
Top-Down
Starter Kit
Documentation
Stylized World “Victoria”

Contents

Introduction

The **Top-Down Starter Kit** project is a starter kit for creating locations of the stylized world. Here is the working scene of the Top-Down RPG game with one location, which will give you a quick start in developing your unique, colorful, stylized RPG or RTS game. In this game, the main character of **Barbarian Girl Victoria** moving around the location destroys **Training Dummy**. Here you will find many objects for filling locations (*trees, plants, stones, ruins, water plants, wooden objects, various visual effects*). The project implemented a system for moving the main character (**Barbarian Girl Victoria**) based on NavMesh, a combat system with 3 attack animations and the effect of causing damage to the target, and a counter for the remaining targets.

The project was developed on the basis of **Lightweight Render Pipeline (LWRP)**, thus achieving high performance using high-quality graphics.

The Lightweight Render Pipeline (LWRP) is a prebuilt Scriptable Render Pipeline (SRP) that is optimized for delivering high graphics performance. It's scalable to mobile platforms, as well as higher-end consoles and PCs. (https://unity.com/lightweight-render-pipeline)

Project Video Review:
https://youtu.be/4mj9TJeJ2kI

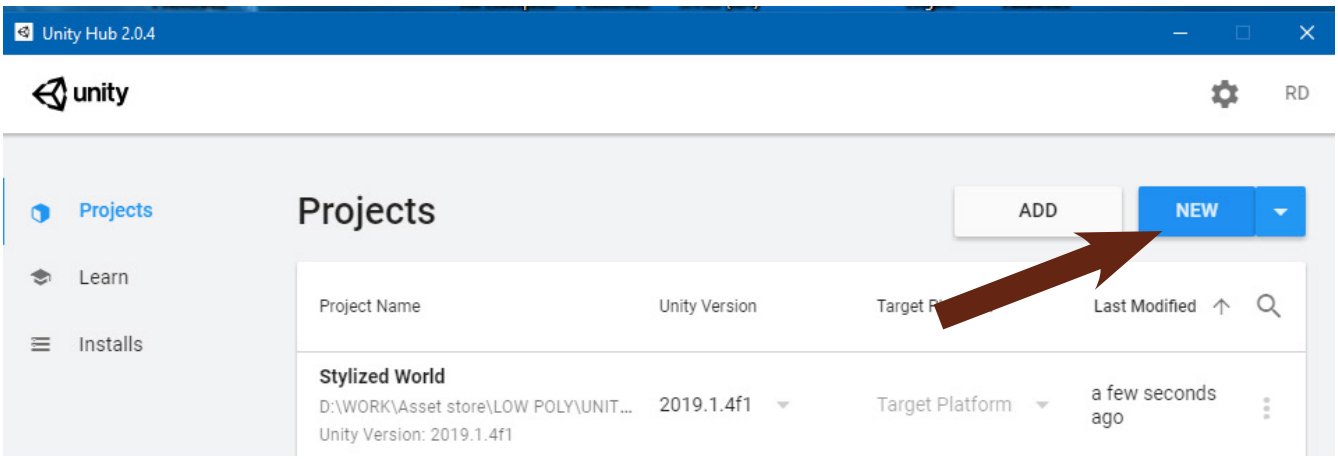
I. Install and setting Project

Project parameters:

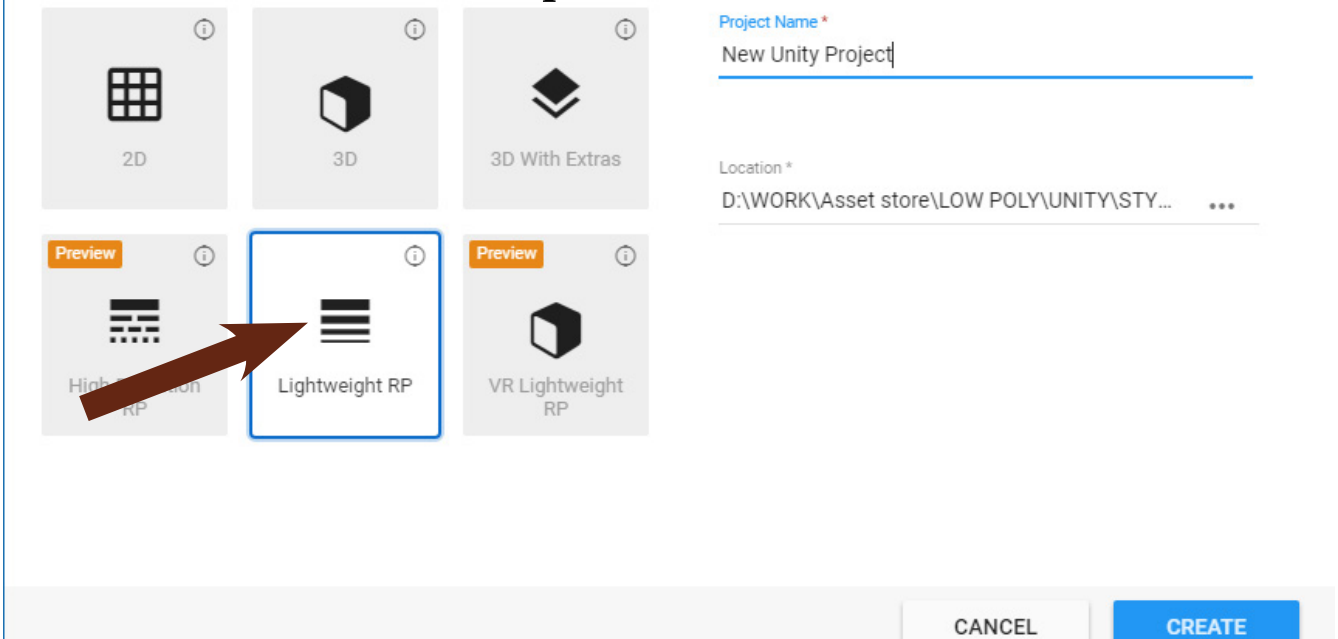
- *Lightweight Render Pipeline v.5.16.1*
- *DirectX 11*
- *Color Space – Linear*
- *Scripting Runtime Version - .NET 4.x Equivalent*

! In order for all project functions to work properly please follow the instructions for installing and setting up the project:

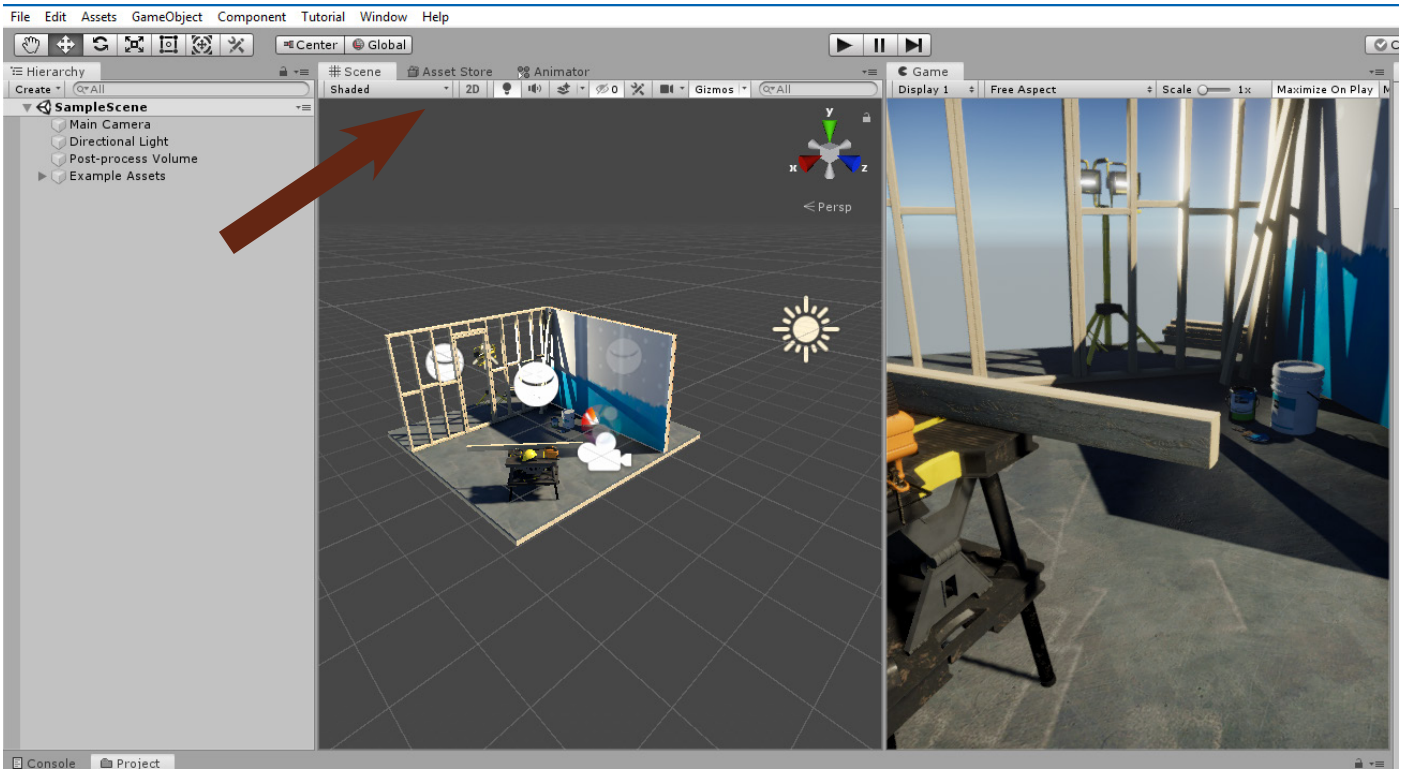
1) Launch Unity and create a new project.



