Top Down Shooter Characters pack

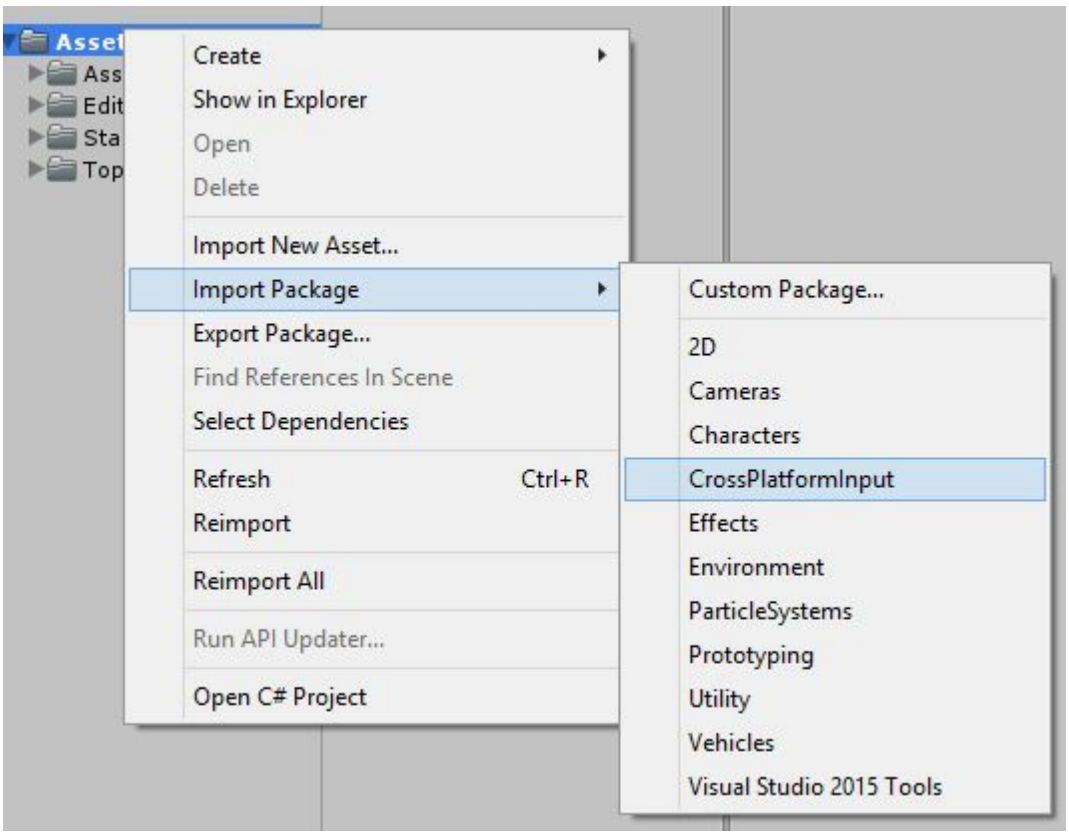
Thank you for buying this pack.

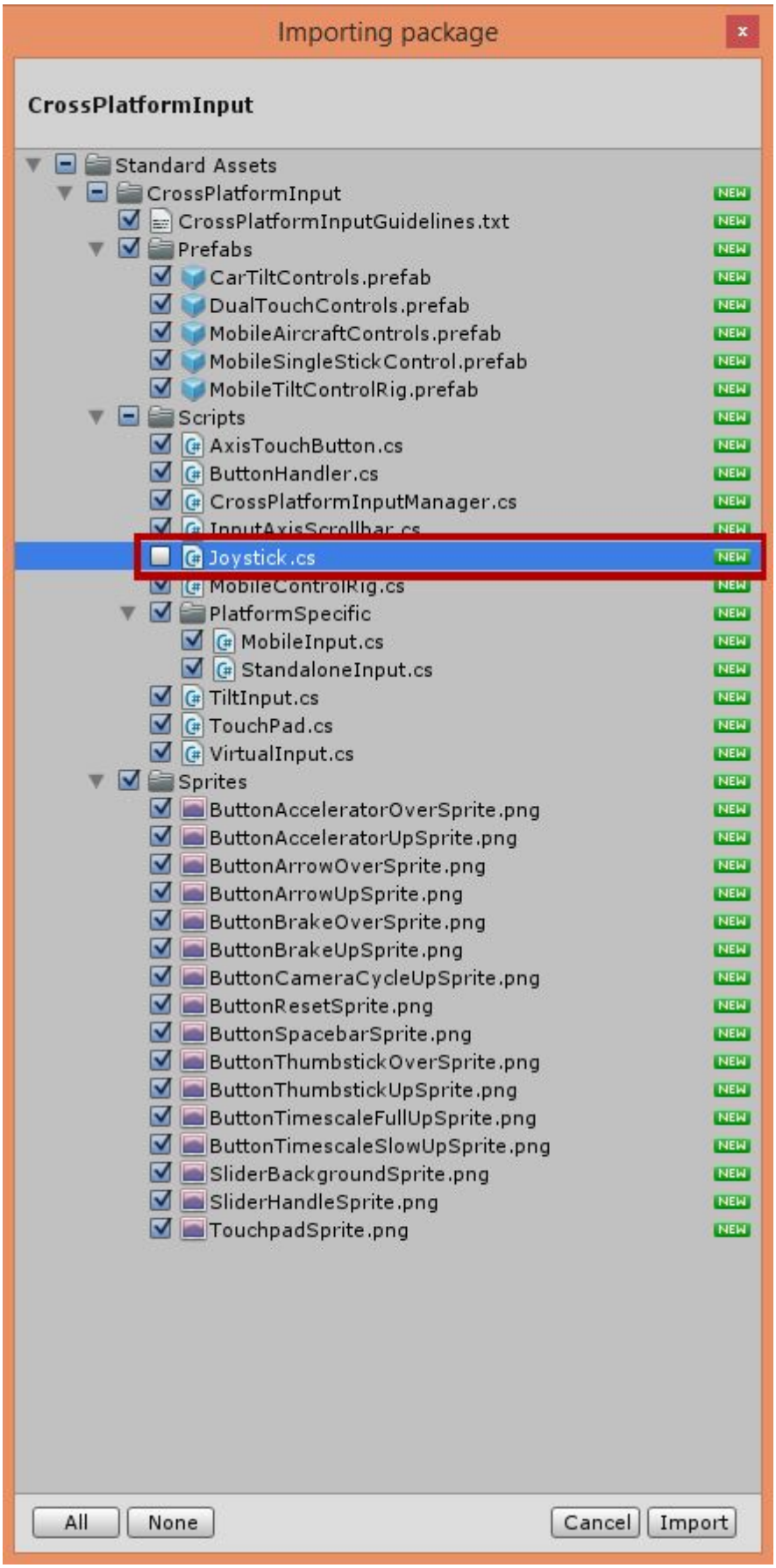
After importing the project, you'll get an error in the log:

Assets/Standard Assets/CrossPlatformInput/Scripts/Joystick.cs(27,17): error CS0246: The type or namespace name `CrossPlatformInputManager' could not be found. Are you missing a using directive or an assembly reference?

