

---

## *Compass Bar Pro*

-

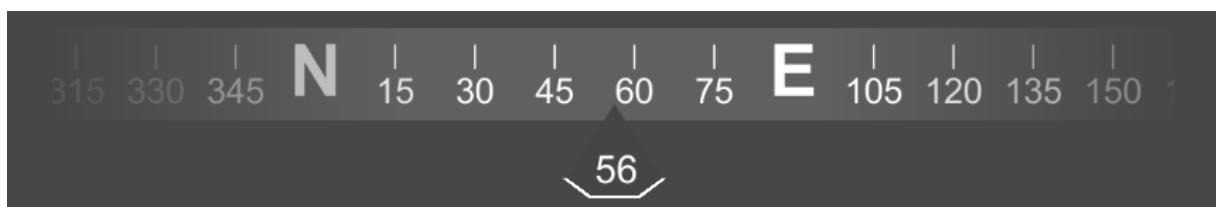
## *Lunar Cats Studio*

---

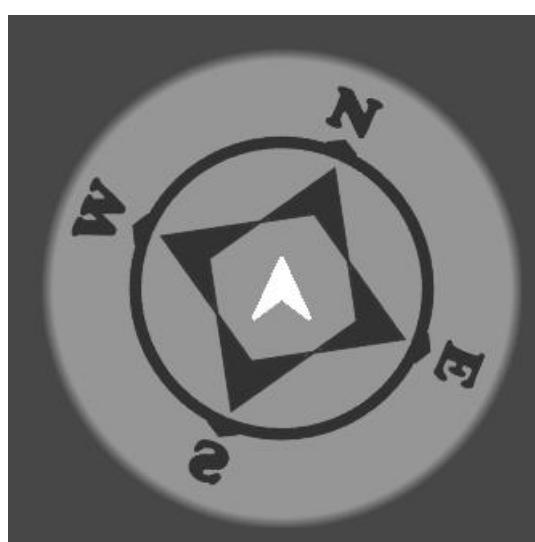
### **1. Presentation**

This asset provides two type of compass,

**Linear Compass :**



**And Circular Compass :**



Compass prefabs are located in « **Assets\CompassBarPro\Prefab** » folder.

