Hola! , I'm Martin Ysa from Argentina developer from this awesome script.

I want to say thank you for download this package and this document is about how use it.

INDEX

*** – What´s new in version 1.1.0 ? – ***

— MAJOR UPDATE !

version 1.1.0 — 2DLight Now Support ROTATION! implemented by Grit Schuster

version 1.1.0 — 2DLight Now Support Angle restriction, as CONE OF SIGHT! implemented by Grit Schuster

version 1.1.0 — More accurate caster works.

version 1.1.0 — Choice optimisation params.

version 1.1.0 — Circle caster added!

version 1.1.0 — Intellider -> Smart optimizer collider for Circles and Hexagons Casters!

version 1.1.0 — Light Car Scene Added!

version 1.1.0 — Radar scene Added!

version 1.1.0 — TimerClass included! (is a helper Script to add a timer in unity in easy way)

*** – What´s new in version 1.0.5 ? – ***

version 1.0.5 — Add a Layer property for interact with casters

version 1.0.5 — fix raycast collider with non shadow caster layer mask

version 1.0.5 — Extension of Documentation.

version 1.0.5 — Z-Fighting Solved! (notable on iOS devices).

version 1.0.5 — Add VertexWorking public property for tracking how vertex are working in scene.

version 1.0.5 — Enhanced 2.MultipleLights scene ->added “vertex used” and how many 2DLights in scene.

version 1.0.5 — Enhanced algorythm for object detection. Scene 3.CoverExplosion(shockwave)

*** – Adding a 2D Dynamic Lights and Shadows in your Game – ***

First to all, you should open at least one Demo Scene and see how this effects work.
If not, no matter, because this script is ready for include in a fastest way.

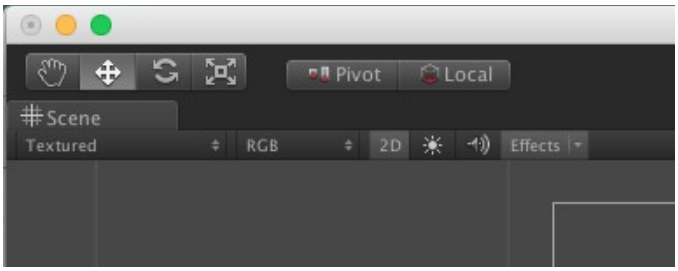
Do this:

1)Download and import 2D Lights and Shadows PRO Package (if you read this is already :))

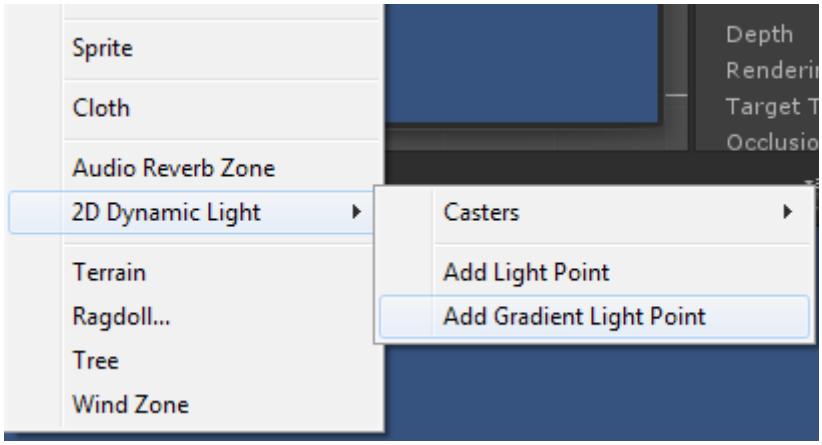
2)Create a empty scene.

3)Set the Camera to Orthographic Proyection in position (X=0 ; Y=1 ; Z=-10) and SIZE=20

4) Set the current project Editor into 2D view design.

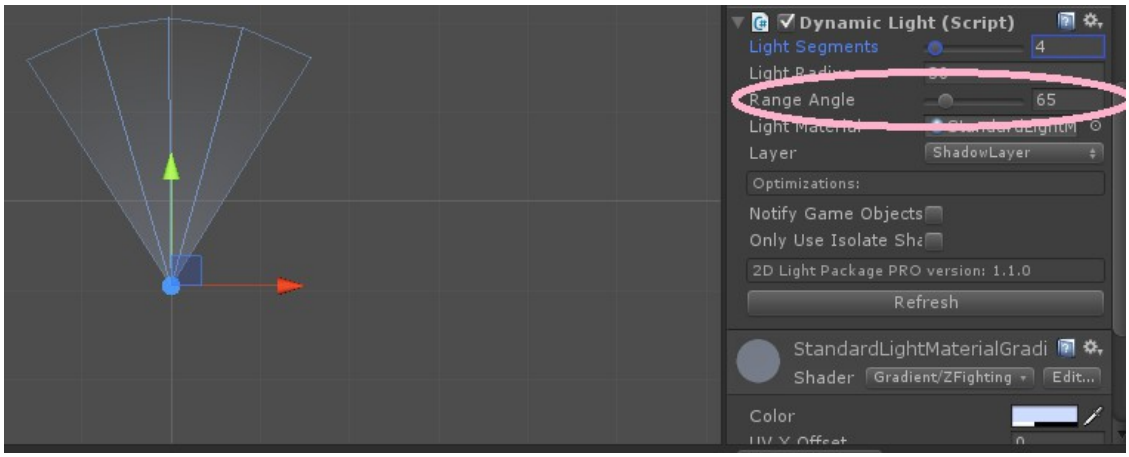


5) From top end Menu, go to GameObject / Create Other / 2D Dynamic Light / Add Gradient Light Point



You will see the point of light like a image below:

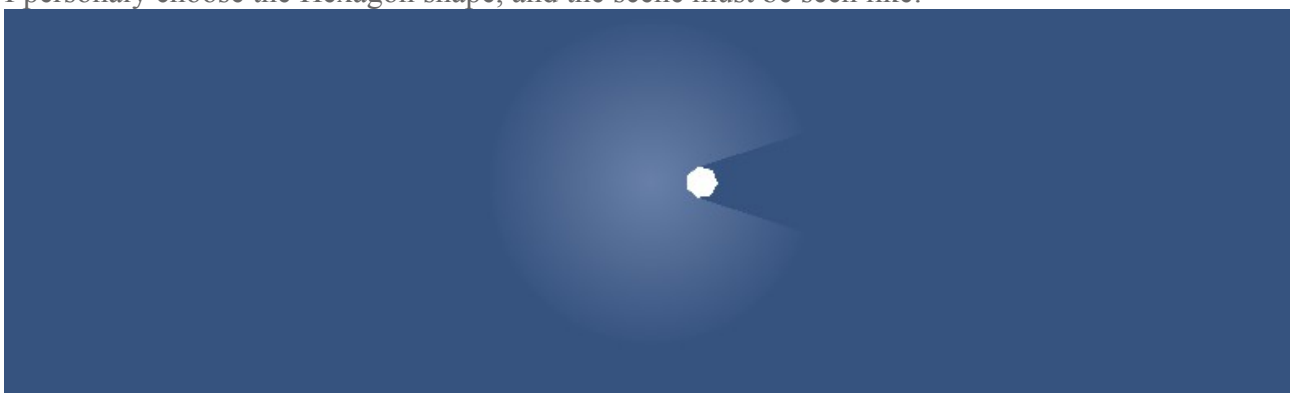
Also you can set the angle restriction and convert the light in a **Spot Light**



6) Light is ready, now we have add some casters:

Go to 2D Dynamic Light menu, and then choose between shapes of casters.

I personaly choose the Hexagon shape, and the scene must be seen like:



REMEMBER that this script run in Editor Mode and also obviously in Game Mode.

CONGRATULATIONS !

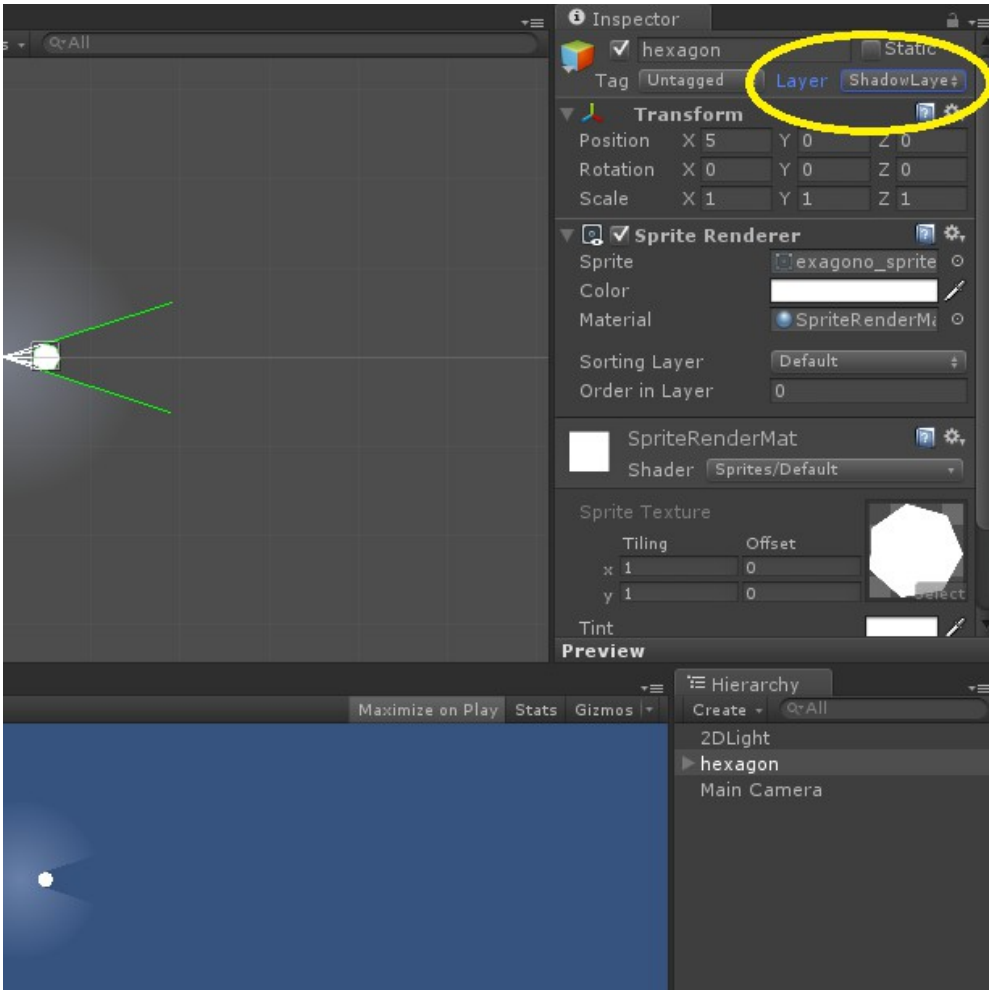
You Already have a First Game Scene with a 2D Lights PRO Effect and looks really cool.
You can combine in many ways this components, you can add Multiple Lights and Casters a lots.
Because this 2D Light PRO package is fully optimized for fast rendering and has a low cost in process to CPU.

