Pinball Creator

CREATE YOUR PINBALL

Asset documentation Part 2.

Table of contents :

Features :

Other :

2 Setup table.

In this section we will see how to setup a new table that could be easily convert to mobile version or desktop version :

First Choose a platform :

1.0 Setup or convert for PC,Mac,Linux.
1b.0 Setup or convert for Mobile.

Then for both desktop and Mobile :

1.2 Prepare your materials	link
1.3 Setup the material on your scene	link
1.4 How to change texture for material on both mobile desktop	link
1.5 Models for Desktop and Mobile version	link
1.6 Setup the model folder on your scene	link
1.7 Switch between desktop models and mobile models	link
1.8 Setup Lighting for Desktop version	link
1.9 Setup Lighting for Mobile version	link
1.10 Setup the ball for mobile version and desktop version	link

Step by step example:

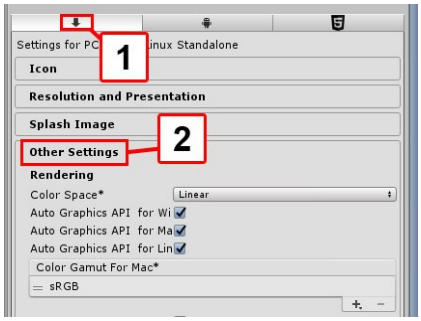
1.0 Setup or convert for PC,Mac,Linux.

1.0.a Open Build Setting tab (file->build Settings)

1.0.b Switch platform to Pc,Mac,Linux if needed (Pic 1)
Press Button Switch Platform (Pic 2)

1.0.c Check if you are on Linear Mode.
Open Player Settings (Edit->ProjectSettings->Player)

- Press on desktop icon button (spot 1)
- Press on Other settings text to open Bar (spot 2)
- Select Linear Mode in ColorSpace



1.0.d Select quality : Choose Good or higher
- Open Player Settings (Edit->ProjectSettings->Quality)
- Clic on good (pic 1). Gray subtitle mean that it is the visualization quality inside Unity Editor.
- Clic on the small triangle and select Good (pic 2). Green tick mean that it is the quality for build export.

1b.0 Setup or convert for Mobile.

1b.0.a Open Build Setting tab (file->build Settings)

1b.0.b Select the platform (Android,iOS...) needed (Pic 1) Example demo scene is create for Android.

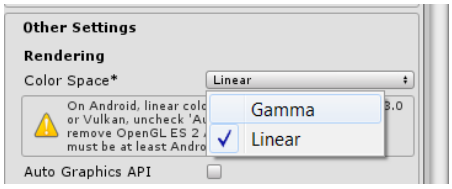
Press Button Switch Platform (Pic 2)

1.0.c Check if you are on Gamma Mode.

Open Player Settings (Edit->ProjectSettings->Player)

Press on Other settings text to open Bar

- Select Gamma Mode in ColorSpace



1b.0.d Open Player Settings (Edit->ProjectSettings->Quality)

- Clic on fastest (pic 1). Gray subtitle mean that it is the visualization Inside Unity Editor.
- Clic on the small triangle and select fastest (pic 2). You will see a green cross on fastest.

It is the setting you have on your build export for Desktop.

1.2 Prepare your materials.

Open scene Start_Pinball. (Create a copy of this scene for safety)
(Pinball Creator->Assets->Scenes-> Start_Pinball)

1.2.a Duplicate the folder Global (Pinball Creator->Assets->Materials-> Global). This folder contain all the basic materials used on Pinball Creator.

A new folder is created named Global 1. Inside this folder there are 2 folders. One that contain all the material for desktop (pic 1) and the other one contain materials for mobile (pic 2).

1.3 Setup the material on your scene

1.3.a Select the gameObject MobileToDesktop on the Hierarchy.

1.3.b Choose the path for the material.

- First write the path inside Folder Desktop Material and Folder Mobile Materials params.

We want to use the materials inside folder Global 1 :
The path for Folder Desktop Material is :
Assets/Pinball Creator/Assets/Materials/Global 1/Materials_Desktop/

The path for Folder Mobile Materials is :
Assets/Pinball Creator/Assets/Materials/Global 1/Materials_Mobile/

- Then Clic on button Validation to update the folder path(pic 2)

If you want to use desktop materials :
Press button MobileToDesktop. (Pic 1)

If you want to use mobile materials :
Press button DesktopToMobile. (Pic 2)

1.3.c Add your own materials :

Info : For your own materials :
Put the desktop material version inside the folder you use for Desktop materials
Put the mobile material version inside the folder you use for Mobile materials

It will be easy to switch your materials between desktop and mobile version

1.4 How to change texture for material on both mobile desktop

Mobile use a material. Desktop use an other material.
So when you want to use new texture, you need to add this texture on both mobile and desktop materials.

To do this follow the procedure bellow :

1.4.a For Example : Select the material Playfield_Mat_11 inside folder Global 1->Materials_Desktop->Playfield_Mat_11 . (Pic 1 next page)

1.4.b Clic on the little circle beside albedo texture . (Pic 2 next page)
A new window appear : Select your texture. (Pic 3 next page)

1.4.c Select the material Playfield_Mat_11 inside folder Global 1->Materials_Mobile->Playfield_Mat_11 .

1.4.d Clic on the little circle beside albedo texture .
A new window appear : Select your texture. (Pic 3 next page)

1.5 Models for Desktop and Mobile version

1.5.a How it works :

Models use same system as material. There is a folder for your Desktop models and a folder for mobile models.

Put your desktop models on folder : Assets/Pinball Creator/Assets/Models/Models_Desktop/
Put your mobile models on folder : Assets/Pinball Creator/Assets/Models/Models_Mobile/

With this system you have a low res mobel for mobile and a high res for desktop

1.5.b Add your own Models

Info : For your own models :

Put the high res version inside the folder for Desktop models

Put the low res version inside the folder for Mobile models

It will be easy to switch your models between desktop and mobile version

1.6 Setup the model folder on your scene

1.6.a Select the gameObject MobileToDesktop on the Hierarchy.

1.6.b Model Path.

By default the path is ok. You don't have to change it. Model path is used only for advanced techniques.

To change the path : Write the path inside Folder Desktop Models and Folder Mobile Models params :

The path for Folder Desktop Models is : Assets/Pinball Creator/Assets/Models/Models_Desktop/

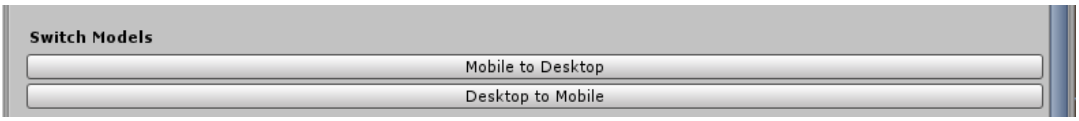
The path for Folder Mobile Models is : Assets/Pinball Creator/Assets/Models/Models_Mobile/

1.7 Switch between desktop models and mobile models

1.7.a Select the gameObject MobileToDesktop on the Hierarchy.

If you want to use desktop models :
Press button Mobile to Desktop. (Pic 1)

If you want to use mobile models :
Press button Desktop to Mobile. (Pic 2)



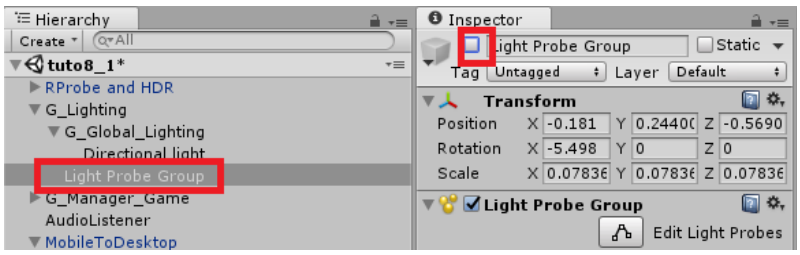
1.8 Setup Lighting for Desktop version

(If you are creating a table for mobile go to next section 1.9 Setup Lighting for Mobile version)

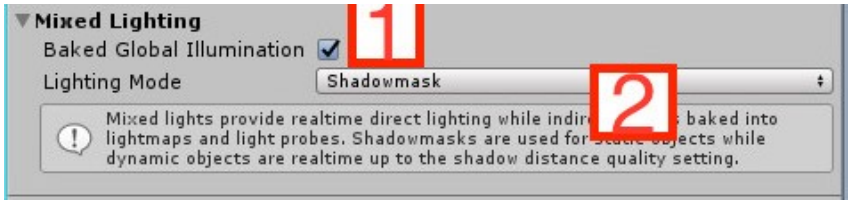
1.8.a Drag and drop prefab RProbe and HDR on the root of Hierarchy (Assets/Pinball Creator/Assets/Prefabs/Grp_Leds/RProbe and HDR).

Note that if you use Start_Pinball this prefab is already on the Hierarchy.

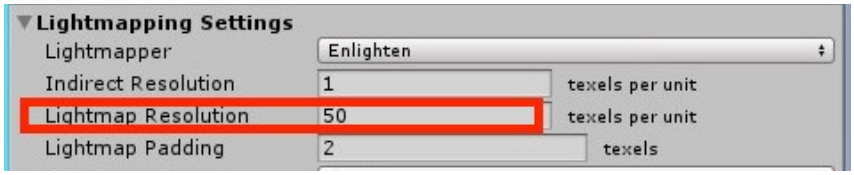
1.8.b Deactivate gameObject Light Probe Group (Hierachy->G_Lighting->Light Probe Group)



1.8.c Open Lighting tab (Window -> Rendering -> Lightings Settings)
Check box Bake Global Illumination(Pic 1)
Choose Ligthing Mode : Shadowmask (Pic 2)

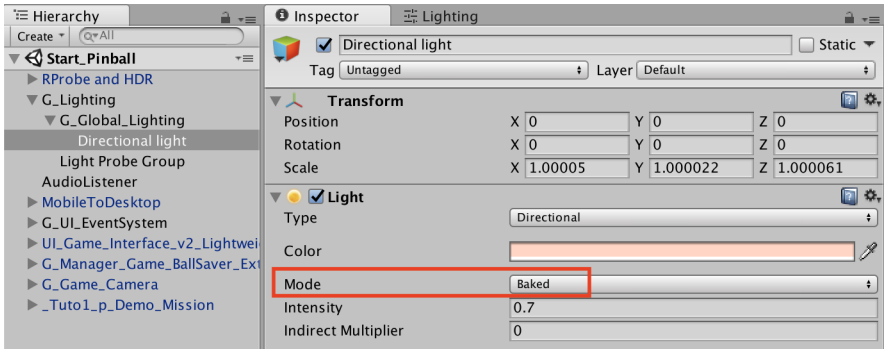


Check if Ligthmap resolution is equal to 50.
Tips : 15 is a good value when you are working because it faster to calculate.

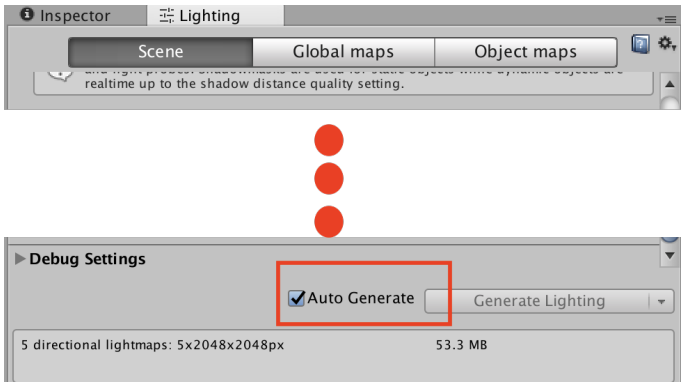


Select Directional .

1.8.d Select the gameObject Directional light on the hierarchy.
Choose Mode = Baked



1.8.e activate the lightmap check box Auto on Lighting tab



1.8.f IMPORTANT : Intensity could be a little bit different when you switch between desktop to mobile.

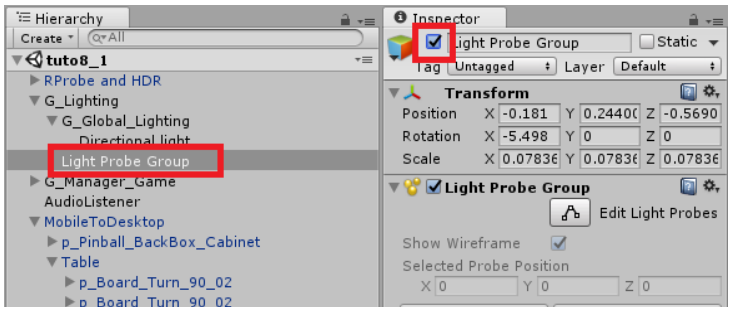
Tweak the light intensity parameter of your :

- Point lights.
- Directional Light (G_Lighting->G_Global_Lighting->Directional light)

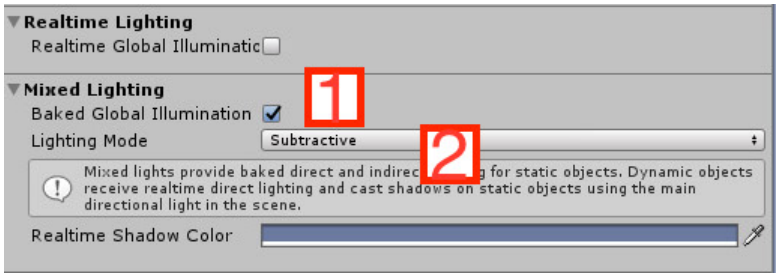
1.9 Setup Lighting for Mobile version

1.9.a Delete prefab RProbe and HDR if you find it on the root of Hierarchy

1.9.b Activate gameObject Light Probe Group (Hierachy->G_Lighting->Light Probe Group)

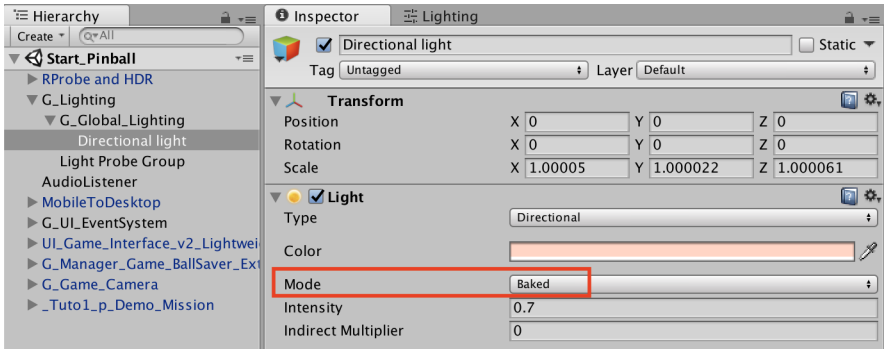


1.9.c Open Lighting tab (Window -> Rendering -> Lightings Settings)
Check box Bake Global Illumination(Pic 1)
Choose Ligthing Mode : Substractive (Pic 2)

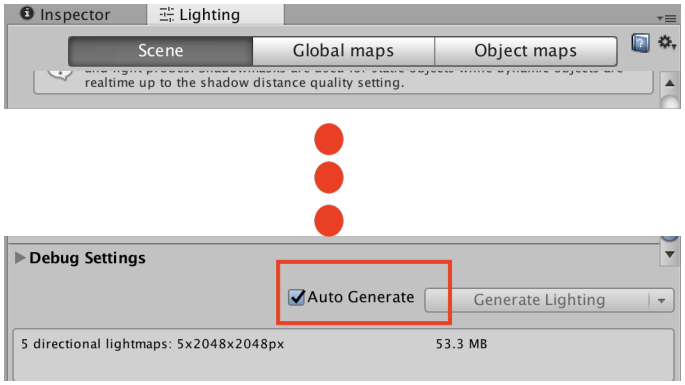


Select None Directional .

1.9.d Select the gameObject Directional light on the hierarchy.
Choose Mode = Baked



1.9.e activate the lightmap check box Auto on Lighting tab



1.9.f IMPORTANT : Intensity could be a little bit different when you switch between desktop to mobile.
Tweak the light intensity parameter of your :
- Point lights.
- Directional Light (G_Lighting -> G_Global_Lighting -> Directional light)

1.10 Setup the ball for mobile version and desktop version

We've create a specific ball for mobile version and an other one for desktop.

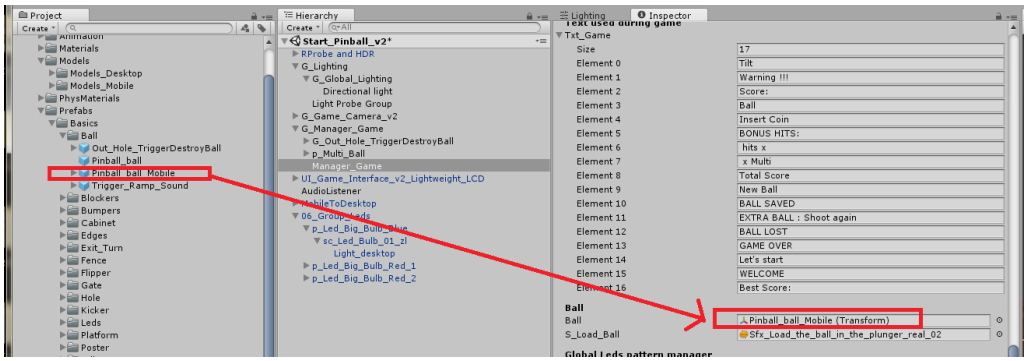
Mobile Ball : (pic 1 next page)
Clic on Manager_Game gameObject on the Inspector (G_Manager_Game_BallSaver_ExtraBall_ leds ready-> Manager_Game)

Inside Ball parameter on script Manager_Game.js drag and drop prefab Pinball_ball_Mobile (Assets->Prefabs->Basic->Ball->Pinball_ball_Mobile)

Desktop Ball :

Clic on Manager_Game gameObject on the Inspector (G_Manager_Game_BallSaver_ExtraBall_ leds ready-> Manager_Game)

Inside Ball parameter on script Manager_Game.js drag and drop prefab Pinball_ball (Assets->Prefabs->Basic->Ball->Pinball_ball)



3 Optimization (usefull for mobile and desktop) :

3.1 Introduction :

Desktop : If your pinball tables are only for desktop, optimization is optional. But if you use our optimization suggestions you could target older desktop and add a lot more gameObjects on your scene.

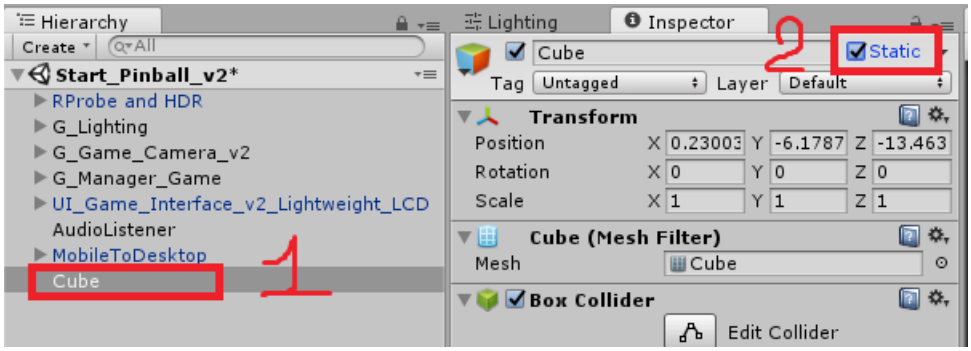
Mobile : For mobile this section is essential. If you don't use these optimizations your FPS will be slow. Combine Meshes is very usefull to increase the FPS of your game about +10 to 15 FPS.
Octopus Island Table run on Galaxy tab 3 at 50-60 FPS.

3.2 Memento Static and non static gameObjects

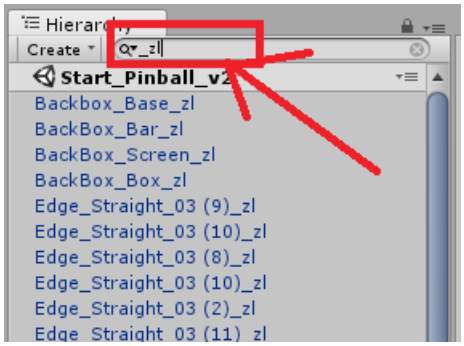
3.2.a Static gameObject are interesting because Unity automaticaly Batch them.
Pros : It increase the performance of the game.
Cons : Static gameObject can't move or rotate at runtime.

Example Static gameObject : Ramp, pipe, stationary target, Room ...
Example Non Static : ball, flipper, drop targets, animated toys, switch, spinner, plunger ...

3.2.b To make a gameObject static on your scene :
Select the gameObject on Hierarchy (Pic 1)
Check the box Static on the inspector. (Pic 2)

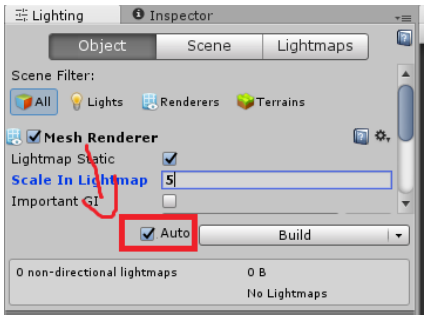


3.2.c To easily find static gameObjects on Pinball Creator :
Search all the gameobject with tag _zl inside there gameObject name :
On Hierarchy search : _zl



3.3 Memento Lightmap

Important : To activate the lightmap check box Auto on Lighting tab



Using Lightmap is usefull for optimize your scene. Desktop or mobile device don't have to calculate shadow and global Illumination (GI) at runtime.

Lightmap is used to pre-calculate shadow and GI.
Lightmap only calculate Static objects.

More info about setup lightmap parameters here :
Setup Lighting for Desktop version
Setup Lighting for Mobile version

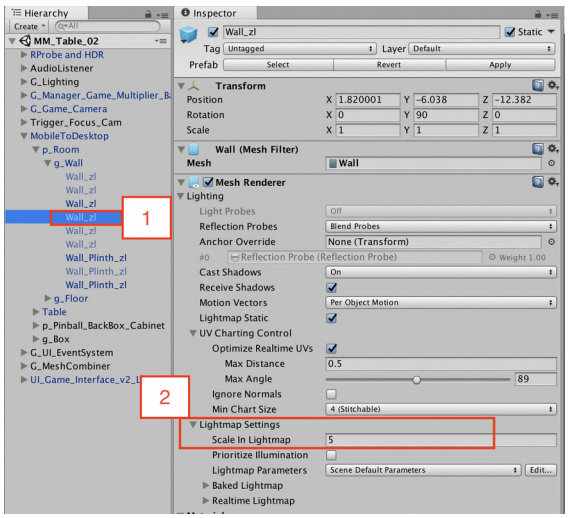
Info : On Unity 5.5 versions if you have strange results try to uncheck Auto button and press instead button Build

3.4 Optimize Object scale in Lightmap

If a static object is big, it could reach the max lightmap atlas size.
If an object reach the max lightmap size Unity could be slow to calculate the lightmap and create strange result.

How to check :
Select a static object (Pic 1)

On the Inspector Tab if this message appear : ''Object's size in lightmap has reach the max atlas size'' decrease Scale In Lightmap value until the message disappear. (Pic 2)



Info : Scale In Lightmap = 10 is good starting point in most cases.

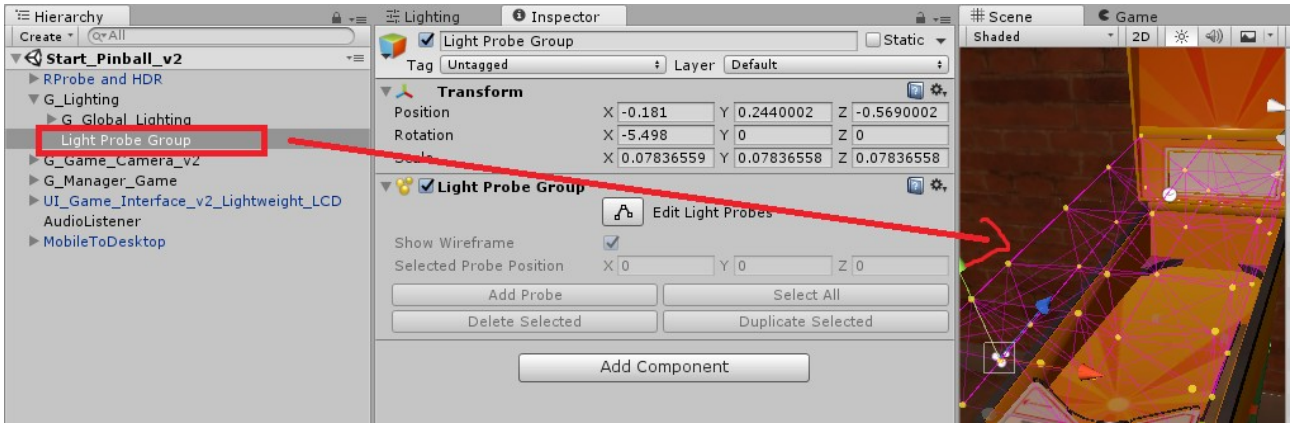
3.5 Light probes

Light probes are used to light moving object (non static) when we use baked lights.

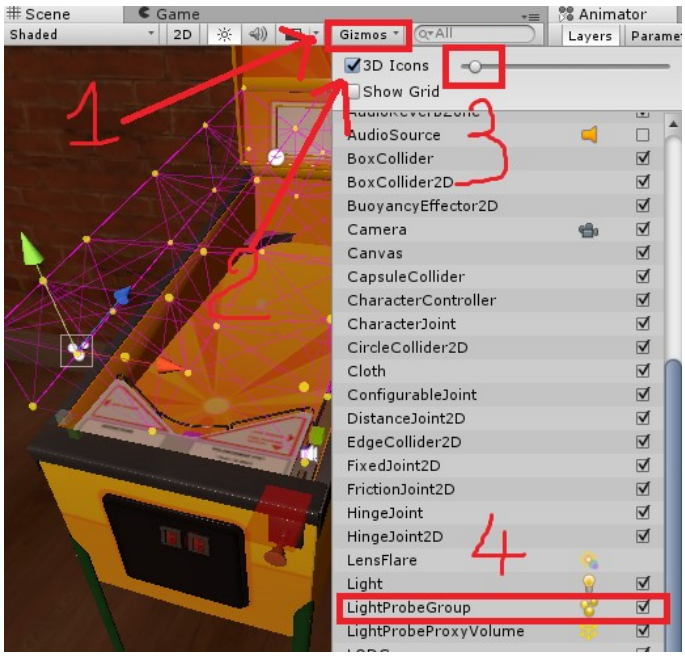
On mobile it is better to use realtime point lights only for ponctual events (bumper light , slingshot light) because it really expensive on resources.
Lightmap pre-compute lighting for Static object.
Light probes simulate lighting for moving objects.

Info : Light Probes doesn't create shadows on moving objects.

Example : scene Start_Pinball (assets->Scenes->Start_Pinball)
Find on the Inspector gameObject light Probe Group (G_Lighting->Light Probe Group):
This object is used to light all the moving object inside the cabinet



If you don't see the light probes group on scene view :
-On scene view clic on Gizmos button (pic 1)
-check if 3d Icons is checked (pic 2)
-Check if the slidebar is not equal to 0 (pic 3)
-Check if LightProbGroup is checked (pic 4)



3.6 Pixel Light count

If a light disappear on your scene it is probably because your Pixel Light Count value is too low.

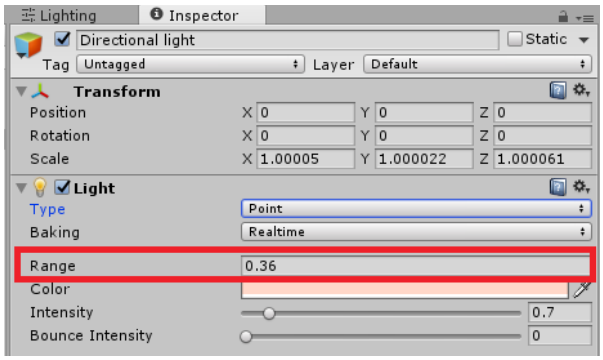
This parameter is use to determine the maximum number of pixel lights when Forward Rendering is used.

You find this parameter on Quality tab (Edit > Project Settings > Quality)

Info : Pixel light count is expensive on resources.

3.7 Realtime Point Light Range

Realtime Point lights are very expensive on resources.
Decrease Range parameter is a good solution to reduce real time light impact.



3.8 Combine Meshes with prefab CombinerMesh_V2

How it works :

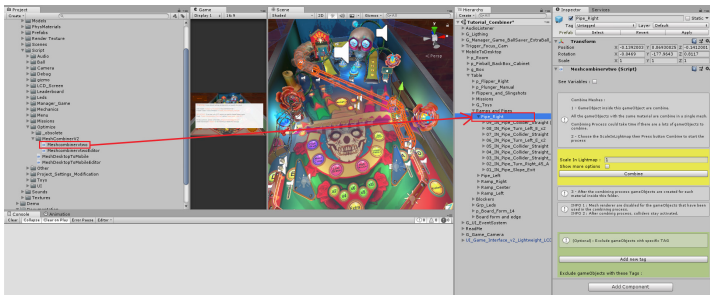
CombinerMesh prefab combine all the mesh that have the same material on a single new mesh. This a good solution to **drastically reduce drawcalls and reduce lighmap precomputed time**.

How to use :

1 Open Tuto scene Tutorial_Combiner

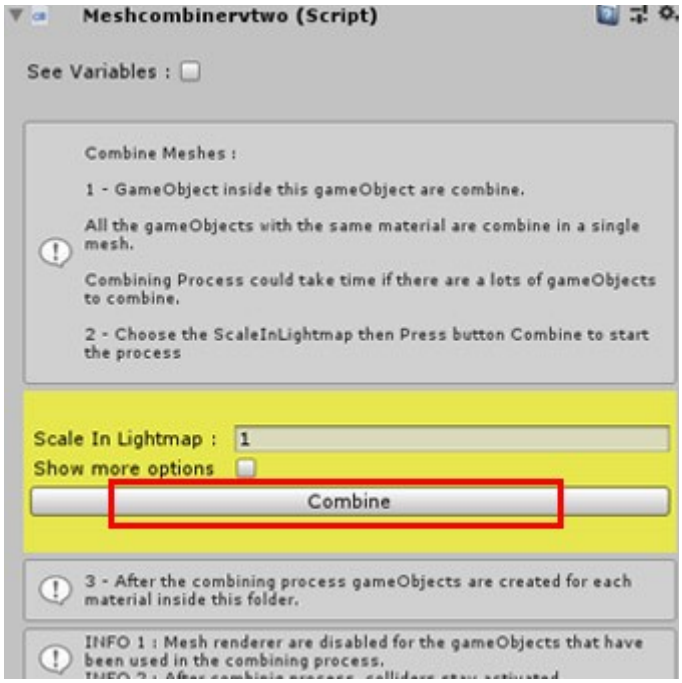
Assets->Scenes->Tuto ->Tuto_Graphics >Tutorial_Combiner

2 Drag and drop the script name Meshcombinervtwo (Assets->Script->Optimize->MeshCombinerV2->CombinerMesh_v2) on Pipe_Right group in hierarchy tab (MobileToDesktop->Table->Ramps and Pipes->Pipe_Right)



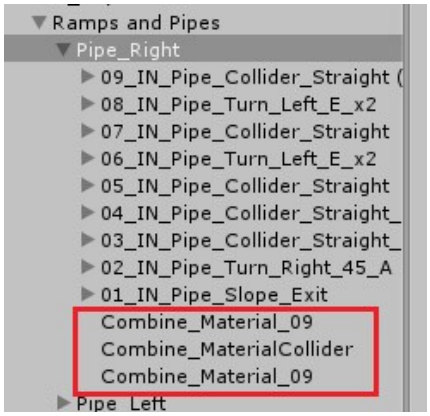
3 Select Pipe_Right group in hierarchy tab.

4 On the Inspector press Combine.

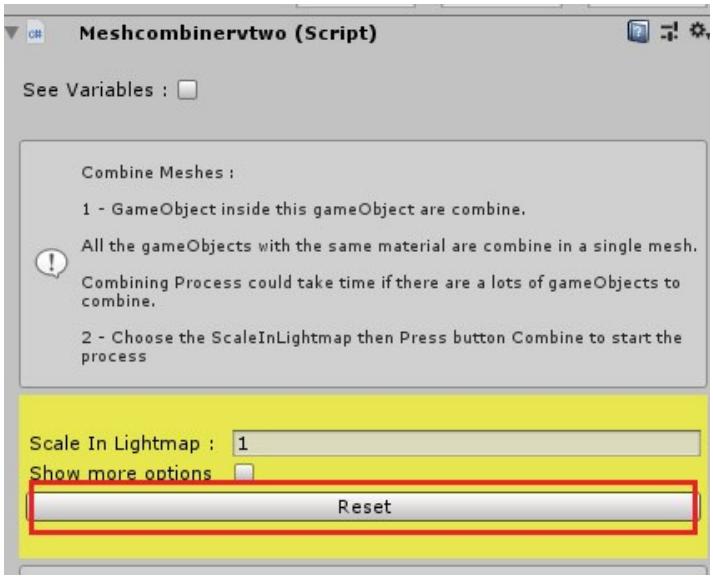


After the process new Combine gameObjects are created inside Objects_ Grp group.

All other objects in the group are hide.



You could revers the process by pressing the button Reset or Ctrl+Z



Troubleshooting :

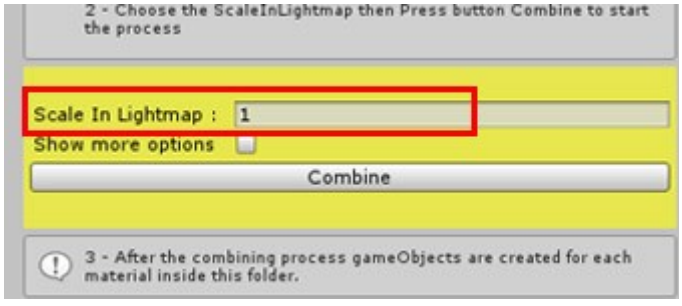
- If you have a lot of objects (or large objects) in group we recommande to separate into several pieces to avoid poor quality lightmaps.

- If you have strange results, this is probably because the number of tris of combine objects are too important.
To solve this issue separate into several groups and combine each group separately.

Options:

Some objects, especially those with rounded edges require more lightmap definition.

To increase the definition of these objects increase Scale in lightmap value.



Tips:

- Create 2 groups

- Add Meshcombinervtwo combiner script on each

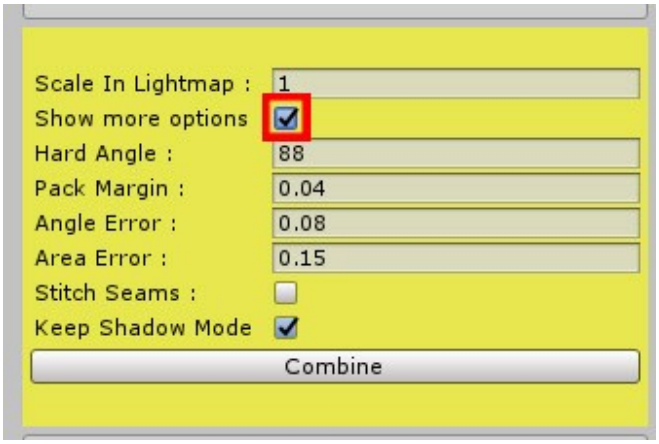
- Add to the first group objects with sharp edges
- Set Scale in lightmap value to 1

- Add to the second group objects with smooth edges
- Set Scale in lightmap value to 4

Caution: If you change the value you must decombine (reset) and then recombine the group (combine)

Tips: if you want to increase the quality of lightmaps for the whole scene, you can increase lightmap resolution value in lightings settings tab.
On the other hand the size of the lightmaps will be larger and the lightmaps precomputing time too.
So it's best to put a low value of lightmap resolution in lightings settings and choose a scale in lightmap value depending on the type of objects(smooth or sharp).

To see more options check Show more options box

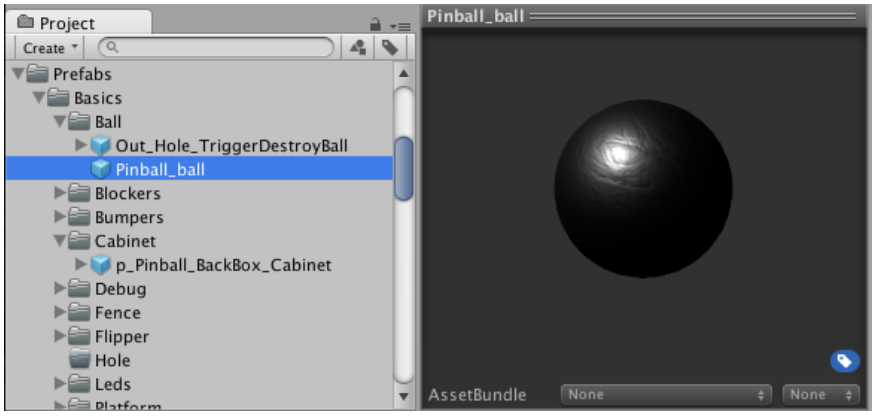


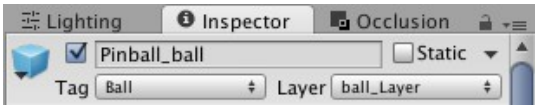
Stitch seams improves the quality of lightmaps

Keep shadow Mode allows you to keep the shadows options (for example cast shadow off)

Tables Mechanics :
(prefabs)

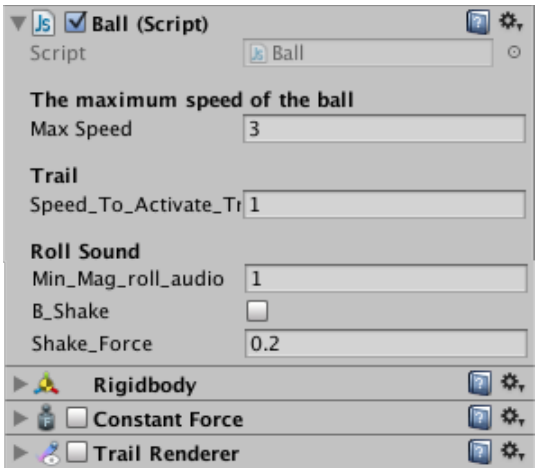
Ball :
Project -> Assets -> Prefabs -> Basics -> Ball -> Pinball_ball





Tag : Ball

Layer : ball_Layer



Find Ball.js on Pinball_ball gameObject

Max Speed : Limit ball speed

Speed_To_Activate_Trail : Minimum speed to activate the trail

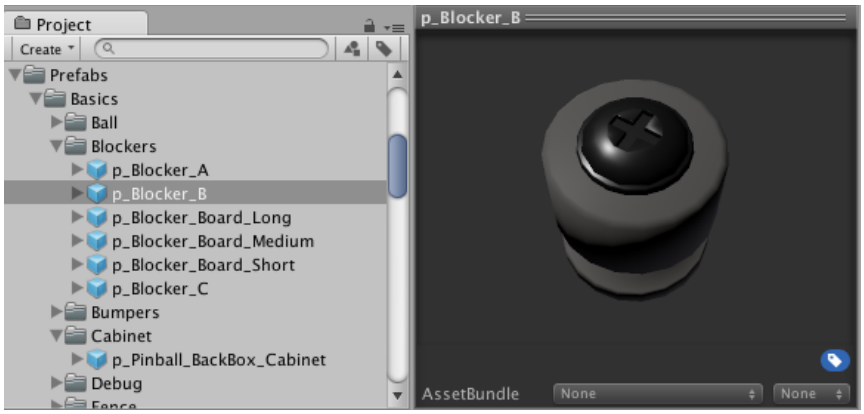
Min_Mag_roll_audio : Minimum speed to activate the rolling sound

b_shake : Activate shake force.

Shake_Force : Force added to the ball if the player use nudge technique.

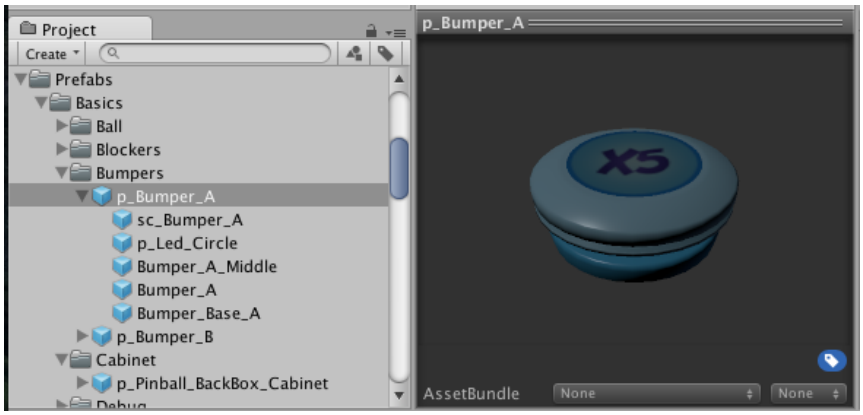
Blocker :

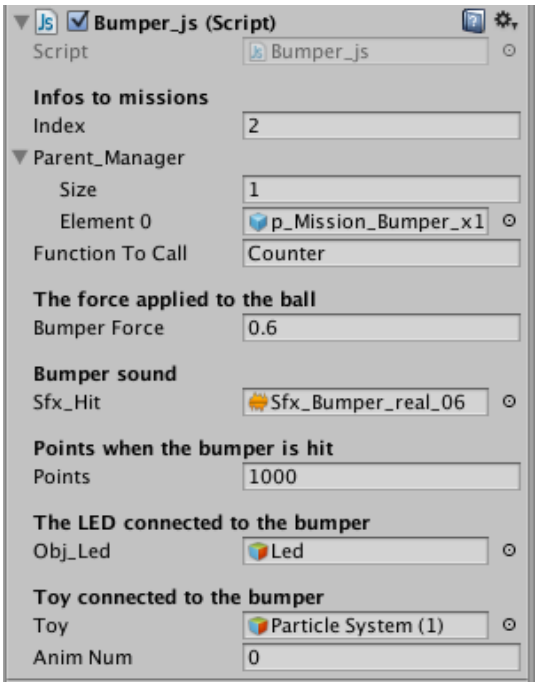
Project -> Assets -> Prefabs -> Basics -> Blockers -> p_Blocker_A or p_Blocker_B or p_Blocker_C



Bumper :

Project -> Assets -> Prefabs -> Basics -> Blockers -> p_Bumper_A or p_Bumper_B





Find Bumper_Js.js on sc_Bumper_A or sc_Bumper_B

Index : Choose a unique ID

Parent_Manager : Connect missions that used this object. You could connect more than one mission.

Function To Call : Call a function

Bumper Force : Force added to the ball

Sfx_Hit : Sound when the ball hit the bumper

Points : Points added to the score when ball hit the bumper

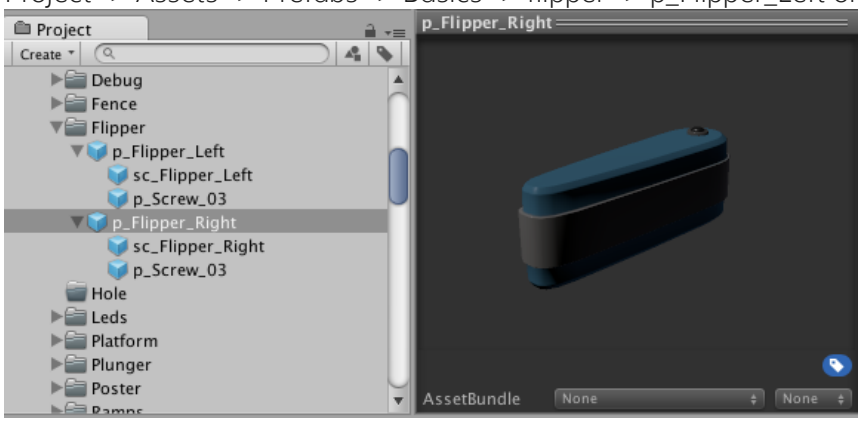
Obj_Led : Connect a led. Led Switch ON when ball hit the bumper

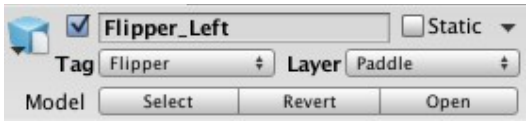
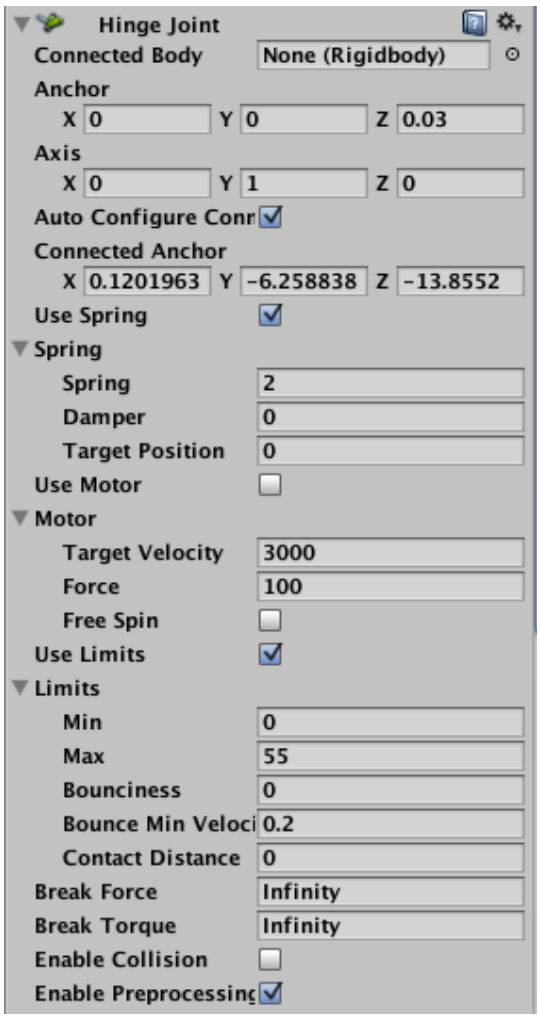
Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.js attached to it. (more about toy)

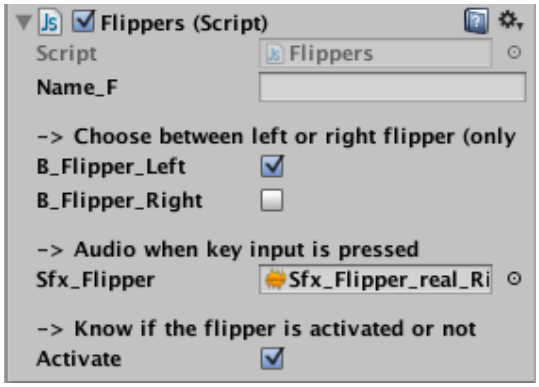
AnimNum : Choose the animation played by the toy.

Flipper :

Project -> Assets -> Prefabs -> Basics -> flipper -> p_Flipper_Left or p_Flipper_Right

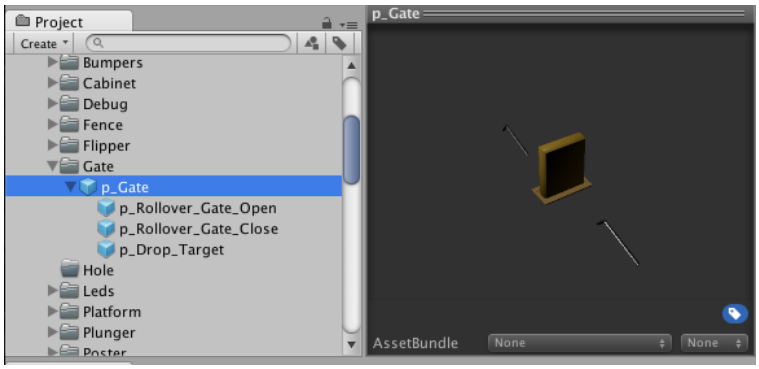


	Find scripts on sc_Flipper_Left or sc_Flipper_Right **Tag :** Flipper. Used by Manager_Game.js to activate or deactivate the plunger. **Layer :** Paddle. Ignore collision between all layers except , ball_Layer" (Flippers.js)
	Manage flipper spring **Spring -> Spring :** Force applied when flipper go back to the init position. **Motor -> Target velocity :** Force applied to the flipper when the player press Input key. **Limits -> Min and Max :** Flipper limits (angle)

	Manage Flipper : flippers.js **Name_F :** Input key. (Auto connect) **B_Flipper_Left :** Check if it is a left Flipper. Uncheck if it is a right flipper. **B_Flipper_Right :** Check if it is right Flipper. Uncheck if it is left flipper. **Sfx_Flipper :** Play sound when button is pressed **Activate :** Activate or deactivate the flipper
	When the ball hit the flipper, play a sound. **volume_Max :** Maximum volume **b_Flipper :** TRUE **Flipper :** Play this sound

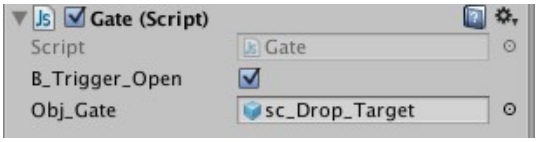
Gate :

Project -> Assets -> Prefabs -> Basics -> Gate -> p_Gate



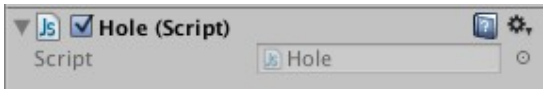
How it work.