Assets/Standard Assets/CrossPlatformInput/Scripts/Joystick.cs(28,17): error CS0246: The type or namespace name `CrossPlatformInputManager' could not be found. Are you missing a using directive or an assembly reference?

To remove the error:
add **CrossPlatformInput** to your project.
mouse right click on **Assets -> Import Package -> CrossPlatformInput ->** disabling paragraphs **Joystick.cs** in Importing package menu,
according to the screenshot below.




Importing package
CrossPlatformInput
Standard Assets
CrossPlatformInput
CrossPlatformInputGuidelines.txt
Prefabs
CarTiltControls.prefab
DualTouchControls.prefab
MobileAircraftControls.prefab
MobileSingleStickControl.prefab
MobileTiltControlRig.prefab
Scripts
AxisTouchButton.cs
ButtonHandler.cs
CrossPlatformInputManager.cs
Joystick.cs
MobileControlRig.cs
PlatformSpecific
MobileInput.cs
StandaloneInput.cs
TiltInput.cs
TouchPad.cs
VirtualInput.cs
Sprites
ButtonAcceleratorOverSprite.png
ButtonAcceleratorUpSprite.png
ButtonArrowOverSprite.png
ButtonArrowUpSprite.png
ButtonBrakeOverSprite.png
ButtonBrakeUpSprite.png
ButtonCameraCycleUpSprite.png
ButtonResetSprite.png
ButtonSpacebarSprite.png
ButtonThumbstickOverSprite.png
ButtonThumbstickUpSprite.png
ButtonTimescaleFullUpSprite.png
ButtonTimescaleSlowUpSprite.png
SliderBackgroundSprite.png
SliderHandleSprite.png
TouchpadSprite.png
NEW
All
None
Cancel
Import

To use the pack, simply open demo scene, located in the “Demos” folder, and set up the project.

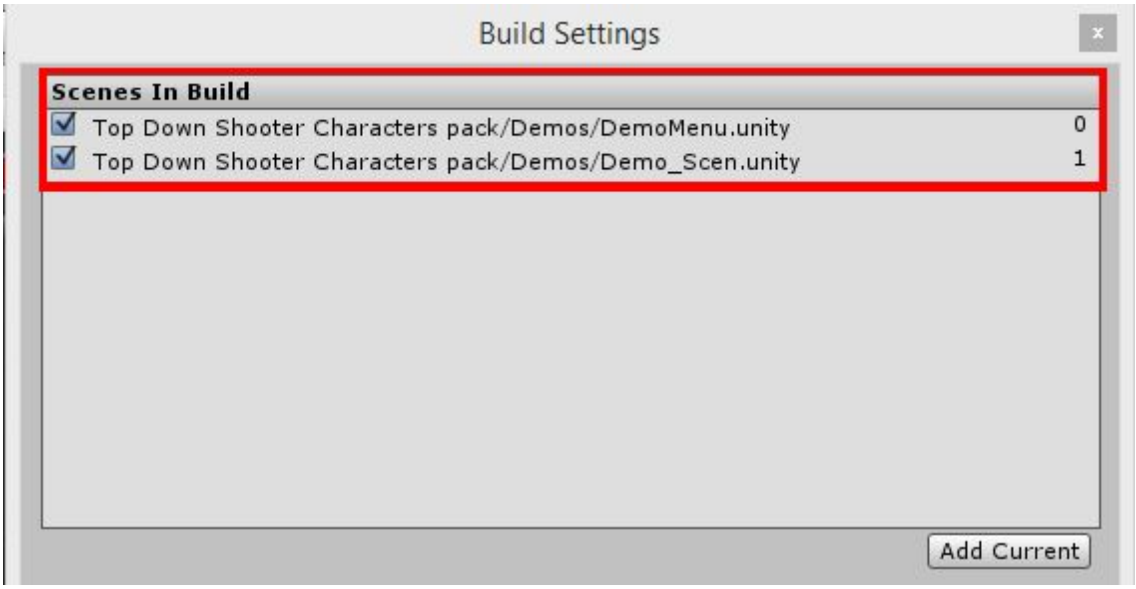
Setup

Settings must be imported from the package if it did not adjust the project to the instructions below.

1. Go to the **“File -> Build Settings”** menu,

add scenes **DemoMenu and Demo_Scen,** located in the “Demos” folder**,**

according to the screenshot below.

