My Project

AUTHOR Ruchmair Dixon
Version
Sun Sep 9 2018

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Class Documentation

DaiMangou.RadarBuilder3D.DMMath Class Reference

DMMath functions

Static Public Member Functions

- static float **REQS** (float screenHeight, float valiue)
- static float **REQS** (float screenHeight)

Public Attributes

- const float **Mv** = 2.287128713f
- const float **ScalingConstant** = 0.00215f

Detailed Description

DMMath functions

The documentation for this class was generated from the following file:

- C:/MyProjects/Visual Studio/DaiMangou/RadarBuilder3D/RadarBuilder3D/RadarBuilder3D/_3DRadarBuilder.cs

DaiMangou.RadarBuilder3D.Editor.MiniMapModule Class Reference

Minimap Module

Public Member Functions

- Sprite **MaskSprite** ()
  generates are sprite specificially used for the mask layer of the Radar
- Mesh **ProceduralMapQuad** ()

Public Attributes

- **MapType mapType** = **MapType.Realtime**
  Choose between Realtime minimap or a stati minimap
- Sprite **MapTexture**
  Texture to be used for static minimaps
- Sprite **CustomMapMaskShape**
  take in a sprite value and will use it as the mask for the minimap
- bool **generated**
  Check if the map has been generated
- bool **calibrate**
  Determine if the static minimap is being calibrated
- bool **UseCustomMapMaskShape**
  if teue , will allow you to set a custom map mask sprite
- GameObject **Map**
  the objet which will use the Map texture and Masked Material
- GameObject **Mask**
  the object which will use the mask material
- GameObject **MapPivot**
  The parent gameobject of the Map
- float **SavedSceneScale**
  Cashe of the SceneScale vlaue
- float **MapScale** = 1
  The value set during calibratin of ststic minimap
- float **SavedMapScale**
  Cashe of the MapScale vlaue
- float **Scalingfactor**
  Determines by what rate the minmap is scales at rintime
- Material **MapMaterial**
  Masked material
- Material **MaskMaterial**
  Mask Material
- LayerMask **layer**
  The layer on which the minimap will be rendered
- RenderTexture **renderTexture**
  the RenderTexture to be used with the realtime minimap
- Camera **RealtimeMinimapCamera**

The camera reading the RenderTexture for the Minimap

- float **CameraHeight**
  the position of the RealtimeMinimapCamera in the Y axis
- int **OrderInLayer** = -1
  the order in layer of the blip

Detailed Description

Minimap Module

Member Function Documentation

Sprite DaiMangou.RadarBuilder3D.Editor.MiniMapModule.MaskSprite ()

generates are sprite specificially used for the mask layer of the Radar

Returns:

Mesh DaiMangou.RadarBuilder3D.Editor.MiniMapModule.ProceduralMapQuad ()

Returns:

Member Data Documentation

bool DaiMangou.RadarBuilder3D.Editor.MiniMapModule.calibrate

Determine if the static minimap is being calibrated

float DaiMangou.RadarBuilder3D.Editor.MiniMapModule.CameraHeight

the position of the RealtimeMinimapCamera in the Y axis

Sprite DaiMangou.RadarBuilder3D.Editor.MiniMapModule.CustomMapMaskShape

take in a sprite value and will use it as the mask for the minimap

bool DaiMangou.RadarBuilder3D.Editor.MiniMapModule.generated

Check if the map has been generated

LayerMask DaiMangou.RadarBuilder3D.Editor.MiniMapModule.layer

The layer on which the minimap will be rendered

GameObject DaiMangou.RadarBuilder3D.Editor.MiniMapModule.Map

the objet which will use the Map texture and Masked Material

Material DaiMangou.RadarBuilder3D.Editor.MiniMapModule.MapMaterial

Masked material

GameObject DaiMangou.RadarBuilder3D.Editor.MiniMapModule.MapPivot

The parent gameobject of the Map

float DaiMangou.RadarBuilder3D.Editor.MiniMapModule.MapScale = 1

The value set during calibratin of ststic minimap

Sprite DaiMangou.RadarBuilder3D.Editor.MiniMapModule.MapTexture

Texture to be used for static minimaps

MapType DaiMangou.RadarBuilder3D.Editor.MiniMapModule.mapType = MapType.Realtime

Choose between Realtime minimap or a stati minimap

GameObject DaiMangou.RadarBuilder3D.Editor.MiniMapModule.Mask

the object which will use the mask material

Material DaiMangou.RadarBuilder3D.Editor.MiniMapModule.MaskMaterial

Mask Material

int DaiMangou.RadarBuilder3D.Editor.MiniMapModule.OrderInLayer = -1

the order in layer of the blip

Camera DaiMangou.RadarBuilder3D.Editor.MiniMapModule.RealtimeMinimapCamera

The camera reading the RenderTexture for the Minimap

RenderTexture DaiMangou.RadarBuilder3D.Editor.MiniMapModule.renderTexture

the RenderTexture to be used with the realtime minimap

float DaiMangou.RadarBuilder3D.Editor.MiniMapModule.SavedMapScale

Cashe of the MapScale vlaue

float DaiMangou.RadarBuilder3D.Editor.MiniMapModule.SavedSceneScale

Cashe of the SceneScale vlaue

float DaiMangou.RadarBuilder3D.Editor.MiniMapModule.Scalingfactor

Determines by what rate the minmap is scales at rintime

bool DaiMangou.RadarBuilder3D.Editor.MiniMapModule.UseCustomMapMaskShape

if teue , will allow you to set a custom map mask sprite

The documentation for this class was generated from the following file:

- C:/MyProjects/Visual Studio/DaiMangou/RadarBuilder3D/RadarBuilder3D/RadarBuilder3D/_3DRadarBuilder.cs

DaiMangou.RadarBuilder3D.Editor.OptimizationModule Class Reference

Options for optimizing the radars functions

Public Attributes

- int **poolSize**
  pool size for objects you wish to track
- bool **SetPoolSizeManually** = false
  Determines if the blip will be using pooling
- **ObjectFindingMethod objectFindingMethod** = **ObjectFindingMethod.Recursive**
  Options for usng two diferent object finding methods
- bool **RemoveBlipsOnTagChange**
  if true , blips will be removed if the object they track has lost its original tag
- bool **RemoveBlipsOnDisable**
  if true , blips will be removed if the object they track has been disabled
- bool **RequireInstanceObjectCheck**
  if true and you are using Recursive optimization method then you can call _3DRadar.radar3D.doInstanceObjectCheck() trigger object search;
- bool **RecalculatePoolSizeBasedOnFirstFoundObjects**
  By setting this to true, you can ensure that evne if your pool value at atart is greater then the actual amount of objects that can be found , your pool value will be reset to the amount of objects found ao that recusrsive searching is avoided

---

Detailed Description

Options for optimizing the radars functions

Member Data Documentation

ObjectFindingMethod DaiMangou.RadarBuilder3D.Editor.OptimizationModule.objectFindingMethod = ObjectFindingMethod.Recursive

Options for usng two diferent object finding methods

int DaiMangou.RadarBuilder3D.Editor.OptimizationModule.poolSize

pool size for objects you wish to track

bool DaiMangou.RadarBuilder3D.Editor.OptimizationModule.RecalculatePoolSizeBasedOnFirstFoundObjects

By setting this to true, you can ensure that evne if your pool value at atart is greater then the actual amount of objects that can be found , your pool value will be reset to the amount of objects found ao that recusrsive searching is avoided

bool DaiMangou.RadarBuilder3D.Editor.OptimizationModule.RemoveBlipsOnDisable

if true , blips will be removed if the object they track has been disabled

bool DaiMangou.RadarBuilder3D.Editor.OptimizationModule.RemoveBlipsOnTagChange

if true , blips will be removed if the object they track has lost its original tag

bool DaiMangou.RadarBuilder3D.Editor.OptimizationModule.RequireInstanceObjectCheck

if true and you are using Recursive optimization method then you can call _3DRadar.radar3D.doInstanceObjectCheck() trigger object search;

bool DaiMangou.RadarBuilder3D.Editor.OptimizationModule.SetPoolSizeManually = false

Determines if the blip will be using pooling

The documentation for this class was generated from the following file:

- C:/MyProjects/Visual Studio/DaiMangou/RadarBuilder3D/RadarBuilder3D/RadarBuilder3D/_3DRadarBuilder.cs

DaiMangou.RadarBuilder3D.Editor.RadarBlips3D Class Reference

Public Attributes

- bool **DoRemoval** = false
  Tell the blip to do a removal of blips if the Recursive optimization method is used
- bool **Instanced**
  checks if all blips have ben instanced
- bool **IsActive**
  check if the blip is set turned on or off
- bool **ShowBLipSettings**
  INTERNAL USE ONLY
- bool **ShowSpriteBlipSettings**
  INTERNAL USE ONLY
- bool **ShowMeshBlipSettings**
  INTERNAL USE ONLY
- bool **ShowPrefabBlipSettings**
  INTERNAL USE ONLY
- bool **IsTrackRotation**
  Determines if the blip will be tracking the rotation of its target
- bool **BlipCanScleBasedOnDistance**
  Determines if th blips can scale by distance
- bool **ShowTrackingLineSettings**
  INTERNL USE ONLY
- bool **UseTrackingLine**
  Determines if we should use tracking lines or not.
- bool **UseBaseTracker**
  Determines if we should use basetrackers or not
- bool **ShowBaseTrackerSettings**
  INTERNAL USE ONLY
- bool **lockX**
  Determines if the X rotation of the tracked object should be locked to 0
- bool **lockY**
  Determines if the Y rotation of the tracked object should be locked to 0
- bool **lockZ**
  Determines if the Z rotation of the tracked object should be locked to 0
- bool **UseLOD**
  Determines if the mesh blip will use a the Radar Builder LOD system
- bool **ShowLODSettings**
  INTERNAL USE ONLY
- bool **ShowGeneralSettings**
  INTERNAL USE ONLY
- bool **ShowAdditionalOptions**