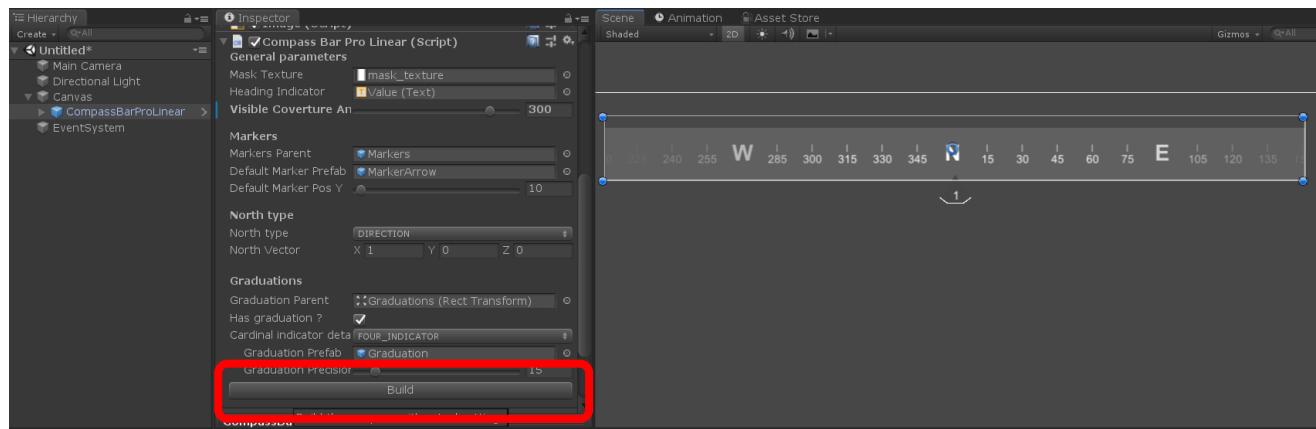
## 2. How to use

### 2.1. Linear Compass

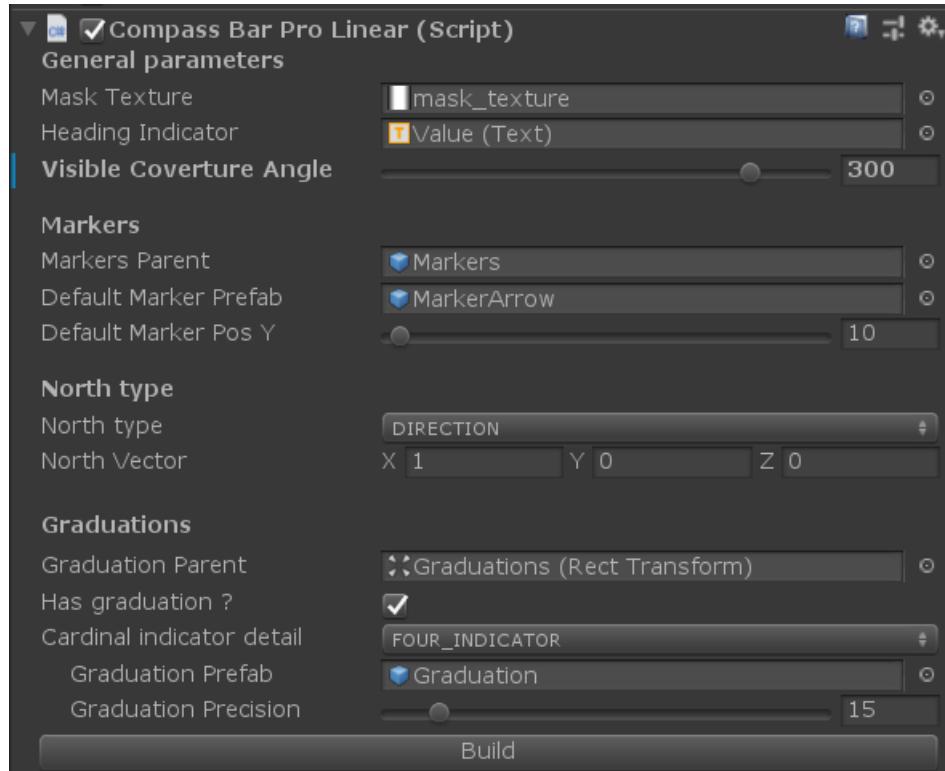
#### 2.1.1. Import LinearCompass in your project

Drag and drop the « **CompassBarProLinear** » prefab into your scene under a Canvas component. Set up position and size of the compass and then click « **Build** » button under **CompassBarProLinear** component in the GameObject. This will build the graduations in the compass, based on the parameter values (coverture angle, graduation precision, cardinals indicator).

You can then adjust the parameters of the script to fit your needs. If you resize the LinearCompass, you will need to click « **Build** » button again so that compass graduations adjust to the new size.



#### 2.1.2. LinearCompass settings



- **Mask Texture** : This is the mask texture that will be applied to the compass in order to handle the transparency fade effect. Black pixels will be transparent and white pixels will be opaque