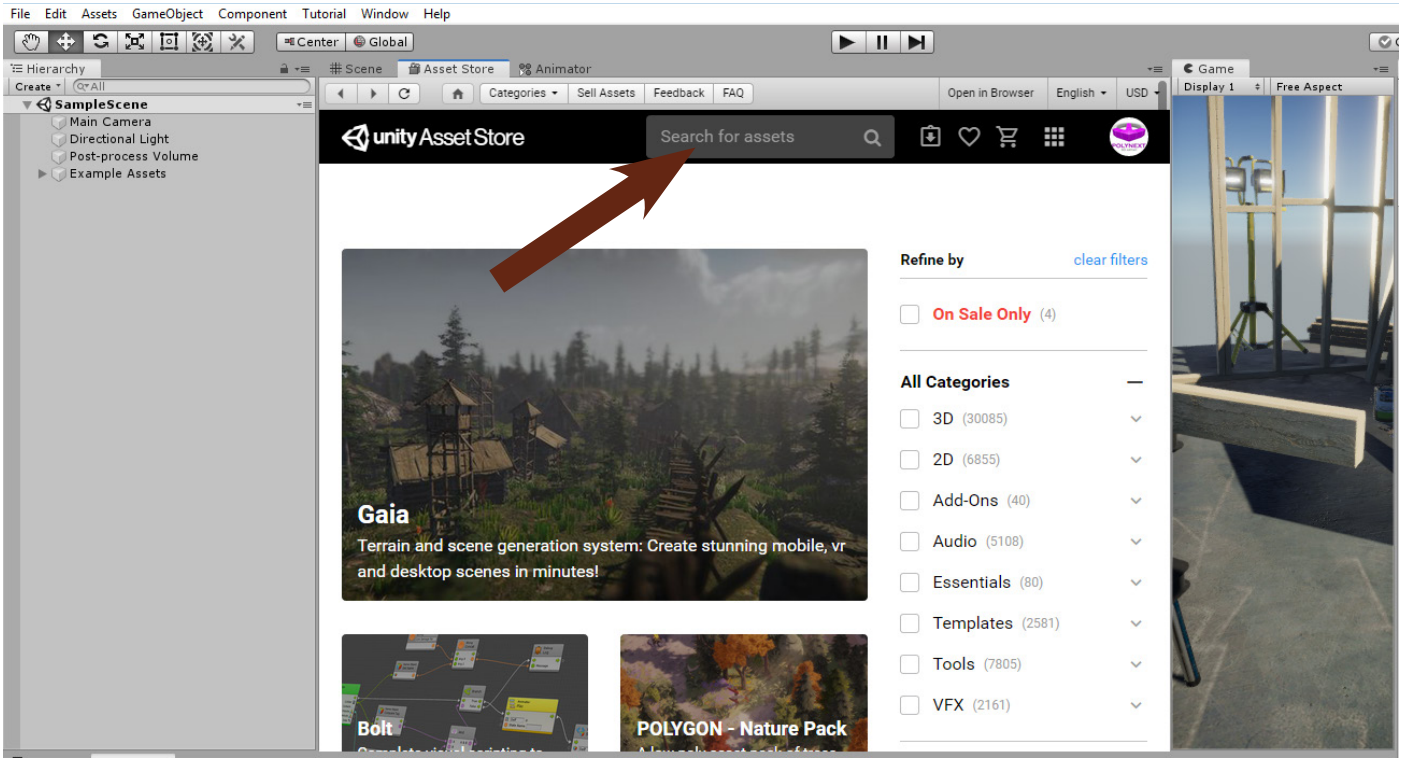
2) In the next window, select the project type «**Lightweight RP**», enter the name, the path and click «CREATE».



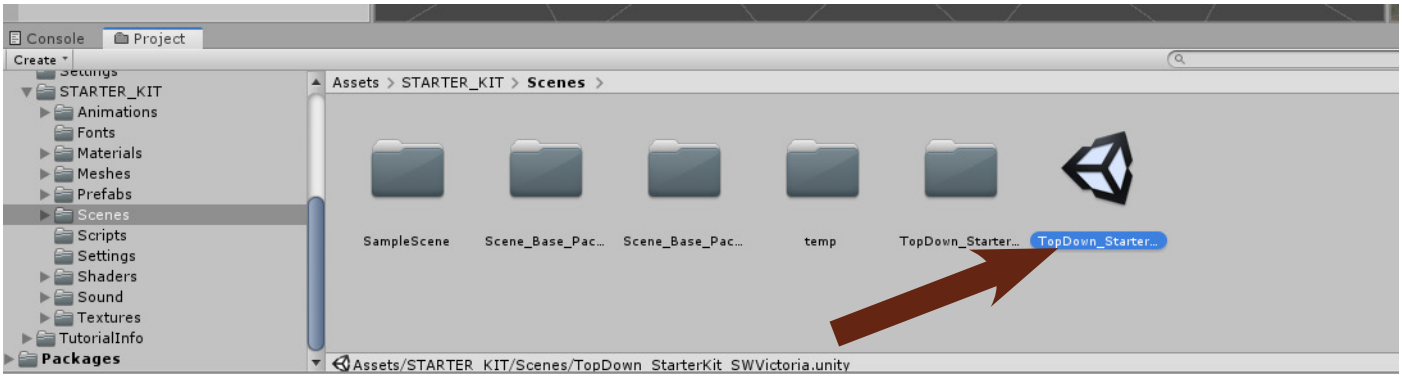
3) Wait for the download of the standard project «**Lightweight RP**» and select the **Asset Store** tab.



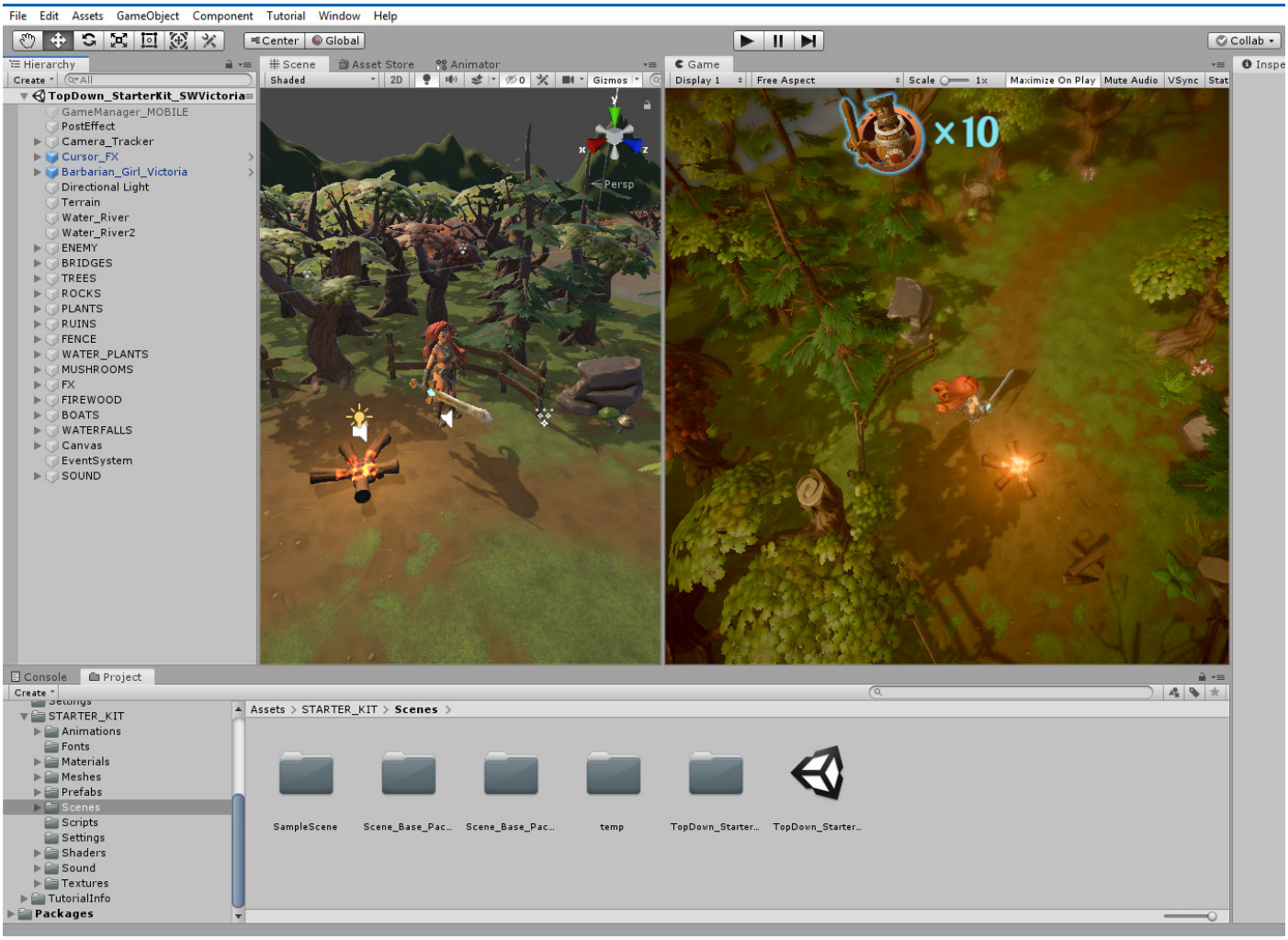
4) Enter the project name «**Top-Down Starter Kit [SW Victoria]**» in the search and click «**Download**»



