Pinball Creator

CREATE YOUR PINBALL

Asset documentation Part 2.

Table of contents :

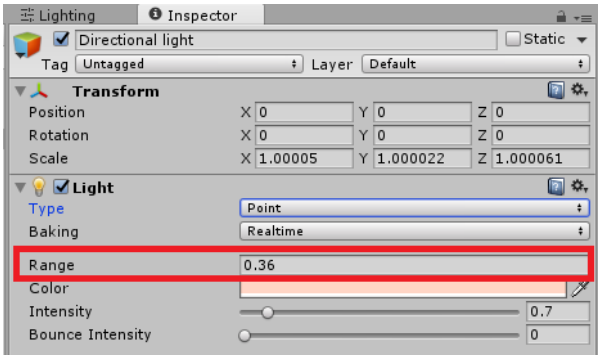
Features :

Other :

Optimization :

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.



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**.

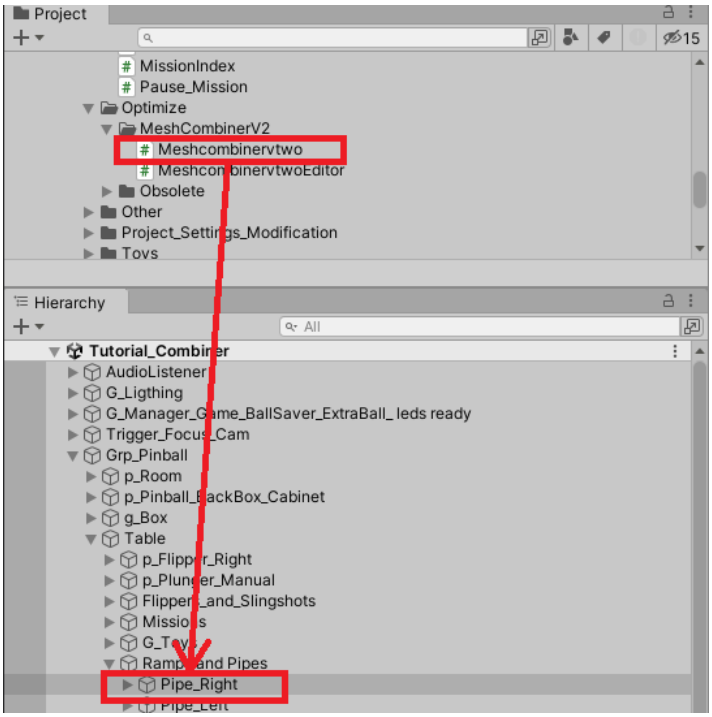
How to use :

- Open Tuto scene Tutorial_Combiner

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

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

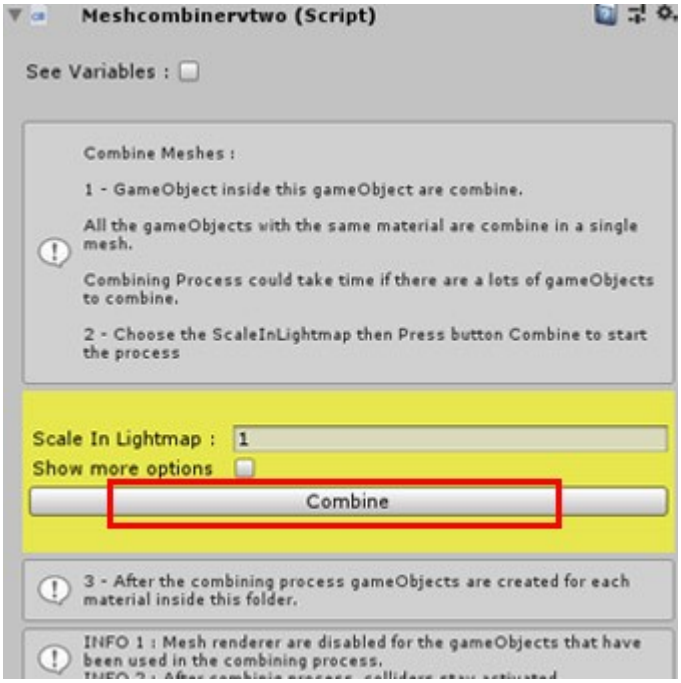
(Grp_Pinball->Table->Ramps and Pipes -> Pipe_Right)



- Select Pipe_Right in hierarchy tab.

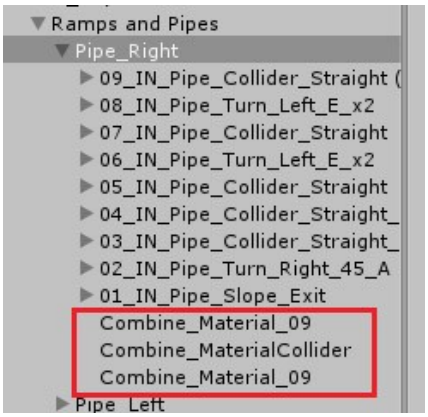
- In the Inspector press Reset tag button.

- On the Inspector press Combine.

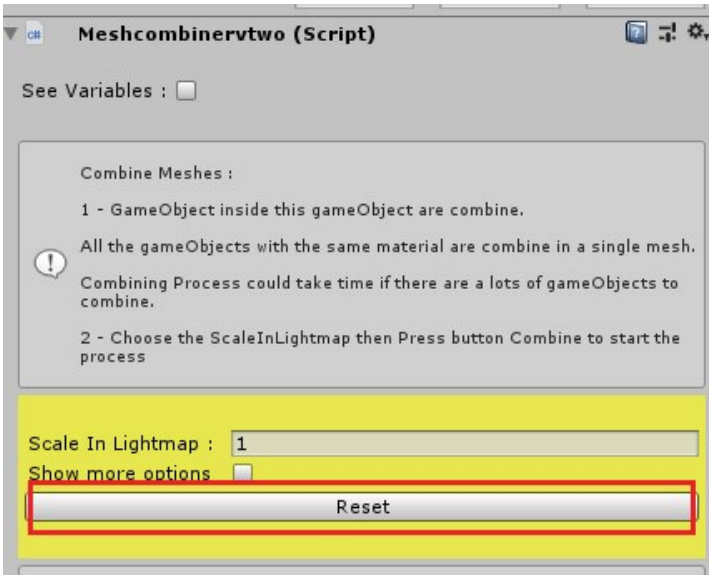


After the process new Combine gameObjects are created inside Objects_Grp group.

All other objects in the group are hide.

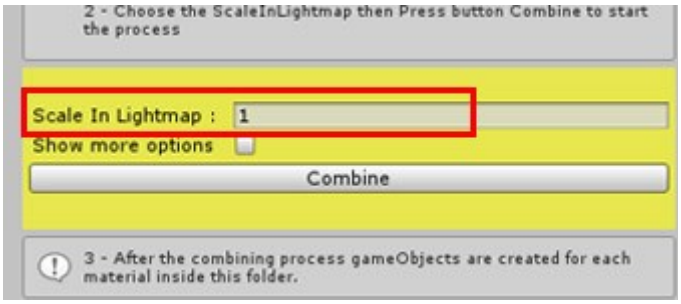


You could revers the process by pressing the button Reset or Ctrl+Z



Troubleshooting :
- If you have a lot of objects (or large objects) in group we recommand 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

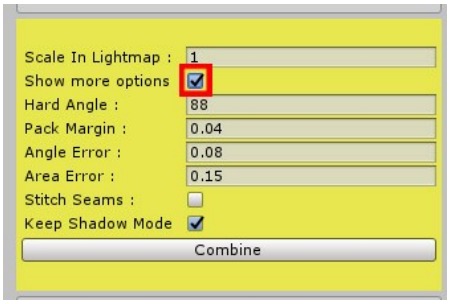
- Add to the first group objects with sharp edges
- Set Scale in lightmap value to 1

- 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

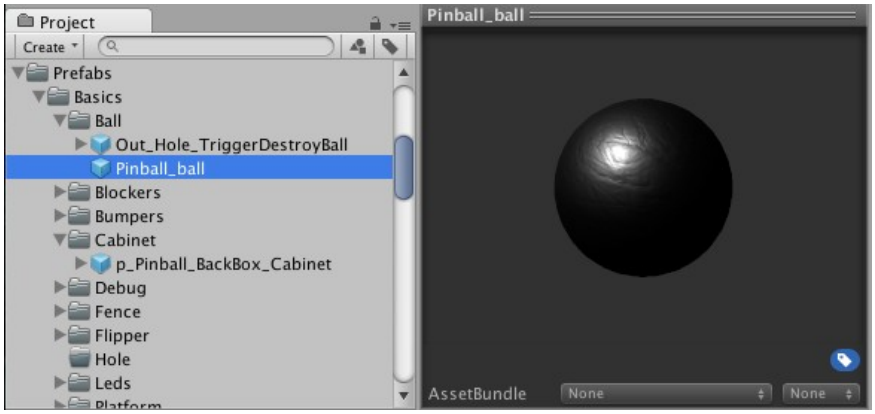


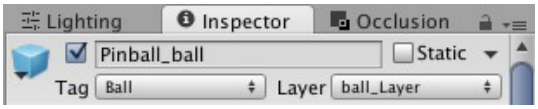
Stitch seams improves the quality of lightmaps

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

Tables Mechanics : (prefabs)

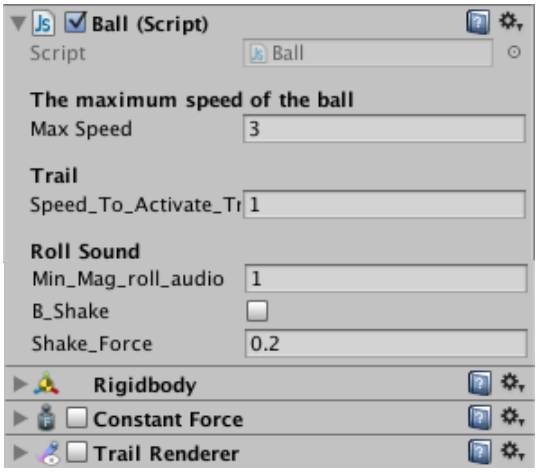
Ball :
Project -> Assets -> Prefabs -> Basics -> Ball -> Pinball_ball





Tag : Ball

Layer : ball_Layer



Find Ball.cs on Pinball_ball gameObject

Max Speed : Limit ball speed

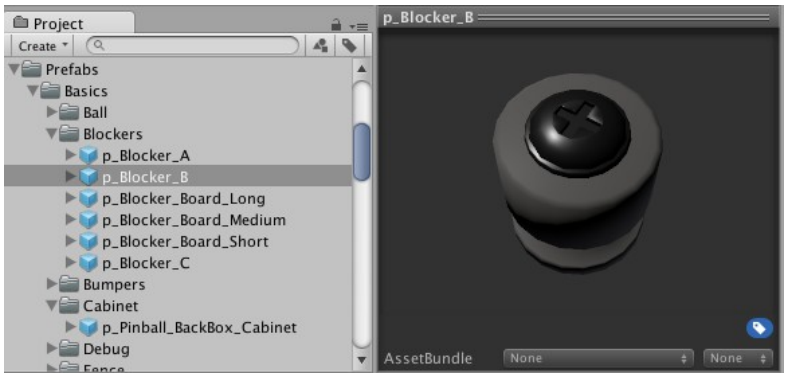
Speed_To_Activate_Trail : Minimum speed to activate the trail

Min_Mag_roll_audio : Minimum speed to activate the rolling sound

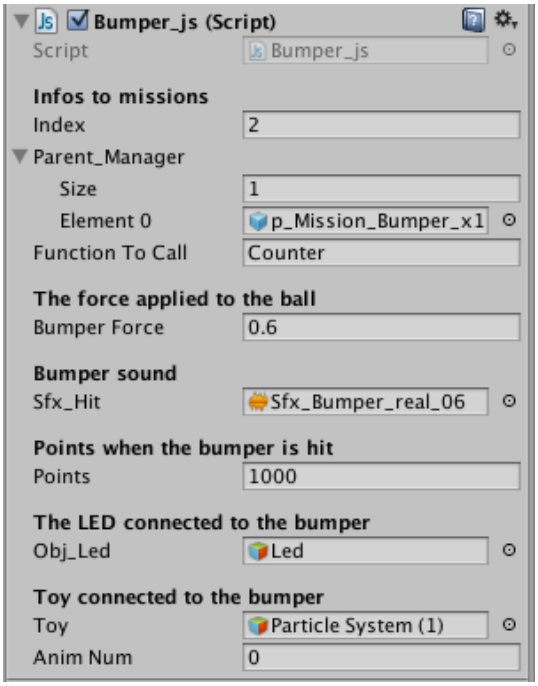
b_shake : Activate shake force.

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.cs 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

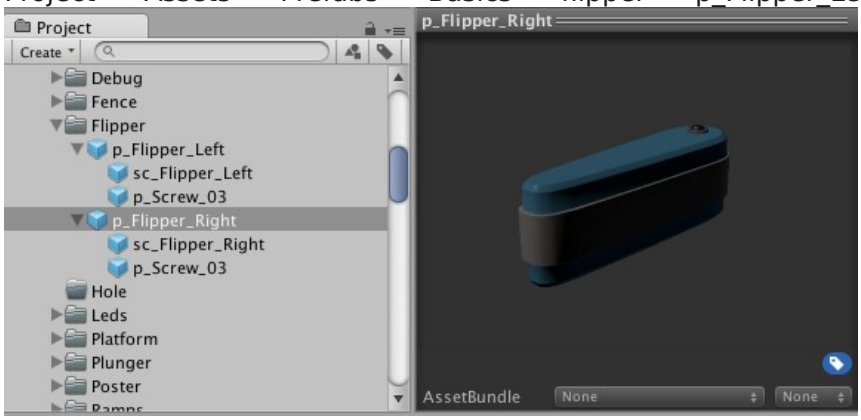
Obj_Led : Connect a led. Led Switch ON when ball hit the bumper

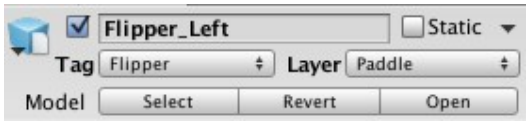
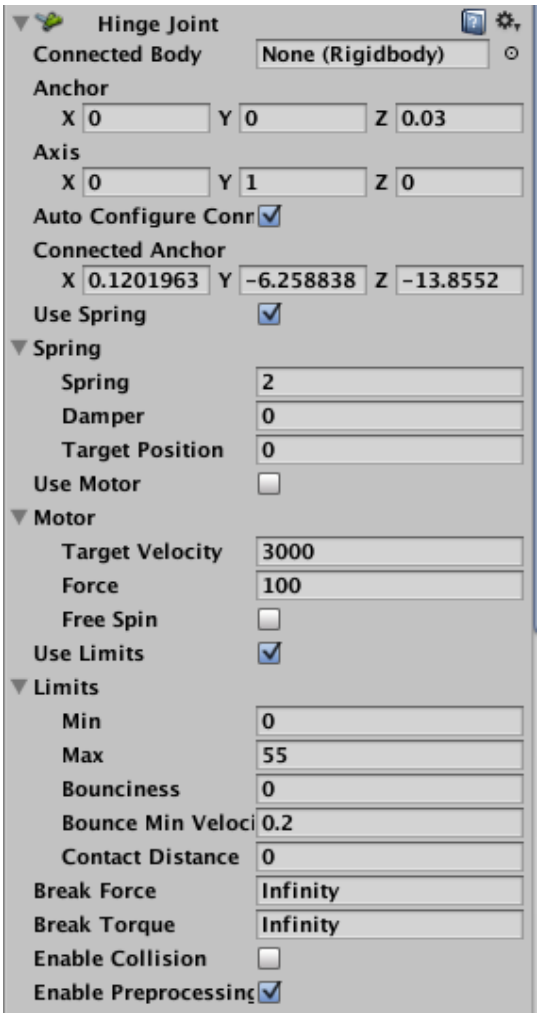
Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.cs attached to it.(more about toy)

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.cs to activate or deactivate the plunger. **Layer :** Paddle. Ignore collision between all layers except “ball_Layer” (Flippers.cs)
	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.cs

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.

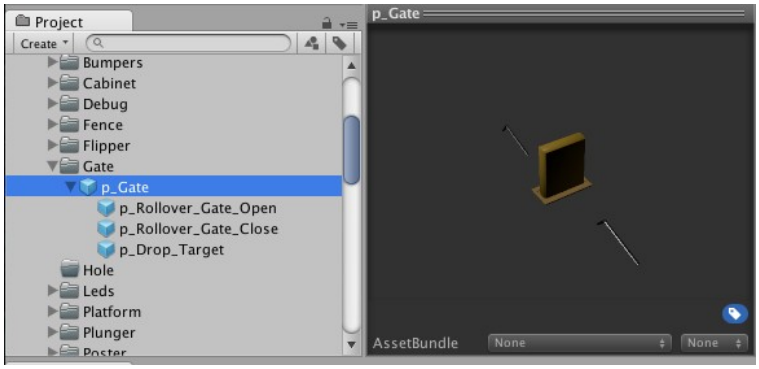
volume_Max : Maximum volume

b_Flipper : TRUE

Flipper : Play this sound

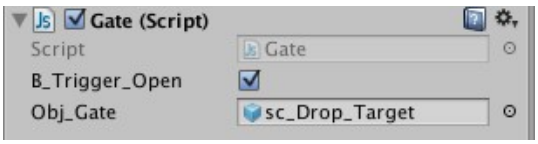
Gate :

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).



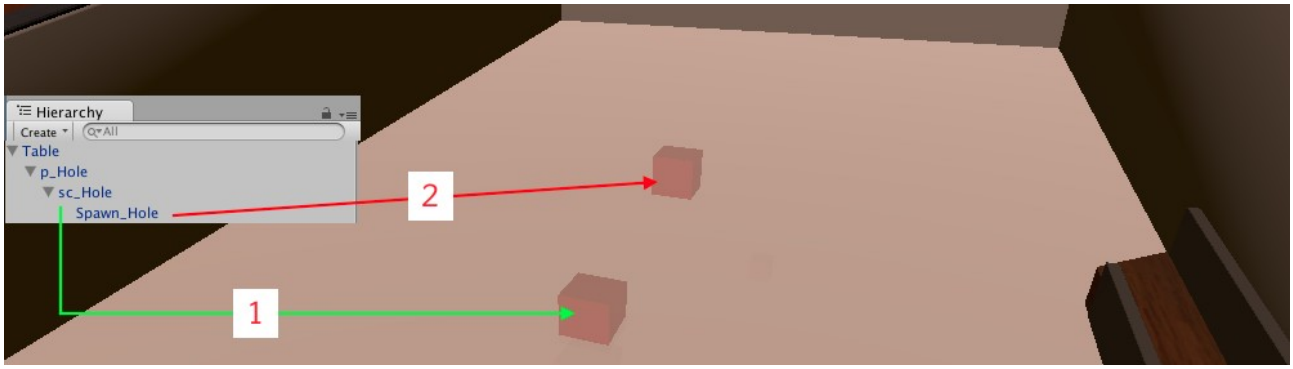
Find Gate.cs on p_rollover_Gate_Open -> sc_Roll_Over_Metal and _rollover_Gate_Close -> sc_Roll_Over_Metal

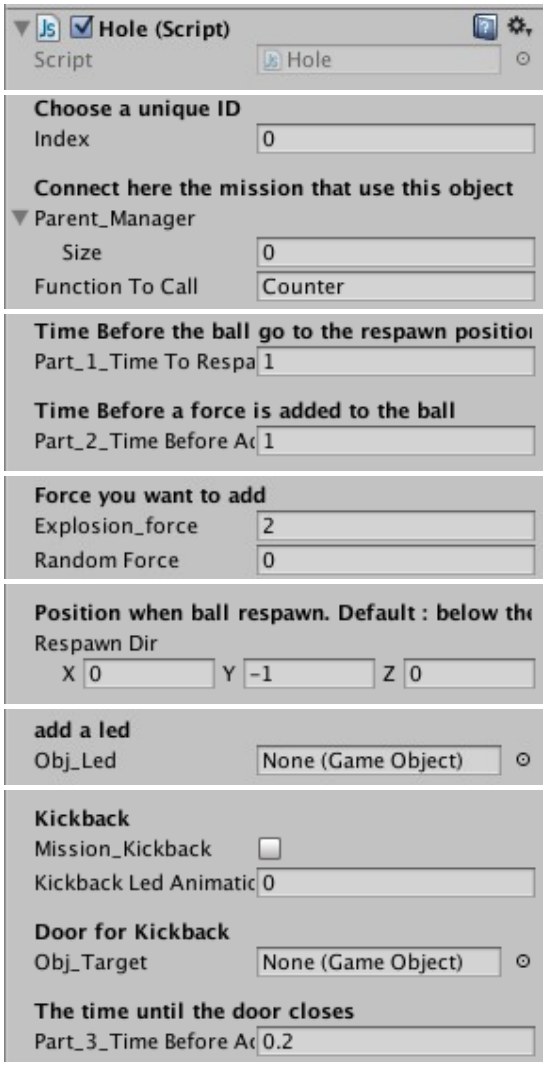
B_Trigger_Open : The trigger you want to open the target must be True, the other trigger must be False

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.cs 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)

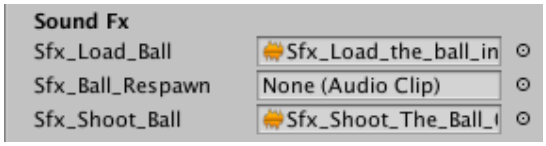
If you want to test kickback use prefab p_Kickback:
Project -> Assets -> Prefabs -> Mechanics -> Gate -> p_Kickback
(more about Kickback)

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

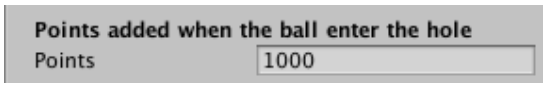
Part_2_TimeBeforeActivateObjDoor : Timer



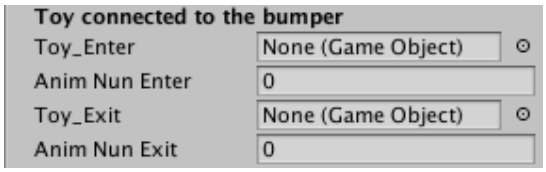
Sfx_Load_Ball : Play a sound when ball is capture

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

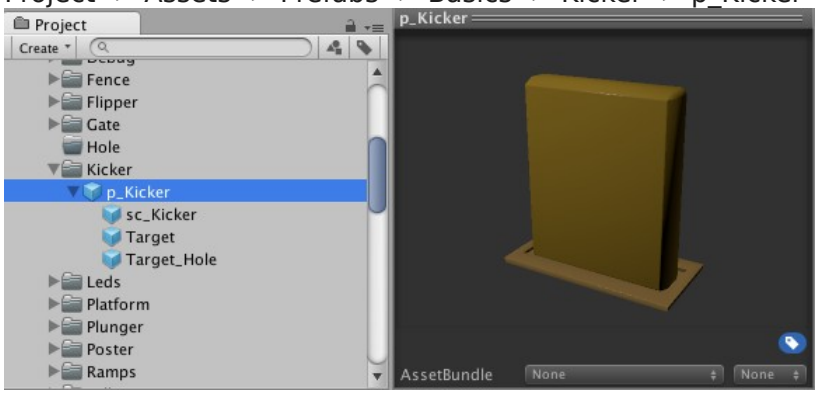
AnimNumEnter : choose animation number.

Toy_Exit : Play toy animation when ball are ejected

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

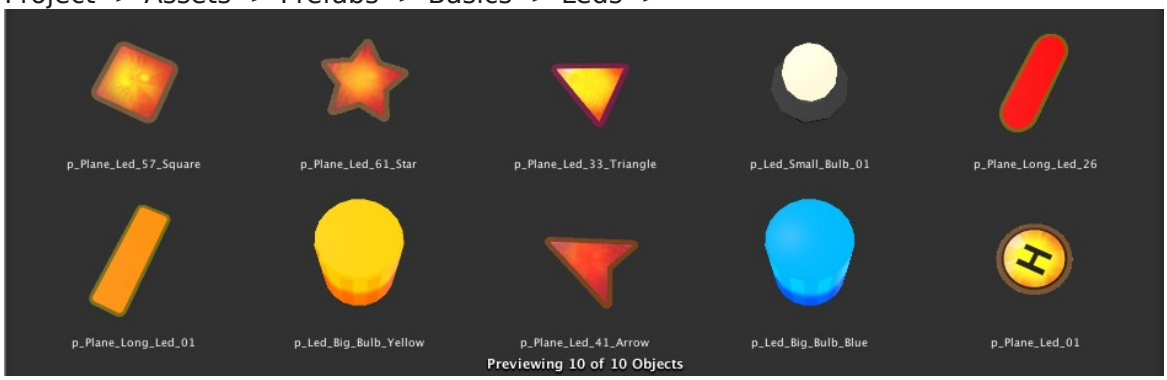
Kicker :

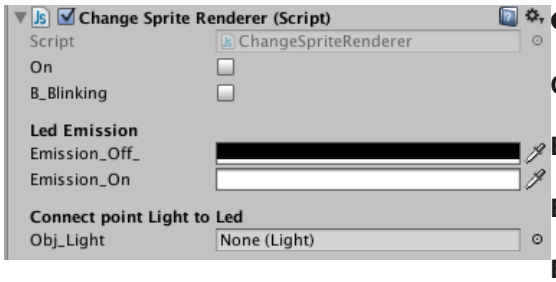
Project -> Assets -> Prefabs -> Basics -> Kicker -> p_Kicker



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.cs

On : if True Led is switch ON

B_Blinking : if True led blinks.

Emission_Off_ : Choose a color

Emission_On : Choose a color

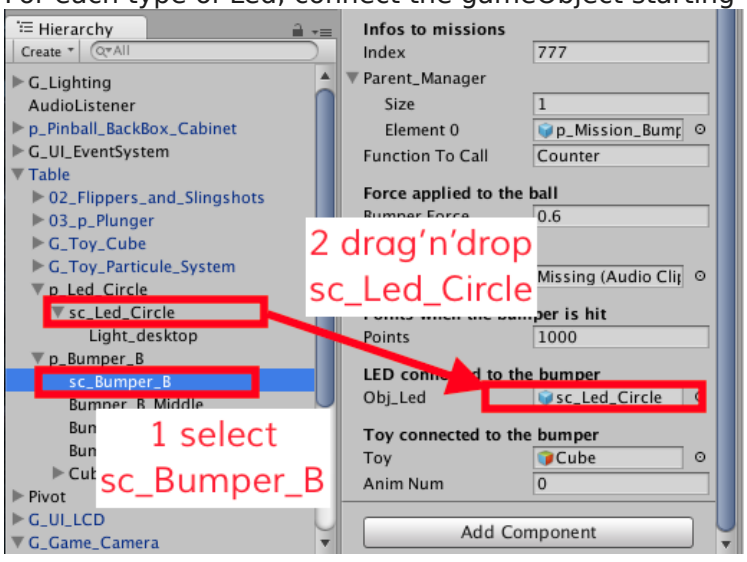
Obj_Light : Connect a point light.

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.cs (Inspector) (pic 2).