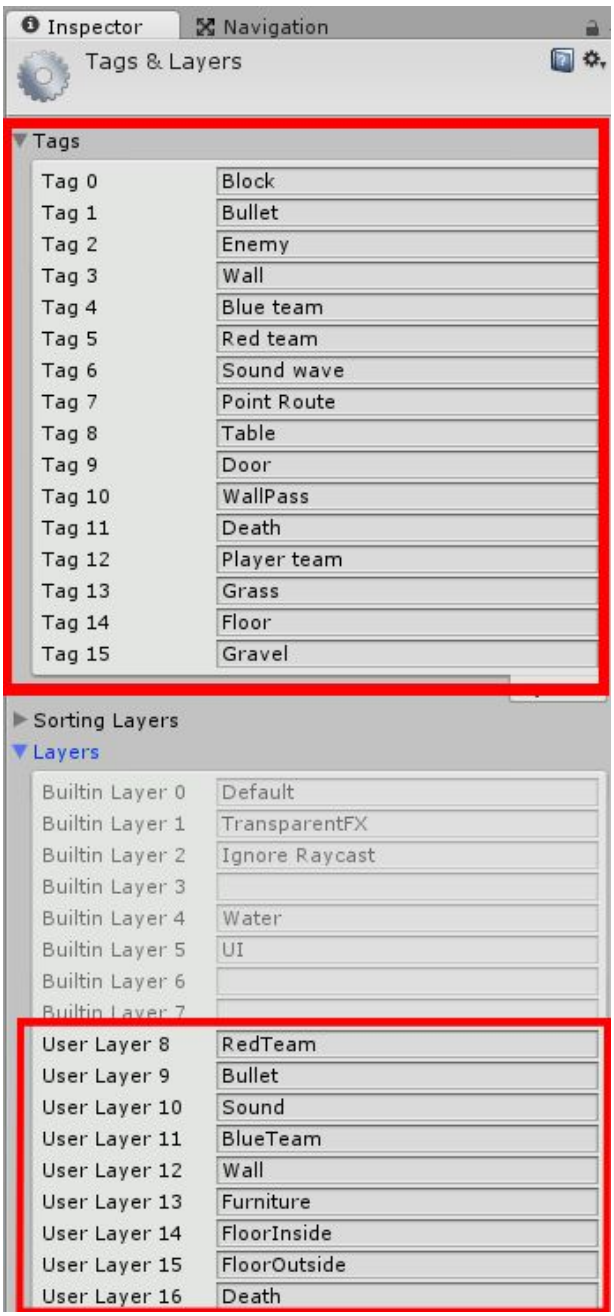


2. Go to the **“Edit -> Project Settings - > Tags and Layers”** menu,

add **Tags:**
Block, Bullet, Enemy, Wall, Blue team, Red team, Sound wave, Point Route, Table, Door, WallPass, Death, Player team, Grass, Floor, Gravel;

add **Layers:**
RedTeam, Bullet, Sound, BlueTeam, Wall, Furniture, FloorInside, FloorOutside, Death;

according to the screenshot below.

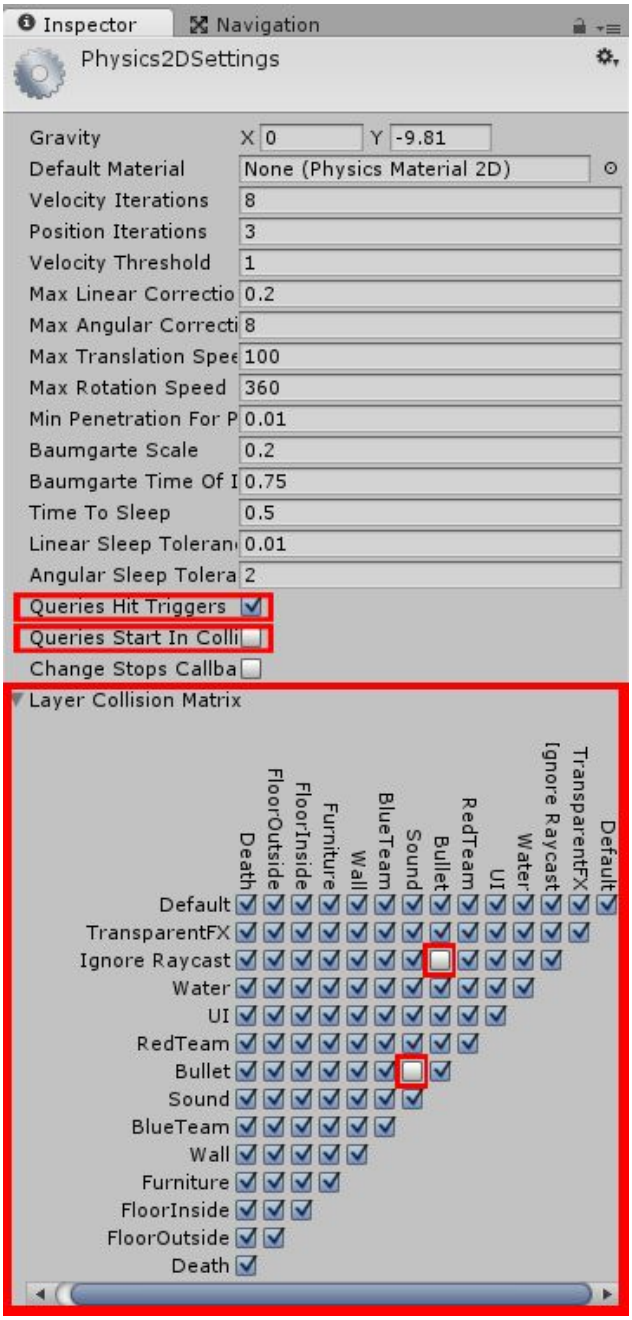


If the tags are not initialized in the prefabs, save and restart the project.

3. Go to the **“Edit -> Project Settings - > Physics 2D”** menu

set “Queries Hit Triggers” to **on**
set “Queries Start in Collider” to **off.**

Setup **“Layer Collision Matrix”** according to the screenshot below.