INTERNAL USE ONLY

- bool **AlwaysShowBlipsInRadarSpace**
  determines if the blip should always remeing inside the radar
- bool **ShowLowMeshSetings**
  INTERNAL USE ONLY
- bool **ShowMediumMeshSettings**
  INTERNAL USE ONLY
- bool **ShowHighMeshSettings**
  INTERNAL USE ONLY
- bool **ShowOptimizationSettings**
  INTERNAL USE ONLY
- bool **SmoothScaleTransition**
  if you are using Always Show and Scale By Distance , this will ensure that you have a smooth ttansition from the moment your blip passes the Tracking Bounds to the moment is is scales to its minimaum scale
- bool **UseCustomRotation**
  Set to true if you wish to give the blips a custom rotation
- bool **KeepBlipsAboveRadarPlane**
  ensures that the blips do not go under the radar
- Sprite **icon**
  The blip icon if the blip is a sprite
- Sprite **BaseTracker**
  The base tracker sprite
- Transform **prefab**
  Prefab blip
- string **State** = ""
  INTERNAL USE ONLY
- string **Tag** = "Untagged"
  INTERNAL USE ONLY
- Material **SpriteMaterial**
  The material used for the sprite blip
- Material **TrackingLineMaterial**
  The material used for the tracking line ///
- Material **BaseTrackerMaterial**
  The material used for the base tracker
- Mesh **mesh**
  The mesh blip mesh when LOD is not active
- Mesh **Low**
  The low poly mesh when LOD is active
- Mesh **Medium**
  The medium poly count mesh when the LOD is active
- Mesh **High**
  The high poly count mesh when the LOD is active
- Material[] **MeshMaterials** = new Material[1]
  All mesh materials usd by the Mesh
- Color **colour** = new Color(1F, 0.6F, 0F, 0.8F)
  THe colour of the material

- Color **TrackingLineStartColour** = new Color(1F, 0.435F, 0F, 0.5F)
  The colour start of the tracking line
- Color **TrackingLineEndColour** = new Color(1F, 0.435F, 0F, 0.5F)
  The end colour of the tracking line
- Color **BaseTrackerColour** = new Color(1F, 0.435F, 0F, 0.5F)
  The colour ued by the base tracker material
- float **BlipSize** = 1
  The size of the blip
- const float **DynamicBlipSize** = 0.025f
  The default minimum scale of the blip
- float **BlipMinSize** = 0.5f
  The minimum size of the blip
- float **BlipMaxSize** = 1
  The maximum size of the blip
- float **TrackingLineDimention** = 0.02F
  The width of the tracking line
- float **LowDistance**
  The distance at which the LOW mesh will replace the current mesh of the mesh blip
- float **MediumDistance**
  The distance at which the MEDIUM mesh will replace the current mesh of the mesh blip
- float **HighDistance**
  The distance at which the HIGH mesh will replace the current mesh of the mesh blip
- float **BaseTrackerSize** = 0.5f
  The scale of th base tracker
- float **CustomXRotation** = 0
  Custom X Rotation For BLips
- float **CustomYRotation** = 0
  Custom Y Rotation For BLips
- float **CustomZRotation** = 0
  Custom Z Rotation For BLips
- int **NumberOfBLips**
  INTERNAL USE ONLY
- int **count**
  INTERNAL USE ONLY
- int **MatCount** = 1
  INTERNAL USE ONLY
- int **Layer** = 0
  INTERNAL USE ONLY
- List< GameObject > **TrackingLineObject** = new List<GameObject>()
  A list of All tracking lines
- List< GameObject > **gos** = new List<GameObject>()
  A list of the objects being tracked
- List< Transform > **RadarObjectToTrack** = new List<Transform>()
  A list of the actual blips you see in your radar
- List< GameObject > **BaseTrackers** = new List<GameObject>()
- **CreateBlipAs _CreateBlipAs**

Determines what the blip should be created as , prefab or sprite

- int **ObjectCount** = -1
  records the amount of tracked objects in the radr far this blip type
- int **OrderInLayer** = 1
  the order in layer of the blip
- SortingLayer **sortingLayer**
  Sorting layer of the sprite blip
- **OptimizationModule optimization** = new **OptimizationModule**()
  Methods of optimizing radar performance