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

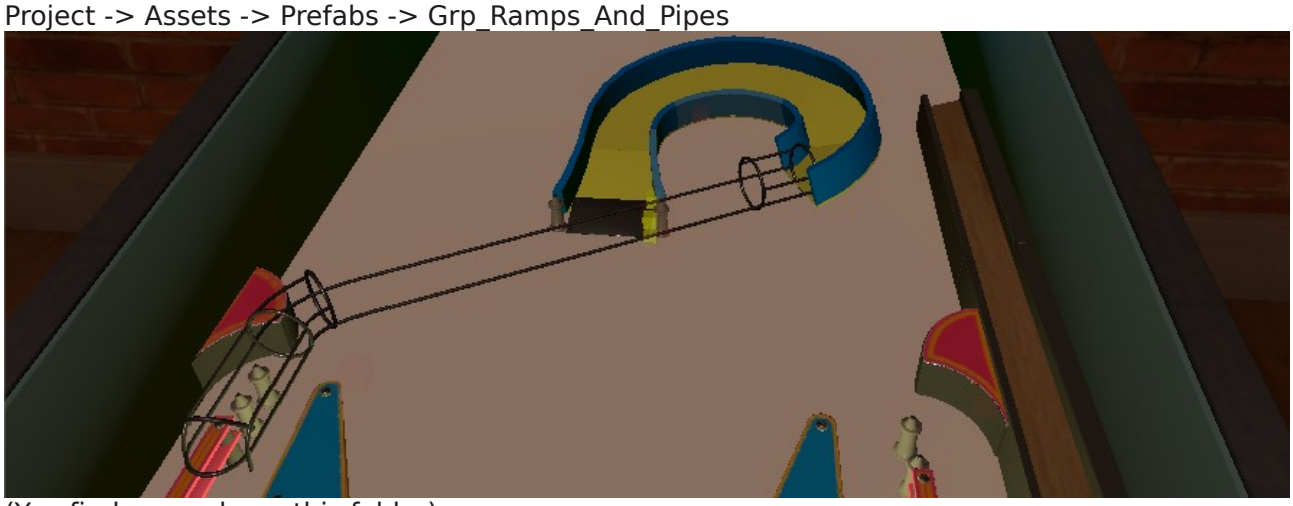


Led : Where to connect Led

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

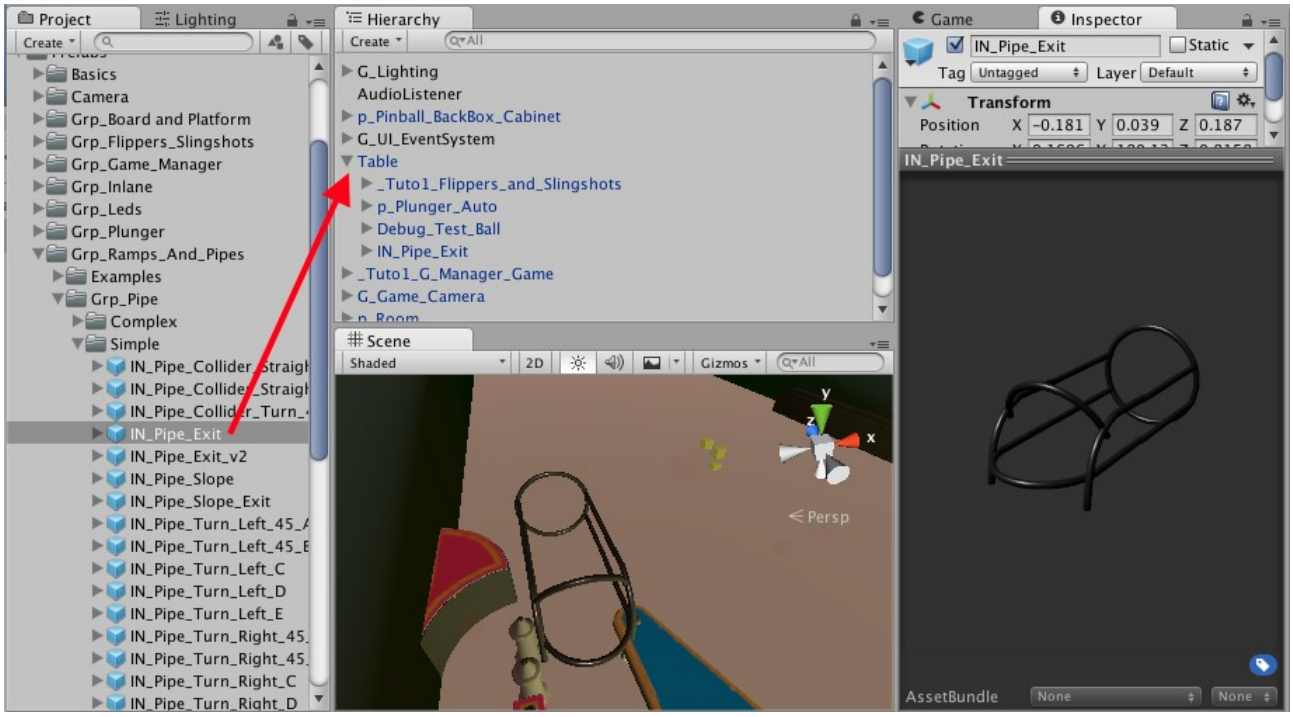


(You find example on this folder).

How to create a pipe :

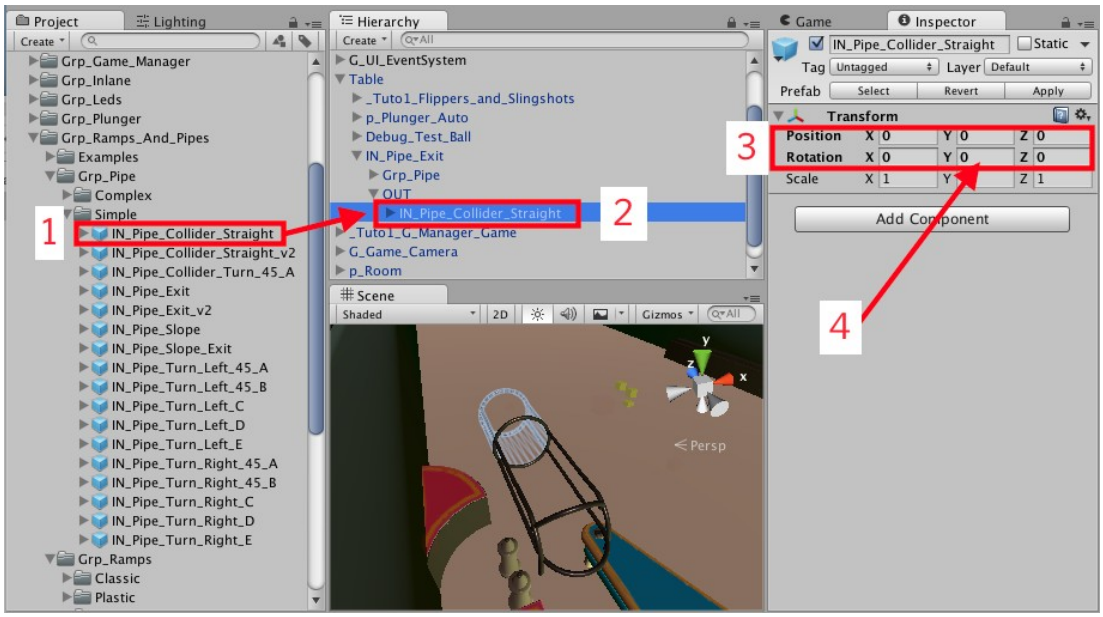
Example : Create this pipe

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)

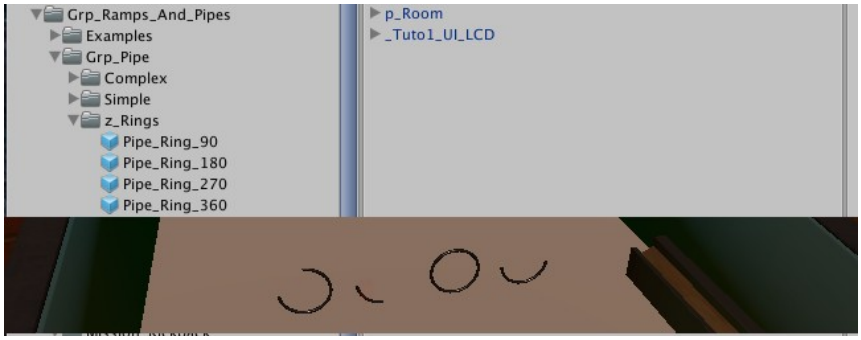


Step 2 : Drag'n'drop IN_Pipe_Collider_Straight inside IN_Pipe_Exit -> OUT on Hierarchy.(Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes -> Grp_Pipe -> Simple -> IN_Pipe_Collider_Straight). (see next page : pic 1 and 2)
Reset IN_Pipe_Collider_Straight transform. Position (0,0,0) and rotation (0,0,0)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

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



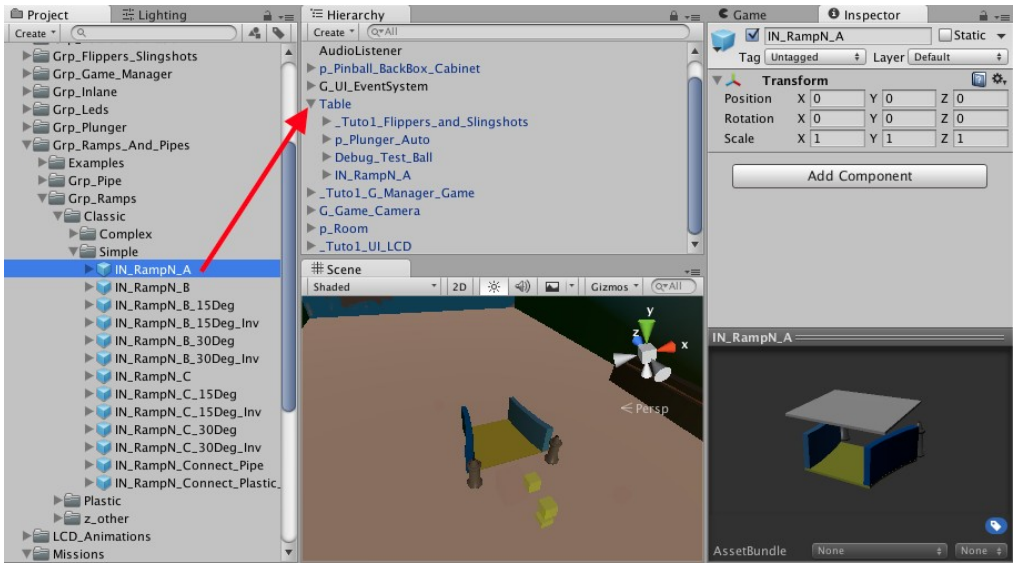
INFO: Inside folder z_ring you could find different type of rings to customize your pipe.



How to create a ramp :

Example : Create this ramp

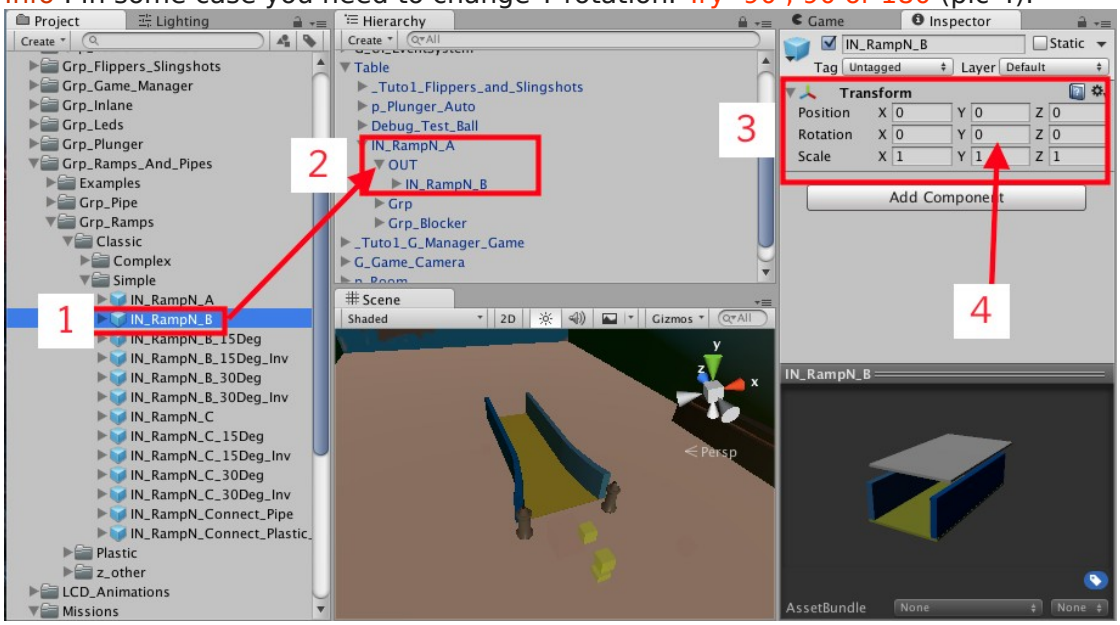
Step 1 : Drag'n'drop IN_RampN_A inside Table on Hierarchy.(Project -> Assets -> Prefabs -> Grp_Ramps_And_Pipes -> Grp_Ramps -> -> Classic -> Simple -> IN_RampN_A). (see next page pic 1)



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).

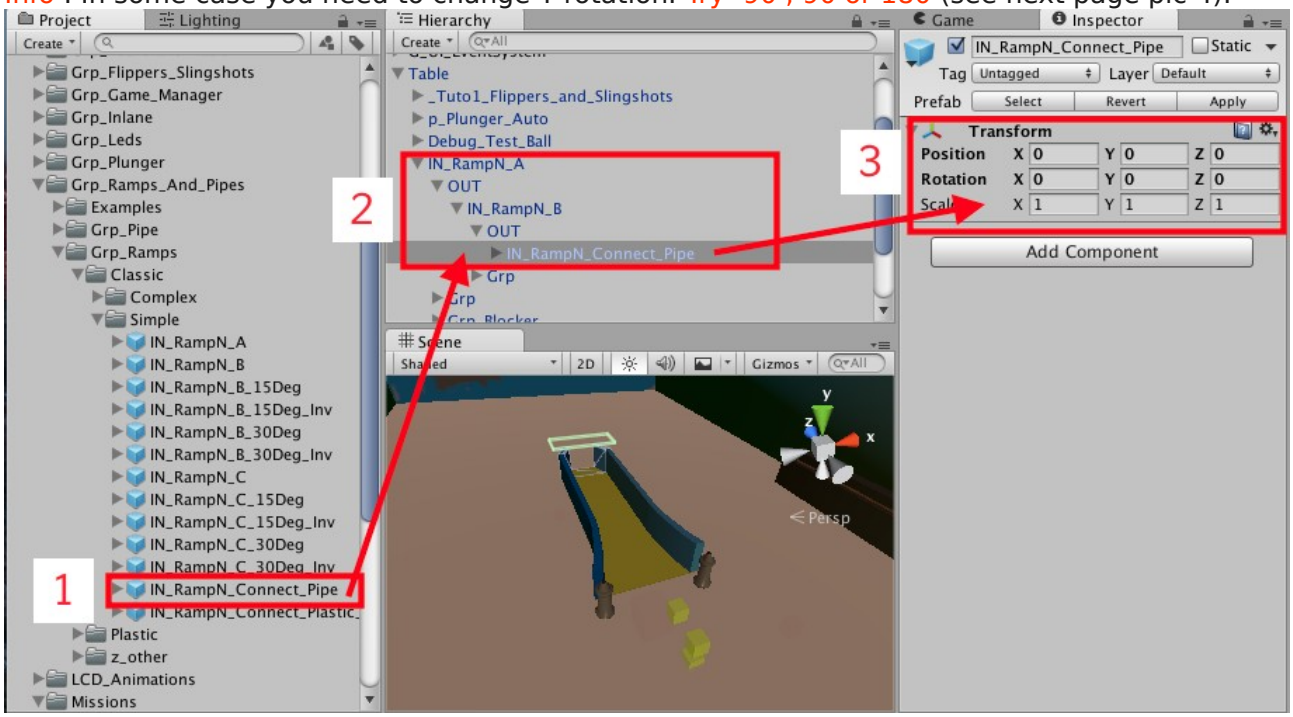
Info : In some case you need to change Y rotation. Try -90 , 90 or 180 (pic 4).



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).



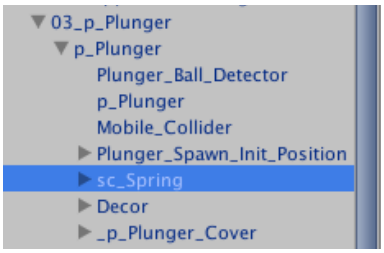
Plunger :
Project -> Assets -> Prefabs -> Basics -> Plunger -> p_Plunger_Auto, p_Plunger_Manual ...

How it work.
Press return to pull the spring.

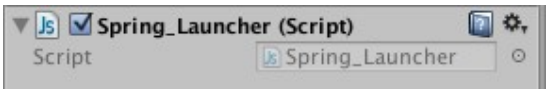
Change Inputs on gameObject Manager_Game on Hierarchy (script manager_Input_Setting.cs).

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

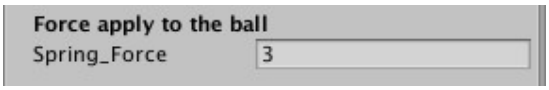


Tag : Plunger . This tag is used by Manager_Game on Hierarchy.

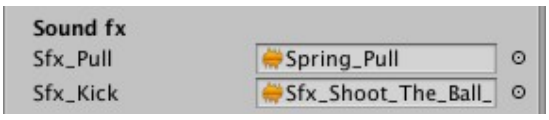


Find spring_Launcher.cs on p_Plunger_Manual -> p_Plunger -> sc_Spring

Auto_Mode : True (plunger manual)
False (plunger auto)

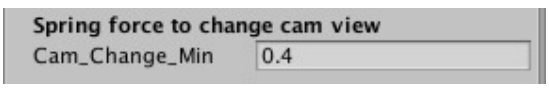


Spring_Force : Force apply to ball.



Sfx_Pull : Sound when player pull the plunger

Sfx_Kick : Sound when ball is ejected

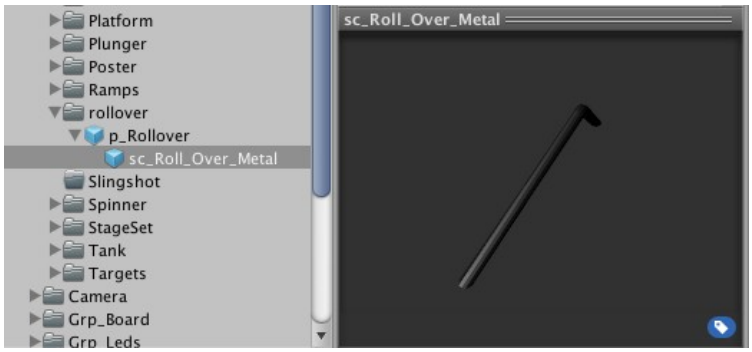


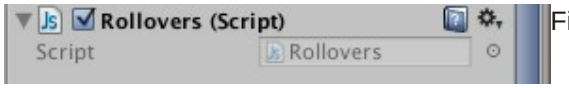
Cam_Change_Min : If Spring force > Cam_Change_Min the view change.

Timer : Time to prevent bug when camera switch between playfield camera and plunger camera.

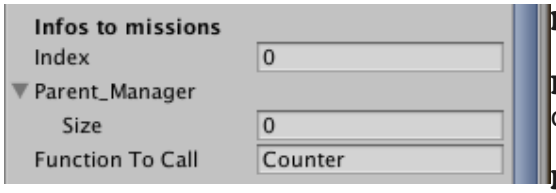
Rollover :

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





Find Rollovers.cs on sc_Rollover_Metal gameObject

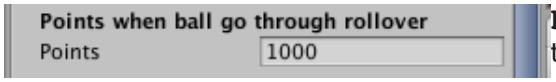


Index : Choose a unique ID

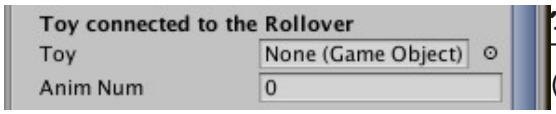
Parent_Manager : Connect missions that used this object. You could connect more than one mission.

Function To Call : Call a function

Sfx_Hit : Sound when the ball hit the bumper



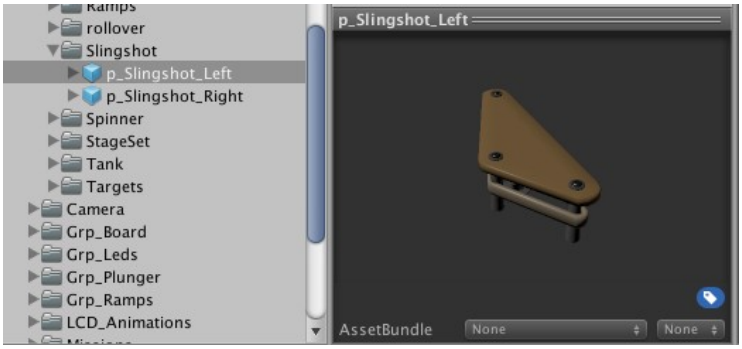
Points : Points added to the score when ball go through the rollover

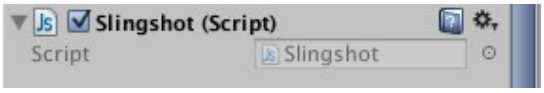


Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.cs attached to it. (more about toy)

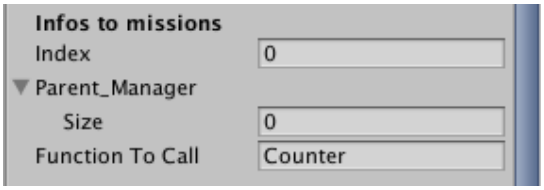
AnimNum : Choose the animation played by the toy.

Slingshot : and kicker
Project -> Assets -> Prefabs -> Basics -> Slingshot -> p_Slingshot_Left or p_Slingshot_Right





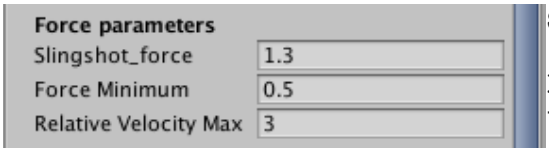
Find Slingshot.cs on sc_Slingshot gameObject



Index : Choose a unique ID

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

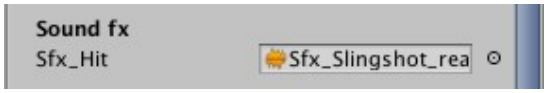
Function To Call : Call a function



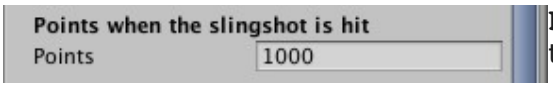
Slingshot_force : Force apply to ball

ForceMinimum : Minimum impact velocity to apply force to ball

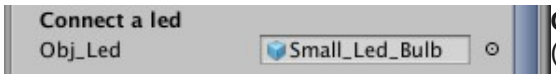
RelativeVelocityMax : Maximum force apply to ball



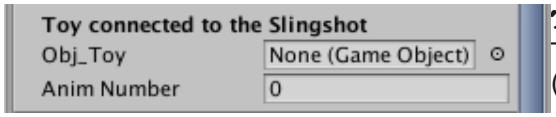
Sound_fx : Sound when the ball hit the slingshot



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



Obj_Light : Connect a led.
(more about led)

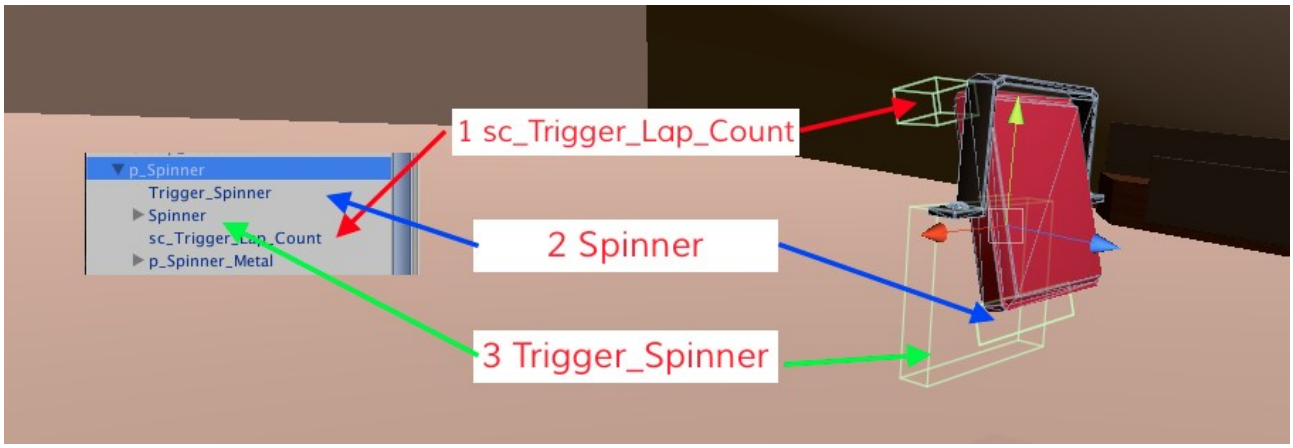


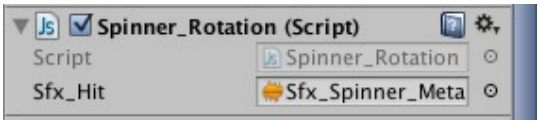
Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.cs attached to it.
(more about toy)

AnimNum : Choose the animation played by the toy.

Spinner :

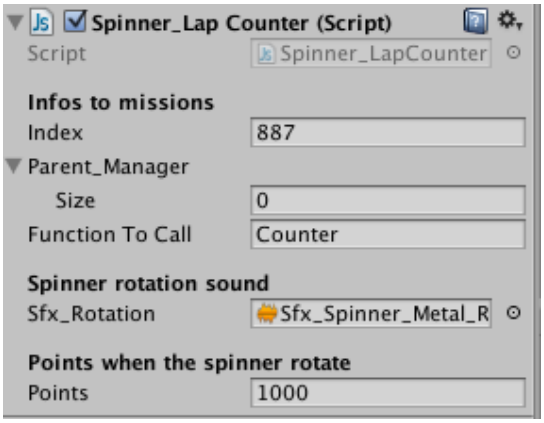
Project -> Assets -> Prefabs -> Basics -> Spinner -> p_Spinner





2 : Find Spinner_Rotation.cs on sc_Spinner gameObject

Sfx_Hit : play a sound.



1 : Find Spinner_Lap_Counter.cs 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.

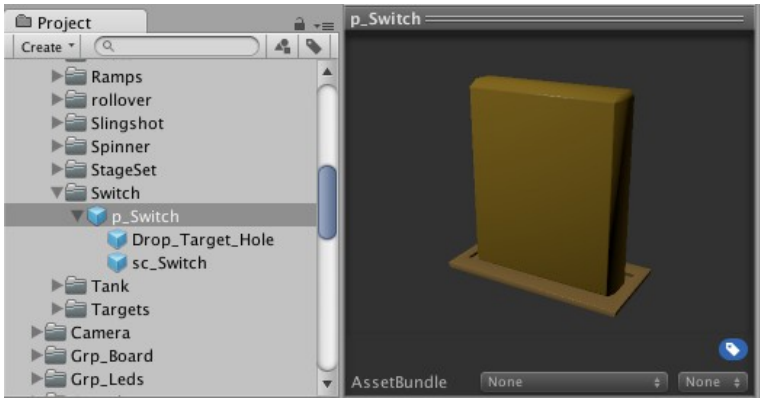
Function To Call : Call a function

Sfx_Rotation : play a sound.