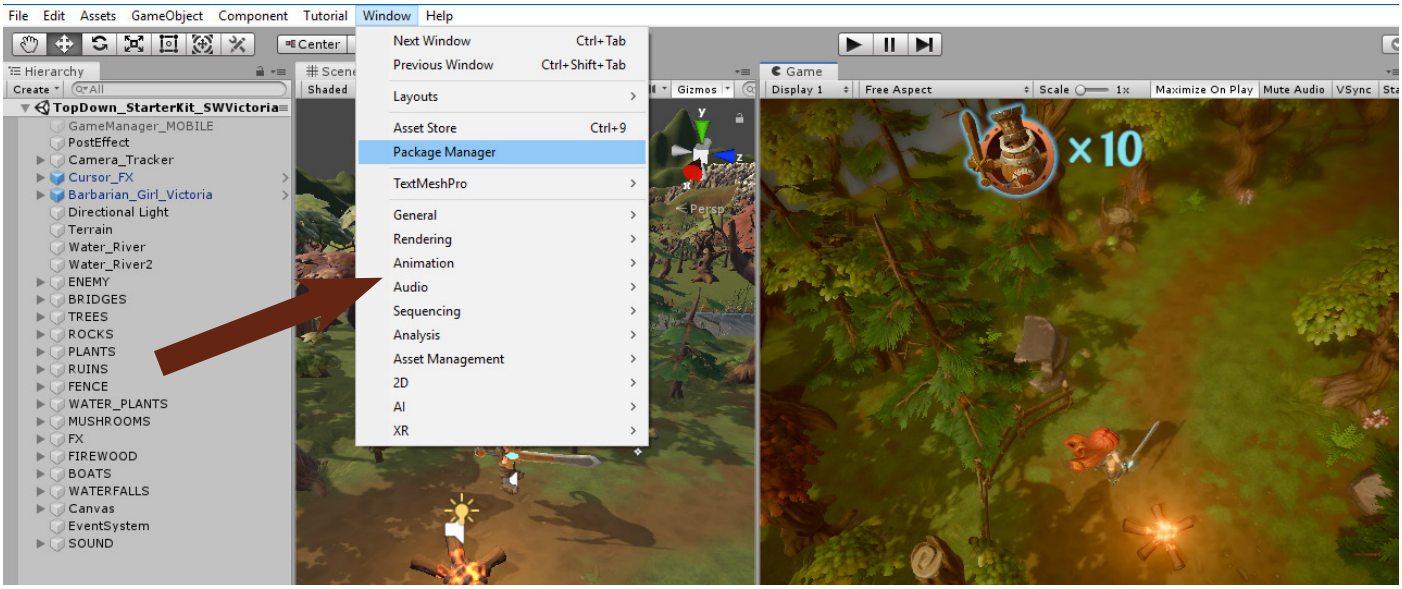
5) Open the project scene (**STARTER_KIT->Scenes-> TopDown_StarterKit_SWVictoria**)



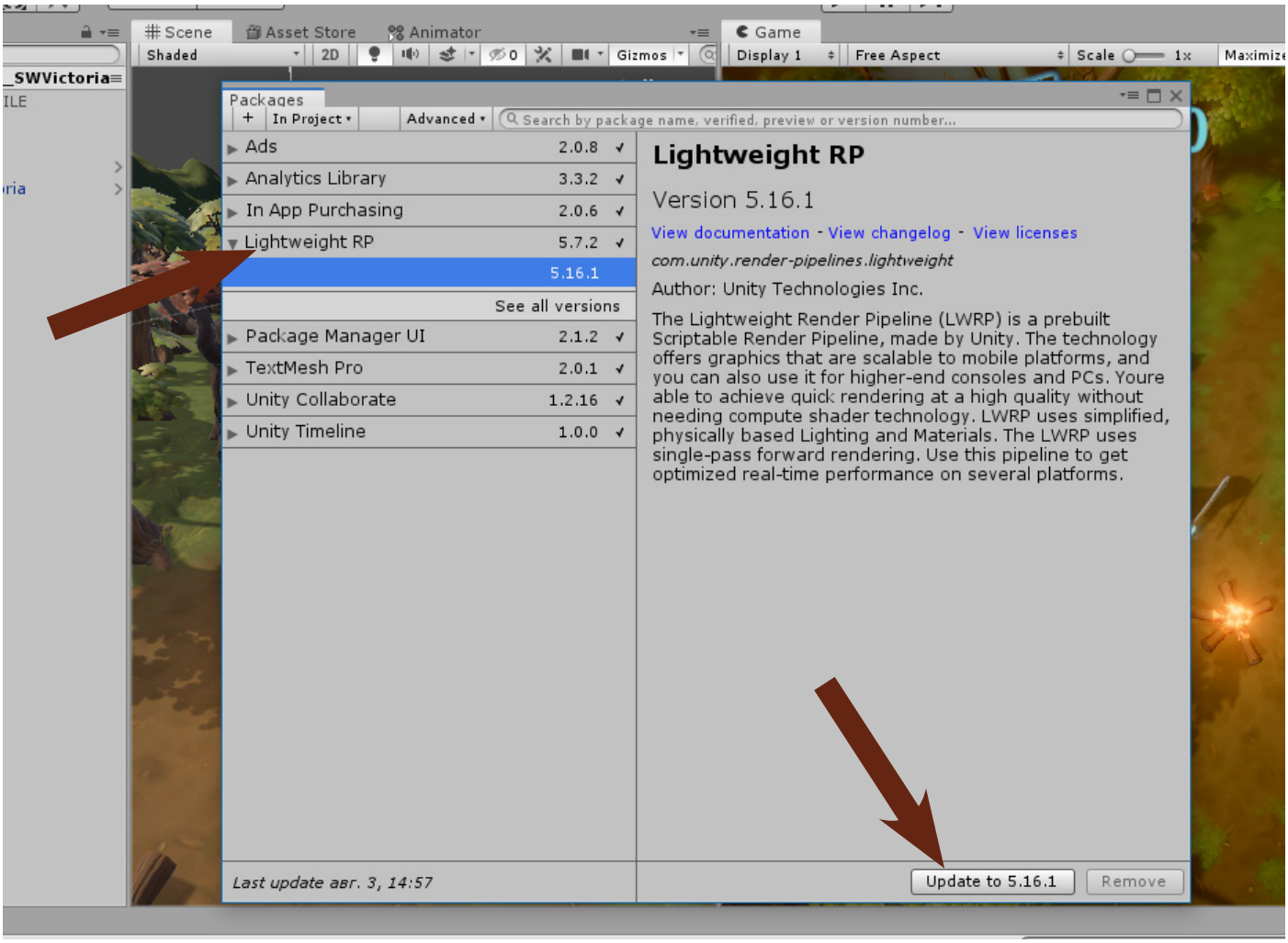
6) The scene will open, but the function of the ghostly character will not work yet! To fix, you must update the version of LWRP.