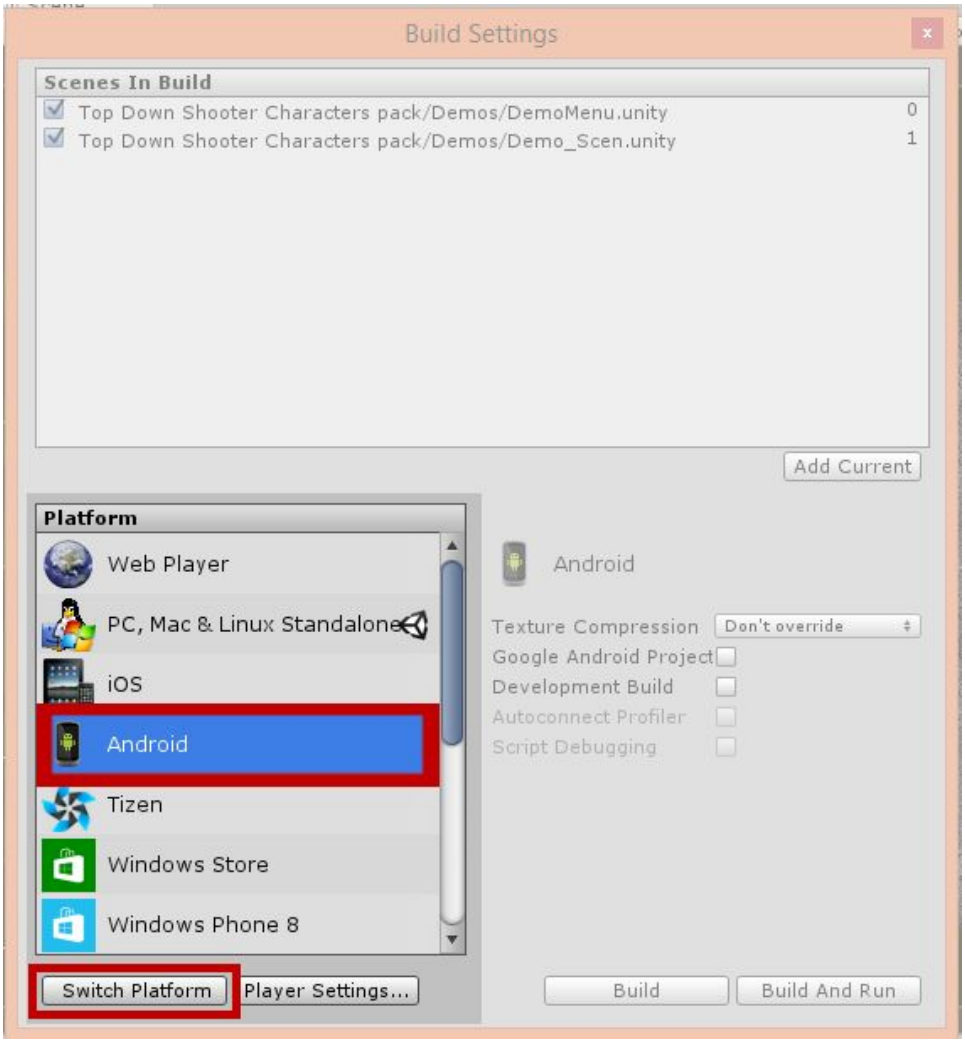


Demo Scene Setup:

For use mobile control:

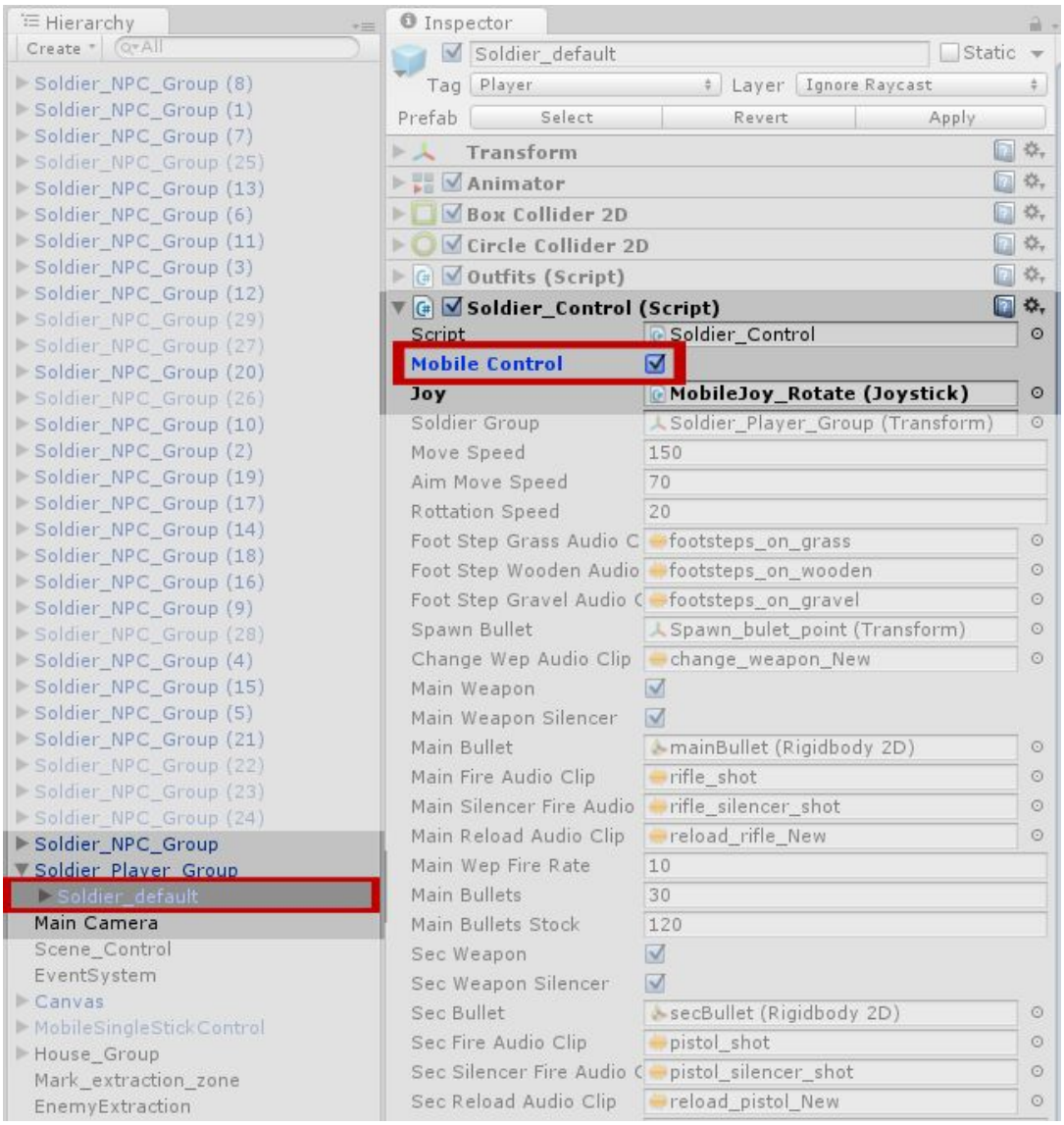
Go to the **“File -> Build Settings”** menu
Change platform Android

according to the screenshot below.