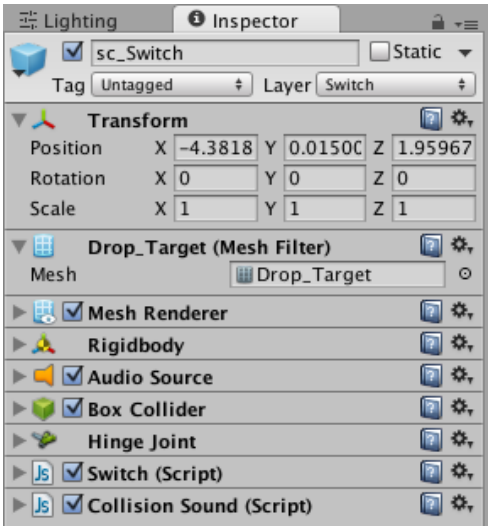
Points : Points earn when spinner rotate

Switch :

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

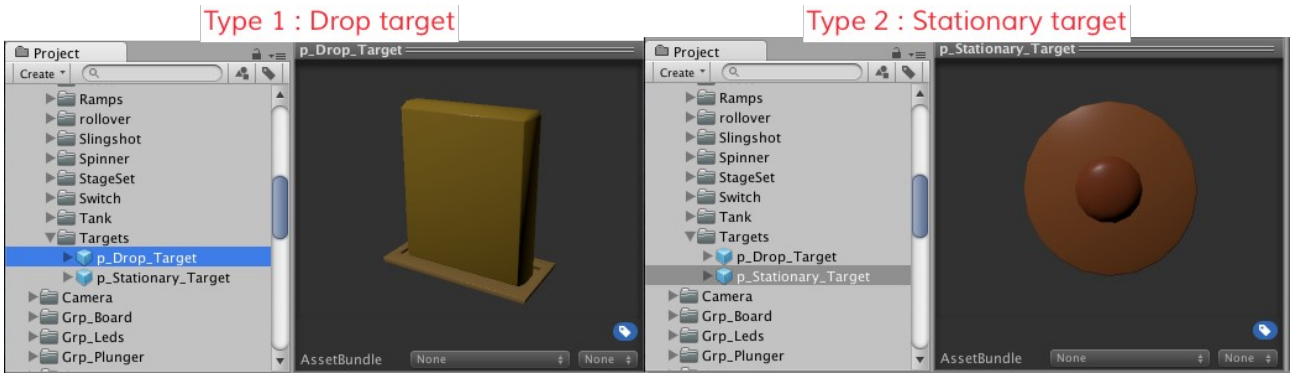


How it works.
Switch is similar to gate but it use a spring.
Ball is restricted to one direction.

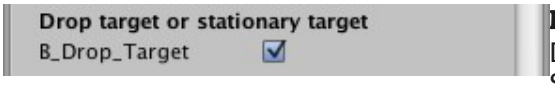


Find Switch.cs, Hinge joint and CollisionSound.cs on sc_Switch gameObject

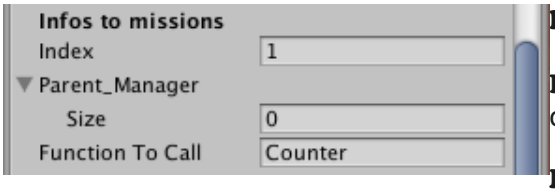
Targets : drop target, stationary target and vari-target
Project -> Assets -> Prefabs -> Basics -> Targets -> p_Drop_Target and p_Stationary_Target



Find Target.cs on sc_Drop_Target and sc_Stationary_Target gameObject



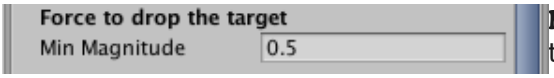
B_Drop_Target :
Drop target and vari-target = True
Stationary target = False



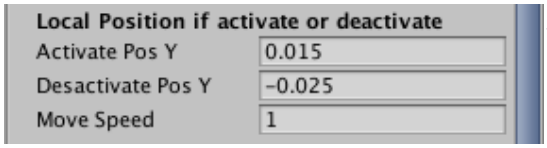
Index : Choose a unique ID

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

Function To Call : Call a function



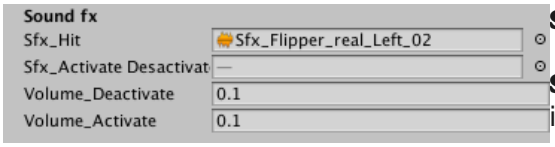
MinMagnitude : minimum magnitude when ball hit target



ActivatePosY : local Position when target is activated

DeactivatePosY : local position when target is deactivated

MoveSpeed : speed to reach the new target position



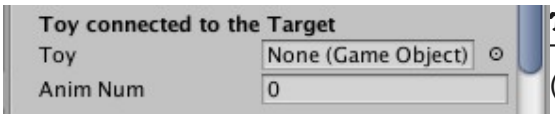
Sfx_Hit : play a sound when ball hit target.

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.

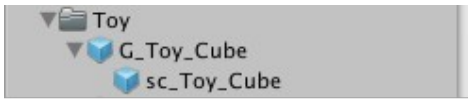
Points : Points earn when ball hits target.

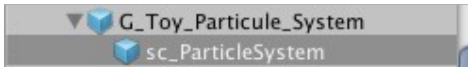


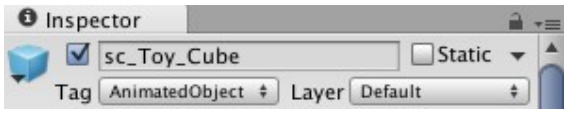
Toy : Connect a toy or a particle system to this object. This toy must have a script Toys.cs attached to it. (more about toy)

Toys :

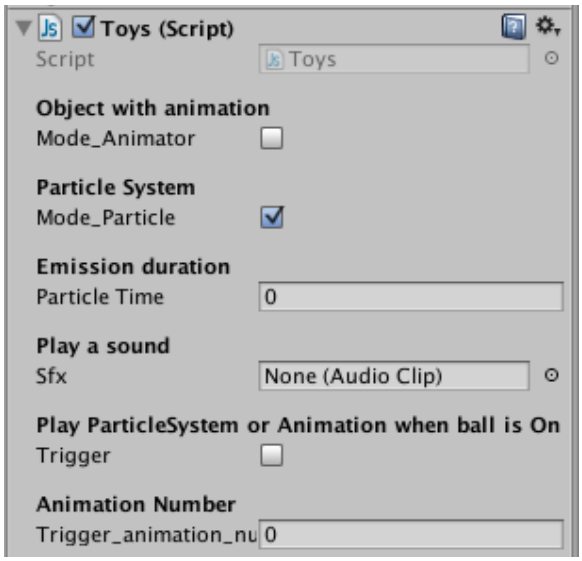
There are two types of Toy. Toy with animation and Toy with particle system. Look at example prefabs G_Toy_Cube and G_Toy_particle_System (Project -> Assets -> Prefabs -> Toy ->)







Tag : AnimatedObject



Find script Toys.cs on gameObject sc_Toy_Cube or sc_ParticleSystem

Mode_Animator : True if toy use animation.

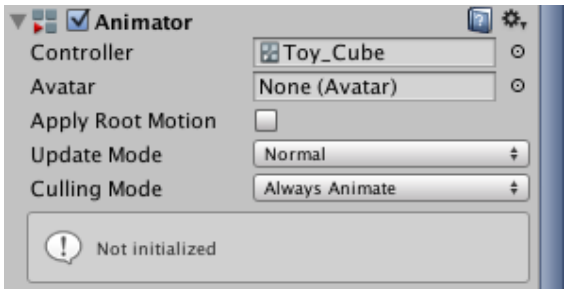
Mode_Particle : True if toy use particle system.

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.

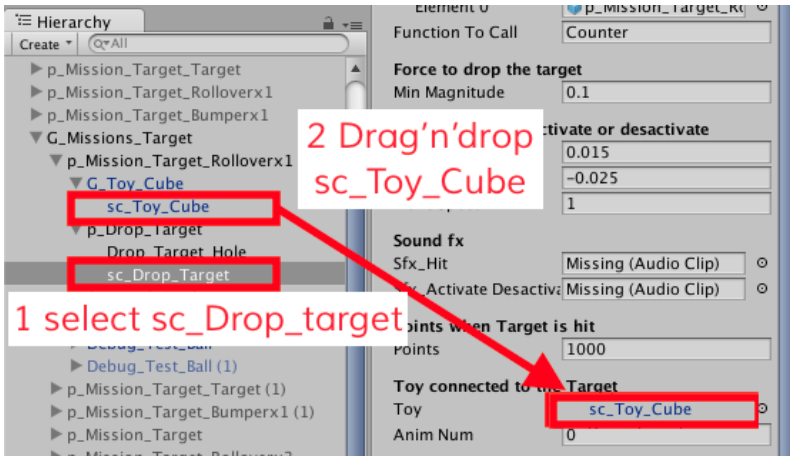
You can connect Toy in different ways.

Example : add Toy to Target.

Case 1 : Add Toy with animation.

Step 1 : Select sc_Drop_Target on hierarchy (pic 1).

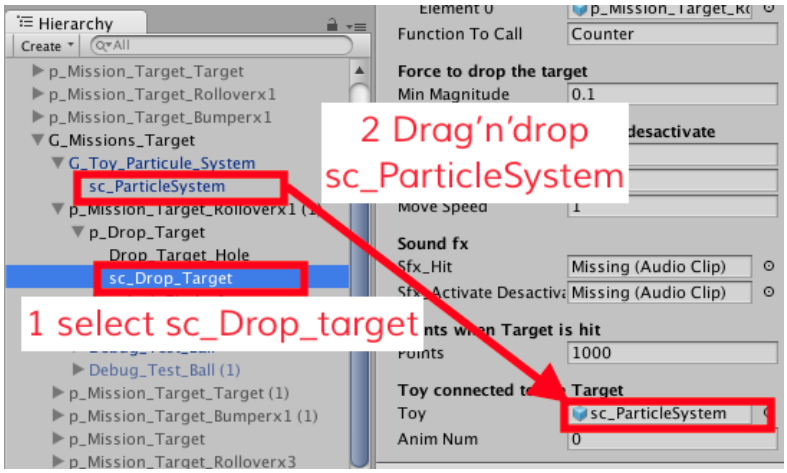
Step 2 : Drag'n'drop sc_Toy_Cube inside variable Toy on script Target.cs (Inspector) (pic 2).



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.cs (Inspector) (pic 2).

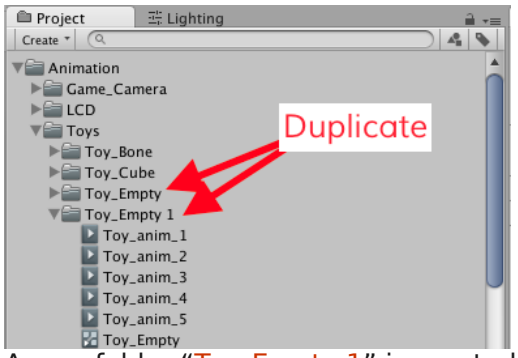


2° Toys : Where to connect Toy

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

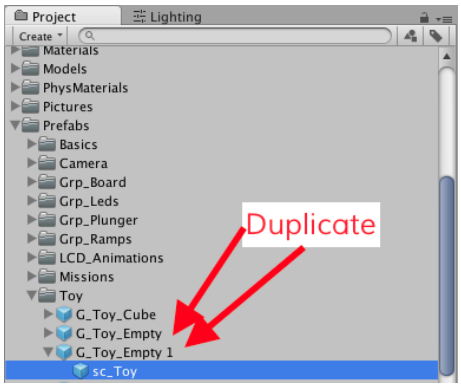
3° Toys : Create a new Toy with animation.

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



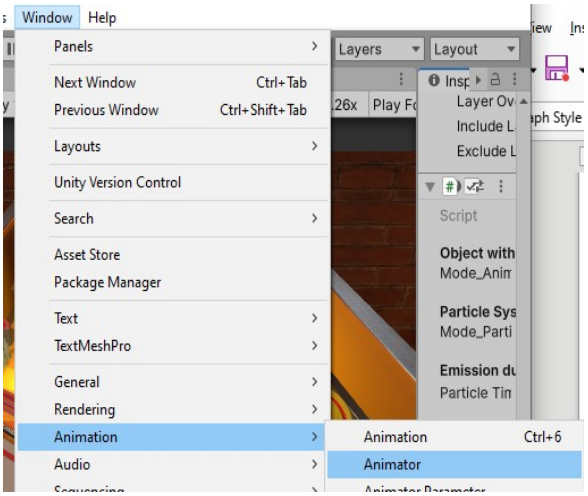
A new folder "Toy_Empty 1" is created.

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.



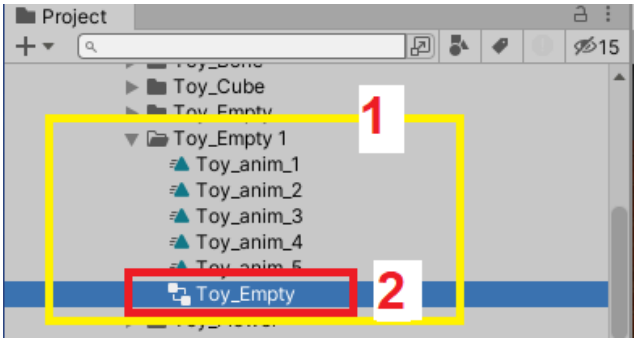
Step 3 : Configure the animator Toy

- First open Animator Window.

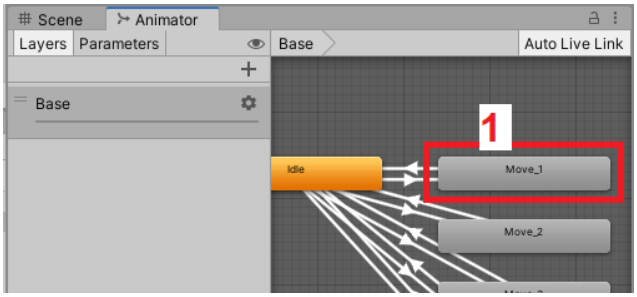


- In Project folder inside Toy_Empty 1 (pic 1) select Toy_Empty (pic 2).

(Project -> Assets -> Animation -> Toys -> Toy_Empty 1 -> Toy_Empty)

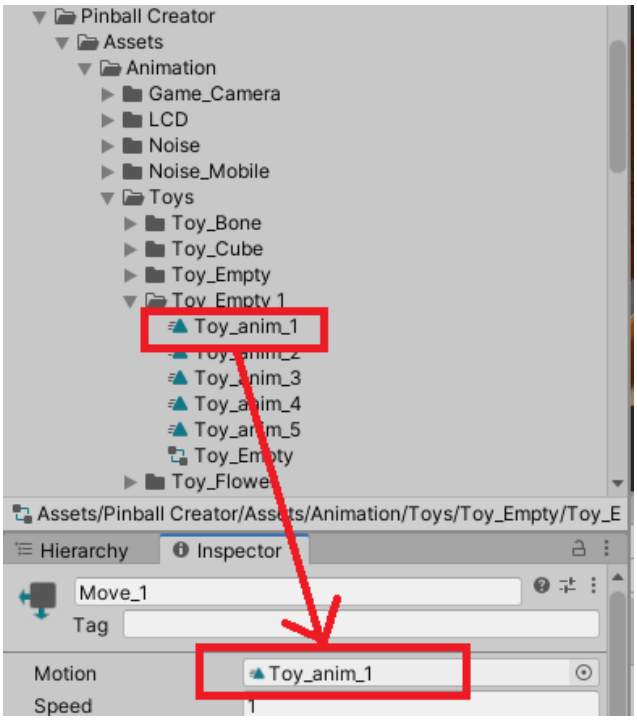


- In the animator window click on Move_1 animation state to select it (pic 1).



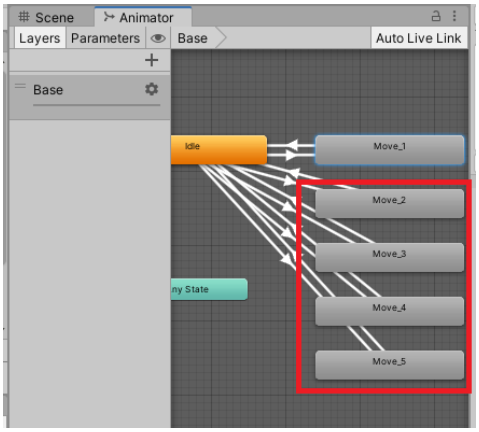
- In the Inspector:
Drag'n'drop Toy_anim_1 inside Motion.

(Project -> Assets -> Animation -> Toys -> Toy_Empty 1 -> Toy_anim_1)



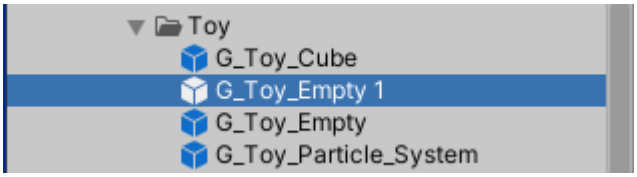
- Do the same for process with
Move_2,
Move_3,
Move_4,
Move_5

Drag'n'drop Toy_anim_2 inside Motion_2.
Drag'n'drop Toy_anim_3 inside Motion_3.
Drag'n'drop Toy_anim_4 inside Motion_4.
Drag'n'drop Toy_anim_5 inside Motion_5.

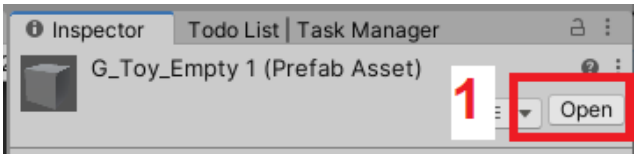


- In Project folder select G_Toy_Empty 1

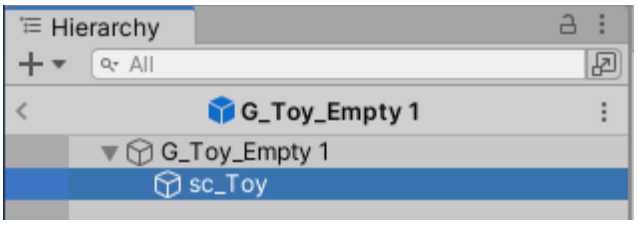
(Project -> Assets -> Prefabs -> Toy -> G_Toy_Empty 1)



- In the Inspector press Open button to open the prefab.



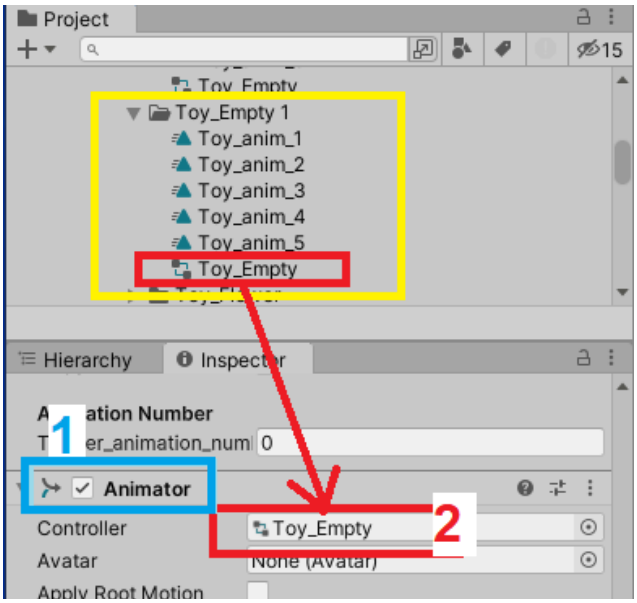
- In the Hierarchy select sc_Toy.



- In the Inspector go to Animator script (pic 1)

- From the Project tab drag'n'drop Toy_Empty inside Controller slot (pic 2).

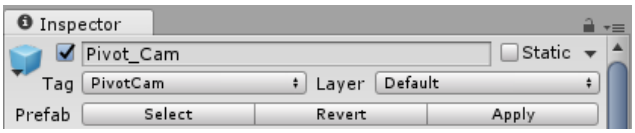
(Project -> Assets -> Animation -> Toys -> Toy_Empty 1 -> Toy_Empty)



Step 5 : Toy is ready.

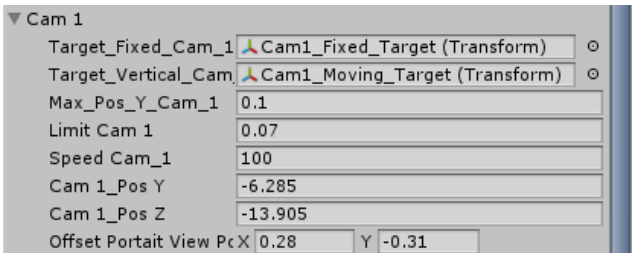
Now you could put object on sc_Toy and create your own animations for the Toy.

Features :
Camera

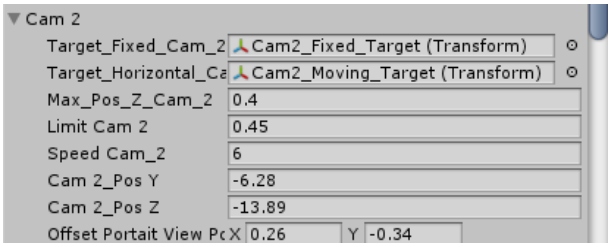


Tag : PivotCam

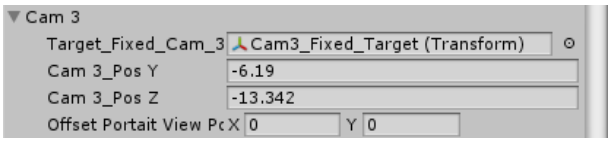
DebugCamView : true if you want to tweak your camera view.
IMPORTANT : Uncheck the box when you finish your tweaks



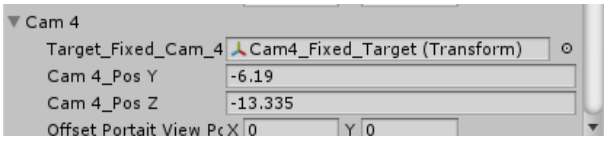
Target_Fixed_Cam_1 : Camera look at this gameObject if LimitCam1 < 0.1
Target_Vertical_Cam_1 : Camera follow this gameObject if LimitCam1 > 0.1
Max_Pos_Y_Cam_1 : Max movement for the camera
LimitCam1 : Limit in Z position to switch betweenTarget_Fixed_Cam_1 and Target_Vertical_Cam_1
SpeedCam1 : Camera 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



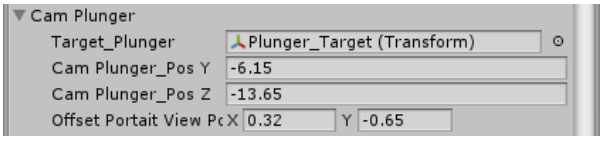
Same as Cam 1



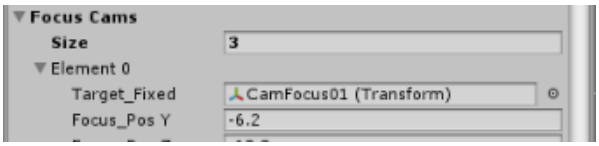
Same as Cam 1



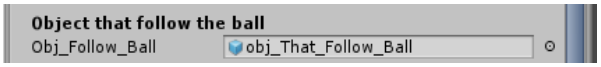
Same as Cam 1



Plunger Cam



Focus Cam



Obj_Follow_Ball : This object follow the ball

5.2 How to tweak a camera view position and rotation ?

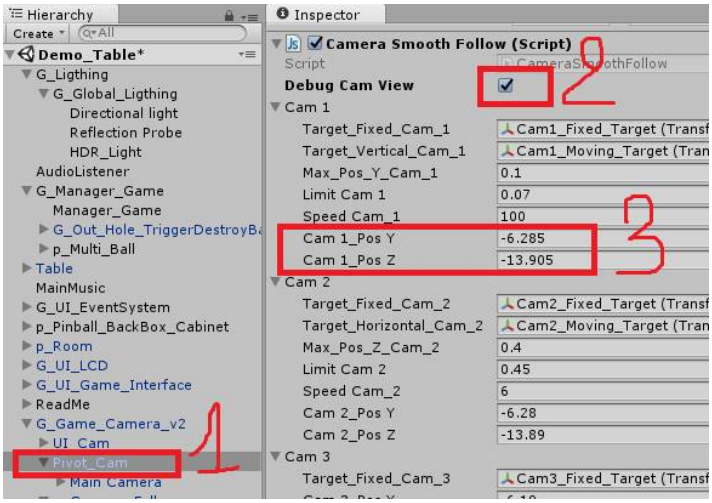
Example : Cam view 1

Change position :
-Select PivotCam on gameObject G_Game_Camera (pic1)

- Check box DebugCamView (pic 2)

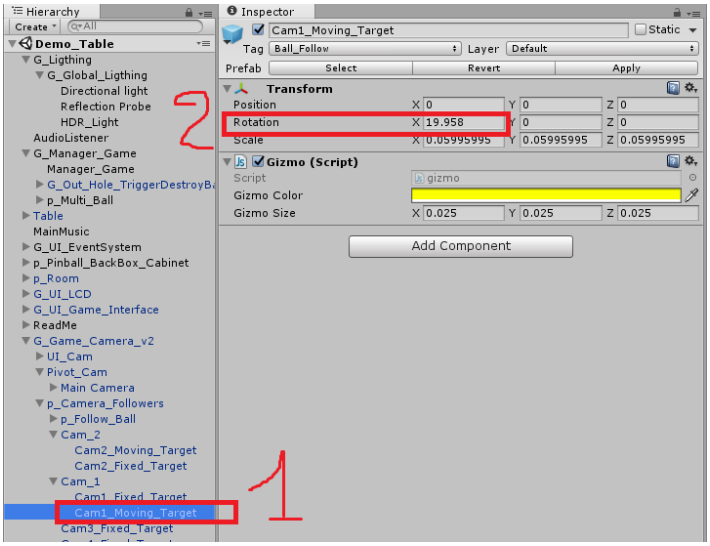
IMPORTANT Don't forget to uncheck the box when you finish to tweak the camera)

- 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

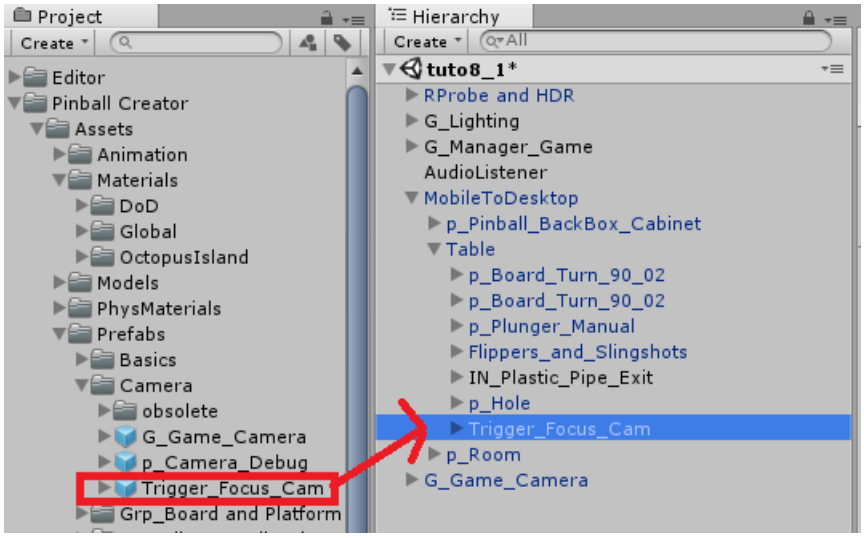
Example 1 : How to use FocusCam0 and FocusCam1 ?

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

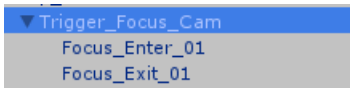
In this scene :
Ball is ejected outside the plunger and enter a hole.
Then the ball is eject from the pipe
You could press play and try the scene.