A gate is composed with one drop target and two triggers (one to deactivate the target and the other one to activate the target).

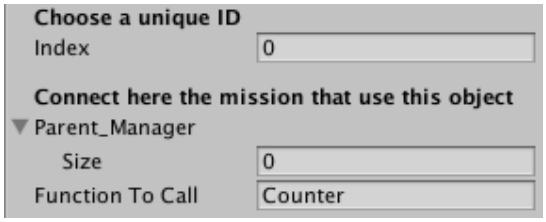
	Find Gate.js on p_rollover_Gate_Open -> sc_Roll_Over_Metal and _rollover_Gate_Close -> sc_Roll_Over_Metal **B_Trigger_Open :** The trigger you want to open the target must be True, the other trigger must be False **Obj_Gate :** Connect the target

Hole : hole and gobble hole
Project -> Assets -> Prefabs -> Basics -> Hole -> p_Hole

How it work.
Step 1 : when a ball enter on collision with sc_Hole (pic 1) the ball is captured.
Step 2 : Then ball is ejected by Spawn_Hole using his position and Z direction (pic 2).



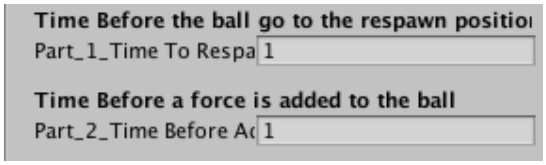
Find Hole.js on sc_Hole gameObject



Index : Choose a unique ID

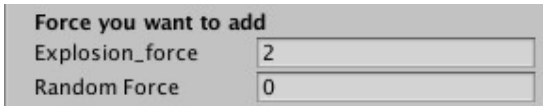
Parent_Manager : Connect missions that used this object. You could connect more than one mission.

Function To Call : Call a function



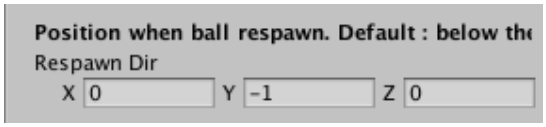
Part_1_TimeToRespawn : Time before ball respawn

Part_2_TimeBeforeAddingForce : Time before adding a force to the ball. This timer start when ball is respawn.



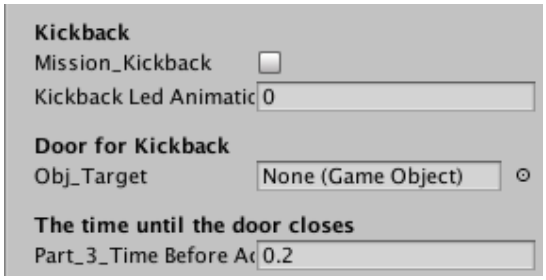
Explosion_force : Force added to the ball.

RandomForce : Add a random force.



Respawn Dir : Position when ball respawn.

obj_Led : Connect a led. (More about led)



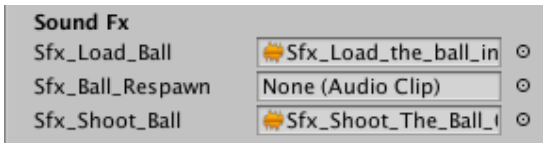
If you want to test kickback use prefab p_Kickback:
Project -> Assets -> Prefabs -> Mechanics -> Gate -> p_Kickback
(more about Kickback)

Mission_Kickback : True if you use hole for kickback.

KickBackLedAnimation : Play animation when ball enter on kickback. (more about Led animation)

Obj_Target : Connect a target

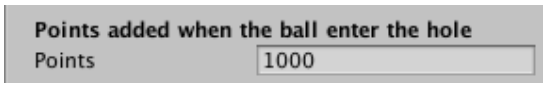
Part_2_TimeBeforeActivateObjDoor : Timer



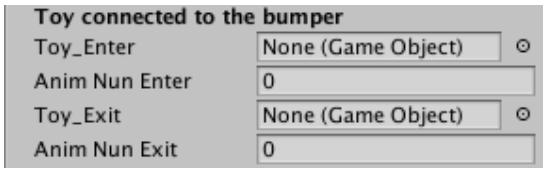
Sfx_Load_Ball : Play a sound when ball is capture

Sfx_Ball_Respawn : Play a sound when ball respawn

Sfx_Shoot_Ball : Play a sound when ball is ejected.



Points : add points to score when ball enter on Hole



Toy_Enter : Play toy animation when ball enter the hole

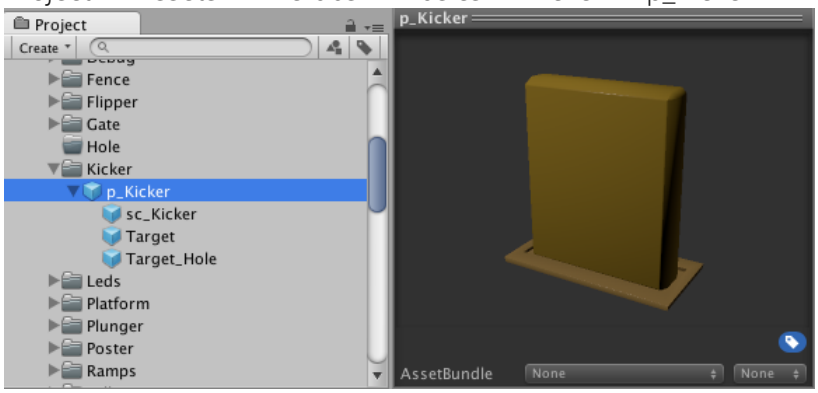
AnimNumEnter : choose animation number.

Toy_Exit : Play toy animation when ball are ejected

AnimNumExit : choose animation number.
(more about toy animation)

Kicker :

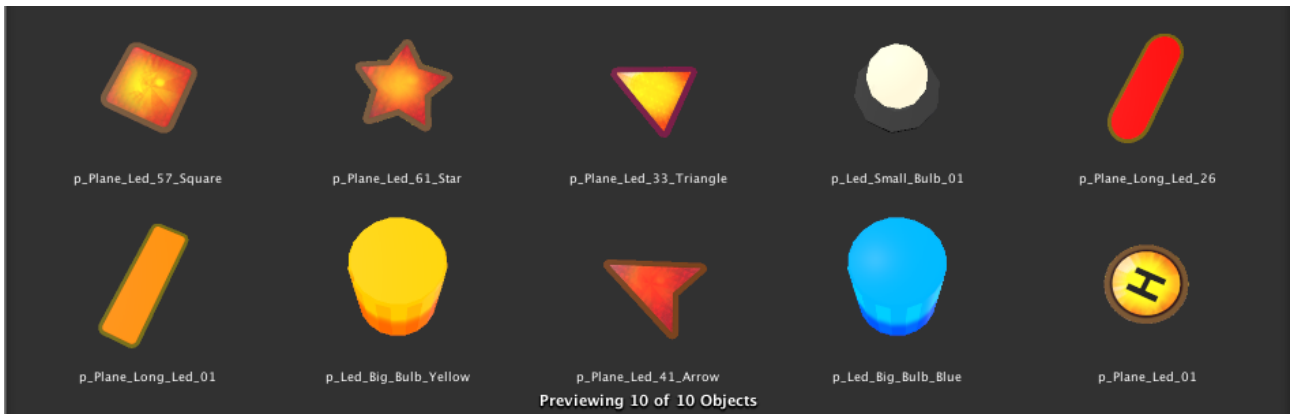
Project -> Assets -> Prefabs -> Basics -> Kicker -> p_Kicker

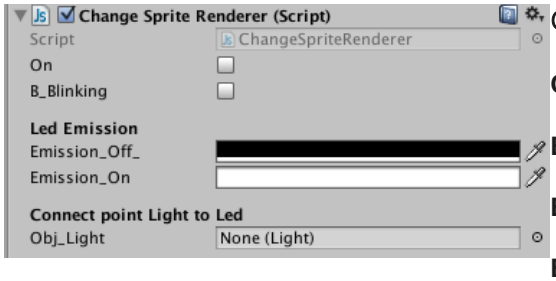


How it work.
When ball hit the kicker a force is apply to the ball.
Kicker works in the same way as Slingshot. (more here)

Led :

Project -> Assets -> Prefabs -> Basics -> Leds ->





Change Sprite Renderer : ChangeSpriteRenderer.js

On : if True Led is switch ON

B_Blinking : if True led blinks.

Emission_Off_ : Choose a color

Emission_On : Choose a color

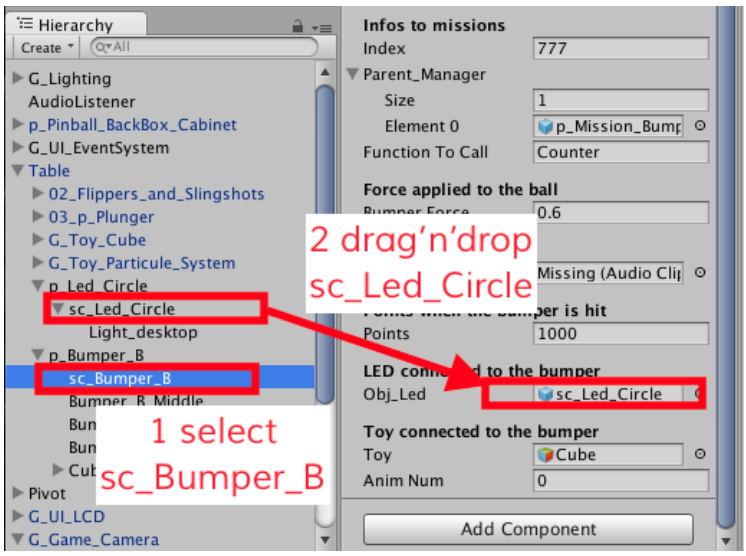
Obj_Light : Connect a point light.

Led : How to connect Led.

You can connect Led to different gameObject.

Example : add Led to Bumper.
Step 1 : Select sc_Bumper_B on Hierarchy (pic 1).
Step 2 : Drag'n'drop sc_Led_Circle inside variable Obj_Led on script Bumper.js (Inspector) (pic 2).

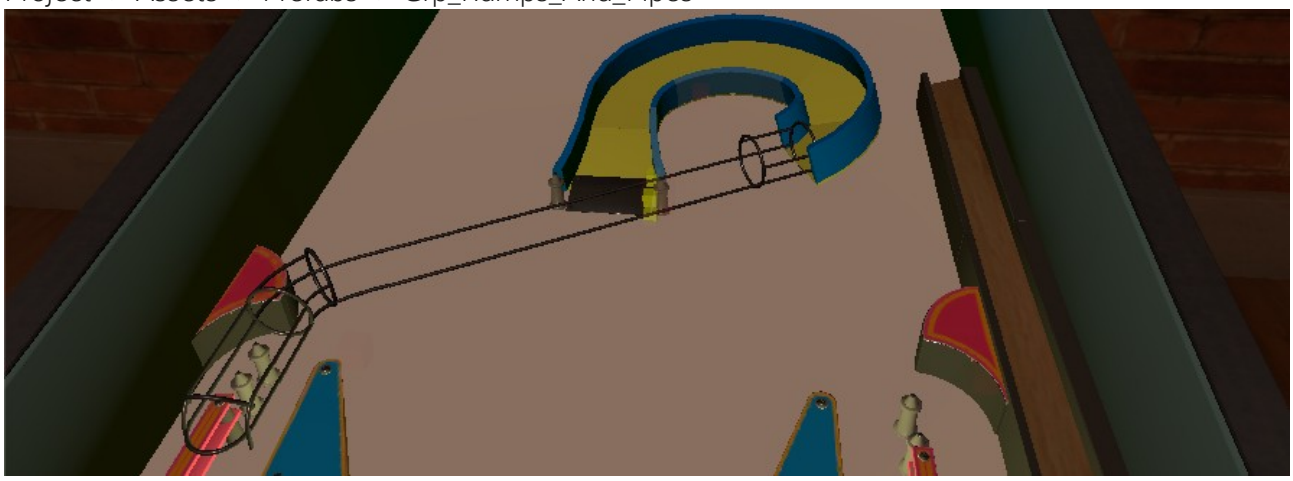
For each type of Led, connect the gameObject starting with sc_



Led : Where to connect Led

You could connect Led to :
- Mission (more info)
- Manager_Game (more info)
- Bumper (more info)

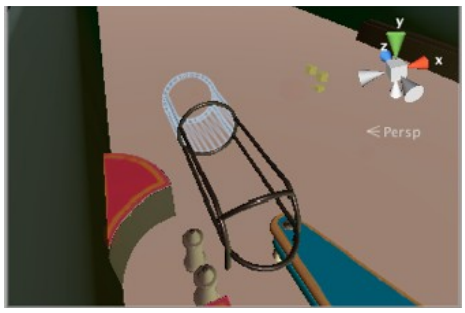
Ramp and pipe :
Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes



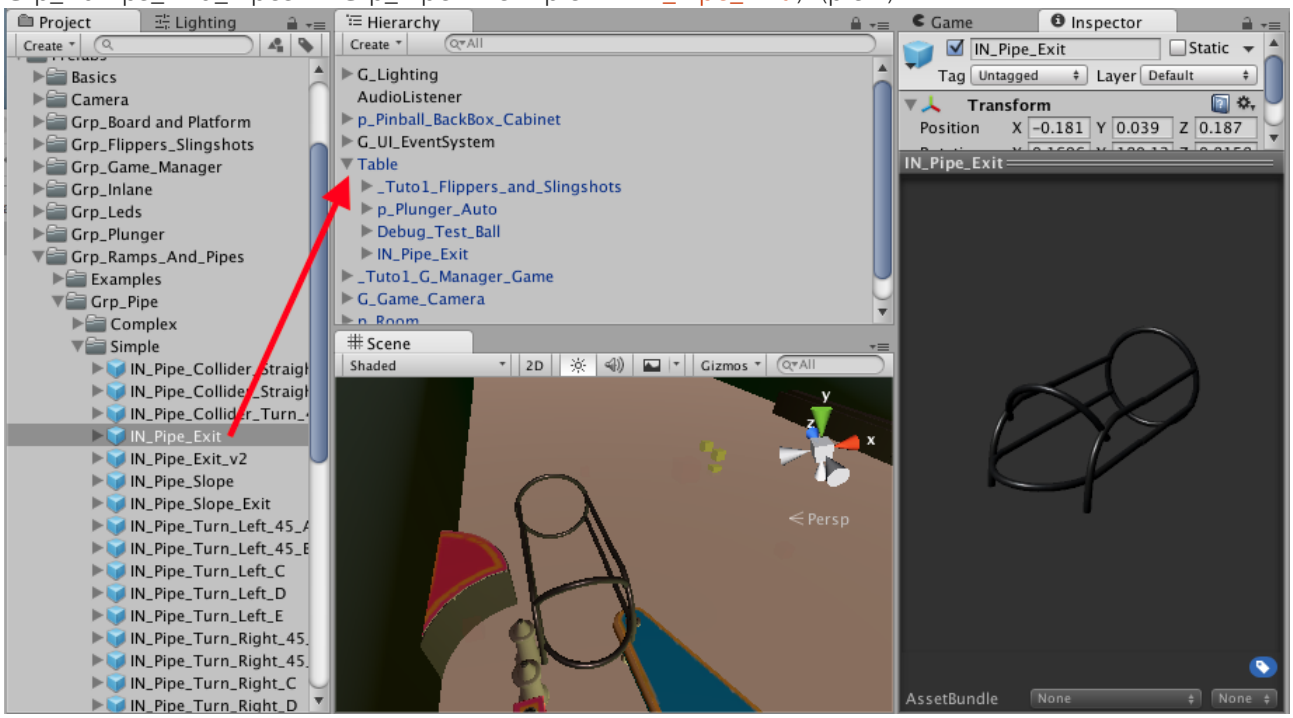
(You find example on this folder).

How to create a pipe :

Example : Create this pipe

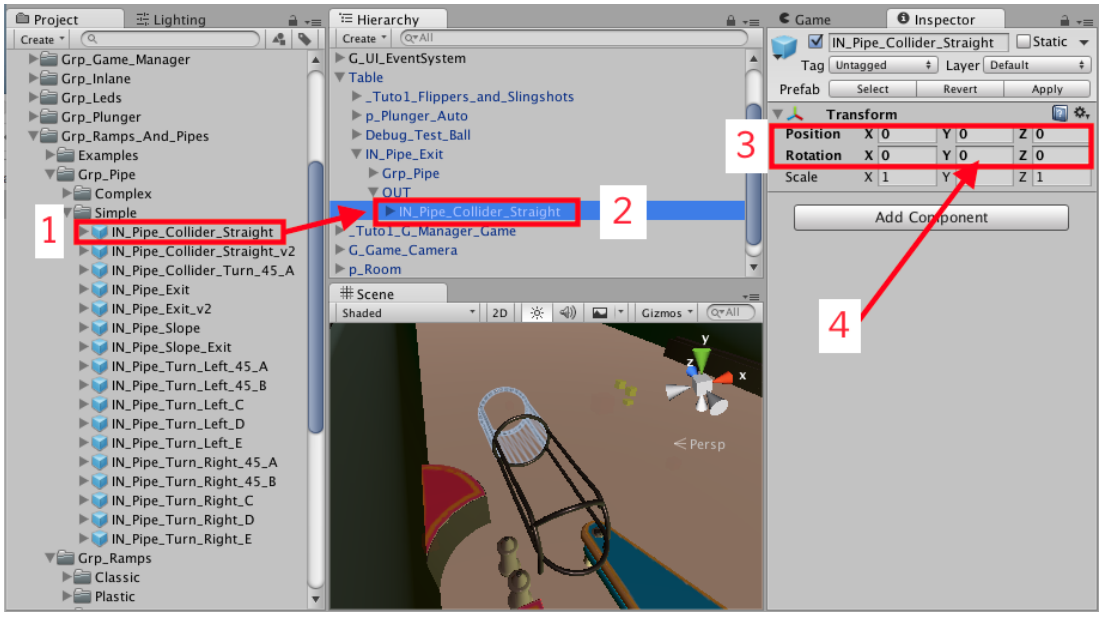


Step 1 : Drag'n'drop IN_Pipe_Exit inside Table on Hierarchy.(Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes -> Grp_Pipe -> Simple -> IN_Pipe_Exit). (pic 1)

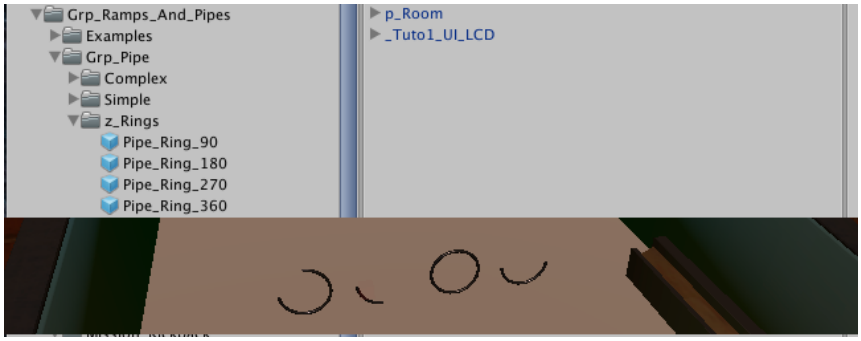


Step 2 : Drag'n'drop IN_Pipe_Collider_Straight inside IN_Pipe_Exit -> OUT on Hierarchy.(Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes -> Grp_Pipe -> Simple -> IN_Pipe_Collider_Straight). (see next page : pic 1 and 2)
Reset IN_Pipe_Collider_Straight transform. Position (0,0,0) and rotation (0,0,0) (pic 3).

IMPORTANT : In some case you need to change Y rotation. Try -90 , 90 or 180 (pic 4).



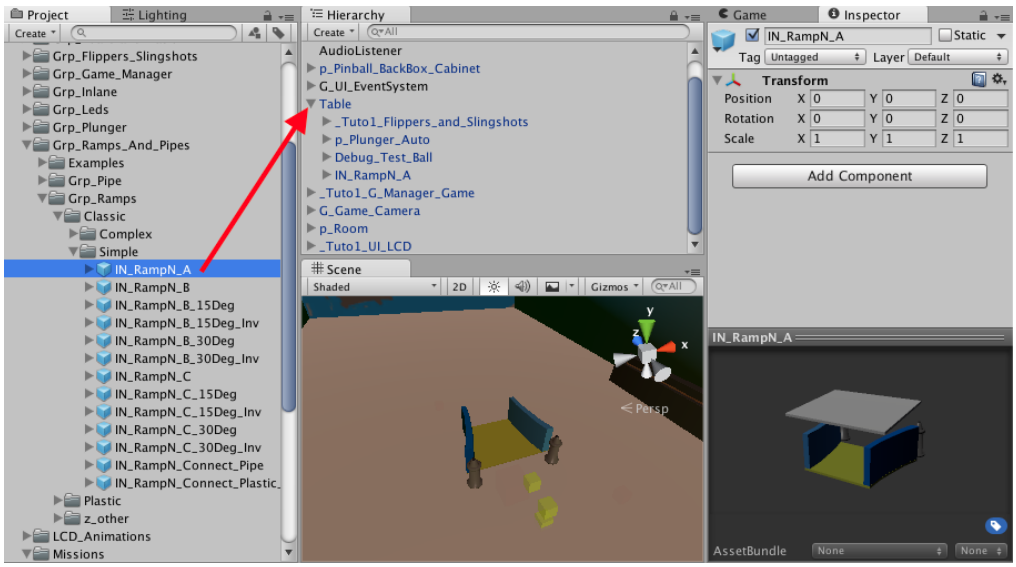
INFO: Inside folder z_ring you could find different type of rings to customize your pipe.



How to create a ramp :

Example : Create this ramp

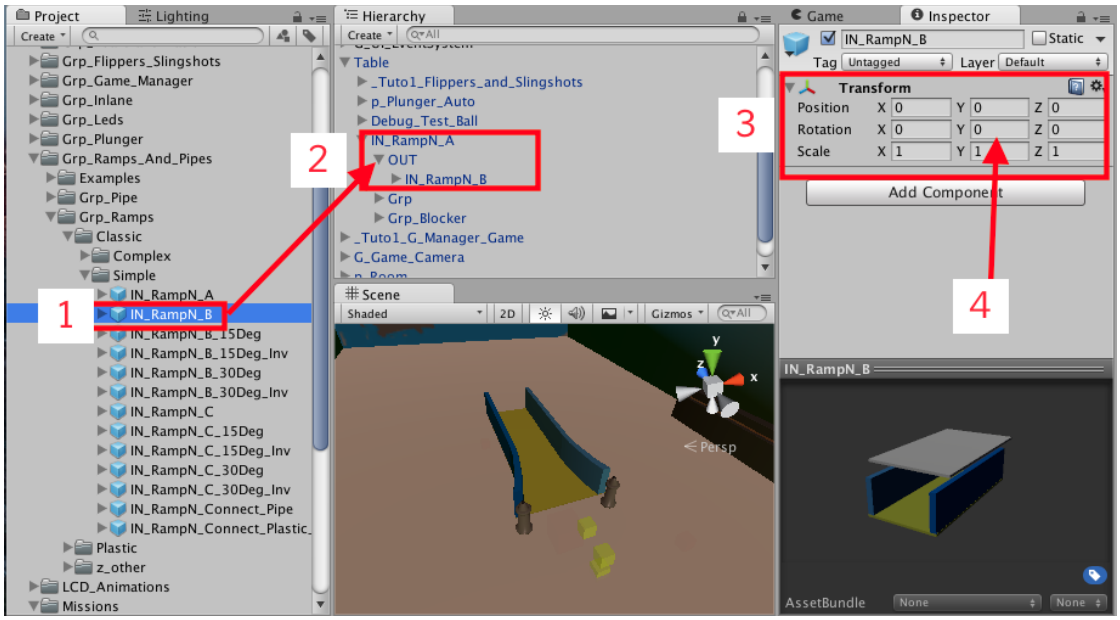
Step 1 : Drag'n'drop IN_RampN_A inside Table on Hierarchy.(Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes -> Grp_Ramps -> -> Classic -> Simple -> IN_RampN_A). (see next page pic 1)



Step 2 : Drag'n'drop IN_RampN_B inside IN_RampN_A -> OUT on Hierarchy.(Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes -> Grp_Pipe -> Simple -> IN_RampN_B). (pic 1 and 2)

Then reset IN_RampN_B transform. Position (0,0,0) and rotation (0,0,0) (pic 3).

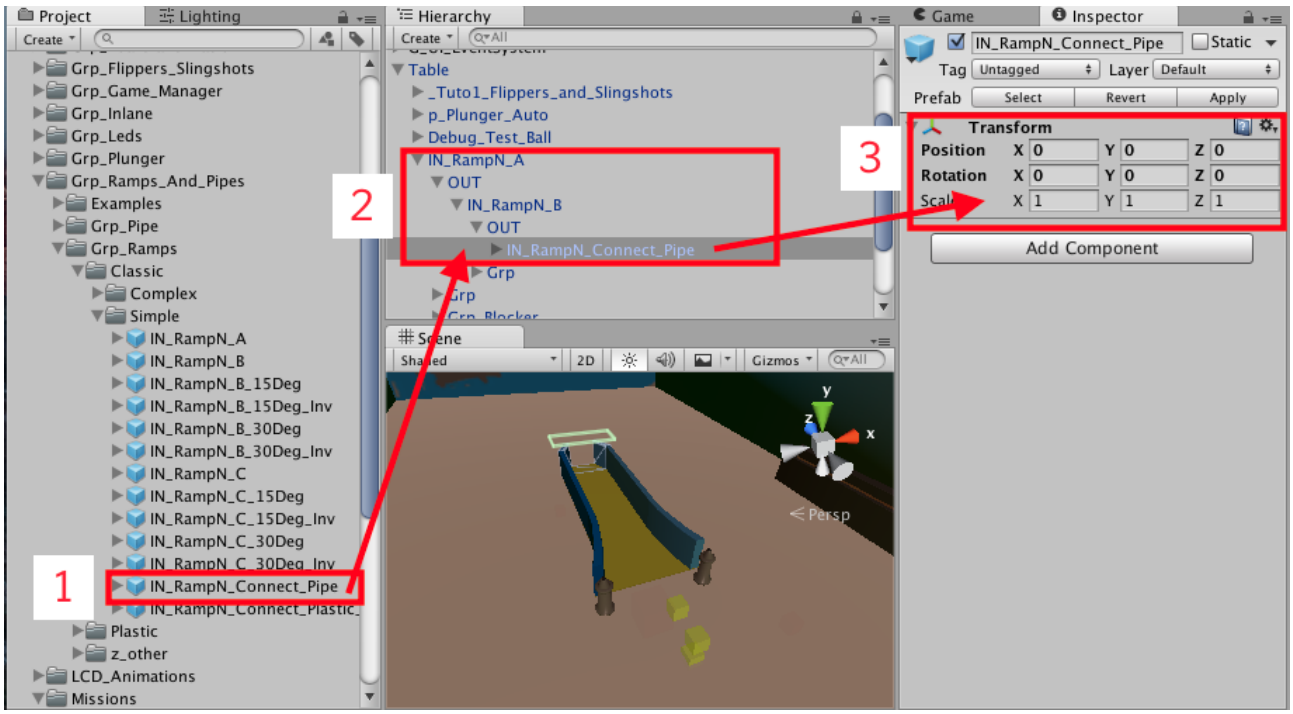
Info : In some case you need to change Y rotation. Try -90 , 90 or 180 (pic 4).



Step 3 : Drag'n'drop IN_RampN_Connect_Pipe inside IN_RampN_B -> OUT on Hierarchy.(Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes -> Grp_Pipe -> Simple -> IN_RampN_Connect_Pipe). (pic 1)

Then reset IN_RampN_Connect_Pipe transform. Position (0,0,0) and rotation (0,0,0) (pic 3).

Info : In some case you need to change Y rotation. Try -90 , 90 or 180 (see next page pic 4).



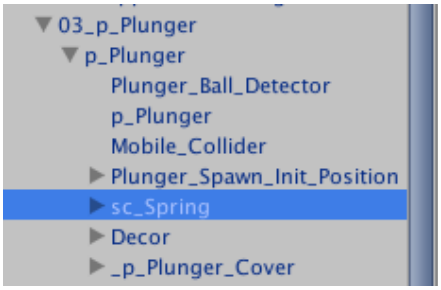
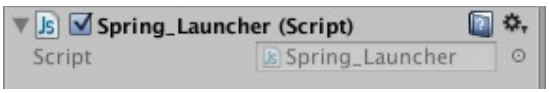
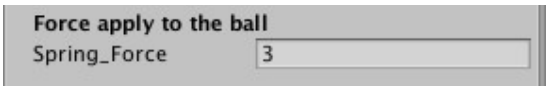
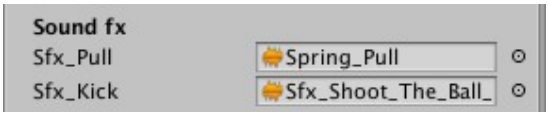
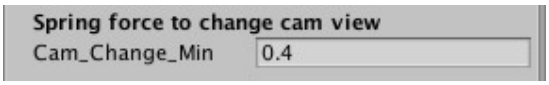
Plunger :
Project -> Assets -> Prefabs -> Basics -> Plunger -> p_Plunger_Auto, p_Plunger_Manual ...

How it work.
Press return to pull the spring.

Change Inputs on gameObject Manager_Game on Hierarchy (script manager_Input_Setting.js).

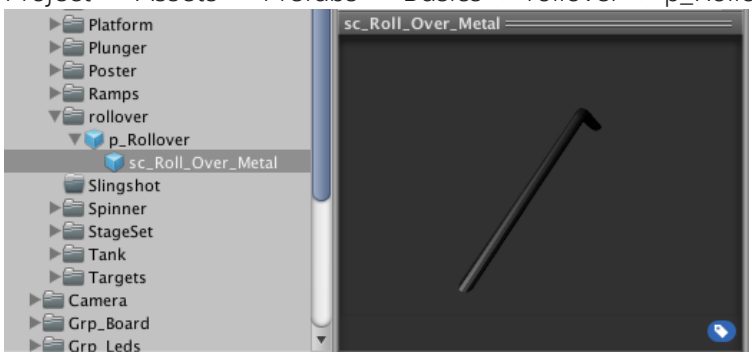
IMPORTANT 1 : Don't rename plunger_Spawn because it call by Manager_Game on Hierarchy. It is used to respawn ball when ball is lost.

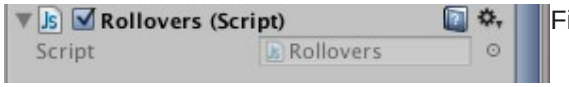
IMPORTANT 2 : If you are using a plunger other than p_Plunger_Manual you probably need to set the gameObject Multi_Ball localPosition.y to .09

	
	**Tag :** Plunger . This tag is used by Manager_Game on Hierarchy.
	Find spring_Launcher.js on p_Plunger_Manual -> p_Plunger -> sc_Spring
	**Auto_Mode :** True (plunger manual) False (plunger auto)
	**Spring_Force :** Force apply to ball.
	**Sfx_Pull :** Sound when player pull the plunger **Sfx_Kick :** Sound when ball is ejected
	**Cam_Change_Min :** If Spring force > Cam_Change_Min the view change.
	**Timer :** Time to prevent bug when camera switch between playfield camera and plunger camera.

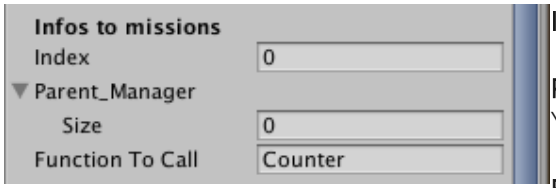
Rollover :

Project -> Assets -> Prefabs -> Basics -> rollover -> p_Rollover





Find Rollovers.js on sc_Rollover_Metal gameObject

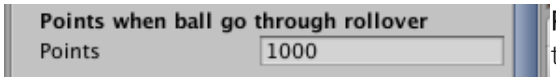


Index : Choose a unique ID

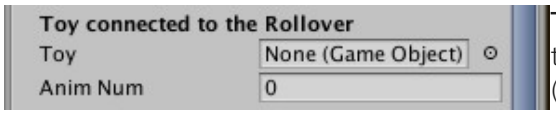
Parent_Manager : Connect missions that used this object. You could connect more than one mission.

Function To Call : Call a function

Sfx_Hit : Sound when the ball hit the bumper



Points : Points added to the score when ball go through the rollover

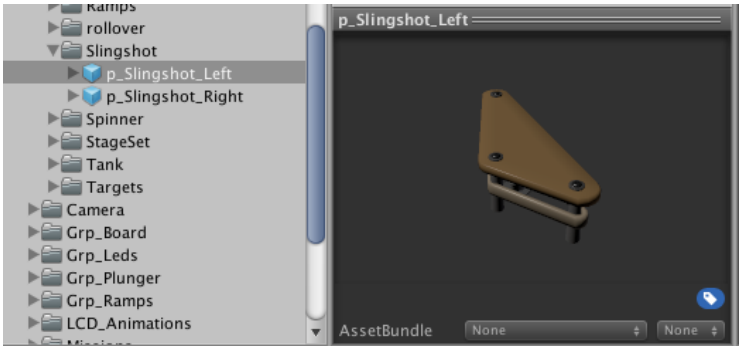


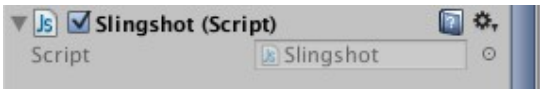
Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.js attached to it.
(more about toy)

AnimNum : Choose the animation played by the toy.

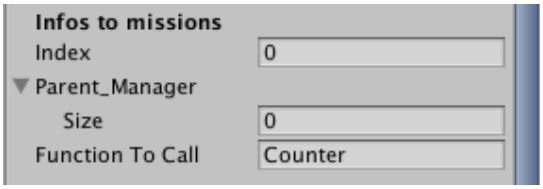
Slingshot : and kicker

Project -> Assets -> Prefabs -> Basics -> Slingshot -> p_Slingshot_Left or p_Slingshot_Right





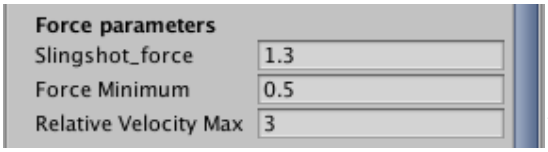
Find Slingshot.js on sc_Slingshot gameObject



Index : Choose a unique ID

Parent_Manager : Connect missions that used this object. You could connect more than one mission.

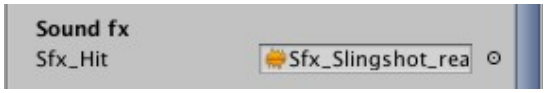
Function To Call : Call a function



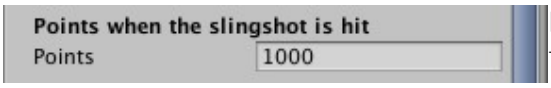
Slingshot_force : Force apply to ball

ForceMinimum : Minimum impact velocity to apply force to ball

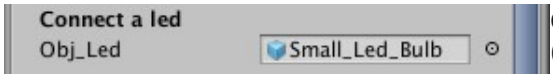
RelativeVelocityMax : Maximum force apply to ball



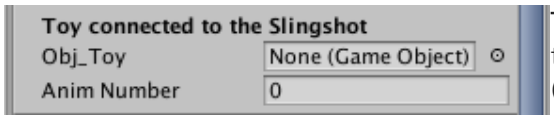
Sound_fx : Sound when the ball hit the slingshot



Points : Points added to the score when ball go through the rollover



Obj_Light : Connect a led.
(more about led)

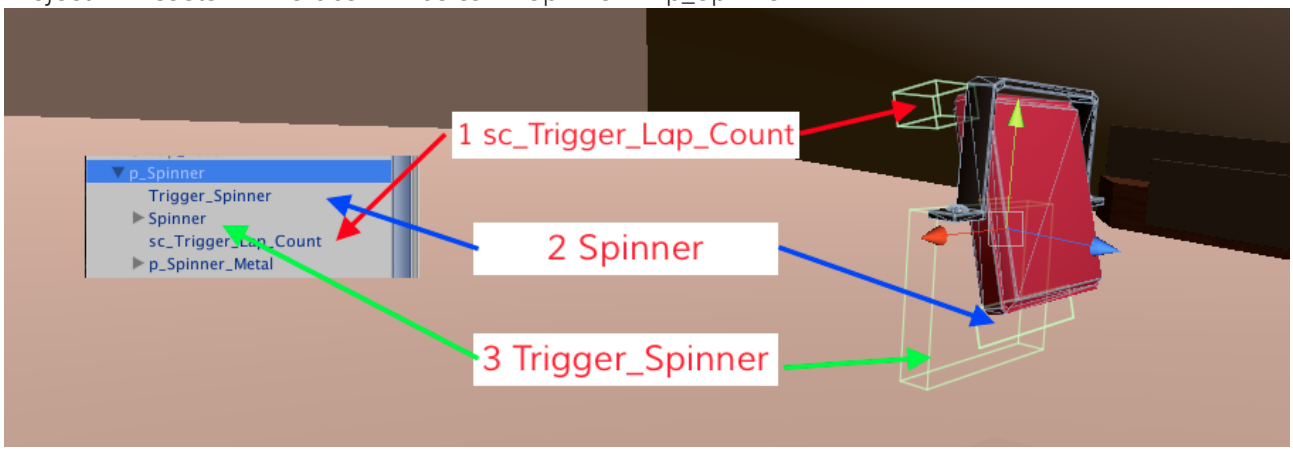


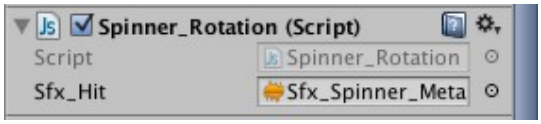
Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.js attached to it.
(more about toy)

AnimNum : Choose the animation played by the toy.

Spinner :

Project -> Assets -> Prefabs -> Basics -> Spinner -> p_Spinner



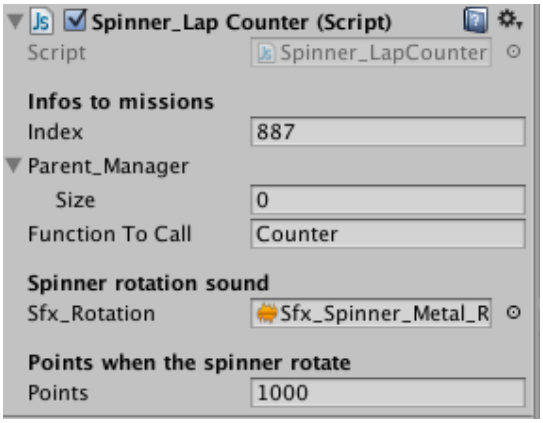


2 : Find Spinner_Rotation.js on sc_Spinner gameObject

Sfx_Hit : play a sound.

1 : Find Spinner_Lap_Counter.js on sc_Trigger_Lap_Count gameObject

Index : Choose a unique ID



Parent_Manager : Connect missions that used this object. You could connect more than one mission.

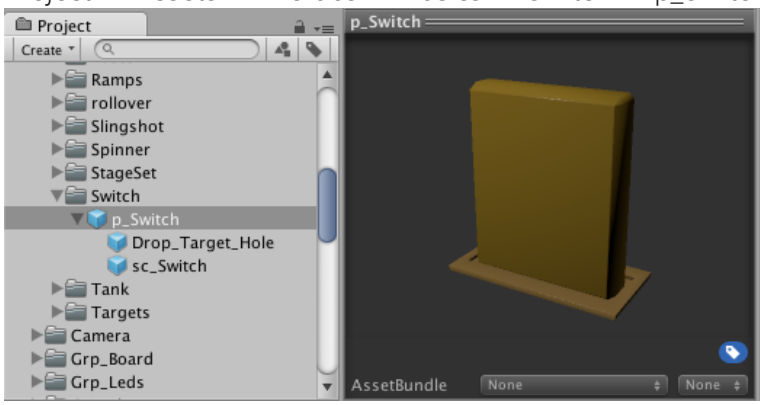
Function To Call : Call a function

Sfx_Rotation : play a sound.

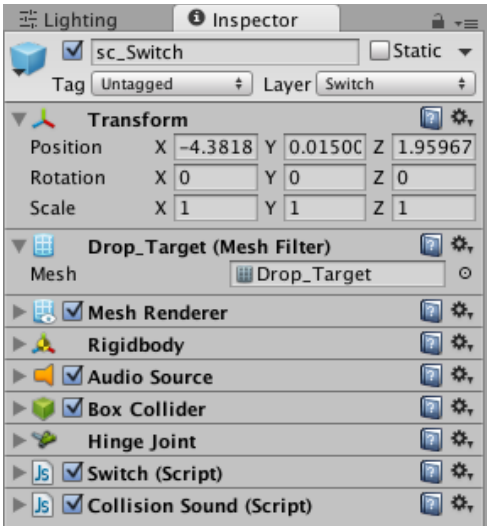
Points : Points earn when spinner rotate

Switch :

Project -> Assets -> Prefabs -> Basics -> Switch -> p_Switch



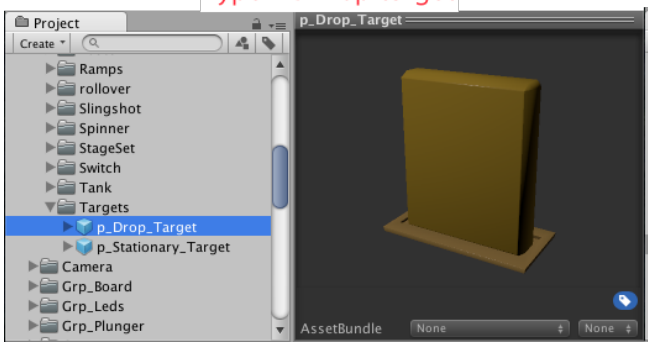
How it works.
Switch is similar to gate but it use a spring.
Ball is restricted to one direction.



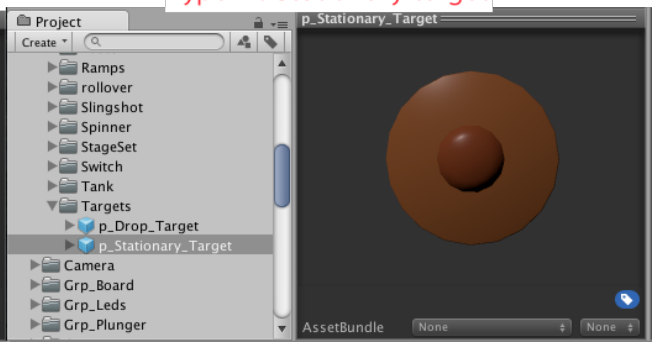
Find Switch.js, Hinge joint and CollisionSound.js on sc_Switch gameObject

Targets : drop target, stationary target and vari-target
Project -> Assets -> Prefabs -> Basics -> Targets -> p_Drop_Target and p_Stationary_Target

Type 1 : Drop target

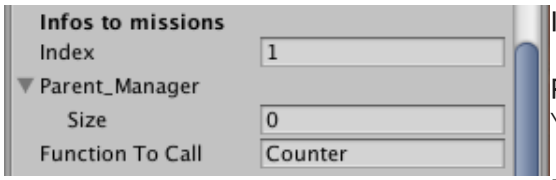


Type 2 : Stationary target



Find Target.js on sc_Drop_Target and sc_Stationary_Target gameObject

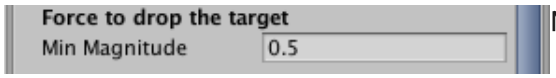
B_Drop_Target :
Drop target and vari-target = True
Stationary target = False



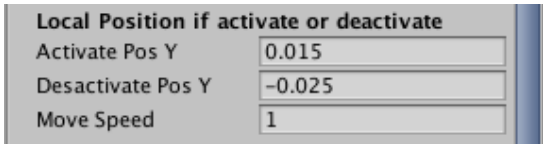
Index : Choose a unique ID

Parent_Manager : Connect missions that used this object. You could connect more than one mission.

Function To Call : Call a function



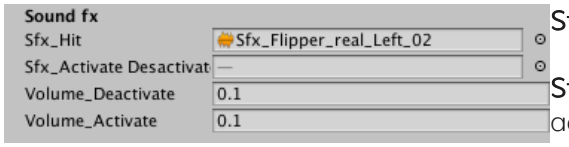
MinMagnitude : minimum magnitude when ball hit target



ActivatePosY : local Position when target is activated

DeactivatePosY : local position when target is deactivated

MoveSpeed : speed to reach the new target position



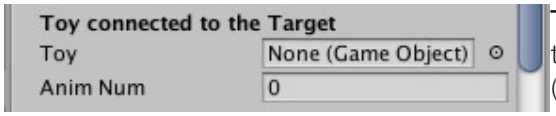
Sfx_Hit : play a sound when ball hit target.

Sfx_Activate_Deactivate : play a sound when target is activate or deactivate.

Volume_Deactivate : volume when the target is deactivated.

Volume_Activate : volume when the target is activated.

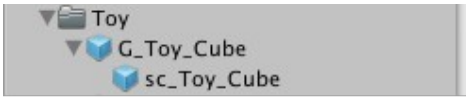
Points : Points earn when ball hits target.

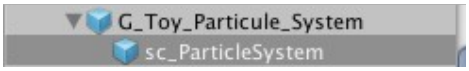


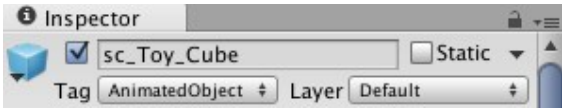
Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.js attached to it.
(more about toy)

Toys :

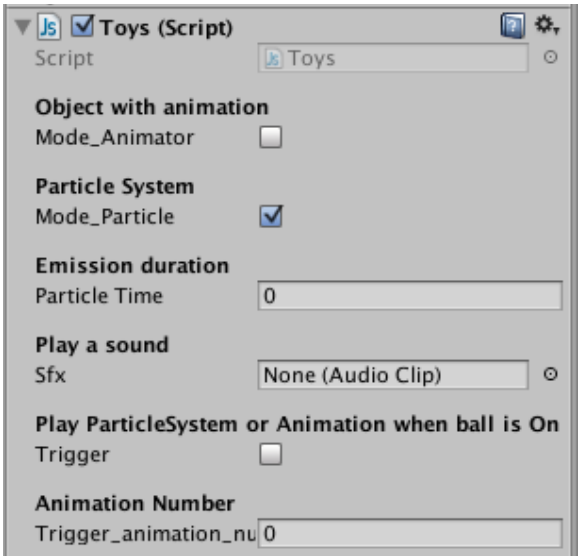
There are two type of Toy. Toy with animation and Toy with particle system. Look at example prefabs G_Toy_Cube and G_Toy_particle_System (Project -> Assets -> Prefabs -> Toy ->)







Tag : AnimatedObject



Find script Toys.js on gameObject sc_Toy_Cube or sc_ParticleSystem

Mode_Animator : True if toy use animation.

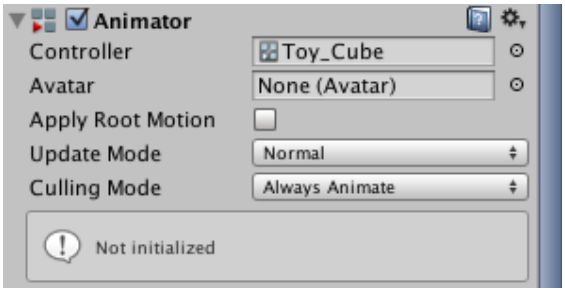
Mode_Particle : True if toy use particle system.

Particle Time : choose particle emission duration (second).

Sfx : Connect a sound.

Trigger : if True. Play animation or Particle when ball enter Toy trigger.

Trigger_animation_number : Choose animation when ball enter Toy trigger (between 0 to 4). (more info here)



If Toy use animations :
(more info here)

If Toy use particle system

If Toy need to play a sound

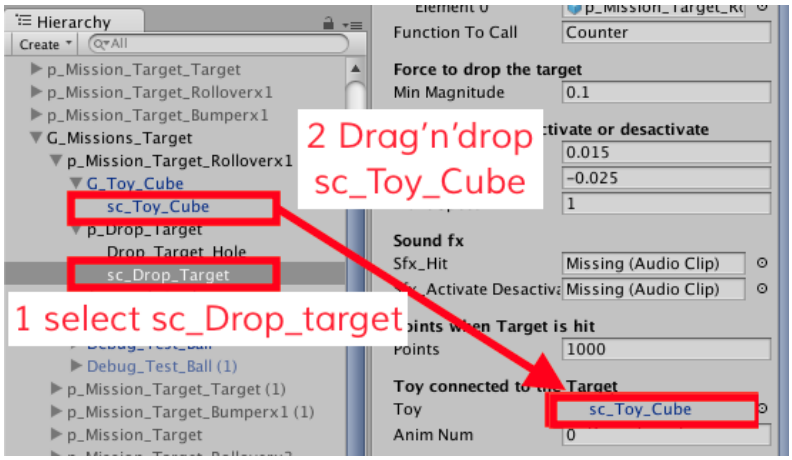
1° Toys : How to connect Toys.