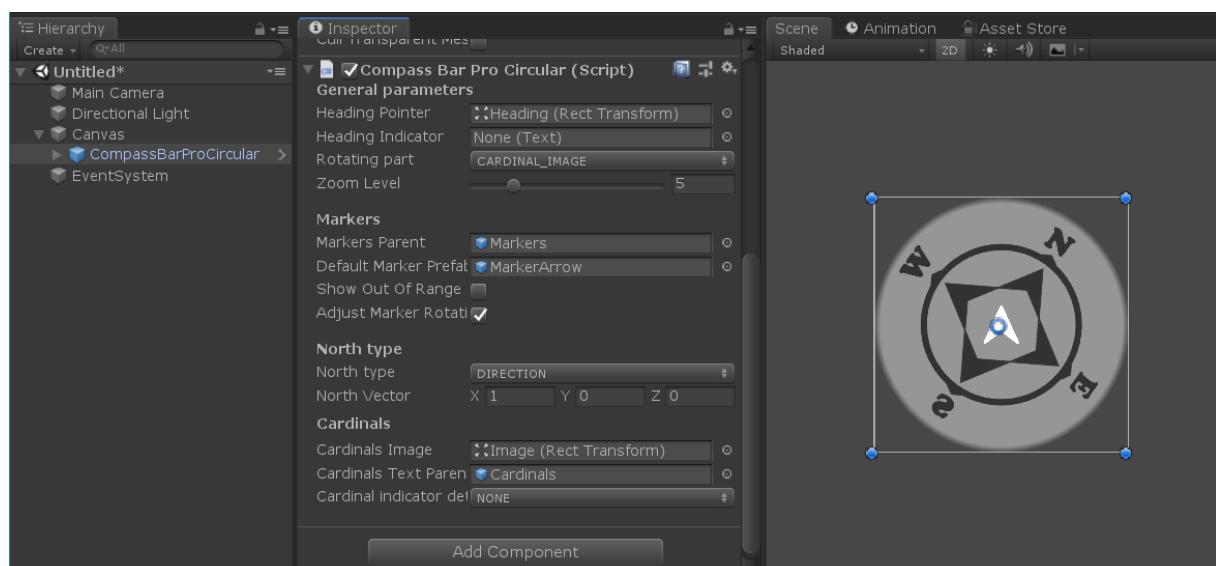
- **Heading Indicator** : (Optional) The prefab of the heading value indicator
- **Visible Coverage Angle** : The total angle coverage of the compass in degree. This value must be between 1 and 360
- **Markers Parent** : Reference to the parent GameObject of markers
- **Default marker prefab** : The default marker prefab that will be used when adding a marker if no prefab specified while using API
- **Default marker Pos Y** : The default marker's Y position in compass if no specified while using API. Y position is relative to the « Markers » GameObject under Compass.
- **North Type** : Choose between a Direction (Vector3) or a GameObject's transform in the scene to represent the North
- **North Vector** : The absolute vector pointing to the north in case of North Type set to Direction
- **North Transform** : The transform representing the north in case of North Type set to Transform
- **Graduation Parent** : Reference to the parent GameObject of graduations
- **Has Graduation** : Set it to true to display graduations in the compass
- **Cardinal indicator detail** : Choose between None (no cardinal indicator), Four Indicators (N, S, E, W cardinals) or Eight Indicators (N, NE, E, SE, S, SW, W, NW)
- **Graduation Prefab** : The prefab used for graduations in the compass
- **Graduation precision** : Determines the interval in degree between each graduation

**Build button** : Click build button after having resized the compass in order for graduations to adjust size and position, or if you changed the graduation prefab.