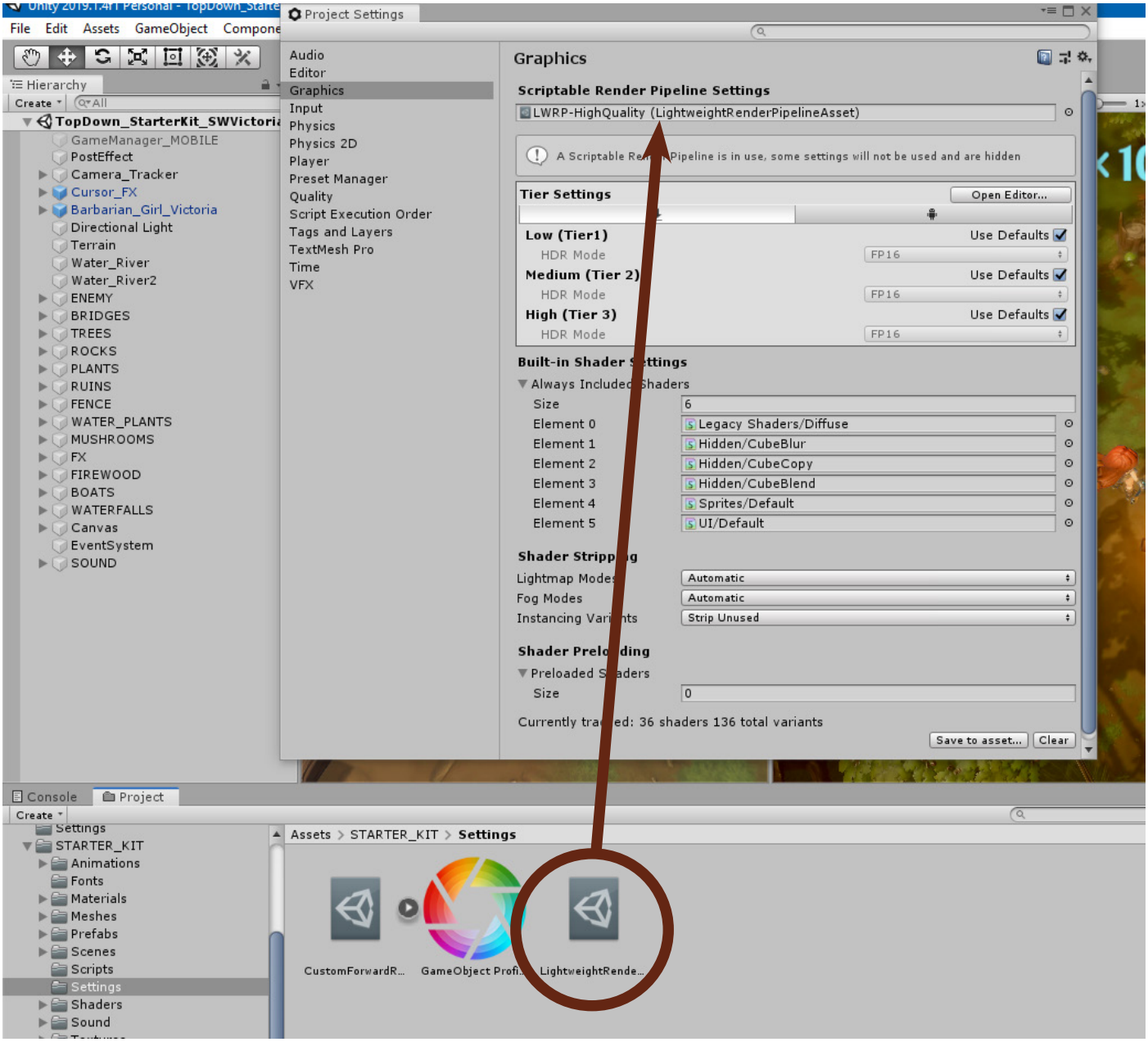
7) Open the Package Manager (Window-> Package Manager)



8) Expand the list of «Lightweight RP», select the latest version (at the time of writing 5.16.1) and click Update to 5.16.1



9) Open «**Project Settings**» (Edit-> Project Settings). Go to the «**Graphics**» section and drag the file «**LightweightRenderPipelineAsset**» from the «**Settings**» project folder into the «**Scriptable Render Pipeline Settings**» field.



10) Now all project functions are working properly and you can use it to create high-quality games.

II. Project Content

1. Trees

- All textures resolution 2048

- All textures type (Albedo, Normal, MetallicSmoothness, AO)

-2 variants material leaves (red, green) on every tree with foliage

- *Snag1* (Verts- **2294**, Tris -**2366**,)
- *Snag2* (Verts- **3096**, Tris -**2934**)
- *Stump1_1* (Verts- **1585**, Tris -**1633**)
- *Stump1_2* (Verts- **1199**, Tris -**1953**)
- *Tree_Oak1_1* (Verts- **4548**, Tris -**4608**)
- *Tree_Oak1_2* (Verts- **4288**, Tris -**4432**)
- *Tree_Oak1_3* (Verts- **4548**, Tris -**4608**)
- *Tree_Oak2_1*(Verts- **6056**, Tris -**5838**)
- *Tree_Oak2_2* (Verts- **5965**, Tris -**5550**)
- *Tree_Pine1* (Verts- **1519**, Tris -**1860**)
- *Tree_Pine2* (Verts- **1586**, Tris -**1912**)
- *Tree_Pine3* (Verts- **894**, Tris -**1140**)

2. Plants

- All textures resolution 2048

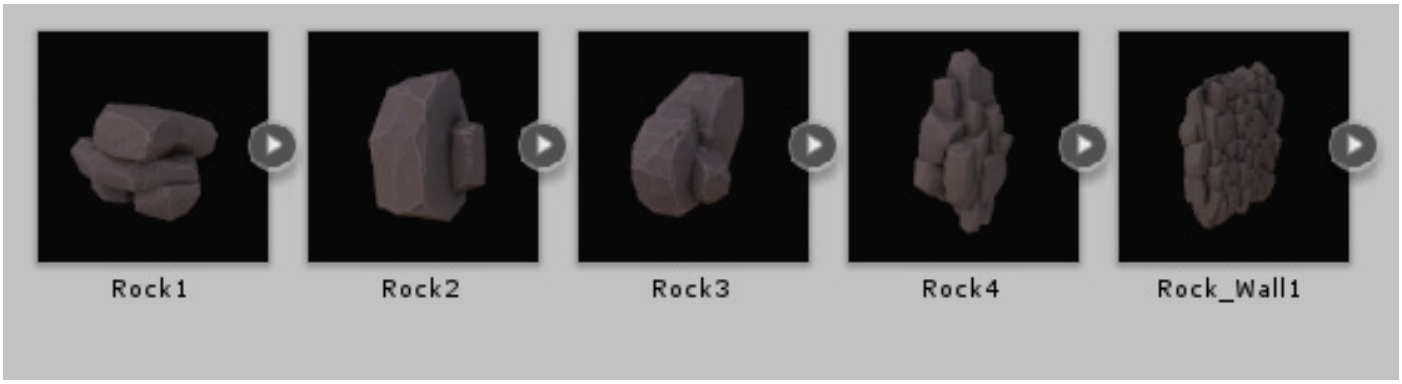
- Grass_Bush Textures type (Albedo_transparency), to other objects textures type (Albedo, Normal, MetallicSmoothness, AO)

- *Grass_Bush* (Verts- **72**, Tris -**64**)
- *Grass_Bush2* (Verts- **72**, Tris -**64**)
- *Grass_Bush3* (Verts- **135**, Tris -**120**)
- *Mushroom1* (Verts- **215**, Tris -**304**)
- *Mushroom2* (Verts- **215**, Tris -**304**)
- *Mushroom3* (Verts- **215**, Tris -**304**)
- *Mushroom4* (Verts- **98**, Tris -**160**)
- *Mushroom5* (Verts- **98**, Tris -**160**)
- *Mushroom6* (Verts- **98**, Tris -**160**)
- *Mushroom7* (Verts- **188**, Tris -**288**)
- *Mushroom8* (Verts- **188**, Tris -**288**)
- *Mushroom9* (Verts- **188**, Tris -**288**)
- *Plant* (Verts- **110**, Tris -**150**)
- *Plant_Fern* (Verts- **240**, Tris -**320**)