You can connect Toy in different ways.

Example : add Toy to Target.

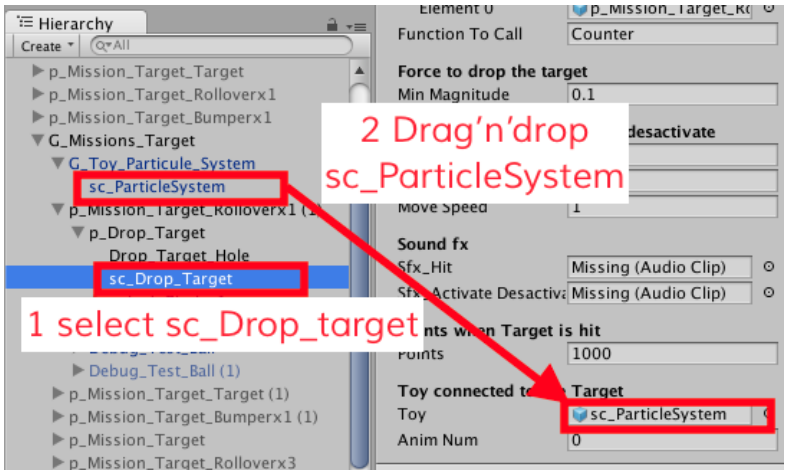
Case 1 : Add Toy with animation.

Step 1 : Select sc_Drop_Target on hierarchy (pic 1).
Step 2 : Drag'n'drop sc_Toy_Cube inside variable Toy on script Target.js (Inspector) (pic 2).



Case 2 : Add Toy with paricleSystem.

Step 1 : Select sc_Drop_Target on Hierarchy (pic1).
Step 2 : Drag'n'drop sc_ParticleSystem inside variable Toy on script Target.js (Inspector) (pic 2).

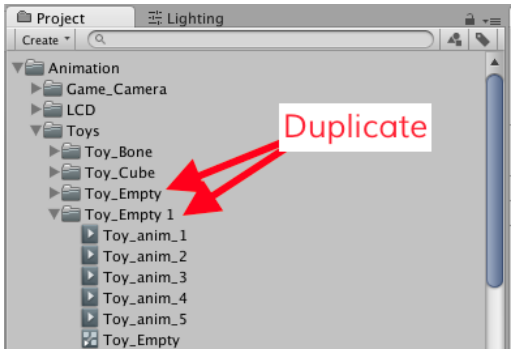


2° Toys : Where to connect Toy

You could connect Toy to :
- Mission (more info)
- Target (more info)
- Rollover (more info)
- Bumper (more info)
- Hole (more info)

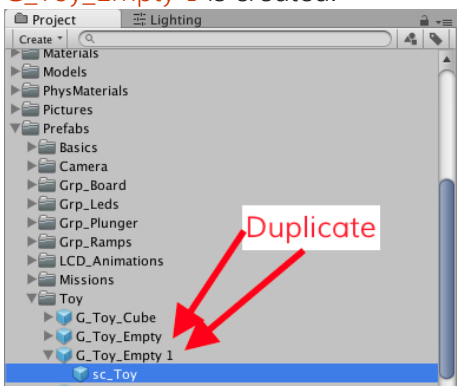
3° Toys : Create a new Toy with animation.

Step 1 : Duplicate the folder that contains the animations for this toy.
Duplicate Toy_Empty (Project -> Assets -> Animation -> Toys -> Toy_Empty)

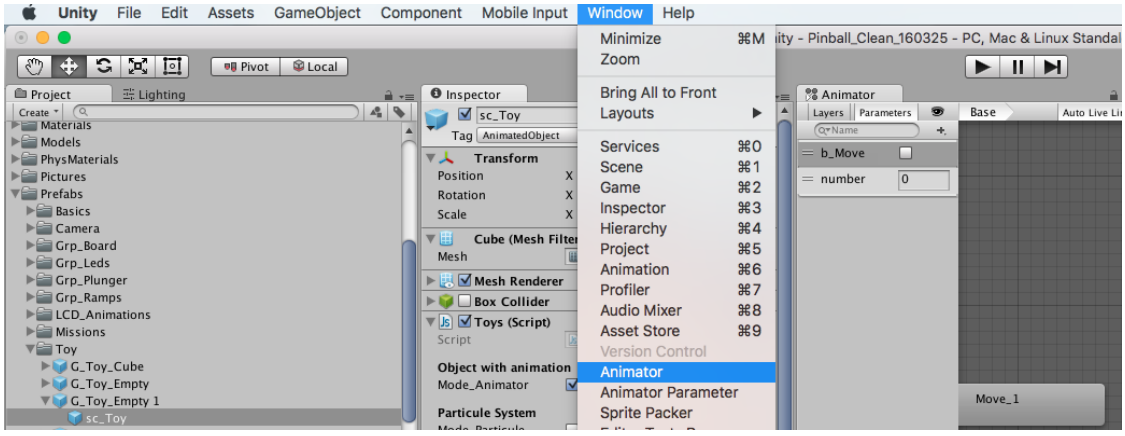


A new folder , Toy_Empty 1" is created.

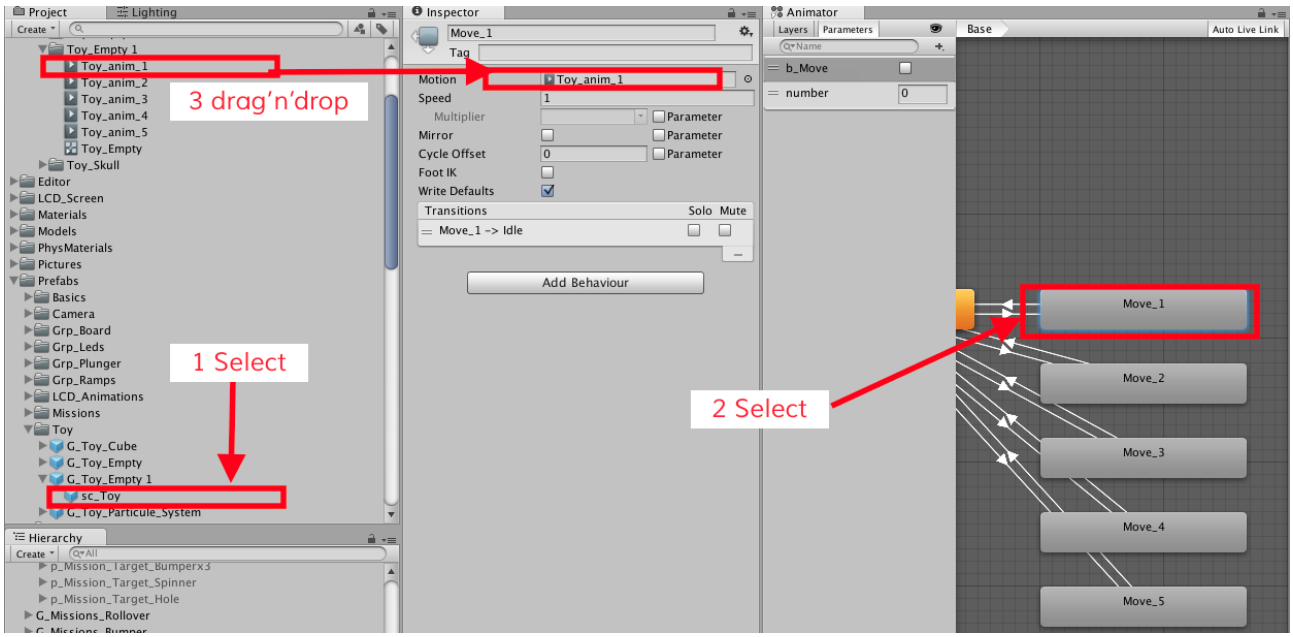
Step 2 : Duplicate the Toy prefab
Duplicate G_Toy_Empty (Project -> Assets -> Prefabs -> Toy -> G_Toy_Empty) (see picture next page)
G_Toy_Empty 1 is created.



Step 3 : Configure the animator Toy
- First open Animator Window.

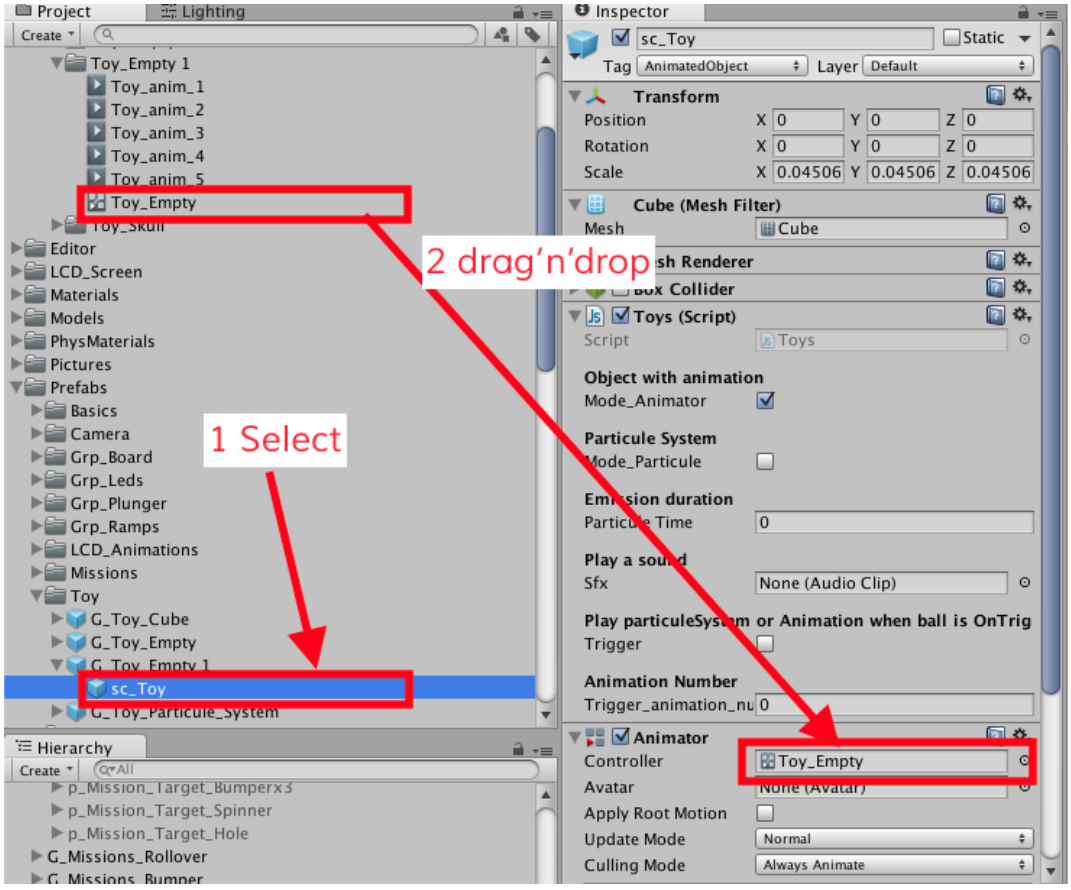


- Then open G_Toy_Empty 1 on Project folder and select sc_Toy (pic 1).
- Select Animation State named Move_1 (pic 2).
- Drag'n'drop Toy_anim_1 (Project -> Assets -> Animation -> Toys -> Toy_Empty 1 -> Toy_anim_1) inside Motion on the Inspector (pic 3).
- Do the same for state Movie_2, Movie_3, Movie_4 and Movie_5 with Toy_Empty 2, Toy_Empty 3, Toy_Empty 4, Toy_Empty 5.



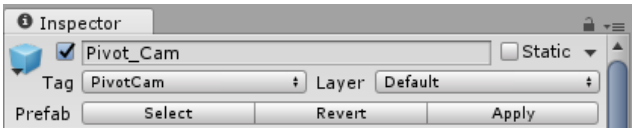
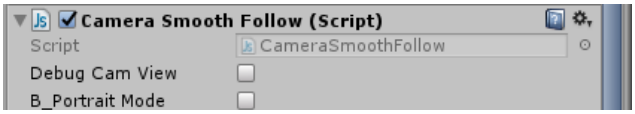
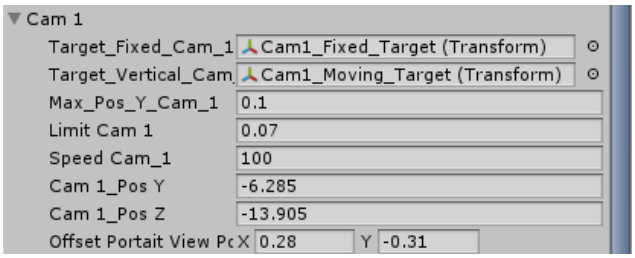
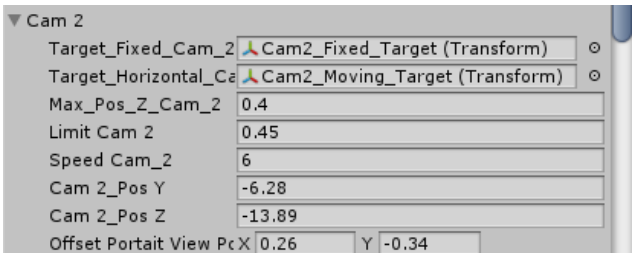
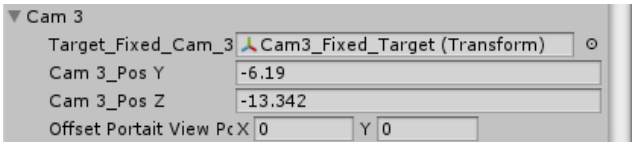
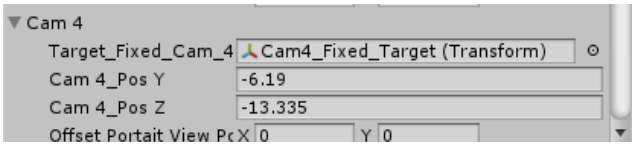
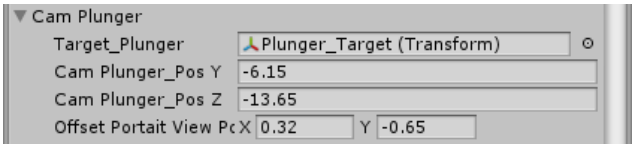
Step 4 : Connect animator to the toy prefab

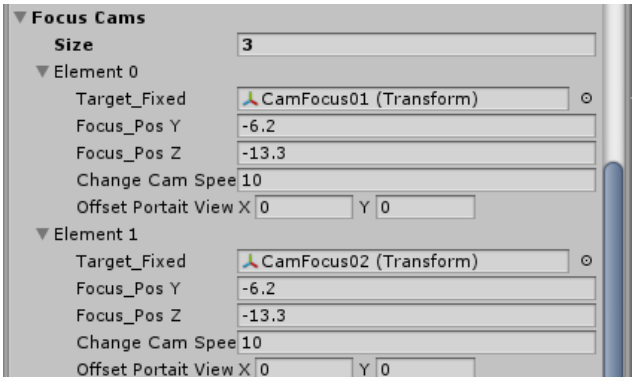
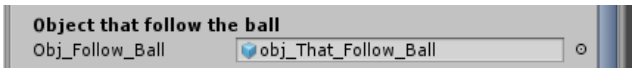
- Select sc_Toy on Project folder (pic 1) and drag'ndrop Toy_Empty (Project -> Assets -> Animation -> Toys -> , Toy_Empty 1" -> Toy_Empty) inside Animator -> Controller on the Inspector (pic 2).



Step 5 : Toy is ready.
Now you could put object on sc_Toy and create your own animations for the Toy.

Features :
Camera

	**Tag :** PivotCam
	**DebugCamView :** true if you want to tweak your camera view. **IMPORTANT :** Uncheck the box when you finish your tweaks
	**Target_Fixed_Cam_1 :** Camera look at this gameObject if LimitCam1 < 0.1 **Target_Vertical_Cam_1 :** Camera follow this gameObject if LimitCam1 > 0.1 **Max_Pos_Y_Cam_1 :** Max movement for the camera **LimitCam1 :** Limit in Z position to switch betweenTarget_Fixed_Cam_1 and Target_Vertical_Cam_1 **SpeedCam1 :** Camera speed **Cam1PosY :** Use to tweak your camera **Cam1PosZ :** Use to tweak your camera **OffsetPortraitViewPos :** Use to tweak the camera position on mobile when orientation = portrait
	Same as Cam 1
	Same as Cam 1
	Same as Cam 1
	Same as Cam 1

	Same as Cam 1
	**Obj_Follow_Ball :** This object follow the ball

5.1 What's new ?

Camera system from version is obsolete and probably doesn't work any more on your project. Replace G_Game_Camera with G_Game_Camera (more info)

G_Game_Camera :

- allow you to tweak camera position.
- allow you to add Focus Camera point of view

5.2 How to tweak a camera view position and rotation ?

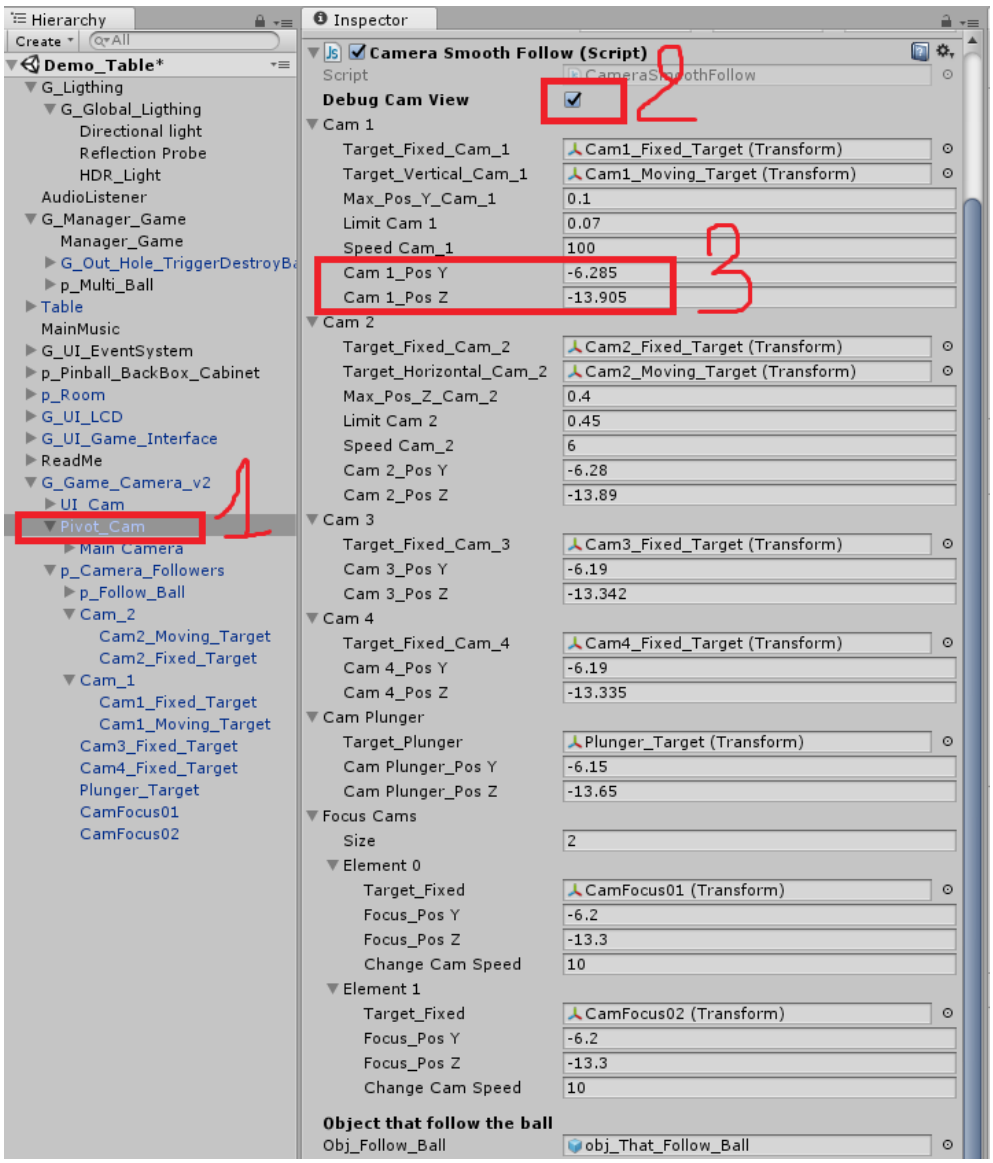
Example : Cam view 1

Change position :

Select PivotCam on gameObject G_Game_Camera (pic1)

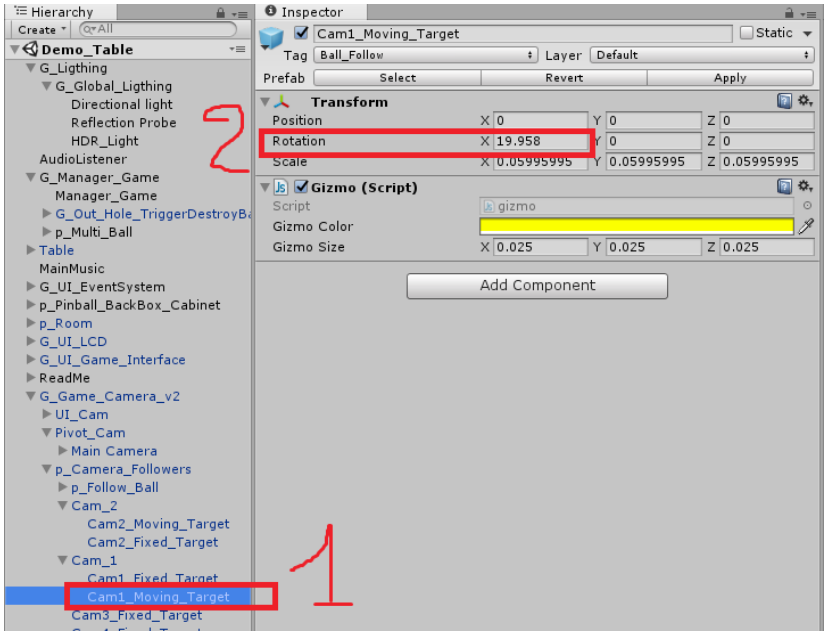
Check box DebugCamView (pic 2) IMPORTANT Don't forget to uncheck the box when you finish to tweak the camera)

Change cam 1_Pos Y and Cam1_Pos Z (pic 3)



Change rotation :

Select Cam1_Fixed_Target and change the rotation X on Inspector.



Do the same for Camera 2 with cam 2_Pos Y, Cam2_Pos Z and Cam2_Fixed_Target
for Camera 3 with cam 3_Pos Y, Cam 3_Pos Z and Cam3_Fixed_Target
for Camera 4 with cam 4_Pos Y, Cam 4_Pos Z and Cam4_Fixed_Target
for Camera Plunger with cam Plunger_Pos Y, Cam Plunger_Pos Z and Cam4_Plunger_Target

5.3 How to use Focus Camera ?

You find the focus view parameters on gameObject pivotCam (G_Game_Camera->pivotCam)

1 - When the ball enter on the first trigger camera is zooming
2 - When the ball enter on the second trigger camera zoom out to the last camera view

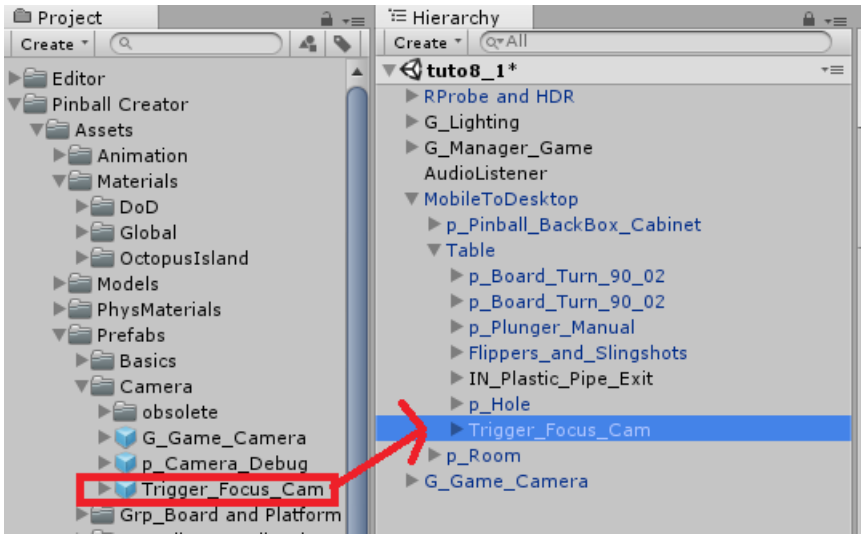
Example 1 : How to use FocusCam0 and FocusCam1 ?

Step 1 : Open Tuto8_1 (Asset->Tuto-> Tuto8_1)

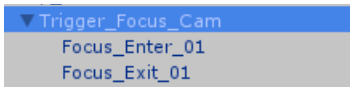
In this scene :
Ball is ejected outside the plunger and enter a hole.
Then the ball is eject from the pipe
You could press play and try the scene.

Step 2 : Activate FocusCam0 :

Drag and drop the gameObject Trigger_Focus_Cam inside gameObject Table on Hierarchy (Assets->Prefabs->Camera->Trigger_Focus_Cam)



Inside gameObject Trigger_Focus_Cam there are two chidren :
Focus_Enter_01 : When the ball enter this object the camera Zoom In
Focus_Exit_01 : When the ball enter this object the ball zoom out to the las camera view

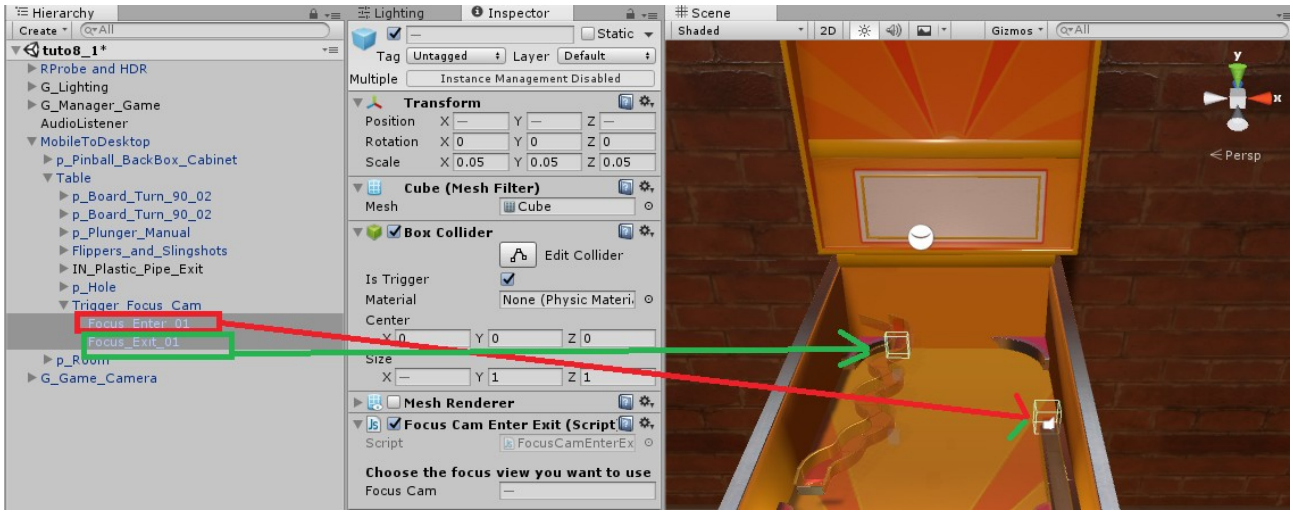


Step 3 : Choose where the camera zoom in and zoom out

Select gameObject Focus_Enter_01 on hierarchy and choose :
position X : 0.14
position Y : 0.03
position Z : -0.23

Select gameObject Focus_Exit_01 on hierarchy and choose :
position X : 0.36
position Y : 0.09
position Z : 0.1

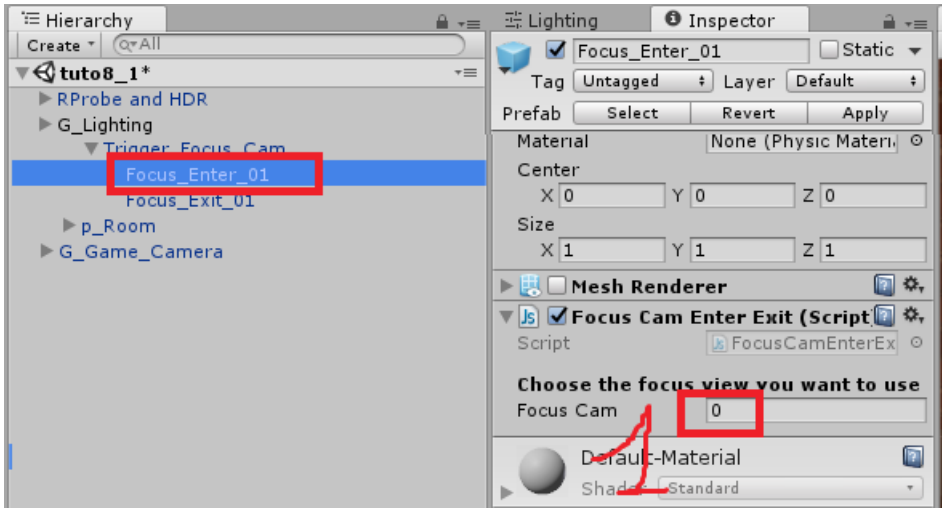
You need to have this :



Step 4 : Select the Focus Camera you want to use.

Select gameObject Focus_Enter_01 on hierarchy :
On script FocusCamEnterExit on Inspector Change the value of FocusCam parameter (pic 1)

if FocusCam = 0 you use FocusCam0
if FocusCam = 1 you use FocusCam1



Select gameObject Focus_Exit_01 on hierarchy :
On script FocusCamEnterExit on Inspector FocusCam parameter = -1.

-1 mean that you go back to the last camera view.

If you have a problem open scene Tuto8_2 (Asset->Tuto-> Tuto8_2)

5.4 How to use Change Focus Camera Position ?

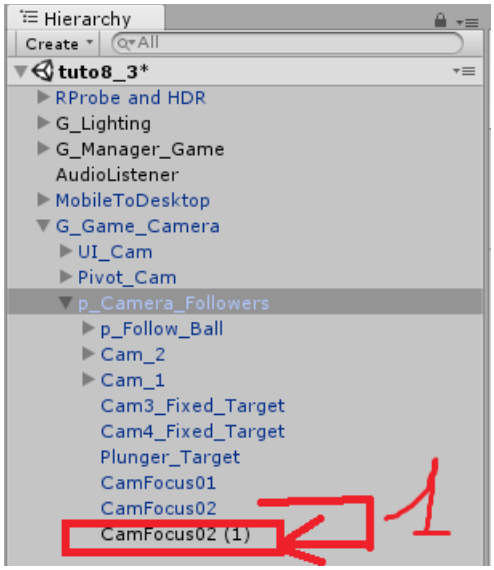
For the first Focus Cam :
Move and rotate the gameObject CamFocus01
For the second Focus Cam :
Move and rotate the gameObject CamFocus02

You find these two gameobjects on gameObject p_Camera_Followers (G_Game_Camera->p_Camera_Followers)

5.5 create a new Focus Cam

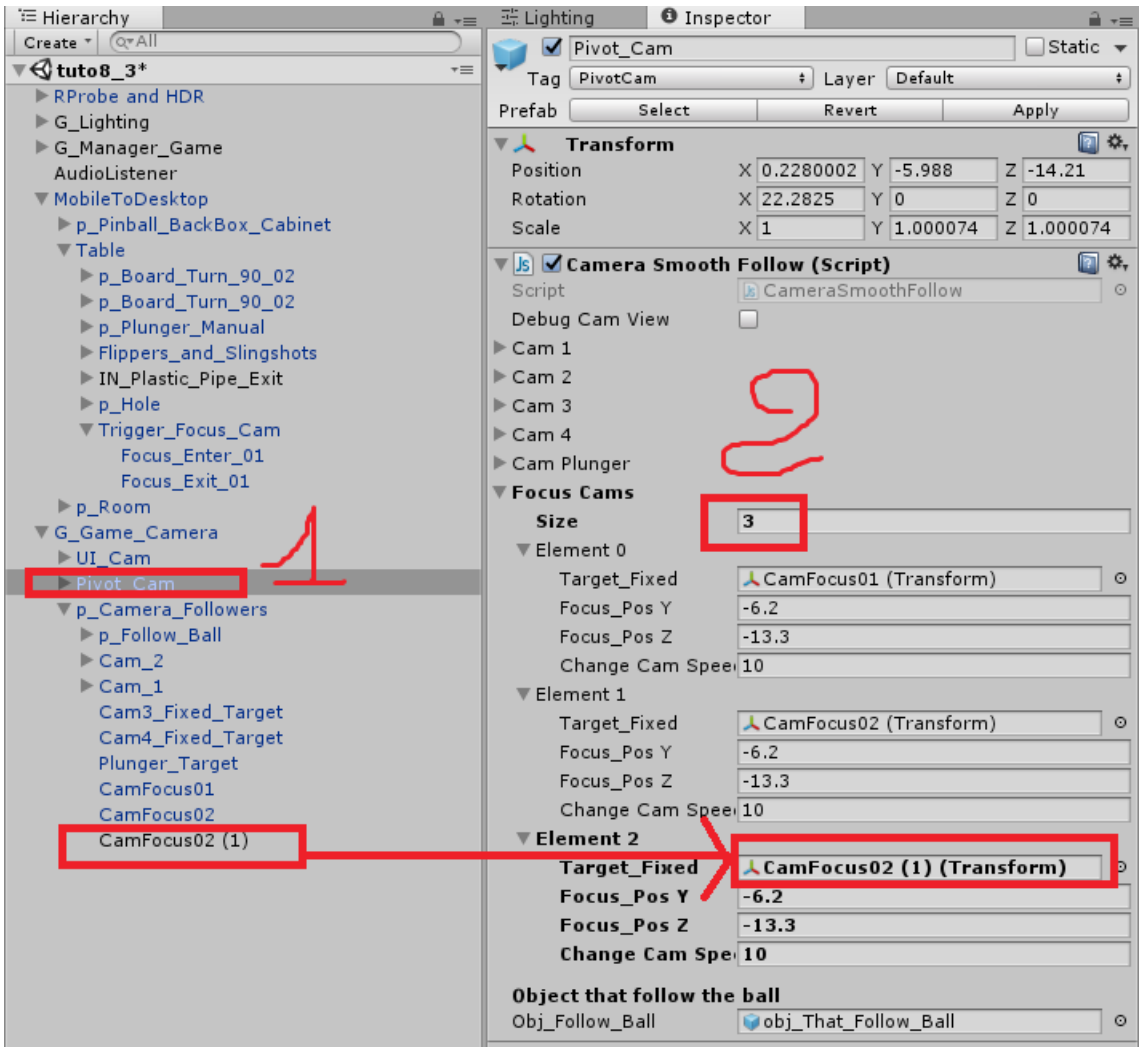
Step 1 : Open Tuto8_3 (Asset->Tuto-> Tuto8_3)

Duplicate CamFocus02 on the Hierarchy. (G_Game_Camera->p_Camera_Followers)
CamFocus02 (1) is created (pic 1)



Step 2 : Setup this new Cam

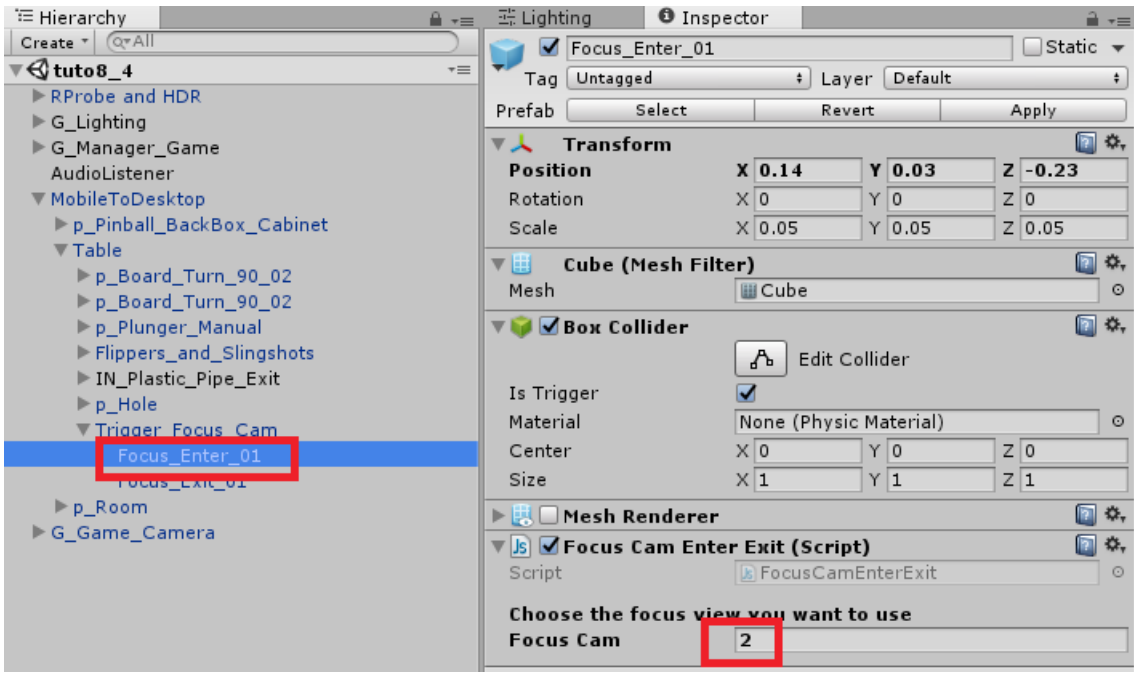
Select GameObject Pivot_Cam (G_Game_Camera->Pivot_Cam) (pic 1)
On Inspector Change Size to 3 On the FocusCams parameter (pic 2)
Drag and Drop CamFocus02 (1) Element 2-> Target_Fixed (pic 3)



Step 3 : Select Focus Cam number when the ball the trigger Focus_Enter_01

Select gameObject Focus_Enter_01 on the Hierarchy (MobileToDesktop->Table->Trigger_Focus_Cam-> Focus_Enter_01)

Change parameter FocusCam on the Inspector to 2 because we want to use Focus Cams 3 that we setup on the last step.



Step 4 : Choose your New position and rotation for the Focus Cam

Change CamFocus02 (1) position and rotation

Position : x = 0.15 y = 0 z = 0.04
Rotation : x = 34 y = 36 z = 0

Press Play to test

If you have a problem open scene tuto8_4

Camera : How to disconnect the camera system.

Delete the game Object named G_Game_Camera on the Hierarchy . That's it.
Then you could create a new camera. Menu --> GameObject --> Camera

2D orthographic camera :

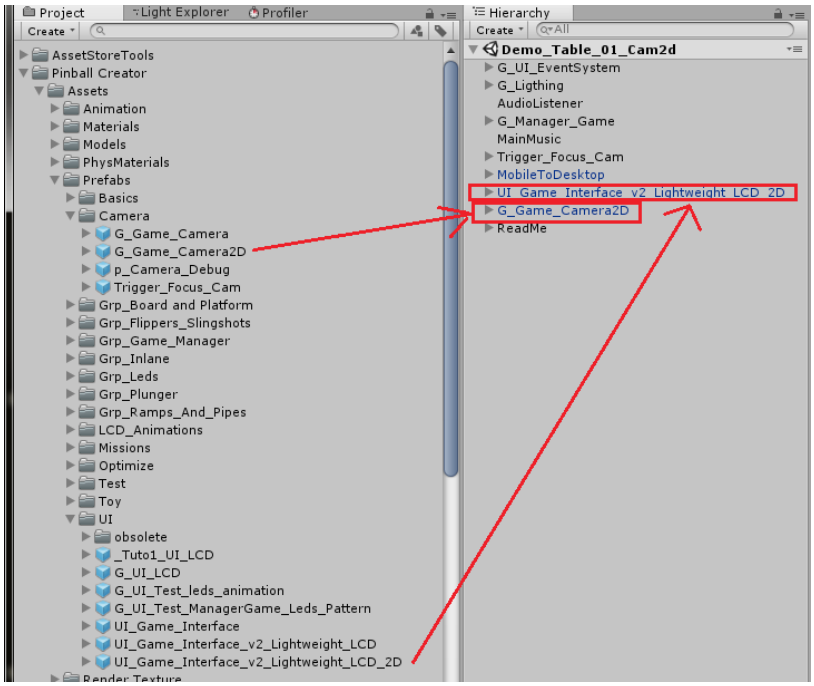
1a π How to use:

1.0 - Open scene Demo_Table_01_Cam2d (Assets->Demo->Demo 2D Camera-> Demo_Table_01_Cam2d)

This scene is setup to use 2D camera.

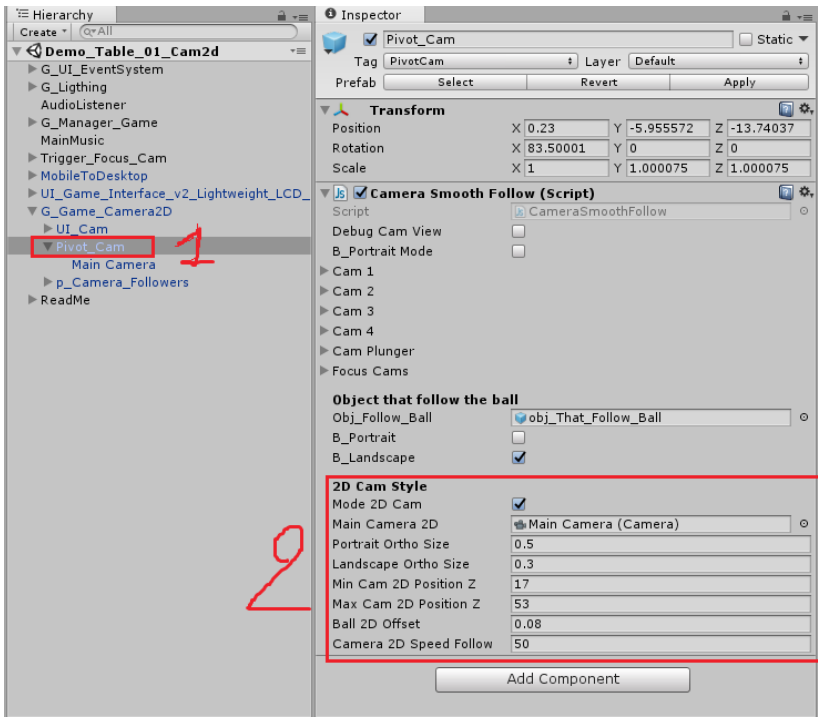
Camera 2D needs two prefabs to work:
- G_Game_Camera2D
- UI_Game_Interface_v2_Lightweight_LCD_2D

You find these prefabs here : (Assets->Prefabs->Camera-> G_Game_Camera2D) and (Assets->Prefabs->UI-> UI_Game_Interface_v2_Lightweight_LCD_2D)



1b - Camera 2D settings :

You could modify 2D camera on gameObject Pivot_Cam (see picture next page spot 1) (Hierarchy->G_Game_Camera2D->Pivot_Cam)

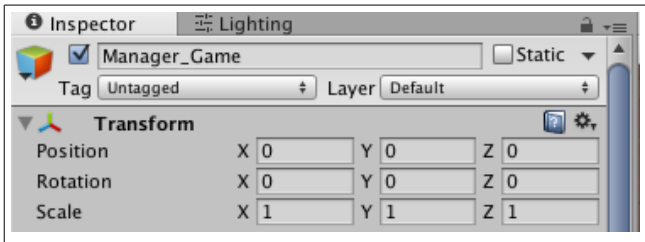
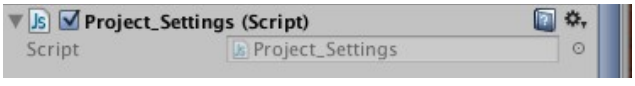


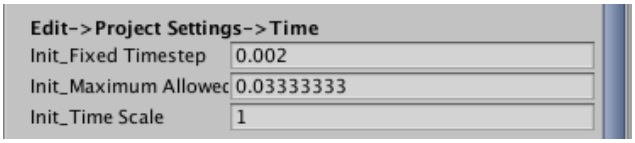
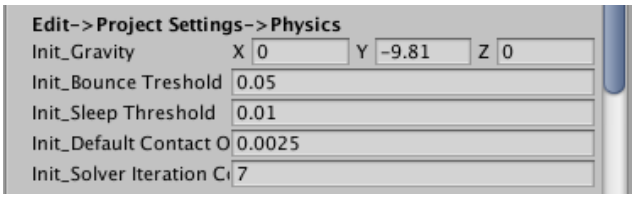
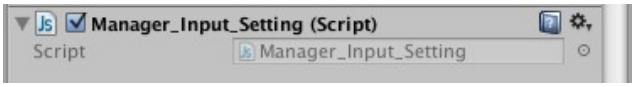
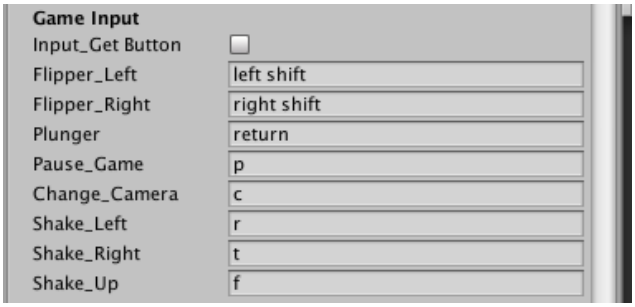
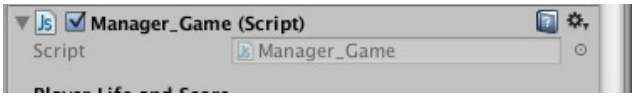
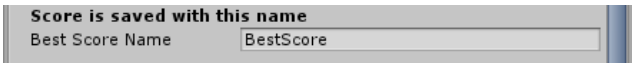
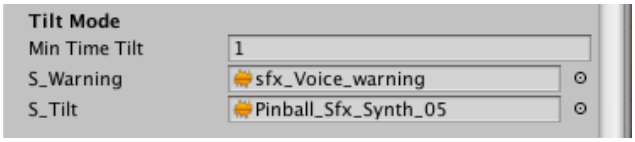
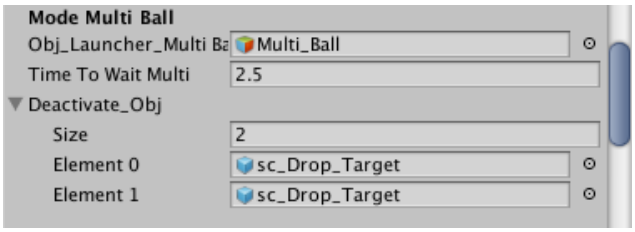
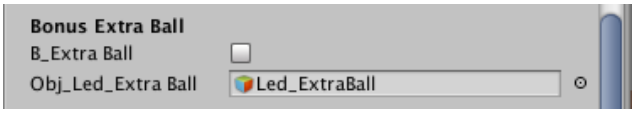
2D camera (spot 2) :

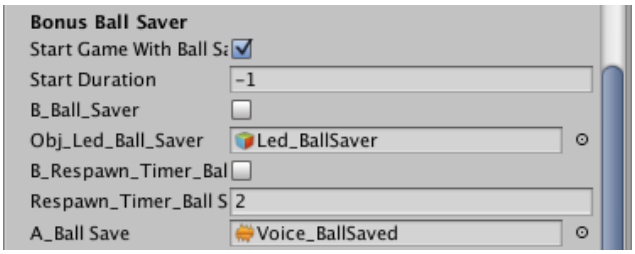
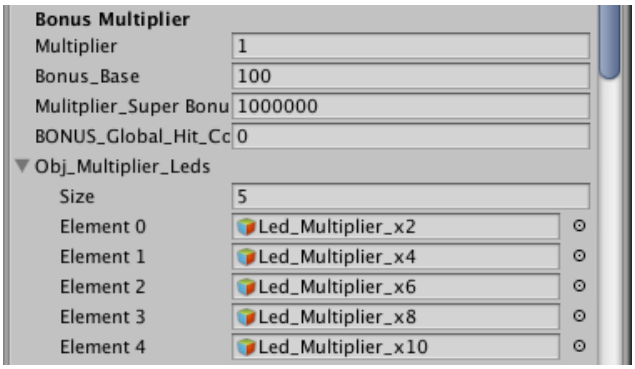
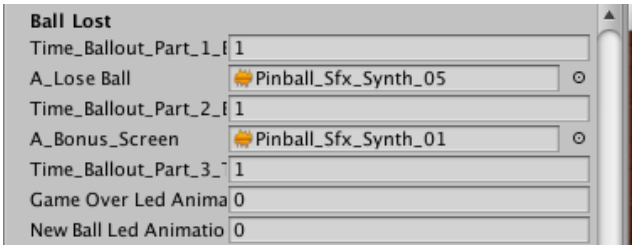
Mode 2D Cam : need to be checked
Main Camera 2D : Need to be connected
Portrait Ortho Size : Camera Orthographic size when a mobile device is on portrait mode
Landscape Ortho Size : Camera Orthographic size when a mobile device is on landscape mode
Min Cam 2D Position Z : Minimum Camera position
Max Cam 2D Position Z : Maximum Camera Position
Ball 2D Offset : Offset position between ball and camera

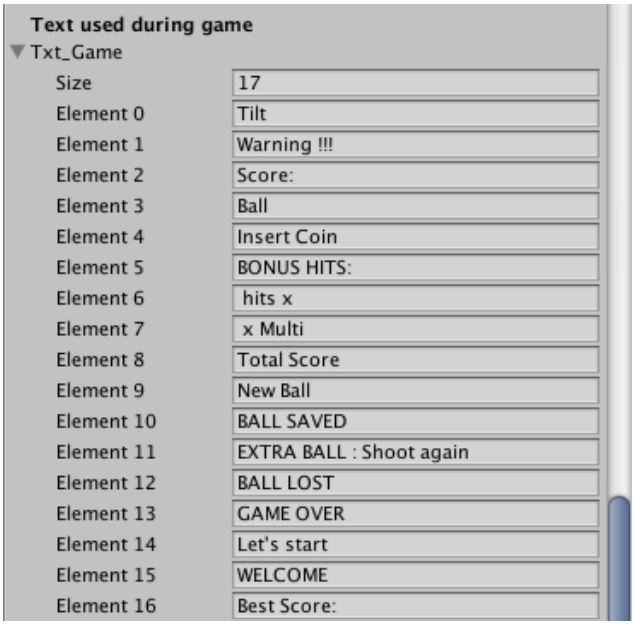
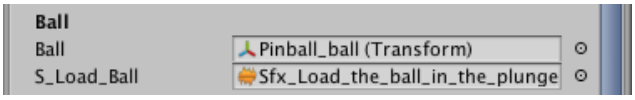
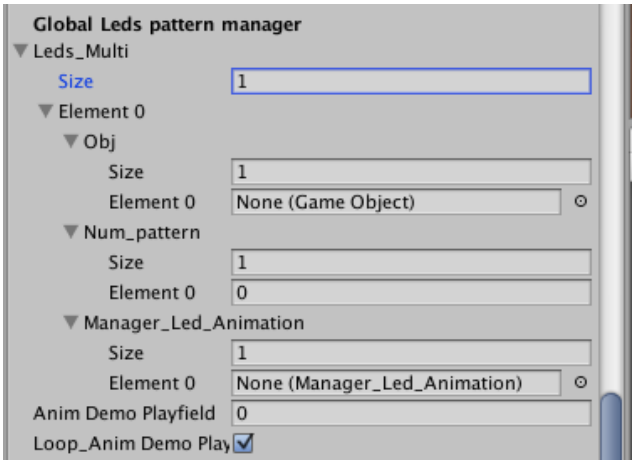
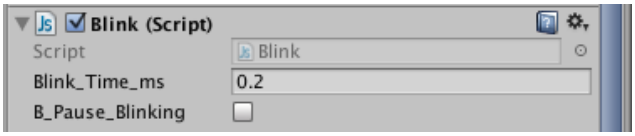
Game Manager :

On Hierarchy, game object Manager_Game manage ProjectSettings, Inputs and pinball rules. (Hierarchy -> G_Manager_Game -> ManagerGame) .