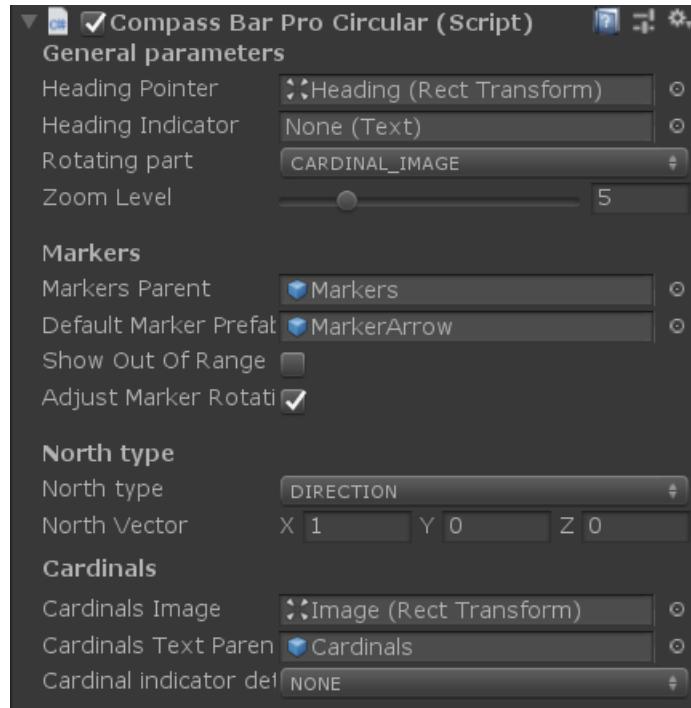
## 2.2.Circular Compass

### 2.2.1. Import CompassBarPro in your project

Drag and drop one of the « **CompassBarProCircular** » prefab into your scene under a Canvas component. Set up position and size of the compass and then adjust the parameters of the script to fit your needs.



## 2.2.2. Circular Compass settings



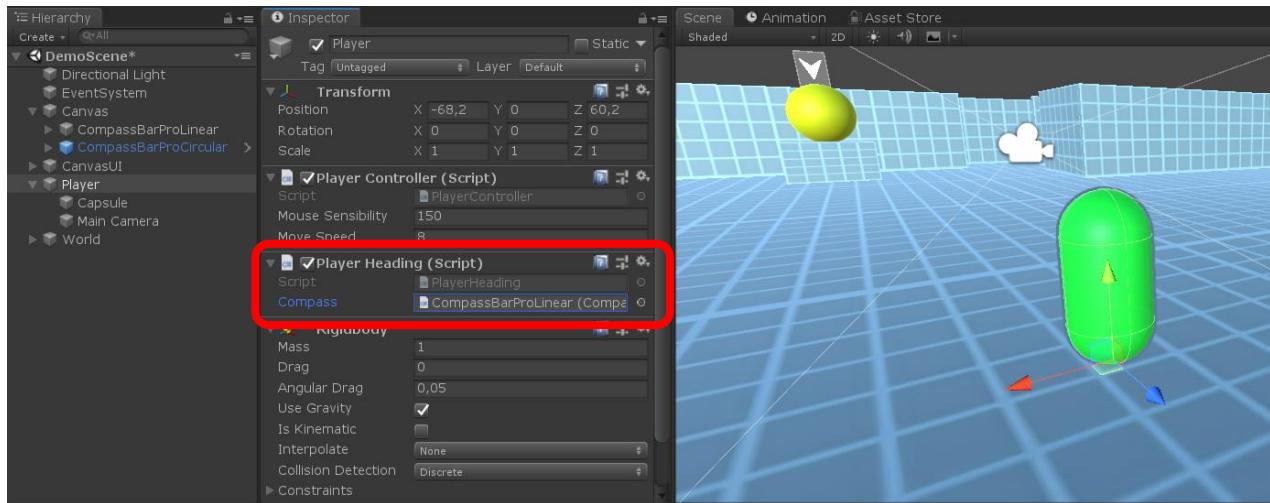
- **Heading Pointer** : Reference to the needle GameObject of the compass, indicating the player's heading
- **Heading Indicator** : (Optional) The prefab of the heading value indicator
- **Rotating part** : Indicates which part of the compass will rotate with the player's heading : the Cardinal Image or the needle pointer.
- **Zoom level** : The current zoom level of the compass, this will affect the markers position relative to the player in the compass
- **Markers Parent** : Reference to the parent GameObject of markers
- **Default marker prefab** : The default marker prefab that will be used when adding a marker if no prefab specified while using API
- **Show out of range** : Set to true if you want markers to still be visible in the compass even if their position (given the current zoom level) is out of the compass range, they will be shown in the border of the compass. If set to false, far away markers will not be visible in compass
- **Adjust marker rotation** : Set to true if you want the markers to always keep their absolute orientation, even when the compass rotates
- **North Type** : Choose between a Direction (Vector3) or a GameObject's transform in the scene to represent the North
- **North Vector** : The absolute vector pointing to the north in case of North Type set to Direction
- **North Transform** : The transform representing the north in case of North Type set to Transform
- **Graduation Parent** : Reference to the parent GameObject of graduations
- **Cardinals image** : (Optional) Reference to the compass's cardinal image
- **Cardinals Text parent** : Reference to the Parent GameObject of the cardinals Text GameObjects
- **Cardinal indicator detail** : Choose between None (no cardinal indicator), Four Indicators (N, S, E, W cardinals) or Eight Indicators (N, NE, E, SE, S, SW, W, NW)

### 2.3. Connect Compass Bar with a player

In order to connect the compass with your player, you just have to add the « **PlayerHeading** » script in your player's GameObject and set the compass reference.