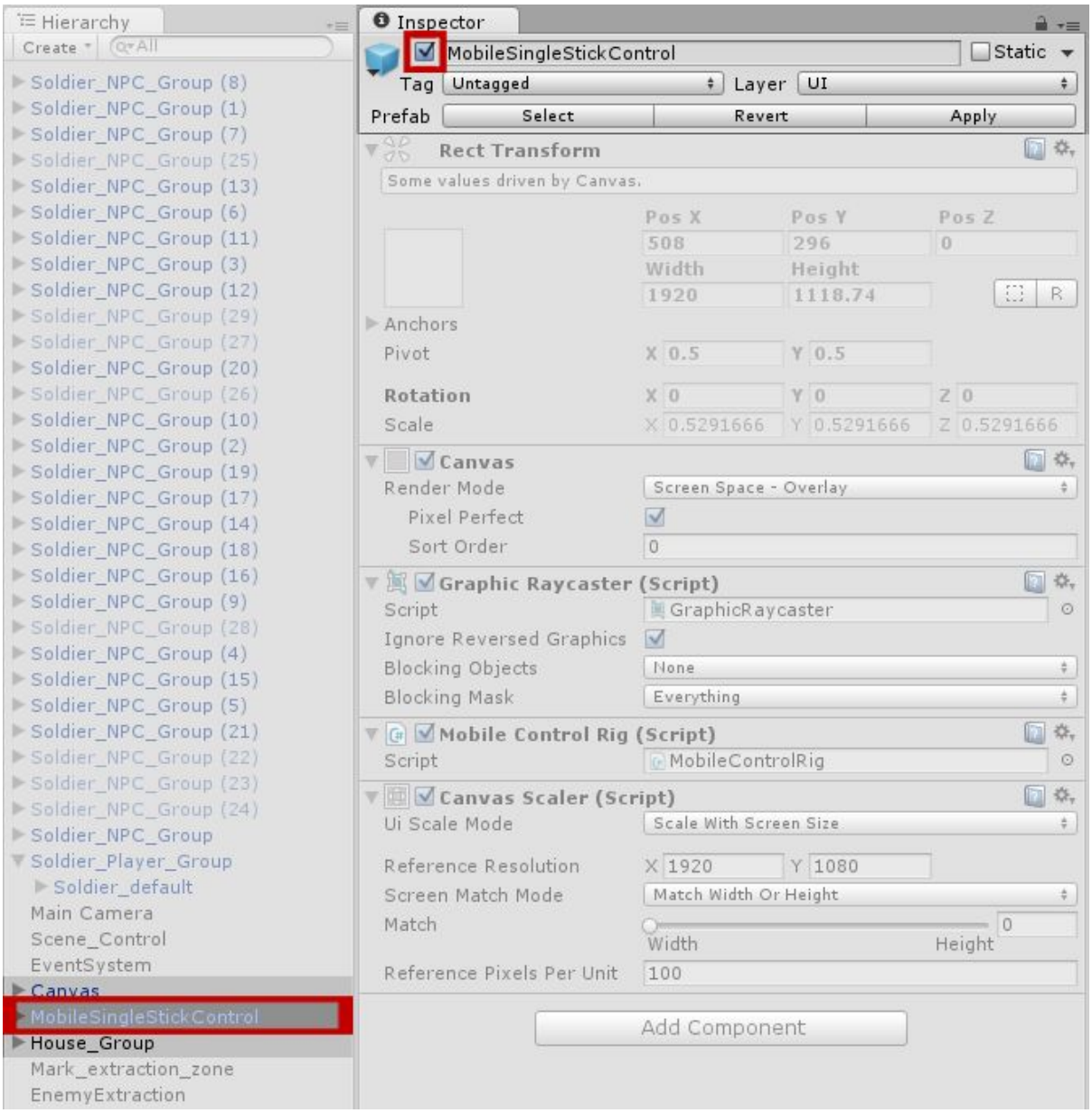
In Hierarchy **“Soldier_Player_Group -> Soldier_default -> Soldier_Control” set “Mobile Control”** to **on**

according to the screenshot below.



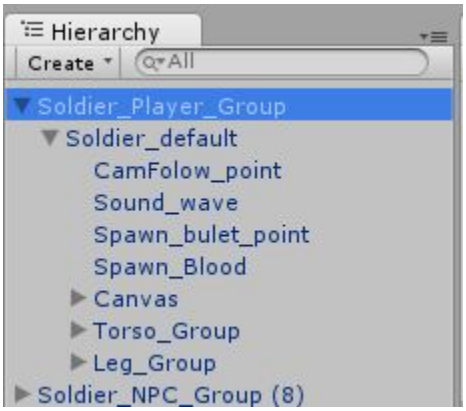
In Hierarchy **“MobileSingleStickControl”**
set “Game Object” to **on**

according to the screenshot below.



In the demo scene you can try typical usage of this pack.

Soldier_Group
GameObject and it's hierarchy.



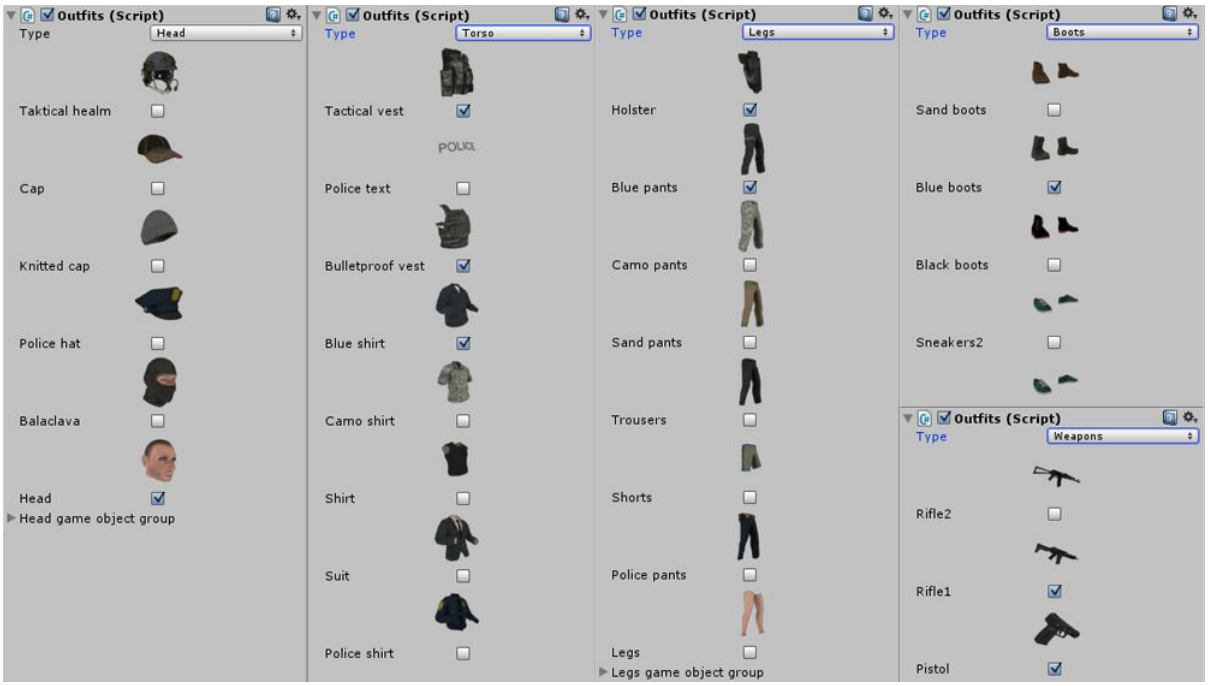
The most interesting parts are where the scripts located (see below).

