

Content

| | |
|--------------------------------------|---|
| Fog of War | 1 |
| 0. Introduction and setup..... | 1 |
| 0.0 Introduction to fog of war. | 1 |
| 0.1 Support platform: | 1 |
| 0.2 How to use the utility..... | 1 |
| 1. The process of warfog..... | 2 |
| 1.0 Introduction. | 2 |
| 1.1 Basic concepts | 2 |
| 1.2 API Information. | 3 |

Fog of War

0. Introduction and setup

0.0 Introduction to fog of war.

The utility can create fog on screen. It's used for 2.5D RGP games, like LOL. It's especially suitable for mobile games because of the utility is based on mesh render, not post process.

I'm add algorithms to cacualte fog on cpu, which is based on Greg McIntyre's libfov:<https://sourceforge.net/projects/libfov/>。

0.1 Support platform:

Platform:

Android, iOS, Windows.

Graphic API:

OpenGL ES 2.0 + , Metal, DirectX 9.0 +

0.2 How to use the utility.

1 First, you need to invoke the API:

```
WarFogApi.getInstance().initWarFog(WarFogType    pCurrentType,    Transform  
pParentTransForPlaneWarFogTrans);
```

To init warfog api instance. WarFogType is fog type, only Plane supported now. Root is the root gameobject of warfog plane hook, it can't be null.

2 You need set warfog parments like this:

WarFogApi.getInstance().setParments(*Material* pRenderMat, *Vector3* pOriginPos, float pWidthPerCell, float pHeightPerCell, int pH_Cell_Counts, int pV_Cell_Counts);
The invoking will create plane mesh and set fog parments.

3 Caculater must ben init like this.

```
WarFogApi.getInstance().WarFogCaculater.setMapData(WarFog_Map_Element[,]  
pMapData, int pWidth, int pHeight);
```

4 You can add static check point like this:

```
WarFogApi.getInstance().WarFogCaculater.addOrUpdateStaticCheckPoints(LinkedList  
st<WarFog_CheckPoint> pPointList);
```

Add dynamic check point like this:

```
WarFogApi.getInstance().WarFogCaculater.updateDynamicCheckPoints(LinkedList<War  
Fog_CheckPoint> pPointList);
```

5 You need to set fog texture when complete calculate fog data like this:

```
WarFogApi.getInstance().PlaneDisplayer.setFogData(int[,] pData);
```

6 You need invoke WarFogApi.getInstance().PlaneDisplayer.updateFog(); in Update on MonoBehaviour script. The function is used to fade fog from old data to new data.

I write a demo in demo directory. Please read [WarFogDemo.cs](#) before use the utility.

1. The process of warfog.

1.0 Introduction.

The utility will init warfog while start games. Create fog plane according to parments, and will update fog texture when set fog data. Then fog material will render the fog according to fog texture. I will detail those api later.

1.1 Basic concepts

WarFog_Map_Element: the class is used for store map block information.

WarFog_Element: the class is used for store current fog data.

WarFog_CheckPoint: the class is an vision/sight in game.

WarFog_CheckPointStoreElement: is used for store static sight data for improve efficiency.

WarFog_Plane: the class is used for create fog plane and set fog texture in shader.

WarFog_Caculate: the class is used for cacualte valid fog data in blocked map. The algorithm is based on Greg McIntyre's libfov , you can get more information on :<https://sourceforge.net/projects/libfov/>

ScriptObjectPool is for improve efficiency.

WarFogDemo is a demo, please read it before use the utility.

1.2 API Information.

1.2.1 WarFogApi.getInstance().initWarFog(WarFogType pCurrentType, Transform pParentTransForPlaneWarFogTrans);

:Init warfog .

pCurrentType: type of warfog, only WarFog is supported now.

pParentTransForPlaneWarFogTrans: is the parent transform of warfog plane. You need to sure the root will not be destroyed when change scene.

1.2.2 WarFogApi.getInstance().setParments(Material pRenderMat, Vector3 pOriginPos, float pWidthPerCell, float pHeightPerCell, int pH_Cell_Counts, int pV_Cell_Counts);

:Set parments of warfog panel.

pRenderMat: the material for render fog.

pOriginPos: the origin position of warfog plane. Fog plane will created origin by the position.

pWidthPerCell: width of an cell.

pHeightPerCell: height of an cell.

pH_Cell_Counts: count of cells in horizontal direction.

pV_Cell_Counts: count of cells in verital direction.

1.2.3

WarFogApi.getInstance().WarFogCaculater.setMapData(WarFog_Map_Element[,]

pMapData, int pWidth, int pHeight);

:Init caculater.

pMapData: block data in map

pWidth: width of map.

pHeight: height of map.

1.2.4

WarFogApi.getInstance().WarFogCaculater.addOrUpdateStaticCheckPoints(LinkedList<WarFog_CheckPoint> pPointList);

:Add or update new static check points.

pPointList: list of new check points.

WarFogApi.getInstance().WarFogCaculater.removeStaticCheckPoints

(LinkedList<WarFog_CheckPoint> pPointList);

:delete not used check points.

pPointList: list of deleted check points.

WarFogApi.getInstance().WarFogCaculater.setDynamicCheckPoints(LinkedList<WarFog_CheckPoint> pPointList);

:Set new dynamic check points.

pPointList: list of new check points.

1.2.5

`WarFogApi.getInstance().PlaneDisplayer.setFogData(int[,] pData);`

:After calculate fog data, the fog data need been set in shader and rendered.

pData: valid data of fog.

1.2.6

`WarFogApi.getInstance().PlaneDisplayer.updateFog();`

:the function will update fade process. Because fade fog is a process in display, for fade fog from old to new, you muse invoke this. If not, the display fog is old data, not lastest data.