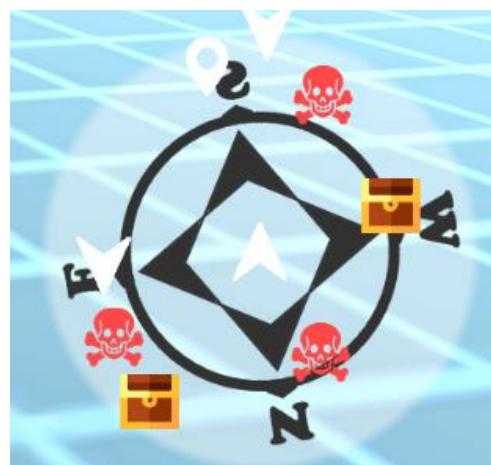
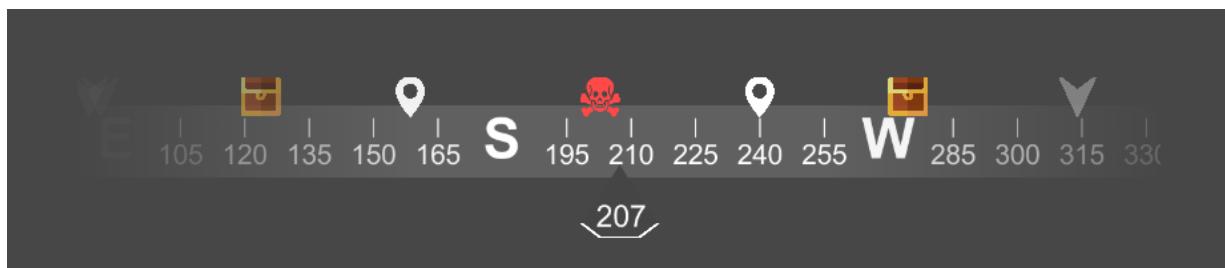
This script update the heading every Unity's Update, using the « **SetHeading** » API.

The heading is calculated with the forward vector of the player's transform and the north definition.



### 2.4. Markers

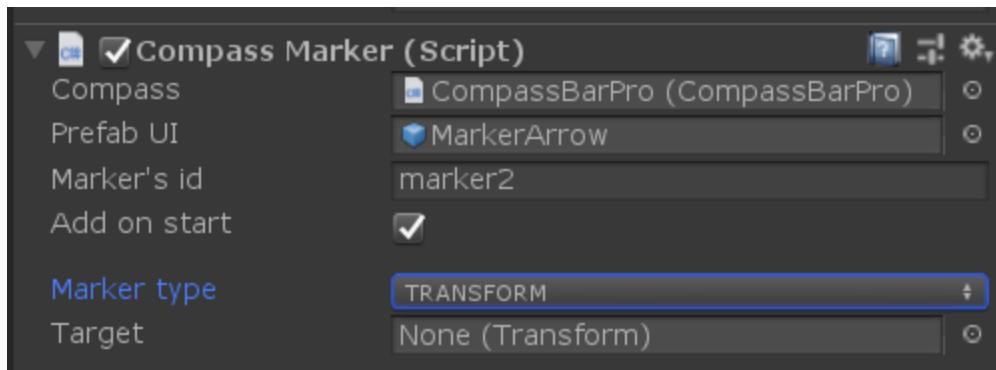
Markers are icons displayed in the Compass, they are usefull to indicate special locations, targets or enemies.