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

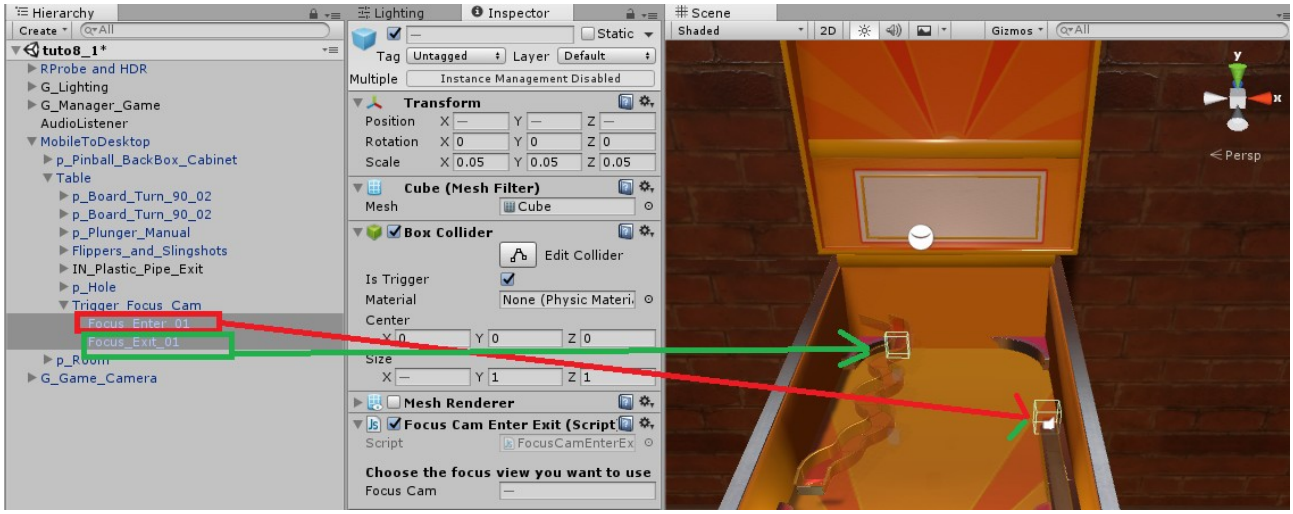


Step 3 : Choose where the camera zoom in and zoom out

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

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

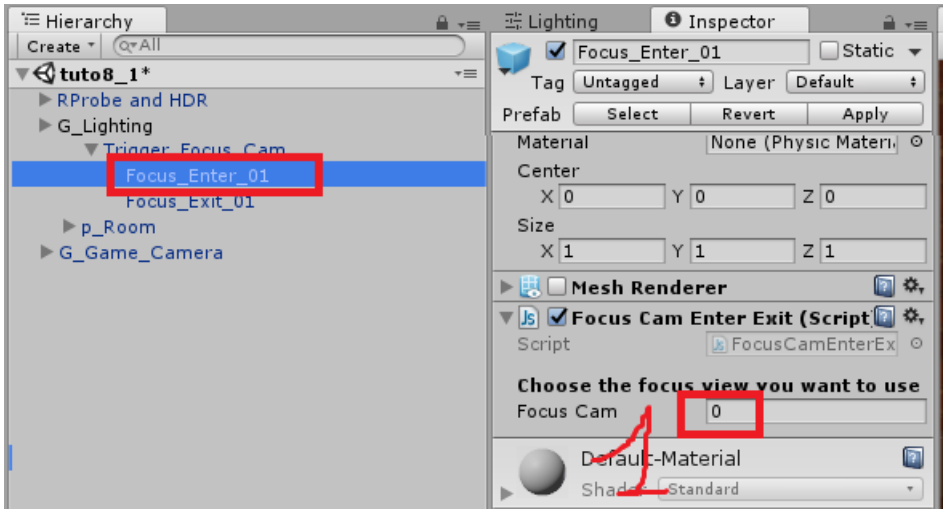
You should have this :



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 ?

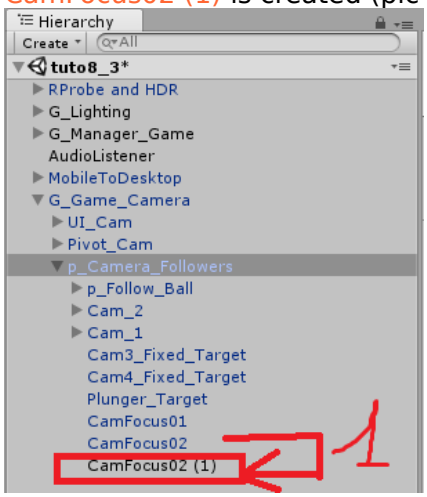
For the first Focus Cam : Move and rotate the gameObject CamFocus01
For the second Focus Cam : Move and rotate the gameObject CamFocus02

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

5.5 create a new Focus Cam

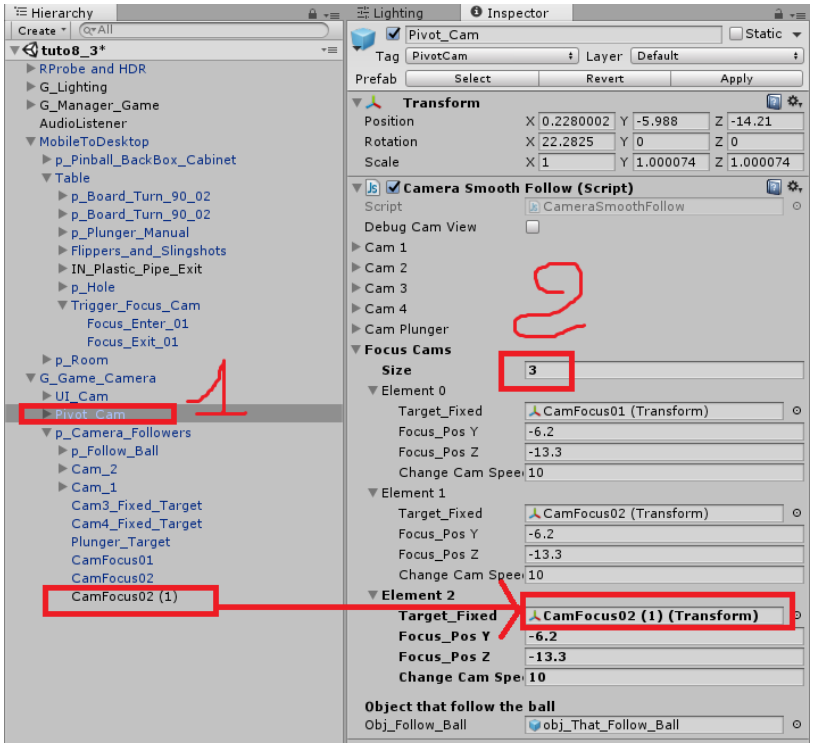
Step 1 : Open Tuto8_3 (Asset->Tuto-> Tuto8_3)

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



Step 2 : Setup this new Cam

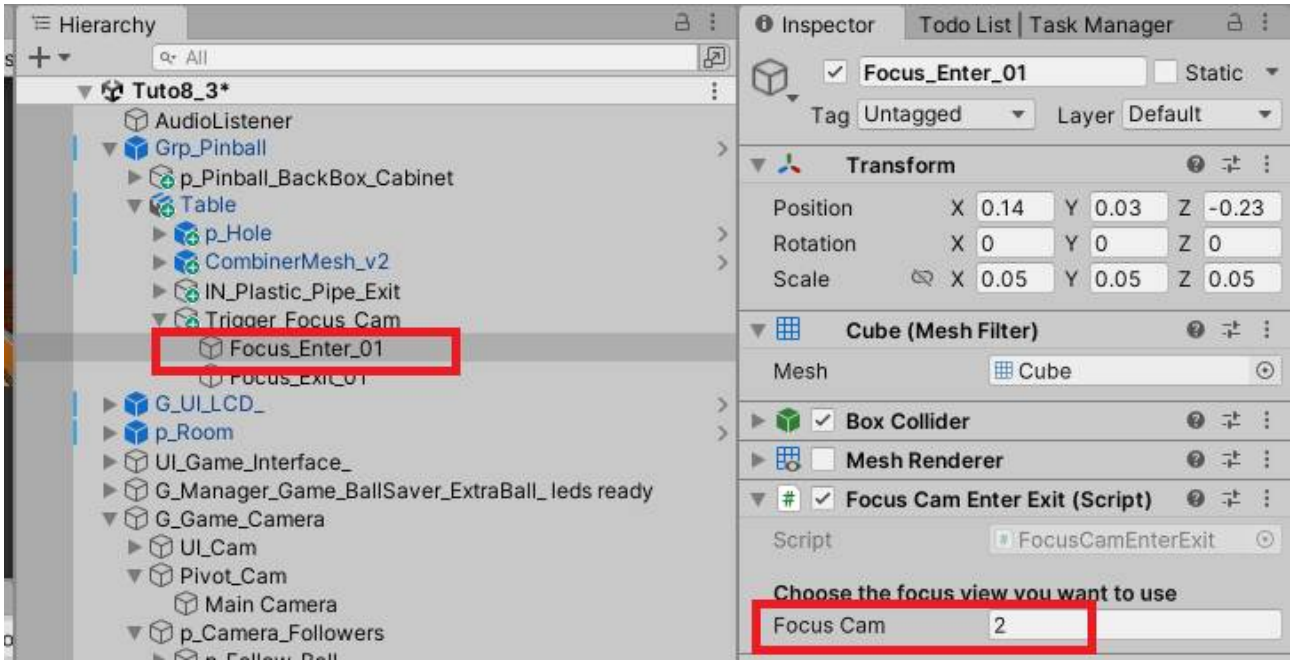
Select GameObject Pivot_Cam (G_Game_Camera->Pivot_Cam) (pic 1)
On Inspector Change Size to 3 On the FocusCams parameter (pic 2)
Drag and Drop CamFocus02 (1) Element 2-> Target_Fixed (pic 3)



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

Select gameObject Focus_Enter_01 on the Hierarchy (Grp_Pinball->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 :

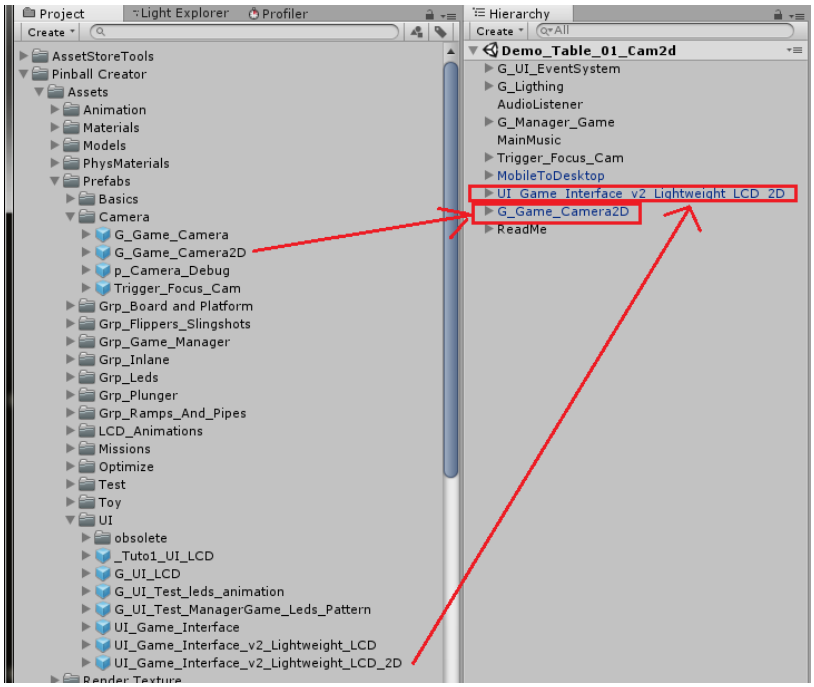
1a – How to use:

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.

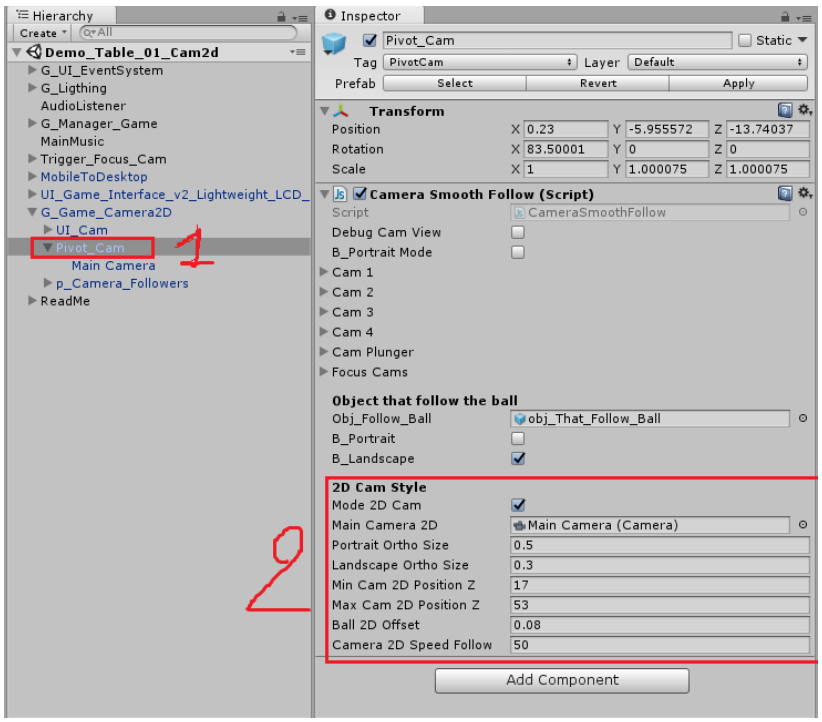
Camera 2D needs two prefabs to work:
- G_Game_Camera2D
- UI_Game_Interface_v2_Lightweight_LCD_2D

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)

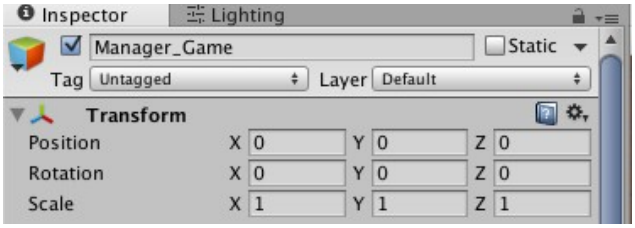


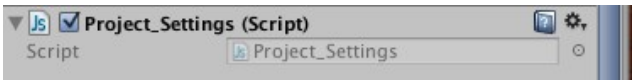
2D camera (spot 2) :

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

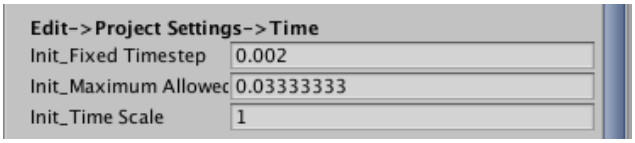
Game Manager :

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

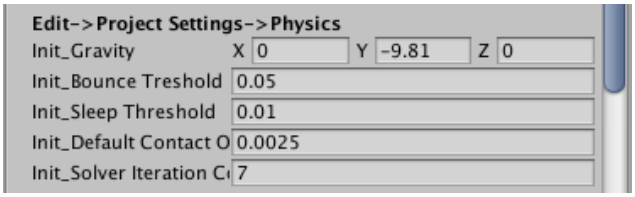




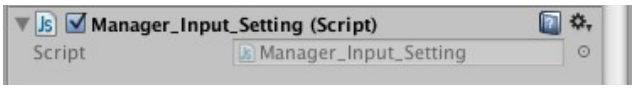
Project_Settings.cs
Manage Project Settings : Time and Physics



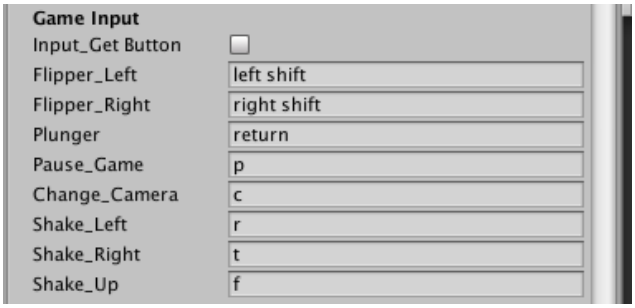
Project Settings : Time



Project Settings -> Physics

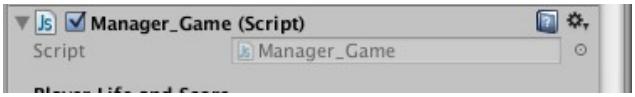


Manager_Input_Setting.cs
Manage Inputs

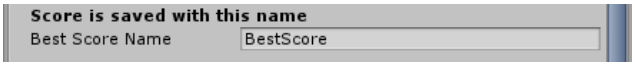


Choose your Inputs.

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)

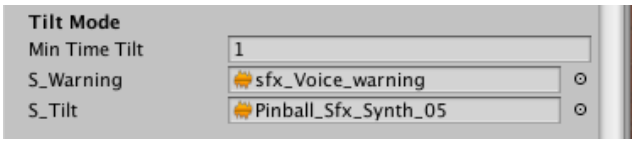


Manager_Game.cs
Manage game rules

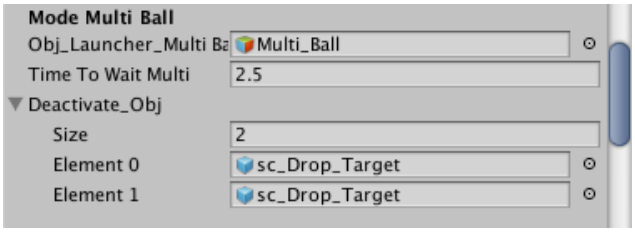


BestScoreName : Choose a name for the PlayerPrefs that save the score for this table. Choose a unique name For each Table

Life : number of life



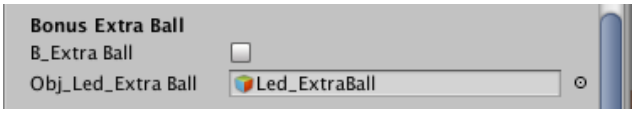
Tilt Mode : (nudge technique)
MinTimeTilt : Minimum time between two shakes. (seconds)
S_Warning : Sound if the table was shaken
S_Tilt : Sound if Tilt Mode Start
(More about Tilt Mode)



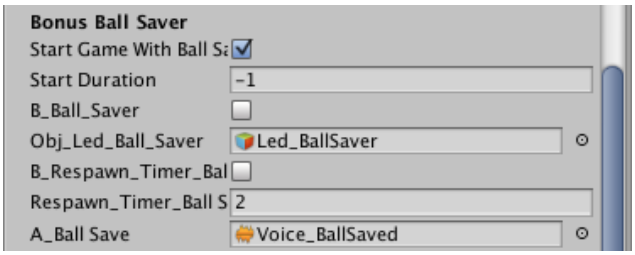
Obj_Launcher_MultiBall : Connect the gameObject Multi_Ball.

Time To Wait : Time to wait before multiball start

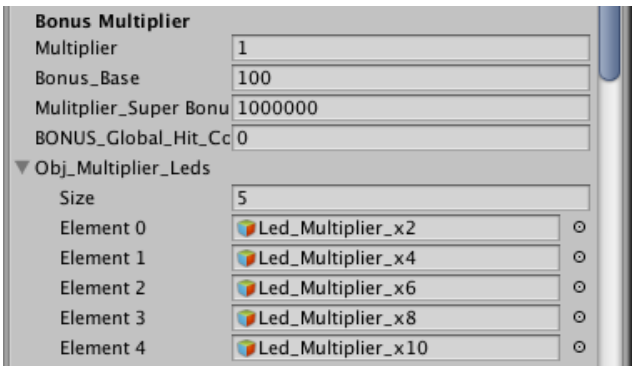
DeactivatObj : Connect target here if you want to activate and deactivate one or more targets when multiball start and stop



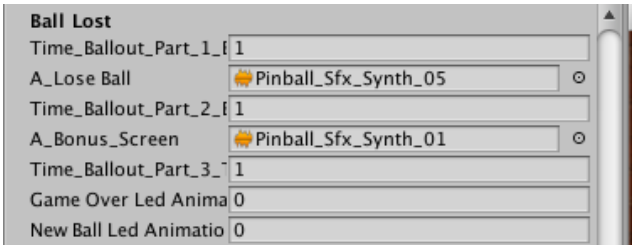
Extra Ball :
B_ExtraBall : true when Extra Ball Enable (automatically managed)
Obj_Led_ExtraBall : Connect a led. Led Switch On when Extra ball is enable for the player.
(more about how to connect a led)
(more about ExtraBall)



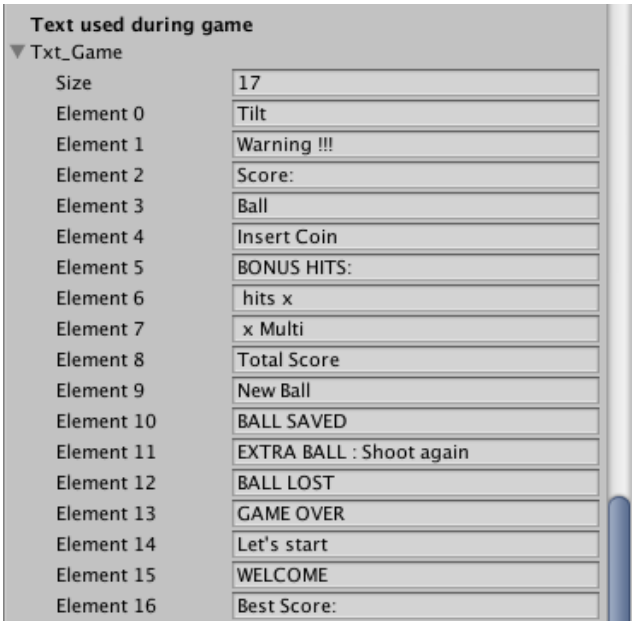
Ball Saver :
StartGameWithBallSaver : Every ball start with Ball Saver activated
StartDuration : Ball Saver duration. If StartDuration = -1 Ball Saver stop only when the ball is lost.
B_Ball_Saver : if true Ball Saver is activated. (automatically managed)
Obj_Led_Ball_Saver : Connect a led. Led Switch 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)

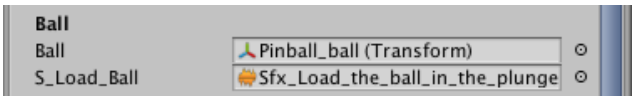


Txt_Game[0] : Tilt
Txt_Game[1] : Warning

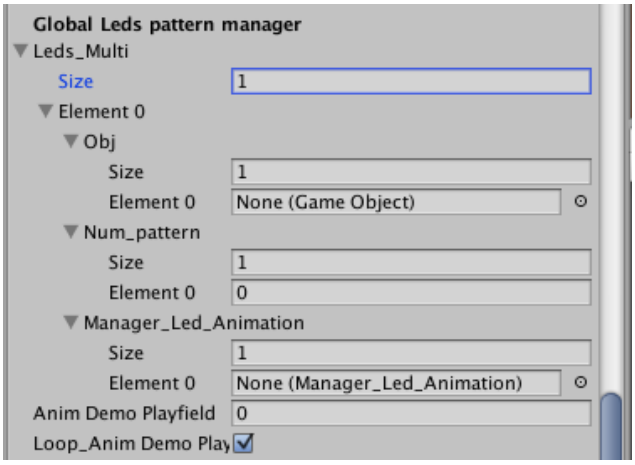
Txt_Game[2] + player_Score : Display the score when there is nothing else to display
Txt_Game[3] + (Ball_num+1) : Display the the ball number
Txt_Game[4] : Display a text when we are wait a player start the game

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



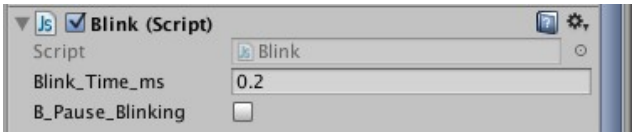
Ball :
Ball : Connect here ball prefab.
S_Load_Ball : Play this sound when ball is load on plunger.



(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 allows you to synchronize all the leds and make them blinking.

How to :

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

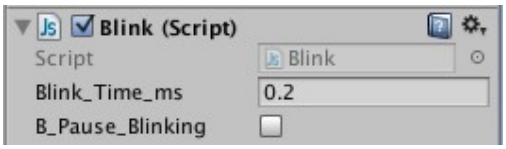
2 – Check box B_Blinking on each Led you want to blink. (You find this variable on each Led object. Change_Sprite_Renderer.cs)

How it works.

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

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

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



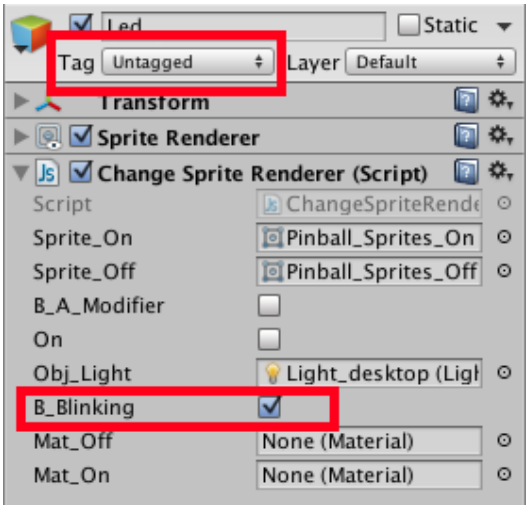
You find this script on Manager_Game in the hierarchy

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

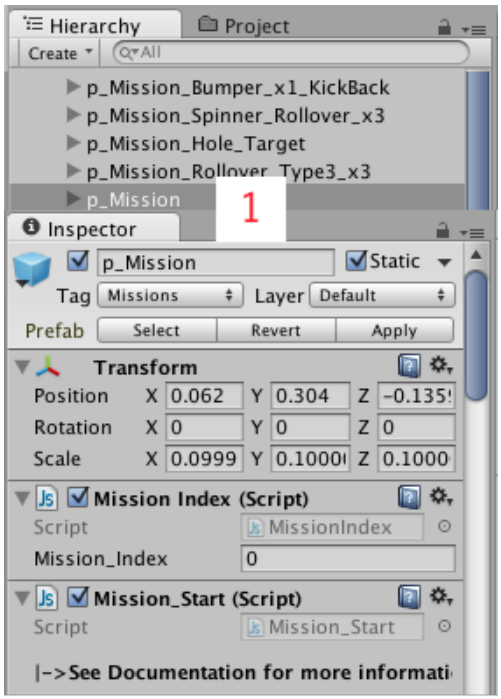
If true the will be blink when the scene start

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

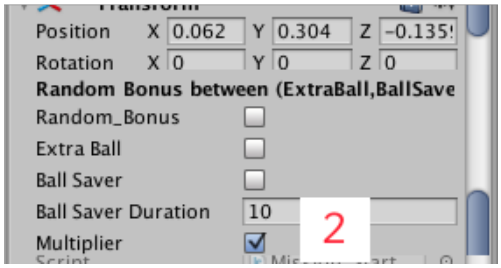
A mission could increase Multiplier when the mission is completed.

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 :
Kickback mission is explains in the documentation Part 1.

Multi-ball :
(more about multi-ball)

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

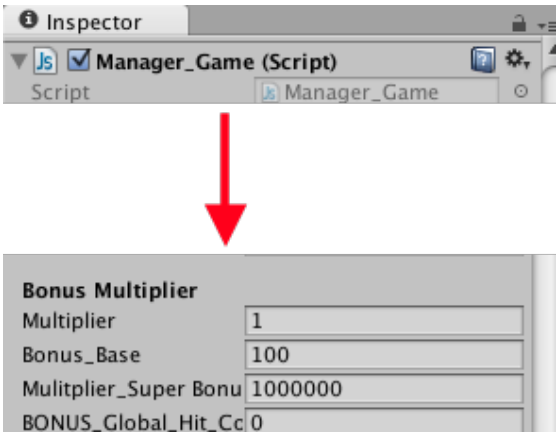
The score is calculated by adding :
- Points earned when ball hit a table mechanic on playfield.
- Points earned when a mission is completed.
- Points earned with Bonus Score (more info)

After the game has ended, Manager_Game.cs 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.

