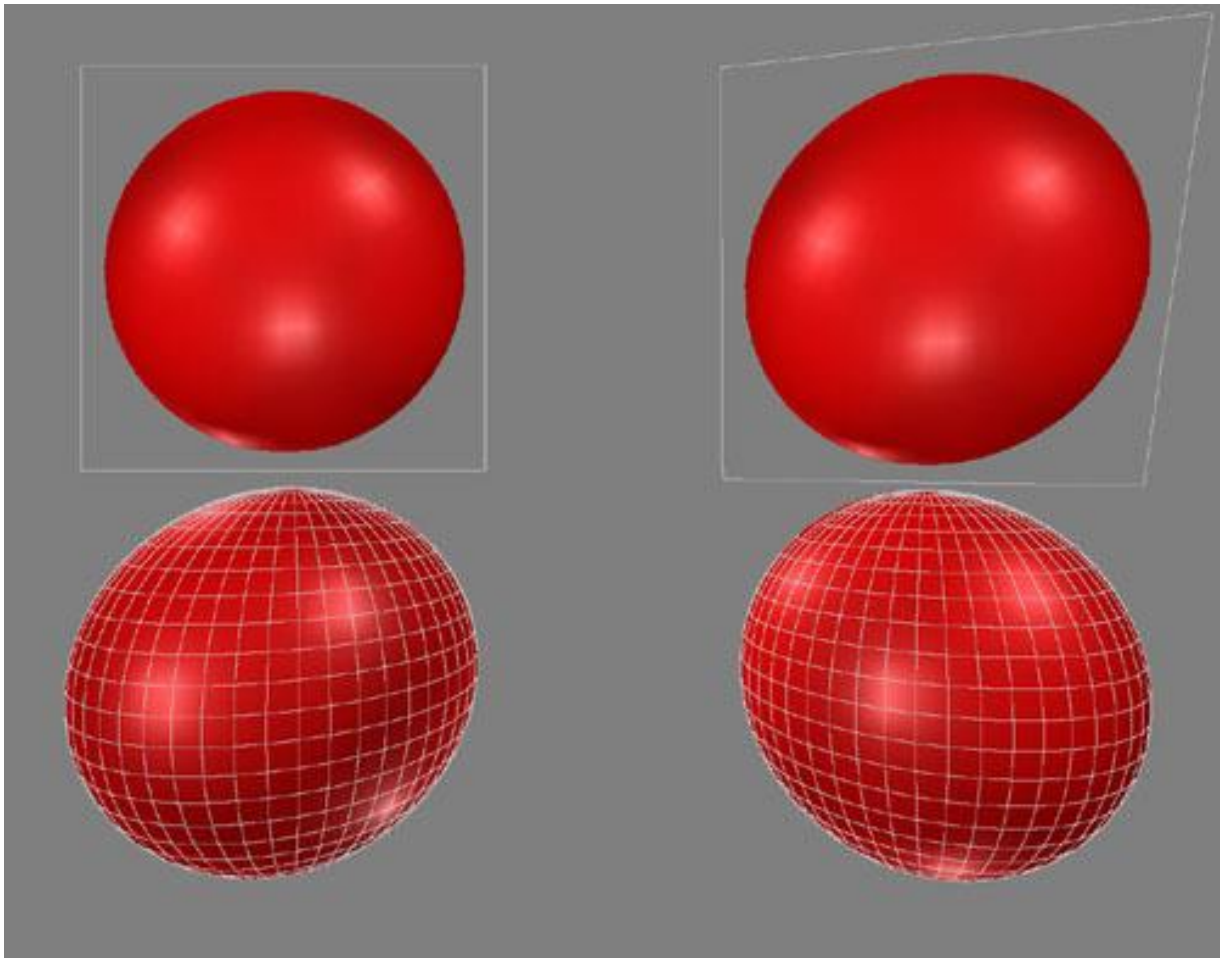