3. Rocks

- All textures resolution 2048

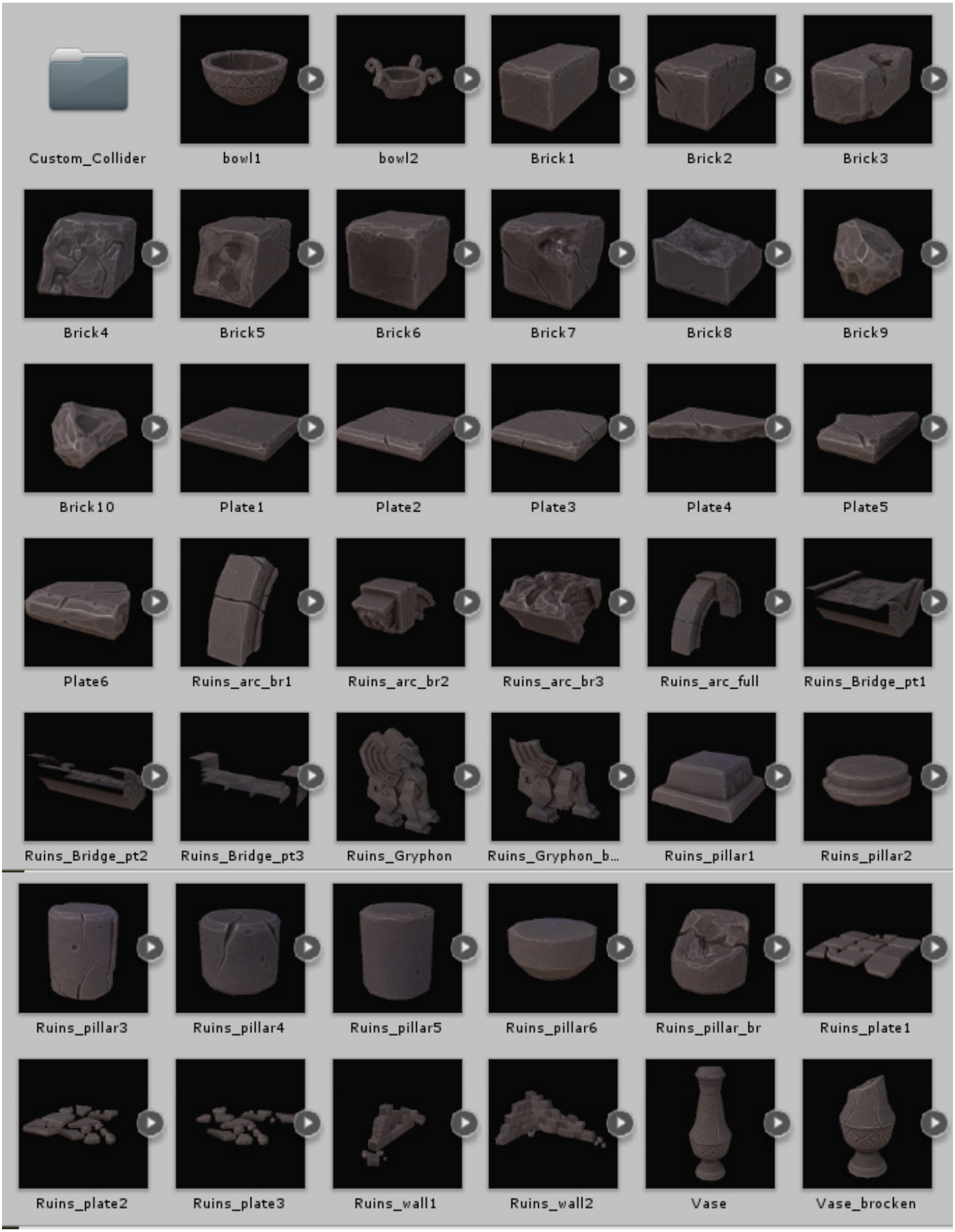
- All textures type (Albedo, Normal, MetallicSmoothness, AO)



- *Rock1* (Verts- **173**, Tris -**234**)
- *Rock2* (Verts- **106**, Tris -**134**)
- *Rock3* (Verts- **102**, Tris -**136**)
- *Rock4* (Verts- **376**, Tris -**608**)
- *Rock_Wall1* (Verts- **1235**, Tris -**2124**)

4. Ruins

- *bowl1* (Verts- **148**, Tris -**168**)
- *bowl2* (Verts- **500**, Tris -**488**)
- *Brick1* (Verts- **26**, Tris -**28**)
- *Brick2* (Verts- **26**, Tris -**28**)
- *Brick3* (Verts- **95**, Tris -**102**)
- *Brick4* (Verts- **57**, Tris -**62**)
- *Brick5* (Verts- **68**, Tris -**70**)
- *Brick6*(Verts- **26**, Tris -**28**)
- *Brick7* (Verts- **75**, Tris -**70**)
- *Brick8* (Verts- **55**, Tris -**62**)
- *Brick9* (Verts- **29**, Tris -**34**)
- *Brick10* (Verts- **48**, Tris -**64**)
- *Plate1* (Verts- **40**, Tris -**28**)
- *Plate2* (Verts- **40**, Tris -**28**)
- *Plate3* (Verts- **56**, Tris -**52**)



- All textures resolution 2048

- All textures type (Albedo, Normal, MetallicSmoothness, AO)

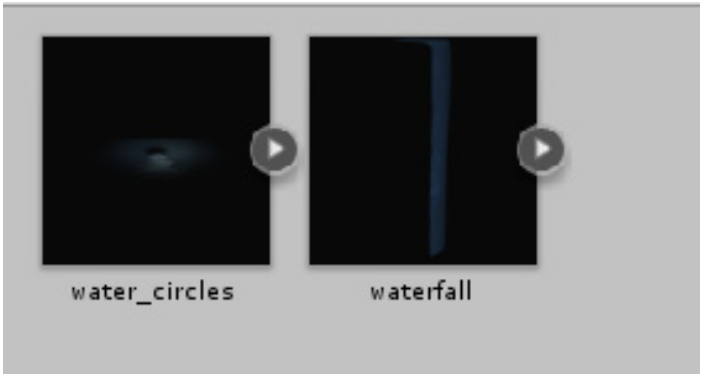
- *Plate4* (Verts- **60**, Tris -**48**)
- *Plate5* (Verts- **58**, Tris -**48**)
- *Plate6* (Verts- **50**, Tris -**40**)
- *Ruins_arc_br1* (Verts- **138**, Tris -**168**)
- *Ruins_arc_br2* (Verts- **176**, Tris -**216**)
- *Ruins_arc_br3* (Verts- **118**, Tris -**130**)
- *Ruins_arc_full* (Verts- **286**, Tris -**392**)
- *Ruins_Bridge_pt1* (Verts- **166**, Tris -**180**)
- *Ruins_Bridge_pt2* (Verts- **128**, Tris -**130**)
- *Ruins_Bridge_pt3* (Verts- **140**, Tris -**106**)
- *Ruins_Gryphon* (Verts- **1387**, Tris -**1366**)
- *Ruins_Gryphon_brocken* (Verts- **795**, Tris -**713**)
- *Ruins_pillar1* (Verts- **52**, Tris -**52**)
- *Ruins_pillar2* (Verts- **99**, Tris -**120**)
- *Ruins_pillar3* (Verts- **214**, Tris -**264**)
- *Ruins_pillar4* (Verts- **64**, Tris -**80**)
- *Ruins_pillar5* (Verts- **64**, Tris -**80**)
- *Ruins_pillar6* (Verts- **88**, Tris -**100**)
- *Ruins_pillar_br* (Verts- **71**, Tris -**90**)
- *Ruins_plate1* (Verts- **743**, Tris -**654**)
- *Ruins_plate2* (Verts- **832**, Tris -**768**)
- *Ruins_plate3* (Verts- **743**, Tris -**702**)
- *Ruins_wall1* (Verts- **815**, Tris -**884**)
- *Ruins_wall2* (Verts- **2039**, Tris -**2182**)
- *Vase* (Verts- **182**, Tris -**288**)
- *Vase_brocken* (Verts- **170**, Tris -**244**)

For some objects (bridges, arches) low-poly custom colliders were created

- *Collider_Ruins_arc_full* (Verts- **24**, Tris -**44**)
- *Collider_Ruins_Bridge_pt1* (Verts- **85** Tris -**44**)
- *Collider_Ruins_Bridge_pt2* (Verts- **106** Tris -**56**)
- *Collider_Ruins_Bridge_pt3*(Verts- **78** Tris -**42**)
- *Collider_Ruins_wall2* (Verts- **28** Tris -**28**)

5. Water

To implement the water in the project uses custom shaders. To create rivers and lakes, a standard plane with a «Water_Stylized» shader is used. To create a waterfall, 2 meshes were developed (Waterfall with a Waterfall_Stylized shader and Water_circle with a Water_Circle_ Stylized shader)

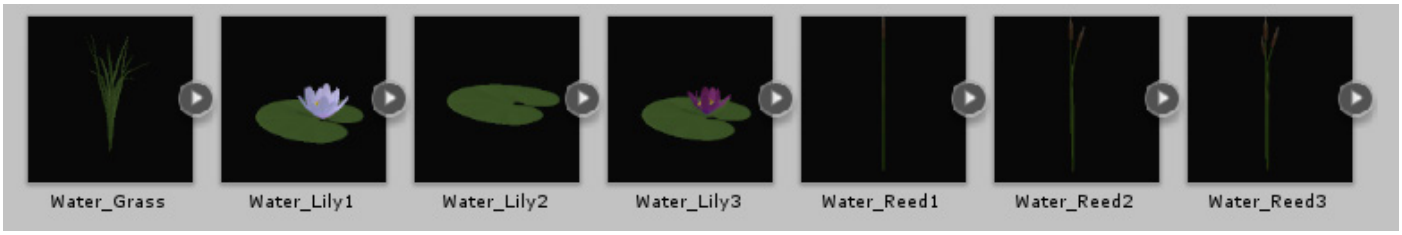


- *water_circles* (Verts- **231**, Tris -**384**)
- *waterfall* (Verts- **50**, Tris -**72**)

6. Water Plants

- All textures resolution 2048

- All textures type (Albedo, Normal, MetallicSmoothness, AO)



- *Water_Grass* (Verts- **190**, Tris -**220**)
- *Water_Lily1* (Verts- **251**, Tris -**290**)
- *Water_Lily2* (Verts- **6**, Tris -**4**)
- *Water_Lily3* (Verts- **223**, Tris -**258**)
- *Water_Reed1* (Verts- **135**, Tris -**184**)
- *Water_Reed2* (Verts- **270**, Tris -**368**)
- *Water_Reed3* (Verts- **405**, Tris -**552**)