Detailed Description

Member Data Documentation

CreateBlipAs DaiMangou.RadarBuilder3D.Editor.RadarBlips3D._CreateBlipAs

Determines what the blip should be created as , prefab or sprite

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.AlwaysShowBlipsInRadarSpace

determines if the blip should always remeing inside the radar

Sprite DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BaseTracker

The base tracker sprite

Color DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BaseTrackerColour = new Color(1F, 0.435F, 0F, 0.5F)

The colour ued by the base tracker material

Material DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BaseTrackerMaterial

The material used for the base tracker

List<GameObject> DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BaseTrackers = new List<GameObject>()

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BaseTrackerSize = 0.5f

The scale of th base tracker

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BlipCanScleBasedOnDistance

Determines if th blips can scale by distance

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BlipMaxSize = 1

The maximum size of the blip

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BlipMinSize = 0.5f

The minimum size of the blip

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.BlipSize = 1

The size of the blip

Color DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.colour = new Color(1F, 0.6F, 0F, 0.8F)

THe colour of the material

int DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.count

INTERNAL USE ONLY

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.CustomXRotation = 0

Custom X Rotation For BLips

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.CustomYRotation = 0

Custom Y Rotation For BLips

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.CustomZRotation = 0

Custom Z Rotation For BLips

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.DoRemoval = false

Tell the blip to do a removal of blips if the Recursive optimization method is used

const float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.DynamicBlipSize = 0.025f

The default minimum scale of the blip

List<GameObject> DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.gos = new List<GameObject>()

A list of the objects being tracked

Mesh DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.High

The high poly count mesh when the LOD is active

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.HighDistance

The distance at which the HIGH mesh will replace the current mesh of the mesh blip

Sprite DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.icon

The blip icon if the blip is a sprite

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.Instanced

checks if all blips have ben instanced

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.IsActive

check if the blip is set turned on or off

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.IsTrackRotation

Determines if the blip will be tracking the rotation of its target

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.KeepBlipsAboveRadarPlane

ensures that the blips do not go under the radar

int DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.Layer = 0

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.lockX

Determines if the X rotation of the tracked object should be locked to 0

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.lockY

Determines if the Y rotation of the tracked object should be locked to 0

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.lockZ

Determines if the Z rotation of the tracked object should be locked to 0

Mesh DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.Low

The low poly mesh when LOD is active

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.LowDistance

The distance at which the LOW mesh will replace the current mesh of the mesh blip

int DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.MatCount = 1

INTERNAL USE ONLY

Mesh DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.Medium

The medium poly count mesh when the LOD is active

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.MediumDistance

The distance at which the MEDIUM mesh will replace the current mesh of the mesh blip

Mesh DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.mesh

The mesh blip mesh when LOD is not active

Material [] DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.MeshMaterials = new Material[1]

All mesh materials usd by the Mesh

int DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.NumberOfBLips

INTERNAL USE ONLY

int DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ObjectCount = -1

records the amount of tracked objects in the radr far this blip type

OptimizationModule DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.optimization = new OptimizationModule()

Methods of optimizing radar performance

int DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.OrderInLayer = 1

the order in layer of the blip

Transform DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.prefab

Prefab blip

List<Transform> DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.RadarObjectToTrack = new List<Transform>()

A list of the actual blips you see in your radar

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowAdditionalOptions

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowBaseTrackerSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowBLipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowGeneralSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowHighMeshSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowLODSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowLowMeshSetings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowMediumMeshSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowMeshBlipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowOptimizationSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowPrefabBlipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowSpriteBlipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.ShowTrackingLineSettings

INTERNL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.SmoothScaleTransition

if you are using Always Show and Scale By Distance , this will ensure that you have a smooth ttansition from the moment your blip passes the Tracking Bounds to the moment is is scales to its minimaum scale

SortingLayer DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.sortingLayer

Sorting layer of the sprite blip

Material DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.SpriteMaterial

The material used for the sprite blip

string DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.State = ""

INTERNAL USE ONLY

string DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.Tag = "Untagged"

INTERNAL USE ONLY

float DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.TrackingLineDimention = 0.02F

The width of the tracking line

Color DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.TrackingLineEndColour = new Color(1F, 0.435F, 0F, 0.5F)

The end colour of the tracking line

Material DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.TrackingLineMaterial

The material used for the tracking line ///

List<GameObject> DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.TrackingLineObject = new List<GameObject>()

A list of All tracking lines

Color DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.TrackingLineStartColour = new Color(1F, 0.435F, 0F, 0.5F)

The colour start of the tracking line

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.UseBaseTracker

Determines if we should use basetrackers or not

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.UseCustomRotation

Set to true if you wish to give the blips a custom rotation

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.UseLOD

Determines if the mesh blip will use a the Radar Builder LOD system

bool DaiMangou.RadarBuilder3D.Editor.RadarBlips3D.UseTrackingLine

Determines if we should use tracking lines or not.