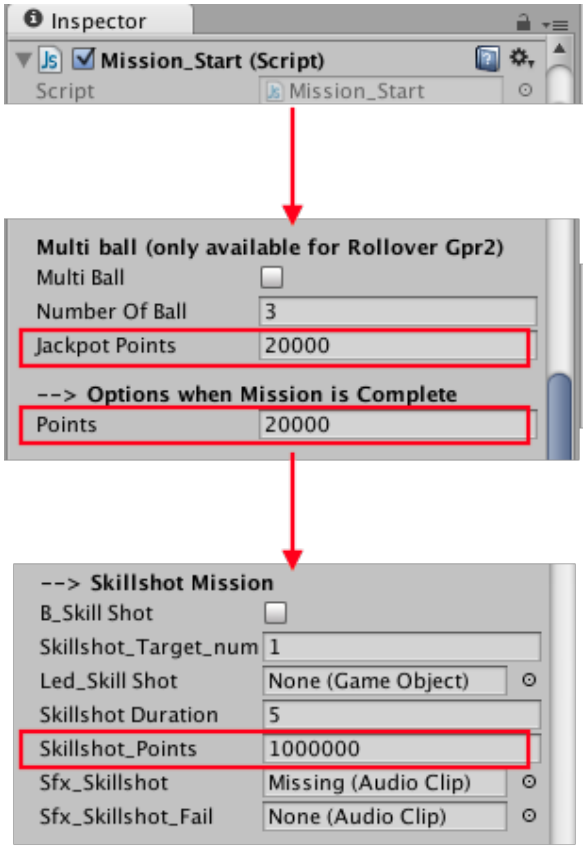


Inside Manager_Game (Manager_Game.cs) **:**

Section : Bonus Multiplier

Bonus_Base : (more info)

Multiplier_SuperBonus : Points earned if Multiplier = 10



Inside a misison : (script Mission_Start.cs)

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)

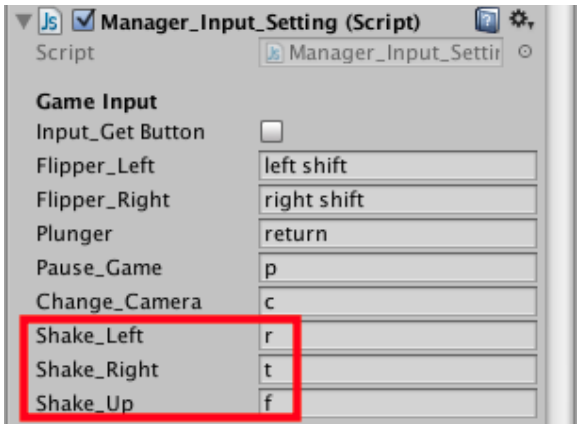
Skillshot :
(more about Skillshot here)

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.

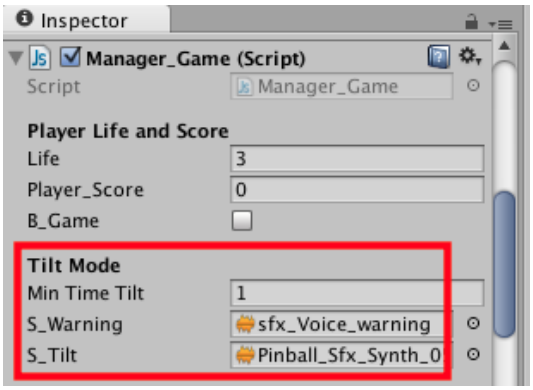


Player can hit the machine by pressing key input.

By default :

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

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



Tilt Mode is manage by script Manager_Game.cs 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

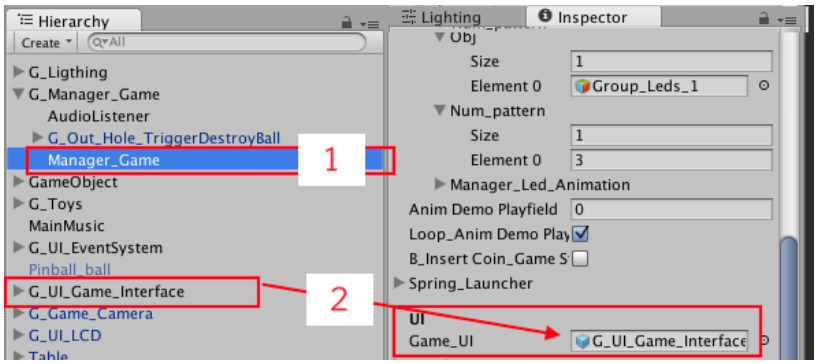
UI Interface :

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

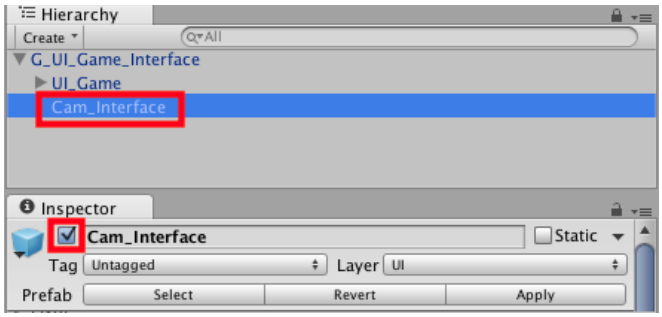
Important : Don't change buttons name inside the prefab. These names are used by script manager_Game.cs

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)



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.

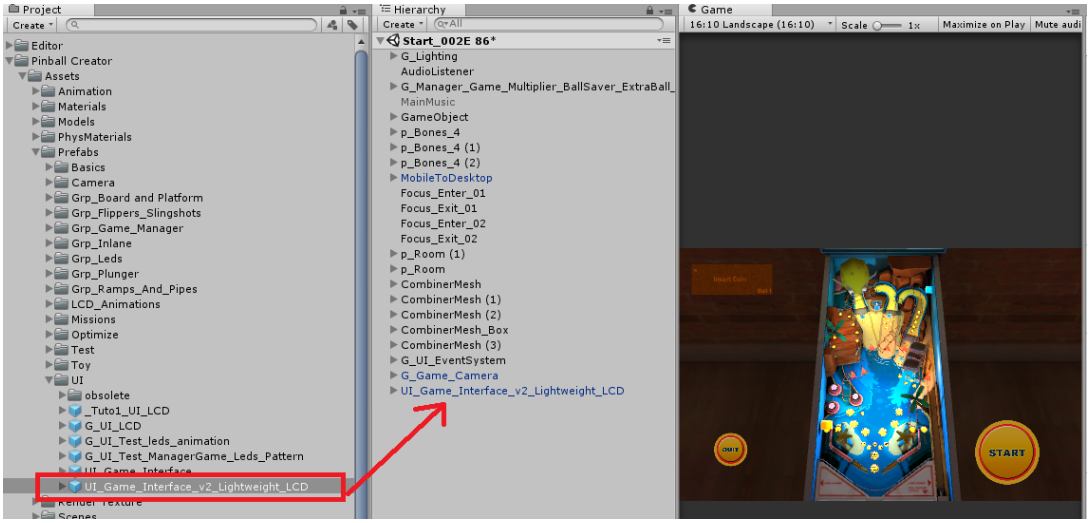
The UI system v1 (Optimize) associate to LCD screen v1 (prefab UI_Game_Interface + prefab G_UI_LCD)

Pros : LCD is display on the cabinet.

Cons : Loss about 5 FPS

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 :

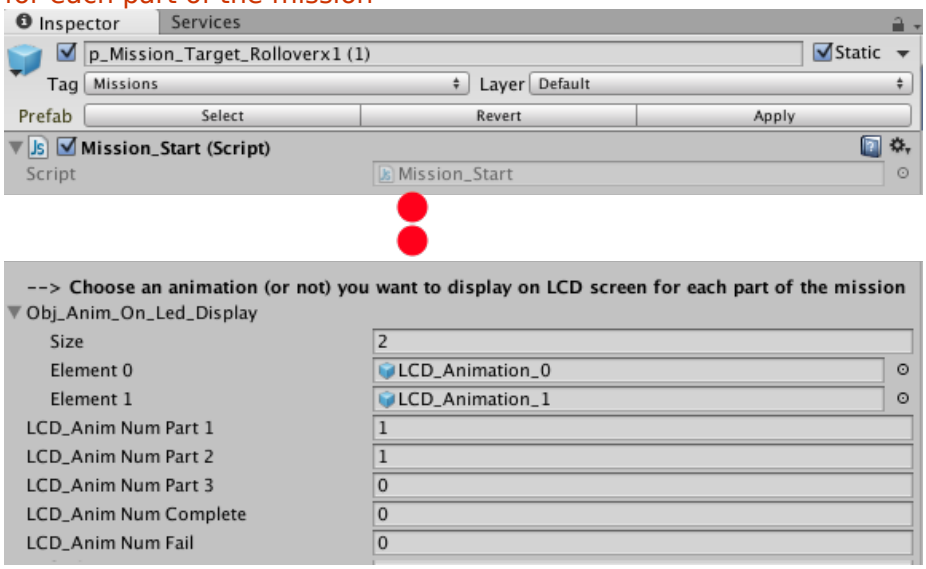
Drag and drop UI_Game_Interface on Hierarchy (assets->Prefabs-> UI-> UI_Game_Interface) (pic 1)

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.cs 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''

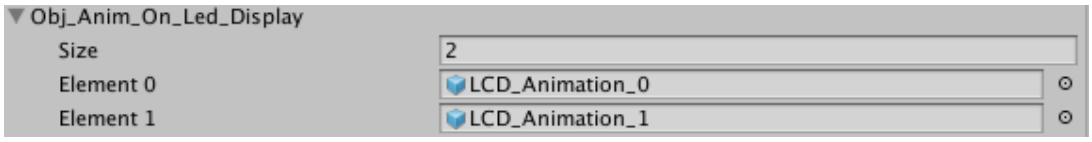


Step 2 : Choose LCD animations you want to use for mission

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.

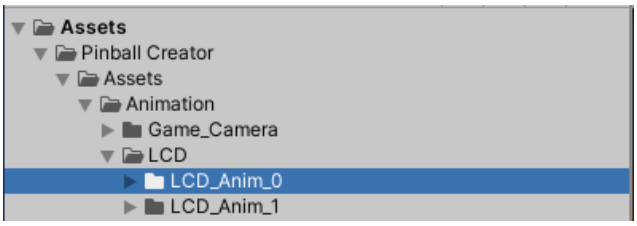
In the next example :
0 correspond to animation Element 0 : LCD_Animation_0
1 correspond to animation Element 1 : LCD_Animation_1.
Important : -1 mean no animation for this part.

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

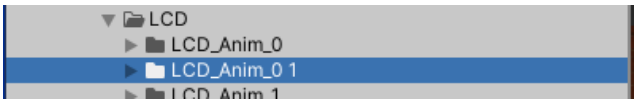
Step 1 : Preparation of elements Part 1

Duplicate LCD_Anim_0

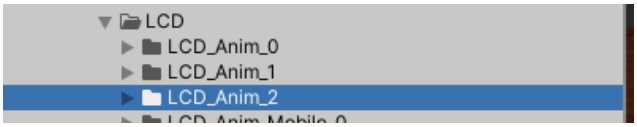
(Project -> Assets -> Animation -> LCD -> LCD_Anim_0)



The duplicated file is automatically selected.

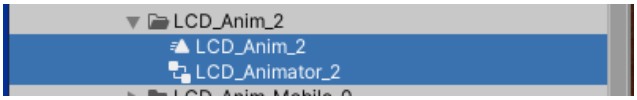


Rename it LCD_Anim_2



Rename files inside the folder.

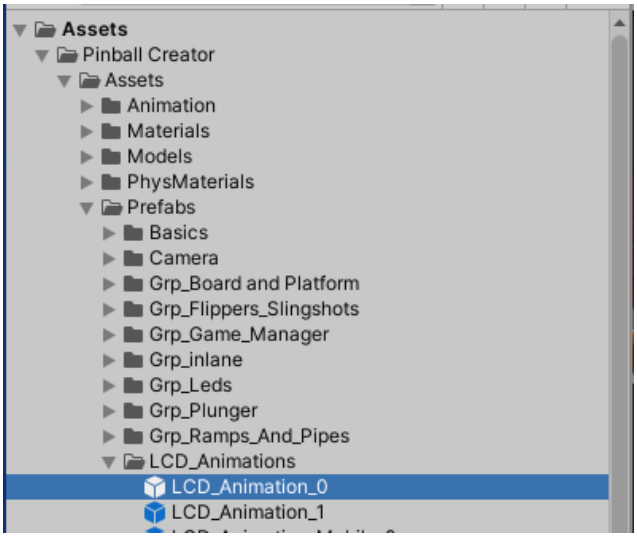
Change LCD_Anim_0 to LCD_Anim_2
and
LCD_Animator_0 to LCD_Animator_2.



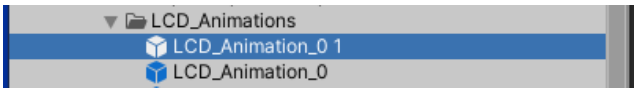
Step 2 : Preparation of elements Part 2

Duplicate LCD_Animation_0

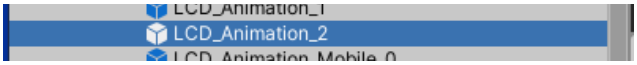
(Project -> Assets -> Prefabs -> LCD_Animations -> LCD_Animation_0)



The duplicated file is automatically selected.



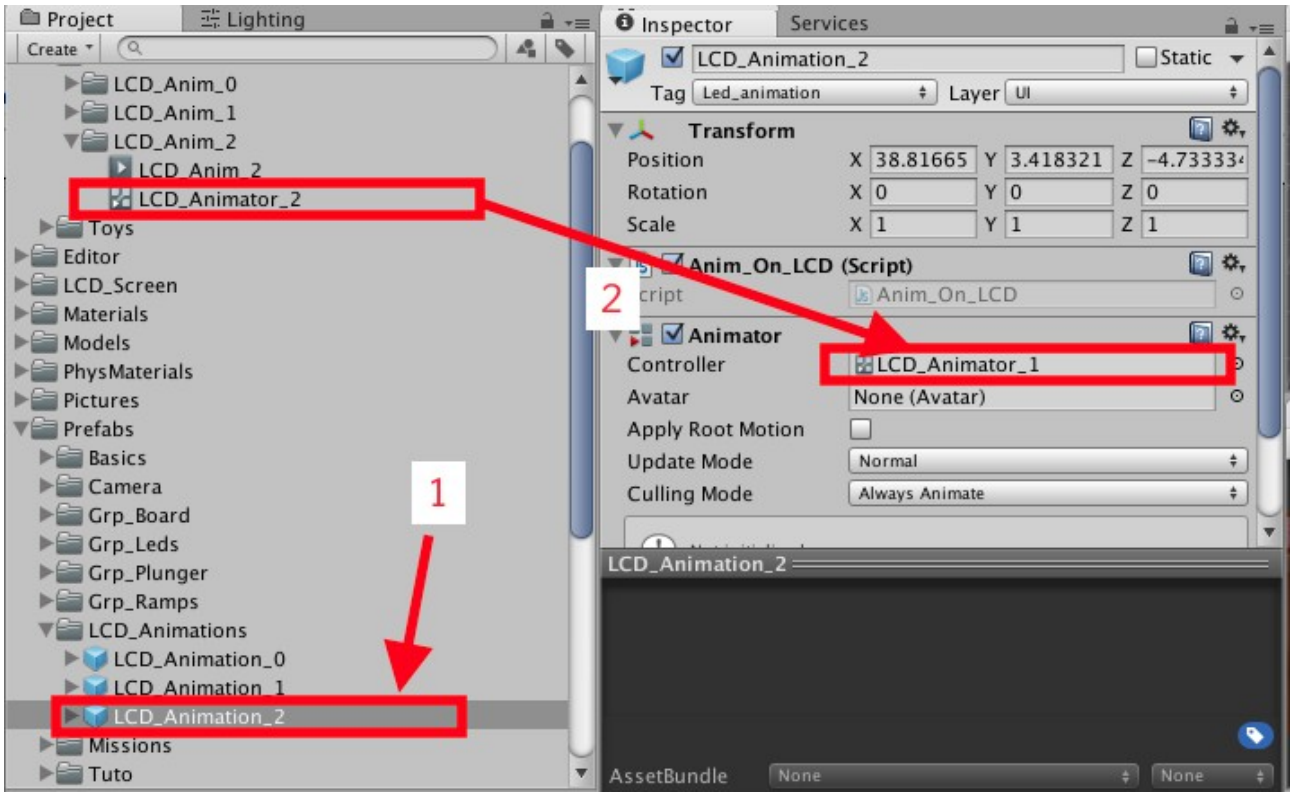
Rename it LCD_Animation_2 is created.



Step 3 : Connect Controller on LCD Animation Prefab

Select LCD_Animation_2 on Project folder (pic 1)

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



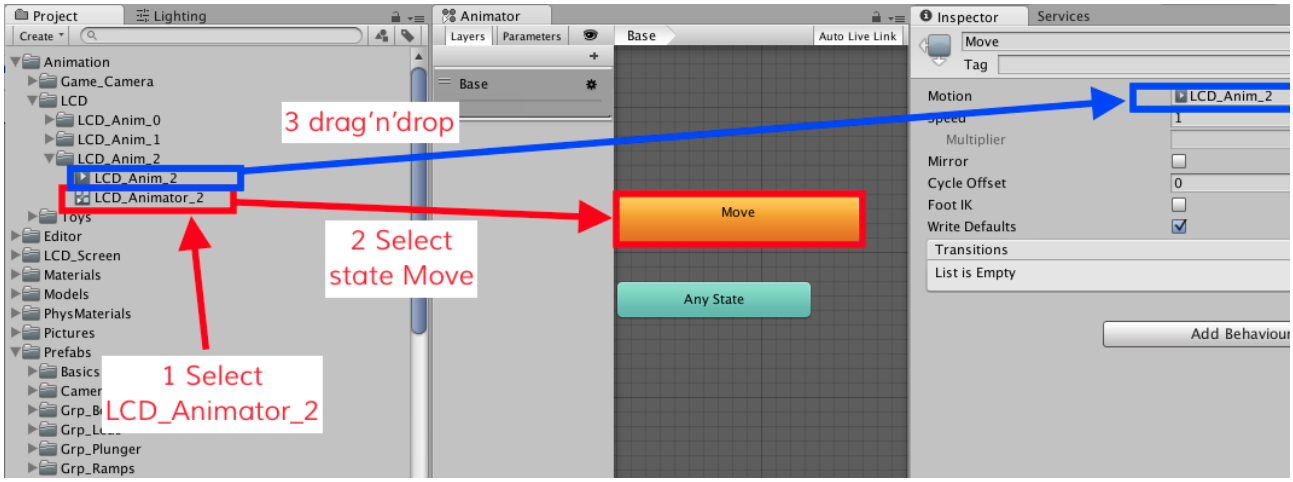
Step 4 : Connect animation on Animator

Open Animator window.

Select LCD_Animator_2 on Project folder (pic 1).

Select Animation State named Move (pic 2).

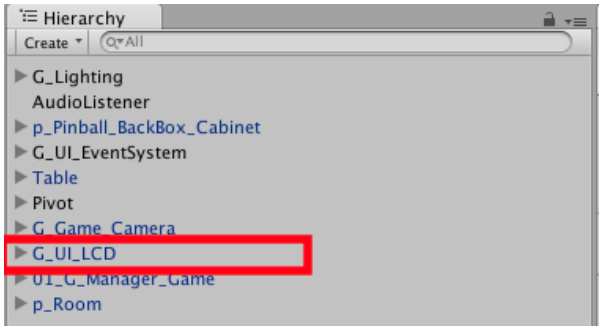
Drag'ndrop LCD_Anim_2 inside Motion on the Inspector (pic 3).



Step 5 : Put the new animation on scene

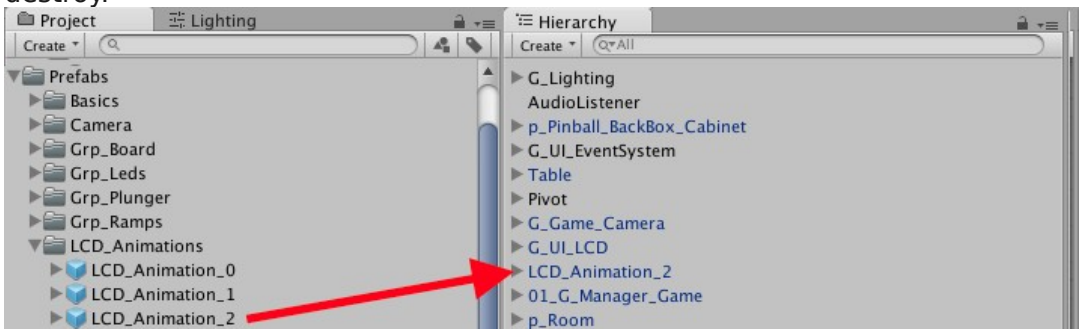
First Check if you have the prefab G_UI_LCD on Hierarchy.

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.



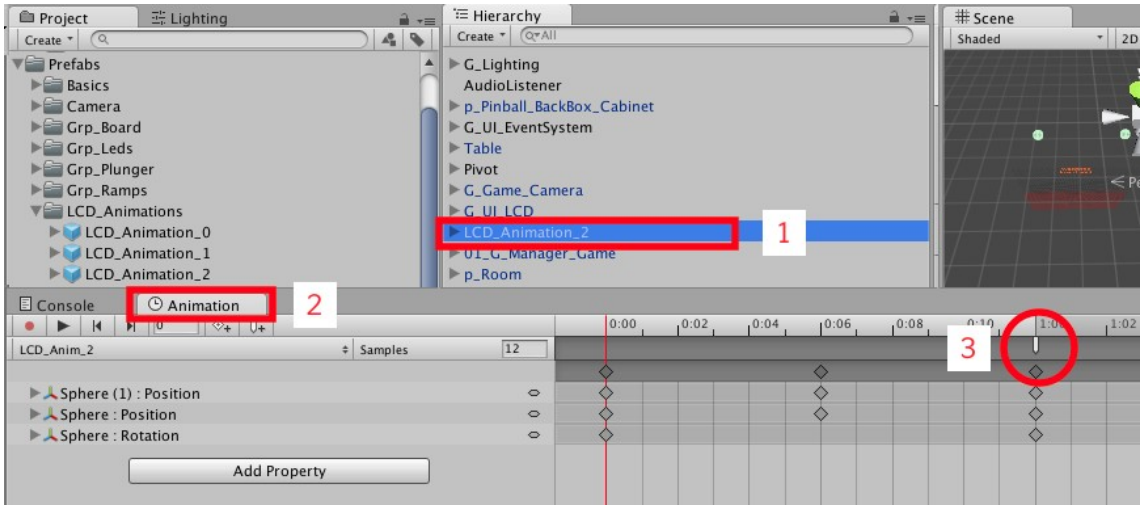
Step 6 : Create a new animation

Open Animation Window

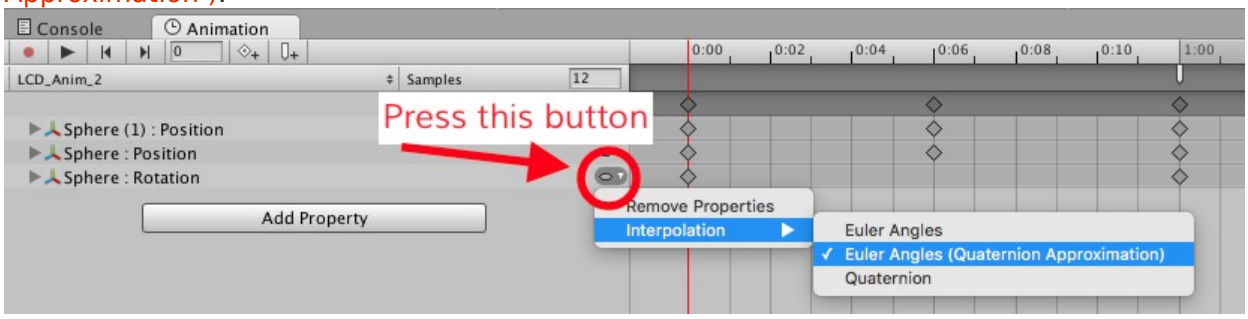
Then select LCD_Animation_2 on Hierarchy (pic 1).

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.

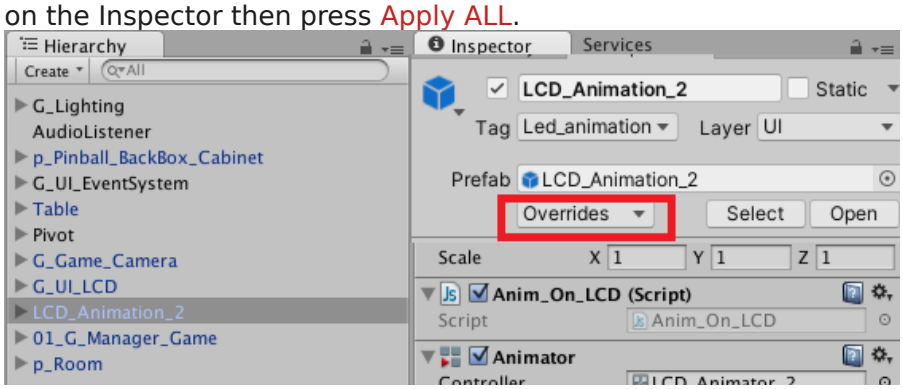


Tips : If you modify objects rotation choose Interpolation -> EulerAngles(Quaternion Approximation).



Step 7 : You could now delete the spheres. Create your own objects. Put them inside LCD_Animation_2 on Hierarchy. Create your animation.

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



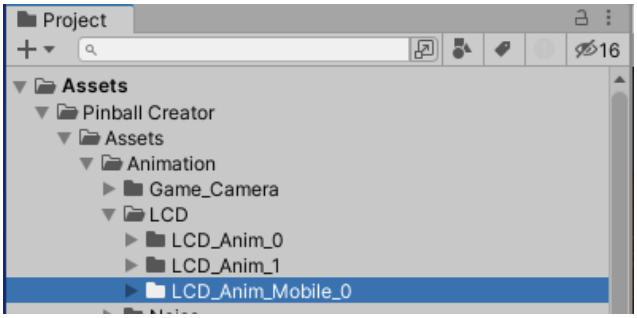
LCD Screen : Create Animation for LCD Screen

(if you use UI_Game_Interface_v2_Lightweight_LCD prefab).

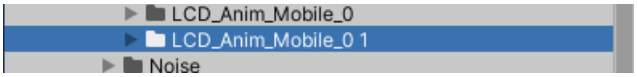
Step 1 : Preparation of elements Part 1

Duplicate LCD_Anim_Mobile_0

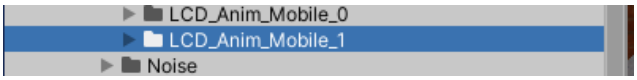
(Project -> Assets -> Animation -> LCD -> LCD_Anim_Mobile_0)



The duplicated file is automatically selected.



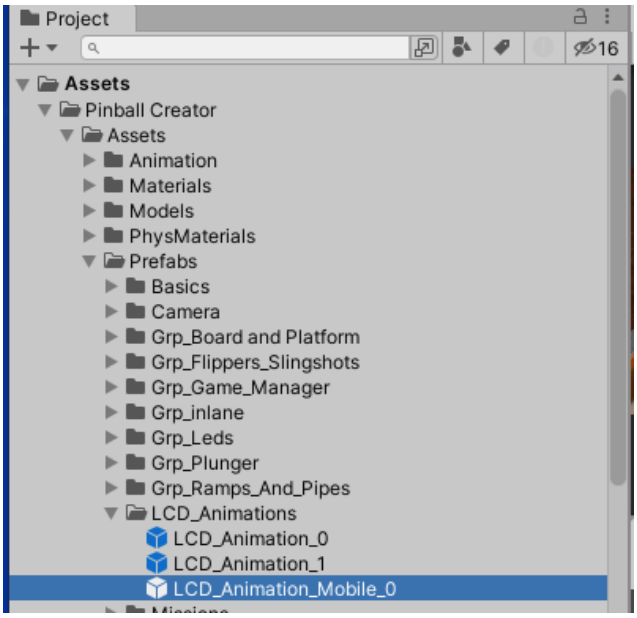
Rename it LCD_Anim_Mobile_1



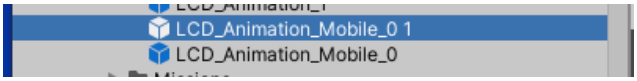
Step 2 : Preparation of elements Part 2

Duplicate LCD_Animation_Mobile_0

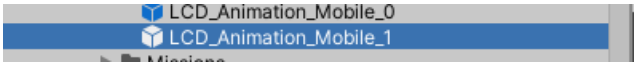
(Project -> Assets -> Prefabs -> LCD_Animations -> LCD_Animation_Mobile_0)



The duplicated file is automatically selected.



Rename it LCD_Animation_Mobile_1 is created.