Soldier's hierarchy

- **Soldier_Group** - main object.
  Rigidbody 2D & Collider 2D for physics.
    - **Soldier_default** - Animator & Collider 2D for hit bullets.
      Scripts:
      Outfits - Main outfits setup.
      Soldier_Control - Soldier setup, Player input and Movement logic.
        - **CamFolow_point** - transform point for camera.
        - **Sound_wave -** object for sound wave simulation.
        - **Spawn_bulet_point** - bullets spawn point.
        - **Torso_Group** - Group with the types of torso (optional activation in the **Outfits** script).
        - **Leg_Group** - Group with the types of legs (optional activation in the **Outfits** script).

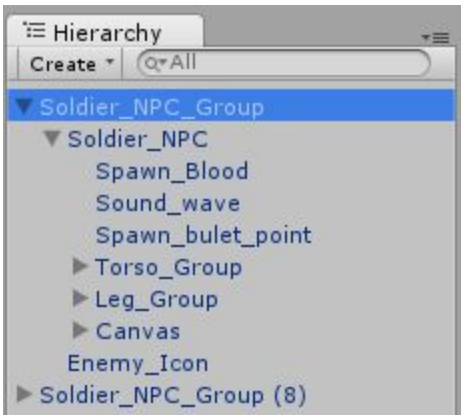
Settings:
For Settings see sctipts properties.

Under the Soldier_Group**Soldier_default**:
Outfits - Main outfits setup (all changes are visible right away in the editor mode).



Soldier_Control - Soldier setup, Player input and Movement logic.

Soldier_Enemy_Group
GameObject and it's hierarchy.



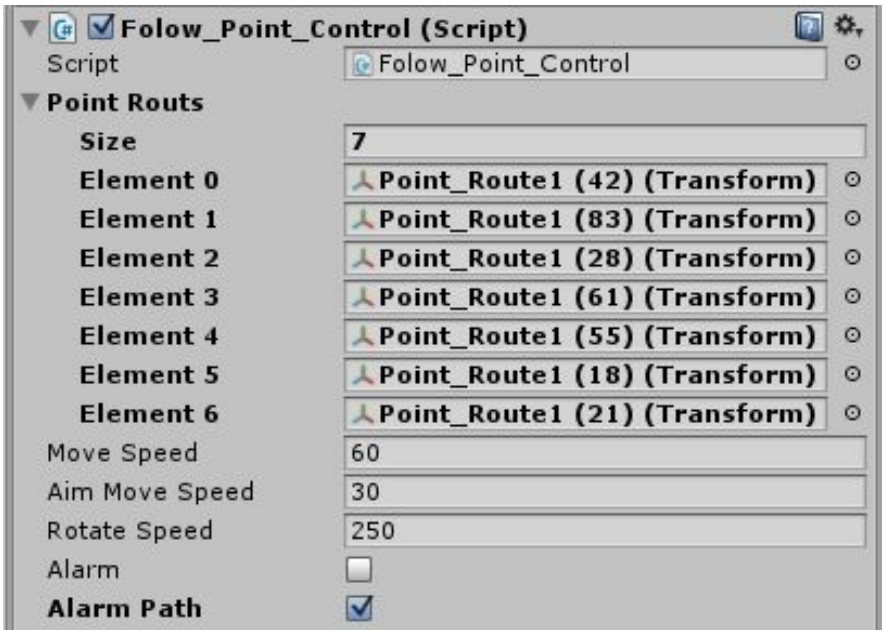
The most interesting parts are where the scripts located (see below).

Soldier's hierarchy

- **Soldier_Enemy_Group** - main object.
  Rigidbody 2D & Collider 2D for physics, and simulation field of view.
  Scripts:
  Folow_Point_Control - Movement logic.
  Enemy_FieldOfView - Field of view logic.
    - **Soldier_Enemy** - Animator & Collider 2D for hit bullets.
      Scripts:
      Outfits - Main outfits setup.
      Enemy_Control - Enemy soldier setup.
        - **Spawn_bulet_point** - spawn point for bullets.
        - **Torso_Group** - Group with the torso types (optional activation in the **Outfits** script).
        - **Leg_Group** - Group with the legs types (optional activation in the **Outfits** script).

Settings:
For Settings see sctipts properties.

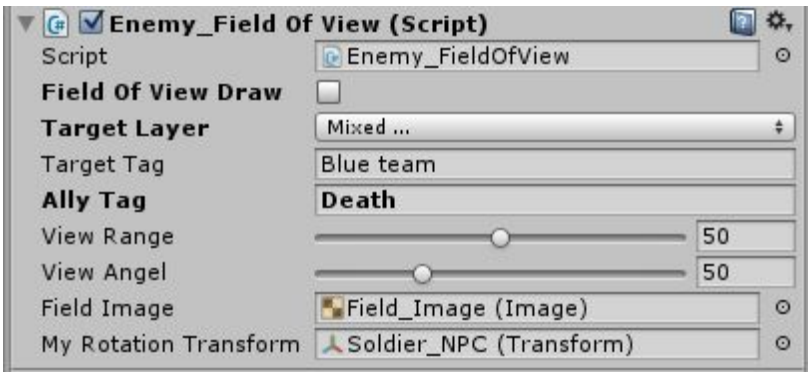
Soldier_Enemy_Group:
Folow_Point_Control - Movement logic.



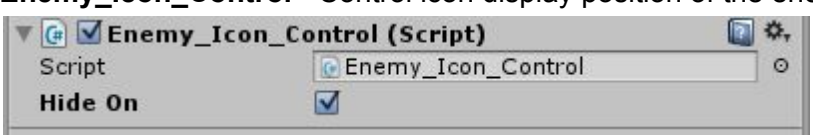
Add one point. for enemy stay.
Two and more points, for enemy patrolling area.

If you enable **Alarm Path** soldier takes position of the first point, and begins to move after the alarm.

Enemy_FieldOfView - logic field of view.