*** – CASTERS: Building my Own – ***

If you want build your own GameObject that Caster Shadow, you may consider the following conditions:

For example, i'll create a cube

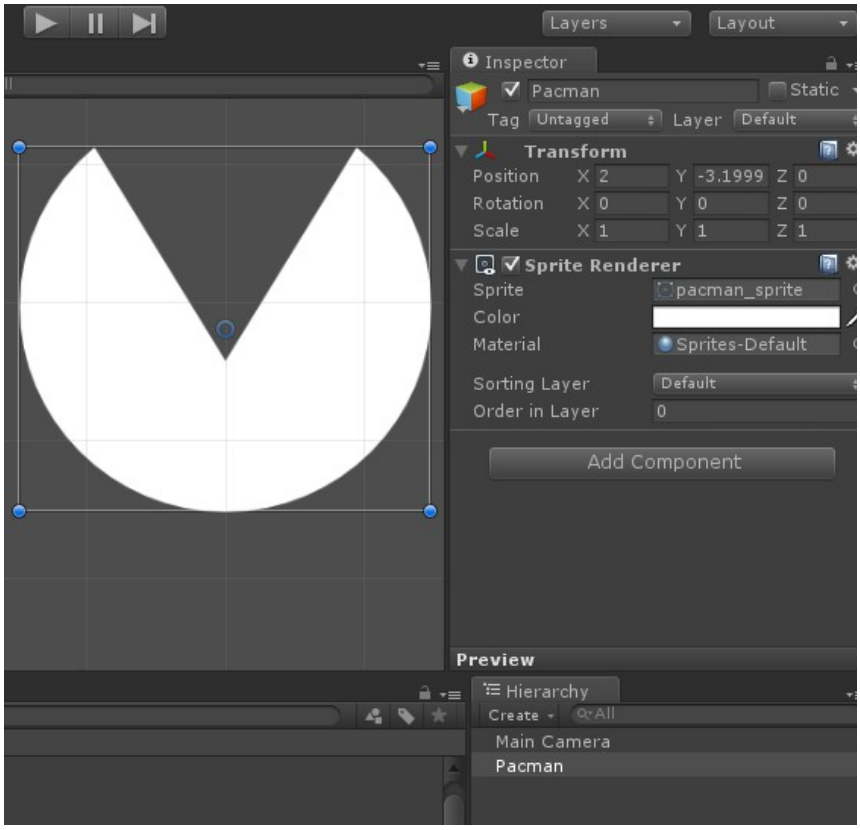
1) Create a Cube Game Object (GameObject → Create other → Cube)
2) Delete any collider, BoxCollider in this case.
3) Create a empty game object and add a PolygonCollider2D component
4) Do it child of main Game Object, this allow work with vertices more freely.
5) **Important to set Z position to 0 (Z=0)** main game object and also child collider2D. Why? Because need alignment between the Light and the Collider for the ray clash against him.
6) Check the Layer for interact with 2DLight be the desired layer (Default = ShadowLayer)



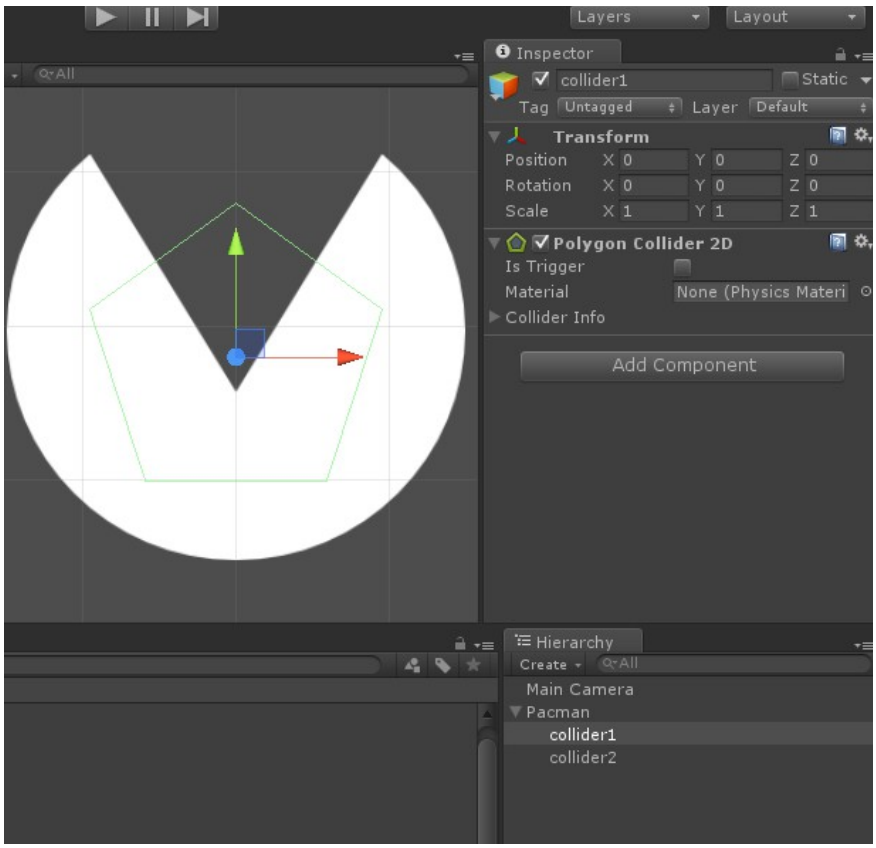
*** – CONCAVE COLLIDERS: Pacman case – ***

If you want insert in a scene a CONCAVE shape or sprite, the collider must be a CONVEX. e.g:

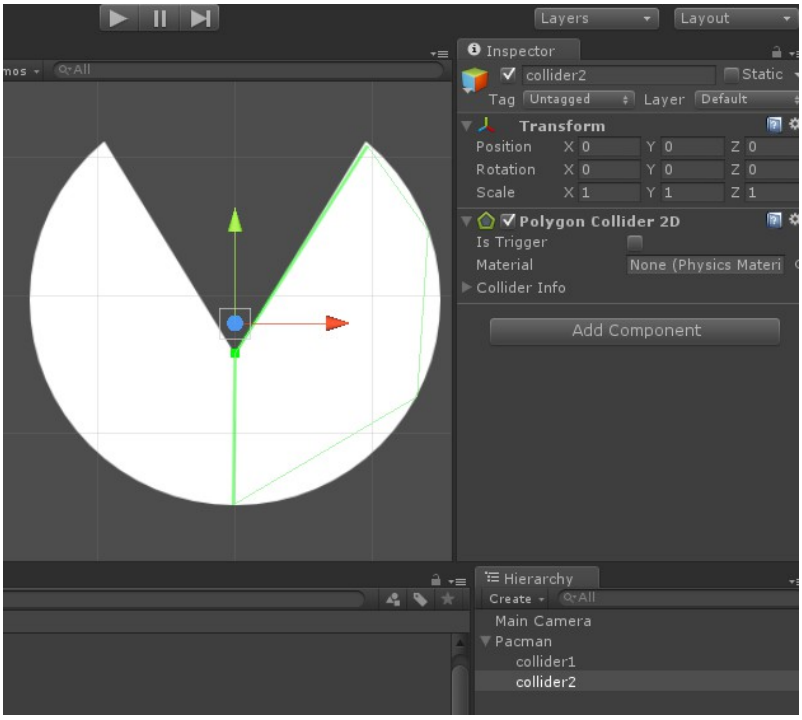
1) Create a new sprite (GameObject → Create other → Sprite). Call it “pacman”
2) Drag Pacman_sprite from textures to Sprite property field.