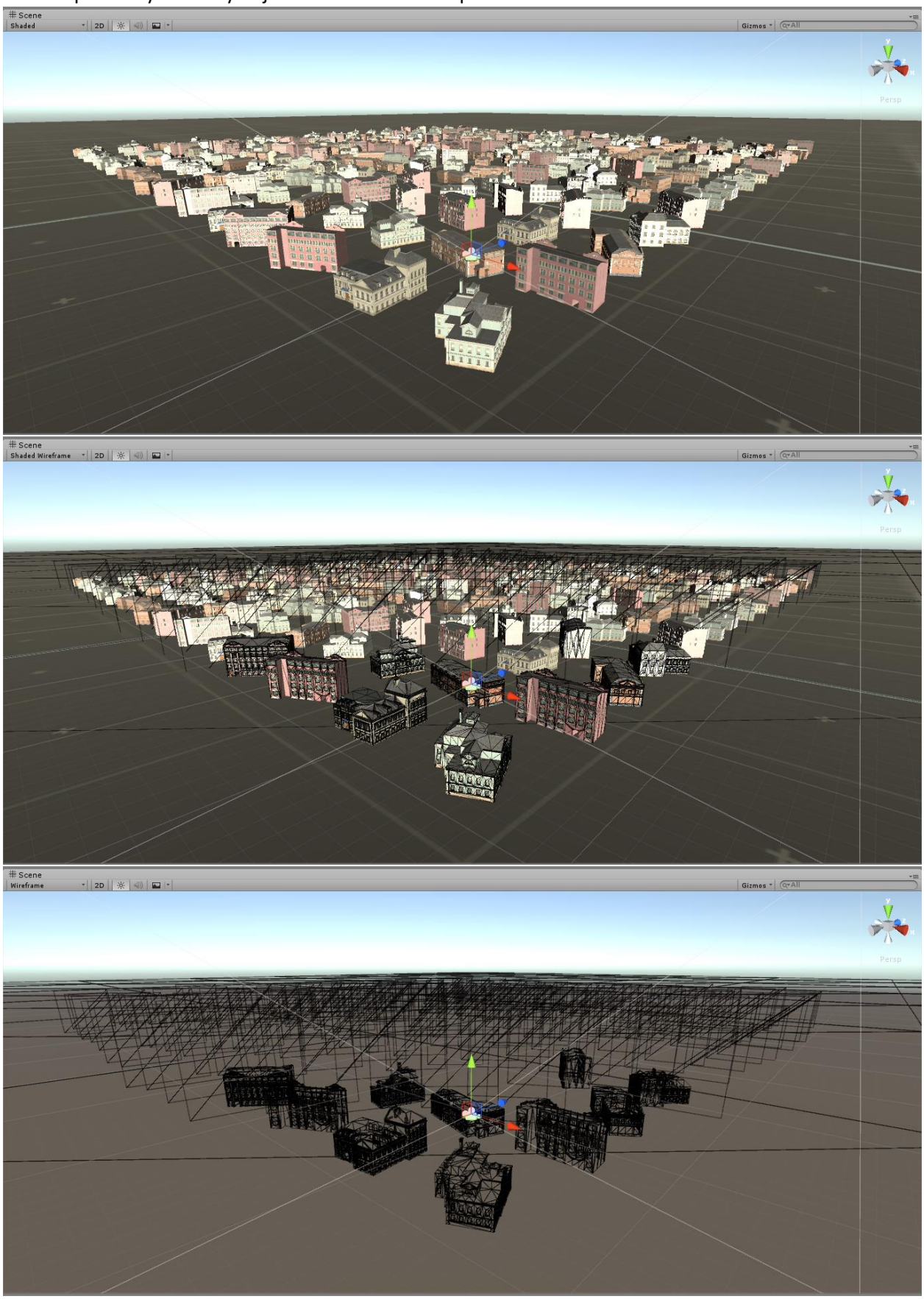
Imposter System v1.2

