

# 2D TOPDOWN SHOOTER



DEAD PIXEL

**Support : [volkov40@hotmail.com](mailto:volkov40@hotmail.com)**

## Table of Contents

Tiles type .....	2
Editor Overview .....	3
Editor GUI .....	5
Grid.....	6
GameManager .....	7



# Tile types

**Ground** : It is the default tile where the player will be able to walk on it without any collision. The player will always be above.

**Wall** : - Player can't pass through.

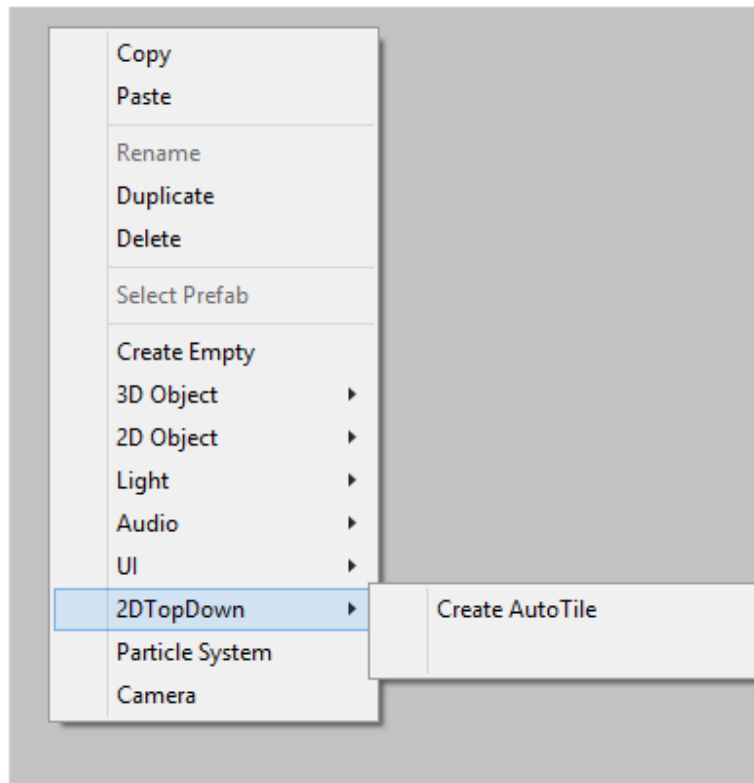
**Above** : the tile will always be above the player. Can be used on fog, cloud...

**Destructible** : this can be destroyed when shooting at it. You can put a item in it.



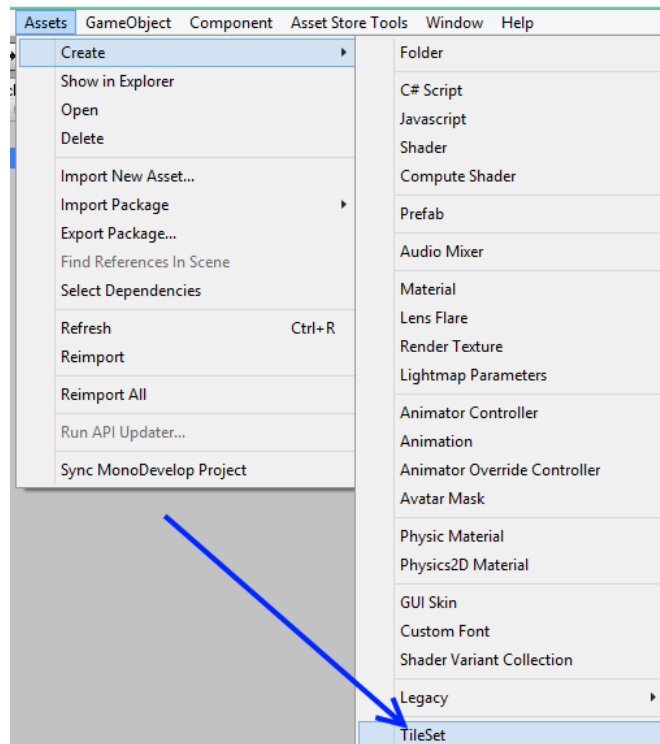
# Editor Overview

Before starting, you need to create a Grid object in your scene. To do so, right click in Hierarchy ->2DTopDown->Create AutoTile.