	
	Project_Settings.js Manage Project Settings : Time and Physics

	Project Settings : Time
	Project Settings -> Physics
	Manager_Input_Setting.js Manage Inputs
	Choose your Inputs. **Input_GetButton :** if True inputs use Edit -> Project Settings-> Input For example : If you want to use Fire 1 for Flipper_Left. Replace left shift with Fire 1 (more on setting inputs here)
	Manager_Game.js Manage game rules
	**BestScoreName :** Choose a name for the PlayerPrefs that save the score for this table. Choose a unique name For each Table
	**Life** : number of life
	Tilt Mode : (nudge technique) **MinTimeTilt** : Minimum time between two shakes. (seconds) **S_Warning :** Sound if the table was shaken **S_Tilt** : Sound if Tilt Mode Start (More about Tilt Mode)
	**Obj_Launcher_MultiBall :** Connect the gameObject Multi_Ball. **Time To Wait :** Time to wait before multiball start **DeactivatObj :** Connect target here if you want to activate and deactivate one or more targets when multiball start and stop
	Extra Ball : **B_ExtraBall** : true when Extra Ball Enable (automatically managed) **Obj_Led_ExtraBall** : Connect a led. Led Switch On

	when Extra ball is enable for the player. (more about how to connect a led) (more about ExtraBall)
	Ball Saver : **StartGameWithBallSaver** : Every ball start with Ball Saver activated **StartDuration** : Ball Saver duration. If StartDuration = -1 Ball Saver stop only when the ball is lost. **B_Ball_Saver** : if true Ball Saver is activated. (automatically managed) **Obj_Led_Ball_Saver** : Connect a led. Led Switch On when Extra ball is enable for the player. (more about how to connect a led) **B_Respawn_Timer_Ball_Saver** : use to respawn the ball. (automatically managed) **Respawn_Timer_Ball_Saver** : use to respawn the ball. **A_BallSave** : Play a sound when ball is lost **BallSaverLedAnimation** : Play Led Animation when ball is lost. 0 is the default animation. (more about how to play led animation during the game) (more about ball saver)
	Bonus Multiplier : **Multiplier** : current multiplier (read only) **Bonus Base** : (more about Bonus Score) **Multiplier_SuperBonus** : Bonus earned if multiplier > 10; **Bonus_Global_Hit_Counter** : How many ball hits pinball mechanics (more about Bonus Score) **Obj_Multiplier_Leds** : Connect leds. Important. Connect 0 led or 5 leds. Other size may create bugs. (more about Bonus Score) (See section Score for information about player score).
	Ball Lost : **Time_Ballout_Part_1** : After this time Bonus Score is displayed on LCD. **A_LoseBall** : Play a sound when ball is lost. **Time_Ballout_Part_2** : After this time Score is displayed on LCD. **A_Bonus_Screen** : Play a sound for during Bonus Score. **Time_Ballout_Part_3** : After this time , New ball" or , Game Over" is displayed on LCD. **GameOverLedAnimation** : play Leds animation if player is game Over. -1 mean no animation. 0 is the default animation. (more about animation) **NewBallLedAnimation** : play Leds animation if player lose a ball. -1 mean no animation. 0 is the default animation. (more about animation)

	Txt_Game[0] : Tilt Txt_Game[1] : Warning Txt_Game[2] + player_Score : Display the score when there is nothing else to display Txt_Game[3] + (Ball_num+1) : Display the the ball number Txt_Game[4] : Display a text when we are wait a player start the game Bonus text when player lose a ball (There is 3 parts) Part1 : Txt_Game[5], Txt_Game[6], Txt_Game[7] : Txt_Game[5] + "\n" + tmp_BONUS_Global_Hit_Counter + Txt_Game[6] + Bonus_Base + "\n" + tmp_Multiplier + Txt_Game[7], Time_Ballout_Part_2_Bonus); Part 2 : Txt_Game[8] : Txt_Game[8] + "\n" + player_Score.ToString() (display the player score) Part 3 : Txt_Game[9] : Display this text if player life > 0 Txt_Game[10] : Text for ball saver Txt_Game[11] : Text for Extra ball Txt_Game[12] : Text for Ball lost Txt_Game[13] : Text for Game Over Txt_Game[14] : Text when a game start Txt_Game[15] : Text when the scene start
	Ball : **Ball** : Connect here ball prefab. **S_Load_Ball** : Play this sound when ball is load on plunger.
	(More here)
	(automatically managed)
	UI : **Game_UI** : Connect Project -> Assets -> prefabs -> UI -> G_UI_Game_interface. (more here)
	Blink : **Blink_Time_ms** : Choose global blinking time (ms) **B_Pause_Blinking** : (automatically managed)
	Audio Component

BallSaver :
Ball is ejected on table, if the player lose the ball and if the Ball Saver is activated on Manager_Game. (more about BallSaver options)

Best score :
Best score is saved on PlayerPrefs(, BestScore"). You could call it with PlayerPrefs.GetInt(, BestScore"). (See section Score for information about player score).

Blink :
Blink system allow you to synchronize all the leds and make them blinking.

How to :

1 π Choose the blinking time you want with Blink_Time_ms (You find this variable on Manager_Game in the Hierarchy. Blink.js)

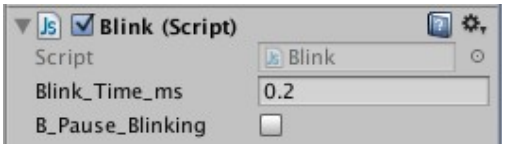
2 π Check box B_Blinking on each Led you want to blink. (You find this variable on each Led object. Change_Sprite_Renderer.js)

How it works.

a) When scene start, if B_Blinking = true the led change tag to Blink

b) When scene start, blink.js check every object with the tag Blink.

c) Every Blink_Time_ms blink.js send a message to leds with tag Blink. Leds blink only if they are switch On.



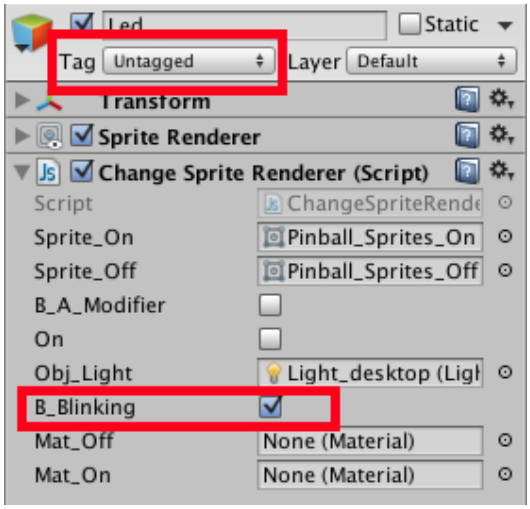
You find this script on Manager_Game in the hierarchy

Blink_Time_ms

Choose your blinking time. It's the same for every Leds on scene

B_Pause_Blinking

Used if you want to pause blinking leds. (automatically managed)
You find this script on each Led gameobject



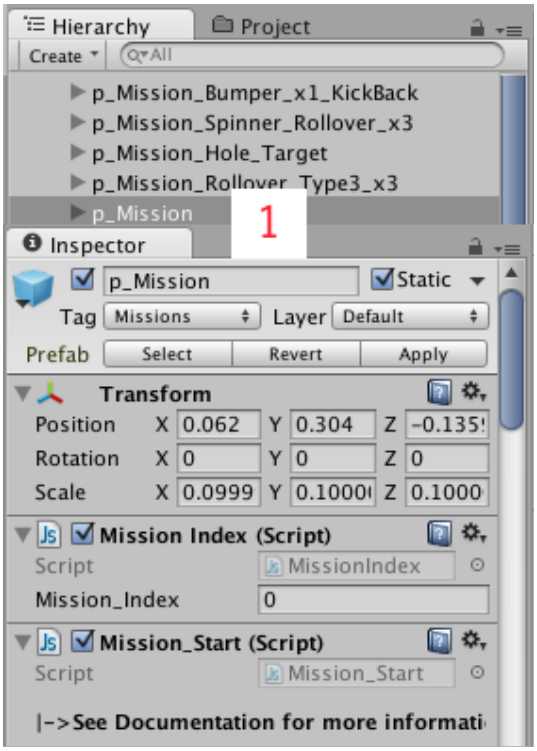
B_blinking

If true the will be blink when the scene start

Bonus Score :
During game, player earned Bonus points.
It is determined when player lose a ball.
Bonus Score is displayed on LCD SCREEN at the end of the ball.
Bonus Score = How many table mechanics has been hit during the ball x Bonus Base x Multiplier

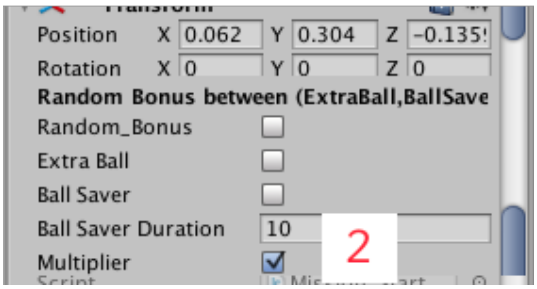
A mission could increase Multiplier when the mission is completed.

How to configure a mission to increase Multiplier ?



Select a mission on Hierarchy (pic 1)

Selection Option Multiplier (pic 2)
IMPORTANT : You could choose only one option

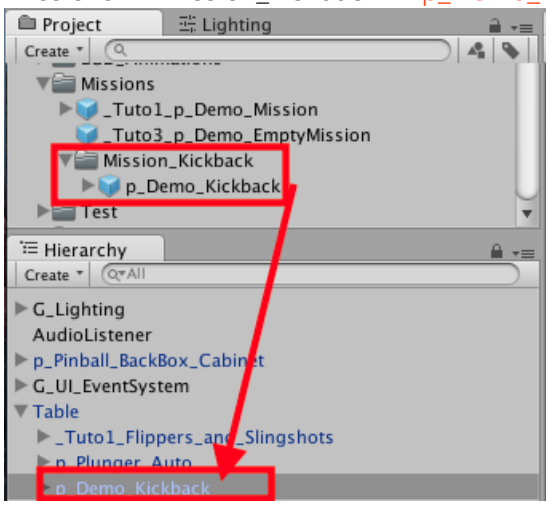


Multiplier increase : x2 x4 x6 x8 x10.
If Multiplier > 10 player earn the SuperBonus.

ExtraBall :
Ball is saved if ExtraBall is activated.
To activate an extra ball check the box ExtraBall inside a mission (more here)

Kickback :
Project -> Assets -> Prefabs -> Basics -> Hole -> p_KickBack
It's a feature that saves the ball and eject it back into playfield.
Some missions using kickback is included inside folder Project -> Assets -> Prefabs -> Missions -> z_Missions_Special_Kickback

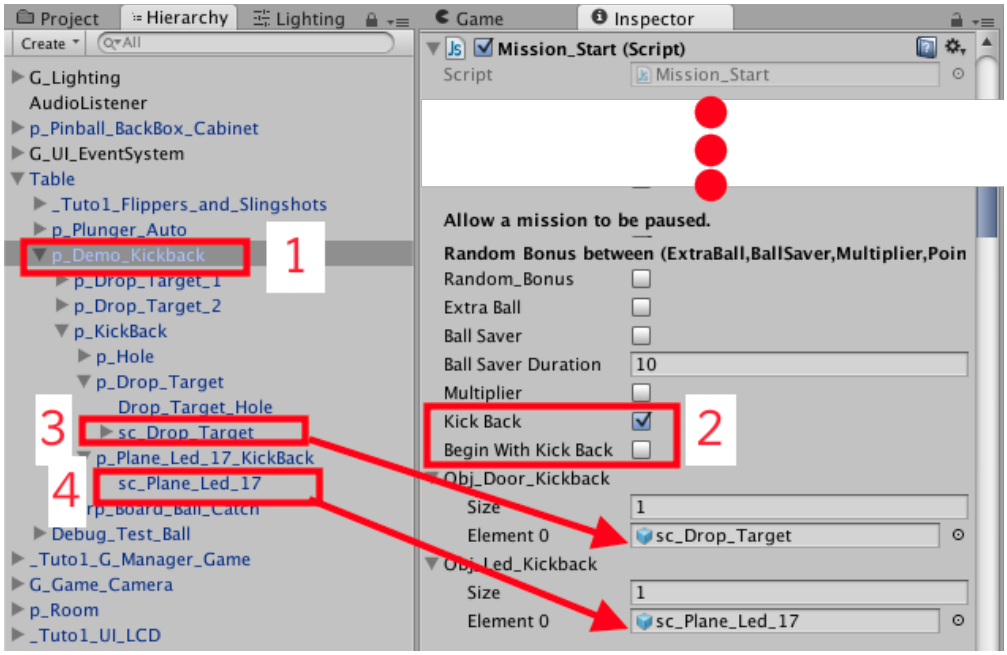
Example : How to use and connect kickback
Step 1 : Drag'n'drop p_Demo_Kickback inside Table on the Hierarchy (Project -> Assets -> Prefabs -> Missions -> Mission_kickback -> p_Demo_Kickback)



Step 2 : Select p_Demo_kickback. Inside you find the script Mission_Start.js (pic 1)

Step 3 : Check box Kickback (pic 2). When the player finish this mission kickback is enabled.
Check box Begin with kickback if you want to enable kickback when the player start playing a ball.

Step 4 : drag'n'drop sc_Drop_Target inside Obj_Door_Kickback (pic 3)
drag'n'drop sc_Plane_Led_17 inside Obj_Led_Kickback (pic 4)
Info : You could put more than one kickback inside Obj_Door_Kickback and Obj_Led_Kickback.



Multi-ball :
(more about multi-ball)

Score :
Score is manage by Manager_Game on Hierarchy. (Manager_Game.js)

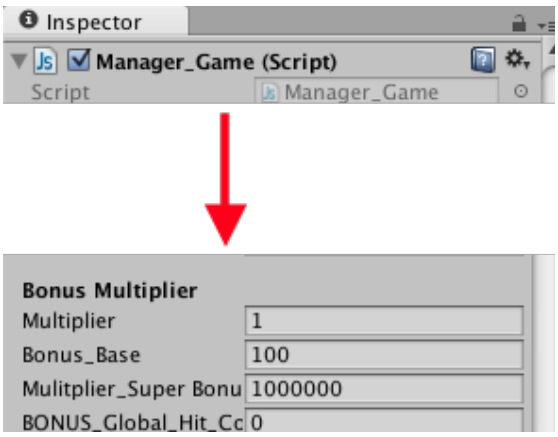
The score is calculated by adding :
- Points earned when ball hit a table mechanic on playfield.
- Points earned when a mission is completed.
- Points earned with Bonus Score (more info)

After the game has ended, Manager_Game.js check if player score is bigger than Best Score. If true, New Best score is saved on PlayerPrefs(, BestScore").
Maximum score is 999999999 points.

How to modify points earned by player.

Tables Mechanics : In every table mechanic you could choose points earned when ball hits this object.

Default points : 1000.

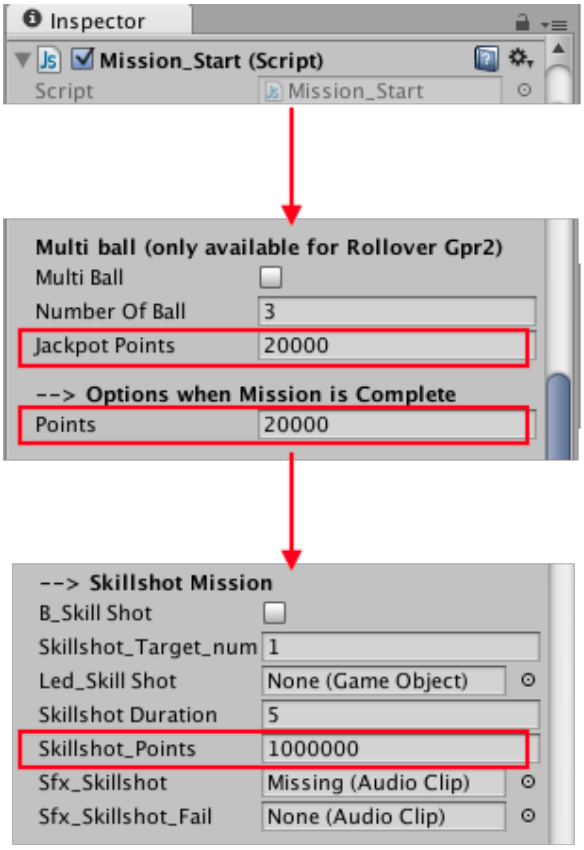


Inside Manager_Game (Manager_Game.js) **:**

Section : Bonus Multiplier

Bonus_Base : (more info)

Multiplier_SuperBonus : Points earned if Multiplier = 10



Inside a misison : (script Mission_Start.js)

Section : Multi-Ball
JackpotPoints : Points earned when multi-ball mode is on. (more about multi-ball)

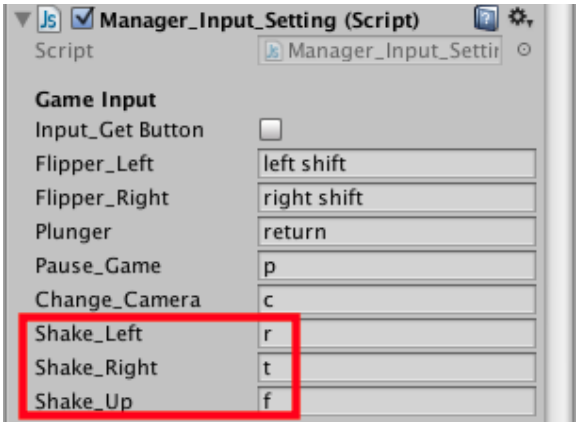
Section : Options when mission is complete
Points : Points eared when player complete the mission

Section : Skillshot Mission
Skillshot_Points : Points earned when player hit the skillshot table mechanic
(more about Skillshot here)

Skillshot :
(more about Skillshot here)

Tilt :
How Tilt Mode Works.
1- A warning message appears the first time the player hits the table (nudge technique). A force is applied to the ball. Camera play an animation.

2- The table starts Tilt mode, if the player hits the table again (immediately after the first shake). The flippers are locked and all the balls are ejected toward the exit.

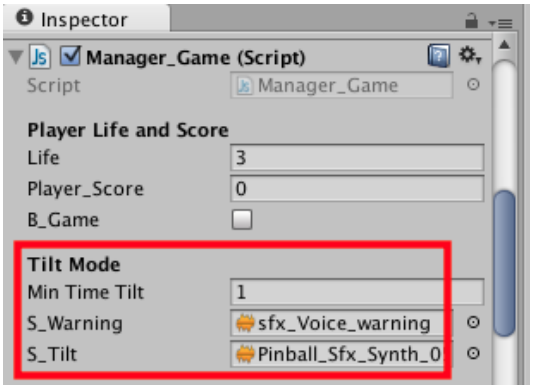


Player can hit the machine by pressing key input.

By default :

Hit Left : r
Hit Right : t
Hit Up : f

You can modify the input key on script Manager_Input_Setting.js on gameObject Manager_Game (hierarchy).



Tilt Mode is manage by script Manager_Game.js on gameObject Manager_Game (hierarchy).

MinTimeTilt : The minimum time between two hit on the table.

S_Warning : play a sound when player hit the table

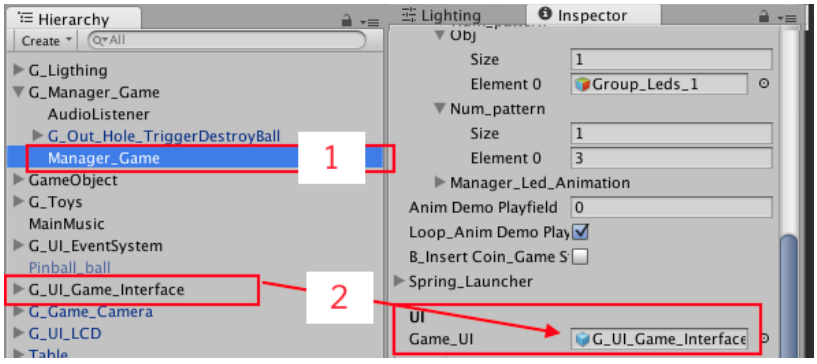
S_Tilt : play a sound when tilt mode start

UI Interface :
Project -> Assets -> prefabs -> UI -> G_UI_Game_Interface
Use this prefab to have a simple UI Interface.

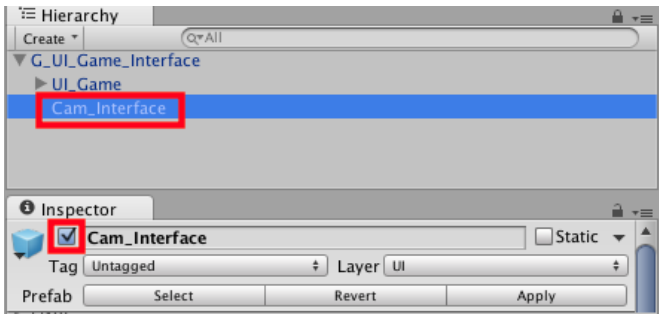
Important : Don't change buttons name inside the prefab. These names are used by script manager_Game.js

How to connect the UI Interface.
Step 1 : drag'n'drop G_UI_Game_Interface on Hierarchy.

Step 2 : Select Manager_Game on Hierarchy (pic 1). Drag'n'drop G_UI_Game_Interface inside Game_UI on the Inspector. (See picture next page)



Step 3 (if needed) : By default the UI Interface is renderer by the camera that you can find on G_UI_LCD Prefab (Project -> Assets -> prefabs -> UI -> LCD). So if you don't use this prefab, enable the camera Cam_Interface inside prefab G_UI_Game_Interface (pic 1)


Hierarchy
Create
All
G_UI_Game_Interface
UI_Game
Cam_Interface
Inspector
Cam_Interface
Static
Tag
Untagged
Layer
UI
Prefab
Select
Revert
Apply

LCD Screen :

Choose between the two camera system :

You could use :

The lightweight UI interface + LCD (prefab UI_Game_Interface_v2_Lightweight_LCD) It's the UI interface + LCD

Pros : Increase FPS

Cons : LCD is not display on the cabinet.

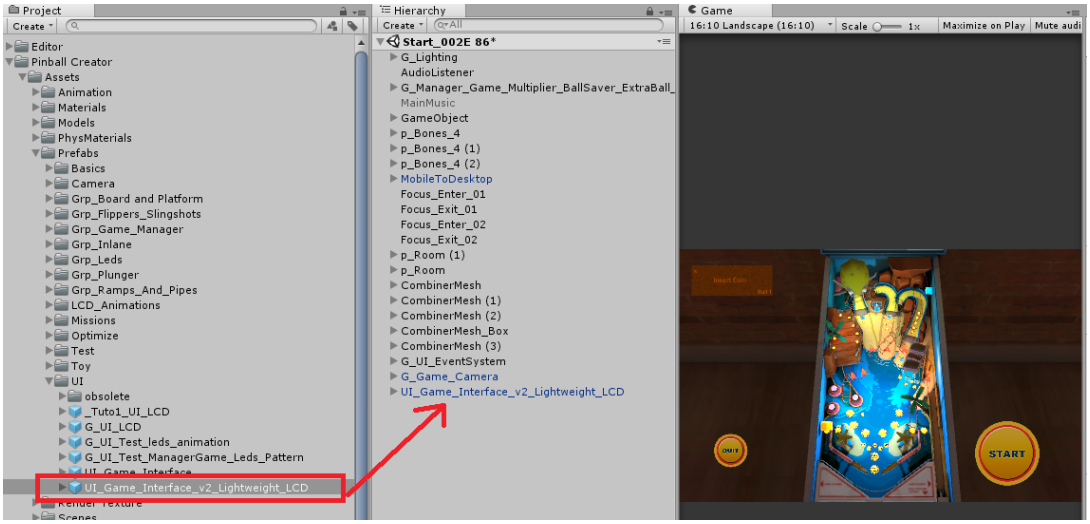
The UI system v1 (Optimize) associate to LCD screen v1 (prefab UI_Game_Interface + prefab G_UI_LCD)

Pros : LCD is display on the cabinet.

Cons : Loss about 5 FPS on mobile device

How to connect The lightweight UI interface + LCD :

Drag and drop UI_Game_Interface_v2_Lightweight_LCD on Hierarchy (assets->Prefabs-> UI-> UI_Game_Interface_v2_Lightweight_LCD)



How to connect The UI system v1 (Optimize) associate to LCD screen v1 :

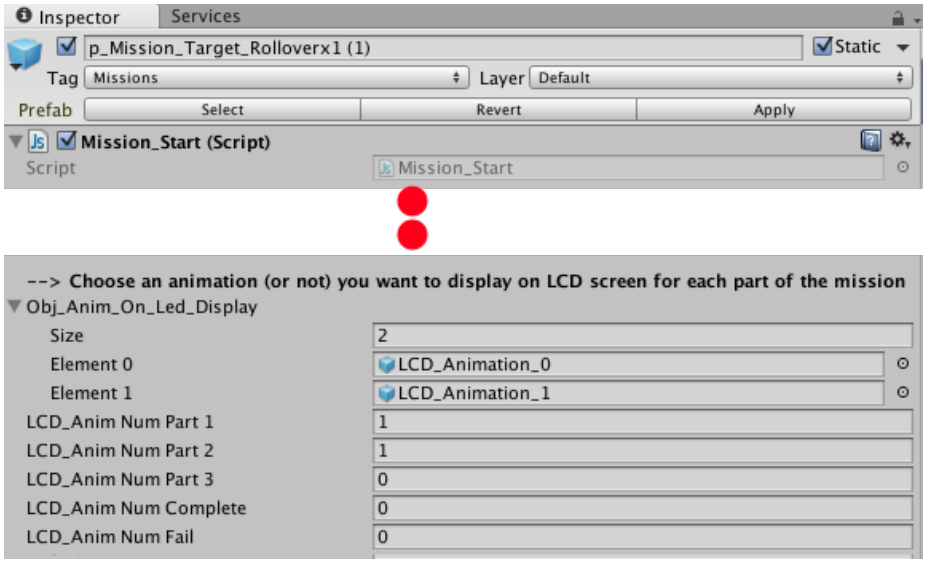
Drag and drop UI_Game_Interface on Hierarchy (assets->Prefabs-> UI-> UI_Game_Interface) (pic 1)

Drag and drop G_UI_LCD on Hierarchy (assets->Prefabs-> UI-> G_UI_LCD) (pic 2)

How to invoke Animation for LCD Screen.

Animation for LCD Screen is invoke from the mission. (script Mission_Start.js inside a mission).

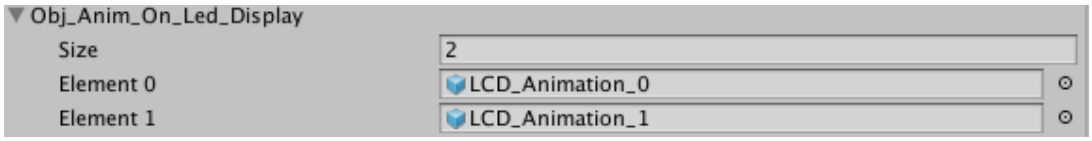
Step 1 : Look at section : ''--> Choose an animation (or not) you want to display on LCD Screen for each part of the mission''



Step 2 : Choose LCD animations you want to use for mission
In the next example you use two animations.

Put Size = 2 and drag'n'drop the needed animations inside Obj_Anim_On_Led_Display -> Element 0 and Element 1.

You could use LCD_Animation_0 and that you could find on Project -> Assets -> Prefabs -> LCD_Animations .

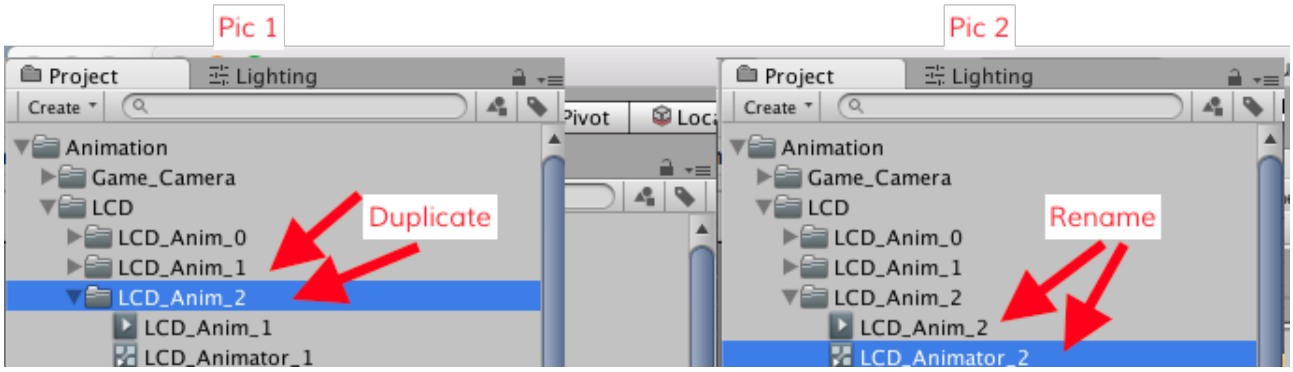


Step 3 : Choose LCD animations you want to play for each mission part.
You could play an animation for each mission part.

In the next example :
0 correspond to animation Element 0 : LCD_Animation_0
1 correspond to animation Element 1 : LCD_Animation_1.
Important : -1 mean no animation for this part.

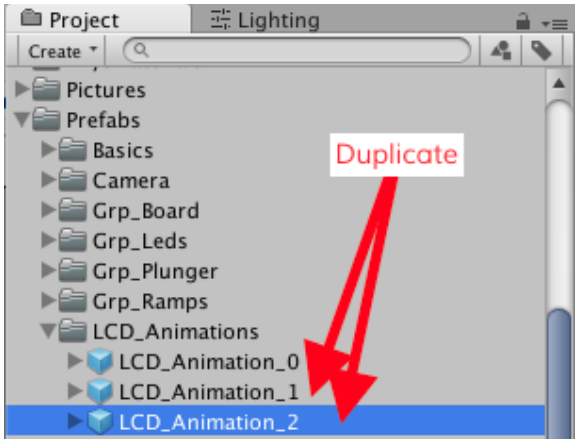
LCD Screen : Create Animation for LCD Screen (if you use G_UI_LCD prefab).

Step 1 : Preparation of elements Part 1
Duplicate LCD_Anim_0 (Project -> Assets -> Animation -> LCD -> LCD_Anim_0)
A new folder LCD_Anim_2 is created (pic 1).
Rename files inside the folder. Change LCD_Anim_0 to LCD_Anim_2 and LCD_Animator_0 to LCD_Animator_2 (pic2).



Step 2 : Preparation of elements Part 2

Duplicate LCD_Animation_0 (Project -> Assets -> Prefabs -> LCD_Animations -> LCD_Animation_0) LCD_Animation_2 is created.



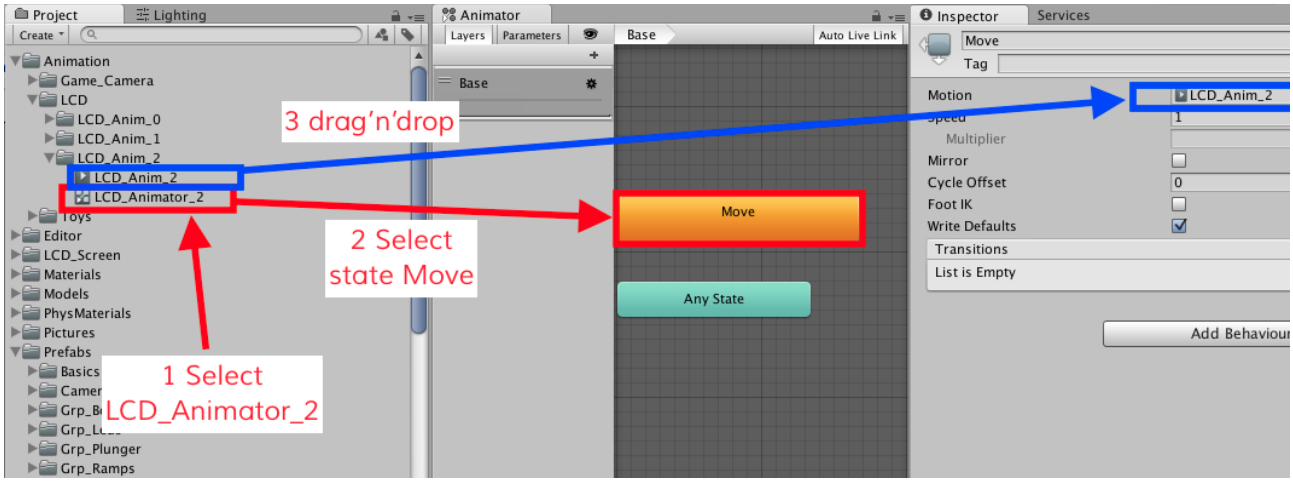
Step 3 : Connect animation on Animator

Open Animator window.

Select LCD_Animator_2 on Project folder (pic 1).

Select Animation State named Move (pic 2).

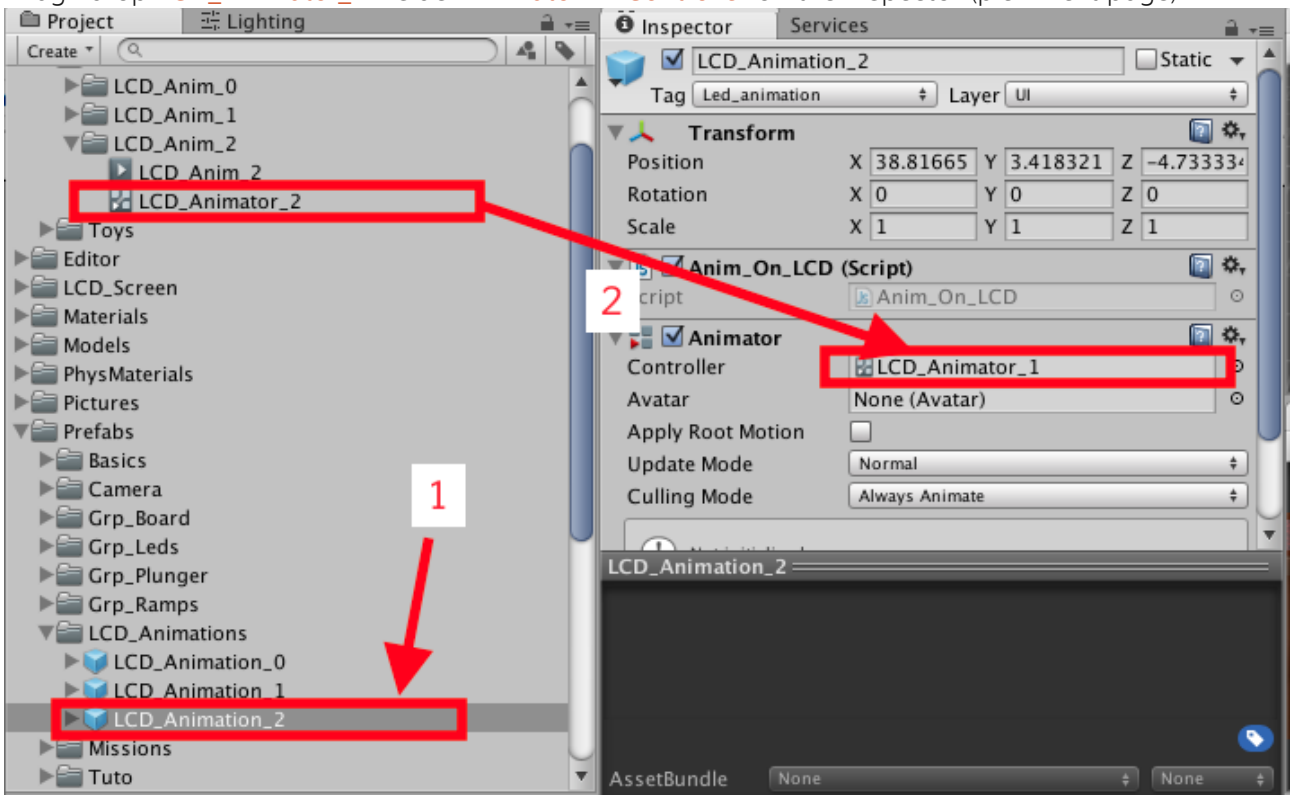
Drag'ndrop LCD_Anim_2 inside Motion on the Inspector (pic 3).



Step 4 : Connect Controller on LCD Animation Prefab

Select LCD_Animation_2 on Project folder (pic 1 next page)

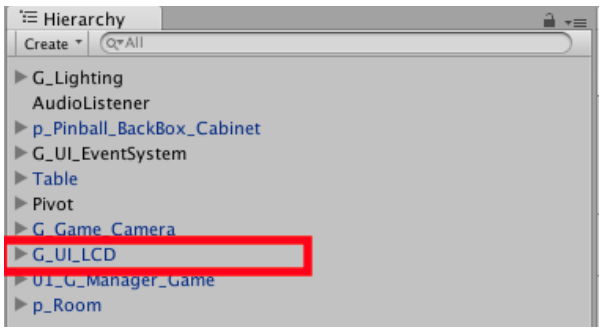
Drag'ndrop LCD_Animator_2 inside Animator -> Controller on the Inspector (pic 2 next page).



Step 5 : Put the new animation on scene

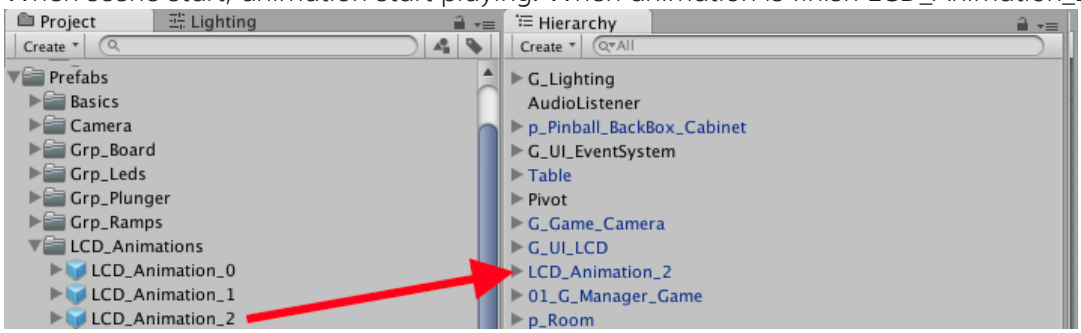
First Check if you have the prefab G_UI_LCD on Hierarchy.

If you don't have this object. Drag'n'drop G_UI_LCD from Project -> Assets -> Prefabs -> UI -> G_UI_LCD in the root Hierarchy folder.



Drag'n'drop LCD_Animation_2 in the root of Hierarchy. On scene Press , f" to focus on gameObject LCD_Animation_2.
Press play. You should see two spheres.

When scene start, animation start playing. When animation is finish LCD_Animation_2 is destroy.



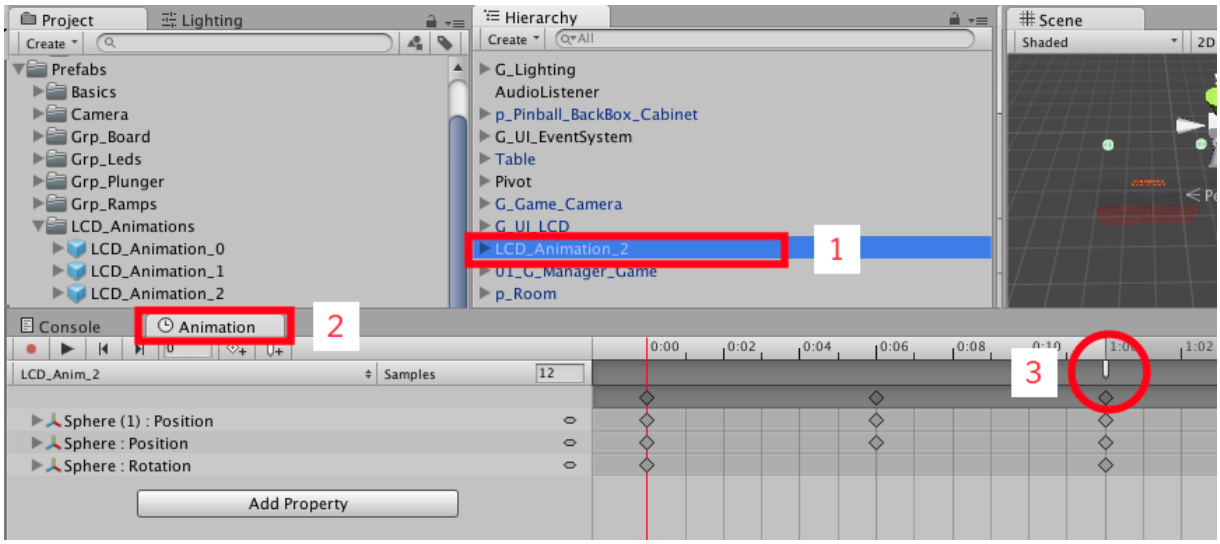
Step 6 : Create a new animation

Open Animation Window

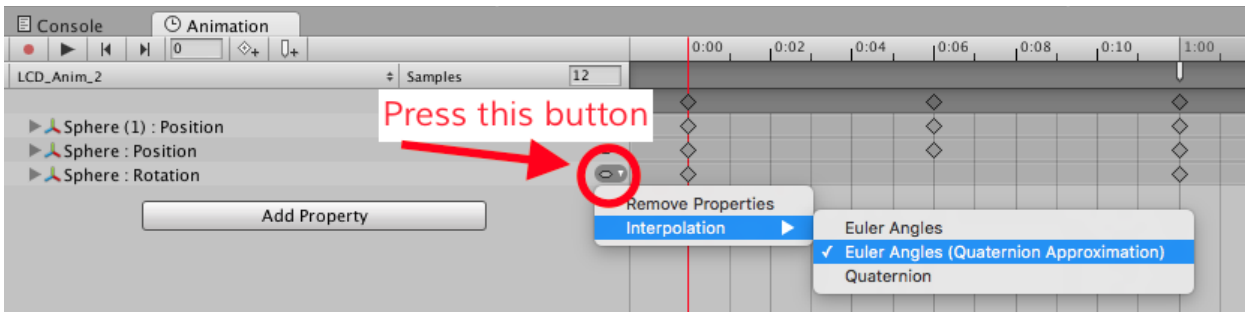
Then select LCD_Animation_2 on Hierarchy (pic 1).

You could modify animation (pic 2).

Important. Don't delete the event (pic 3) because it used to destroy the gameObject when animation is finish.

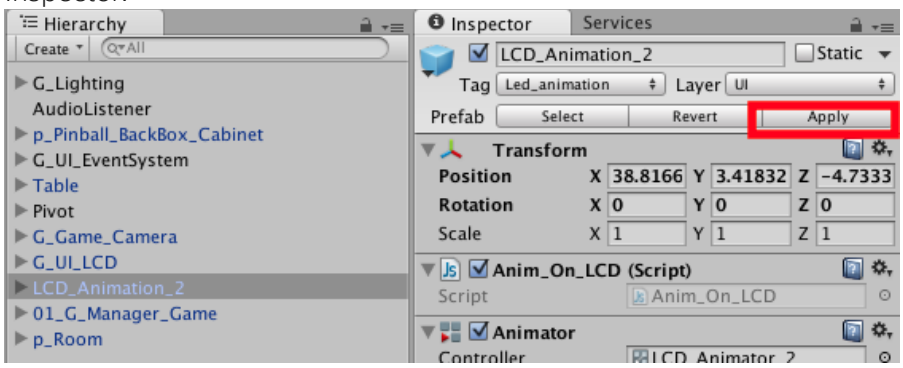


Tips : If you modify objects rotation choose Interpolation -> EulerAngles(Quaternion Approximation).



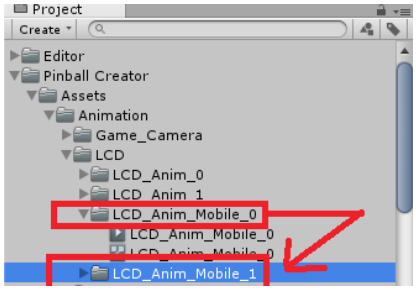
Step 7 : You could now delete the spheres. Create your own objects. Put them inside LCD_Animation_2 on Hierarchy. Create your animation.
Tips : To see your animation : Open Window -> Game . Animation appears on LCD Screen.

Step 8 : Important. When it's done. Select LCD_Animation_2 on Hierarchy and press Apply on the Inspector.



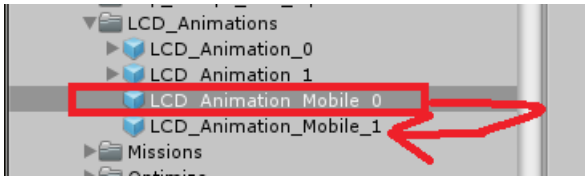
LCD Screen : Create Animation for LCD Screen (if you use UI_Game_Interface_v2_Lightweight_LCD prefab).

Step 1 : Preparation of elements Part 1
Duplicate LCD_Anim_Mobile_0 (Project -> Assets -> Animation -> LCD -> LCD_Anim_Mobile_0)
A new folder LCD_Anim_Mobile_1 is created (pic 1).