7. Wooden objects

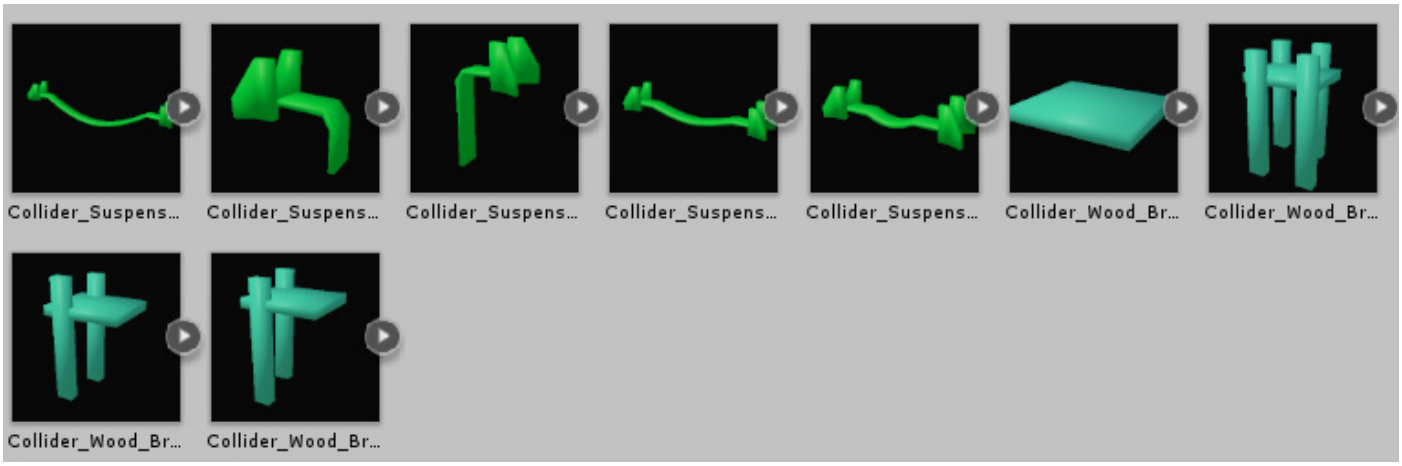
- All textures resolution 2048

- All textures type (Albedo, Normal, MetallicSmoothness, AO)

- Prefab animated Boat

-For some objects low-poly custom colliders were created

- *Coal* (Verts- **108**, Tris -**166**)
- *Fire1* (Verts- **423**, Tris -**566**)
- *Fire2* (Verts- **297**, Tris -**406**)
- *Firewood1* (Verts- **63**, Tris -**80**)
- *Firewood2* (Verts- **54**, Tris -**64**)
- *Firewood3* (Verts- **63**, Tris -**80**)
- *Suspension_Bridge_Big* (Verts- **12032**, Tris -**16226**)
- *Suspension_Bridge_Brocken1* (Verts- **3340**, Tris -**4430**)
- *Suspension_Bridge_Brocken2* (Verts- **4798**, Tris -**6436**)
- *Suspension_Bridge_Medium* (Verts- **8402**, Tris -**11280**)
- *Wood_Boat* (Verts- **420**, Tris -**592**)
- *Wood_Bridge1* (Verts- **966**, Tris -**1312**)
- *Wood_Bridge2* (Verts- **1904**, Tris -**2414**)
- *Wood_Bridge3* (Verts- **1540**, Tris -**2026**)
- *Wood_Bridge4* (Verts- **1474**, Tris -**1904**)
- *Wood_Fence1* (Verts- **572**, Tris -**740**)
- *Wood_Fence2* (Verts- **488**, Tris -**596**)
- *Wood_Fence3* (Verts- **392**, Tris -**470**)
- *Wood_Fence4* (Verts- **284**, Tris -**352**)



For some objects (bridges, arches) low-poly custom colliders were created

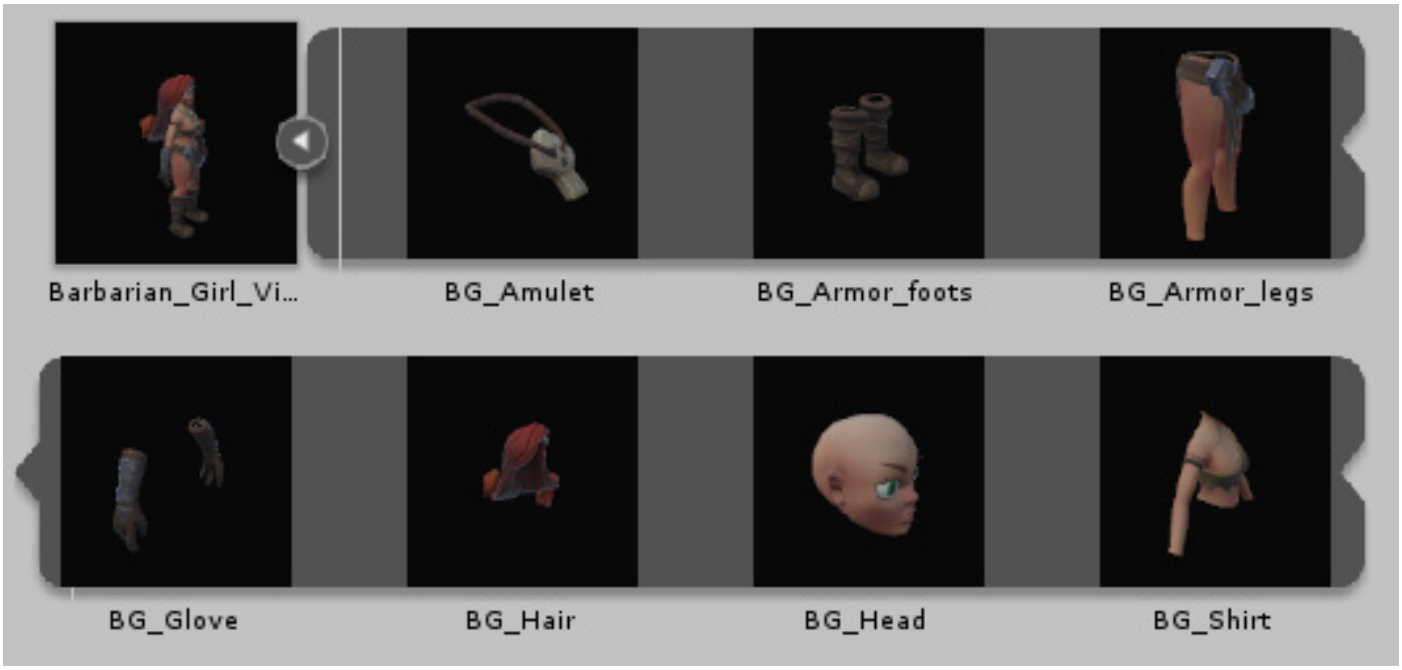
- *Collider_Suspension_Bridge_Big* (Verts- **198**, Tris -**220**)
- *Collider_Suspension_Bridge_Brocken1* (Verts- **69**, Tris -**72**)
- *Collider_Suspension_Bridge_Brocken2* (Verts- **69**, Tris -**72**)
- *Collider_Suspension_Bridge_Medium* (Verts- **156**, Tris -**172**)
- *Collider_Suspension_Bridge_Small* (Verts- **144**, Tris -**156**)
- *Collider_Wood_Bridge1* (Verts- **14**, Tris -**12**)
- *Collider_Wood_Bridge2* (Verts- **70**, Tris -**60**)
- *Collider_Wood_Bridge3* (Verts- **42**, Tris -**36**)
- *Collider_Wood_Bridge4* (Verts- **42**, Tris -**36**)

For instructions on designing bridges and using custom colliders, see the video:
https://youtu.be/iJVHFpbht-M

8. Character «Barbarian Girl Victoria»

Character **Barbarian Girl «Victoria»** designed as the main acting character of the **Stylized World «Victoria»**. The character rig is designed for the **"Humanoid"** animation type, which allows the developer to attach third-party animations designed for this type to a barbarian girl.

The character is divided into 7 meshes (*BG_Amulet, BG_Armor_foots, BG_Armor_legs, BG_Glove, BG_Hair, BG_Head, BG_Shirt*) attached to one rig. This is done specifically for the next update in which character customization will be implemented.