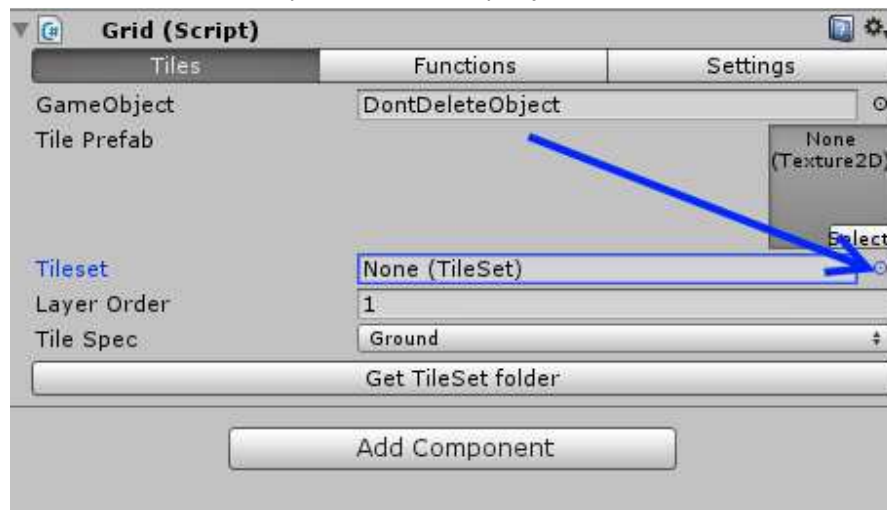
Now, you got to create a TileSet scriptable object to save your PNG which will be used as tiles

**Assets-> Create-> TileSet.**



Select the Grid object in your scene and in your inspector, you should have this.

Now click on this to link your Tileset that you just created and select it.



When it is done, now you can import all your PNGs from your folder. In this tutorial we will get the PNGs that comes with it so Click on Get TileSet folder and choose :

*Assets\2d\_TDP\Resources\PNG\Grounds*



# Editor GUI



: You can change how many tiles you want to create at once  
1x1, 2x2 or 3x3.



: Create a tile on the map



: Delete a tile on the map

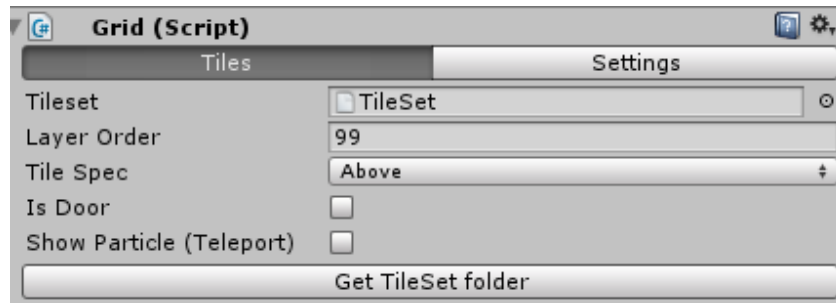


: When selected, if you click on the map, it will find all the tiles there and even  
change the inspector so you can re-use this tile with the same spec. (fast)



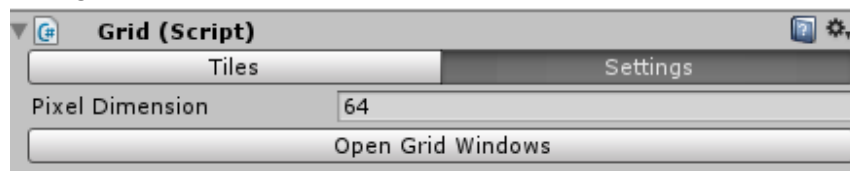
# Grid

There are 3 main section in the editor (Tiles, Functions && Settings)