Step 2 : Preparation of elements Part 2

Duplicate LCD_Animation_Mobile_0 (Project -> Assets -> Prefabs -> LCD_Animations -> LCD_Animation_Mobile_0)
LCD_Animation_Mobile_1 is created.



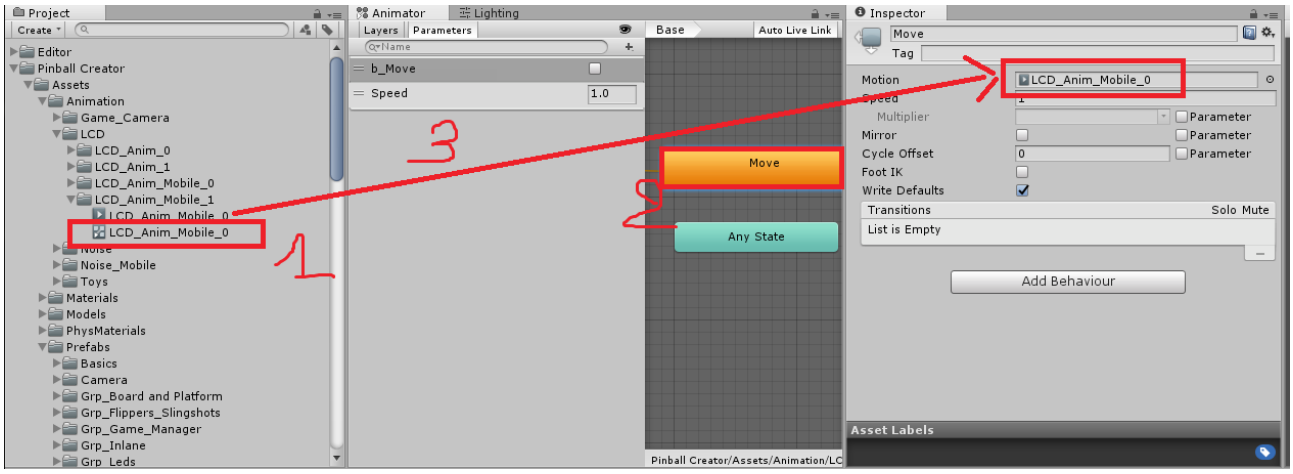
Step 3 : Connect animation on Animator

Open Animator window.

Select LCD_Anim_Mobile_0 on Project folder (pic 1).

Select Animation State named Move (pic 2).

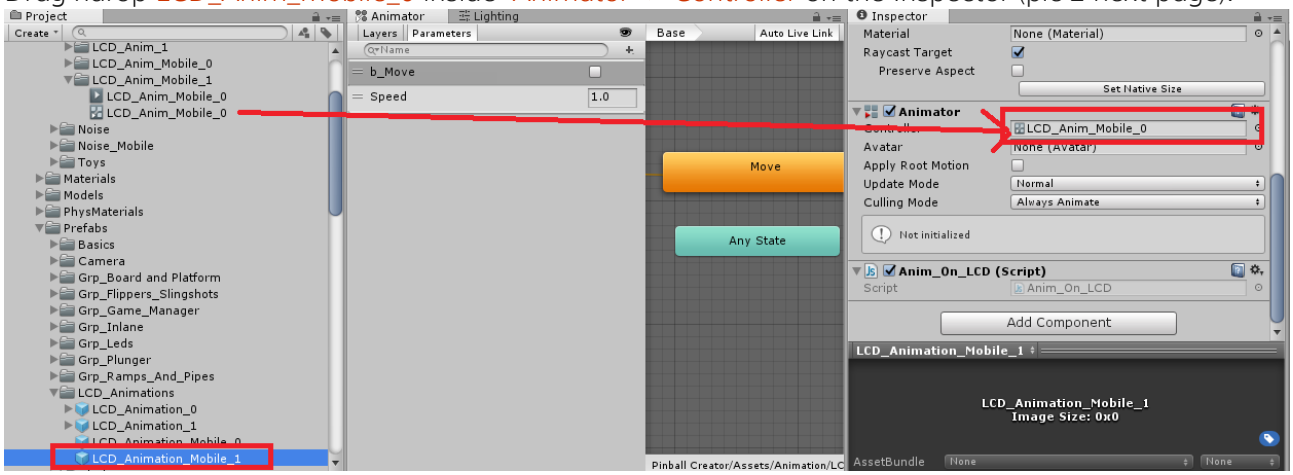
Drag'ndrop LCD_Anim_Mobile_0 inside Motion on the Inspector (pic 3).



Step 4 : Connect Controller on LCD Animation Prefab

Select LCD_Animation_Mobile_1 on Project folder (pic 1 next page)

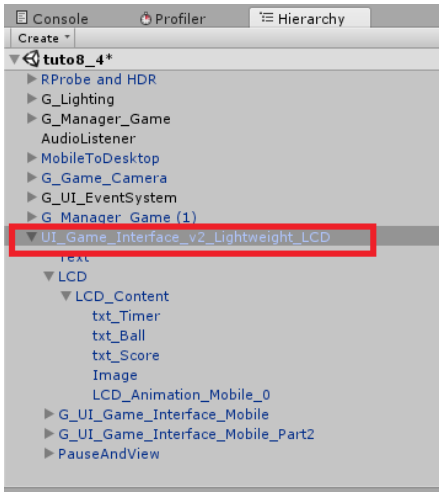
Drag'ndrop LCD_Anim_Mobile_0 inside Animator -> Controller on the Inspector (pic 2 next page).



Step 5 : Put the new animation on scene

First Check if you have the prefab G_UI_LCD on Hierarchy.

If you don't have this object. Drag'n'drop UI_Game_Interface_v2_Lightweight_LCD from Project -> Assets -> Prefabs -> UI -> UI_Game_Interface_v2_Lightweight_LCD in the root Hierarchy folder.



Drag'n'drop LCD_Animation_Mobile_1 in the root of Hierarchy. On scene Press , f" to focus on gameObject LCD_Animation_2.
Press play. Animation is playing

When scene start, animation become child of gameObject LCD_Content (Hierarchy->UI_Game_Interface_v2_Lightweight_LCD->LCD->LCD_Content->) start playing. When animation is finish LCD_Animation_Mobile_1 is destroy.

Step 6 : Create a new animation

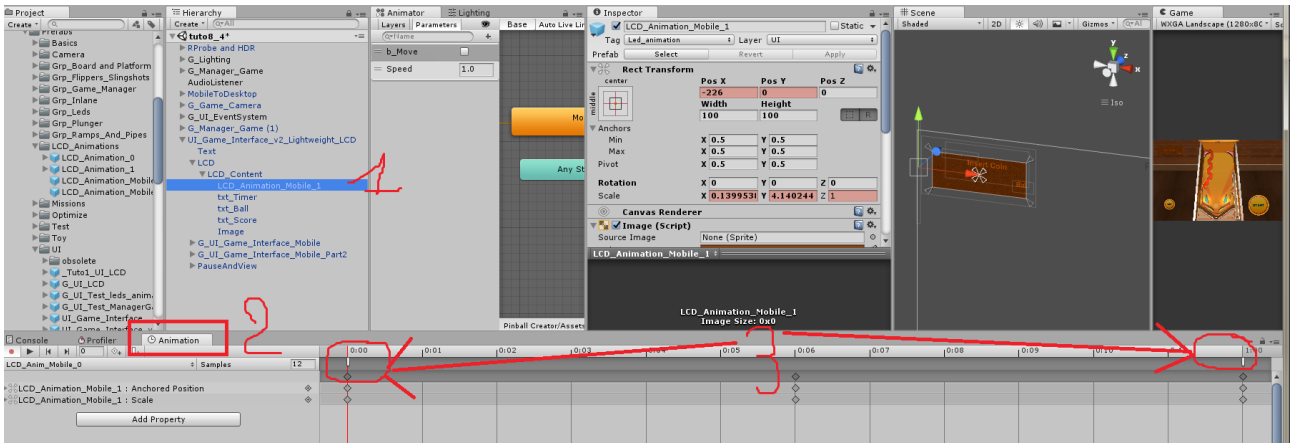
Open Animation Window

Then select LCD_Animation_Mobile_1 on Hierarchy.

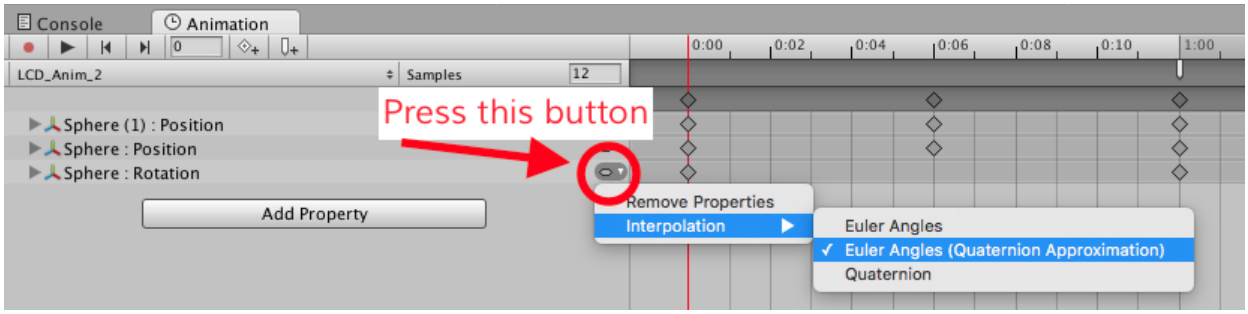
Drag and drop LCD_Animation_Mobile_1 inside (Hierarchy->UI_Game_Interface_v2_Lightweight_LCD->LCD->LCD_Content->) pic 1

You could modify animation (pic 2).

Important. Don't delete the event (pic 3) because it used to destroy the gameObject when animation is finish.

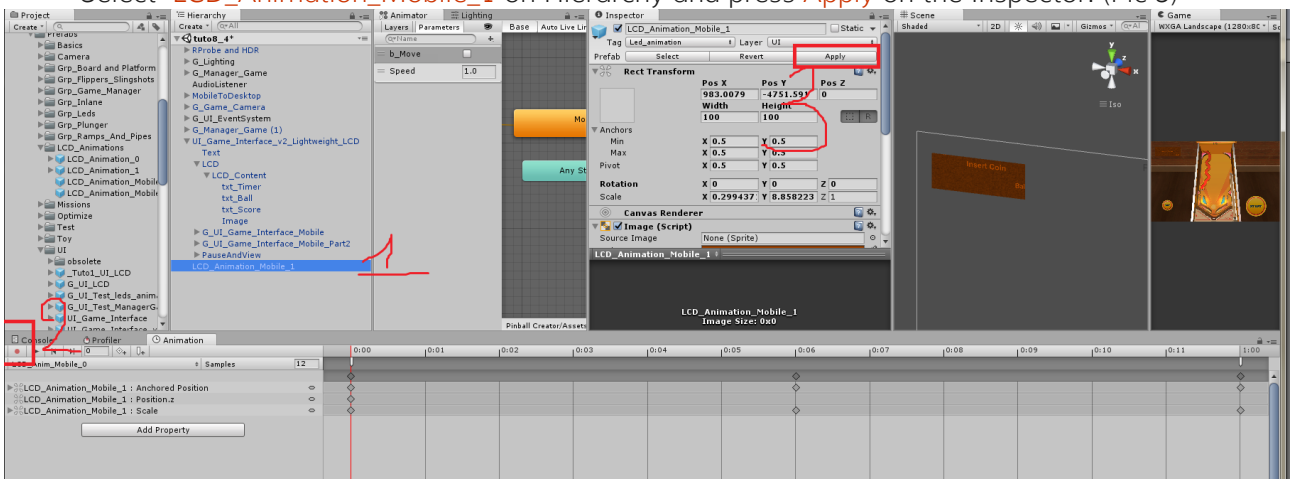


Tips : If you modify objects rotation choose Interpolation -> EulerAngles(Quaternion Approximation).



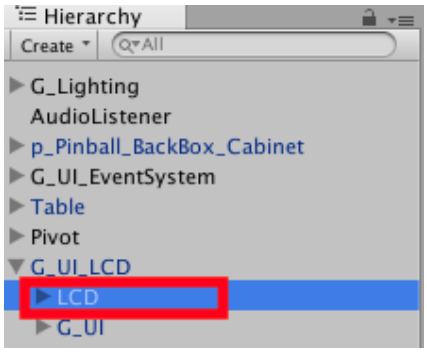
Step 7 : Create your animation.
Tips : To see your animation : Open Window -> Game . Animation appears on LCD Screen.

Step 8 : VERY VERY IMPORTANT . When it's done.
Put t LCD_Animation_Mobile_1 on the root of hierarchy (pic 1)
Uncheck red button on Animation Window (pic 2)
Select LCD_Animation_Mobile_1 on Hierarchy and press Apply on the Inspector. (Pic 3)



How to disconnect LCD Screen, if you want to create your own system.

Delete the game Object named LCD (Hierarchy -> G_UI_LCD -> LCD) on Hierarchy.



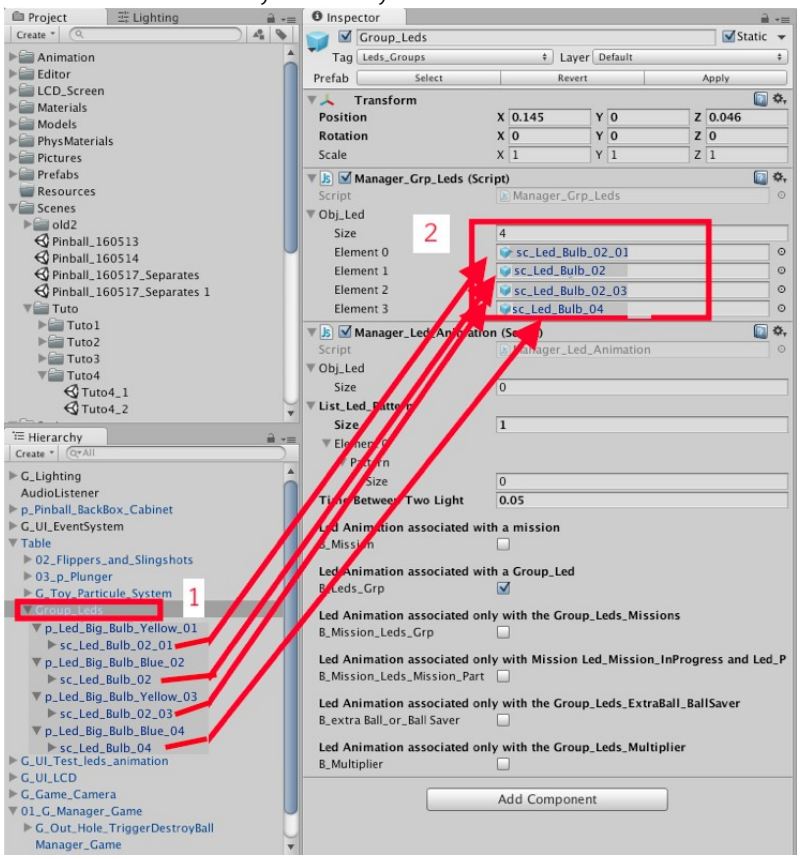
That's it.

LED animation system :

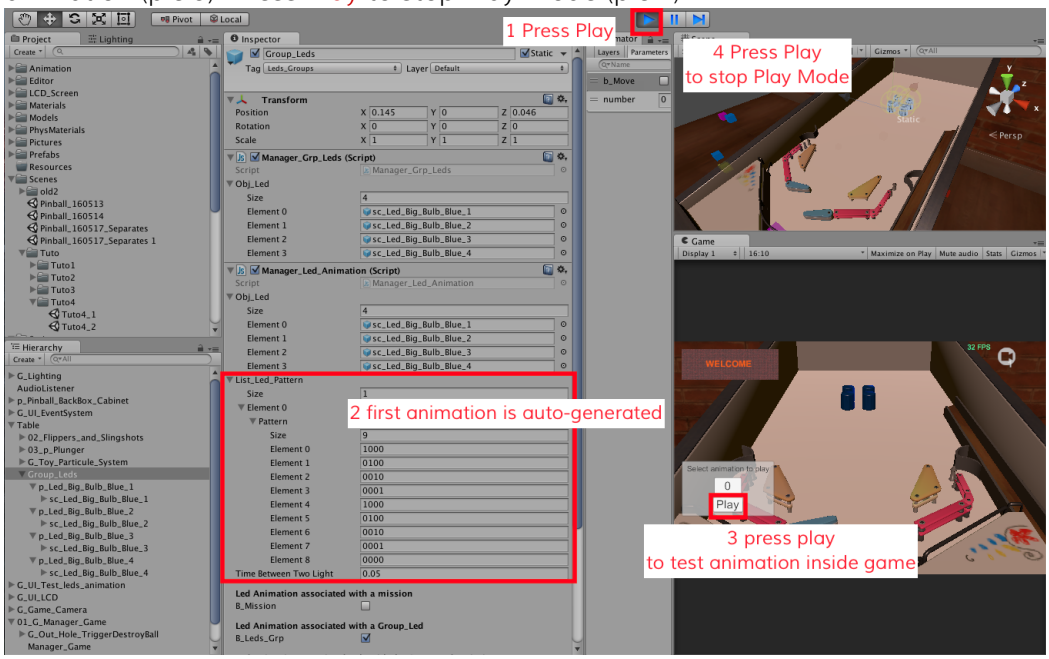
How to create Leds animation.

Step 1 : Open Tuto_4_1 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_1)

Step 2 : On Hierarchy open Table and Select Group_Leds (pic 1).
In this example, 4 leds are connected to variable Obj_Led on script Manager_Grp_Leds.js (pic 2). You can connect as many leds as you want. Read more about how to connect led here

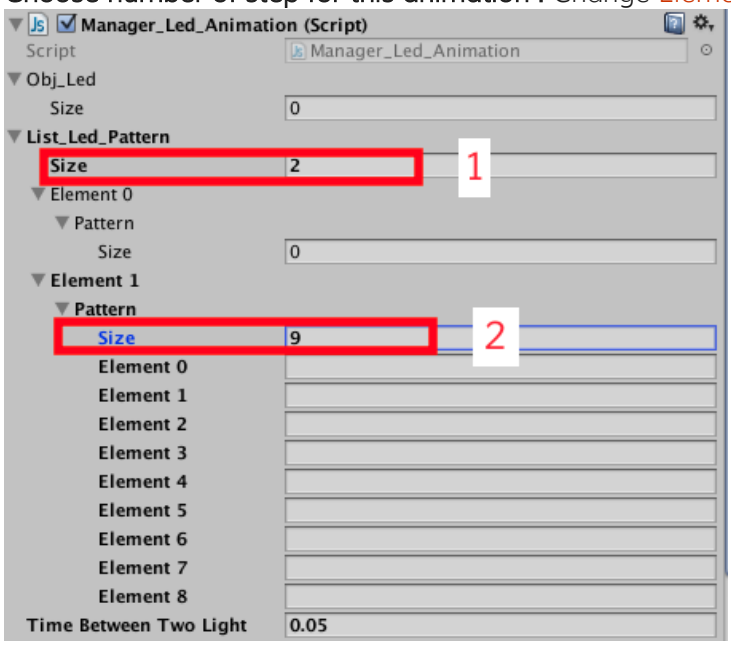


Step 3 : Press Play (pic 1). First animation is auto-generated. (pic 2). Press button Play on Game to see animation (pic 3). Press Play to stop Play Mode (pic 4)



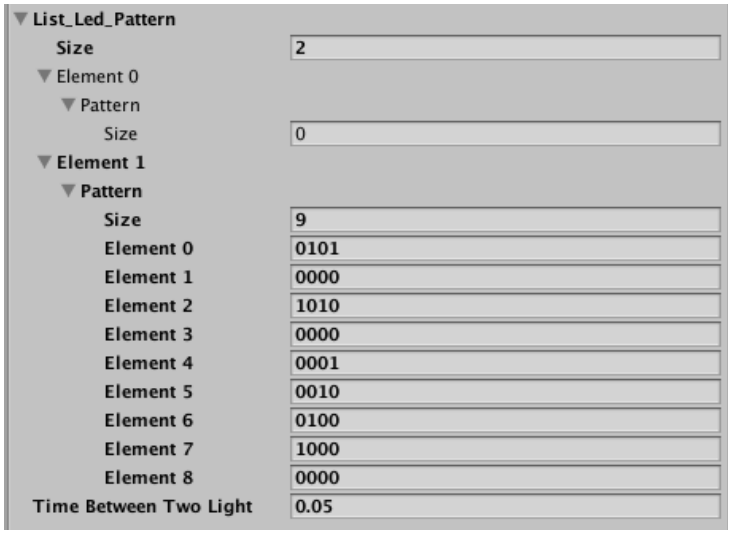
Step 4 :
Create a new animation : Change List_Led_Pattern -> Size to 2 (pic 1).

Choose number of step for this animation : Change Element 1 -> Pattern -> Size to 9 (pic 2)

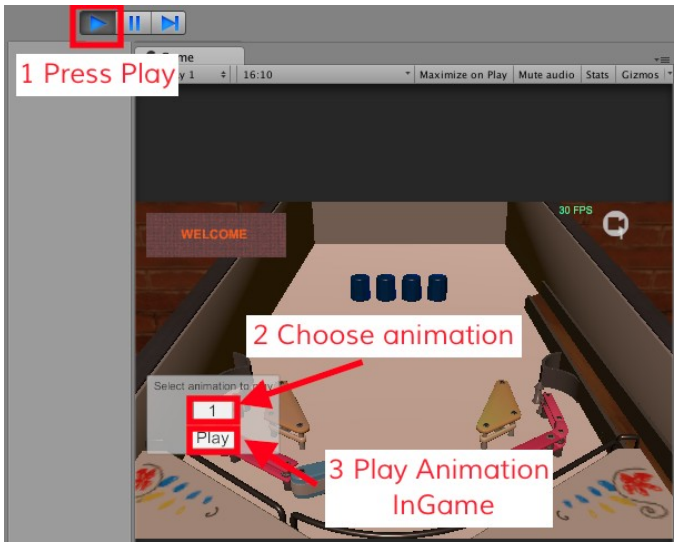


Create pattern : You need to describe the state of each led for each step of the animation.
In this example : if you want all the leds OFF. Write 0000.
If you want all the leds ON. Write 1111
If you want led 0 et 3 On. Write 1001.
Last example : if you have 6 leds and you want led 1 and led 4 On. Write 010010

In this example Write : (pic 1)
Element 1 -> Pattern -> Element 0 : **1010**
Element 1 -> Pattern -> Element 1 : **0000**
Element 1 -> Pattern -> Element 2 : **0101**
Element 1 -> Pattern -> Element 3 : **0000**
Element 1 -> Pattern -> Element 4 : **0001**
Element 1 -> Pattern -> Element 5 : **0010**
Element 1 -> Pattern -> Element 6 : **0100**
Element 1 -> Pattern -> Element 7 : **1000**
Element 1 -> Pattern -> Element 8 : **0000 :** Important Switch OFF all leds at the end of animation.



Test : Press Play (pic1). Press button (pic 2) to choose animation. Press Play (pic 3) to play animation.

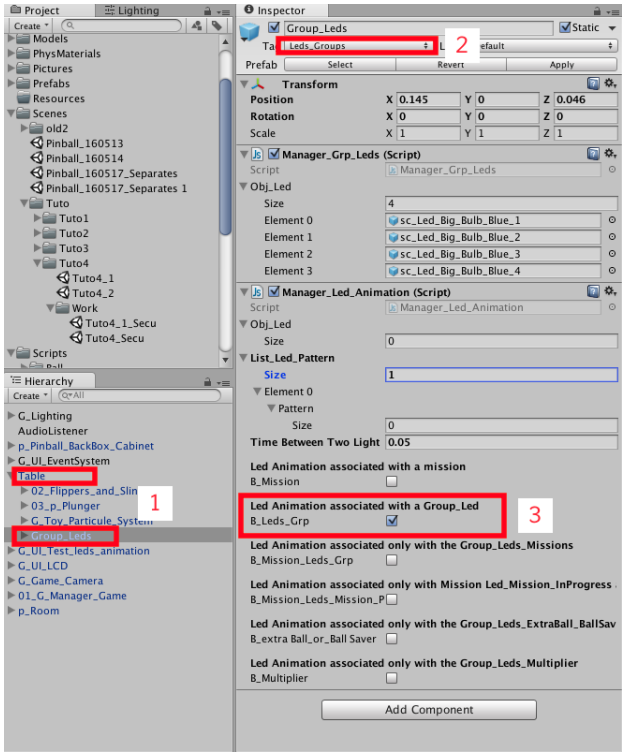


What you need to know depending of Led type (Examples) :

Case 1 : Group of Leds that is not associated with mission or Manager_Game.

Step 1 : Open Tuto_4_2 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_2)

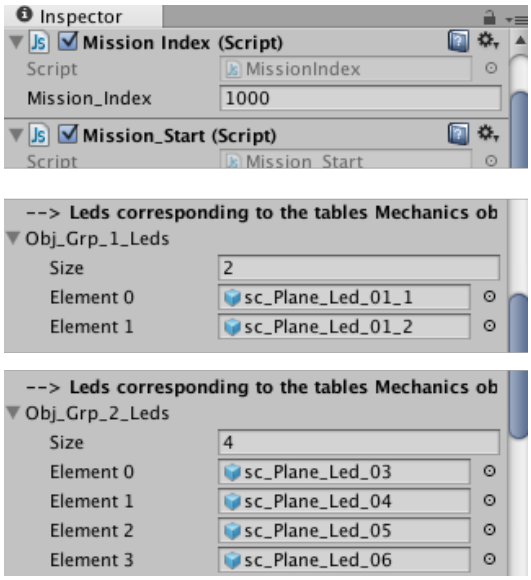
Step 2 : On Hierarchy open Table and Select Group_Leds (pic 1).
Tag = Leds_Groups (pic 2).
B_Leds_Grp = true (pic 3)
To create this type of Group you could start with prefab Group_Leds (Project -> Assets -> Prefabs -> Grp_Leds -> Group_Leds)



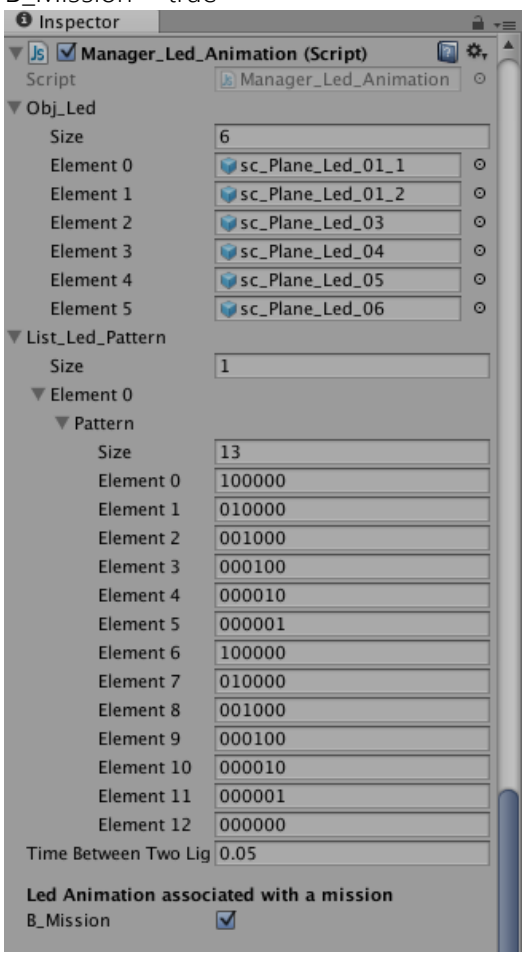
Case 2 : Group of Leds that is associated with mission (Obj_Led_Grp1 and Obj_Led_Grp2).

Step 1 : Open Tuto_4_3 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_3)

Step 2 : On Hierarchy open Table and Select _Tuto4_p_Mission_Led_Example.
Led animation is created from Obj_Led_Grp1 and Obj_Led_Grp2 (pic 1).



B_Mission = true



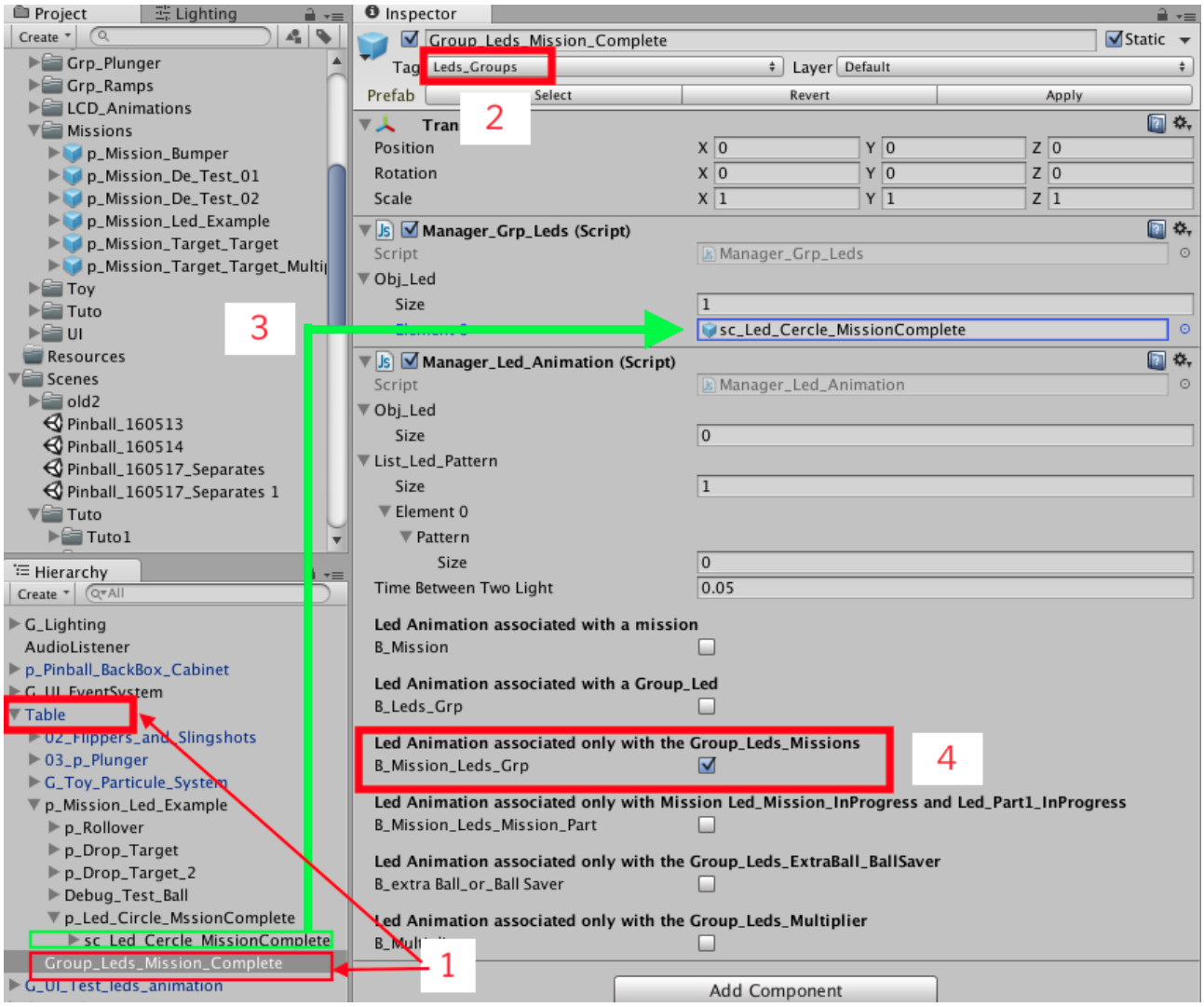
To create this type of Group you could start with prefab _Tuto4_p_Mission_Led_Example (Project -> Assets -> Prefabs -> Missions -> _Tuto4_p_Mission_Led_Example)

Case 3 : Group of Leds that is associated with Led_Mission_Complete.
When a mission is complete a led could be switch ON (more info here)

Step 1 : Open Tuto_4_4 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_4)

Step 2 : On Hierarchy open Table and Select Group_Leds_Mission_Complete. (pic 1)
Tag = Leds_Groups (pic 2).
Put your leds inside Obj_Led (pic 3)
B_Mission_Leds_Grp = true (pic 4)

To create this type of Group you could start with prefab Group_Leds_Mission_Complete (Project -> Assets -> Prefabs -> Grp_Leds -> Group_Leds_Mission_Complete)

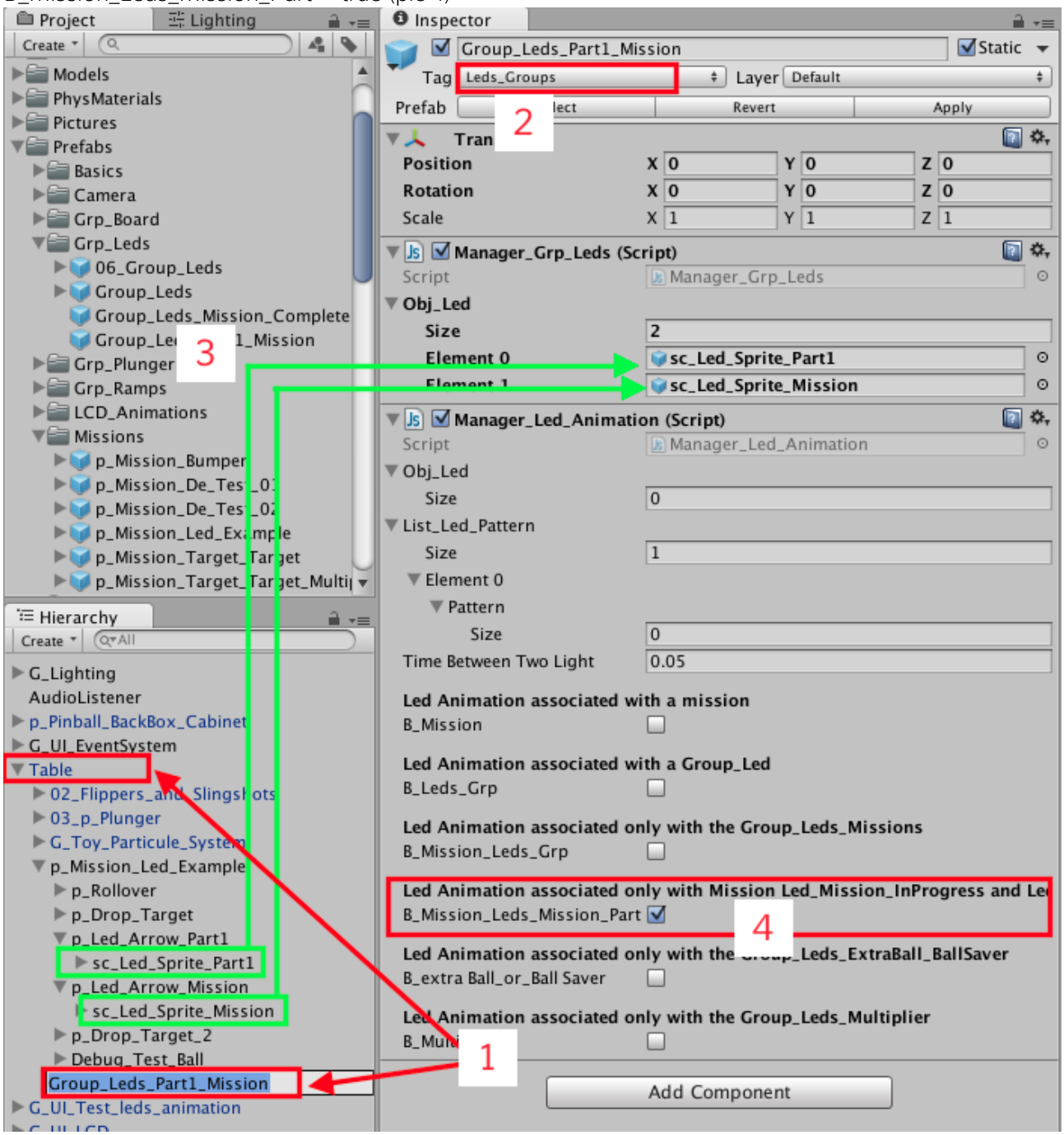


More Example : Open Tuto_4_4_2 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_4_2)
In this example you find 3 missions.
On Hierarchy open Table and Select Group_Leds_Mission_Complete
There are 3 Leds on Obj_Led. Each led corresponds to one mission.

Case 4 : Group of Leds that is associated with Led_Part1_InProgress and Led_Mission_InProgress.
When a mission is ''in progress'', leds could be switch ON (more here)

Step 1 : Open Tuto_4_5 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_5)

Step 2 : On Hierarchy open Table and Select Group_Leds_Part1_Mission. (pic 1)
Tag = Leds_Groups (pic 2).
Put your leds inside Obj_Led (pic 3)
B_Mission_Leds_Mission_Part = true (pic 4)

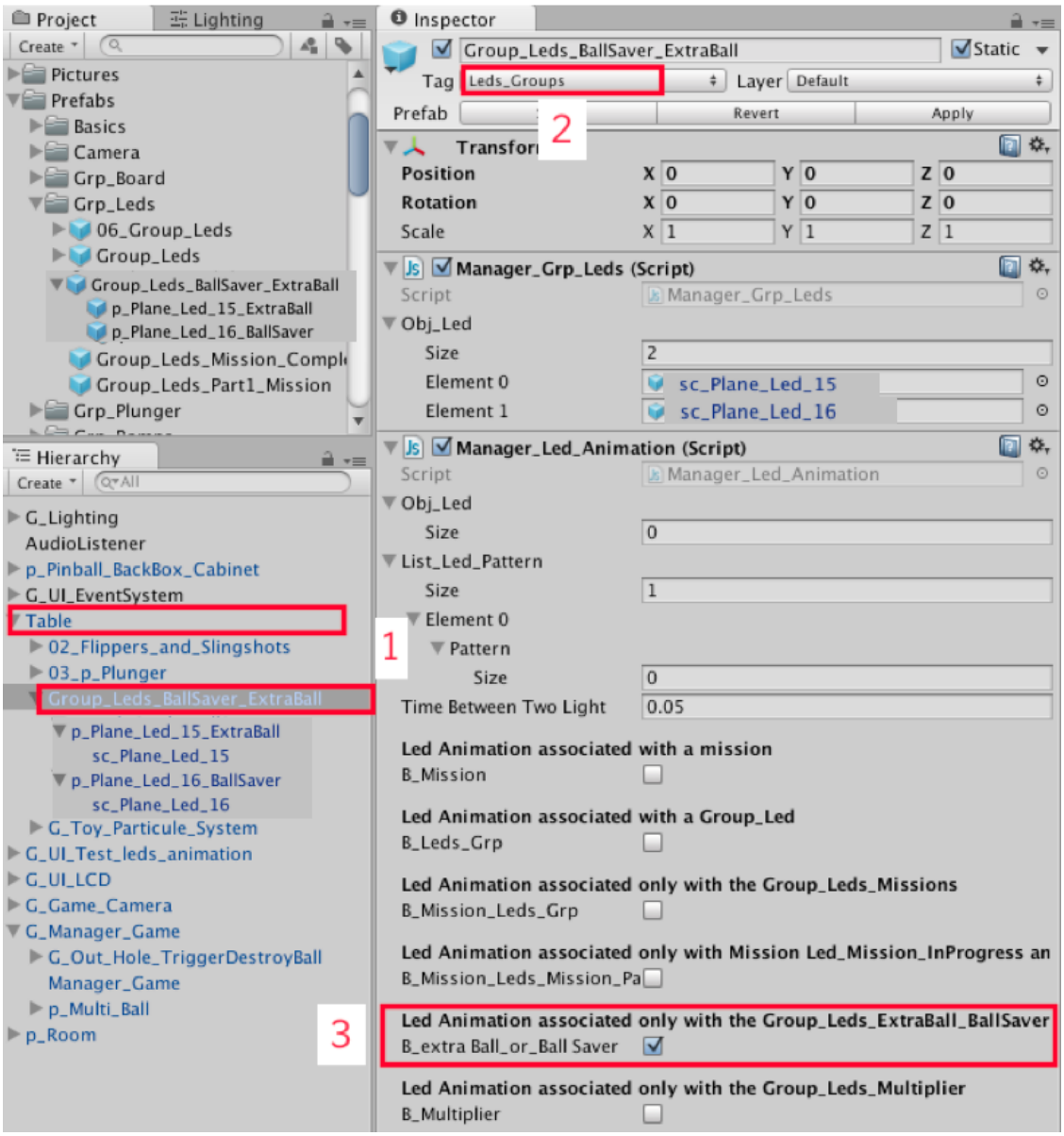


To create this type of Group you could start with prefab Group_Leds_Part1_Mission (Project -> Assets -> Prefabs -> Grp_Leds -> Group_Leds_Part1_Mission)

Case 5 : Group of Leds that is associated with Obj_Led_ExtraBall and Obj_Led_BallSaver.

Step 1 : Open Tuto_4_6 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_6)

Step 2 : On Hierarchy open Table and Select Group_Leds_BallSaver_ExtraBall. (pic 1)
Tag = Leds_Groups (pic 2).
B_Extra_or_BallSaver = true (pic 3)

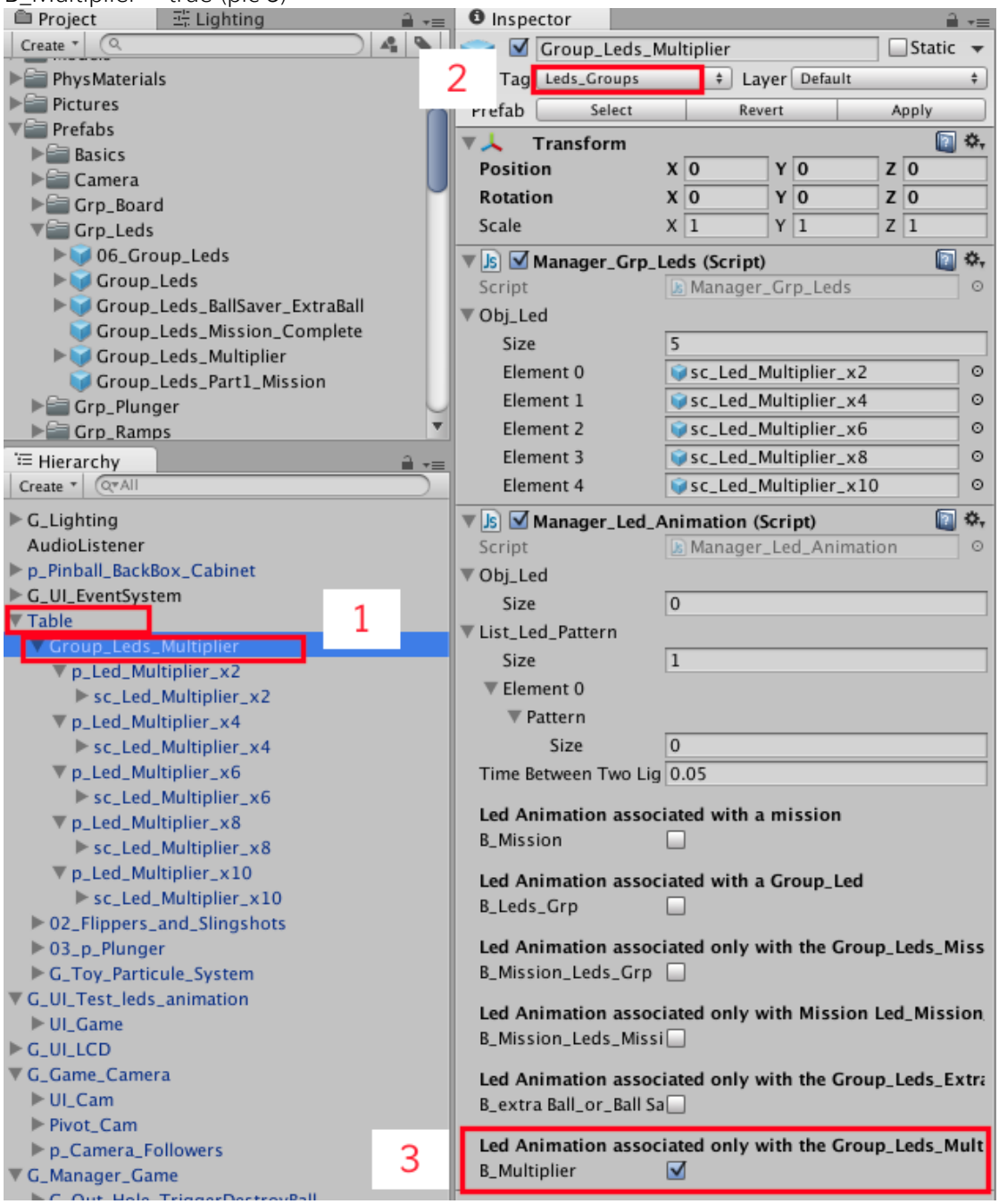


To create this type of Group you could start with prefab Group_Leds_BallSaver_ExtraBall (Project -> Assets -> Prefabs -> Grp_Leds -> Group_Leds_BallSaver_ExtraBall)

Case 6 : Group of Leds that is associated with Obj_Multiplier_Leds on Manager_Game.

Step 1 : Open Tuto_4_7 (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_7)

Step 2 : On Hierarchy open Table and Select Group_Leds_Multiplier. (pic 1)
Tag = Leds_Groups (pic 2).
B_Multiplier = true (pic 3)



To create this type of Group you could start with prefab Group_Leds_Multiplier (Project -> Assets -> Prefabs -> Grp_Leds -> Group_Leds_Multiplier)

How to play led animation during the game.

First way to play led animation during game is in Mission (more info here)
Second way to play led animation during game is in Manage_Game (more info here)

How To create new led combination

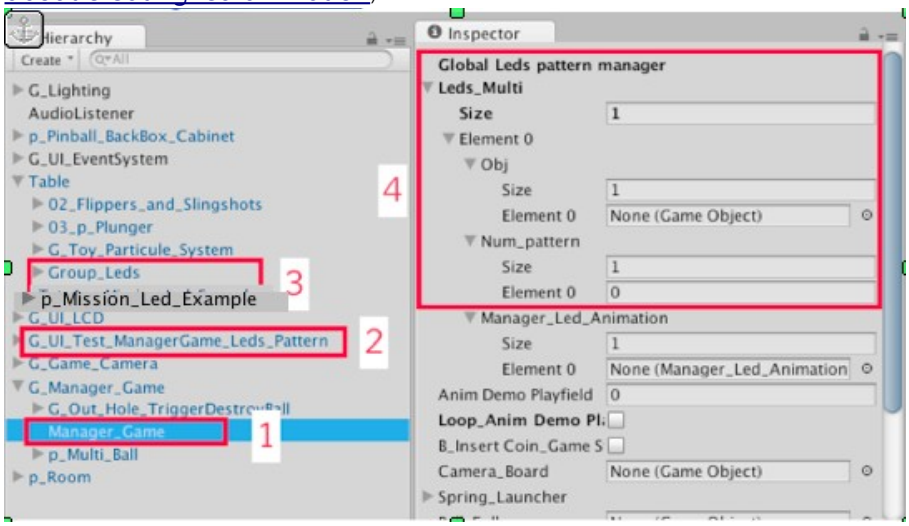
Step 1 : Open Tuto_5_1 (Project -> Assets -> Scenes -> Tuto -> Tuto5 -> Tuto5_1)

Step 2 :
Led animations are played by Manager_Game.
On Hierarchy open G_Manager_Game and Select Manager_Game. (pic 1)

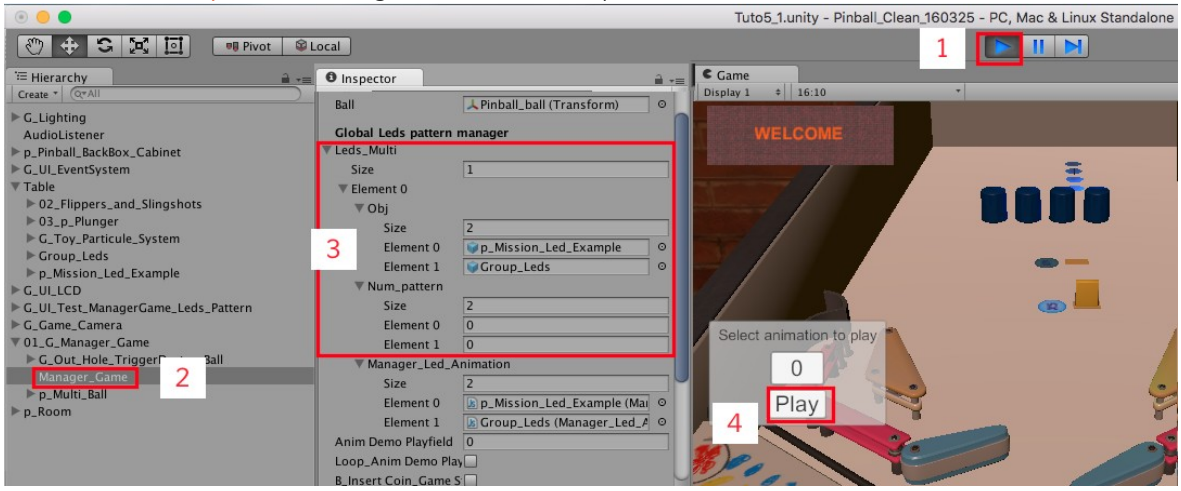
Led animations are manage by Leds_Multi on Manager_Game.js (Inspector). (Pic 4)

G_UI_Test_ManagerGame_Leds_Pattern is used to test animation from Manager_Game gameObject. (pic 2)

In this example we use two groups of leds. Groups_Leds and p_Mission_Led_Example (pic 3) . (More about creating led animation).