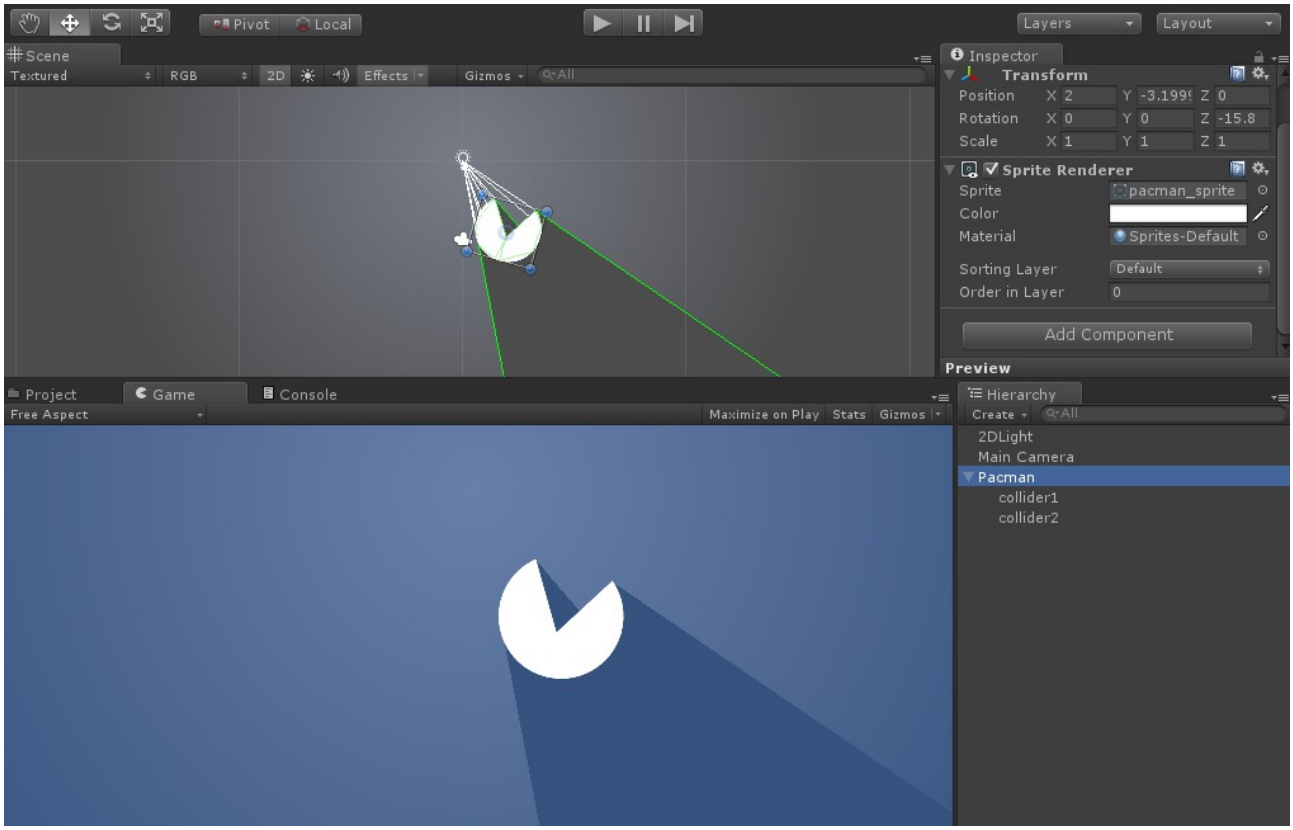
3) Create a Empty GameObject, call it “collider1” and doit child of “pacman”.
4) Clone the “collider1” Game Object and put “collider2” as name.
5) Add PolygonCollider2D in both childs.



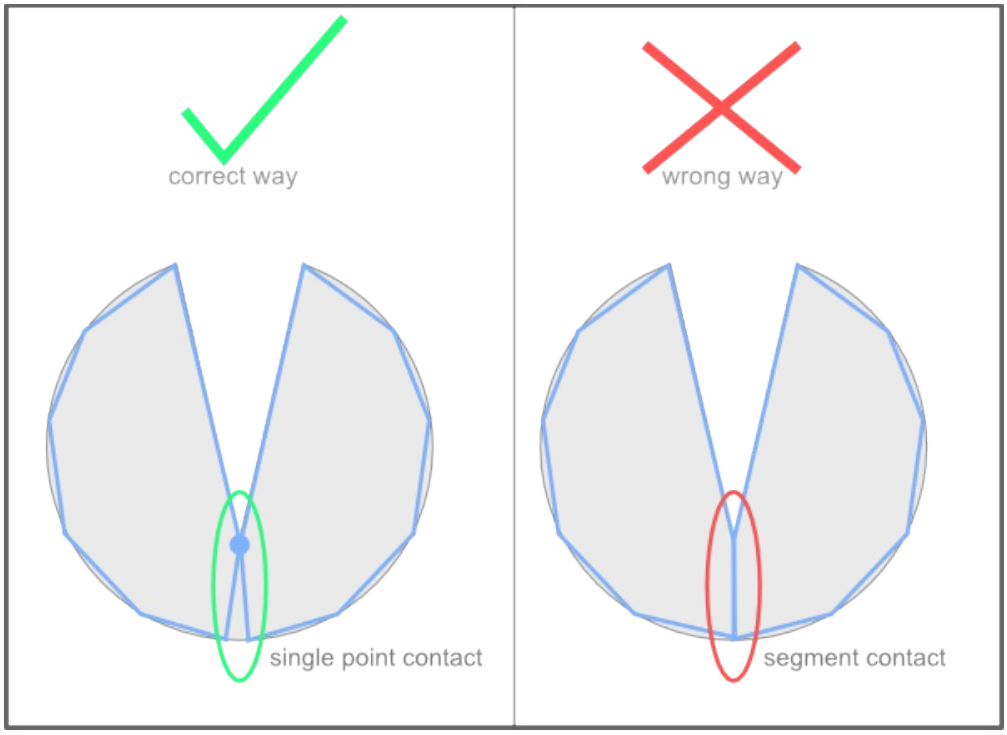
6) Why child? Well, for avoid that Unity generate automatic collider for us.
7) "collider1" will for the first half of pacman, and "collider2" for the second.
8) Use the keyoard keys for edit PolygonCollider2D points.
(http://docs.unity3d.com/Manual/class-PolygonCollider2D.html)



9) We should get something like this:



10) Here is the trick. Now, to prevent an strange behavior or light need only a one contact point between “collider1” and “collider2” like the image:

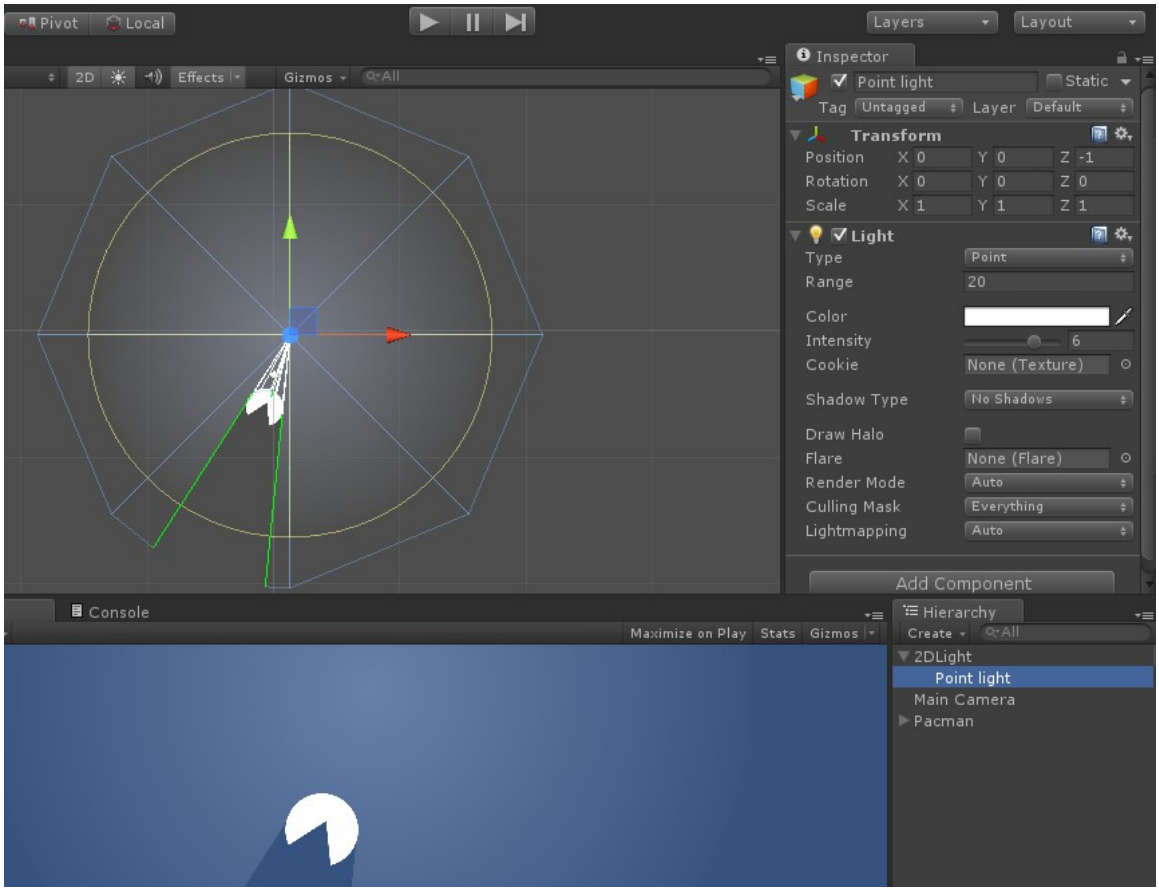


11) And your custom Concave Collider is Ready!