The documentation for this class was generated from the following file:

- C:/MyProjects/Visual Studio/DaiMangou/RadarBuilder3D/RadarBuilder3D/RadarBuilder3D/_3DRadarBuilder.cs

DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D Class Reference

Public Attributes

- bool **Instanced**
  checks if all blips have ben instanced
- bool **IsActive**
- bool **ShowBLipSettings**
  INTERNAL USE ONLY
- bool **ShowSpriteBlipSettings**
  INTERNAL USE ONLY
- bool **ShowMeshBlipSettings**
  INTERNAL USE ONLY
- bool **ShowPrefabBlipSettings**
  INTERNAL USE ONLY
- bool **IsTrackRotation**
  Determines if the blip will be tracking the rotation of its target
- bool **lockX**
  Determines if the X rotation of the tracked object should be locked to 0
- bool **lockY**
  Determines if the Y rotation of the tracked object should be locked to 0
- bool **lockZ**
  Determines if the Z rotation of the tracked object should be locked to 0
- bool **ShowGeneralSettings**
  INTERNAL USE ONLY
- bool **AlwaysShowCenterObject**
  Determines if the enter object or center blip should alwats be shown in th radar
- bool **CenterObjectCanScaleByDistance**
  Determines if the center object (center blip) can scale by distance
- bool **ShowAdditionalOptions**
  INTERNAL USE ONLY
- bool **ShowTrackingLineSettings**
  INTERNL USE ONLY
- bool **UseTrackingLine**
  Determines if we should use tracking lines or not.
- bool **UseBaseTracker**
  Determines if we should use basetrackers or not
- bool **ShowBaseTrackerSettings**
  INTERNAL USE ONLY
- bool **SmoothScaleTransition**
  if you are using Always Show and Scale By Distance , this will ensure that you have a smooth ttansition from the moment your blip passes the Tracking Bounds to the moment is is scales to its minimaum scale
- bool **UseCustomRotation**

Set to true if you wish to give the center blip a custom rotation

- Sprite **icon**
  The blip icon if the blip is a sprite
- Sprite **BaseTracker**
  The base tracker sprite
- Transform **prefab**
  Prefab blip
- string **State** = ""
  INTERNAL USE ONLY
- string **Tag** = "Player"
  INTERNAL USE ONLY
- Material **SpriteMaterial**
  The material used for the sprite blip
- Material **TrackingLineMaterial**
  The material used for the tracking line ///
- Material **BaseTrackerMaterial**
  The material used for the base tracker
- Mesh **mesh**
  The mesh blip mesh when LOD is not active
- Material[] **MeshMaterials** = new Material[1]
  All mesh materials usd by the Mesh
- Color **colour** = new Color(1F, 0.435F, 0F, 0.5F)
  THe colour of the material
- Color **TrackingLineStartColour** = new Color(1F, 0.435F, 0F, 0.5F)
  The colour start of the tracking line
- Color **TrackingLineEndColour** = new Color(1F, 0.435F, 0F, 0.5F)
  The end colour of the tracking line
- Color **BaseTrackerColour** = new Color(1F, 0.435F, 0F, 0.5F)
  The colour ued by the base tracker material
- float **BlipSize** = 1
  The size of the blip
- float **CustomXRotation** = 0
  Custom X Rotation For Center Blip
- float **CustomYRotation** = 0
  Custom Y Rotation For Center Blip
- float **CustomZRotation** = 0
  Custom Z Rotation For Center Blip
- float **TrackingLineDimention** = 0.2f
  The width of the tracking line
- const float **DynamicBlipSize** = 0.025f
  The default minimum scale of the blip
- float **BlipMinSize** = 0.5f
  The minimum size of the blip
- float **BlipMaxSize** = 1
  The maximum size of the blip
- float **BaseTrackerSize** = 0.5f

The scale of th base tracker

- int **Layer** = 0
  INTERNAL USE ONLY
- int **OrderInLayer** = 1
  the order in layer of the blip
- GameObject **CenterBlip**
  The blip at the center of the radar
- Transform **CenterObject**
  The object being tracked to and used to represent the CenterBlip
- int **MatCount** = 1
  INTERNAL USE ONLY
- **CreateBlipAs _CreateBlipAs**
  Determines what the blip should be created as , prefab or sprite
- GameObject **BaseTrackerObject**
  Object which will sit on the y plane of the radar at all time
- GameObject **TrackingLine**
  Line wich will indicate distance in height from the centerobject to the radar </summary

---

Detailed Description

Member Data Documentation

CreateBlipAs DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D._CreateBlipAs

Determines what the blip should be created as , prefab or sprite

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.AlwaysShowCenterObject

Determines if the enter object or center blip should alwats be shown in th radar

Sprite DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BaseTracker

The base tracker sprite

Color DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BaseTrackerColour = new Color(1F, 0.435F, 0F, 0.5F)