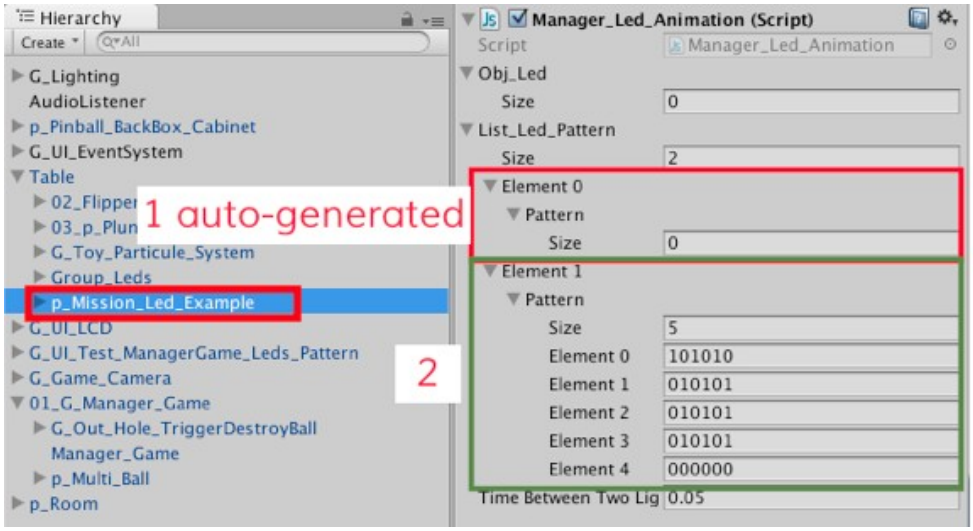
Step 3 : Test scene
Start Play Mode (pic 1)
Select Manager_Game on hierarchy (pic 2).
On Leds_Multi (Inspector) a combination of leds animations is auto-generated (pic 3). It combine all group of leds on Hierarchy. In this example Groups_Leds and _Tuto4_p_Mission_Led_Example.
Press button Play to test this global animation (pic 4).



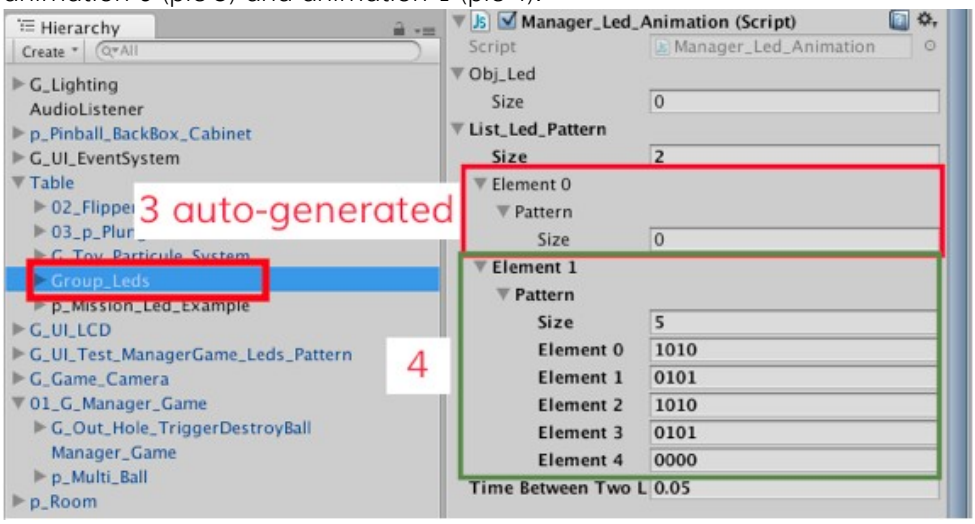
Press Stop before moving to the next part.

Step 3-a : Create a new combination of animation.

p_Mission_Led_Example has 2 animations :
animation 0 (pic 1) and animation 1 (pic 2).
(More about creating Leds animation)



Group_Leds has 2 animations :
animation 0 (pic 3) and animation 1 (pic 4).



Step 3-B : Create a new combination of animation.

a- Select Manager_Game on Hierarchy (pic 1).

b- To create new combination change Leds_Multi -> Size to 3 on the Inspector(pic 2).

Important : We don't want to change first combination (Leds_Multi -> Element 0) because it is auto-generated pic 3).

c- Example 1 : Create a new combination for Leds_Multi -> Element 1 :

- Change Element 1 -> Obj -> Size to 2
- Then drag'n'drop Group_Leds and p_mission_Led_Example inside Element 1 -> Obj -> Element 0 and Element 1 (pic 4).

Change Element 1 -> Num_pattern -> Size to 2 then choose the animation you want to play for each group of leds (pic 5).

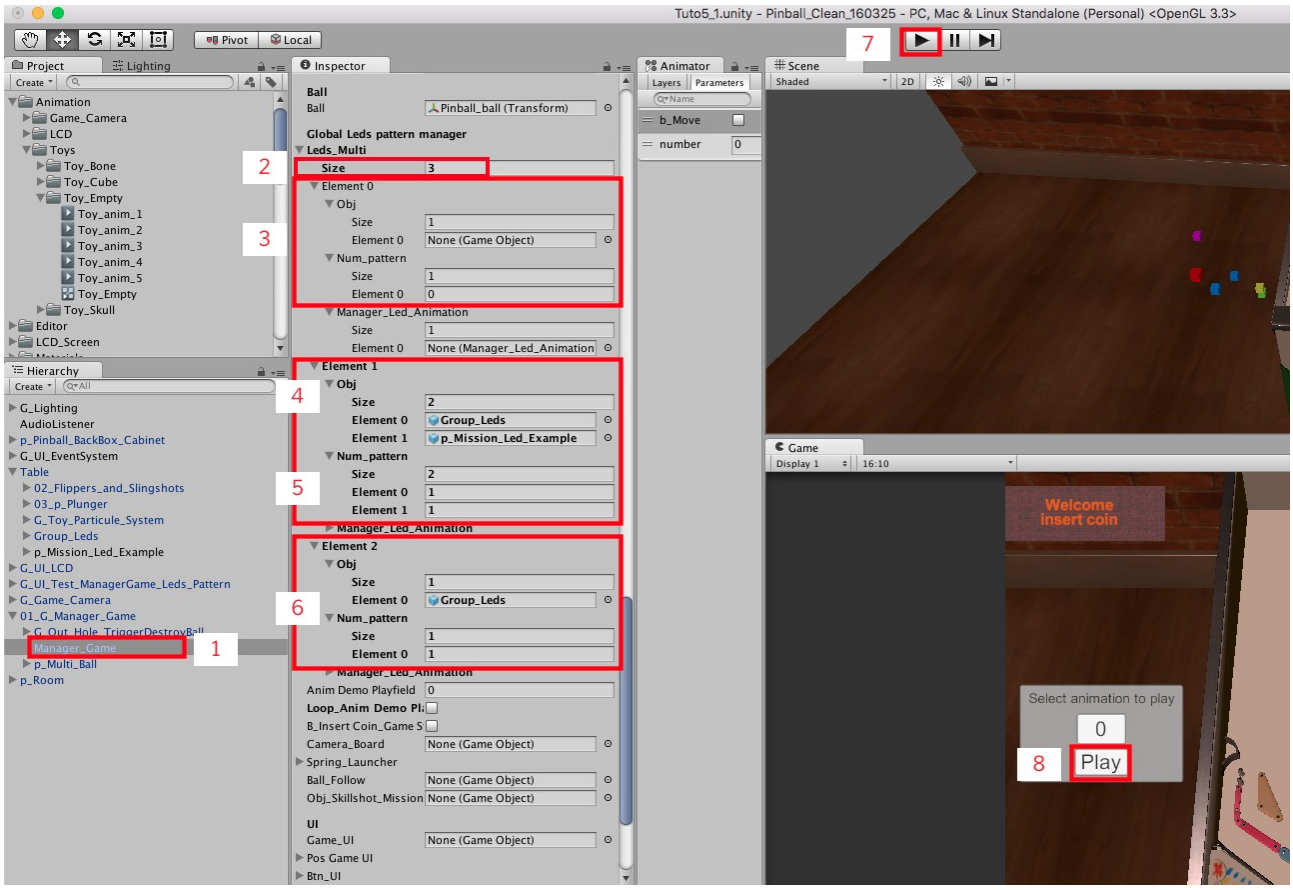
We choose to play animation 1 for Group_Leds and p_mission_Led_Example (pic 5) (Element 0 -> Num_pattern -> Element 0 and Element 1 = 1)

d- Example 2 : Create a new combination for Leds_Multi -> Element 2 :
- Drag'n'drop Group_Leds inside Element 2 -> Obj -> Element 0 (pic 6).
- We choose to play animation 1 (Element 1 -> Num_pattern -> Element 0 = 1)

e- Start Play Mode (pic 7).

f- Test animations by clicking button Play on Game (pic 8)

If you have a problem, Open Tuto_5_2 (Project -> Assets -> Scenes -> Tuto -> Tuto5 -> Tuto5_2).



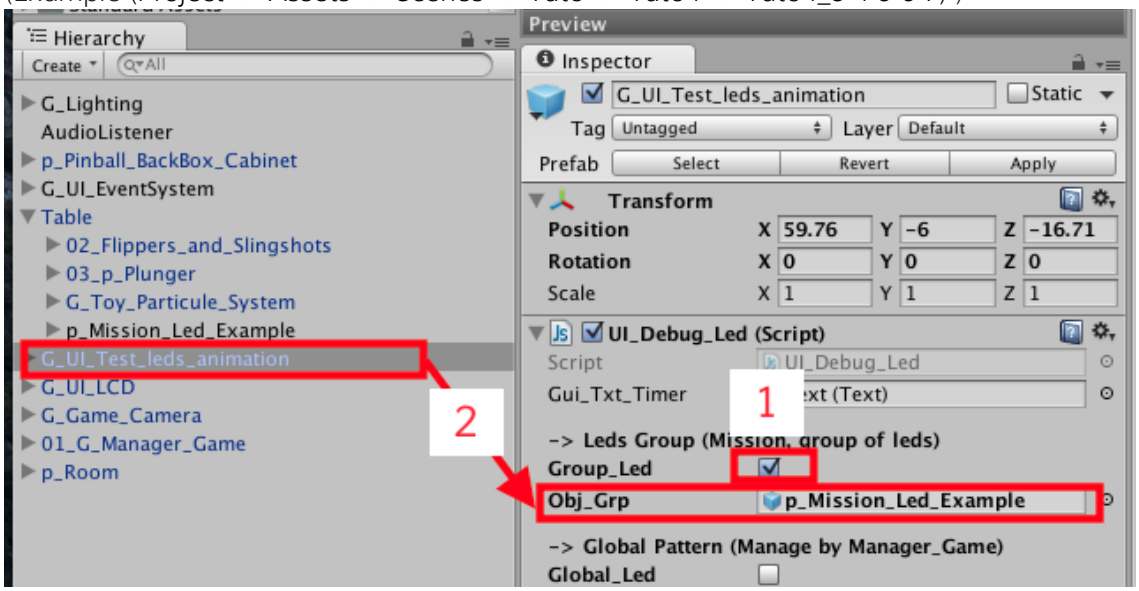
How to use G_UI_Test_Led_animation and G_UI_Test_ManagerGame_Leds_Pattern prefab.

Cas 1 : G_UI_Test_Led_animation :

This prefab is used to test Leds animation from Missions or a group of Led.

Step 1 : Drag'n'drop G_UI_Test_Led_animation on Hierarchy (Project -> Assets -> Prefabs -> UI -> G_UI_Test_Led_animation).
Step 2 : Group_Led = true (pic 1). Drag'n'drop a mission or a group of leds you wanted to test inside Obj_Grp (pic 2).

(Example (Project -> Assets -> Scenes -> Tuto -> Tuto4 -> Tuto4_3 4 5 6 7))

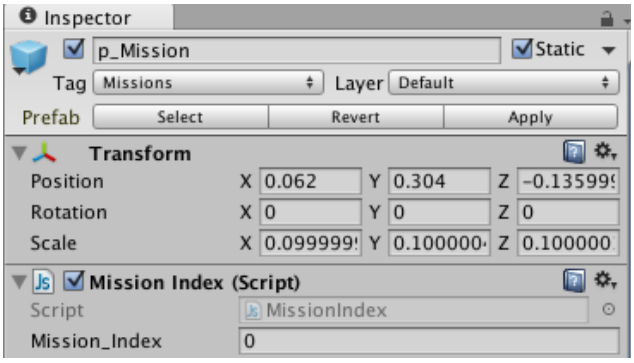


Cas 2 : G_UI_Test_ManagerGame_Leds :

This prefab is used to test Leds combination from Manager_Game on Hierarchy.

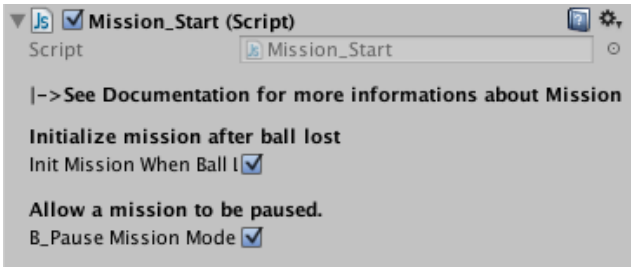
Step 1 : Drag'n'drop G_UI_Test_ManagerGame_Leds_Pattern on Hierarchy (Project -> Assets -> Prefabs -> UI -> G_UI_Test_ManagerGame_Leds_Pattern).
(Example (Project -> Assets -> Scenes -> Tuto -> Tuto5 -> Tuto5_2))

Mission :

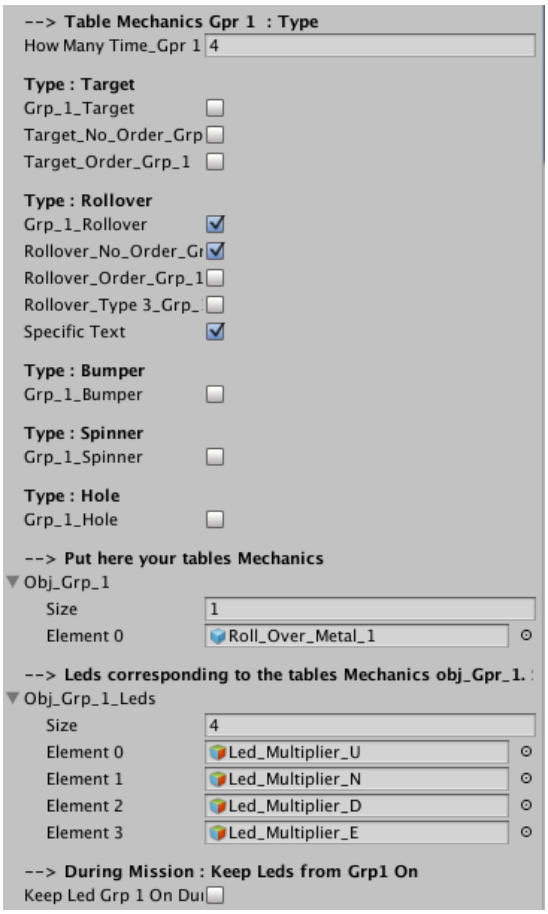


Tag : Missions

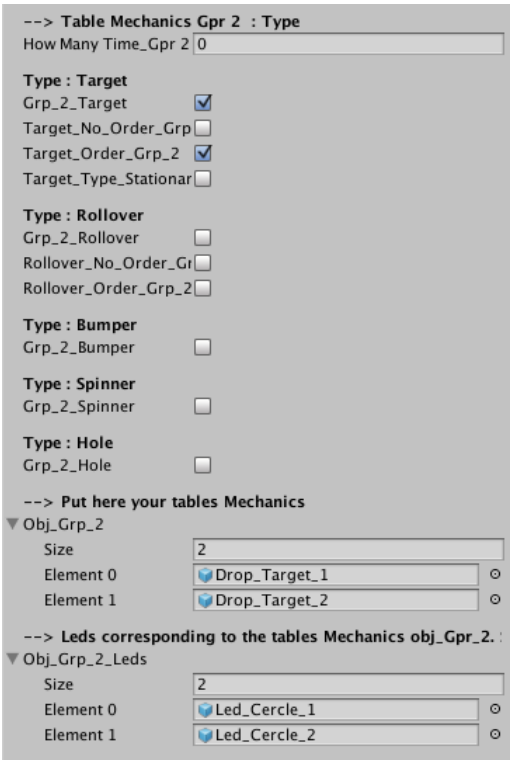
Mission_index : VERY IMPORTANT Choose a unique Index for each mission.



InitMissionWhenBallLost : if True the mission is init when the player lose a ball. False the mission is init only if it's the part 2 of the mission or when the player is game over.
b_PauseMissionMode : If false. Mission is not affected by the pause of other mission. And the mission couldn't pause other mission

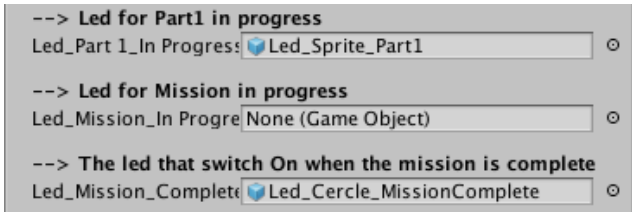


Mission Part 1
see section mission configuration to learn more about mission part 1 configuration.



Mission Part 2

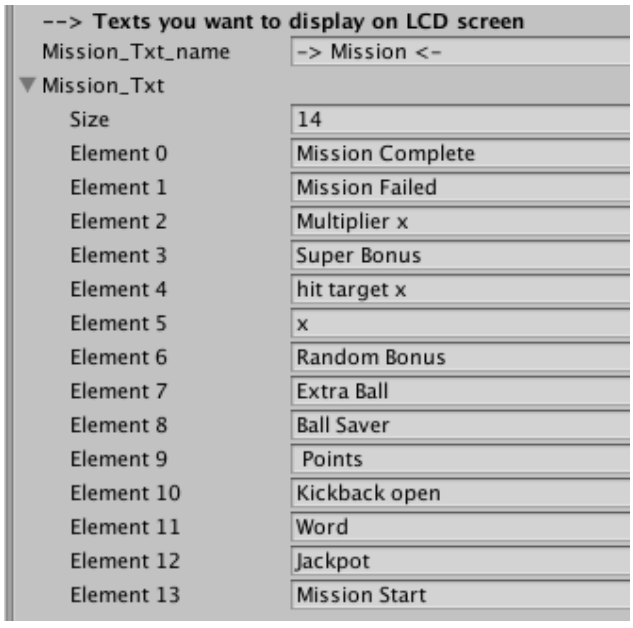
see section mission configuration to learn more about mission part 2 configuration.



Led_Part1_InProgress : Switch On a led when mission Part1 is in progress

Led_Mission_InProgress : Switch On a led when mission Part 2 is in progress

Led_Mission_Complete : Switch On a led when mission is complete. This led stay switch On until the player is game over



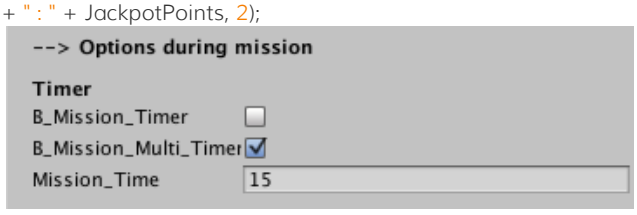
Mission_Txt_name : Mission name

Mission_Txt : An array to manage the text you want to write on LCD screen

All text combination :

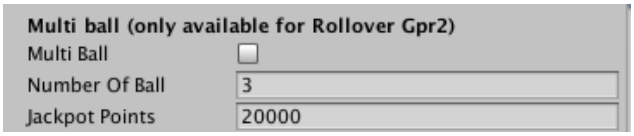
```
gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n"// text : Mission name
+ Mission_Txt[1]);// text : Mission Failed

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
+ Mission_Txt[0] + "\ n"// text : Mission Complete
+  Mission_Txt[7]+"</size>", 3);// text : Extra Ball
```

```
gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"
+ Mission_Txt[0] + "\ n"// text : Mission Complete
+ "\ n" + Mission_Txt[2]// text : Multiplier x
+ gameManager.F_return_multiplier().ToString()// text : x+"</size>", 3);

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
+ Mission_Txt[0] + "\ n"// text : Mission Complete
+ Mission_Txt[3]// text : Super Bonus
+ gameManager.F_return_Mulitplier_SuperBonus().ToString()          // text : Super Bonus value
+"</size>", 3);

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
+ Mission_Txt[0] + "\ n" // text : Mission Complete
+  Mission_Txt[10]+"</size>", 3);// text : Kickback open

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
Mission_Txt[0] + "\ n" + Points// text : Mission Complete
+  Mission_Txt[9]+"</size>", 3);// text : Points

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
+ Mission_Txt[0] + "\ n" // text : Mission Complete
+  Mission_Txt[6]+ "\ n" // text : Random Bonus
+  Mission_Txt[7]+"</size>", 3);// text : Extra Ball

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
+ Mission_Txt[0] + "\ n"// text : Mission Complete
+  Mission_Txt[6]+ "\ n"// text : Random Bonus
+  Mission_Txt[8]+"</size>", 3);// text : Ball Saver

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
+ Mission_Txt[0] + "\ n"// text : Mission Complete
+  Mission_Txt[6]// text : Random Bonus +
 "\ n" + Mission_Txt[2] // text : Multiplier x
+ gameManager.F_return_multiplier().ToString()// text : x
+"</size>", 3);

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n" + "<size= 20>"// text : Mission name
+ Mission_Txt[0] + "\ n" // text : Mission Complete
+  Mission_Txt[6]+ "\ n" + Points// text : Random Bonus
+  Points + Mission_Txt[9]+"</size>", 3);// text : x Points

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n"// text : Mission name
+ (HowManyTime_Gpr1 - Step).ToString()
+ Mission_Txt[4], 3);    // text : More

gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n"// text : Mission name
+ Mission_Txt[5],3);// text : Mission Start

"<color=#FF640078>"
Mission_Txt[11][i] + "</color>";          // text : Specific text

gameManager.Add_Info_To_Array(
Mission_Txt_name // text : Mission name+ "\ n" + Mission_Txt[12]// text : Jackpot
+ " : " + JackpotPoints, 2);
```



When mission Part 2 start you could choose to add a mission Timer. If timer = 0 mission is failed.

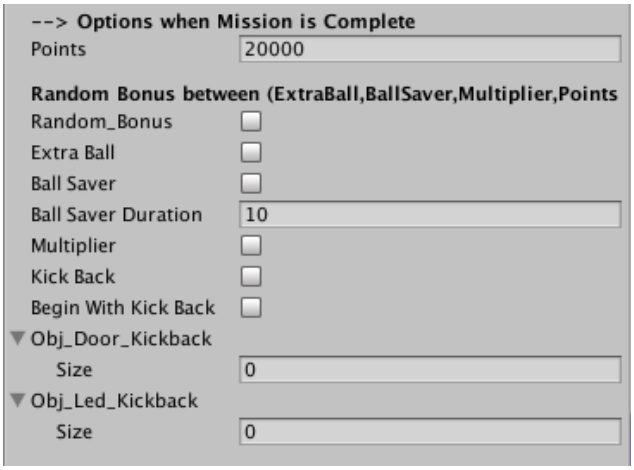
b_Mission_Timer : If true : Timer is not initialized during mission part 2.



b_Mission_Multi_Timer : If true : Timer is initialized when ball hit an object
Mission_Timer : Timer duration
Multi Ball is only available with Rollover on part 2

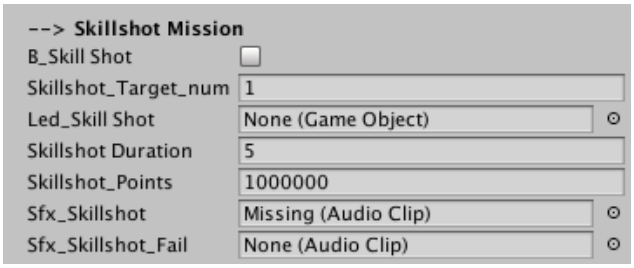
Multi-ball starts when Mission part 1 is ended and stop when there is only one ball on playfield.

MultiBall : if true, multi ball start when mission part 1 is ended.
MultiBall ended when all the multiball are ejected and there is only one ball on playfield
NumberOfBall : The number of multi ball.
JackpotPoints : Points win when ball go through a rollover



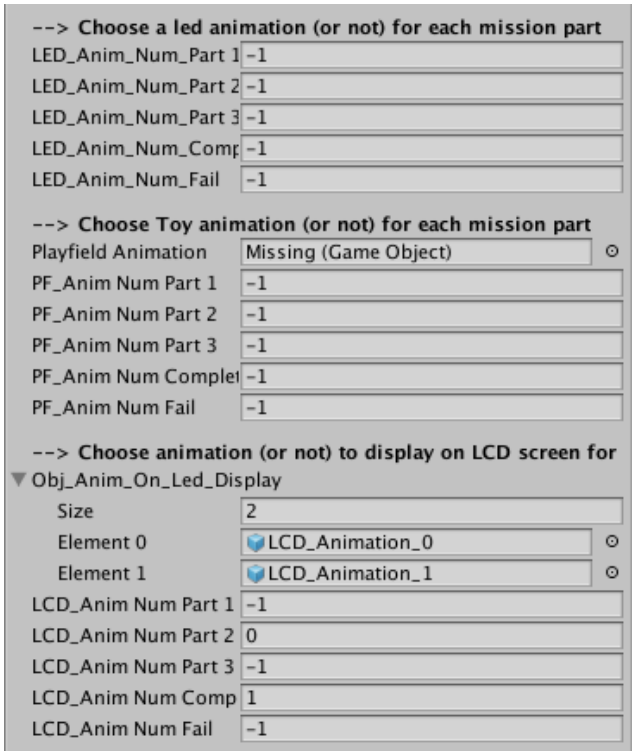
Bonus options when a mission is complete

Choose only one option at a time
Random_Bonus : choose a bonus randomly between Extra Ball, Ball Saver, Multiplier, points
ExtraBall : win an extra ball
BallSaver : Ball saver start
BallSaverDuration : Choose the duration of the ball Saver
Multiplier : increase the Bonus multiplier
KickBack : Open kickback. You need to connect a target. Led is optional.
Obj_Door_Kickback : Connect a target here
Obj_Led_Kickback : Connect a Led here. Optional (more about Kickback)



Skillshot : You could use an object from obj_Grp1 to activate the skillshot. Skillshot start when the player eject ball from the plunger.

B_Skillshot : if true this mission is used for skillshot. Only mission could is variable= true;
Skillshot_Target_num : it is the index number of the object you want to choose for skillshot.
SkillshotDuration : Skillshot duration.
Skillshot_Points : Points win
Sfx_Skillshot : Sound if skillshot is complete
Sfx_Skillshot_Fail : Sound if skillshot is not complete



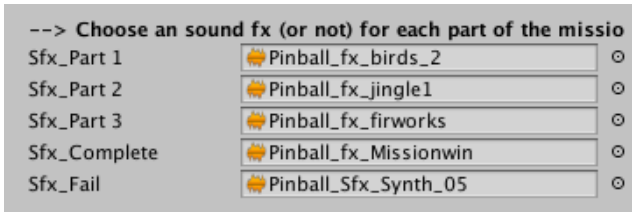
->You could play Led animations, Toy animations and LCD animations for each part of a mission.
Part 1 : Hits objects from obj_Grp_1
Part 2 : Part 1 is finish, Part 3 start.
Part 3 : Hits objects from obj_Grp_2
Part Complete : Mission is complete
Part Fail : Mission is fail.

Important : -1 mean that nothing is played

->Choose Led animation (or not) for each mission part :
Ithe number refers to leds combination you could find on Manager_Game. 0 refers to the default led combination. -1 mean no combination is played for this part.
More Info here

->Choose Toy animation (or not) for each mission part :
Playfield Animation : Connect a Toy.
PF_AnimNumPart 1,2,3,Complete,Fail :
If Toy use animation, choose animation you want to play (0 to 4). -1 mean no Toy animation for this part.
More info here

->Choose animation (or not) you want to display on LCD screen for each mission part : More info here

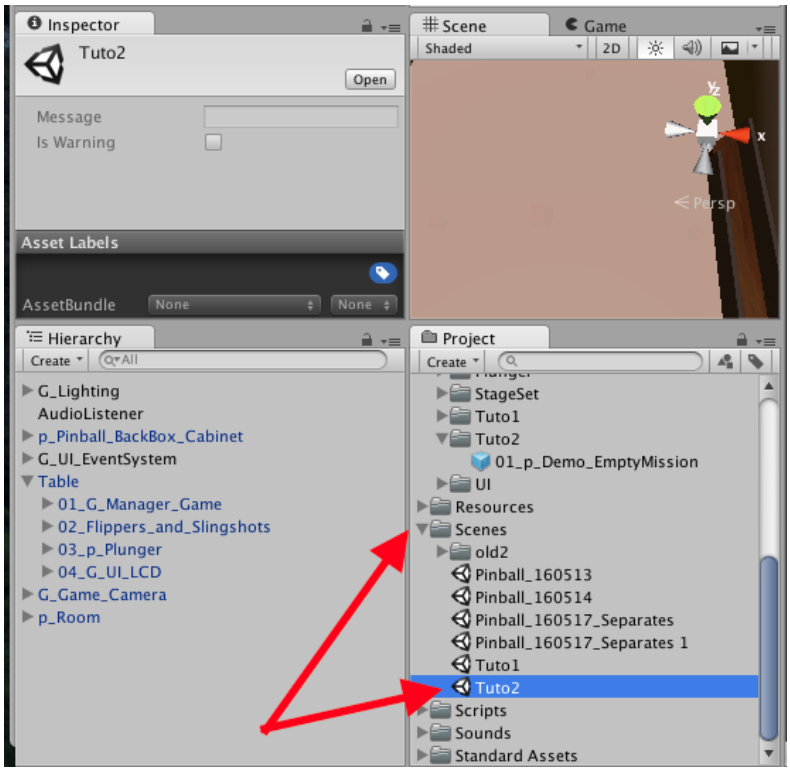


These sounds are played when :
sfx_Part1 : an object in Obj_Grp1 is touched
sfx_Part2 : Part1 is over
sfx_Part3 : an object in Obj_Grp2 is touched
sfx_Complete : Mission is complete
sfx_Fail : Mission fails

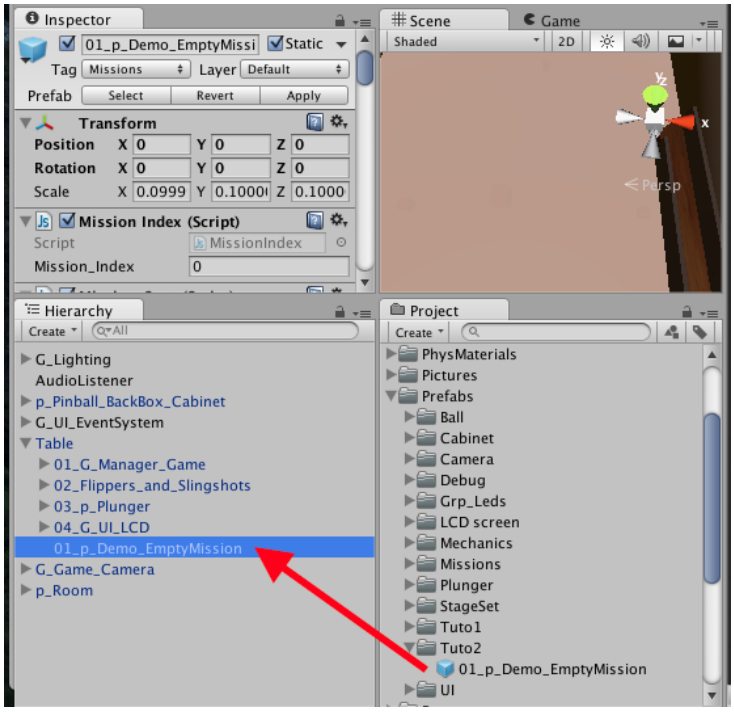
Connect a UI Image.

Tuto 3 : Create a mission from scratch

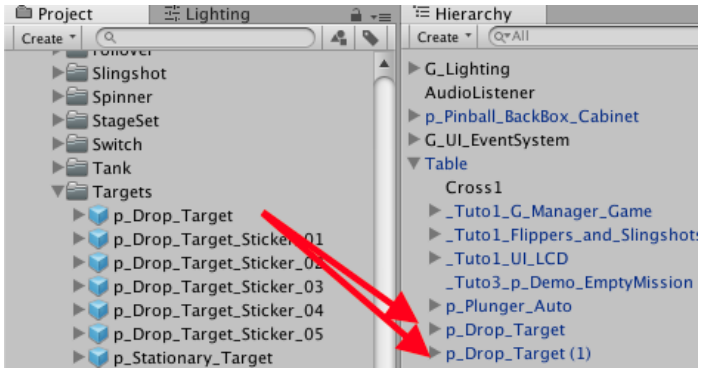
1 Open scene Tuto3 :
(Project -> Assets -> Scenes -> Tuto -> Tuto3 -> Tuto3)



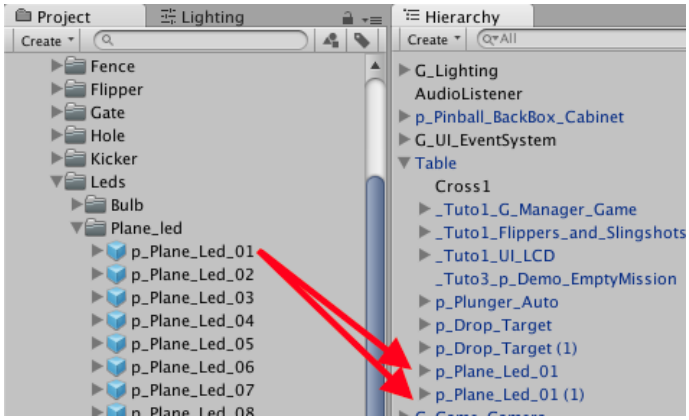
2 Create an empty mission :
Drag'n'drop prefab _Tuto3_p_Demo_EmptyMission inside gameObject Table on the Hierarchy.
(Project -> Assets -> Prefabs ->Missions -> _Tuto3_p_Demo_EmptyMission)



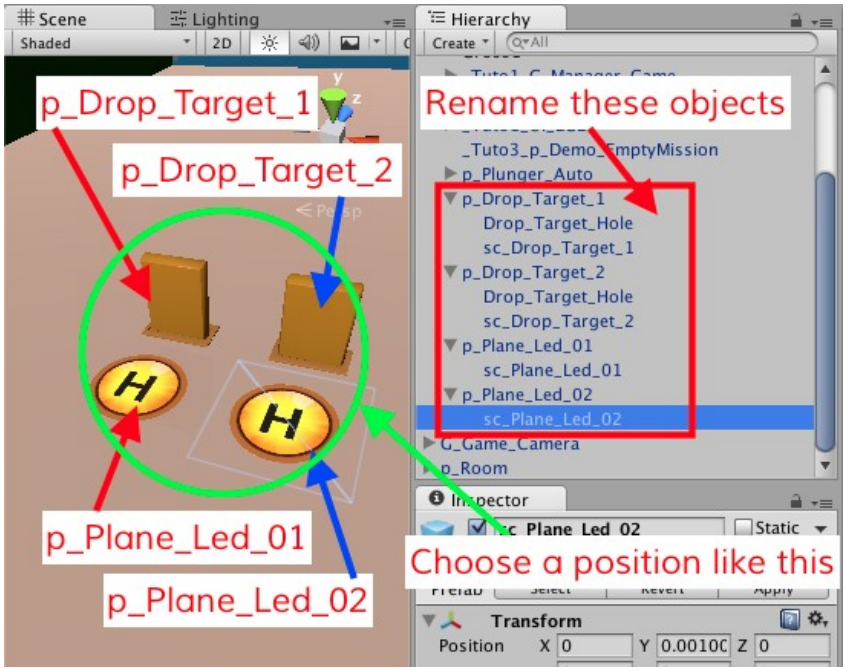
3 Create targets and there associated leds :
Drag'n'drop 2 prefabs p_Drop_Target inside object Table on the Hierarchy. Local position(0,0,0)
(Project -> Assets -> Prefabs -> Basics -> Targets -> p_Drop_Target)



Drag'n'drop 2 prefabs p_Plane_Led_01 inside object Table on the Hierarchy. Local position(0,0,0)
(Project -> Assets -> Prefabs -> Basics -> Leds -> Plane_Led -> p_Plane_Led_01)



Target 1 and 2 : Rename p_Drop_Target and his child sc_Drop_Target.
Led 1 and 2 : Rename p_Plane_Led_01 and his child sc_Plane_Led_01 .
Move prefabs on playfield. See next picture.



If you have a problem open Tuto3_2 (Project -> Assets -> Scenes -> Tuto -> Tuto3 -> Tuto3_2)

3 Set up table mechanics :

3a : (Pic 1) Select sc_Drop_Target_1 on Hierarchy

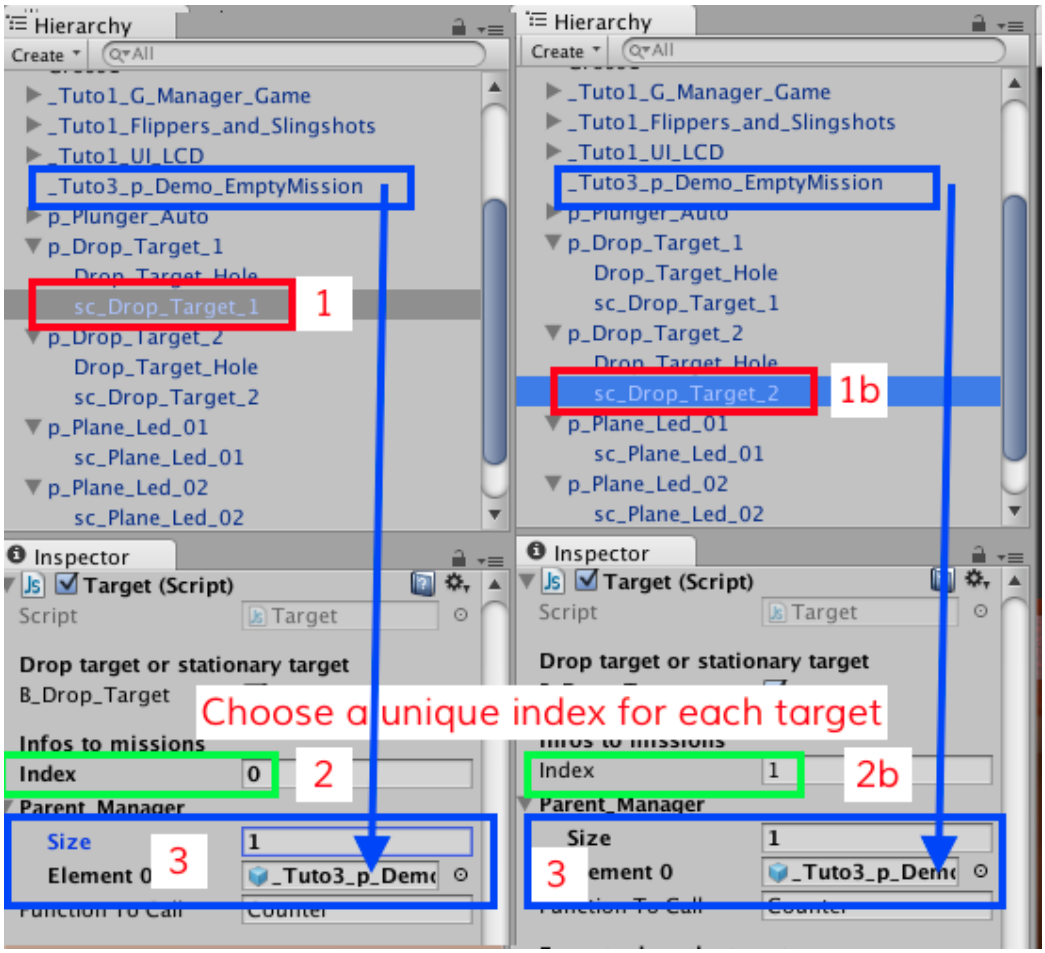
3b : (Pic 2) Very Important Choose an unique index for each target
first target : index = 0, second target : index = 1

The mission recognize an object with his index. So each mechanical prefab must have a unique index

3c : (Pic 1b and 2b) Do the same with sc_Drop_Target_2

3d : (Pic 3) Very Important **Assign mission prefab to targets** :
Select sc_Drop_Target_1 and sc_Drop_Target_2, change Parent_Manager Size to 1. Then drag'n'drop prefab 01_p_Demo_EmptyMission inside Parent Manager Element 0.

If you have a problem open Tuto3_3 (Project -> Assets -> Scenes -> Tuto -> Tuto3 -> Tuto3_3)



4 Create mission Part 1 : Drop 2 targets
Select 01_p_Demo_EmptyMission on Hierarchy (This prefab manage the mission).
On script Mission_Start :
Pic 1 : HowManyTime_Grp1 = 2. This value represent the number of ball hits to complete mission part 1.

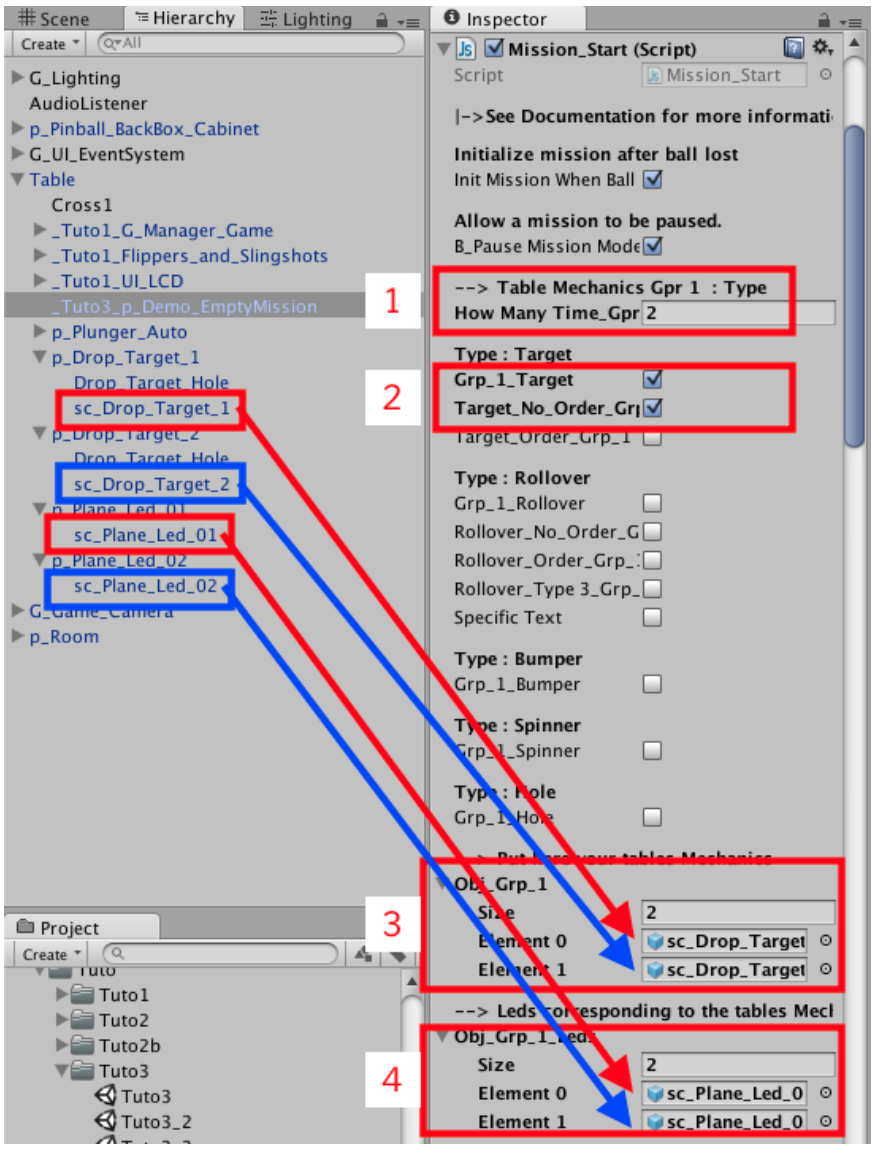
Pic 2 : Check Grp_1_Target (Check this box because mechanics are targets for mission part 1)
Check Target_No_Order_Grp1 (Check this box because player could hit the target in any order)

Pic 3 : Obj_Grp_1 -> size = 2
Drag'n'drop sc_Drop_Target_1 and sc_Drop_Target_2 inside Obj_Grp_1 -> Element 0 and 1.

Pic 4 : Obj_Grp_1_Leds -> size = 2
Drag'n'drop sc_Plane_Led_01 and sc_Plane_Led_02 inside Obj_Grp_1_Leds -> Element 0 and 1.

IMPORTANT Put led on the same order as there associated target.

Press Play. Now mission are ready to play.
After hitting the two targets mission is complete.
If you have a problem open Tuto3_4 (Project -> Assets -> Scenes -> Tuto -> Tuto3 -> Tuto3_4)

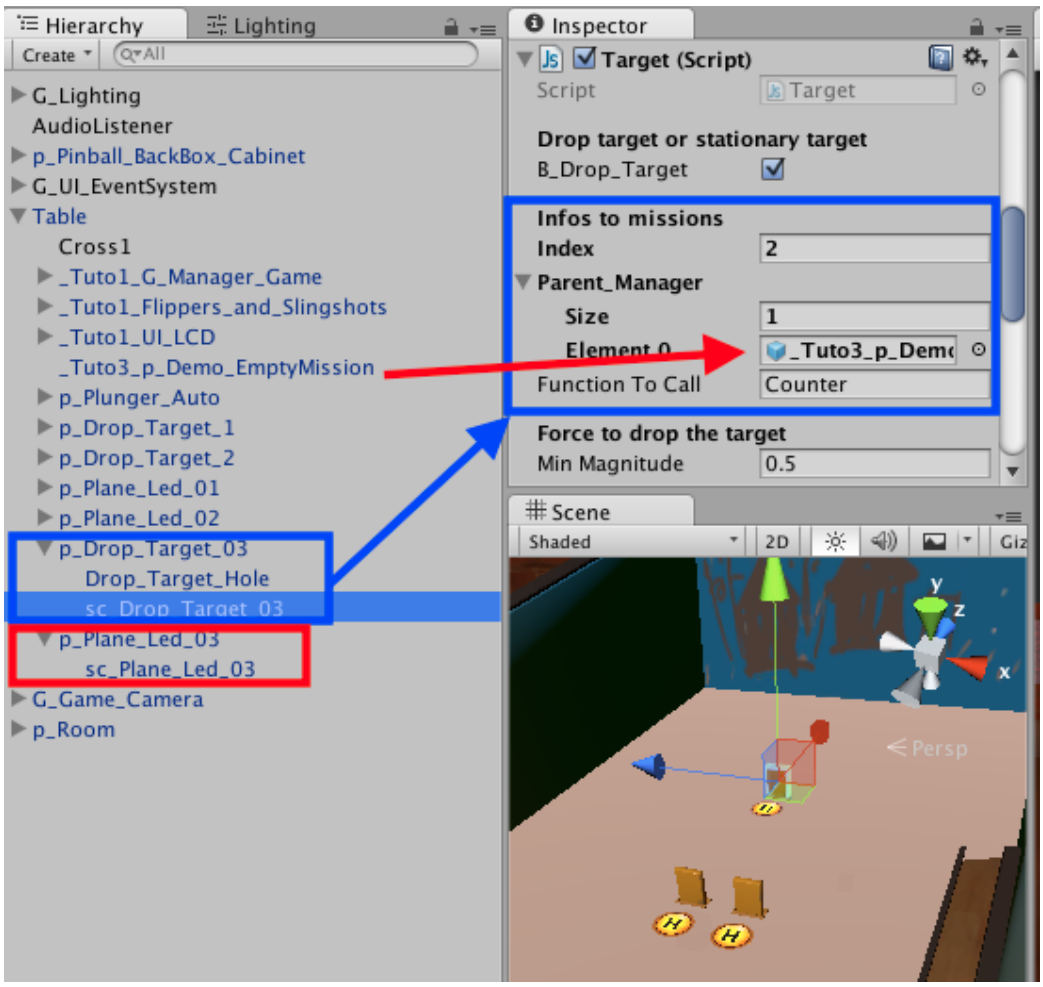


5 Create target and led for mission part 2:
Pic 1 : Drag'n'drop prefab p_Drop_Target inside object Table on Hierarchy.
(Project -> Assets -> Prefabs -> Mechanics -> Targets -> p_Drop_Target)
Pic 2 : Rename p_Drop_Target and his child sc_Drop_Target.
Pic 3 : Select sc_Drop_Target and Choose a unique index.
Pic 4 : Assign mission to targets : change Parent_Manager Size to 1. Then drag'n'drop prefab ''01_p_Demo_EmptyMission'' inside Parent Manager Element 0.