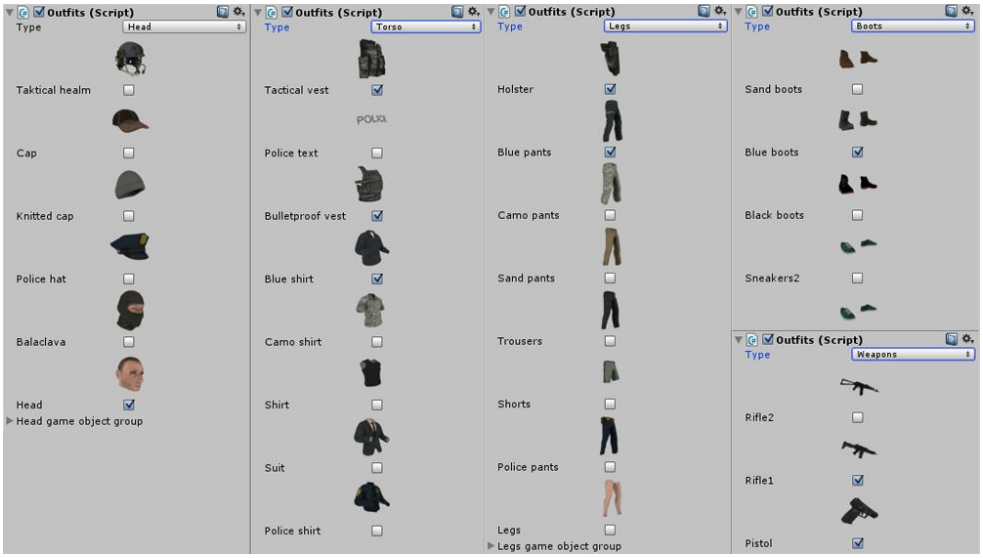
Enemy_Icon_Control - Control icon display position of the enemy behind the screen.



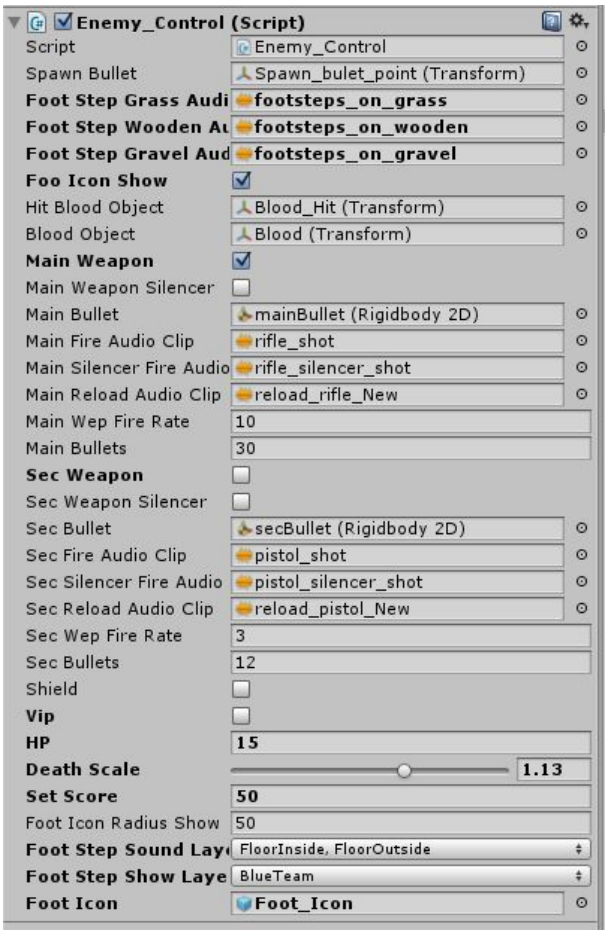
If **Hide On** enabled, the enemy will not be seen out of sight.

Under the Soldier_Enemy_Group**Soldier_Enemy**:

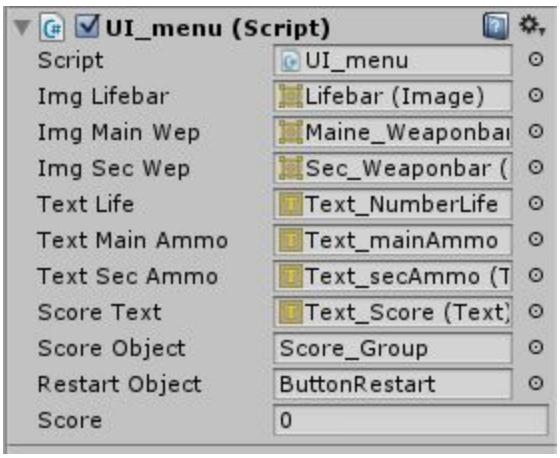
Outfits - Main outfits setup. (all changes visible right away in the editor mode)



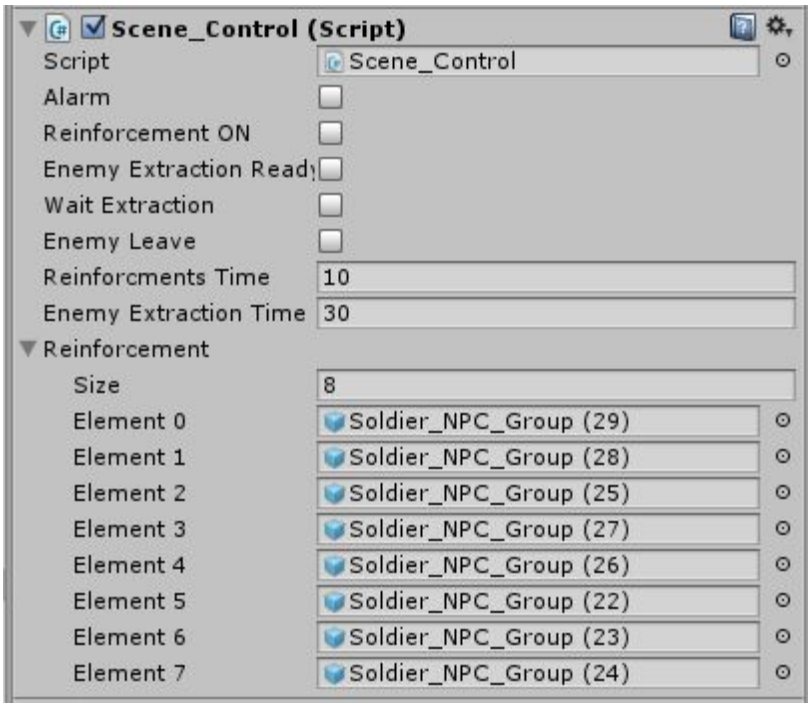
Enemy_Control - Enemy soldier setup.



Canvas GameObject in root, have settings sctipt **UI_menu** properties.

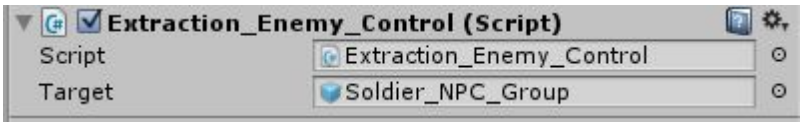


Scene_Control GameObject in root, have settings sctipt **Scene_Control** properties.

