TargetMaker

Cursor Construction Pack

Created by Ripcord Development
info@ripcorddev.com

This project has been thoroughly tested for bugs before being sent to the Unity Asset store. In the event that you do find an issue with this package, please contact us before posting negative feedback on the Unity Asset Store. We are more than willing to help solve any issues you may encounter.

TargetMaker is a set of scripts, models and textures that when combined can quickly create unique assets for your project. Ideal for creating unique looking cursors and markers like those found in Real Time Strategy games..

TargetMaker is composed of a number of different parts.

Rings - These make up the majority of the prefabs in the package. Circular textures applied to flat polygons. Combined with the included scripts these rings can spin and pulse to add extra movement.
Centers - Typically used at the center of a ring, a center is another texture applied to a plane. They are meant to indicate the center of the ring.
Auras - Glowing objects around the ring. These are used to add extra visibility and add some 3D depth to the cursor.
Arrows - Extra effects that are generated to help guide the eye to where the click is taking place.
Markers - A marker is an object that is generated after a click. It stays in place for a limited amount of time before disappearing.

PRESETS

To help you get started quickly, **TargetMaker** comes with a number of pre-built cursors ready to drop into your project. Feel free to pull these presets apart to get a sense of how things can be combined in order to make a unique look that's right for you.

MODELS

These are the basic building blocks in **TargetMaker**. All textures in this package can be applied to any of these meshes in order to create a unique cursor that's right for your project.

TEXTURES

These are sorted into categories based on what they are best suited for. Try experimenting by placing them on other elements. You'd be surprised by some of the unique effects you can create.

All textures are supplied as high-res 1024x1024 PNG's with an alpha channel. All textures are white so that they can be easily tinted to whatever colour you want.

Note *Because of the way Unity displays transparent PNG's all textures will show up as a white square in the inspector. To get a quick preview of each texture please refer to the Preview folder in the asset package for low-resolution previews.*

MATERIALS / SHADERS

All materials in this package are set up to use the included "Unlit Alpha Tint" shader. This is an unlit shader that allows you to specify a tint colour as well as a transparency level.

Note *This shader is not necessary for **TargetMaker** to function.*

SCRIPTS

TargetMaker comes with two sets of scripts. The scripts intended for use with the actual package, and scripts that are necessary for the demo to run. Any script with the "**DEMO-**" tag at the front is **not** needed for **TargetMaker** to function.

Note ***TargetMaker*** *makes use of raycasting to detect and update the position of the target object. In order for raycasting to work, there must be a collider in your scene. The target object will only move above a collider as that's what's needed for raycasting to function. Please see the included targetMaker scene for an example.*

TargetMaker *is set up to reference the MainCamera (Camera.main). This is a camera with the "MainCamera" tag. More than likely this setup will work for your project. If you do not wish to use this setup however, the AttachToCursor script can be easily modified to reference any camera you specify.*

For additional info on all the scripts listed below, please check the comments in the actual script files. The functions and variables are thoroughly commented. If you have further questions, please don't hesitate to ask.

- **CursorManager** - This script contains all the rules for the cursor. Audio and effects related to the cursor are generated from this script. This script must be in your scene for **TargetMaker** to run properly. It should be put on an empty game object. The game object can be named whatever you want, and placed anywhere in the scene.
- **AttachToCursor** - Any object with this will essentially be attached to the mouse cursor. It will follow its movement exactly. This script is set up to follow the mouse cursor on only two axes at a time. It can be easily modified to work on all three axes at the same time, but this type of movement can get confusing for some and is generally not recommended.

    AxisPair suggestions:
    - **XY** - Vertical and horizontal movement such as a puzzle game or sidescroller.
    - **YZ** - Similar to XY, but sort of difficult to manipulate. You probably won't use this.
    - **XZ** - Horizontal movement such as a top down RTS game or anything moving along a ground plane
- **DetachDestroy** - This script detaches all child objects from the parent, then removes the parent object from the scene. This is useful for removing empty container objects to keep your scene clean
- **CursorMovement** - This script controls the basic movement of objects. It can control the rotation as well as modify the scale. This script can also make the object shrink away until it disappears. This can be based on a timer, or as soon as the object is generated
- **Shrink** - This script controls the shrinking of objects based on values in the CursorManager. When an object is below a specified size, it is removed from the scene.
- **Spinner** - A simple script that rotates the object on its Y axis. Positive values rotate it clockwise. Negative values rotate it counterclockwise.

GETTING STARTED

Getting **TargetMaker** up and running is a simple process:

1. Create your target prefab either using one of the included presets or make your own from the pieces provided.
    - Creating your own target from scratch:
        - It's recommended that you start with an empty game object and add all the pieces of your cursor as child objects.
        - In order to make sure your target accurately follows the cursor you should place all child objects at 0,0,0 so they are centered on the transform of the parent object.
        - Convert your parent and attached child objects into your new target prefab. This prefab can be named whatever you wish.

2. Add the **AttachToCursor** script to your target prefab, then place it in your scene.
    - Make sure you set the AxisPair to the pair of axes appropriate for your scene.
    - Make sure there's a collider in your scene for the AttachToCursor script to raycast against.
3. Add the **CursorManager** script to an empty gameObject in your scene and add any audio or effects you want. Make sure you set the cursorTransform to the transform of the target object in your scene. This is necessary so that all the clickMarkers and clickEffects will happen exactly where your mouse cursor is.
4. That's all there is to it!

If you have any questions or comments about **TargetMaker**, please don't hesitate to contact us (info@ripcorddev.com)