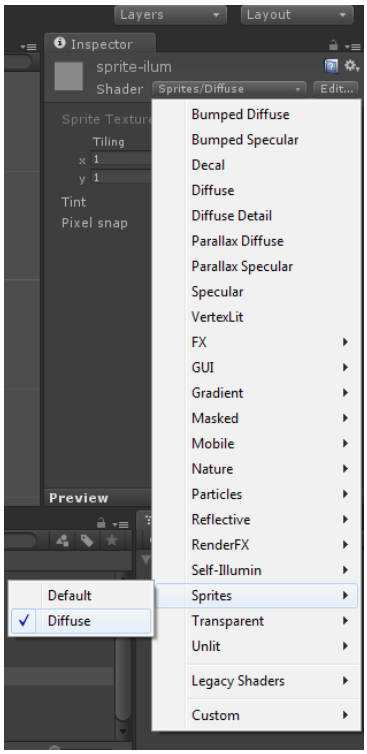
*** – ROOKIE IMPLEMENTATION– ***
2DLight with sprite ilumination

Following the example of Pacman, we will apply a lighting effect surface on the sprite. As shown in the image. (default ilumination on the left and dynamically illuminated on the right)

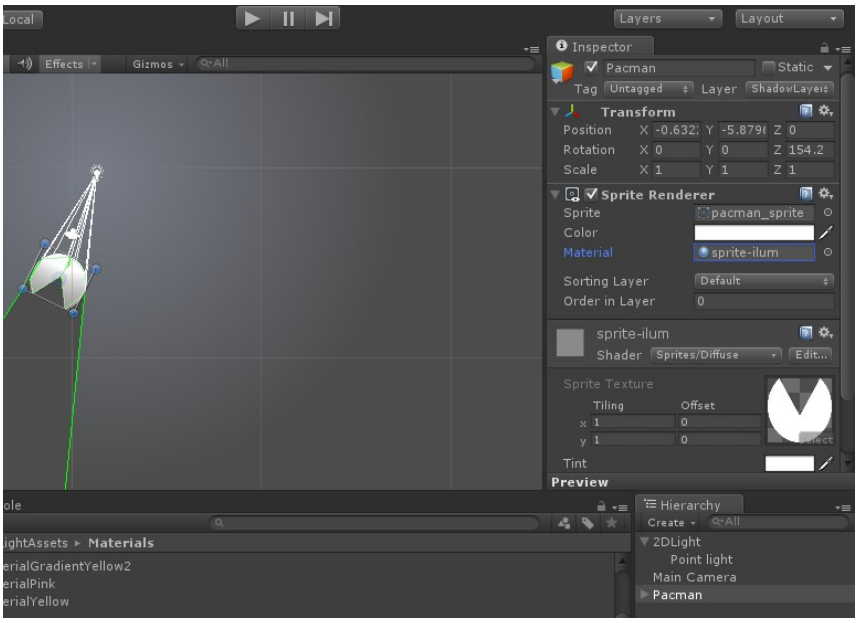
1) Add a Point Light(game Object → Create other → Point Light)
2) Put inside a 2DLight game object and set the local position to (0,0,-1)
3) Light Point Range property set equal 2DLight radius = 20 in my case.

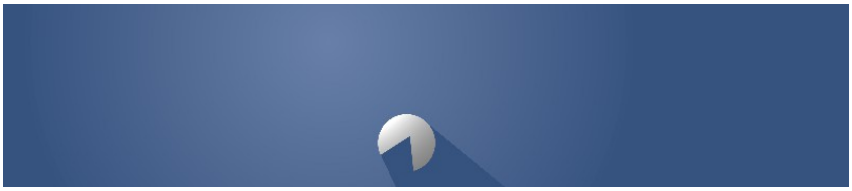


4) Set rest of parameters as you like. Intensity, I used 6.
5) Create a new Material (in Materials Folder, right click, new Material), call it “sprite-ilum”
6) Set shader material to Sprite → diffuse



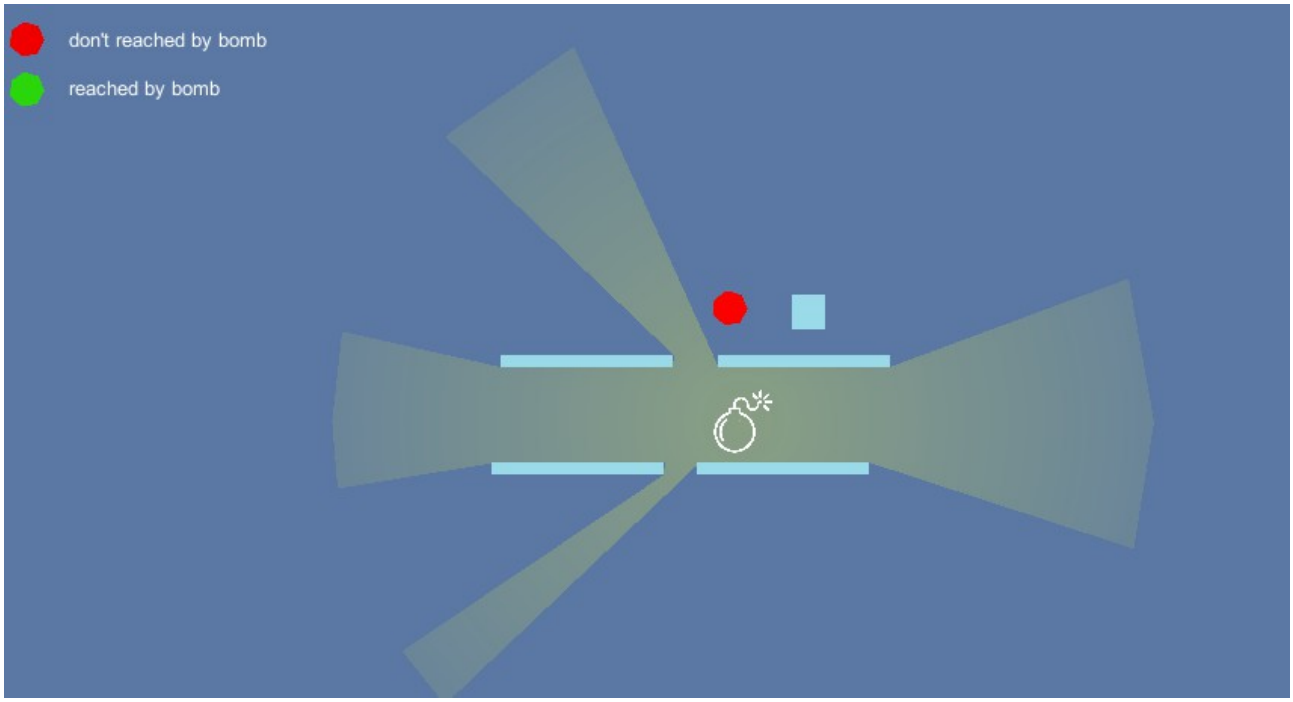
7) Now, drag sprite-ilum Material to Pacman game object, and the Ilumination will start work.





*** – INTERMEDIATE IMPLEMENTATION– ***

3.Cover Explosion(shockwave)



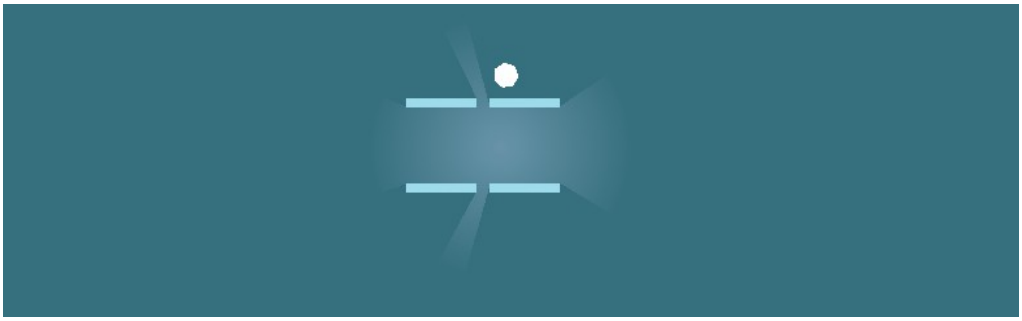
Open the scene 3.CoverExplosion(shockwave) .This scene serves as a bomb blast can simulate an explosion and know that elements have been reached and affected by the explosion..

For the direct implementation, you can use delegates and know that items were passed as parameter. These elements are the gameobject that light has reached.