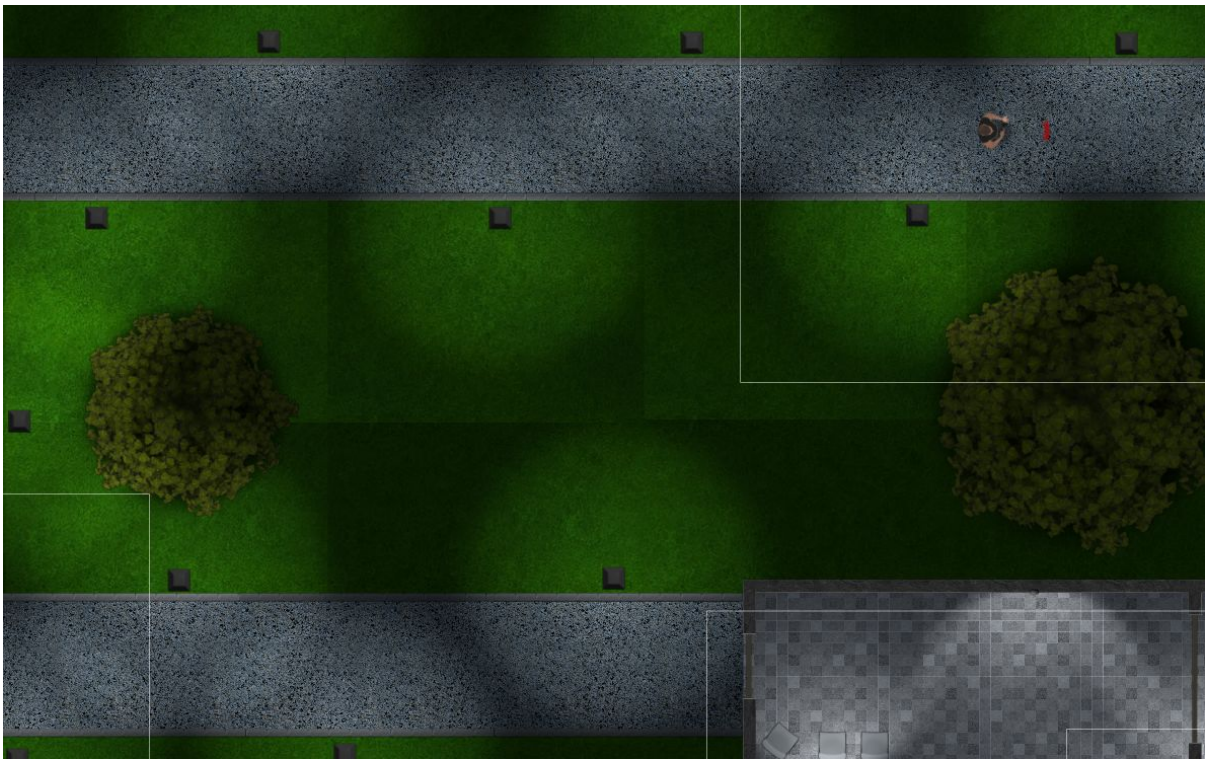


Mark_extraction_zone GameObject in root, have settings sctipt **Extraction_Zone_Control** properties.

EnemyExtraction GameObject in root, have settings sctipt **Extraction_Enemy_Control** properties.

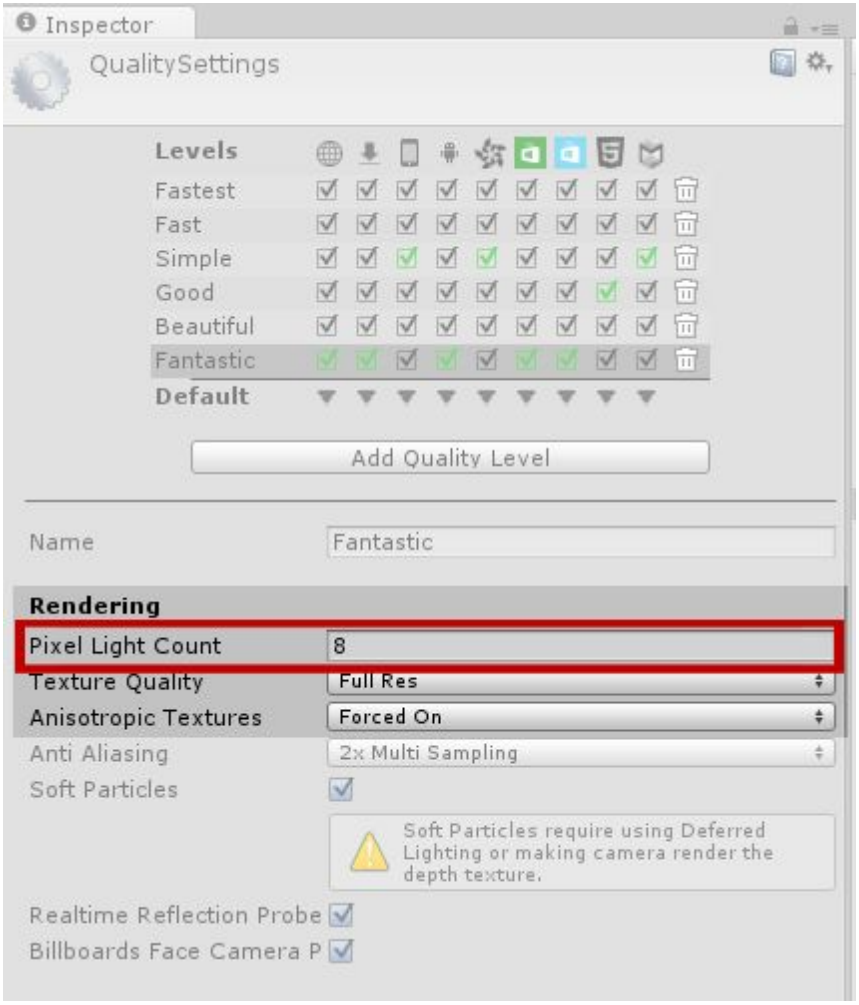


With a low number of Pixel Light Count, visible seams on sprites, like the image below:



that would remove artifacts:
1. Go to the **“Edit -> Project Settings - >Quality”** menu,
increase the value of **Pixel Light Count:**

according to the screenshot below.



You can create a new scene from existing sprites or use your own sprites, using the example settings from the demo scene.

Contact

Contact us at gears.and.brains.labs@gmail.com

We always appreciate ratings on the asset store. Thank you!