Documentation

Imposter System reduces the number of polygons in your scene by converting an 3D objects into a 2D sprites(imposters) without any preprocessing. For the player imposter looks like a 3D object, but consumes much less resources. For example:



With Imposter System any objects could become imposters!



Example scene in project.

Upgrade

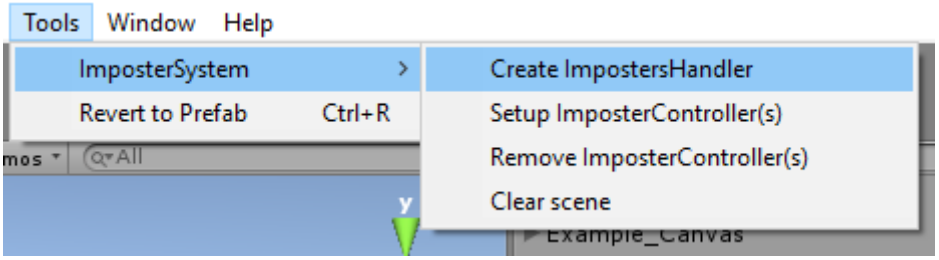
If you upgrading from BillboardSystem, then you must follow this steps, to prevent crashing of your project!

1. **Backup you project!**
2. Open scene that uses BillboardSystem, click on 'Tools/BillboardSystem/Clear scene'
3. Apply all prefabs, who previously had BillboardController component, to prevent 'missed script' warnings
4. Repeat steps 1 and 2 for every scene, that uses BillboardSystem
5. Delete 'BillboardSystem' folder
6. Import new package

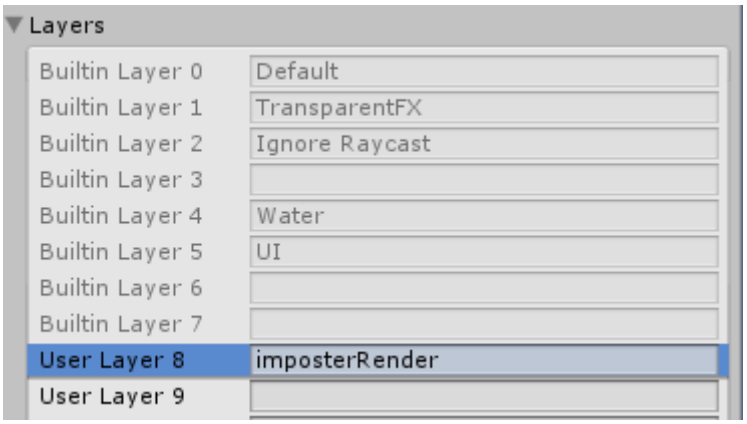
Setup

To begin, I strongly recommend you watch a video tutorial on how to set up and use this package: https://www.youtube.com/watch?v=Zhbu1lOto0M

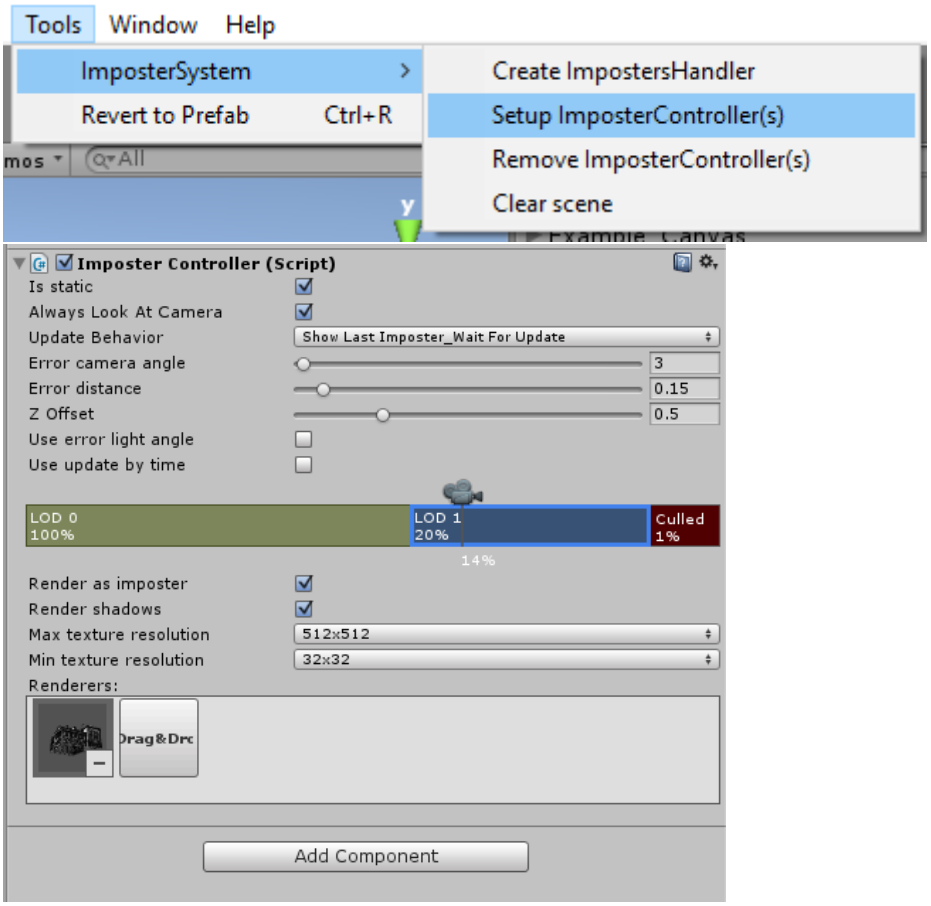
1. Backup your project
2. Import Imposter System unitypackage in your project.
3. On Pop Menu bar click on ImposterSystem-> Create ImpostersHandler, this action will create a game object 'ImpostersHandler', which processes all the information. This step need to do in all scenes, where Imposter System must work.