The colour ued by the base tracker material

Material DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BaseTrackerMaterial

The material used for the base tracker

GameObject DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BaseTrackerObject

Object which will sit on the y plane of the radar at all time

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BaseTrackerSize = 0.5f

The scale of th base tracker

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BlipMaxSize = 1

The maximum size of the blip

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BlipMinSize = 0.5f

The minimum size of the blip

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.BlipSize = 1

The size of the blip

GameObject DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.CenterBlip

The blip at the center of the radar

Transform DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.CenterObject

The object being tracked to and used to represent the CenterBlip

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.CenterObjectCanScaleByDistance

Determines if the center object (center blip) can scale by distance

Color DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.colour = new Color(1F, 0.435F, 0F, 0.5F)

THe colour of the material

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.CustomXRotation = 0

Custom X Rotation For Center Blip

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.CustomYRotation = 0

Custom Y Rotation For Center Blip

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.CustomZRotation = 0

Custom Z Rotation For Center Blip

const float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.DynamicBlipSize = 0.025f

The default minimum scale of the blip

Sprite DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.icon

The blip icon if the blip is a sprite

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.Instanced

checks if all blips have ben instanced

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.IsTrackRotation

Determines if the blip will be tracking the rotation of its target

int DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.Layer = 0

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.lockX

Determines if the X rotation of the tracked object should be locked to 0

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.lockY

Determines if the Y rotation of the tracked object should be locked to 0

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.lockZ

Determines if the Z rotation of the tracked object should be locked to 0

int DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.MatCount = 1

INTERNAL USE ONLY

Mesh DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.mesh

The mesh blip mesh when LOD is not active

Material [] DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.MeshMaterials = new Material[1]

All mesh materials usd by the Mesh

int DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.OrderInLayer = 1

the order in layer of the blip

Transform DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.prefab

Prefab blip

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowAdditionalOptions

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowBaseTrackerSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowBLipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowGeneralSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowMeshBlipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowPrefabBlipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowSpriteBlipSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.ShowTrackingLineSettings

INTERNL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.SmoothScaleTransition

if you are using Always Show and Scale By Distance , this will ensure that you have a smooth ttansition from the moment your blip passes the Tracking Bounds to the moment is is scales to its minimaum scale

Material DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.SpriteMaterial

The material used for the sprite blip

string DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.State = ""

INTERNAL USE ONLY

string DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.Tag = "Player"

INTERNAL USE ONLY

float DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.TrackingLineDimention = 0.2f

The width of the tracking line

Color DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.TrackingLineEndColour = new Color(1F, 0.435F, 0F, 0.5F)

The end colour of the tracking line

Material DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.TrackingLineMaterial

The material used for the tracking line ///

Color DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.TrackingLineStartColour = new Color(1F, 0.435F, 0F, 0.5F)

The colour start of the tracking line

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.UseBaseTracker

Determines if we should use basetrackers or not

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.UseCustomRotation

Set to true if you wish to give the center blip a custom rotation

bool DaiMangou.RadarBuilder3D.Editor.RadarCenterObject3D.UseTrackingLine

Determines if we should use tracking lines or not.

The documentation for this class was generated from the following file:

- C:/MyProjects/Visual Studio/DaiMangou/RadarBuilder3D/RadarBuilder3D/RadarBuilder3D/_3DRadarBuilder.cs

DaiMangou.RadarBuilder3D.Editor.RadarDesign3D Class Reference

Public Attributes

- float **RadarDiameter** = 1
  This is the Diameter of the radar, this value will directly change the scale of the Radars child object "Designs" once UseSceneScale is false
- float **SceneScale** = 100.0f
  This is the amound of the scene that the radar is able to 'see' in order to collect dats on things to track and display
- float **TrackingBounds** = 1
  The range in which all blips can be shown in the radar
- float **InnerCullingZone** = 0f
  The diameter of the zone at the center of the radar in which all blips will ce culled
- float **RadarRotationOffset** = 0f
  INTERNAL USE ONLY
- const float **ConstantRadarRenderDistance** = 4
  Do not replace this value
- float **xPadding**
  The padding on the x and Y axis of the radar system
- **RadarPositioning radarPositioning** = **RadarPositioning.Snap**
  Determins if the radar will ise Manual position or Snap Positioning
- **SnapPosition snapPosition** = SnapPosition.BottomMiddle
  Determines where in scren space the radar system will be positioned
- **FrontIs frontIs** = **FrontIs.North**
  Determining what defines the forward facing position of the radar
- Rect **RadarRect**
  INTERNAL USE ONLY
- int **Count** = 0
  INTERNAL USE ONLY
- int **DesignsCount** = 0
  INTERNAL USE ONLY
- bool **UseLocalScale**
  Determines if we should use the scale of the Radar "Designs" child object instead of the RadarDiameter
- bool **Visualize** = true
  INTERNAL USE ONLY
- bool **LinkToTrackingBounds**
  Determines if the tracking bounds values will always be the same as
- bool **ShowScaleSettings**
  INTERNAL USE ONLY
- bool **ShowRenderCameraSettings**
  INTERNAL USE ONLY
- bool **ShowPositioningSettings**
  INTERNAL USE ONLY