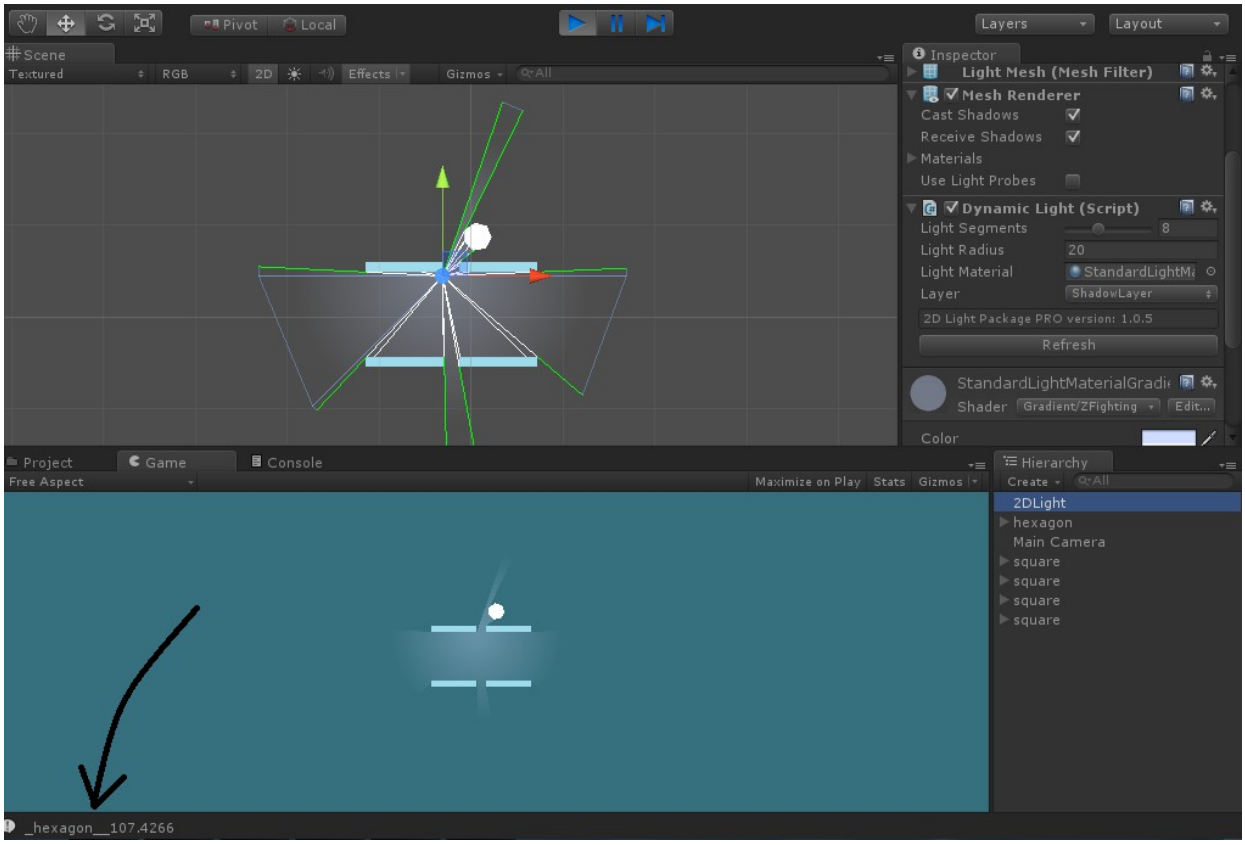
Step by step:

1) Create a new scene.
2) Setup your Camera in ortho size=30.
3) Add a Light2D Gradient and set the radius = 20
4) Put 4 square Casters in the scene forming walls.
5) Add a Hexagon, this caster will be our character.
6) Drag and drop ReachedEventManager script to Hegaxon gameobject.

This is what you get:

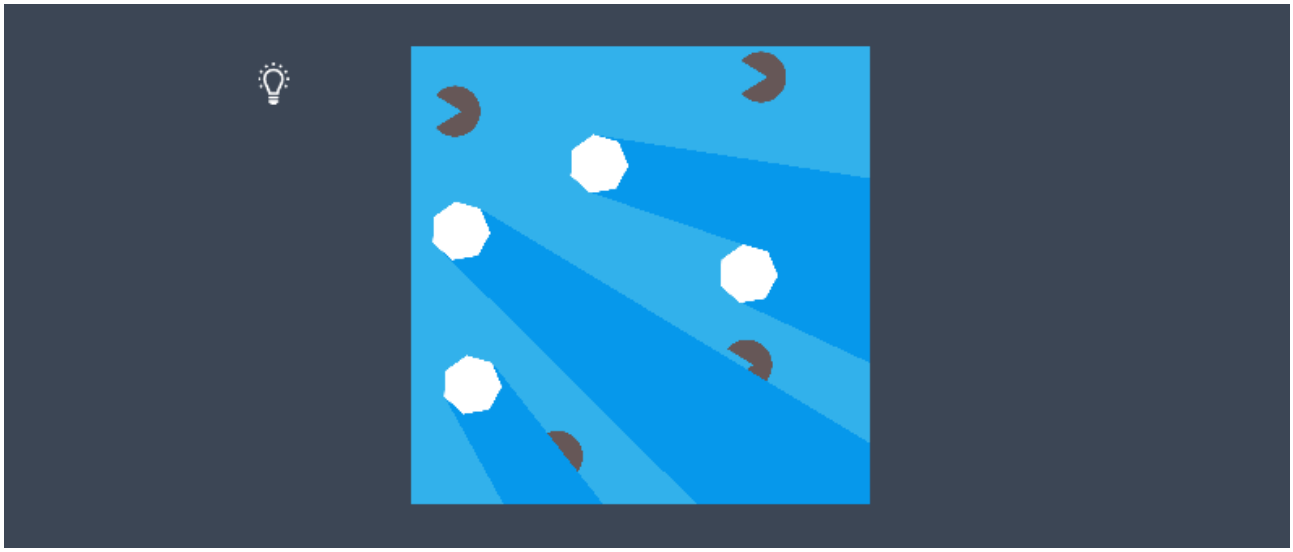


7) Now run the scene and look the console. Will display a message when hexagon is reached by a branch of light.



*** – INTERMEDIATE IMPLEMENTATION 2– ***

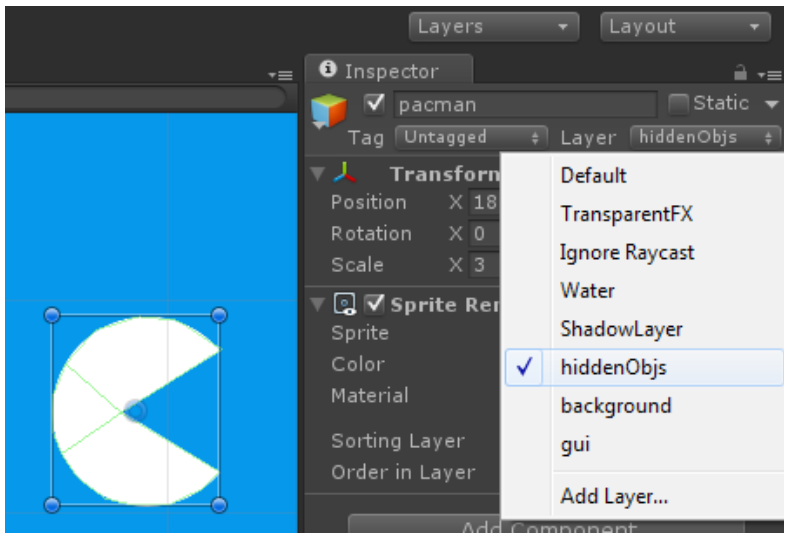
7.VisibleMinimal



In this example, we will use the light mesh to became visible the hidden objects (pacmans).

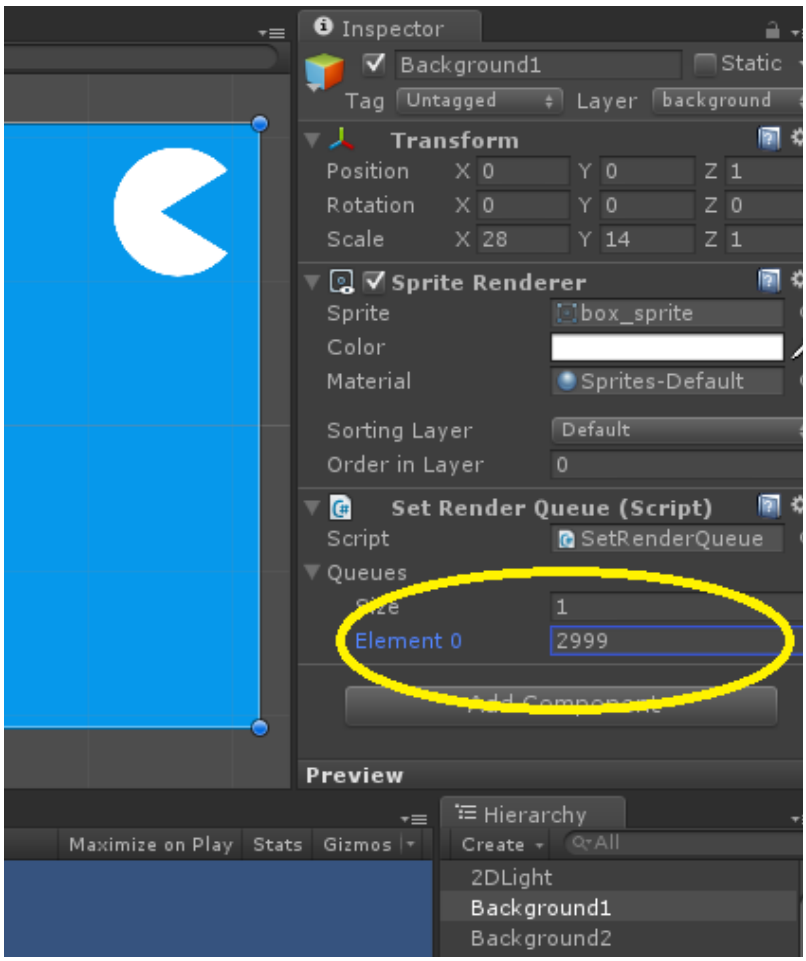
Steps:

1) Create a new scene.
2) Set Camera to ortho in size 30.
3) Add 2 box_sprite in the scene from "/Textures".
4) Set both same position to (0,0,1), same scale and same rotation.
5) Named "Background1" and "Background2"
6) To "Background2" change the color material to RGB(8,173,255)
7) Add 3 caster as Pacman.
8) Create a new layer called "hiddenObject" and put all pacmans inside this layer (the pacmans doesn't cast shadows).