Step 3 : Connect animation on Animator

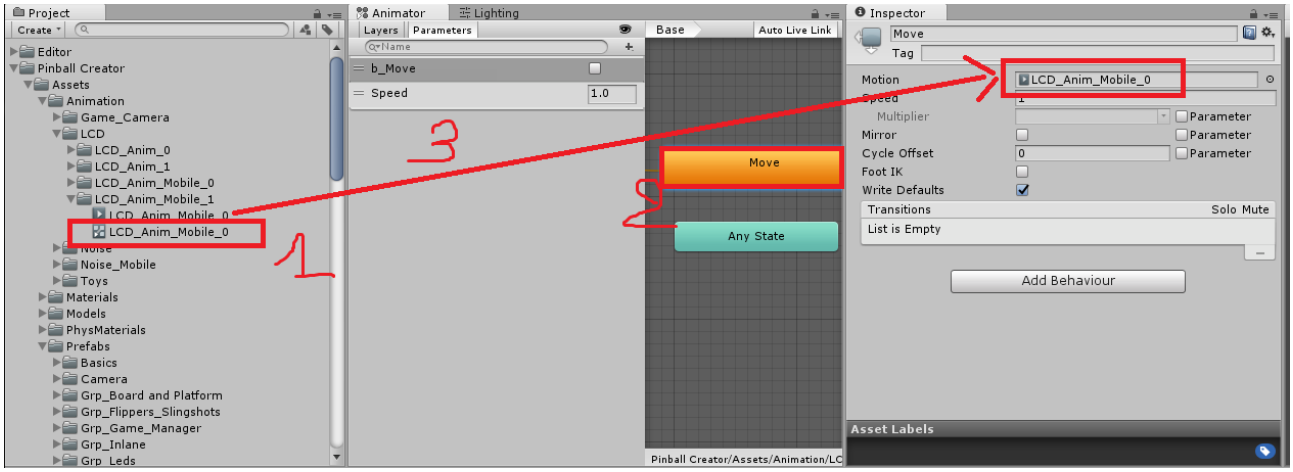
Open Animator window.

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).

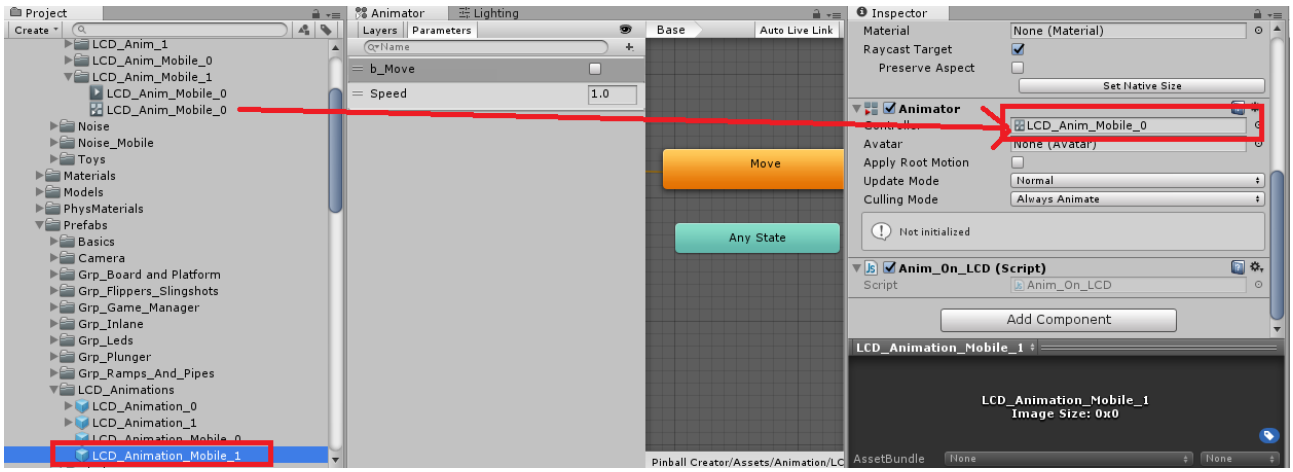
(Project -> Assets -> Animation -> LCD -> LCD_Anim_Mobile_1 -> LCD_Anim_Mobile_0)



Step 4 : Connect Controller on LCD Animation Prefab

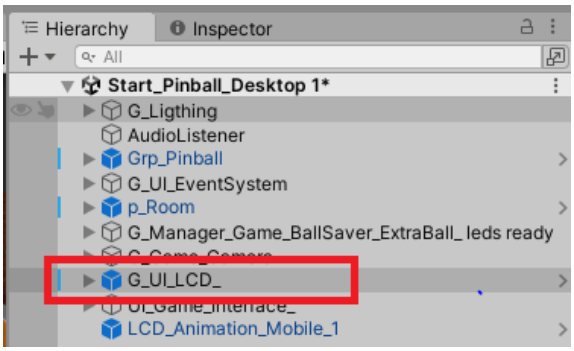
Select LCD_Animation_Mobile_1 on Project folder (pic 1)
(Project -> Assets -> Prefabs -> LCD_Animations -> LCD_Animation_Mobile_1)

Drag'ndrop LCD_Anim_Mobile_0 inside Animator -> Controller on the Inspector (pic 2).
(Project -> Assets -> Animation -> LCD -> LCD_Anim_Mobile_1 -> LCD_Anim_Mobile_0)

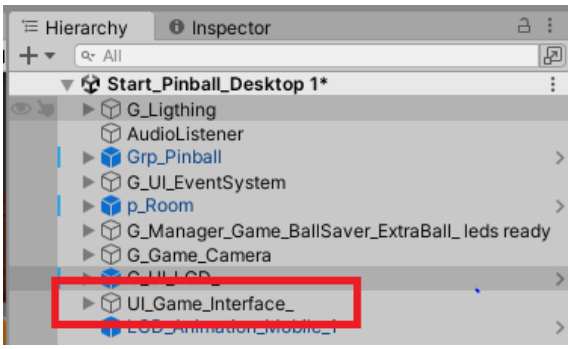


Step 5 : Put the new animation on scene

If you have the prefab G_UI_LCD on Hierarchy:
- Delete it.



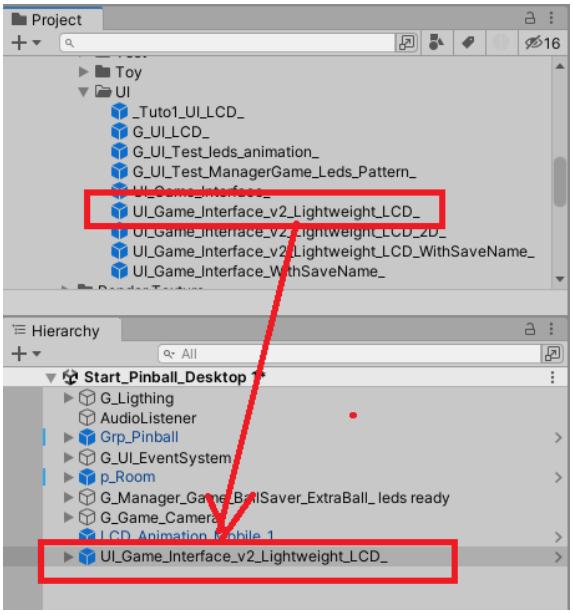
- Delete UI_Game_Interface too.



If you don't have UI_Game_Interface_v2_Lightweight_LCD on Hierarchy:

From project tab drag'n'drop UI_Game_Interface_v2_Lightweight_LCD on the root of the Hierarchy.

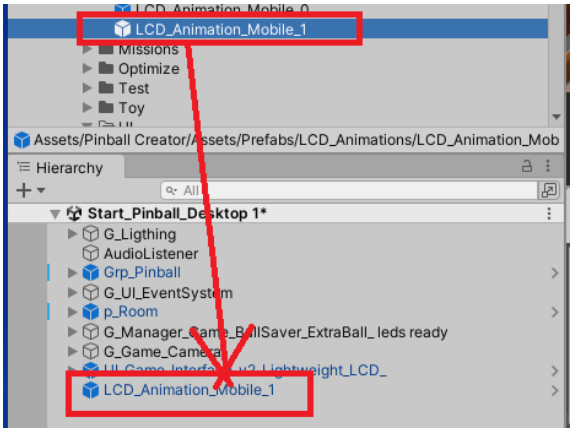
(Project -> Assets -> Prefabs -> UI -> UI_Game_Interface_v2_Lightweight_LCD).



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.



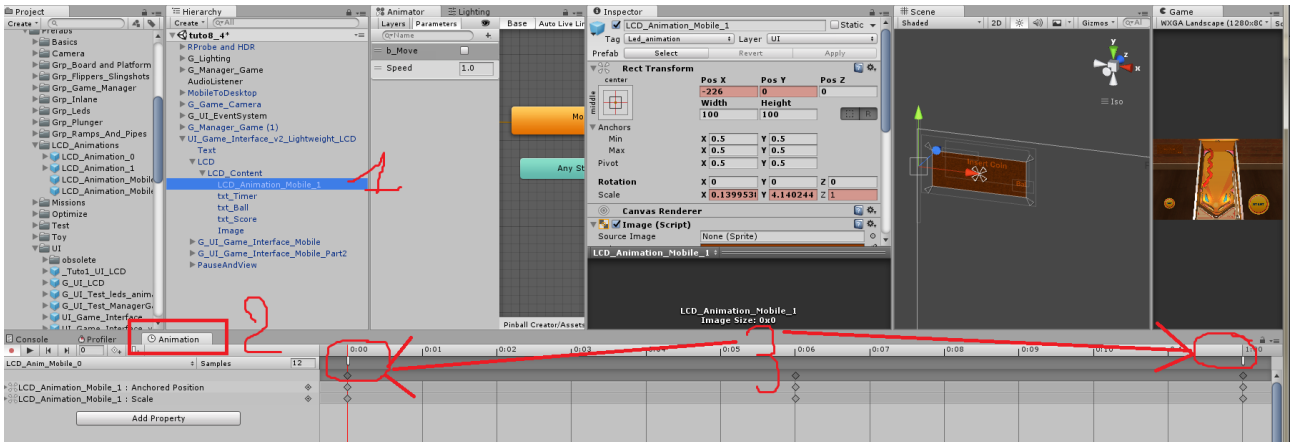
Step 6 : Create a new animation
Open Animation Window

Then select LCD_Animation_Mobile_1 on Hierarchy.

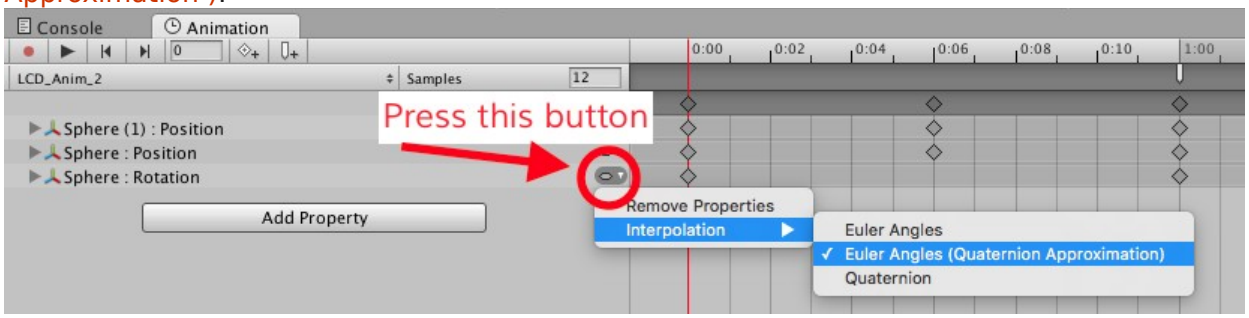
Drag and drop LCD_Animation_Mobile_1 inside (Hierarchy->UI_Game_Interface_v2_Lightweight_LCD->LCD->LCD_Content->) pic 1

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).



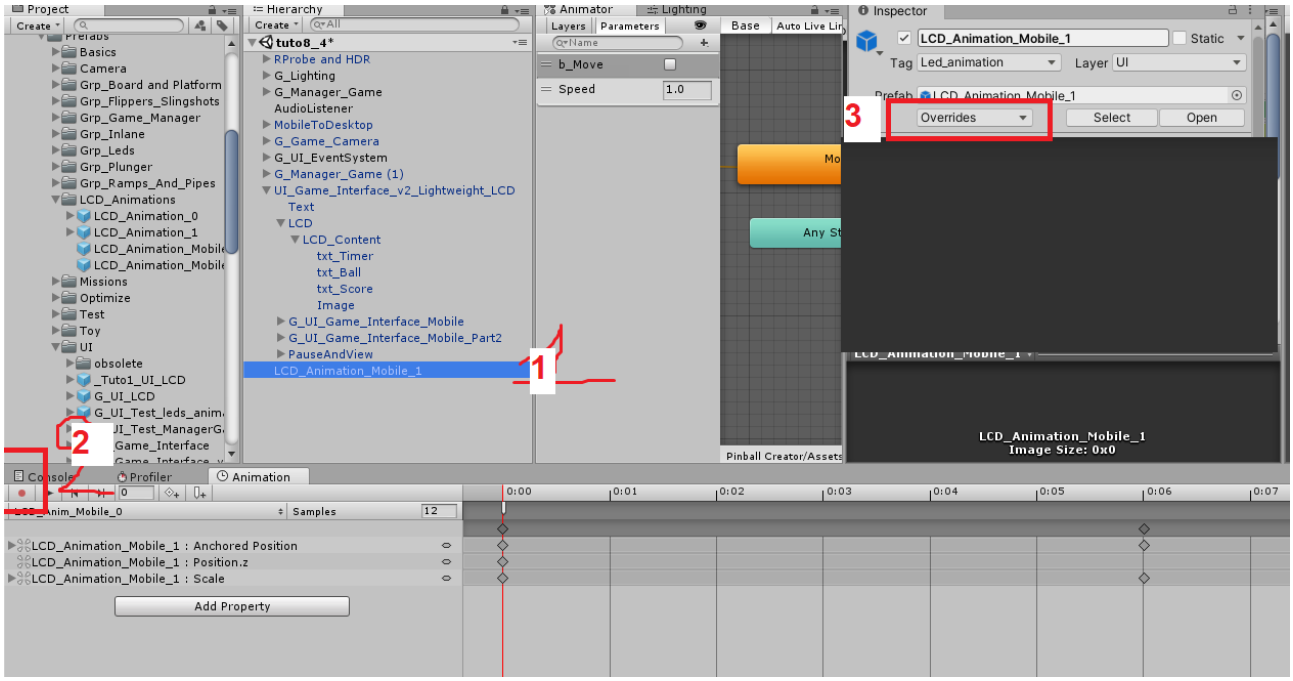
Step 7 : Create your animation.
Tips : To see your animation : Open Window -> Game . Animation appears on LCD Screen.

Step 8 : VERY VERY IMPORTANT . When it's done.

Put LCD_Animation_Mobile_1 on the root of hierarchy (pic 1)

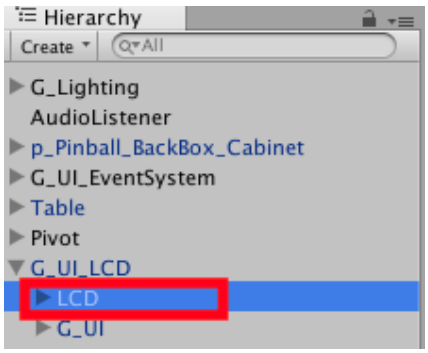
Uncheck red button on Animation Window (pic 2)

Select LCD_Animation_Mobile_1 on Hierarchy and press Overrides on the Inspector then press Apply ALL.



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.

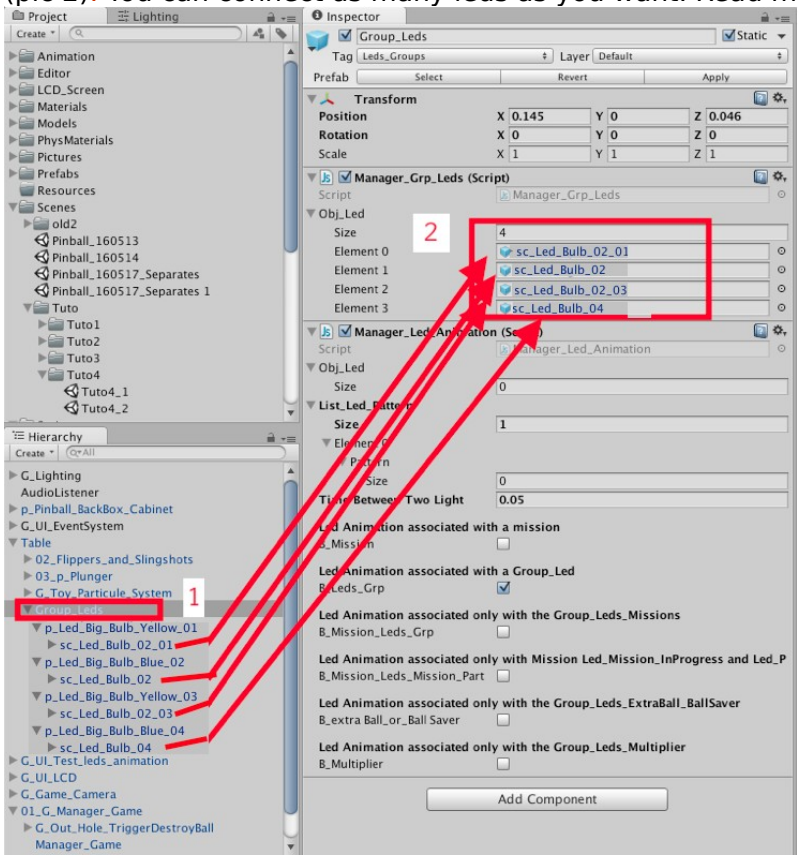


LED animation system :

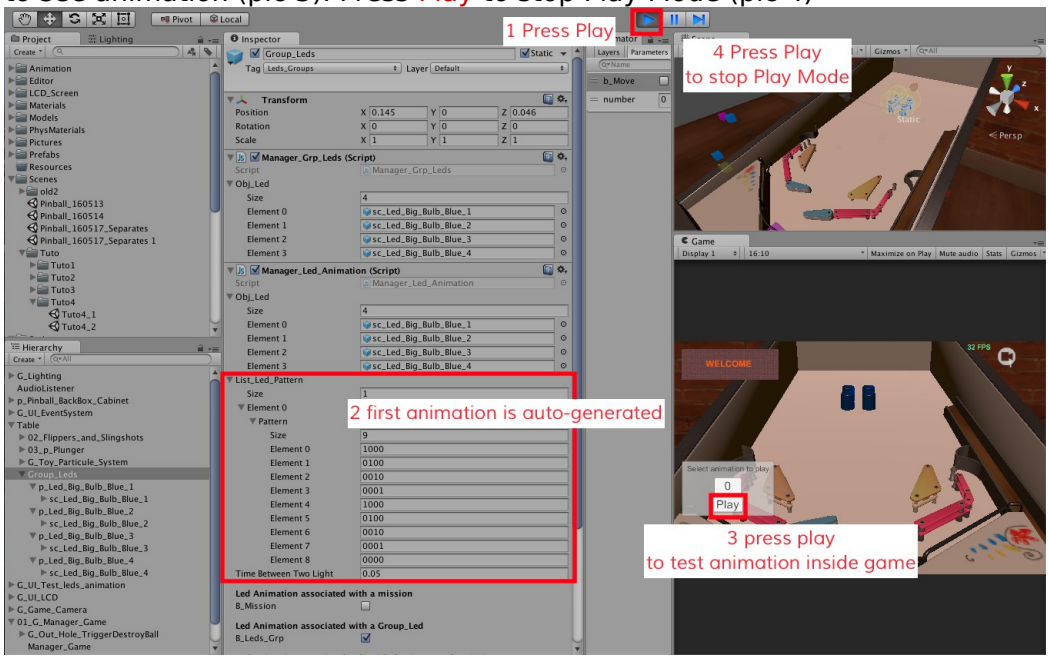
How to create Leds animation.

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.cs (pic 2). You can connect as many leds as you want. Read more about how to connect led here

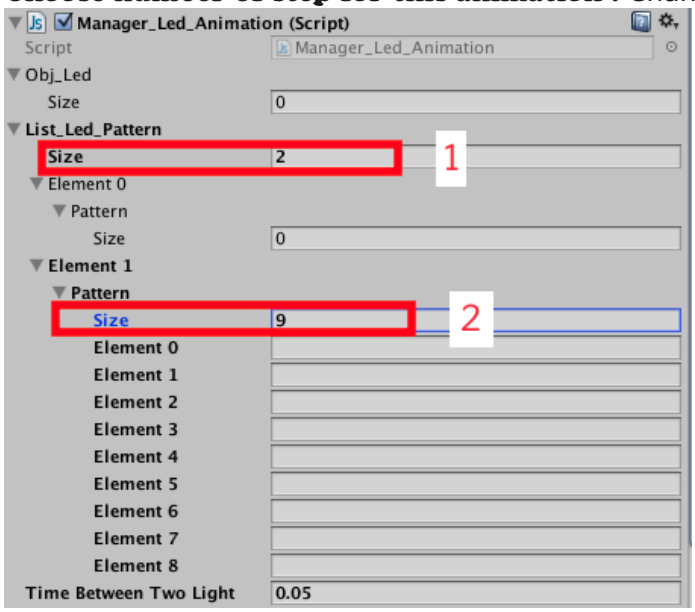


Step 3 : Press Play (pic 1). First animation is auto-generated. (pic 2). Press button Play on Game to see animation (pic 3). Press Play to stop Play Mode (pic 4)



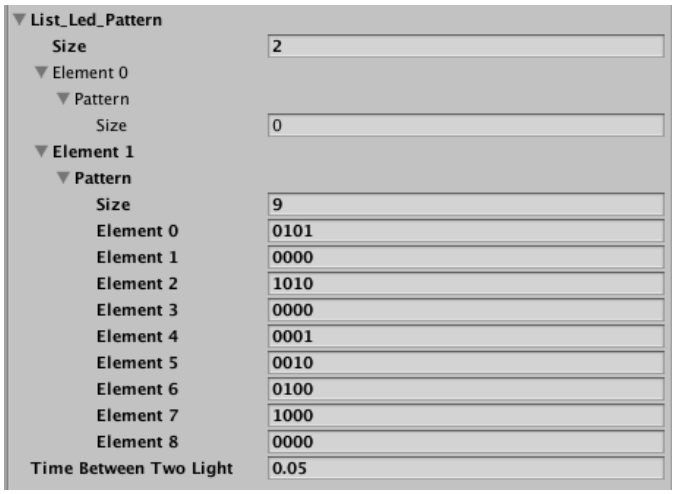
Step 4 :
Create a new animation : Change List_Led_Pattern -> Size to 2 (pic 1).

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



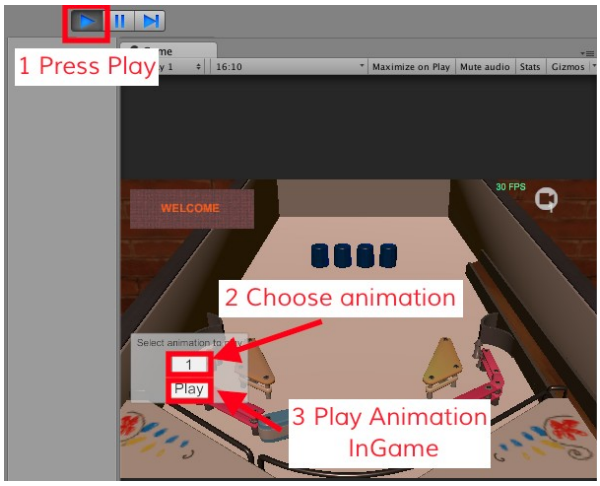
Create pattern : You need to describe the state of each led for each step of the animation.
In this example : if you want all the leds OFF. Write 0000.
If you want all the leds ON. Write 1111
If you want led 0 et 3 On. Write 1001.
Last example : if you have 6 leds and you want led 1 and led 4 On. Write 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 to choose animation (pic 2) .

Press Play to play animation (pic 3).

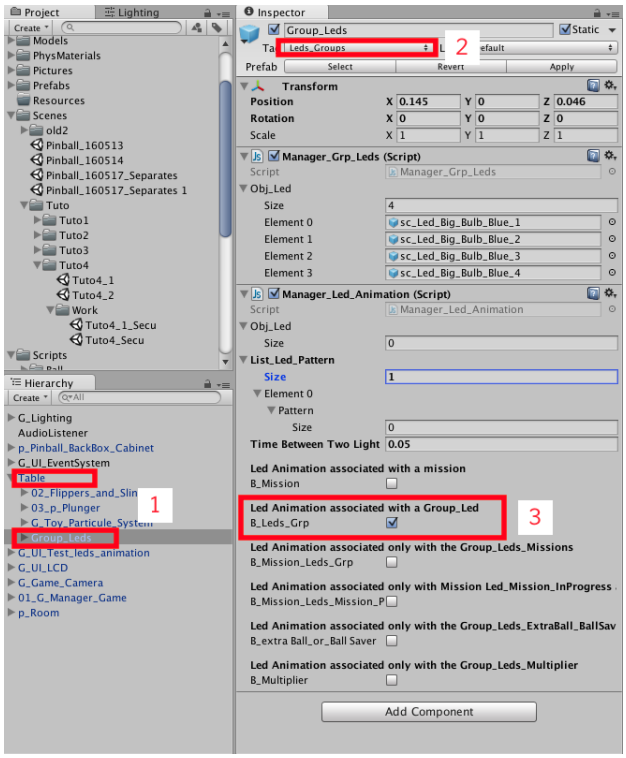


What you need to know depending of Led type (Examples) :

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)

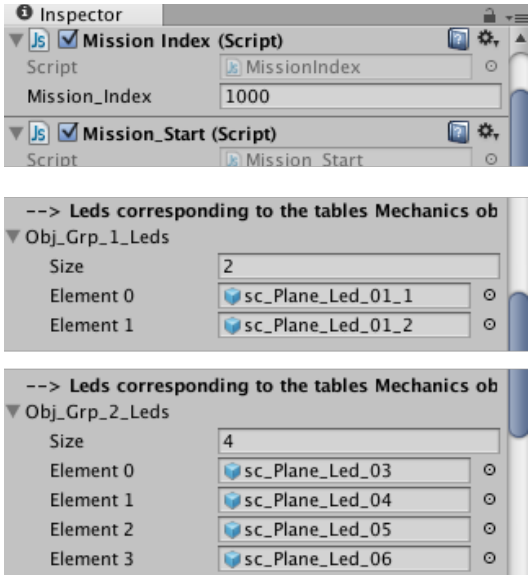
Step 2 : On Hierarchy open Table and Select Group_Leds (pic 1).
Tag = Leds_Groups (pic 2).
B_Leds_Grp = true (pic 3)
To create this type of Group you could start with prefab Group_Leds (Project -> Assets -> Prefabs -> Grp_Leds -> Group_Leds)



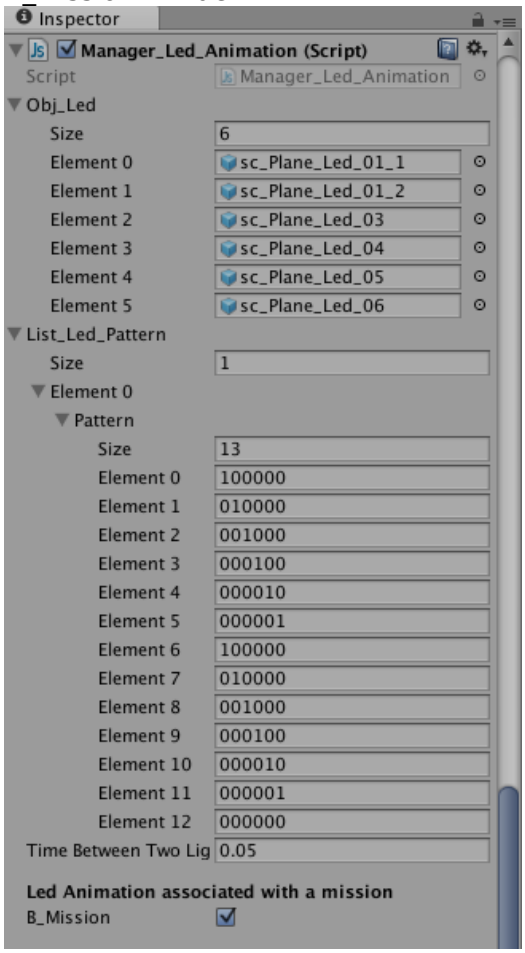
Case 2 : Group of Leds that is associated with mission (Obj_Led_Grp1 and Obj_Led_Grp2).

