Bounding Boxes 1.0.0

Documentation for the initial release of BoundingBoxes.

Currently, it provides the Editor enhancements for viewing and manipulating the bounding box, as well as the MonoBehavior implementation of the Editor behavior and the associated fields. There is also packaged in WASDCamera, an example implementation of a basic WASD camera constrained by a bounding box.

Support

It is recommended to submit bug reports and feature requests on theGithub Issues page.

gestaltengine@gmail.com

Unity

Currently developed on `2019.02.17f1`

Editor Controls

The editor will serialize the ranges and center point. Edit bounding box will toggle the visibility of the box and its controls.

Reset center to parent

> Will reset the box's center point to it's transform position.

Reset box to defaults

> Will reset the box's ranges to default and center point to it's transform position.

WASDCamera

WASDCamera is a simple example implementation of the bounding boxes which can be attached to a camera in a project via `Add Component -> Scripts -> WASDCamera`

```c# using BoundingBoxes;

public class WASDCamera : BoundingBox { public bool CameraLocked = false; public Camera cam;
```

```
public void Start()
{
    if (cam == null){
        cam = Camera.main;
    }
}

private void FixedUpdate()
{
    if (!CameraLocked)
    {
        float xAxisValue = Input.GetAxis("Horizontal");
        float yAxisValue = Input.GetAxis("Vertical");
        Vector3 cameraMove = new Vector3(xAxisValue, yAxisValue, 0.0f);

        if (cam != null)
        {
            cam.transform.Translate(cameraMove);

            if (cam.transform.position.x < GetLeftRange())
            {
                cam.transform.position = new Vector3(GetLeftRange(), cam.transform.posit
            }
            if (cam.transform.position.x > GetRightRange())
            {
                cam.transform.position = new Vector3(GetRightRange(), cam.transform.posi
            }
            if (cam.transform.position.y > GetUpRange())
            {
                cam.transform.position = new Vector3(cam.transform.position.x, GetUpRang
            }
            if (cam.transform.position.y < GetDownRange())
            {
                cam.transform.position = new Vector3(cam.transform.position.x, GetDownRa
            }
        }
    }
}
```

}```

Available Components

BoundingBox2D

> This is the core Class you can interact with. It already inherits from `MonoBehavior` so you do not need to

> inherit from it again.

WASDCamera

> See section above.

DraggableBoundingBox2D

> The custom editor extension that enables the viewing of and manipulation of the bounding box

Available Fields

Note: These ranges are not safe to detect the edges of the box, use the methods further below which will account for the Box's center point. These ranges can be used to detect the SIZE of the box, however.

float UpRange

> The offset used to calculate the top of the box

float DownRange

> The offset used to calculate the bottom of the box

float LeftRange

> The offset used to calculate the left side of the box

float RightRange

> The offset used to calculate the right side of the box

Vector3 Center

> The defined centerpoint of the bounding box (defaults to its parent object)

Methods

float Area()

> Returns the area of the bounding box

void SetTextColor(Color)

> Sets the color of the bounding box label text

Color GetTextColor()

> Returns the Color of the bounding box label text

void SetOutlineColor(Color)

> Sets the color of the bounding box outline

Color GetOutlineColor()

> Get's the Color of the bounding box outline

void SetFillColor(Color)

> Sets the color of the bounding box fill

Color GetFillColor()

> Gets te color of the bounding box fill

bool Visible()

> Whether or not the bounding box and its controls are visible within the editor

void SetVisible(bool)

> Set the visibility of the bounding box and it's controls

Vector3 TopLeft()

> Returns the point representing the top left most corner of the bounding box, offset from the defined center.

Vector3 TopRight()

> Returns the point representing the top right most corner of the bounding box, offset from the defined center.

Vector3 BottomLeft()

> Returns the point representing the bottom left most corner of the bounding box, offset from the defined center.

Vector3 BottomRight()

> Returns the point representing the bottom right most corner of the bounding box, offset from the defined center.

float GetLeftRange()

> Returns the left `x` of the bounding box, offset from the defined center.

Vector3 LeftMiddle()

> Returns a Vector3 representing the left most middle point of the bounding vox offset from the center of the mounding box.

float GetRightRange()

> Returns the right `x` of the bounding box, offset from the defined center.

Vector3 RightMiddle()

> Returns a Vector3 representing the right most middle point of the bounding vox offset from the center of the mounding box.

float GetUpRange()

> Returns the top `y` of the bounding box, offset from the defined center.

Vector3 TopMiddle()

> Returns a Vector3 representing the top most middle point of the bounding vox offset from the center of the mounding box.

float GetDownRange()

> Returns the bottom `y` of the bounding box, offset from the defined center.

Vector3 BottomMiddle()

> Returns a Vector3 representing the bottom most middle point of the bounding vox offset from the center of the mounding box.

void SetLeftRange(float)

> Sets the left range

void SetRightRange(float)

> Sets the right range

void SetUpRange(float)

> Sets the up range

void SetDownRange(float)

> Sets the down range

Authors

- Charles Corbett gestaltengine@gmail.com