Pic 5 : Drag'n'drop prefab p_Plane_Led_01 inside object Table on Hierarchy.
(Project -> Assets -> Prefabs -> Mechanics -> Leds -> p_Plane_Led_01)
Pic 6 : Rename p_Plane_Led_01 and his child sc_Plane_Led_01 .

Pic 7 : Move target and led where you want on playfield

You should have this.



If you have a problem open Tuto3_5 (Project -> Assets -> Scenes -> Tuto -> Tuto3 -> Tuto3_5)

6 Create mission Part 2 : Drop 1 target
Select 01_p_Demo_EmptyMission on Hierarchy (This prefab manage the mission).
On script Mission_Start :
Pic 1 : HowManyTime_Grp2 = 1 (This value represent the number of hit to complete mission part 2).

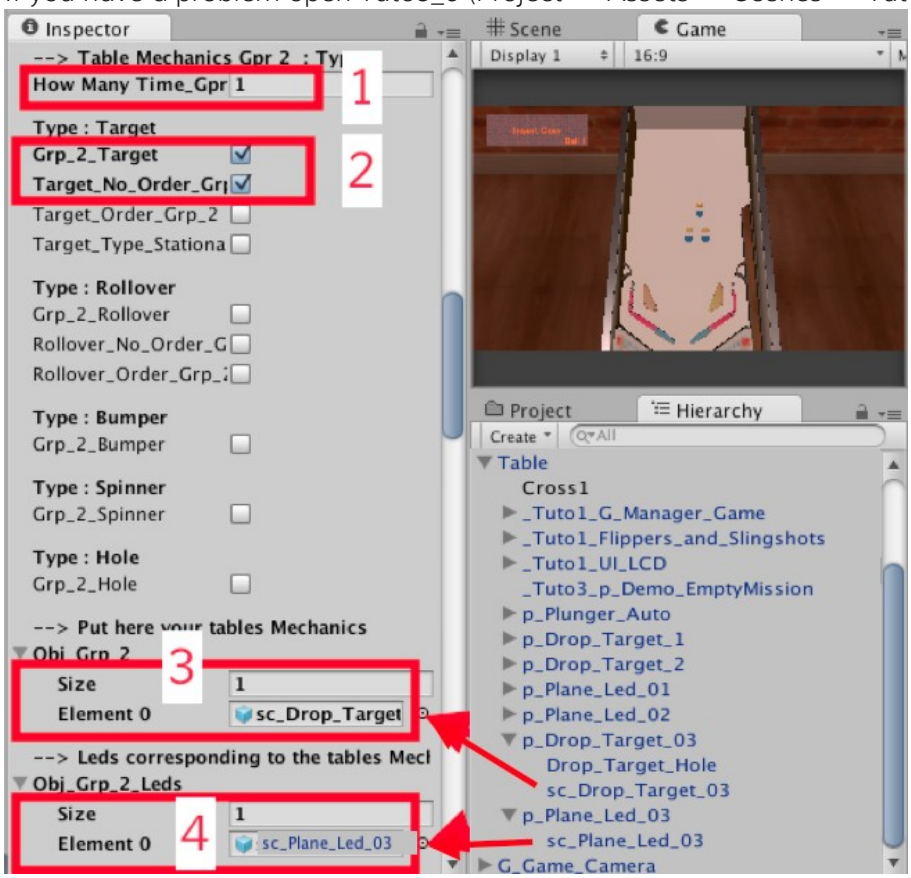
Pic 2 : Check Grp_1_Target. (Check this box because mechanics are targets for mission part 1)
Check Target_No_Order_Grp1 (Check this box because player could hit the target in any order(

Pic 3 : Obj_Grp_2 -> size = 1
Drag'n'drop sc_Drop_Target_3 inside Obj_Grp_2 -> Element 0 .

Pic 4 : Obj_Grp_2_Leds -> size = 1
Drag'n'drop sc_Led_Cercle_1 and sc_Led_Cercle_2 inside Obj_Grp_2_Leds -> Element 0. P

Mission is Complete. Press Play.

If you have a problem open Tuto3_6 (Project -> Assets -> Scenes -> Tuto -> Tuto3 -> Tuto3_6)



Tuto3 is complete.

Mission configuration

Important 1 : Read Tuto 3 First (here).
Important 2 : Each table mechanic must have a unique index. Each mission must have a unique Index
Important 3 : Don't forget to connect your table mechanic to the mission

This section explain how to configure missions. (Script Mission_Start.js)
Mission is divided into two parts (Part1 and Part2). Complete Part 1 to start Part 2. Complete Part 2 to complete the mission.

Configuration : Table of contents

Configuration : MISSSION PART 1 : Case : use TARGET

Step 1 : How many hits to finish Mission Part 1

HowManyTime_Grp1 : Nothing to do for targets. Auto Manage by the script.

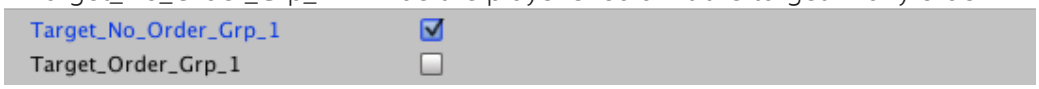
Step 2 : Choose the type of mechanics for mission part 1

Grp_1_Target = True.

Step 3 : Choose Targets options

Check one of the two boxes. Choose between Target_No_Order_Grp_1 or Target_Order_Grp_1.

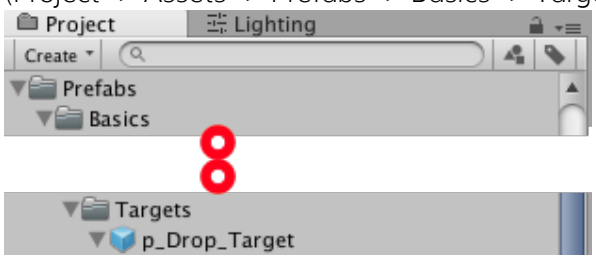
if Target_No_Order_Grp_1 = True the player should hit the target in any order.



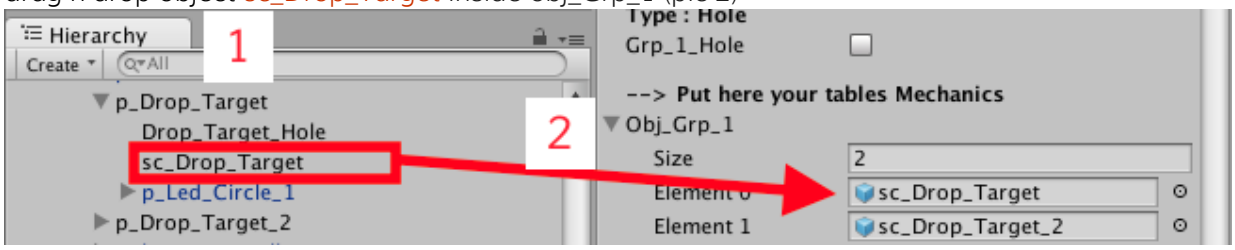
if Target_Order_Grp_1 = True targets appears one by one in the same order as there position on array obj_Grp_1[]

Step 4 : Connect targets to the mission.

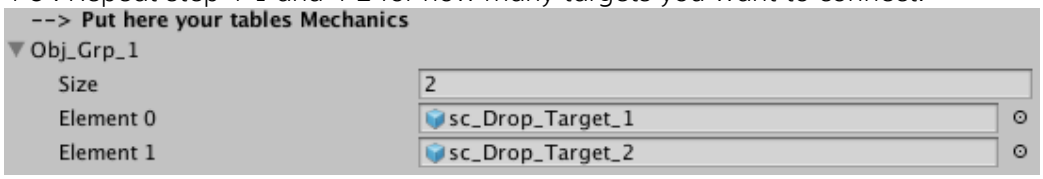
4-1 : Drag'n'drop prefab p_Drop_Target inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Targets -> p_Drop_Target) localPosition (0,0,0)



4-2 : On Hierarchy find object sc_Drop_Target inside your object p_Drop_Target(pic 1). Then drag'n'drop object sc_Drop_Target inside obj_Grp_1 (pic 2)



4-3 : Repeat step 4-1 and 4-2 for how many targets you want to connect.



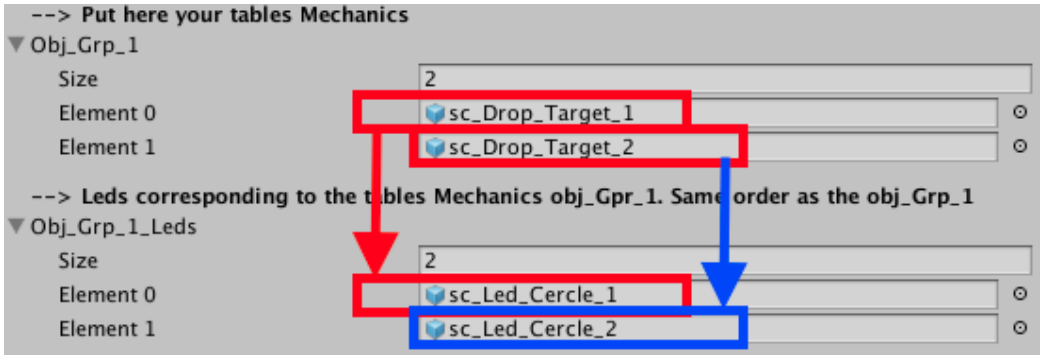
Step 5 : Connect leds to the mission.

5-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

5-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_1

5-3 : Repeat step 5-1 and 5-2 for how many leds you want to connect.

Order is important. The led corresponding to target one (obj_Grp_1 -> Element 0) must have the same position on Obj_Grp_1_leds (Obj_Grp_1_leds -> Element 0)

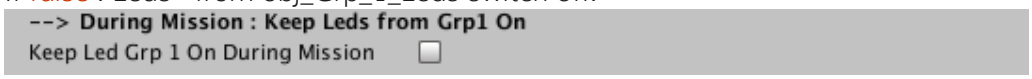


Very Important Inside Obj_Grp_1_leds you must have the same number of leds as there are targets on obj_Grp_1[] .Or you must have no leds on Obj_Grp_1_leds. Other possibilities may create bugs.

Step 6 : Keep Leds from Grp1 ON when Mission Part 2 start.

if keepLedGrp1OnDuringMission = true : Leds from obj_Grp_1_Leds stay ON during mission part 2.

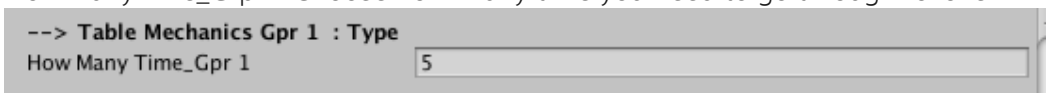
If false : Leds from obj_Grp_1_Leds switch off.



Configuration : MISSSION PART 1 : Case : use Only 1 ROLLOVER

Step 1 : How many time you need to go through rollover to finish Mission Part 1

HowManyTime_Grp1 : Choose how many time you need to go through rollover.



Step 2 : Choose the type of mechanics for mission part 1

Grp_1_Rollover = True.

Step 3 : Choose Rollover options

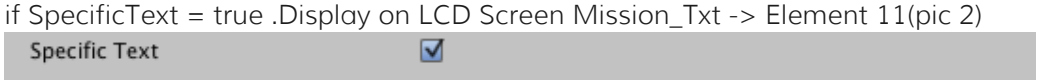
Check one of the two boxes.

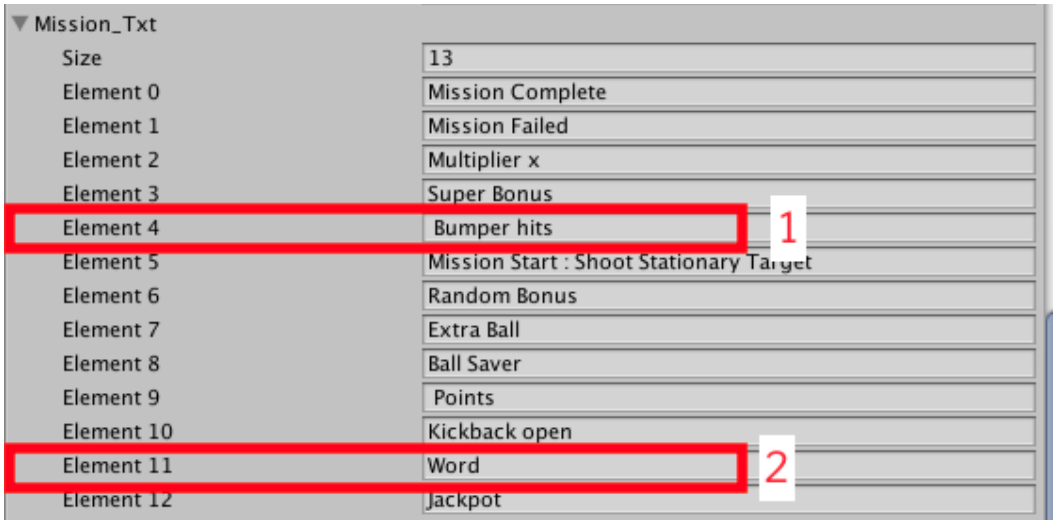
if Rollover_No_Order_Grp_1 = True the player should go through the rollover in any order.

SpecificText is only available for Rollover_No_Order_Grp_1

if SpecificText = false .Display on LCD Screen Mission_Txt -> Element 4 (pic 1)

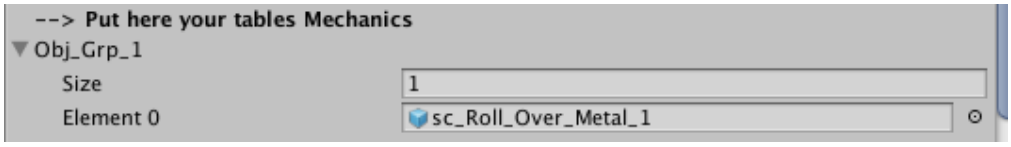
if SpecificText = true .Display on LCD Screen Mission_Txt -> Element 11(pic 2)





Step 4 : Connect Rollover to the mission.
4-1 : Drag'n'drop prefab p_Rollover inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Rollover -> p_Rollover) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Roll_Over_Metal inside your object p_Rollover. Then drag'n'drop object sc_Roll_Over_Metal inside obj_Grp_1

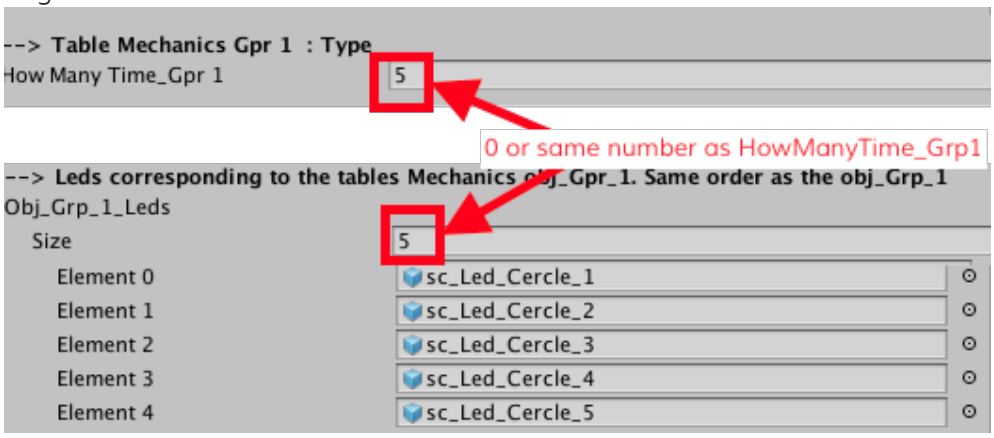


Step 5 : Connect leds to the mission.
5-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

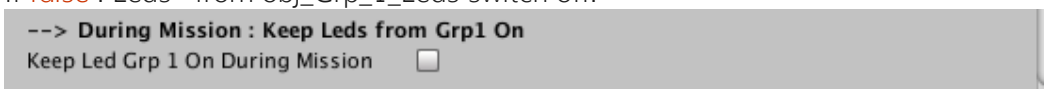
5-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_1

5-3 : Repeat step 5-1 and 5-2 for how many leds you want to connect.

Order is important. Leds indicates mission Part 1 progression
Very Important Inside Obj_Grp_1_leds you must have the same number of leds as HowManyTime_Grp1. Or you must have no leds on Obj_Grp_1_leds. Other possibilities may create bugs.

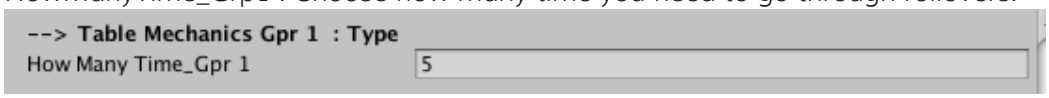


Step 6 : Keep Leds from Grp1 ON when Mission Part 2 start.
if keepLedGrp1OnDuringMission = true : Leds from obj_Grp_1_Leds stay ON during mission part 2.
If false : Leds from obj_Grp_1_Leds switch off.



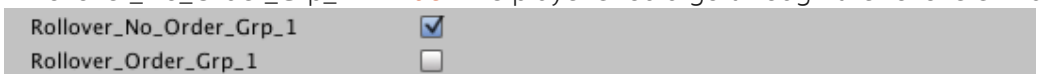
Configuration : MISSSION PART 1 : Case : use more than 1 ROLLOVER

Step 1 : How many time you need to go through rollovers to finish Mission Part 1
HowManyTime_Grp1 : Choose how many time you need to go through rollovers.

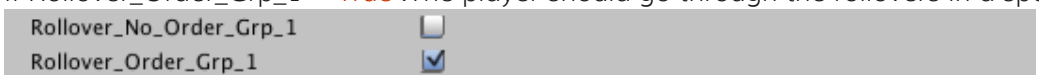


Step 2 : Choose the type of mechanics for mission part 1
Grp_1_Rollover = True.

Step 3 : Choose Rollover options
Check one of the two boxes.
if Rollover_No_Order_Grp_1 = True .The player should go through the rollovers in any order.



if Rollover_Order_Grp_1 = True .The player should go through the rollovers in a specific order.

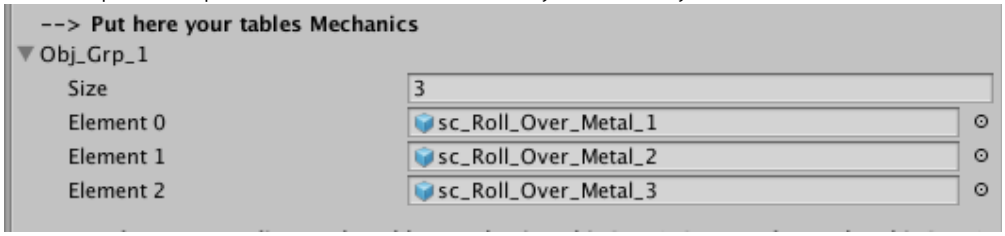


SpecificText is not available for more than 1 Rollover
SpecificText must be equal to false .Display on LCD Screen Mission_Txt -> Element 4

Step 4 : Connect Rollover to the mission.
4-1 : Drag'n'drop prefab p_Rollover inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Rollover -> p_Rollover) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Roll_Over_Metal inside your object p_Rollover. Then drag'n'drop object sc_Roll_Over_Metal inside obj_Grp_1

4-3 : Repeat step 4-1 and 4-2 for how many rollovers you want to connect.

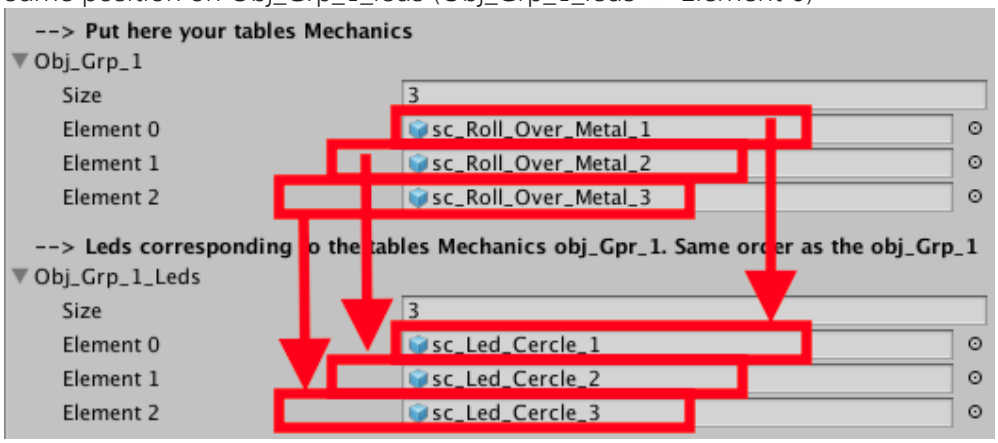


Step 5 : Connect leds to the mission.
5-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

5-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_1

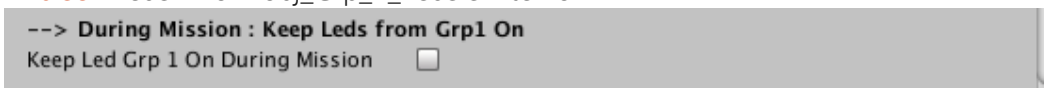
5-3 : Repeat step 5-1 and 5-2 for how many leds you want to connect.

Order is important. The led corresponding to rollover one (obj_Grp_1 -> Element 0) must have the same position on Obj_Grp_1_leds (Obj_Grp_1_leds -> Element 0)



Very Important Inside Obj_Grp_1_leds you must have the same number of leds as there are rollovers on obj_Grp_1 .Or you must have no leds on Obj_Grp_1_leds. Other possibilities may create bugs.

Step 6 : Keep Leds from Grp1 ON when Mission Part 2 start.
if keepLedGrp1OnDuringMission = true : Leds from obj_Grp_1_Leds stay ON during mission part 2.
If false : Leds from obj_Grp_1_Leds switch off.



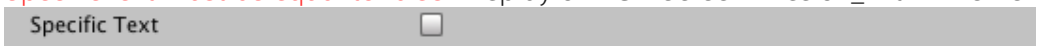
Configuration : MISSSION PART 1 : Case : use ROLLOVER TYPE 3 : (Lane Change)
It is a specific type of rollover. It's available only for Mission Part 1
The player need to switch On all the leds connected to the mission. When the ball go through a rollover the led connected to this rollover switch ON.
When one or more leds are Switch On, the player could modify there position by moving the flippers (Lane Change).

Step 1 : How many time you need to go through rollovers to finish Mission Part 1
HowManyTime_Grp1 : Nothing to do . Auto Manage by the script.

Step 2 : Choose the type of mechanics for mission part 1
Grp_1_Rollover = True.

Step 3 : Choose Rollover options
Rollover_Type3_Grp_1 = True.

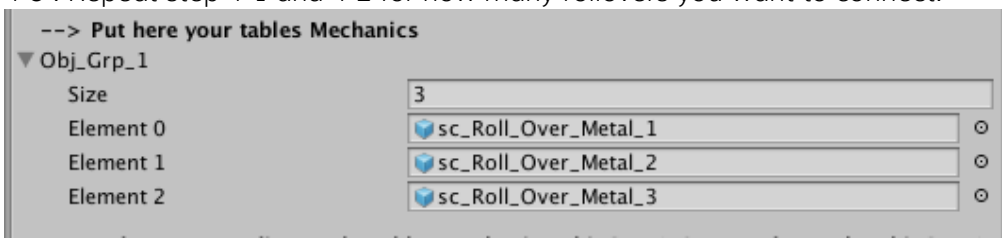
SpecificText must be equal to false .Display on LCD Screen Mission_Txt -> Element 11



Step 4 : Connect Rollover to the mission.
4-1 : Drag'n'drop prefab p_Rollover inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Rollover -> p_Rollover) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Roll_Over_Metal inside your object p_Rollover. Then drag'n'drop object sc_Roll_Over_Metal inside obj_Grp_1

4-3 : Repeat step 4-1 and 4-2 for how many rollovers you want to connect.

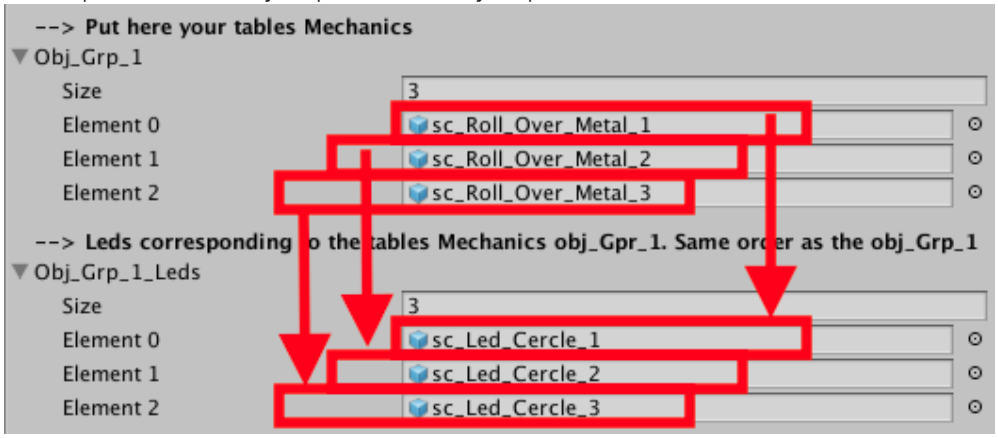


Step 5 : Connect leds to the mission.
5-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

5-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_1

5-3 : Repeat step 5-1 and 5-2 for how many leds you want to connect.

Order is important. The led corresponding to rollover one (obj_Grp_1 -> Element 0) must have the same position on Obj_Grp_1_leds (Obj_Grp_1_leds -> Element 0)

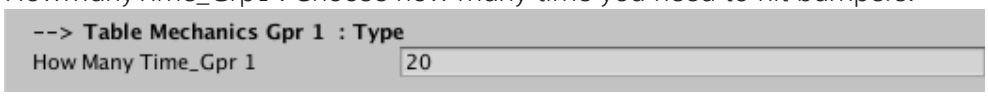


Very Important Inside Obj_Grp_1_leds you must have the same number of leds as there are rollovers on obj_Grp_1[] .Other possibilities may create bugs.

Step 6 : Keep Leds from Grp1 ON when Mission Part 2 start.
if keepLedGrp1OnDuringMission = true : Leds from obj_Grp_1_Leds stay ON during mission part 2.
If false : Leds from obj_Grp_1_Leds switch off.

Configuration : MISSSION PART 1 : Case : use BUMPERS

Step 1 : How many time you need to hit Bumpers to finish Mission Part 1
HowManyTime_Grp1 : Choose how many time you need to hit bumpers.

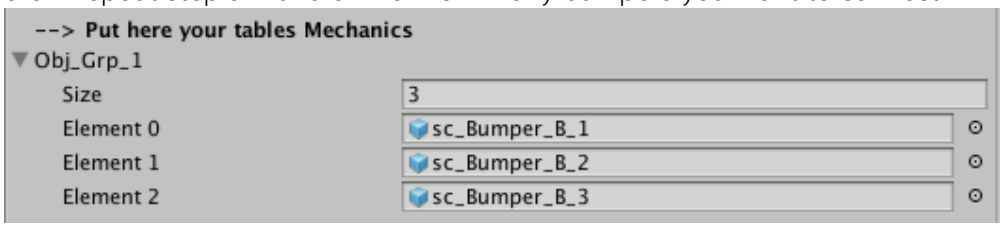


Step 2 : Choose the type of mechanics for mission part 1
Grp_1_Bumper = True.

Step 3 : Connect bumpers to the mission.
3-1 : Drag'n'drop prefab p_Bumper_A inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Bumpers -> p_Bumper_A) localPosition (0,0,0)

3-2 : On Hierarchy find object sc_Bumper_A inside your object p_Bumper_A. Then drag'n'drop object sc_Bumper_A inside obj_Grp_1

3-3 : Repeat step 3-1 and 3-2 for how many bumpers you want to connect.



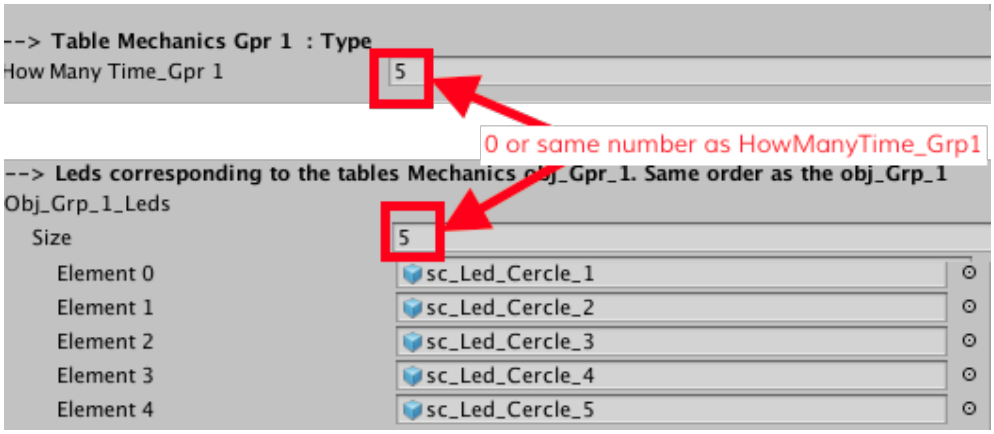
Step 4: Connect leds to the mission.

4-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

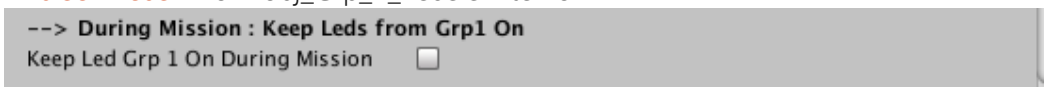
4-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_1

4-3 : Repeat step 4-1 and 4-2 for how many leds you want to connect.

Order is important. Leds indicates mission Part 1 progression
Very Important Inside Obj_Grp_1_leds you must have the same number of leds as HowManyTime_Grp1. Or you must have no leds on Obj_Grp_1_leds. Other possibilities may create bugs.

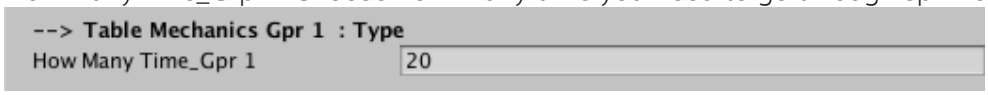


Step 5 : Keep Leds from Grp1 ON when Mission Part 2 start.
if keepLedGrp1OnDuringMission = true : Leds from obj_Grp_1_Leds stay ON during mission part 2.
If false : Leds from obj_Grp_1_Leds switch off.



Configuration : MISSSION PART 1 : Case : use SPINNER

Step 1 : How many time you need to go through spinner to finish Mission Part 1
HowManyTime_Grp1 : Choose how many time you need to go through spinner .

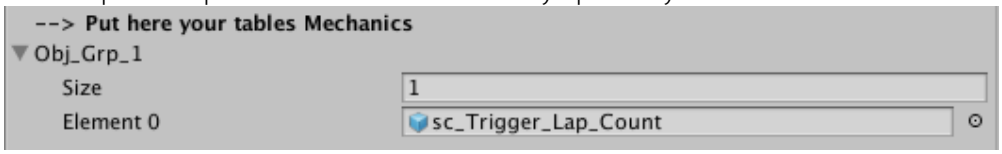


Step 2 : Choose the type of mechanics for mission part 1
Grp_1_Spinner = True.

Step 3 : Connect Spinner to the mission.
3-1 : Drag'n'drop prefab p_Spinner inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Spinner -> p_Spinner) localPosition (0,0,0)

3-2 : On Hierarchy find object Trigger_Lap_Count inside your object p_Spinner. Then drag'n'drop object Trigger_Lap_Count inside obj_Grp_1

3-3 : Repeat step 3-1 and 3-2 for how many spinner you want to connect.



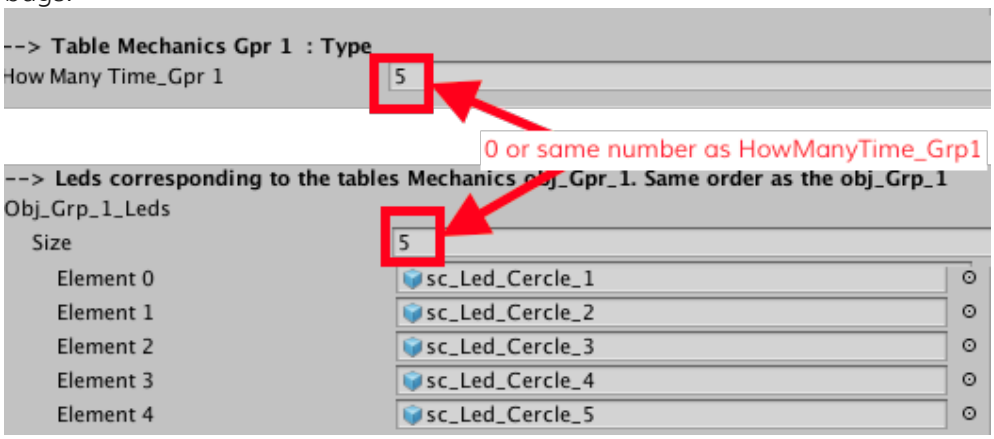
Step 4: Connect leds to the mission.

4-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_1

4-3 : Repeat step 4-1 and 4-2 for how many leds you want to connect.

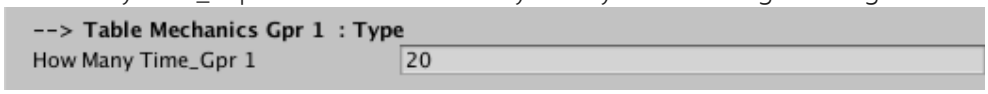
Order is important. Leds indicates mission Part 1 progression
Very Important Inside Obj_Grp_1_leds you must have the same number of leds as HowManyTime_Grp1. Or you must have no leds on Obj_Grp_1_leds. Other possibilities may create bugs.



Step 5 : Keep Leds from Grp1 ON when Mission Part 2 start.
if keepLedGrp1OnDuringMission = true : Leds from obj_Grp_1_Leds stay ON during mission part 2.
If false : Leds from obj_Grp_1_Leds switch off.

Configuration : MISSSION PART 1 : Case : use HOLE

Step 1 : How many time you need to go through spinner to finish Mission Part 1
HowManyTime_Grp1 : Choose how many time you need to go through Hole .



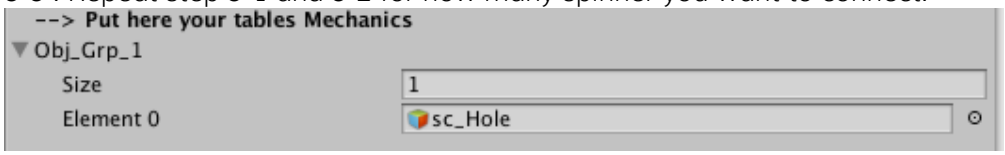
Step 2 : Choose the type of mechanics for mission part 1
Grp_1_Hole = True.

Step 3 : Connect Hole to the mission.
3-1 : Drag'n'drop prefab p_Hole inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Hole -> p_Hole) localPosition (0,0,0)

3-2 : On Hierarchy find object sc_Hole inside your object p_Hole. Then drag'n'drop object sc_Hole

inside obj_Grp_1

3-3 : Repeat step 3-1 and 3-2 for how many spinner you want to connect.

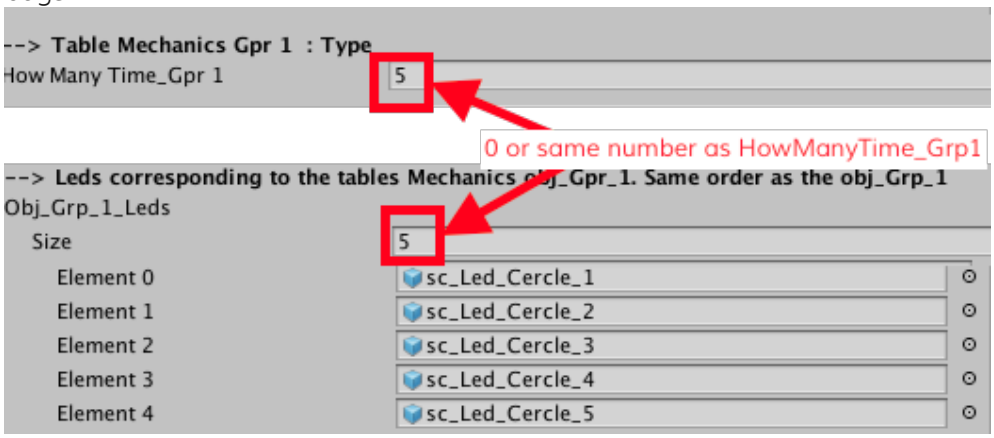


Step 4: Connect leds to the mission.
4-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

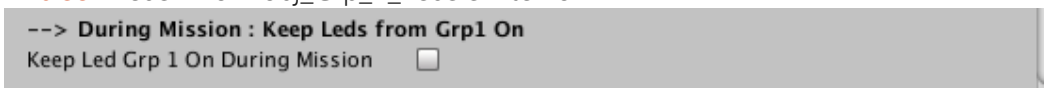
4-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_1

4-3 : Repeat step 4-1 and 4-2 for how many leds you want to connect.

Order is important. Leds indicates mission Part 1 progression
Very Important Inside Obj_Grp_1_leds you must have the same number of leds as HowManyTime_Grp1. Or you must have no leds on Obj_Grp_1_leds. Other possibilities may create bugs.



Step 5 : Keep Leds from Grp1 ON when Mission Part 2 start.
if keepLedGrp1OnDuringMission = true : Leds from obj_Grp_1_Leds stay ON during mission part 2.
If false : Leds from obj_Grp_1_Leds switch off.



Configuration : MISSSION PART 2 : Case : use TARGET

Step 1 : How many hits to finish Mission Part 2
HowManyTime_Grp2 : Nothing to do for targets. Auto Manage by the script.

Step 2 : Choose the type of mechanics for mission part 2

Grp_2_Target = True.

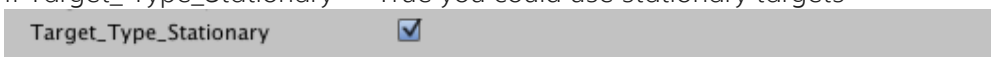
Step 3 : Choose Targets options

Check one of the three boxes. Choose between Target_No_Order_Grp_2, Target_Order_Grp_2 or Target_Type_Stationary. Important : Stationary target only support no order

if Target_No_Order_Grp_2 = True the player should hit the target in any order.

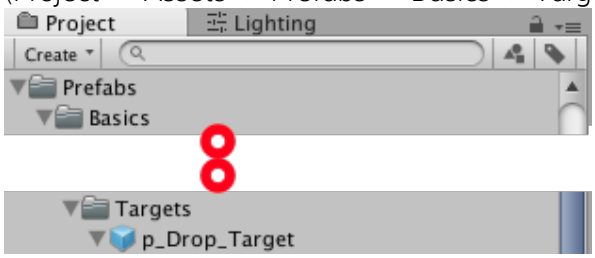
if Target_Order_Grp_2 = True targets appears one by one in the same order as there position on array obj_Grp_2[]

if Target_ Type_Stationary = True you could use stationary targets

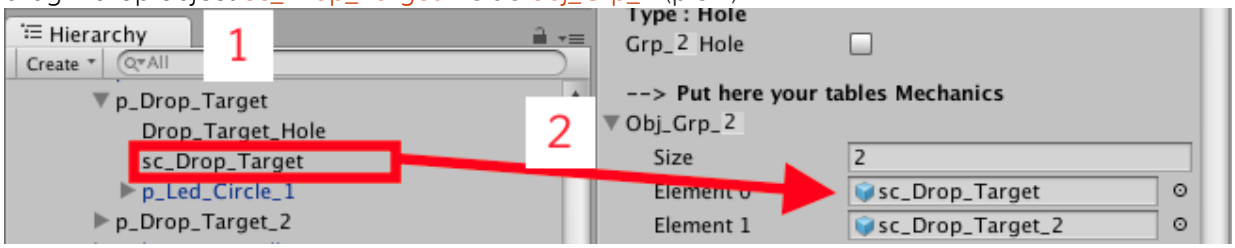


Step 4 : Connect targets to the mission.

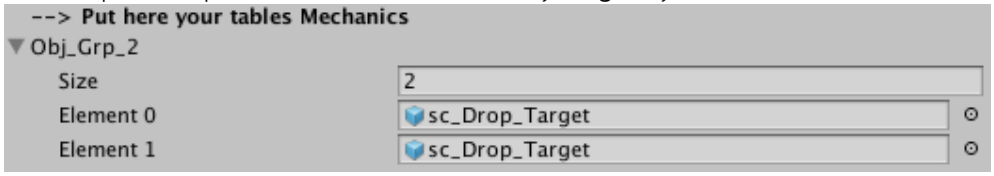
4-1 : Drag'n'drop prefab p_Drop_Target inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Targets -> p_Drop_Target) localPosition (0,0,0)



4-2 : On Hierarchy find object sc_Drop_Target inside your object p_Drop_Target(pic 1). Then drag'n'drop object sc_Drop_Target inside obj_Grp_2 (pic 2)



4-3 : Repeat step 4-1 and 4-2 for how many targets you want to connect.



Step 5 : Connect leds to the mission.

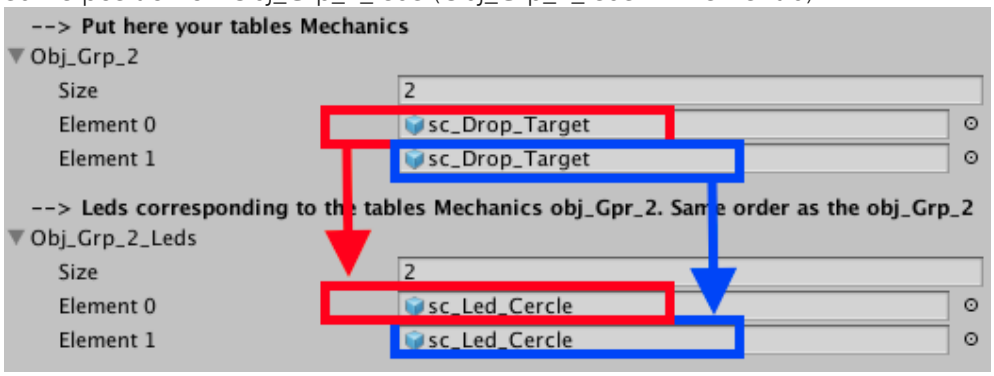
5-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

5-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_2

5-3 : Repeat step 5-1 and 5-2 for how many leds you want to connect.

Order is important. The led corresponding to target one (obj_Grp_2 -> Element 0) must have the

same position on Obj_Grp_2_leds (Obj_Grp_2_leds -> Element 0)

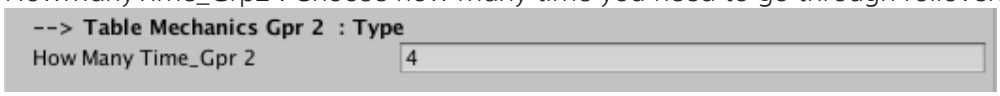


Very Important Inside Obj_Grp_2_leds you must have the same number of leds as there are targets on obj_Grp_2 .Or you must have no leds on Obj_Grp_2_leds. Other possibilities may create bugs.

Configuration : MISSSION PART 2 : Case : use Only 1 ROLLOVER

Step 1 : How many time you need to go through rollover to finish Mission Part 2

HowManyTime_Grp2 : Choose how many time you need to go through rollover.

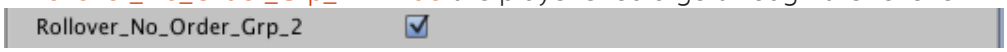


Step 2 : Choose the type of mechanics for mission part 2

Grp_2_Rollover = True.

Step 3 : Choose Rollover options

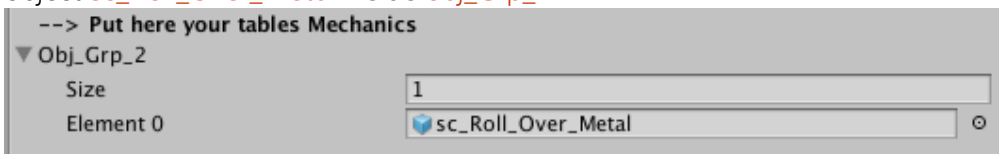
if Rollover_No_Order_Grp_2 = True the player should go through the rollover in any order.



Step 4 : Connect Rollover to the mission.

4-1 : Drag'n'drop prefab p_Rollover inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Rollover -> p_Rollover) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Roll_Over_Metal inside your object p_Rollover. Then drag'n'drop object sc_Roll_Over_Metal inside obj_Grp_2



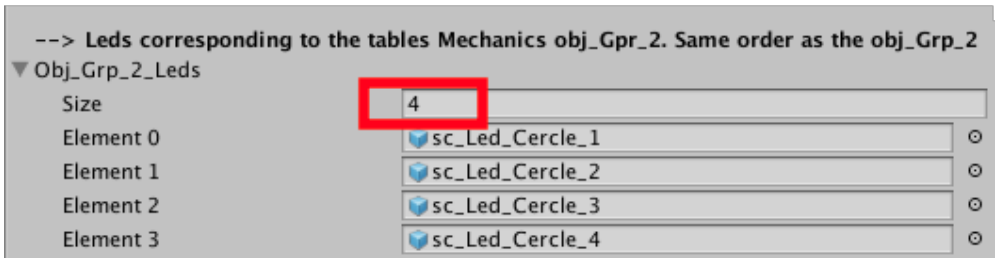
Step 5 : Connect leds to the mission.

5-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

5-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_2

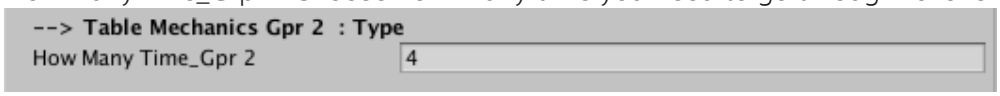
5-3 : Repeat step 5-1 and 5-2 for how many leds you want to connect.

Order is important. Leds indicates mission Part 2 progression
Very Important Inside Obj_Grp_2_leds you must have the same number of leds as HowManyTime_Grp2. Or you must have no leds on Obj_Grp_2_leds. Other possibilities may create bugs.



Configuration : MISSSION PART 2 : Case : use more than 1 ROLLOVER

Step 1 : How many time you need to go through rollovers to finish Mission Part 2
HowManyTime_Grp2 : Choose how many time you need to go through rollovers.



Step 2 : Choose the type of mechanics for mission part 2
Grp_2_Rollover = True.

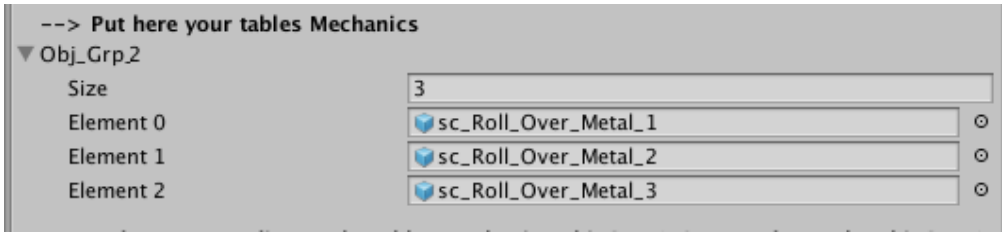
Step 3 : Choose Rollover options
Check one of the two boxes.
if Rollover_No_Order_Grp_2 = True .The player should go through the rollovers in any order.

if Rollover_Order_Grp_2 = True .The player should go through the rollovers in a specific order.

Step 4 : Connect Rollover to the mission.
4-1 : Drag'n'drop prefab p_Rollover inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Rollover -> p_Rollover) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Roll_Over_Metal inside your object p_Rollover. Then drag'n'drop object sc_Roll_Over_Metal inside obj_Grp_2

4-3 : Repeat step 4-1 and 4-2 for how many rollovers you want to connect.