There are two types of markers :

- **Absolute Markers** : markers displayed at an absolute heading value, they are not linked to any GameObject in the scene.
- **Transform Markers** : markers linked to a GameObject in the scene. The heading value of the marker in the compass will depend on the player's position and on the GameObject position

In order to add a marker to the compass, you can use the « **CompassMarker** » script as follow :



- **Compass** : Reference to the compass component
- **Prefab UI** : The prefab of the marker that will be displayed in the compass
- **Marker's id** : The id of the marker. A simple string value unique to this marker used to reference it when using APIs (if you want to delete a previously added marker for example)
- **Add on start** : Set the true if you want this marker to be added to the compass on start. If set to false, simply call the « **AddMarker()** » API of this script whenever you want to add the marker to the compass (example are in demo scene)
- **Marker type** : Absolute marker or Transform (see explanation above)
- **Target** : If Marker type is Transform, the transform component linked to this marker. If null, the transform component of the GameObject where this script is attached to will be used
- **Heading** : If Marker type is Absolute, the absolute heading value of the marker

## How to create a new marker ?

In order to create a new marker, you need to import the associated image in Unity as a Sprite. Then duplicate one of the existing marker prefab and edit it to replace the UI Image. You can now use your new marker prefab !

### 2.5. North

The compass can handle two types of north : Direction or Transform.

- **North is Direction** : If north is set to Direction, an absolute 3D Vector will be taken into account for the north. North will always point at the same direction and will not depend on player's position.

- **North is Transform** : If north is set to Transform, the GameObject's position will be used as the true north. The direction of the north will depend on the player's position, relative to the North GameObject.

### 3. APIs

#### 3.1. Markers

```
/// <summary>
/// Remove all existing markers from compass
/// </summary>
public void RemoveMarkers()
```

```
/// <summary>
/// Remove the marker matching id
/// </summary>
/// <param name="id"></param>
public void RemoveMarker(string id)
```

```
/// <summary>
/// Add a marker
/// </summary>
/// <param name="marker"></param>
public void AddMarker(string id, Transform marker)
```

```
public void AddMarker(string id, Transform marker, GameObject prefab)
```

```
public void AddMarker(string id, Transform marker, GameObject prefab, int posY)
```

```
public void AddMarker(string id, float heading, int posY)
```

```
public void AddMarker(string id, float heading)
```

```
public void AddMarker(string id, GameObject prefab, float heading)
```

```
/// <summary>
/// Add a marker
/// </summary>
/// <param name="markerTransform"></param>
/// <param name="prefab"></param>
/// <param name="heading"></param>
/// <param name="posY"></param>
public void AddMarker(string id, Transform markerTransform, GameObject prefab,
float heading, int posY)
```

```
/// <summary>
/// Return the list of markers
```

```
/// </summary>
/// <returns></returns>
public List<CompassMarkerUI> GetMarkers()
```

### 3.2. North definition

```
/// <summary>
/// Set the north to a new arbitrary transform position
/// </summary>
/// <param name="northPosition"></param>
public void SetNorthPosition(Transform northPosition)
```

```
/// <summary>
/// Set the north to a new Vector3 direction
/// </summary>
/// <param name="northDirection"></param>
public void SetNorthDirection(Vector3 northDirection)
```

### 3.3.Heading

```
/// <summary>
/// Set the heading of the compass. The heading is calculated with the
/// player's Transform and with the selected north type
/// </summary>
/// <param name="playerOrientation"></param>
public void SetHeading(Transform playerOrientation)
/// <summary>
/// Set the heading of the compass, given the heading in parameter
/// </summary>
/// <param name="heading"></param>
public void SetHeading(float heading)
```

## **4. Contact And Support**

If you are facing any issue, please contact us, we will provide support.

As this Asset is in constant evolution, you can also contact me for any suggestion or for new feature you wish to see in this asset.

If you like this asset and want to support us, we'll be very pleased if you could leave a review in the AssetStore 😊.

Facebook: <https://www.facebook.com/LunarCatsStudio/>

Web Site: <http://lunarcatsstudio.com/>

E-mail: [lunar.cats.studio@gmail.com](mailto:lunar.cats.studio@gmail.com)