- bool **IgnoreDiameterScale** = false
  When true, the radar ; diameter (Sale of the Radars "Designs" child object) when scales to a vlue greater or less than one will not prompt the radar system to reposition itslf automatically to maintain a correct position in screen space
- bool **ManualCameraSetup**
  INTERNAL USE ONLY
- bool **UseMainCamera**
  determines if we will be using the gameobject in the scne with the tag "Main Camera"
- bool **_3DSystemsWithScreenSpaceFunction**
  Determines if the 3D Radar will also be using the screen space system
- bool **_3DSystemsWithMinimapFunction**
  Determines if the radar can also be a minimap
- bool **ShowMinimapSettings**
  INTERNAL USE ONLY
- bool **UseOrthographicForSideSnaps** = false
  This makes the Render Camera go into orthographics mode to correct for deault camera distortion
- Transform **BlipsParentObject**
  INTERNAL USE ONLY
- Camera **camera**
  The camera which will be the camera your player views the world through at any time
- Camera **renderingCamera**
  The camera whuch will only render radar systems, (These camera are automatically created for you)
- string **CameraTag** = "MainCamera"
  INTERNAL USE ONLY
- List< **RotationTarget** > **RotationTargets** = new List<**RotationTarget**>()
  The list of Rotation targets
- Vector3 **Pan** = new Vector3()
  The pan of the blips in the radar

Detailed Description

Member Data Documentation

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D._3DSystemsWithMinimapFunction

Determines if the radar can also be a minimap

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D._3DSystemsWithScreenSpaceFunction

Determines if the 3D Radar will also be using the screen space system

Transform DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.BlipsParentObject

INTERNAL USE ONLY

Parent objct which will hold all the blips

Camera DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.camera

The camera which will be the camera your player views the world through at any time

string DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.CameraTag = "MainCamera"

INTERNAL USE ONLY

const float DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.ConstantRadarRenderDistance = 4

Do not replace this value

int DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.Count = 0

INTERNAL USE ONLY

int DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.DesignsCount = 0

INTERNAL USE ONLY

FrontIs DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.frontIs = FrontIs.North

Determining what defines the forward facing position of the radar

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.IgnoreDiameterScale = false

When true, the radar ; diameter (Sale of the Radars "Designs" child object) when scales to a vlue greater or less than one will not prompt the radar system to reposition itslf automatically to maintain a correct position in screen space

float DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.InnerCullingZone = 0f

The diameter of the zone at the center of the radar in which all blips will ce culled

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.LinkToTrackingBounds

Determines if the tracking bounds values will always be the same as

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.ManualCameraSetup

INTERNAL USE ONLY

Vector3 DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.Pan = new Vector3()

The pan of the blips in the radar

float DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.RadarDiameter = 1

This is the Diameter of the radar, this value will directly change the scale of the Radars child object "Designs" once UseSceneScale is false

RadarPositioning DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.radarPositioning = RadarPositioning.Snap

Determins if the radar will ise Manual position or Snap Positioning

Rect DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.RadarRect

INTERNAL USE ONLY

float DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.RadarRotationOffset = 0f

INTERNAL USE ONLY

Camera DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.renderingCamera

The camera whuch will only render radar systems, (These camera are automatically created for you)

List<RotationTarget> DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.RotationTargets = new List<RotationTarget>()

The list of Rotation targets

float DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.SceneScale = 100.0f

This is the amound of the scene that the radar is able to 'see' in order to collect dats on things to track and display

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.ShowMinimapSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.ShowPositioningSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.ShowRenderCameraSettings

INTERNAL USE ONLY

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.ShowScaleSettings

INTERNAL USE ONLY

SnapPosition DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.snapPosition = SnapPosition.BottomMiddle

Determines where in scren space the radar system will be positioned

float DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.TrackingBounds = 1

The range in which all blips can be shown in the radar

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.UseLocalScale

Determines if we should use the scale of the Radar "Designs" child object instead of the RadarDiameter

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.UseMainCamera

determines if we will be using the gameobject in the scne with the tag "Main Camera"

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.UseOrthographicForSideSnaps = false

This makes the Render Camera go into orthographics mode to correct for deault camera distortion

bool DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.Visualize = true

INTERNAL USE ONLY

float DaiMangou.RadarBuilder3D.Editor.RadarDesign3D.xPadding

The padding on the x and Y axis of the radar system

The documentation for this class was generated from the following file:

- C:/MyProjects/Visual Studio/DaiMangou/RadarBuilder3D/RadarBuilder3D/RadarBuilder3D/_3DRadarBuilder.cs

DaiMangou.RadarBuilder3D.Editor.RotationTarget Class Reference

Public Attributes

- bool **ShowDesignSetings**
  called only from editor , and is not necessary at runtime
- bool **UseY**
  When true , the z rotation will be the same as the Y rotation
- bool **FreezeX**
  Freeze rotation around particular axis
- float **RotationDamping**
  Damping used to control rotation of particular design layer
- string **tag**
  the string tag you define
- string **FindingName**
  the name of the object you wish to find
- string **InstancedObjectToTrackBlipName**
  The name of the instanced object you wish to target
- string **InstancedTargetBlipname**
  the name of the instanced blip you wish to track
- **Rotations rotations**
  Selection between Inverse rotation and Proportional rotation
- GameObject **TargetedObject**
  This may be a blip or any other object in scene
- GameObject **Target**
  the object whose rotation you wish to target
- **TargetObject ObjectToTrack** = **TargetObject.ThisObject**
  Selection of the way in which you wish to select and object
- **TargetBlip target** = **TargetBlip.ThisObject**
  The blip you wish to target
- **RetargetRotation RetargetedRotation** = **RetargetRotation.none**
- **RetargetX retargetedXRotation** = **RetargetX.none**
  change the x rotation to the y o z
- **RetargetY retargetedYRotation** = **RetargetY.none**
  change the y rotation to the x or z
- **RetargetZ retargetedZRotation** = **RetargetZ.none**
  change the z rotation to the x or y
- float **AddedRotation** = 90
- float **AddedXRotation**
  X rotation value added to the rotation target
- float **AddedYRotation**
  Y rotation value added to the rotation target
- float **AddedZRotation**
  Z rotation value added to the rotation target

Detailed Description

Defines the object which is to be targeted and the way in which it must be rotated

Member Data Documentation

float DaiMangou.RadarBuilder3D.Editor.RotationTarget.AddedXRotation

X rotation value added to the rotation target

float DaiMangou.RadarBuilder3D.Editor.RotationTarget.AddedYRotation

Y rotation value added to the rotation target

float DaiMangou.RadarBuilder3D.Editor.RotationTarget.AddedZRotation

Z rotation value added to the rotation target

string DaiMangou.RadarBuilder3D.Editor.RotationTarget.FindingName

the name of the object you wish to find

bool DaiMangou.RadarBuilder3D.Editor.RotationTarget.FreezeX

Freeze rotation around particular axis

string DaiMangou.RadarBuilder3D.Editor.RotationTarget.InstancedObjectToTrackBlipName

The name of the instanced object you wish to target

string DaiMangou.RadarBuilder3D.Editor.RotationTarget.InstancedTargetBlipname

the name of the instanced blip you wish to track

TargetObject DaiMangou.RadarBuilder3D.Editor.RotationTarget.ObjectToTrack = TargetObject.ThisObject

Selection of the way in which you wish to select and object

RetargetX DaiMangou.RadarBuilder3D.Editor.RotationTarget.retargetedXRotation = RetargetX.none

change the x rotation to the y o z

RetargetY DaiMangou.RadarBuilder3D.Editor.RotationTarget.retargetedYRotation = RetargetY.none

change the y rotation to the x or z

RetargetZ DaiMangou.RadarBuilder3D.Editor.RotationTarget.retargetedZRotation = RetargetZ.none

change the z rotation to the x or y

float DaiMangou.RadarBuilder3D.Editor.RotationTarget.RotationDamping

Damping used to control rotation of particular design layer

Rotations DaiMangou.RadarBuilder3D.Editor.RotationTarget.rotations

Selection between Inverse rotation and Proportional rotation

bool DaiMangou.RadarBuilder3D.Editor.RotationTarget.ShowDesignSetings

called only from editor , and is not necessary at runtime

string DaiMangou.RadarBuilder3D.Editor.RotationTarget.tag

the string tag you define

GameObject DaiMangou.RadarBuilder3D.Editor.RotationTarget.Target

the object whose rotation you wish to target

TargetBlip DaiMangou.RadarBuilder3D.Editor.RotationTarget.target = TargetBlip.ThisObject

The blip you wish to target

GameObject DaiMangou.RadarBuilder3D.Editor.RotationTarget.TargetedObject

This may be a blip or any other object in scene

bool DaiMangou.RadarBuilder3D.Editor.RotationTarget.UseY

When true , the z rotation will be the same as the Y rotation

---

The documentation for this class was generated from the following file:

- C:/MyProjects/Visual Studio/DaiMangou/RadarBuilder3D/RadarBuilder3D/RadarBuilder3D/_3DRadarBuilder.cs

Index

INDEX