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).



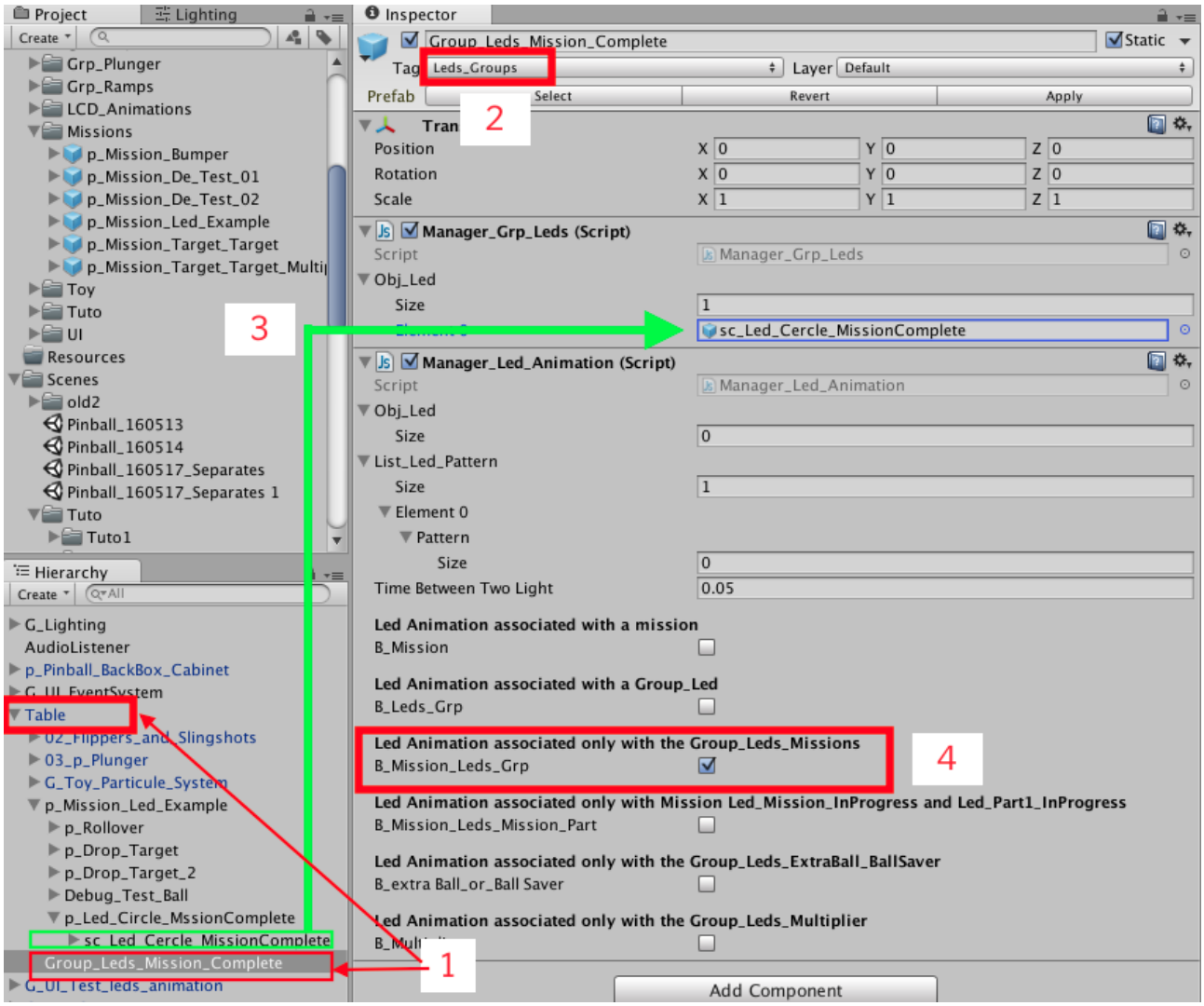
B_Mission = true



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)

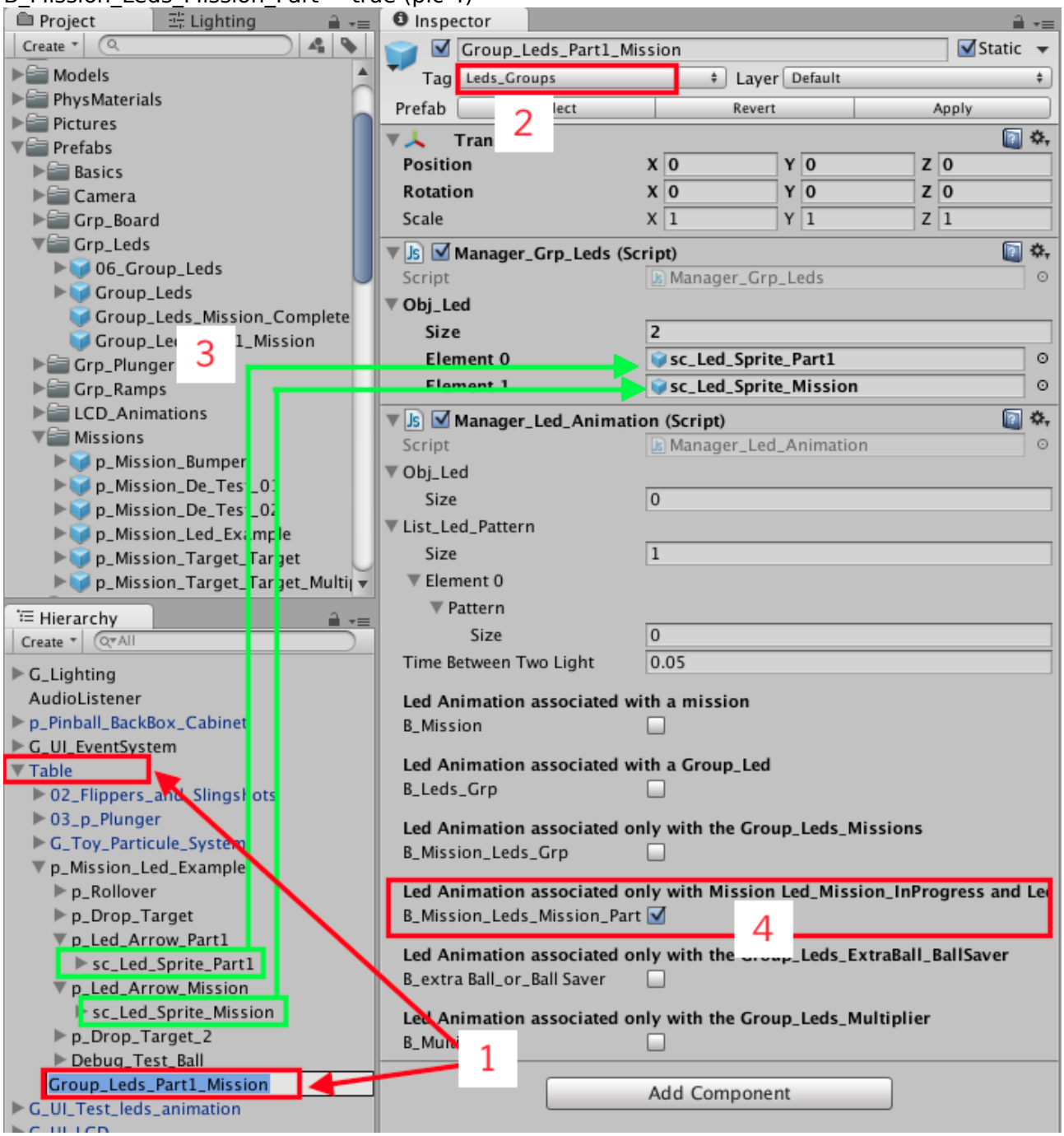
Step 2 : On Hierarchy open Table and Select Group_Leds_Mission_Complete. (pic 1)
Tag = Leds_Groups (pic 2).
Put your leds inside Obj_Led (pic 3)
B_Mission_Leds_Grp = true (pic 4)



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)

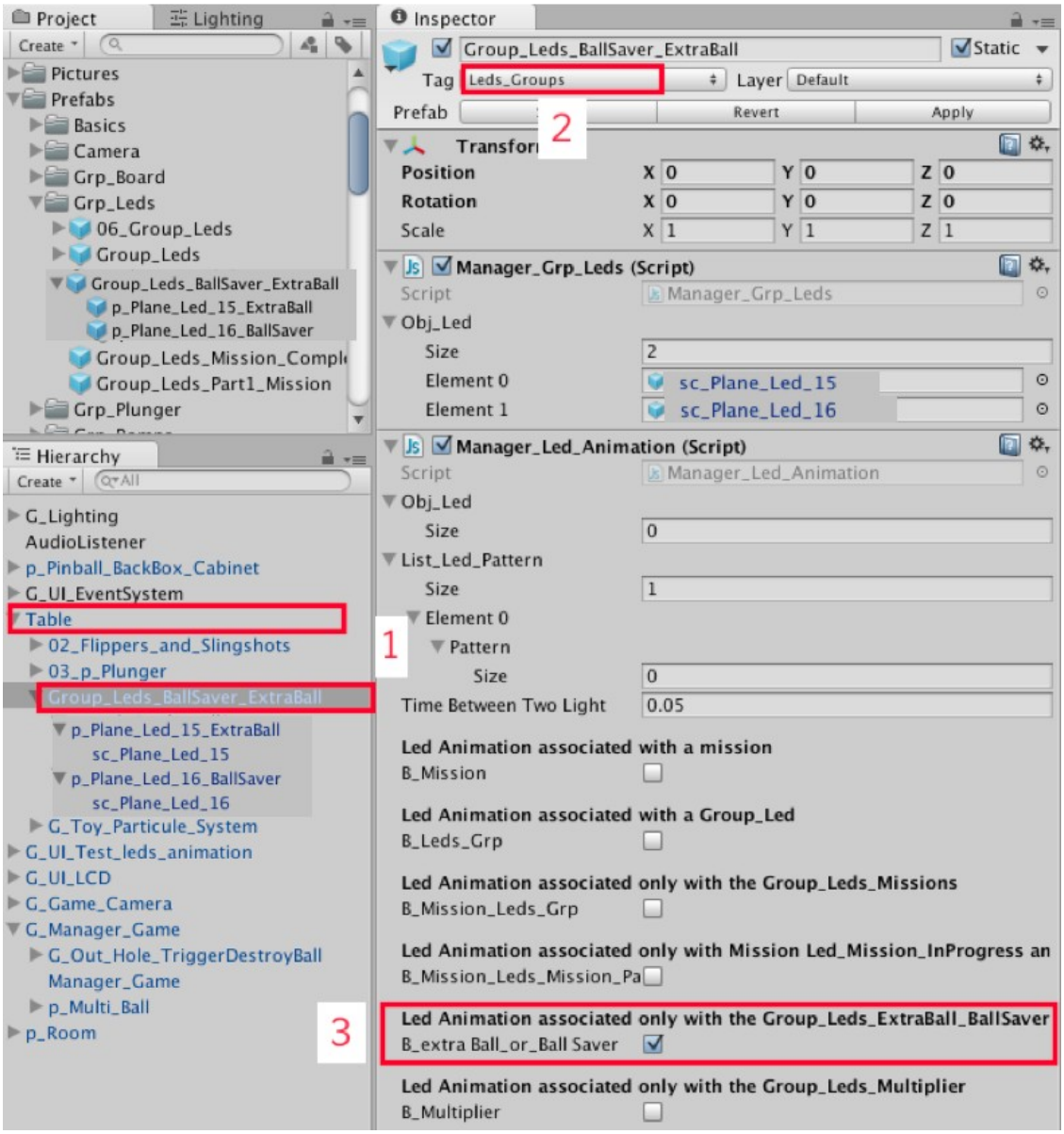
Step 2 : On Hierarchy open Table and Select Group_Leds_Part1_Mission. (pic 1)
Tag = Leds_Groups (pic 2).
Put your leds inside Obj_Led (pic 3)
B_Mission_Leds_Mission_Part = true (pic 4)



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)

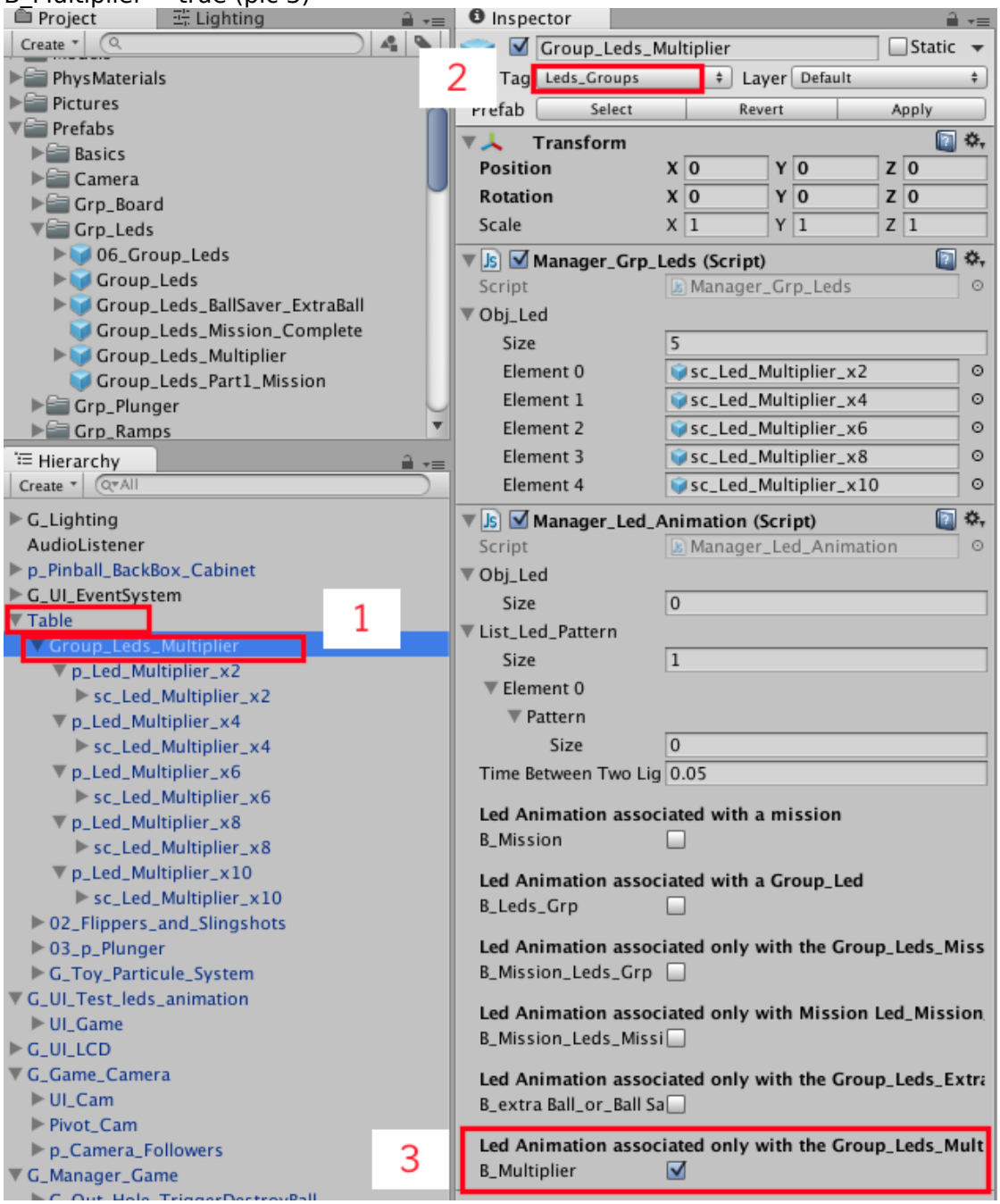
Step 2 : On Hierarchy open Table and Select Group_Leds_BallSaver_ExtraBall. (pic 1)
Tag = Leds_Groups (pic 2).
B_Extra_or_BallSaver = true (pic 3)



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)



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

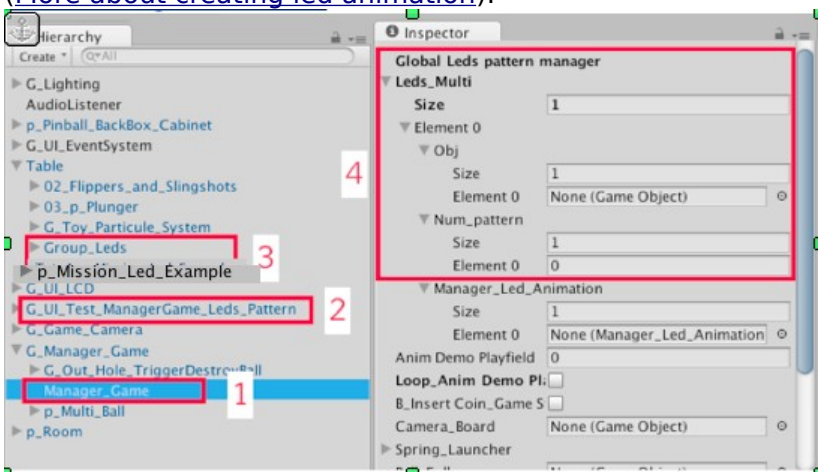
Step 1 : Open Tuto_5_1 (Project -> Assets -> Scenes -> Tuto -> Tuto5 -> Tuto5_1)

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

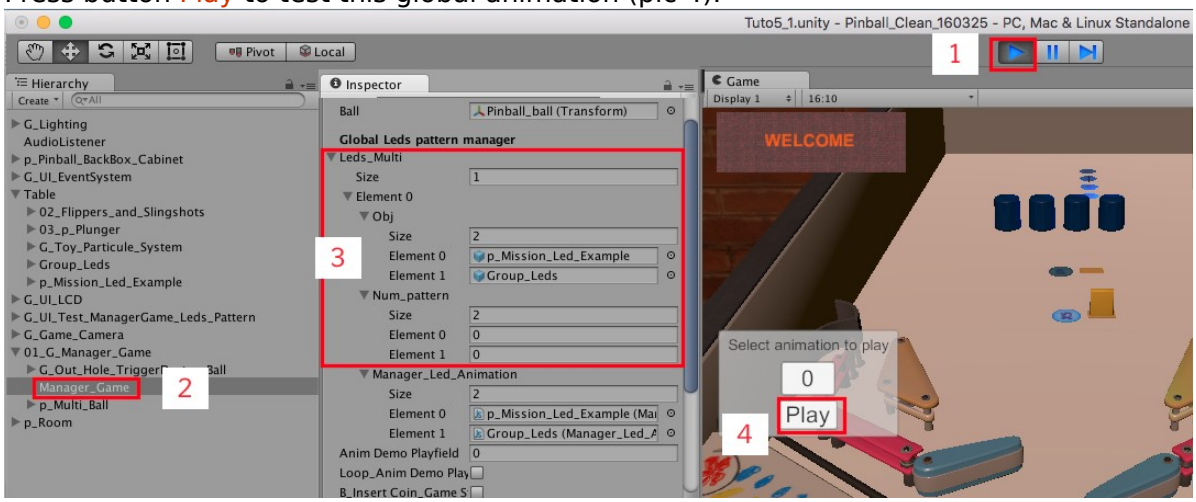
Led animations are manage by Leds_Multi on Manager_Game.cs (Inspector). (Pic 4)

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).



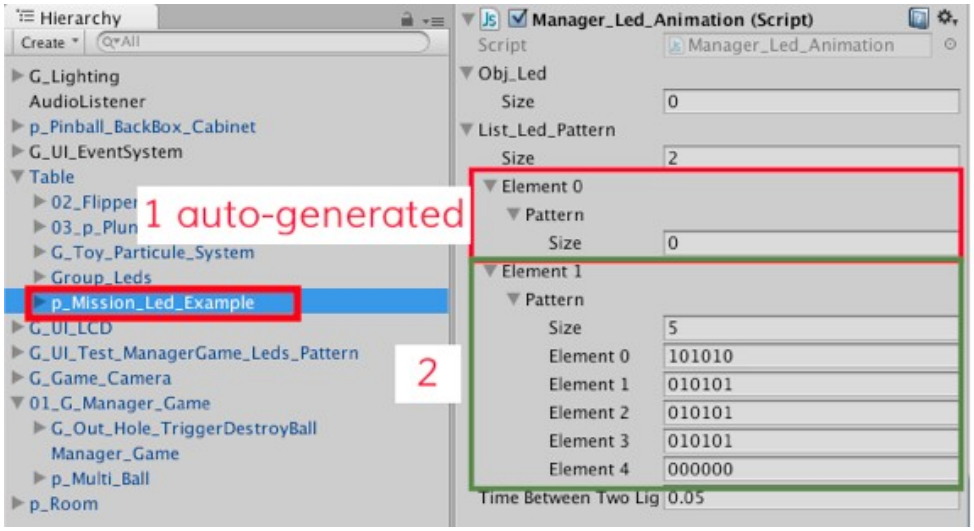
Step 3 : Test scene
Start Play Mode (pic 1)
Select Manager_Game on hierarchy (pic 2).
On Leds_Multi (Inspector) a combination of leds animations is auto-generated (pic 3). It combine all group of leds on Hierarchy. In this example Groups_Leds and _Tuto4_p_Mission_Led_Example.
Press button Play to test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



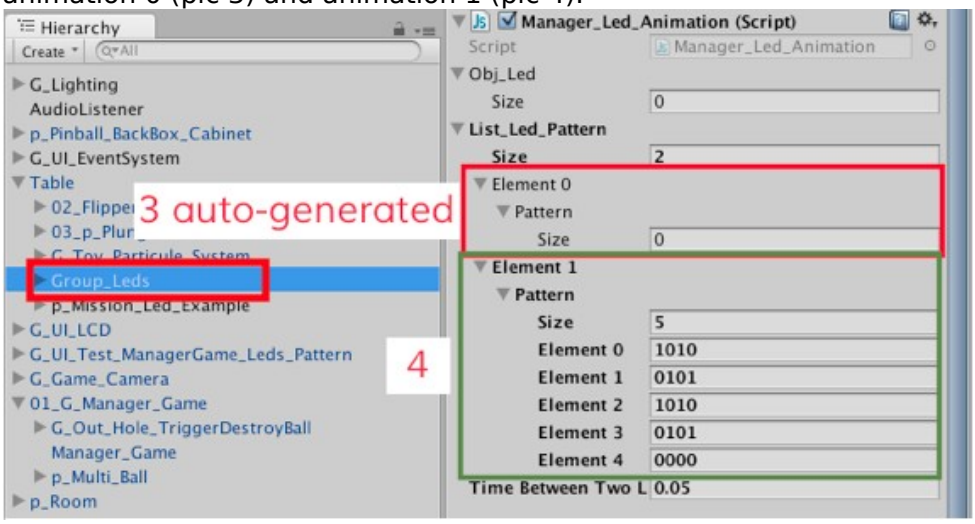
Press Stop before moving to the next part.

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).

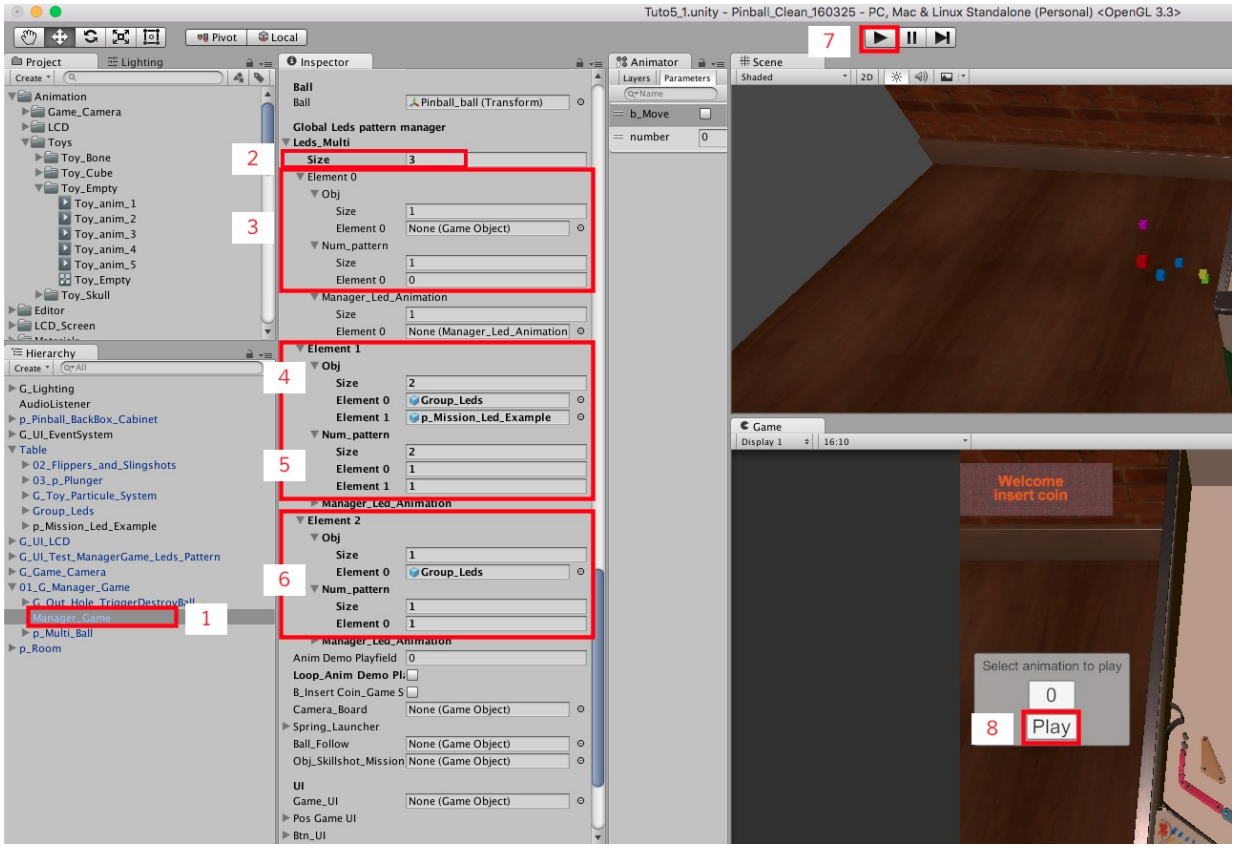
We choose to play animation 1 for Group_Leds and p_mission_Led_Example (pic 5) (Element 0 -> Num_pattern -> Element 0 and Element 1 = 1)

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).



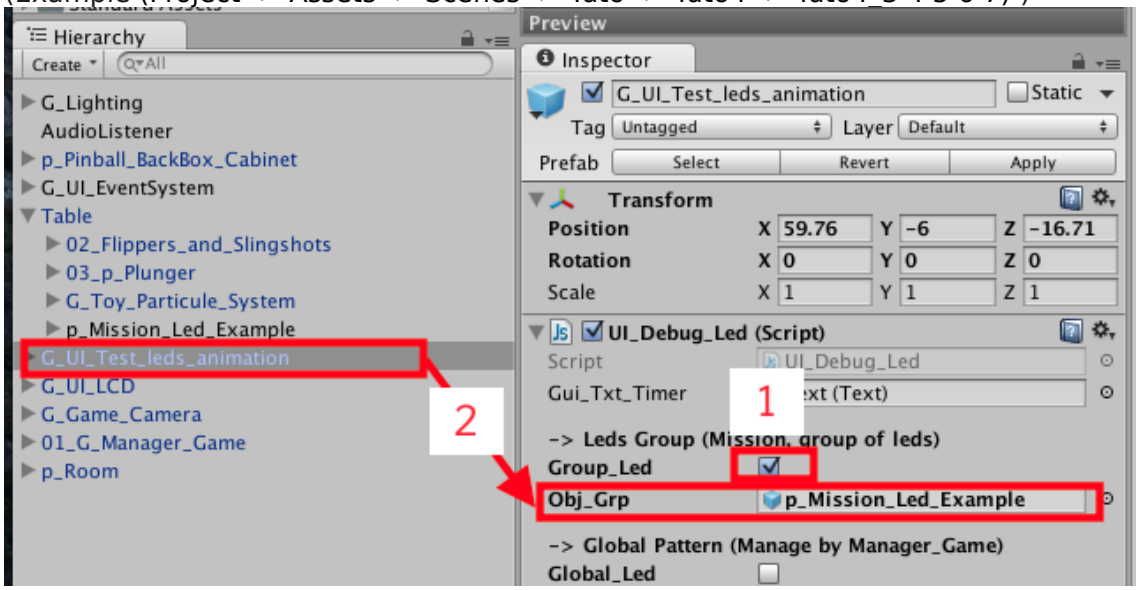
How to use G_UI_Test_Led_animation and G_UI_Test_ManagerGame_Leds_Pattern prefab.