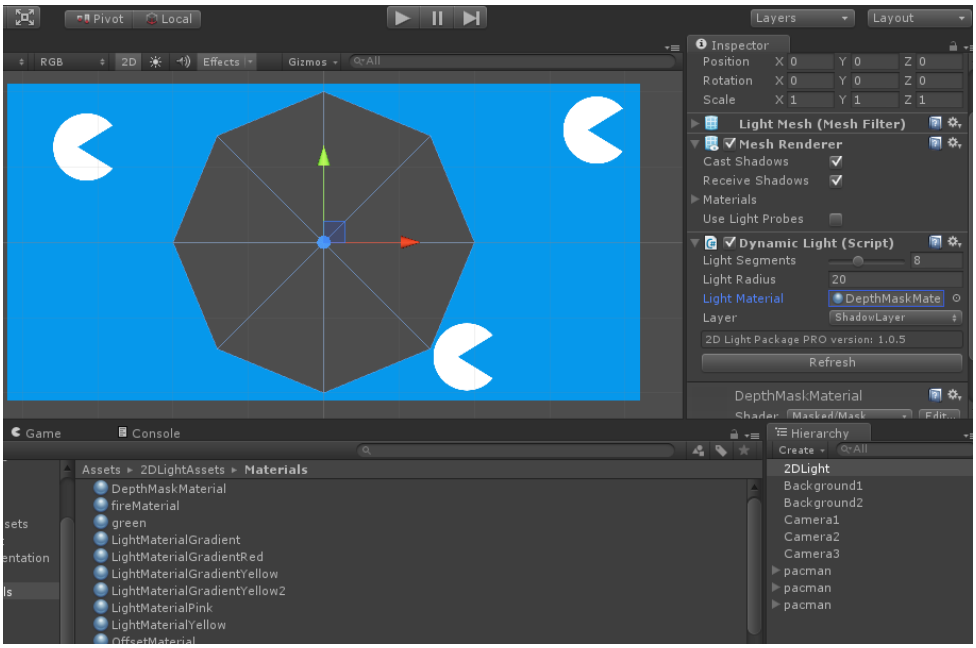


9) Create another new layer called “background1” and put Background1 gameObject within.

10) We need the “background1” and “background2” to be render in last position, for that, drag and drop the script “SetRenderQueue” to both gameObject and only set the Background1 → SetRenderQueue → Element0 = 2999.



11) Now add the Light2D, set radius to 30, and set the Light Material as “DepthMaskMaterial” from Materials folder.



12) CAMERAS !, uppercase because is the most important step:

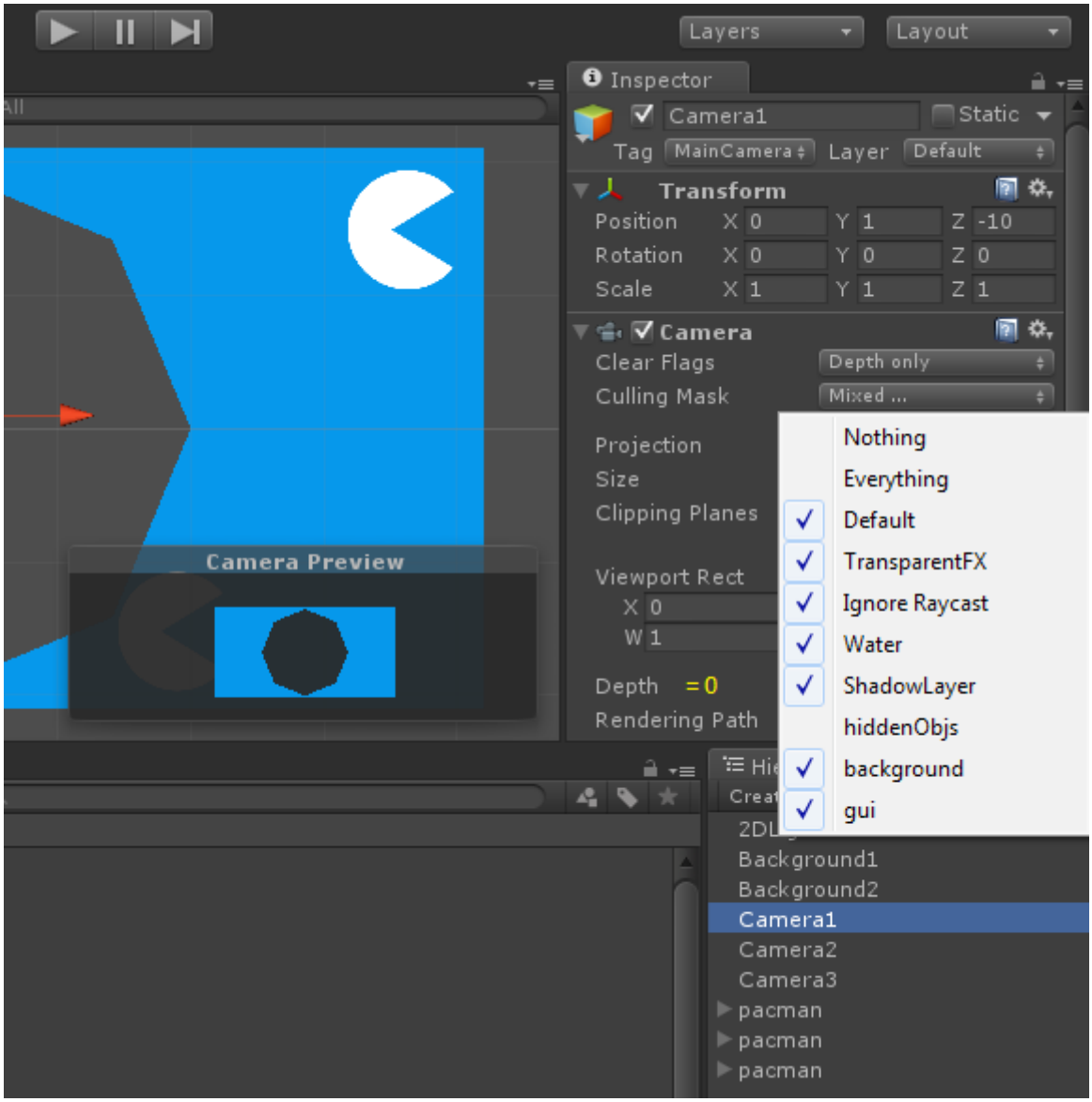
For this scene we need 3 cameras, setted in same position, same size, same rotation.

Camera1.The default Camera (already created) must set:

Clear Flags: Depth Only

Culling Mask: Everything except “hiddenObject” layer

Depth: 0

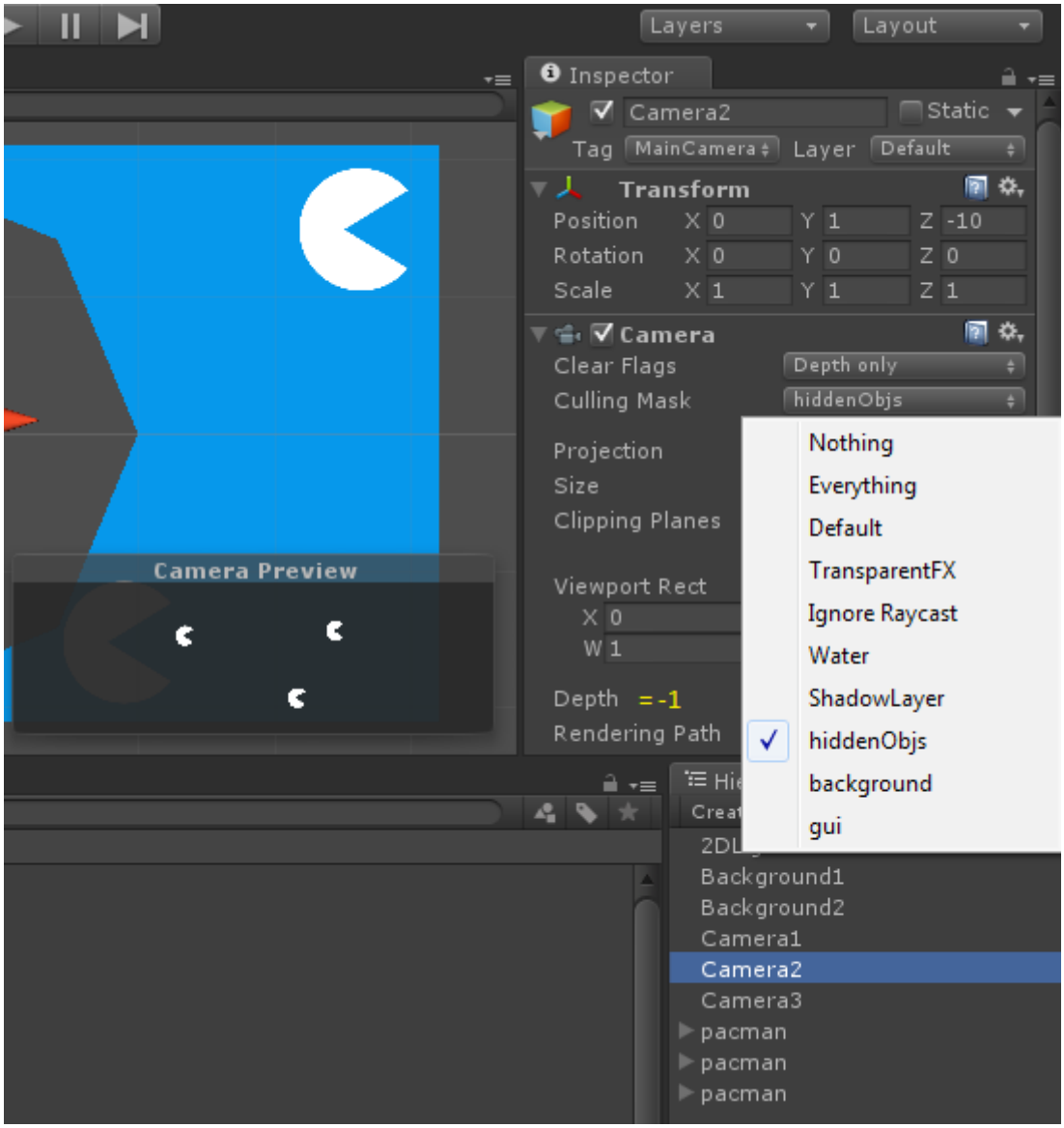


The Camera2 (Cloned from Default Camera) must set:

Clear Flags: Depth Only

Culling Mask: only “hiddenObject” layer

Depth: -1

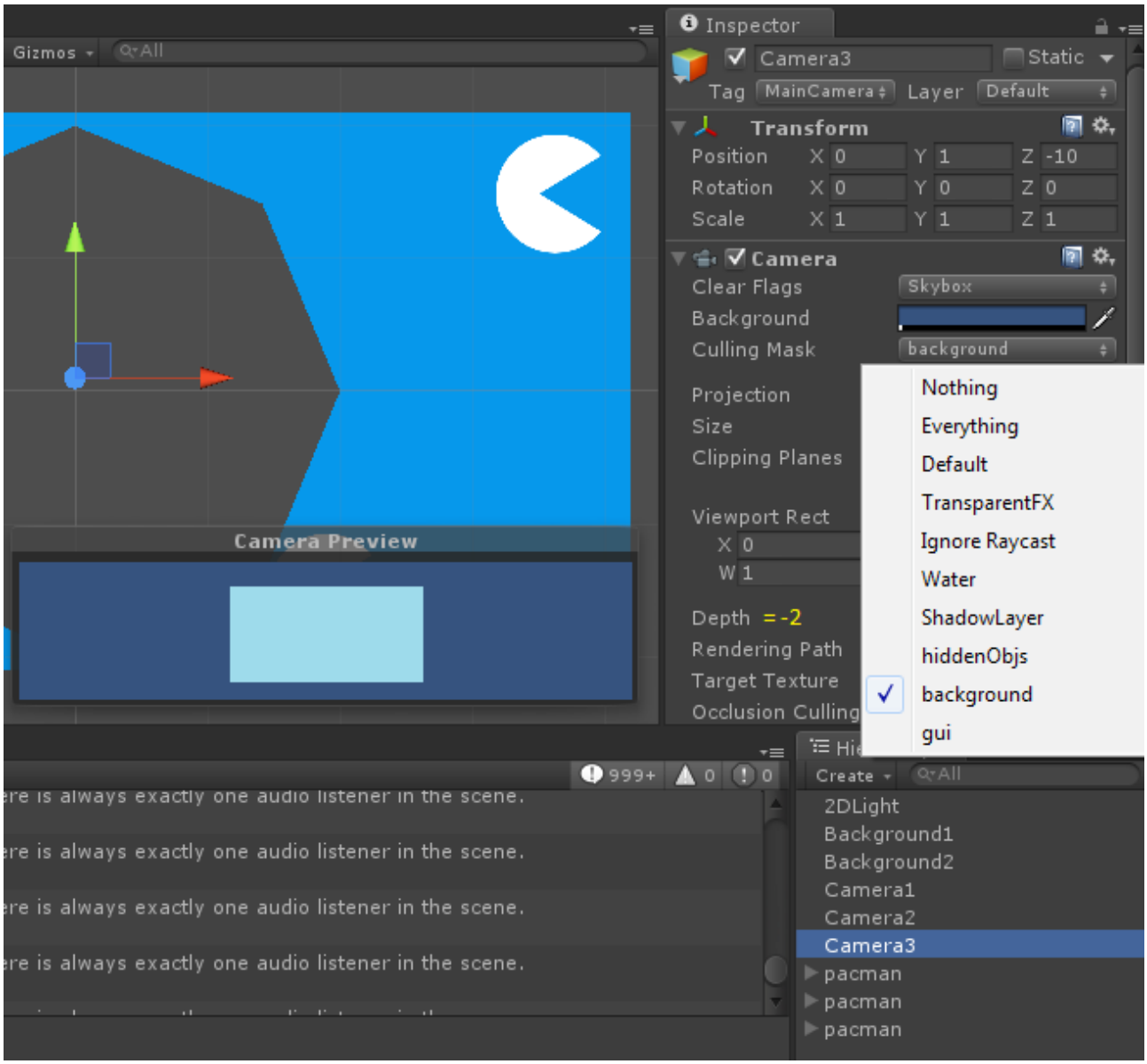


The Camera3 (Cloned from Camera2) must set:

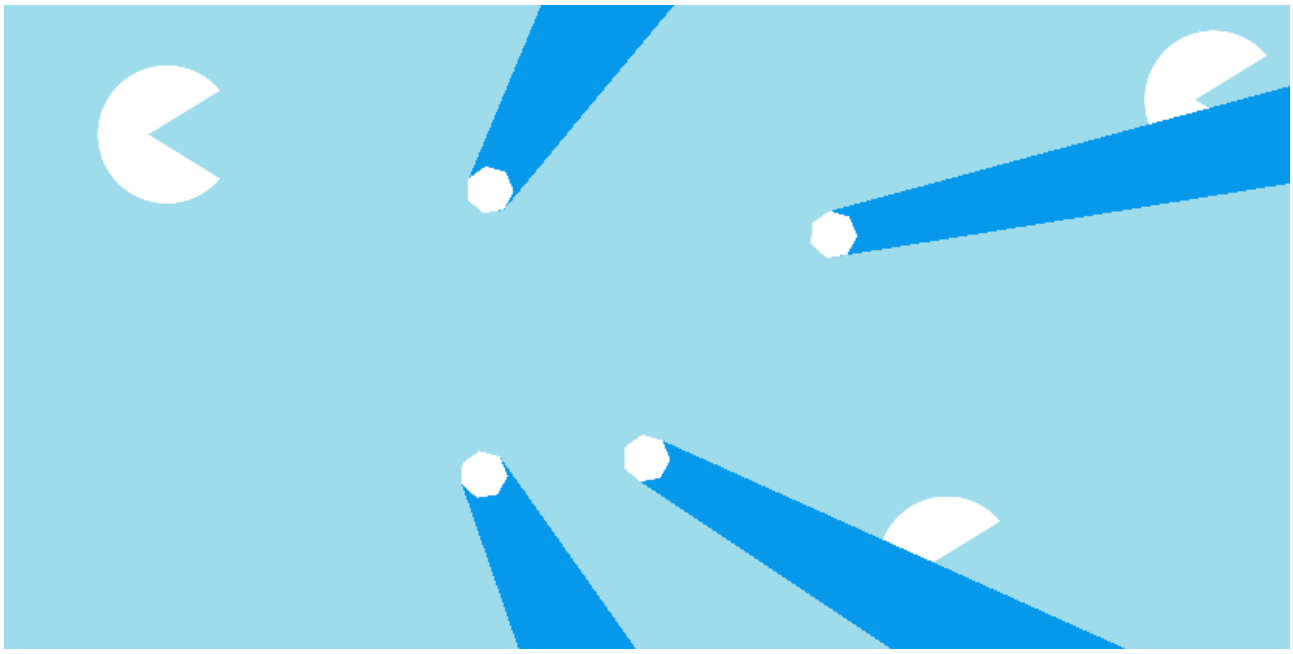
Clear Flags: Skybox

Culling Mask: only “background1” layer

Depth: -2

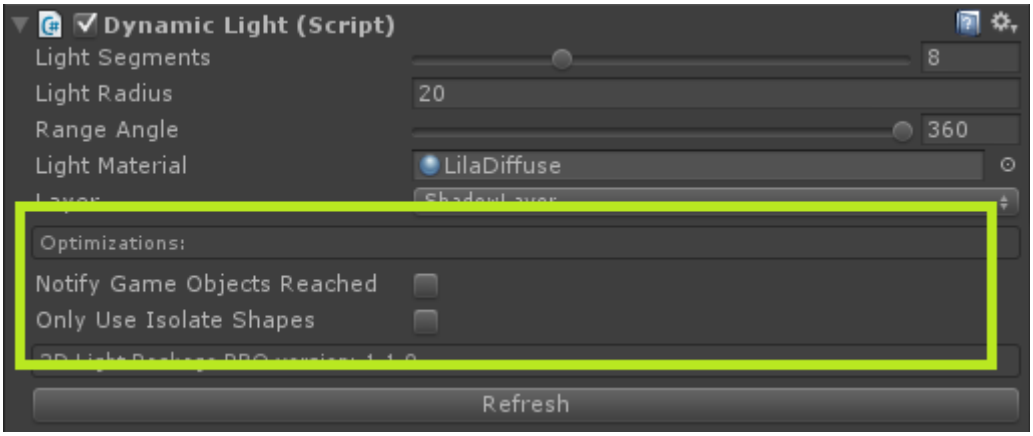


13) For conclude, add some Hexagon Caster, and running the scene you will get :



*** – OPTIMIZATIONS – ***

from **version 1.1.0** you will find the section optimization with the following fields:



by default both fields are deactive, but you must know what each does for use in certain situation.