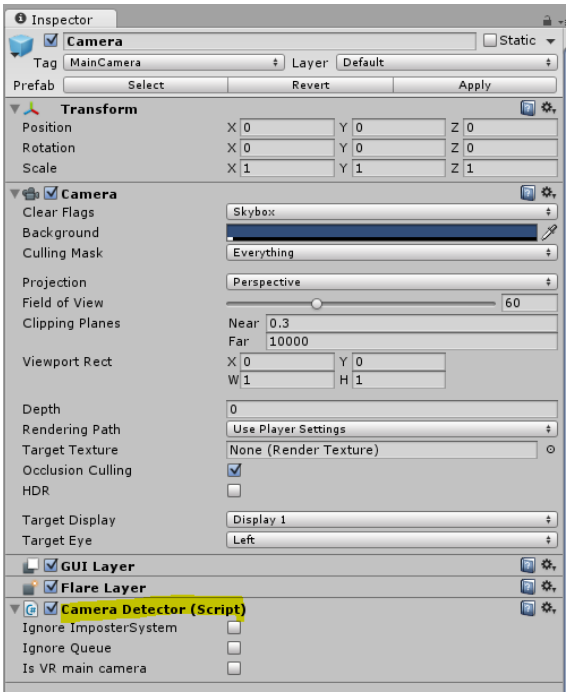
4. ImposterSystem requires layer called 'imposterRender', it should have been created automatically in the previous step. Objects in your scene should not use this layer! If this layer is not created, then add it:



5. Next you need to select the objects that need to be imposters, then on Pop Menu bar click on ImposterSystem -> Setup ImposterController, this action will add to the selected game objects ImposterController component. If game object has LODGroup component, then ImposterController will import LODs settings from LODGroup, and use last LOD to render as imposter. (This way to add ImposterConroller component is **strongly recommended** because it automatically changes the hierarchy for correct work)



6. If game object has no LODGroup component then you need manually add renderers to LODs in ImposterController component(similar to LODGroup setup).
7. The last step is to add a CameraDetector component to the cameras, which should render imposters.



Everything is ready to go!