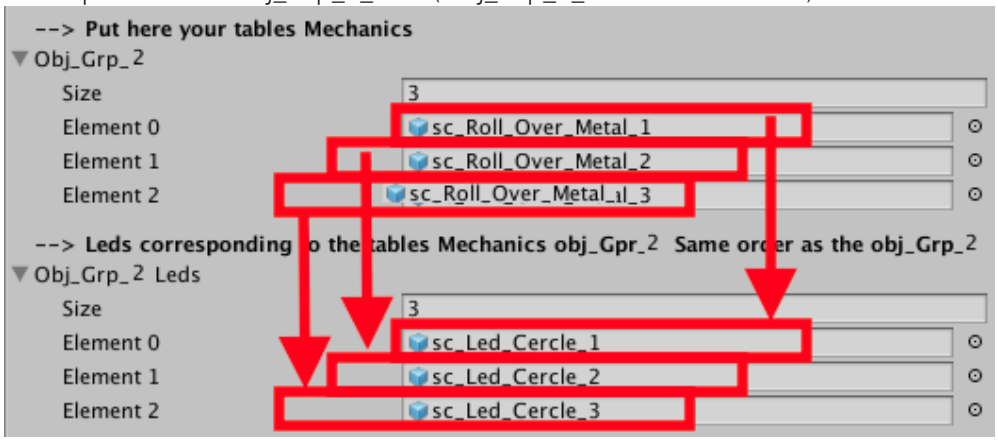


Step 5 : Connect leds to the mission.
5-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

5-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_2

5-3 : Repeat step 5-1 and 5-2 for how many leds you want to connect.

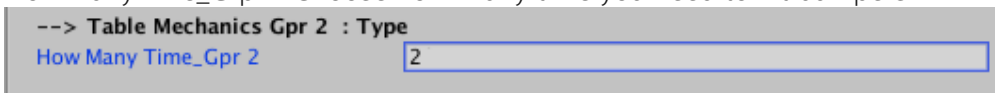
Order is important. The led corresponding to rollover one (obj_Grp_2 -> Element 0) must have the same position on Obj_Grp_2_leds (Obj_Grp_2_leds -> Element 0)



Very Important Inside Obj_Grp_2_leds you must have the same number of leds as there are rollovers on obj_Grp_2[] .Or you must have no leds on Obj_Grp_2_leds. Other possibilities may create bugs.

Configuration : MISSSION PART 2 : Case : use BUMPERS

Step 1 : How many time you need to hit Bumpers to finish Mission Part 2
HowManyTime_Grp2 : Choose how many time you need to hit bumpers.

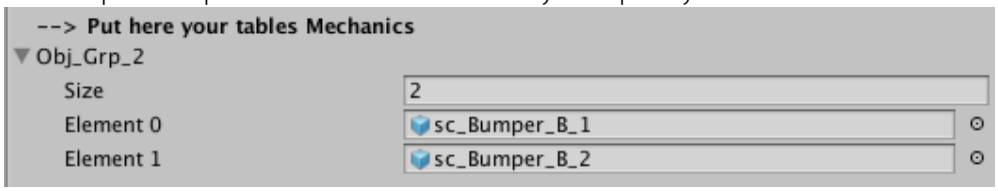


Step 2 : Choose the type of mechanics for mission part 2
Grp_2_Bumper = True.

Step 3 : Connect bumpers to the mission.
3-1 : Drag'n'drop prefab p_Bumper_A inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Bumpers -> p_Bumper_A) localPosition (0,0,0)

3-2 : On Hierarchy find object sc_Bumper_A inside your object p_Bumper_A. Then drag'n'drop object sc_Bumper_A inside obj_Grp_2

3-3 : Repeat step 3-1 and 3-2 for how many bumpers you want to connect.

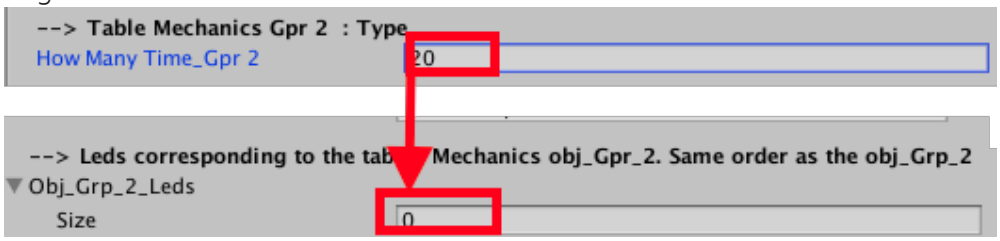


Step 4: Connect leds to the mission.
4-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_2

4-3 : Repeat step 4-1 and 4-2 for how many leds you want to connect.

Order is important. Leds indicates mission Part 2 progression
Very Important Inside Obj_Grp_2_leds you must have the same number of leds as HowManyTime_Grp2. Or you must have no leds on Obj_Grp_2_leds. Other possibilities may create bugs.



Configuration : MISSSION PART 2 : Case : use SPINNER

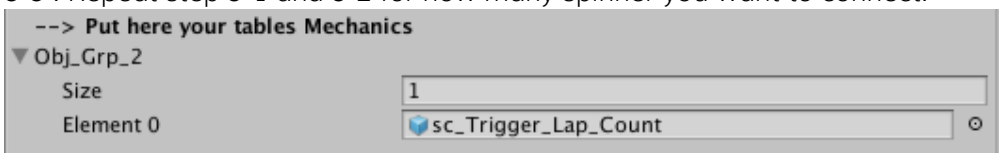
Step 1 : How many time you need to go through spinner to finish Mission Part 2
HowManyTime_Grp2 : Choose how many time you need to go through spinner .

Step 2 : Choose the type of mechanics for mission part 2
Grp_2_Spinner = True.

Step 3 : Connect Spinner to the mission.
3-1 : Drag'n'drop prefab p_Spinner inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Spinner -> p_Spinner) localPosition (0,0,0)

3-2 : On Hierarchy find object Trigger_Lap_Count inside your object p_Spinner. Then drag'n'drop object Trigger_Lap_Count inside obj_Grp_2

3-3 : Repeat step 3-1 and 3-2 for how many spinner you want to connect.

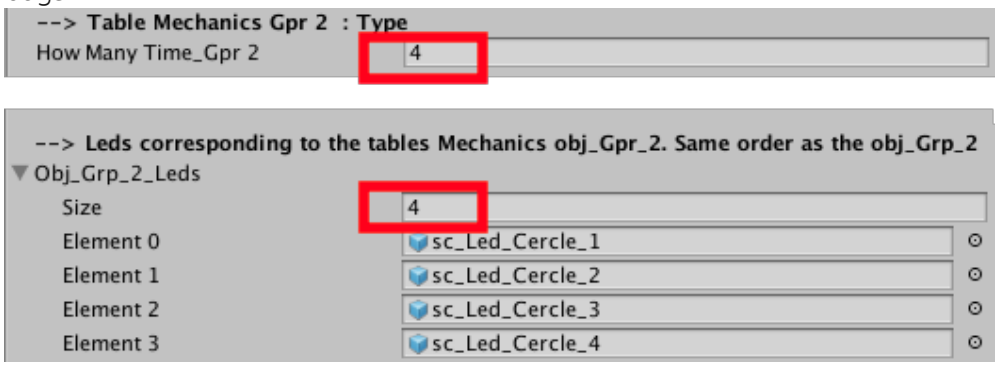


Step 4: Connect leds to the mission.
4-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_2

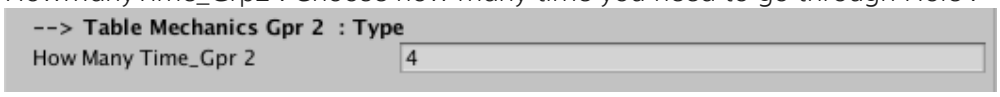
4-3 : Repeat step 4-1 and 4-2 for how many leds you want to connect.

Order is important. Leds indicates mission Part 2 progression
Very Important Inside Obj_Grp_2_leds you must have the same number of leds as HowManyTime_Grp2. Or you must have no leds on Obj_Grp_2_leds. Other possibilities may create bugs.



Configuration : MISSSION PART 2 : Case : use HOLE

Step 1 : How many time you need to go through spinner to finish Mission Part 2
HowManyTime_Grp2 : Choose how many time you need to go through Hole .

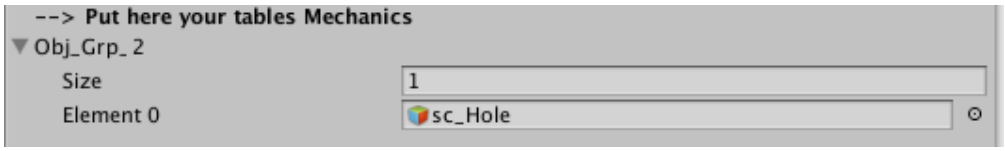


Step 2 : Choose the type of mechanics for mission part 2
Grp_2_Hole = True.

Step 3 : Connect Hole to the mission.
3-1 : Drag'n'drop prefab p_Hole inside GameObject Table on Hierarchy
(Project -> Assets -> Prefabs -> Basics -> Hole -> p_Hole) localPosition (0,0,0)

3-2 : On Hierarchy find object sc_Hole inside your object p_Hole. Then drag'n'drop object sc_Hole inside obj_Grp_2

3-3 : Repeat step 3-1 and 3-2 for how many spinner you want to connect.

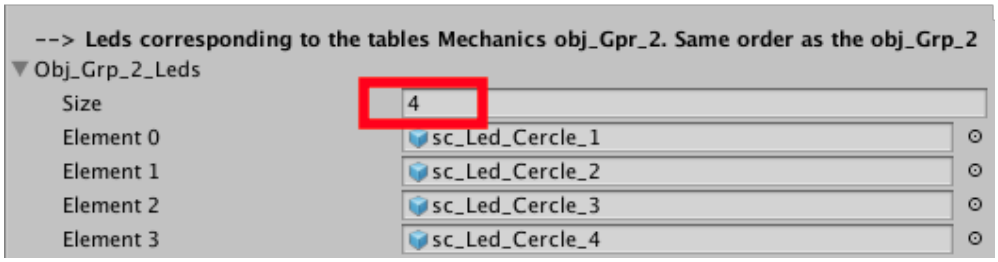


Step 4: Connect leds to the mission.
4-1 : Drag'n'drop prefab p_Led_Circle inside GameObject Table on hierarchy
(Project -> Assets -> Prefabs -> Basics -> Leds -> p_Led_Circle) localPosition (0,0,0)

4-2 : On Hierarchy find object sc_Led_Cercle inside your object p_Led_Circle. Then drag'n'drop object sc_Led_Cercle inside obj_Led_Grp_2

4-3 : Repeat step 4-1 and 4-2 for how many leds you want to connect.

Order is important. Leds indicates mission Part 2 progression
Very Important Inside Obj_Grp_2_leds you must have the same number of leds as HowManyTime_Grp2. Or you must have no leds on Obj_Grp_2_leds. Other possibilities may create bugs.

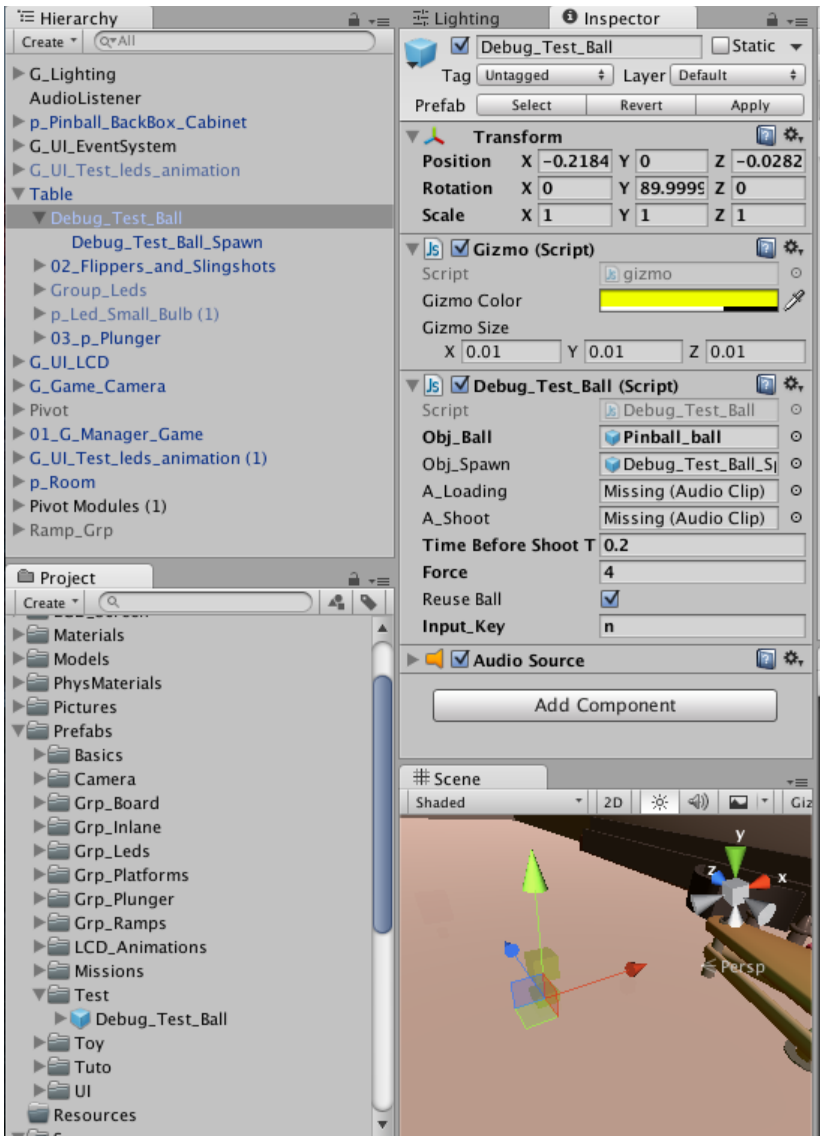


Test Mission with Debug_Test_Ball :

Debug_Test_Ball is used to help you test mission.

Step 1 :
Open Project -> Assets -> Scenes -> Tuto -> Tuto6 -> Tuto6

Step 2 :
Open Hierachy -> Table and select Debug_Test_Ball (pic 1).



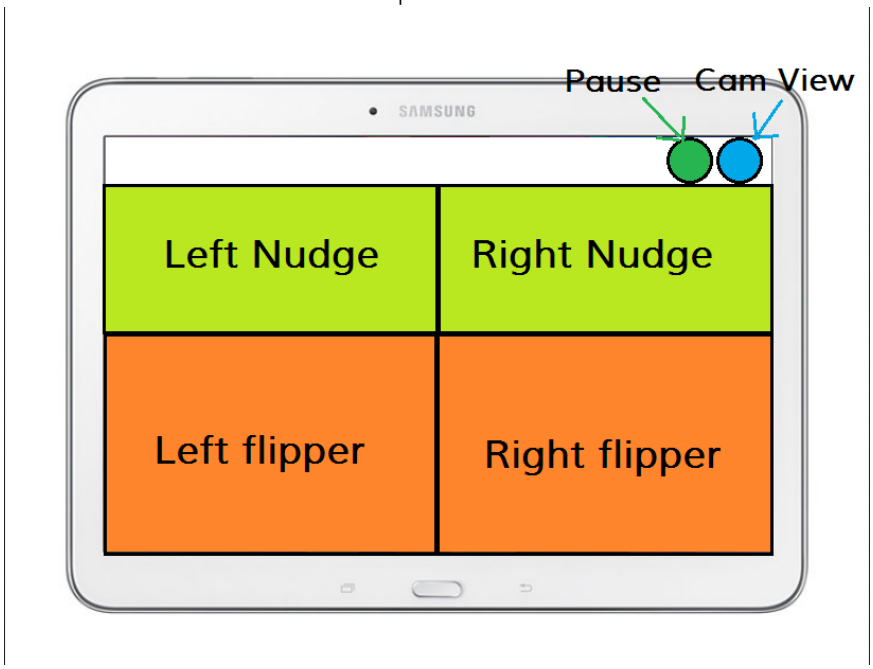
Step 3 :
Start Play Mode.

Press n : A ball spawn.
Release n : ball is ejected on Z Debug_Test_Ball direction.

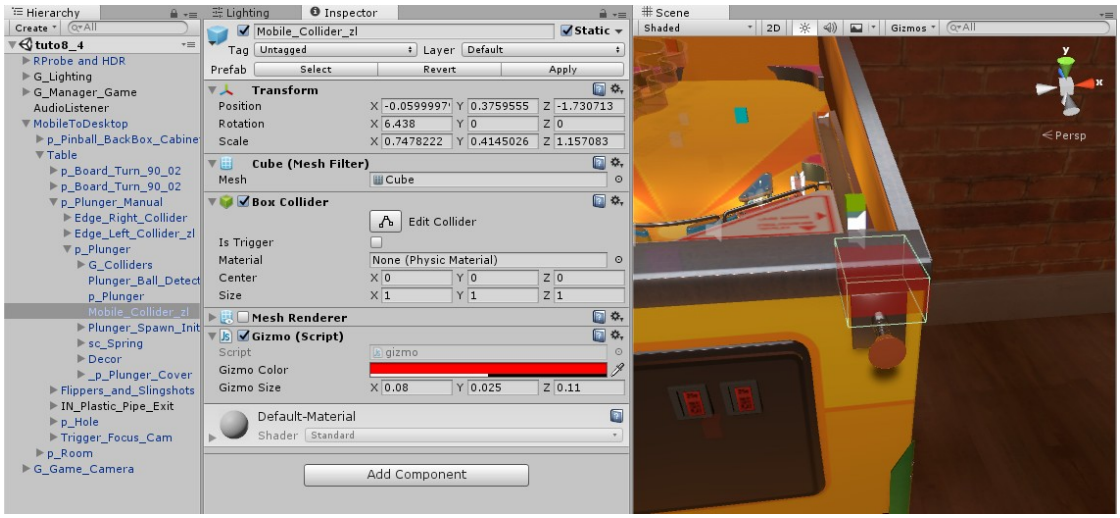
You could put as many Debug_Test_Ball as you want. Choose a unique input for each of them.
It is an easy way to test mission.

Mobile Control :

Find here how buttons are setup on a mobile device.



To use the plunger put your finger on plunger to pull the spring. Then remove your finger from the plunger to eject the ball.
We use raycast to know if the player press is touching the gameObject Mobile_Collider_zl



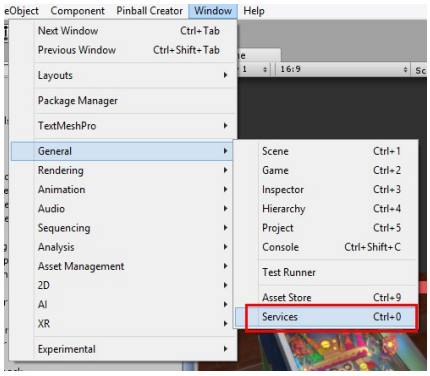
Mobile Input are manage on scripts :

Manager_Game.js
Flippers.js
Spring_Launcher

Unity Ads

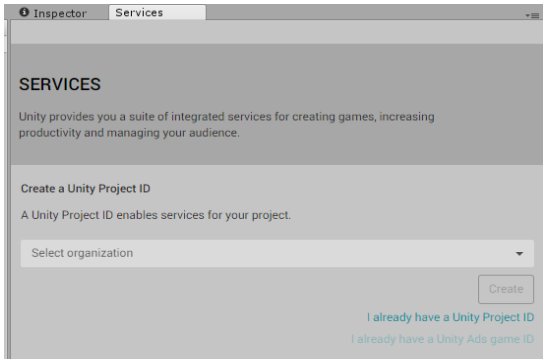
Unity Ads : How to setup

1 Select Menu→ Window → General → Services



2 Select your Organization on the list.

If you don't have a Unity ID go to : https://id.unity.com/en/conversations/7c02d4ab-c395-4ea0-bae3-4ea84cd36dff006f?view=register



3 Press button create

4 Enable Ads by pressing the button Off (spot 1)

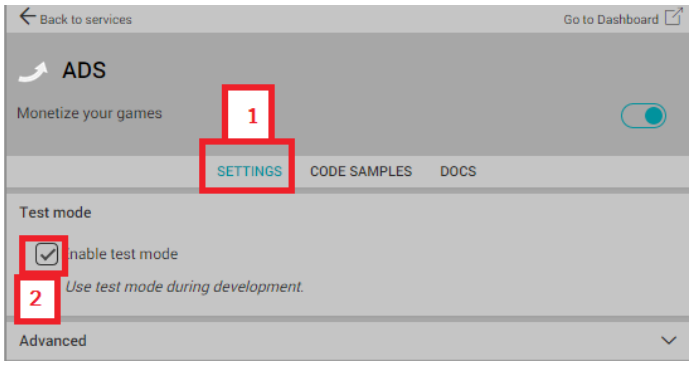
5 Press the switch (spot 1)

6 Press button continue.

7 Next screen appear. Press switch (spot 1)

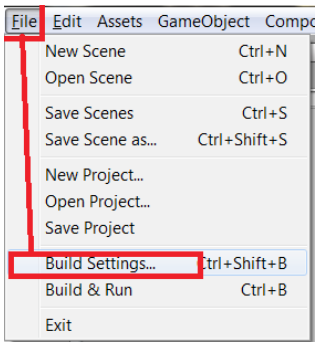
8 Go to Setting section (spot 1)

-Enable test Mode.(Don't forget to disable it when you released your game) (spot 2)

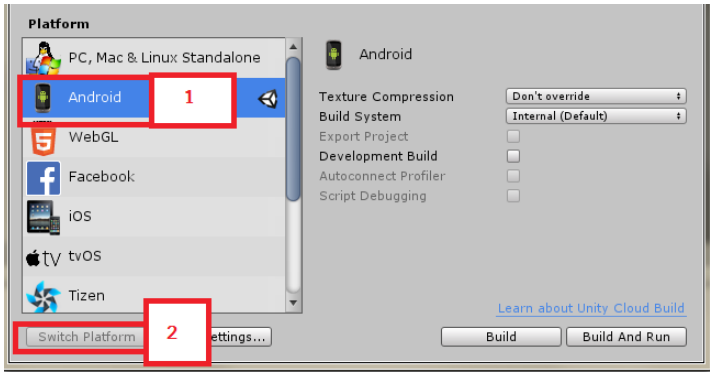


9 IMPORTANT :
Test Mode only works if you've choose a mobile platform the build settings.

Menu → Build Settings

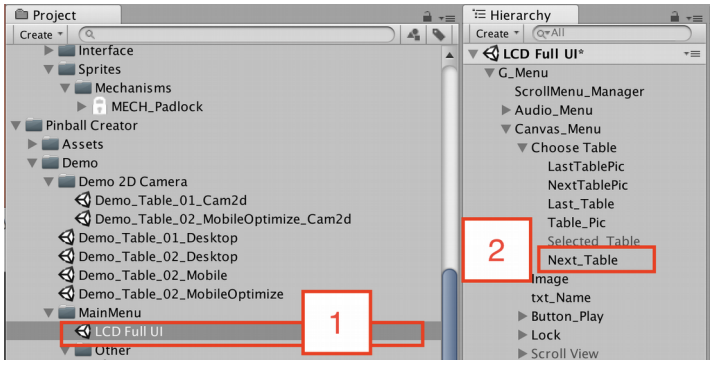


If you are not a mobile platform. Select Android or iOS (spot 1)
Then press switch Platform (spot 2).



Unity Ads : Launch Ads when player pressed a button

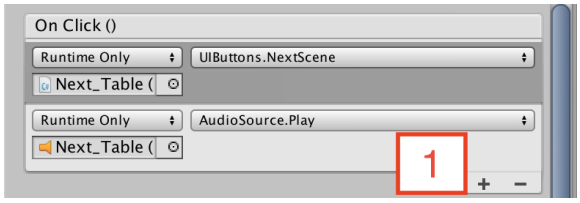
Example : Add Ads on button Next_Table. Before doing these steps you need to setup Unity Ads (more info here)



1 Open scene LCD Full UI (Project Tab → Pinball Creator → Demo → MainMenu → LCD Full UI)

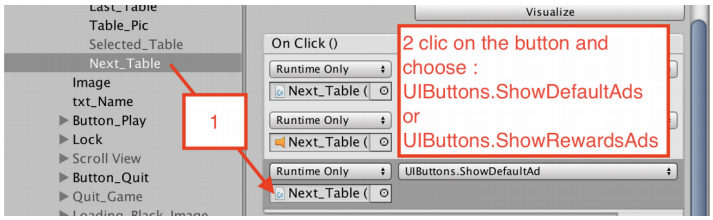
2 On Hierarchy select the button Next_Table (Hierarchy → G_Menu → Canvas_Menu → Choose Table → Next_Table)

3 On the Inspector go to the section OnClick and press the button + (Spot 1)



4 Drag and drop your button inside the new empty slot (spot (1)
(The button need to have the UIButtons.cs attached to it)

- Choose on the list the function UIButtons.ShowDefaultAd (spot 2).

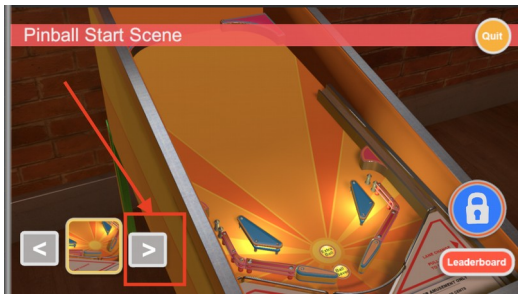


Info : You could choose the function UIButtons.ShowRewardedAds() if you want to use a reward Ads

IMPORTANT : You need to code your reward on script UI_Buttons.cs function F_Rewards()

5 Start your scene.

6 Press the button Next_Table on your scene. An Ad starts.



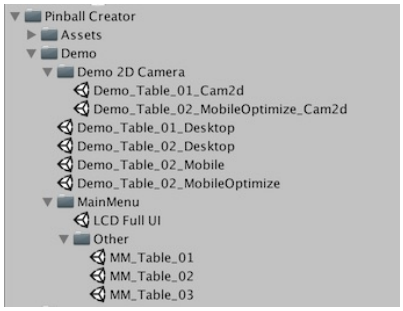
Main Menu

Main Menu : Overview (Read First)

Main menu allow to choose a table, launch a table, see leaderboards and quit game.

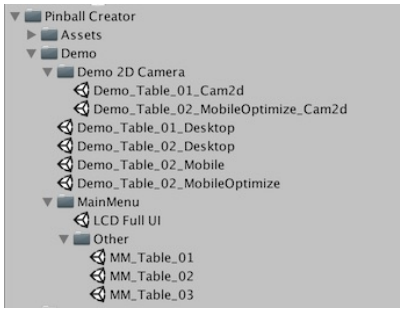
You could use our demo scene as a starting point.

1 Open scene LCD Full UI (Project Tab → Pinball Creator → Demo → MainMenu → LCD Full UI)

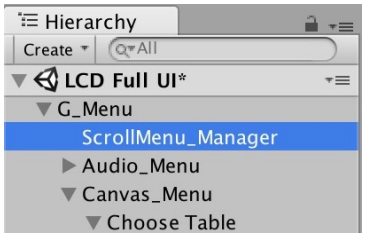


Main Menu : How to add a new table

1 Open scene LCD Full UI (Project Tab → Pinball Creator → Demo → MainMenu → LCD Full UI)

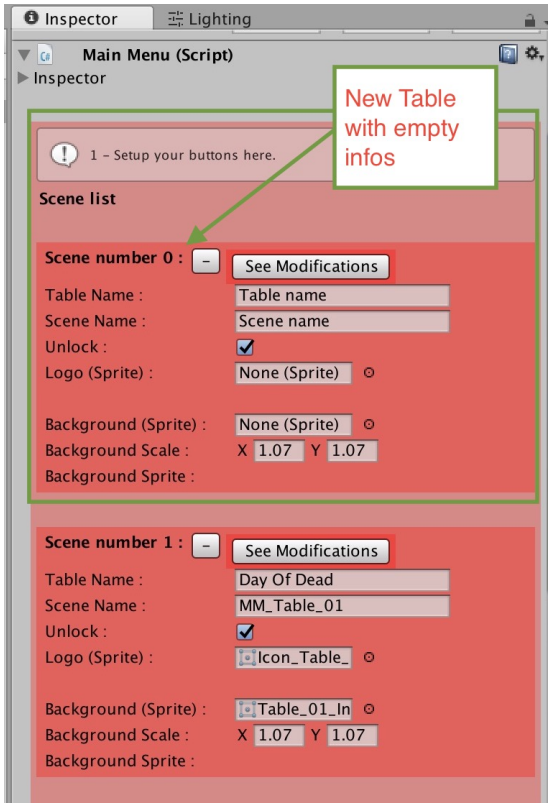


2 On the Hierarchy select the gameObject ScrollMenu_Manager



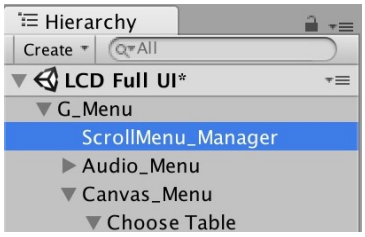
3 On the Inspector press button : Add new Table

4 A new table is created at the top of the other tables

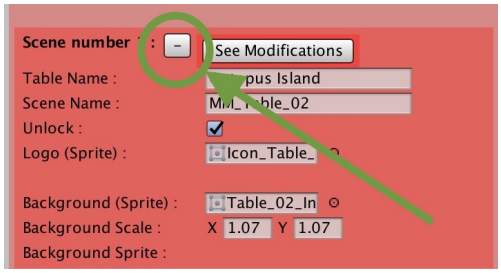


Main Menu : How to delete a table

1 On the Hierarchy select the gameObject ScrollMenu_Manager



2 On the Inspector press the button - beside the table you want to delete.



Main Menu : How to setup a table

1 On the Hierarchy select the gameObject ScrollMenu_Manager

2 To see your modifications on a table press button See Modifications
Button is surrounded in yellow

3 You could choose :
the name of the table that appear on the left up corner of the main menu (Spot1)

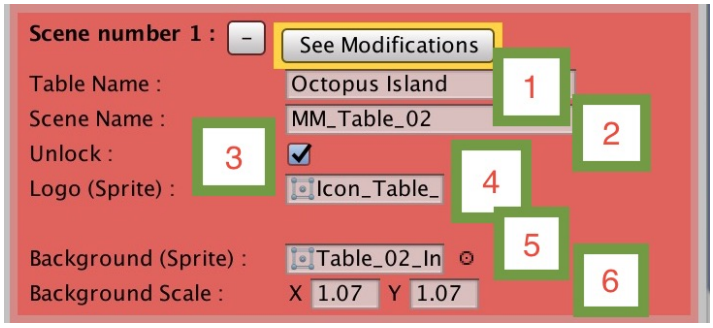
- The name of the scene you want to load for this table. (spot 2)

- If the table table is locked or not when the game start (spot 3)

- Your icon table (Drag and drop a sprite from your project folder) (spot 4)

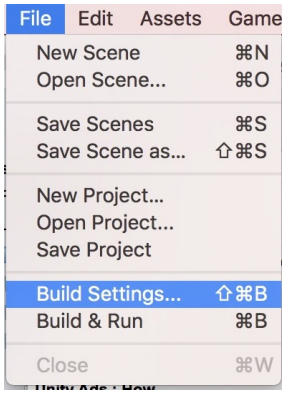
- Your background (Drag and drop a sprite from your project folder) (spot 5)

- the scale of your background sprite



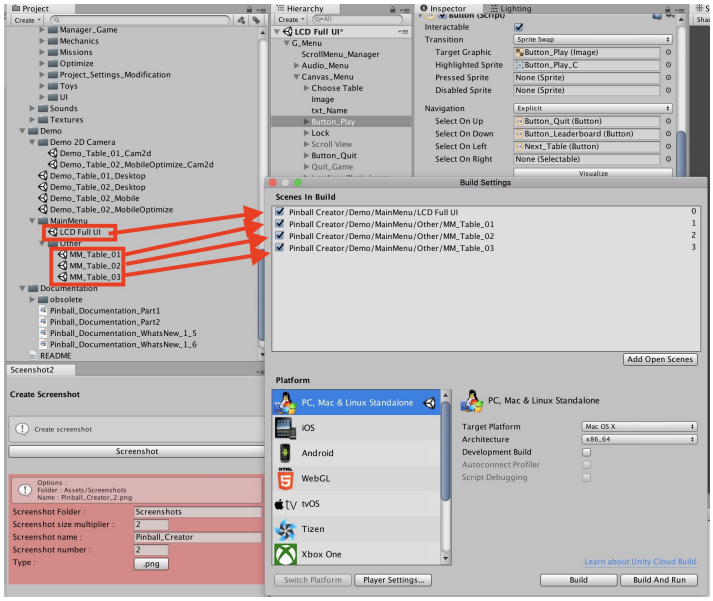
Main Menu : Add your scene to the build Menu

1 Open File → Build Settings



2 Drag and drop your scene on Scene in Build section

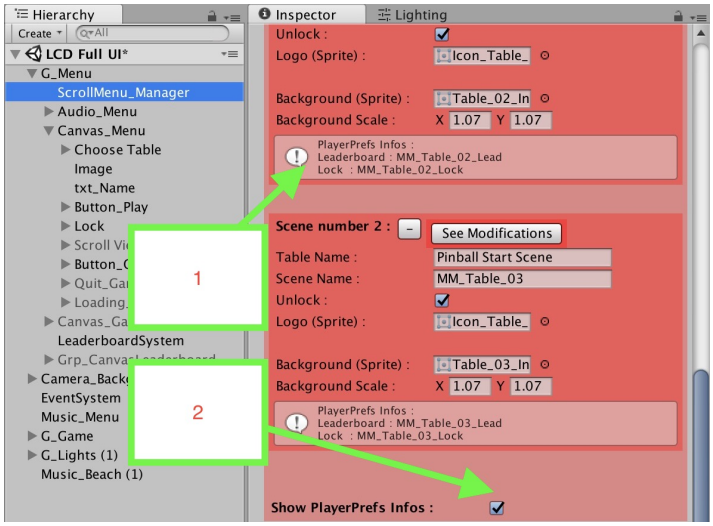
Put the scene LCD Full UI first. It the first scene that Unity need to load



Main Menu : (Script) Access Table leaderboard and Lock PlayerPrefs

It is possible to see PlayerPrefs Infos by checking the button
Show Player Prefs Infos (spot 1)

Then you could see Leaderboard playerPrefs name and Lock playerPrefs for each table (spot 2)



Call the Leaderboard : PlayerPrefs.GetString(Name of your table scene + ‘’_Lead’’)

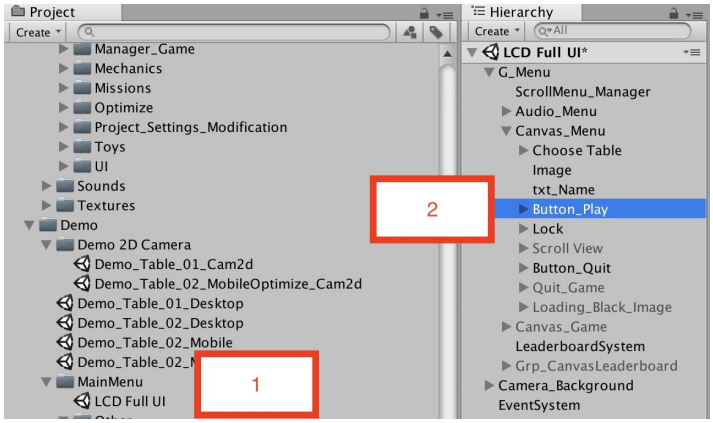
Call the PlayerPrefs to know if the table is locked : PlayerPrefs.GetString(Name of your table scene + ‘’_Lock’’)

The lock is unlocked if PlayerPrefs.GetString(Name of your table scene + ‘’_Lock’’) == ‘’Unlocked’’

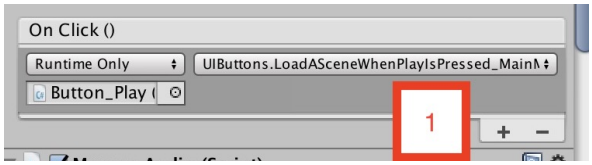
Main Menu : Unlock table with an Ad or anything else

1 Open scene LCD Full UI (Project Tab → Pinball Creator → Demo → MainMenu → LCD Full UI)

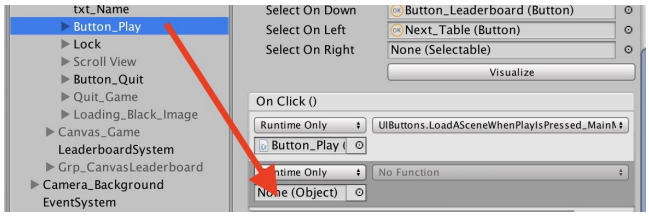
2 On Hierarchy select the button ‘’Button_Play’’ (Hierarchy → G_Menu → Canvas_Menu → Button_Play)



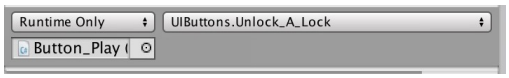
3 On the Inspector go to the section OnClick and press the button + (Spot 1)



4 Drag and drop your button inside the new empty slot
(The button need to have the UIButtons.cs attached to it)



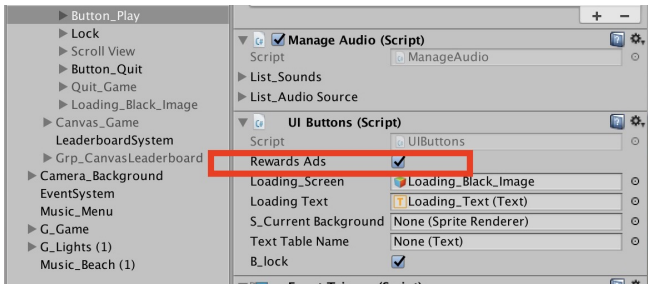
- Choose on the list the function UIButtons.Unlock_A_Lock



Case A : If Unity Ads is enable (more info here)

Check the Box Rewards_Ads

Case B : Unity Ads is not setup.
You could code the function UnlockIfNo_UnityAds() on script UIButtons.cs



5 Start your scene.

6 Press the button Button_Play on your scene. An Ad Rewards starts.

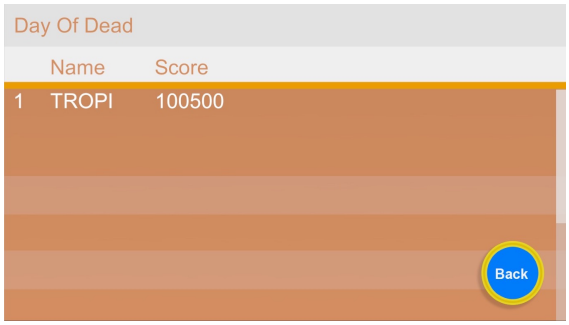
When the ad is closed the lock is unlocked

Leaderboard : Overview (read first)

1 Player could save his name score when he is game over.

2 Player could access the leaderboard

from the main menu

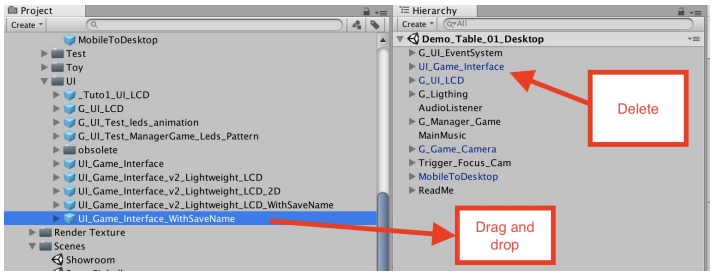


Leaderboard : How to add the prefab that allow to save name and score on a table

Case 1 : You use UI_Game_Interface gameobject on your scene

1 Delete UI_Game_Interface gameobject on your scene

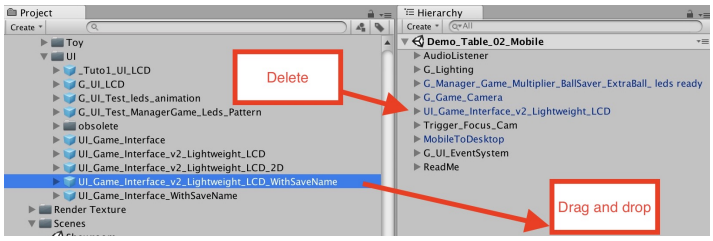
2 Drag and drop the prefab UI_Game_Interface_WithSaveName on the root o f the scene
(Pinball Creator→Assets→Prefabs→UI→)



Case 2 : You use UI_Game_Interface_v2_Lightweight_LCD gameobject on your scene

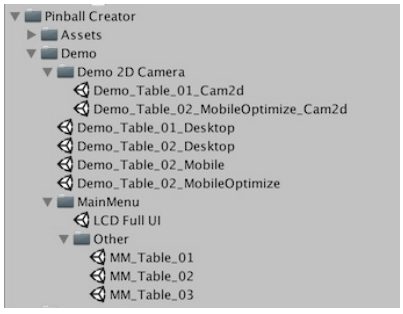
1 Delete UI_Game_Interface_v2_Lightweight_LCD gameobject on your scene

2 Drag and drop the prefab UI_Game_Interface_v2_Lightweight_LCD _WithSaveName on the root o f the scene
(Pinball Creator->Assets->Prefabs->UI->)

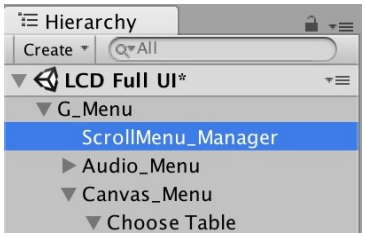


Leaderboard : Create default leaderboard for a table

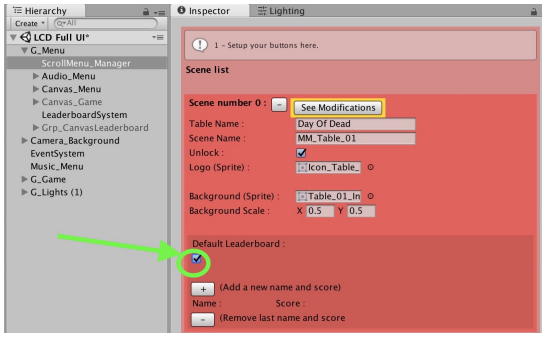
1 Open scene LCD Full UI (Project Tab → Pinball Creator → Demo → MainMenu → LCD Full UI)



2 On the Hierarchy select the gameObject ScrollMenu_Manager



3 On the inspector check the box default leaderboard for the first table



4 Press button + to add a new name and score.
(by default name is John and scrore is a a random value)

Press button - to remove the last name and score.

You could create a default leaderboard for each table .

IMPORTANT : Leaderboard is generate the first time the player launch the application.

You need to init the playerPrefs if you want to generate a new default leaderboard
Menu : Pinball Creator → Init PlayerPrefs

Leaderboard : (script) Access score and name PlayerPrefs

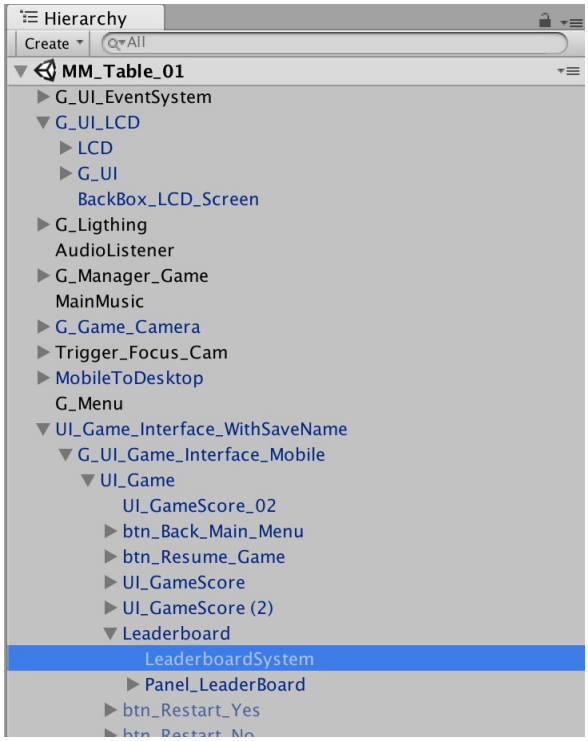
When player is game over you could

Access to the player score with :

PlayerPrefs.GetInt(‘’CurrentScore’’) ;
Or
call PlayerScore() from the script LeaderboardSystem.cs that could find on gameObject LeaderboardSystem
(this function return a integer)

Access to the player name with :

PlayerName() from the script LeaderboardSystem.cs that could find on gameObject LeaderboardSystem
(this function return a string)

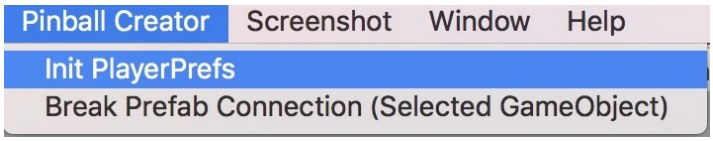



```
string PlayerName(){                        // return the player name
    return txt_PlayerName.text;
}

int PlayerScore(){                          // return the player score
    return PlayerPrefs.GetInt ("CurrentScore");
}
```

Tips : Init PlayerPrefs

If you want to init your PlayerPrefs go to

Pinball Creator → Init PlayerPrefs



Other

Sound

Script Collision_Sound :

Project -> Assets -> Script -> Ball -> Collision_Sound
Add this script to an object if you want to simulate impact between this object and a ball.
If needed, add a AudioSource.

Sound folder :

Project -> Assets -> Sounds
Info : , real“ mean that we have record this sound on a real pinball. Others are creations.

Physics Materials :

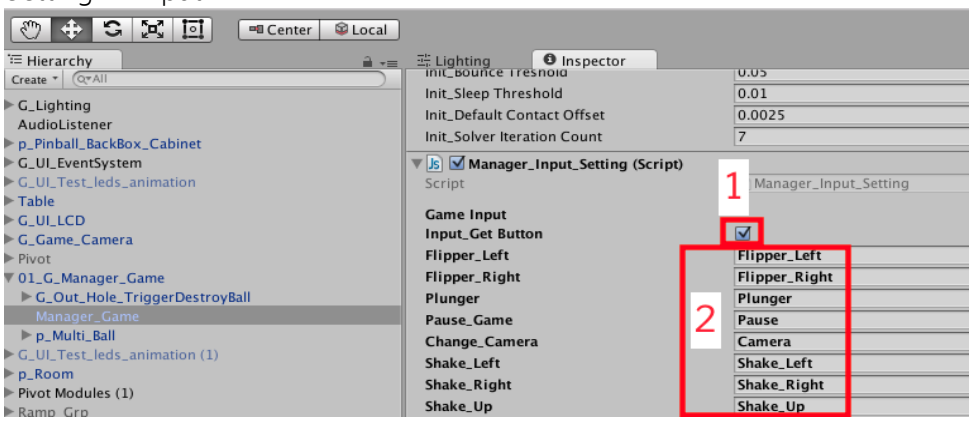
Project -> Assets -> PhysMaterials
You could change the physics material if want more bounce or less bounce.

Pinball_Ball : Use only with ball.
Pinball_Flipper : Use only with flippers.
Pinball_Low_Bounce : Low bounce and low friction.
Pinball_Mid_Bounce : Mid bounce and low friction.
Pinball_Pinball_Plateau : No friction, no bounciness.

Gamepad

This part explain how to configure a XBox 360 gamepad :

Step 1 : Select gameObject Manager_Game on Hierarchy. Check box Input_GetButton on Manager_input_Setting script (pic 1).
Then choose an input for each variable (pic 2). These inputs refer to Inputs created on Edit -> Project Setting -> Input.



Step 2 : Create Inputs.

Change Axes -> Size to add Inputs (pic 1).

Configure each Inputs (pic 2).

Horizontal : Type : Joystick Axis | Axis : 6th axis

Shake_Left : Positive Button : f | Alt Positive Button : Joystick button 4 | Type : Key or Mouse Button | Axis : X axis

Shake_Right : Positive Button : h| Alt Positive Button : Joystick button 5 | Type : Key or Mouse Button | Axis : X axis

Shake_Up : Positive Button : g | Alt Positive Button : Joystick button 2 | Type : Key or Mouse Button | Axis : X axis

Pause : Positive Button : p | Alt Positive Button : Joystick button 7 | Type : Key or Mouse Button | Axis : X axis

Camera : Positive Button : c | Alt Positive Button : Joystick button 3 | Type : Key or Mouse Button | Axis : X axis

Plunger : Positive Button : return | Alt Positive Button : Joystick button 0 | Type : Key or Mouse Button | Axis : X axis

Flipper_Left : Positive Button : left shift | Type : Key or Mouse Button | Axis : X axis

Flipper_Left : Type : Joystick Axis | Axis : 9th axis

Flipper_Right : Positive Button : right shift | Type : Key or Mouse Button | Axis : X axis

Flipper_Right : Type : Joystick Axis | Axis : 10th axis