Notify Game Object Reached:

this allow subscribe to any listen event so as to detect colliding object game.
When is ACTIVE, ***THE PERFORMANCE IS REDUCED***, depending on your cpu will lose between 2 -3 fps for each light
When do I use it? In scenes like 3.CoverExplosion(shockwave).

INTELLIDER CONVEX:

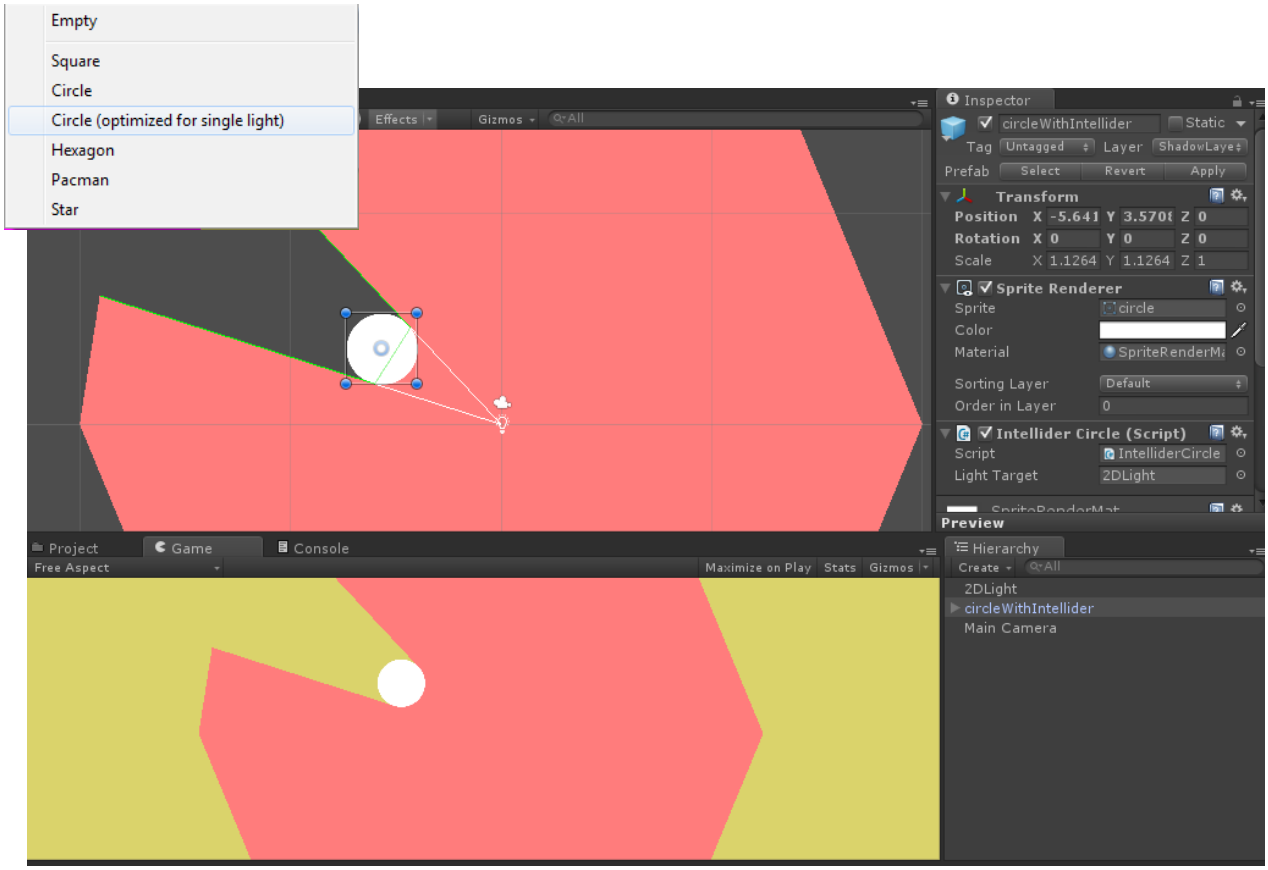
ALREADY INTEGRATED INSIDE 2DLIGHT SYSTEM . This mean that if this field is enabled, you take care to use only convex and separate colliders between each other for prevent weird results.
When is ACTIVE, ***THE PERFORMANCE IS AUGMENTED***
How does it work?: choose the 2 most convenient vertex for each shape and uses.
Activate not recommended

Disable **Enable**

*** – OPTIMIZATIONS: Intellider Circle – ***

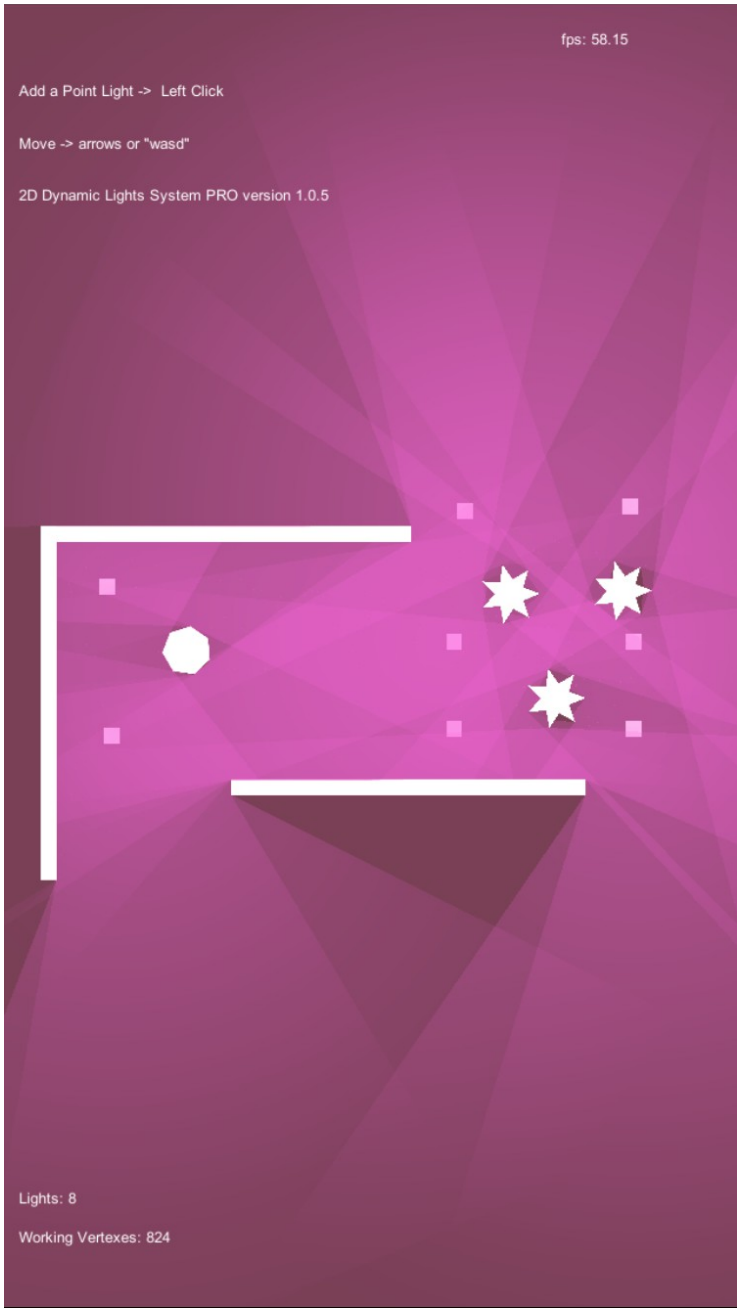
Intellider is a class that allow optimize circle colliders whe is used under a single light.

Operation consists in pointing to the light a collider, with a minimum number of vertices.
wherever the object move or rotate, always project the correct shade.
Available in Casters menu



*** – PERFORMANCE – ***

The following scene is running in iPhone 6 with iOS 8, i added 8 Gradients 2D Lights, 3 Stars and 1 Hexagon:



And could be add more, but is ok for this demostration and keep a High FPS.

*** – LIMITATIONS – ***

Since it is designed for a 2D Enviroment and for reach maximum performance must sacrifice some things. This limitations are:

*)Only can move in X and Y axes. (Z depth movement produce unstable results)

*)~~Can't allow rotation of 2DLight gameObject itself.~~ NOW ALLOW ROTATIONS IN Z AXIS

*)Can't allow scale of 2DLight gameObject itself.

*)Only works in X and Y axes. Don't allow exchange between X → Z or Y → Z

*) Not have a real Concave compatibility. Can't detect Collider segment intersection. BUT HAVE A TRICK FOR GET IT :) see 8.

*** – Acknowledgements – ***

- Unity Forums and Thread: http://forum.unity3d.com/threads/2d-dynamic-shadows.165283/
- Marrt Dual Depth Camera System : http://forum.unity3d.com/threads/skynox.124019/#post-1007954
- Jessy Catterwauld : https://www.youtube.com/channel/UCqkCSwxg2LUkhM0-7M79c9Q
- Grit Schuster : https://www.facebook.com/grit.kit (rotation & light angle approach)
- Clopay : Intellider Circle approach. http://forum.unity3d.com/threads/2d-dynamic-shadows.165283/page-2#post-1929996

Thanks for read this documentation. I hope that enjoy this package.

For contact, tips, question, support or even say hello, please doing at:

info@martinysa.com

www.martinysa.com

twitter: @martinysa