can create your own indicators very simply by adding your own variable to *DirectionIndicatorType*. But it is necessary to maintain indexing, each subsequent value must be 1 greater than the previous one.



```
public enum DirectionIndicatorType
{
    Target = 0,
    Damage = 1,
    Waypoint = 2
}
```

You need to create your indicator based on the *DirectionIndicator* class and create a prefab with it. Next, you need to add a prefab for the new *DirectionIndicatorType* in *Direction Register* prefab.



Direction Register (Script)

Player Data

Script	DirectionRegister
Player Camera	Main Camera (Camera)
Player	Player (Transform)

Direction Indicators

Target Indicator :
TargetIndicator

Damage Indicator :
DamageIndicator

Waypoint Indicator :
BorderDiractionIndicator

Congratulations, you have added your direction indicator.

Script Reference

public class DirectionRegister : MonoBehaviour;

Class for indicators operation.

private *Camera* _playerCamera;

The camera that is or is watching the player.

public DirectionIndicator CreateDirectionIndicator();

Creates an indicator for showing a target.

public *void* ClearIndicators()

Delete all created indicators.

---

public abstract *class* DirectionIndicator : MonoBehaviour;

The class implements the main parameters of the indicator.

public *Action<bool>* ShowDirectionIndicator;

Called when the indicator is about to disappear or reappear.
When isShow is true, the indicator appears.

public *Action* DestroyDirectionIndicator;

Called when an object is deleted.

public *Transform* TargetTransform;

The object we are monitoring.

public *Transform* PlayerTransform;

The player from whom the indicator rotation is calculated.

public *Camera* PlayerCamera;

The camera that is or is watching the player.

public virtual *void* InitIndicator;

Initializes the data required for the indicator to work.

ALWAYS CALL WHEN INSTANTIATE INDICATOR!

---

public class BorderDirectionIndicator : DirectionIndicator;

Script that moves the arrow along the border of the screen.

private *float* _arrowOffset;

Offset arrow from screen border.

private bool _isScale;
Arrow size depending on the player's distance from the target.

private *bool* _isHideWhenLooking;

Hide the arrow when looking at the target.

---

public *class* CircleDirectionIndicator : *DirectionIndicator*

Rotates the arrow 360 degrees depending on the target.

---

public *class* PointDirectionIndicator : *DirectionIndicator*

To the position of the target on the screen draws on top of the indicator that points to the target.

---

public class DamageDirectionIndicator : CircleDirectionIndicator

Deleted after some time.

private *float* _timeToDestroy;

Time after which the indicator is removed.

private *float* _duration;

Camera shake duration.

private *float* _magnitude;

Camera shake force.

private *bool* isShakeCamera;

If True then the camera will shake.

---

public *class* TextDistance : *MonoBehaviour*

Shows the distance from the player to the target.

private *string* _unit;

Number unit.

private *float* _factor;

Range adjustment distance.

---

public *enum* DirectionIndicatorType

The indicator type that will be created when called.

Support

Check the Publisher Page for contact information.