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))

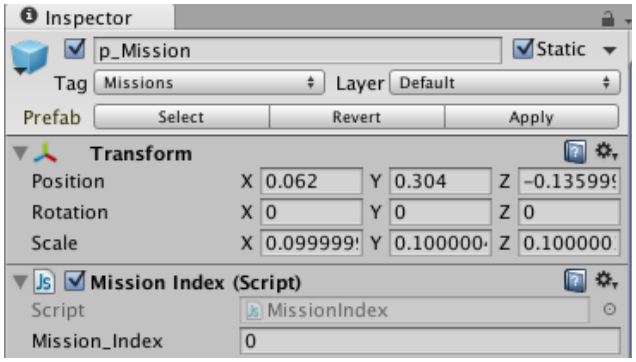


Cas 2 : **G_UI_Test_ManagerGame_Leds :**

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

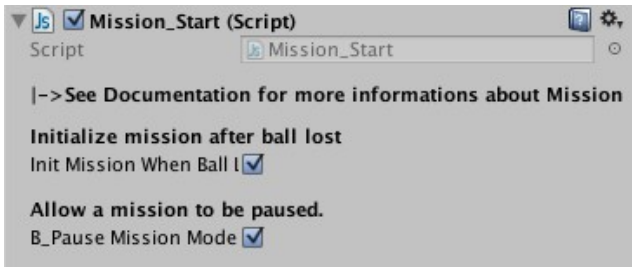
Step 1 : Drag'n'drop G_UI_Test_ManagerGame_Leds_Pattern on Hierarchy (Project -> Assets -> Prefabs -> UI -> G_UI_Test_ManagerGame_Leds_Pattern).
(Example (Project -> Assets -> Scenes -> Tuto -> Tuto5 -> Tuto5_2))

Mission :

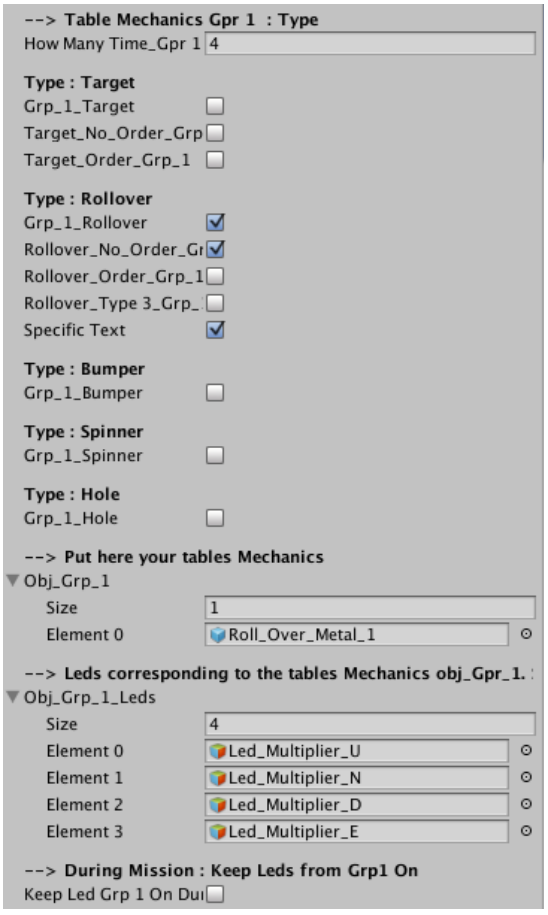


Tag : Missions

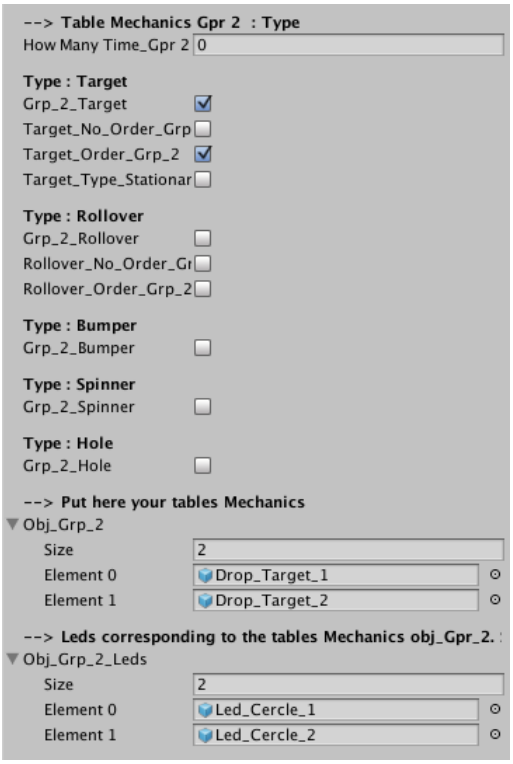
Mission_index : VERY IMPORTANT Choose a unique Index for each mission.



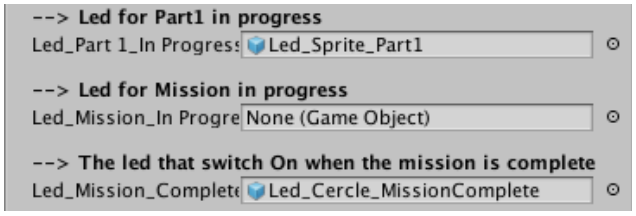
InitMissionWhenBallLost : if True the mission is init when the player lose a ball. False the mission is init only if it's the part 2 of the mission or when the player is game over.
b_PauseMissionMode : If false. Mission is not affected by the pause of other mission. And the mission couldn't pause other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



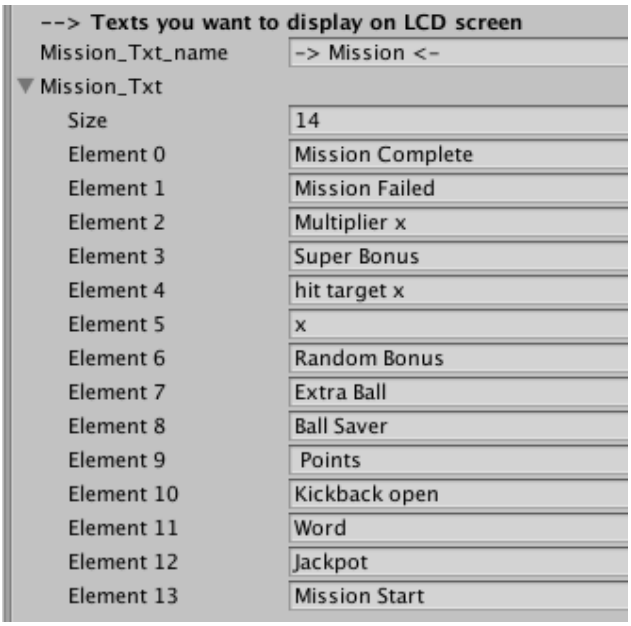
Mission Part 1
see section mission configuration to learn more about mission part 1 configuration.



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



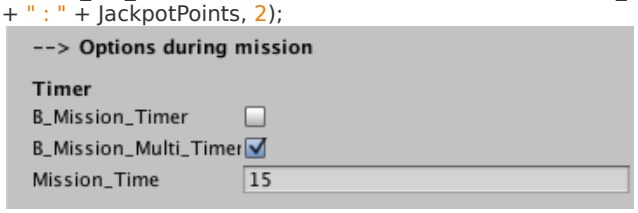
Mission_Txt_name : Mission name

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

All text combination :

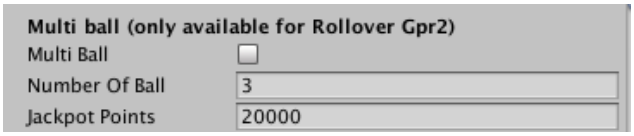
```
gameManager.Add_Info_To_Array(
Mission_Txt_name + "\ n"// text : Mission name
+ Mission_Txt[1]);// text : Mission Failed

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.

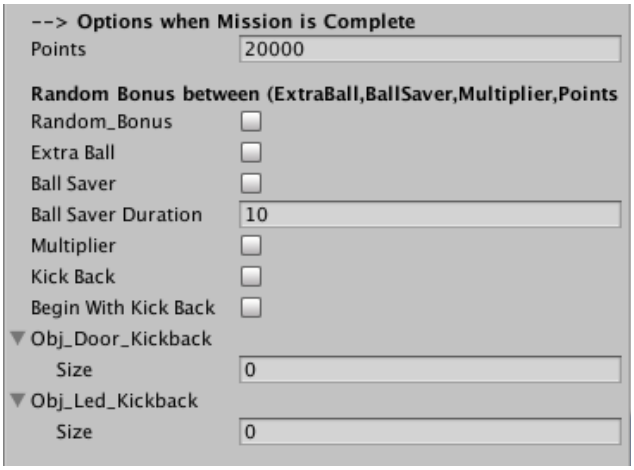
b_Mission_Timer : If true : Timer is not initialized during mission part 2.
b_Mission_Multi_Timer : If true : Timer is initialized when ball hit an object
Mission_Timer : Timer duration



Multi Ball is only available with Rollover on part 2

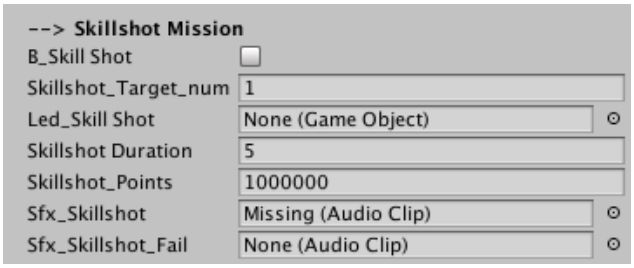
Multi-ball starts when Mission part 1 is ended and stop when there is only one ball on playfield.

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



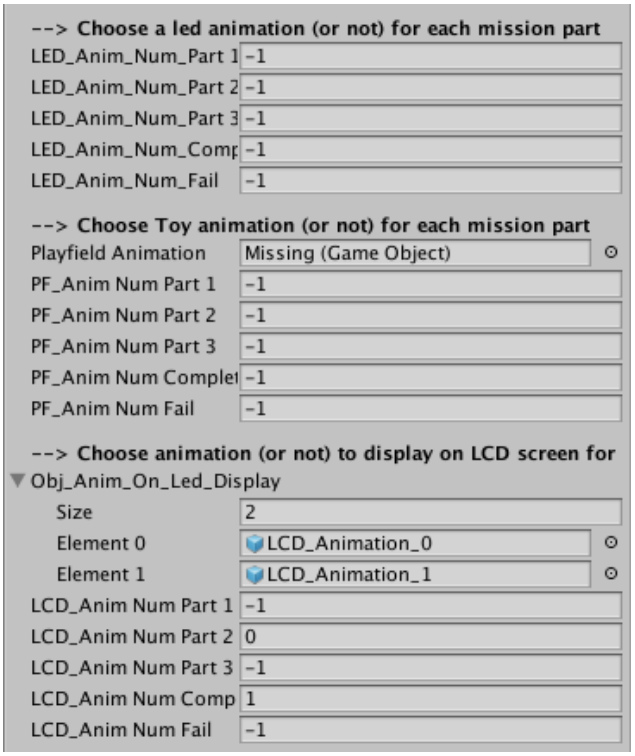
Bonus options when a mission is complete

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



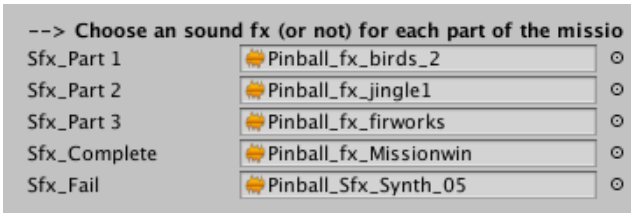
->You could play Led animations, Toy animations and LCD animations for each part of a mission.
Part 1 : Hits objects from obj_Grp_1
Part 2 : Part 1 is finish, Part 3 start.
Part 3 : Hits objects from obj_Grp_2
Part Complete : Mission is complete
Part Fail : Mission is fail.

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.

Main Menu

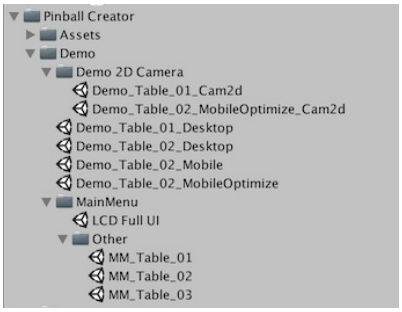
Main Menu : Overview (Read First)

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

You could use our demo scene as a starting point.

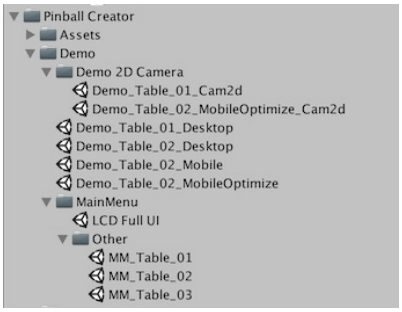
1 Open scene LCD Full UI

(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)

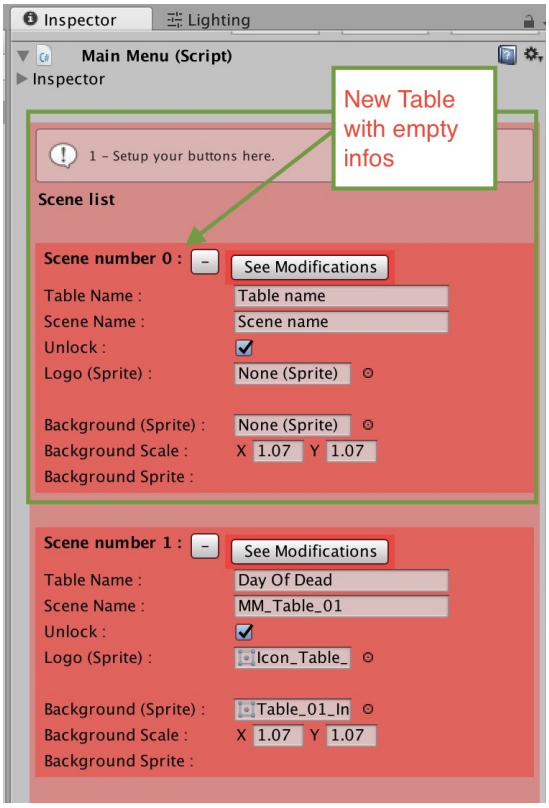


2 On the Hierarchy select the gameObject ScrollMenu_Manager

(G_Menu → ScrollMenu_Manager)

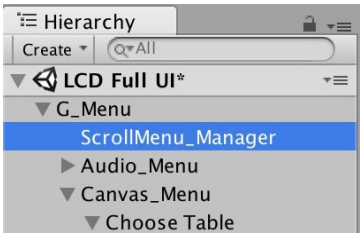
3 On the Inspector press button : Add new Table

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

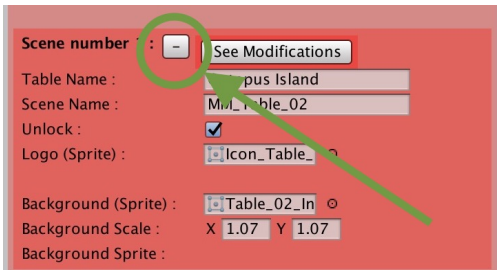


Main Menu : How to delete a table

1 On the Hierarchy select the gameObject ScrollMenu_Manager

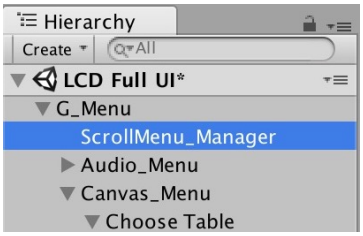


2 On the Inspector press the button - beside the table you want to delete.

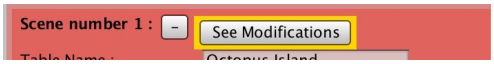


Main Menu : How to setup a table

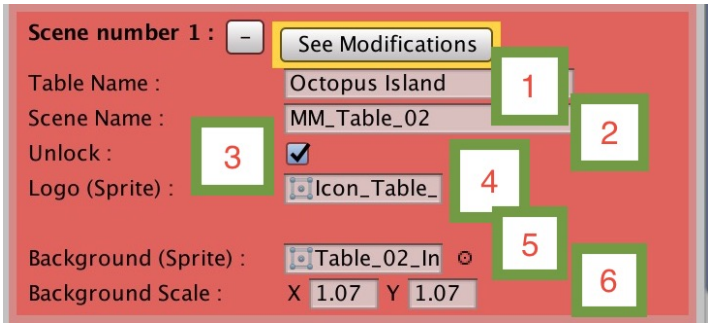
1 On the Hierarchy select the gameObject ScrollMenu_Manager



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 appears 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 starts (spot 3)

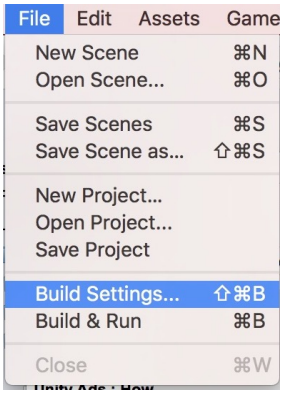
- Your icon table (Drag and drop a sprite from your project folder) (spot 4)

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

- the scale of your background sprite

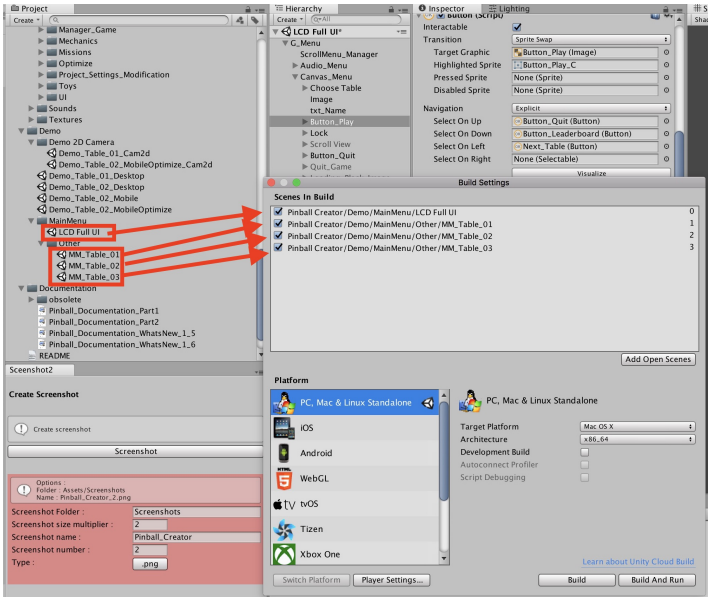
Main Menu : Add your scene to the build Menu

1 Open File → Build Settings



2 Drag and drop your scene on Scene in Build section

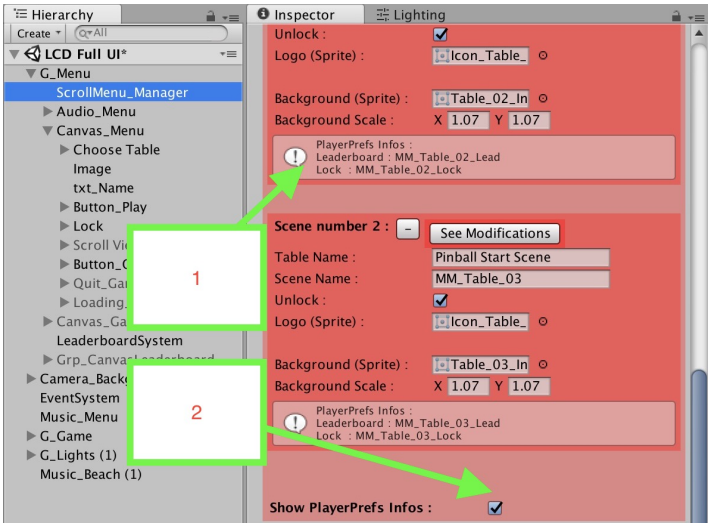
Put the scene LCD Full UI first. It the first scene that Unity need to load



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’’)

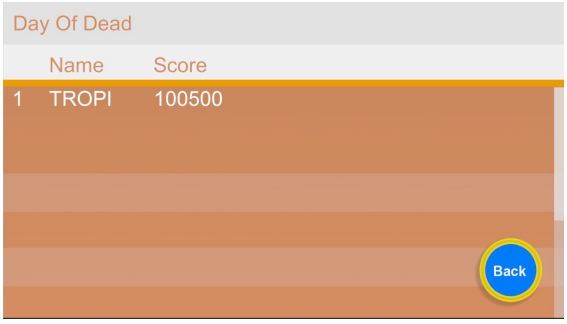
Call the PlayerPrefs to know if the table is locked : PlayerPrefs.GetString(Name of your table scene + ‘’_Lock’’)

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

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

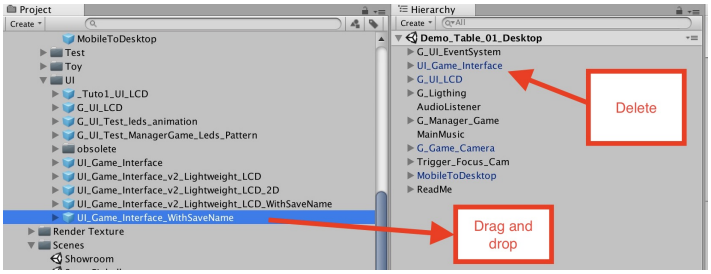


Leaderboard : How to add the prefab that allows to save name and score on a table

Case 1 : You use UI_Game_Interface gameobject on your scene

1 Delete UI_Game_Interface gameobject on your scene

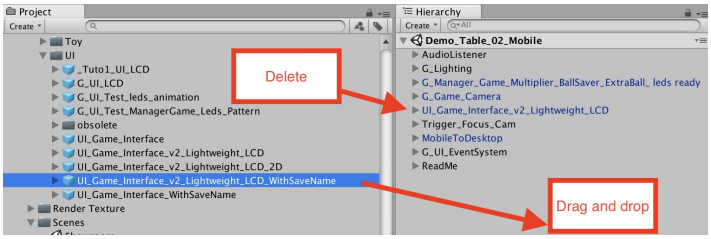
2 Drag and drop the prefab UI_Game_Interface_WithSaveName on the root o f the scene (Pinball Creator→Assets→Prefabs→UI→)



Case 2 : You use UI_Game_Interface_v2_Lightweight_LCD gameobject on your scene

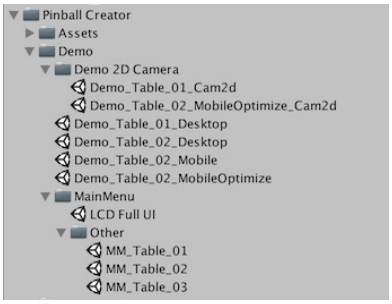
1 Delete UI_Game_Interface_v2_Lightweight_LCD gameobject on your scene

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->)

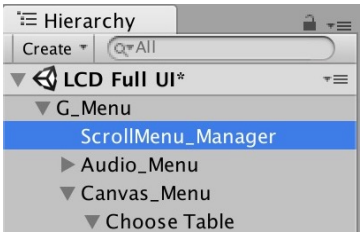


Leaderboard : Create default leaderboard for a table

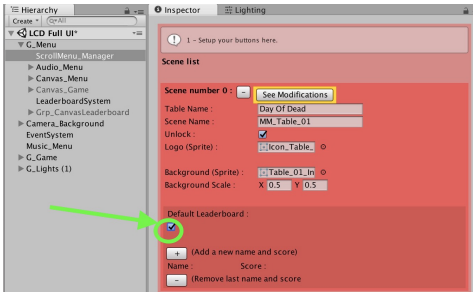
1 Open scene LCD Full UI (Project Tab → Pinball Creator → Demo → MainMenu → LCD Full UI)



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

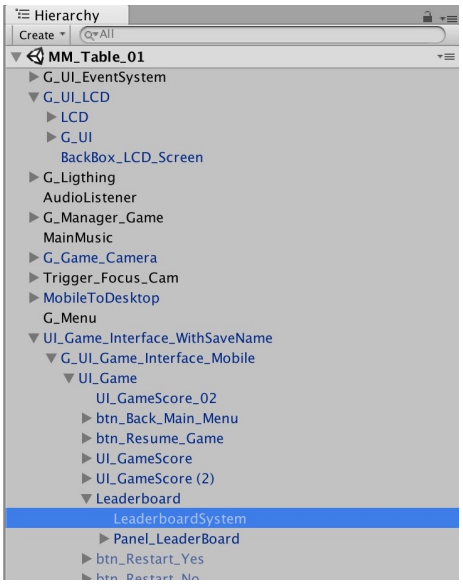
When player is game over you could

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)

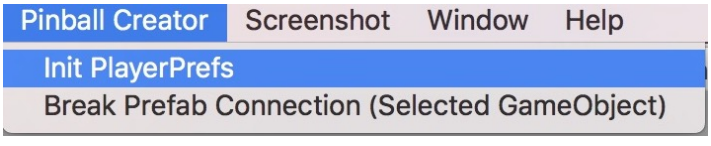



```
string PlayerName(){                        // return the player name
    return txt_PlayerName.text;
}

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.

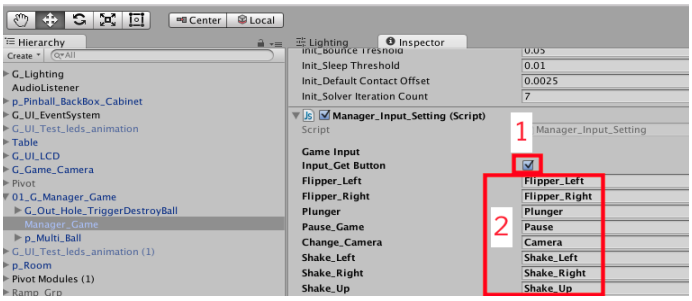
Physics Materials :

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

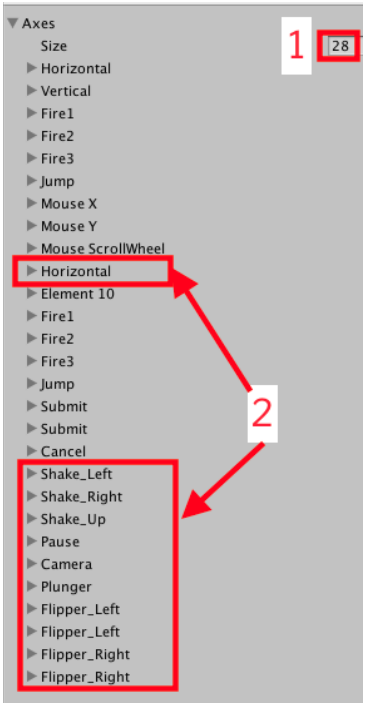
Pinball_Ball : Use only with ball.
Pinball_Flipper : Use only with flippers.
Pinball_Low_Bounce : Low bounce and low friction.
Pinball_Mid_Bounce : Mid bounce and low friction.
Pinball_Pinball_Plateau : No friction, no bounciness.

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