- Layer Order : 0-99. Tiles with a high number will be above the other. 0 = below everything
- Tile Spec : Ground, Walls, Above

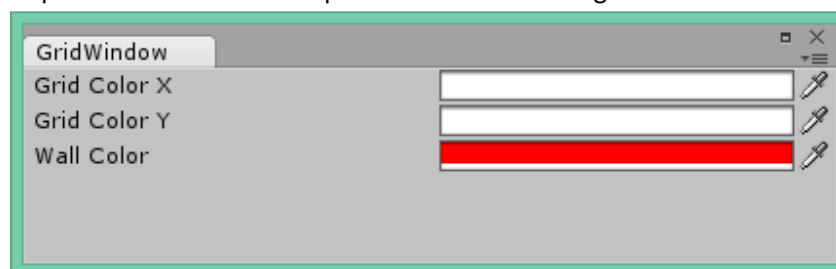
Settings



In this editor, I use my own PNGs wich are 64x64.

If your PNGs are 24x24, just change the Pixel Dimension to 24.

- Open Grid Windows will open this. You can manage the color shown in the editor here.



# GameManager

In the **GameManager** object, everything is stored here so we can instantiate the right weapon on the player or on the enemies.

The screenshot shows the GameManager Inspector with the following settings:

- V Weapon List**
  - Size: 5
  - Pistol**
    - V Name: Pistol
    - V Weapon Name: Pistol
    - V Weapon Type: Ranged
    - Is 2 Handed: ☐
    - V Weapon Icon: Weapons\_0
    - V Weapon Obj: Pistol
    - V Aim Obj: BulletPos
    - V Projectile: Bullet1
    - V Bullet Angle List**
      - V Projectile Speed: 26
      - V Time Btw Shot: 0.4
      - V Time Waited: 0
      - V Impact FX: Impact2(NoSound)
      - V Shot FX: Impact
      - V Dmg: 1
      - Rebound: 0
      - V Ammo Size: 6
      - V Ammo Cur: 0
      - V Clip Obj: clip
      - Attack Animation Used:

**vName** : weapon name showed on the Item ToolTip

**vWeaponName** : This is a enum which mean you have to manually add it in tds\_weapons.cs. Every weapon are linked on this very field so when creating enemies or player, you choose which WeaponName they will have when the game start.

**vWeaponType** : Melee : Do a melee attack animation and if there is a hostile unit nearby, this will hit it

Ranged : Create a bullet in the same direction

**Is2Handed** : Choose between 1 handed or 2handed animation on the character.

**vWeaponIcon** : Sprite which will be used at the TOP of the game to see which weapon we are currently using.



**vWeaponObj** : Create your own fabrik here. Please use any starting fabrik located in 2d\_Tds/fabrik/Weapons. This will be created in the right hand of the player.

**vProjectile** : Choose here what kind of bullet this will be used. (only for visual)

**vBulletAngleList** : If you create a list of 8, you will have to define 8 different angle which mean the weapon will shoot 8x bullets in 1 attack in 8 different angles.

**vProjectileSpeed** : How fast the bullet will go

**vTimeBtwShot** : How much time do we wait until the next shot. Automatically rifle are about 0.1.

**vImpactFX** : Will create this object when the bullet die

**vShotFX** : Will create this object when the shot is fired

**vDmg** : each bullet fired will do this damage when colliding with a hostile unit.

**Rebound** : each bullet will bounce on wall by this number. So for example, if you put 3x, the bullet will bounce on wall 3x time and the 4th time, it will die.

**vAmmoSize** : How many bullet do you have in your clip before reloading?

**vClipObj** : Will drop this obj when reloading. If empty, will do nothing.

**AttackAnimationused** : Put here your very own mechanim animation name (trigger ONLY)