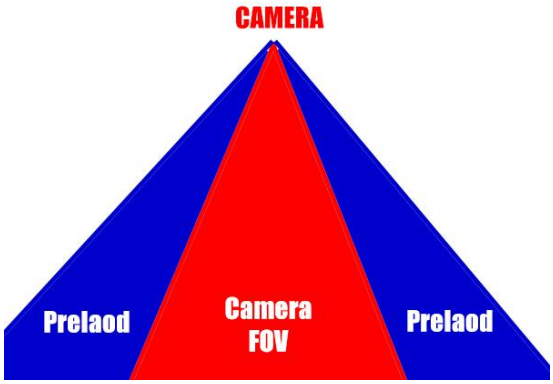
Components

ImposterSystem.ImpostersHandler

#INSPECTOR

- enabled(component toggle) – enable/disable ImposterSystem
- Disable Imposters Updating – enable/disable processing of all imposters, most common use is when you know that player will not move, that's highly increase performance
- Imposters Render Type – the way imposters will be drawn
    - Draw Mesh – (PREFERED) rendering handles by ImposterSystem (mesh combining by ImposterSystem)
    - Mesh Renderer – imposters will be render as gameObjects with MeshRenderer (Unity dynamic batching), doesn't support most of features (fading, shadows…)
- Invisible LODs Action – what need to do with invisible imposters.
    - None - nothing
    - Release Imposter – delete all data about invisible imposters (so if imposter become visible again then its need to recalculate all again)
- Queue Type – how to organize queue of imposters that needs update
    - None – no queue, all imposters updating immediately (may produce fps drops)
    - Simple – simple queue with no sorting
    - Sorted By Screen Size – the more imposter takes screen space, the earlier will be updated
- Light(directional light) – the main light of the scene, need to set if light is dynamic or you want to use Imposter Shadow Casting
- Max Updates Per Frame – how many imposters can be updated per frame(if queue type is not "None")
- Atlas Resolution – It determines the size of the atlas, which will be stored the imposter textures.
- Shader For Render – you can specify a special shader to render imposter textures. Here you can specify a simple shader for less time render. (Replacement Shader)
- Fading (DrawMesh only)
    - Use Fading – if enabled, then create fade transition effect between old and new imposter texture
    - Fade Noise Texture – texture that used to create fade effect (used alpha channel) Example texture located at ImposterSystem/Textures/Noise.psd
    - Fade Time – duration of fade effect in seconds
    - Don't Update When Fading – if enabled, then imposter will not update while fading
- Imposter Shadow Casting (test)
    - Shadow Casting Enabled
    - Light Direction Delta – how often will shadow updated
    - Shadow Texture Resolution – resolution of textures, that are holds shadow data for every single imposter
    - Shadow Atlas Resolution – resolution of atlas, that holds all shadow textures
- Additional Settings

- Preload Factor – multiplies camera fieldOfView by this value (fieldOfView * preloadFactor). New FOV will be used to detect imposters visibility



- Min Angle To Stop Look At Camera – if 'Always Look At Camera' is enabled on ImposterController, then imposter will always look at camera. When camera is right above imposter, then it makes imposter to strangely rotate. To prevent this problem increase 'Min Angle To Stop Look At Camera' value.

#METHODS

- ImpostersHandler Instance – returns the singleton instance of ImpostersHandler
- void UpdateAllImposters(bool immediately = false) – forces all imposters to update their textures
- float GetMemoryUsage() – returns memory amount in KB used by all imposter textures
- void RemoveUnusedTextures(float unusedTime = 0)
- int updatedByFrameImpostersCount(get)
- int currentImpostersCount(get)

ImposterSystem.ImposterController

#INSPECTOR

- IsStatic – if chosen, then object must be static! (Used for optimization)
- Always Look At Camera – if disabled, then imposter rotation will be changing only on texture update
- UpdateBehavior – how imposters will be updating
  - Show3DObject_WaitForUpdate – if imposter is not ready, then camera will render 3D object
  - ShowLastImposter_WaitForUpdate – if imposter is not ready, then camera will render last imposter(maybe incorrect view). If imposter is not generated yet, then it will be created and updated immediately
  - ImmediatelyUpdate – if imposter is need update, then imposter will be updated immediately
- Error camera angle – if the direction changed by this value, then imposter will be updated
- Error distance – if distance(distance between camera and object) relatively changed by this value then imposter will be updated

- Z Offset – imposter'a position relative to the object's center. Used to correctly render imposter with other objects in the scene.

- Use error light angle – allows to update imposter if light direction is changed
    - Error light angle - if light direction changed by this value, then imposter will be updated
- Use update by time – allows to update imposter by time
    - Time interval – time between automatic updates, may be used for animated objects
- m_LODs – imposter levels of detail, the calculation based on their size on the screen. Works as LODGroup.

#METHODS

- void RecalculateBounds()
- void RemoveImposters() – removes all imposters for all cameras
- void UpdateImposter() – forces all already created imposters to update
- void UpdateImposter(Camera camera) – forces already created imposter to update for specific camera
- void UpdateImposterImmediately() – forces all already created imposters to update immediately
- void UpdateImposterImmediately(Camera camera) - forces already created imposter to immediately update for specific camera

ImposterSystem.ImposterLOD

#INSPECTOR

- Render as imposter – if checked then Renderers(described below) will draw as imposter, else as normal renderers.
- Render shadow – allows to render self-shadows(not cast).
- Min/Max texture resolution – imposter texture resolution would be between this values.
- Renderers – renderers that will be drawn

ImposterSystem.CameraDetector

#INSPECTOR

- Ignore ImposterSystem – allows to disable imposters rendering and show original objects
- Ignore Queue – allows to ignore queue of imposters that waiting for update.
- Is VR main camera – allows to use ImposterSystem optimization with VR view(works only without using native Unity VR, because there is some unity limitation with changing camera field of view)

ImposterSystem.Water (Test)

- Default unity water component modified to work with ImposterSystem

Contact developer

http://starasgames.xyz/

All relevant information here

https://forum.unity3d.com/threads/w-i-p-realtime-imposter-system-fake-3d-optimization-plugin.426478/