Polycount:

-***FULL Barbarian Girl «Victoria»*** (Verts- **5794**, Tris -**8705**)

-***BG_Amulet*** (Verts- **281**, Tris -**438**)

-***BG_Armor_foots*** (Verts- **1094**, Tris -**1580**)

-***BG_Armor_legs*** (Verts- **751**, Tris -**1078**)

-***BG_Glove*** (Verts- **926**, Tris -**1160**)

-***BG_Hair*** (Verts- **869**, Tris -**1422**)

-***BG_Head*** (Verts- **1014**, Tris -**1626**)

-***BG_Shirt*** (Verts- **859**, Tris -**1401**)

- **SWORD** (Verts- **357**, Tris -**446**)

Texture:

Barbarian Girl «Victoria» :

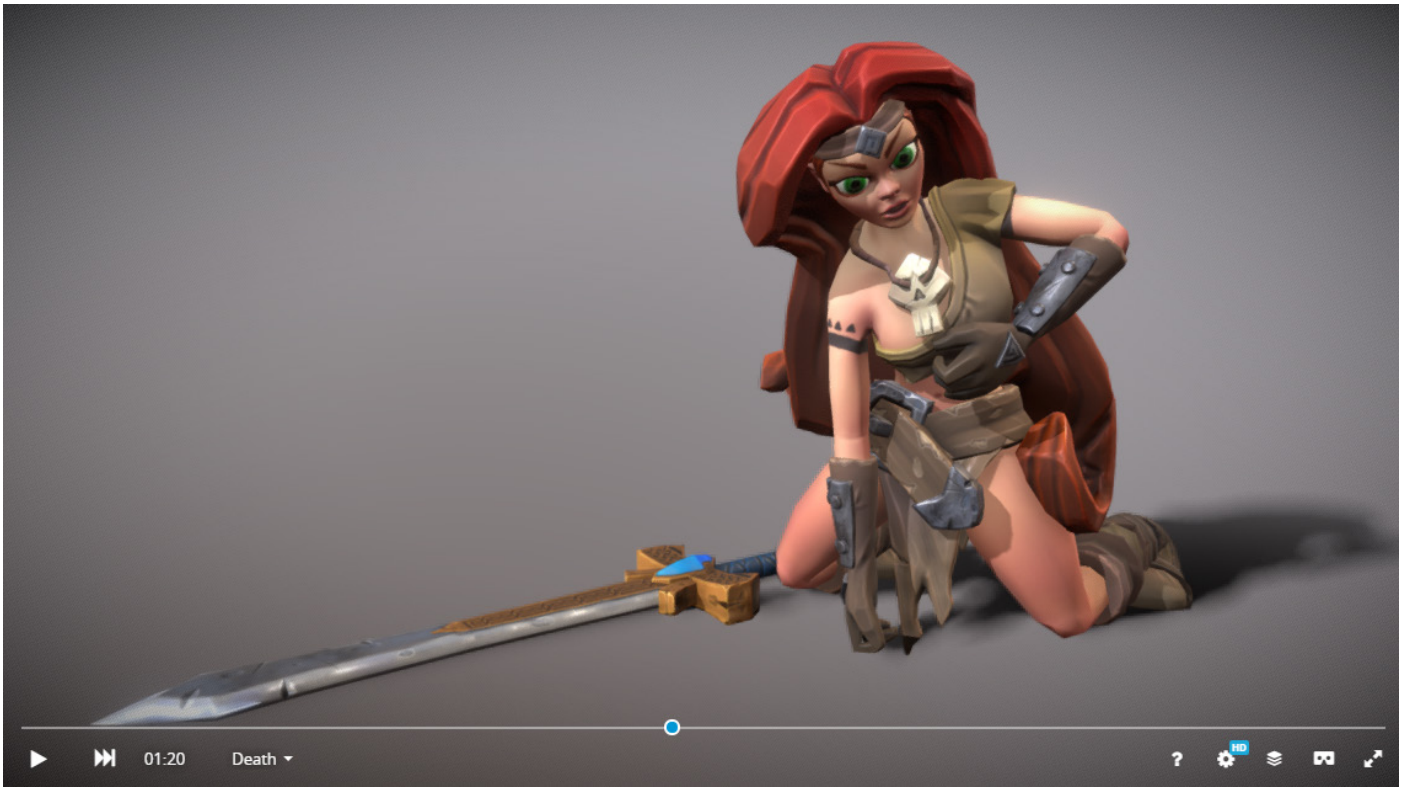
- All textures resolution **2048**
- All textures type (**Albedo, Normal, MetallicSmoothness, AO**)

Sword:

- Textures resolution **2048**
- Textures type (**Albedo**(*2 variants (red, blue)*)**, Normal, MetallicSmoothness, AO, Emission**)

Animations:

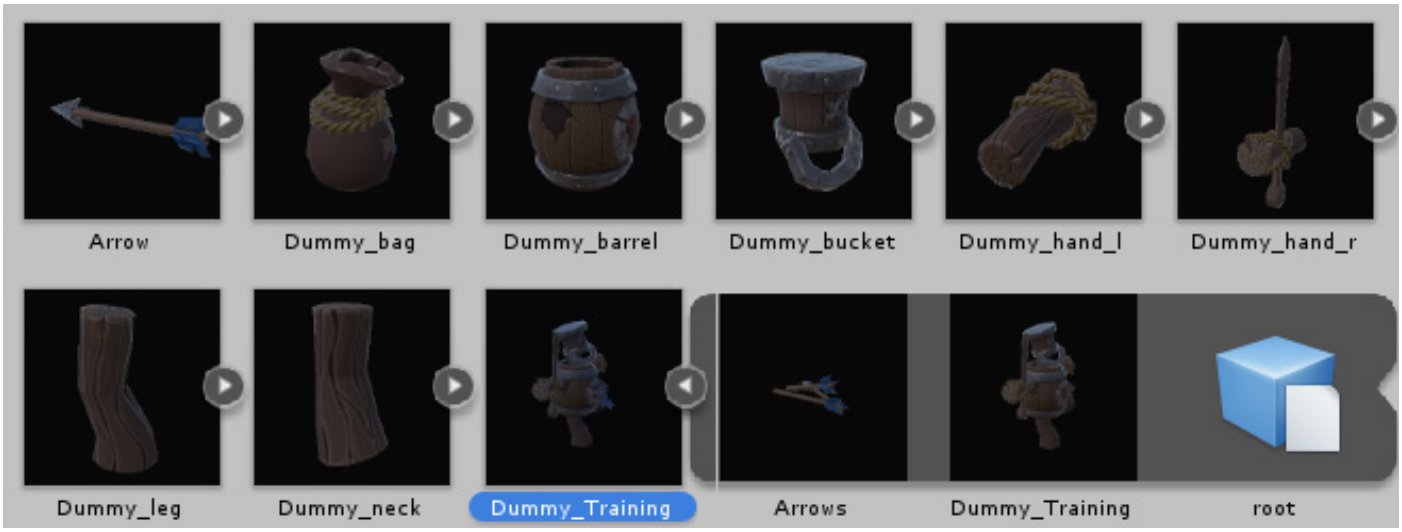
-Idle_Start
-Idle
-Run
-Talk
-Idle_battle
-Attack1
-Attack2
-Attack3
-Hit
-Death



Preview animations:
https://skfb.ly/6MrI6

9. Enemy Character «Training Dummy»

The Enemy character **Training Dummy** is presented in 2 versions (*whole (animated), parts (broken)*). The first version consists of 2 meshes (*Dummy_Training, Arrows*) attached to one rig. This is done to hide if you want a developer with a mesh with arrows. The second option consists of individual objects (*Arrow, Dummy_bag, Dummy_barrel, Dummy_bucket, Dummy_hand_l, Dummy_hand_r, Dummy_leg, Dummy_neck*) to which you can apply physics, which is implemented in the project.



Polycount:

-***FULL Dummy_Training*** (Verts- **1948**, Tris -**2636**)

-***Arrow*** (Verts- **313**, Tris -**328**)

-***Dummy_Training*** (Verts- **1635**, Tris -**2308**)

- **Arrow** (Verts- **117**, Tris -**120**)
- **Dummy_bag** (Verts- **339**, Tris -**476**)
- **Dummy_barrel** (Verts- **287**, Tris -**424**)
- **Dummy_bucket** (Verts- **223**, Tris -**276**)
- **Dummy_hand_l** (Verts- **358**, Tris -**370**)
- **Dummy_hand_r** (Verts- **423**, Tris -**600**)
- **Dummy_leg** (Verts- **69**, Tris -**82**)
- **Dummy_neck** (Verts- **69**, Tris -**82**)

Texture:

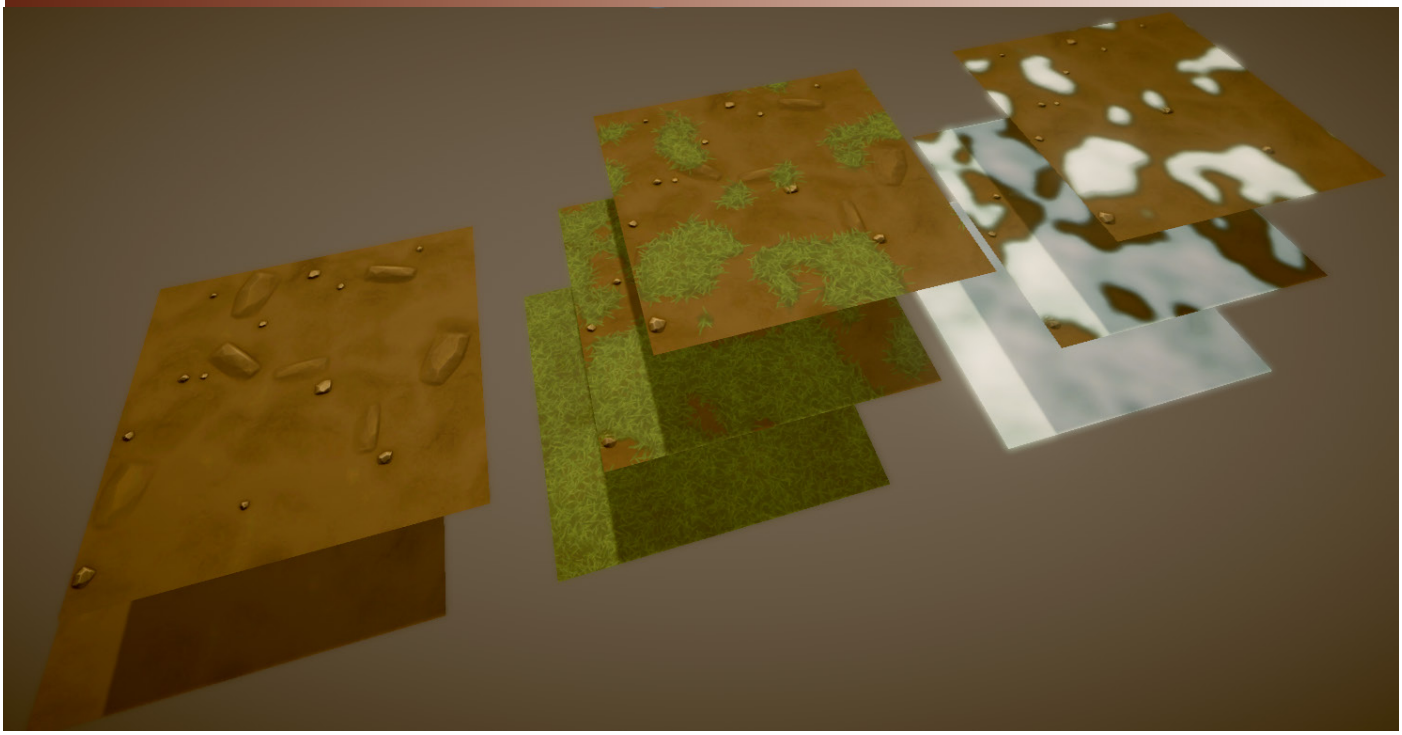
- All textures resolution **2048**
- All textures type (**Albedo, Normal, MetallicSmoothness, AO**)

Animations:

-Idle1
-Idle2
-Hit1
-Hit2
-Death

Preview animations:
https://skfb.ly/6MrIJ

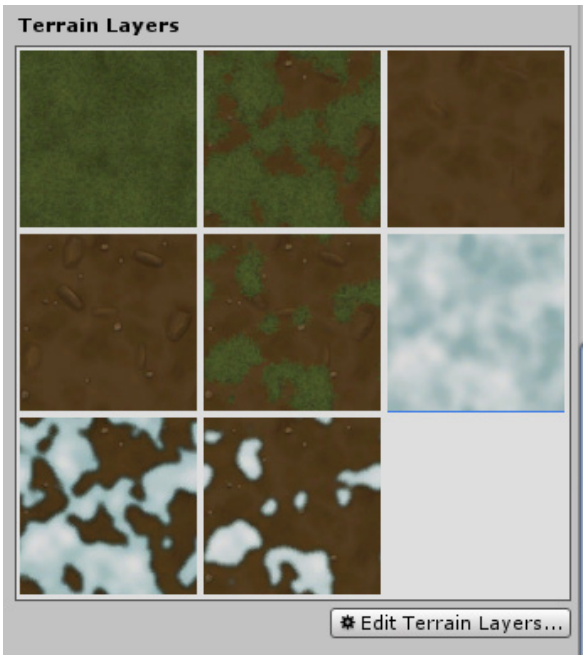
10. Ground Textures



The project contains **8 Tiled Ground Textures** for **Terrains,** which you can use to create high-quality, colorful and non-uniform terrains.

Texture:

- All textures resolution **2048**
- All textures type (**BaseColor, Normal, Metallic, Roughness, AO, Height**)



11. Visual FX

The project implemented visual effects based on particle systems (*snow, defoliation, water splashes, the effect of indicating the target point, fire, the effect of hitting the tree, the sun's rays*)

defoliation

sun's rays

fire

snow

water splashes

hit the tree

12. Cursors

Also for the style of the project developed 2 **unique cursor (res-256)**:

main cursor

attack cursor

13. Custom Shaders (Shader Graphs)

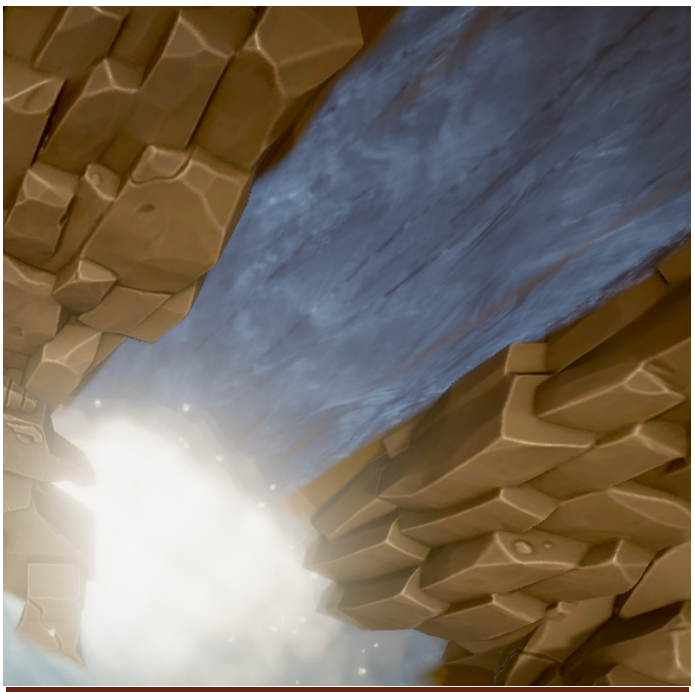
In Shader graph, custom shaders are developed:

a ghost shader to highlight a character if it is blocked by another object.

shader animation foliage, grass and other plants imitating swing in a gust of wind.

water animation shader

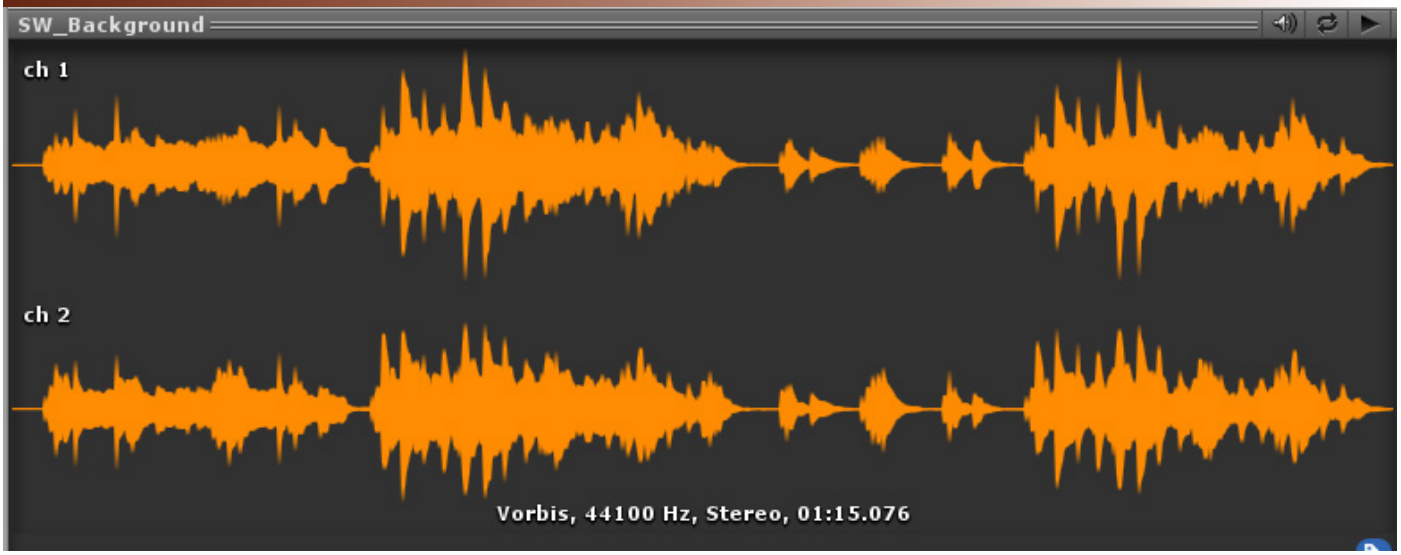
water splash animation shader



waterfall animation shader

snow imitation shader on objects

13. Sounds



Music:

2 unique musical compositions specially written for this project:

- **SW_Background** (calm background music accompanying the main character throughout the game)
- **SW_Battle (**dynamic battle music that comes on when attacking)

Sound FX:

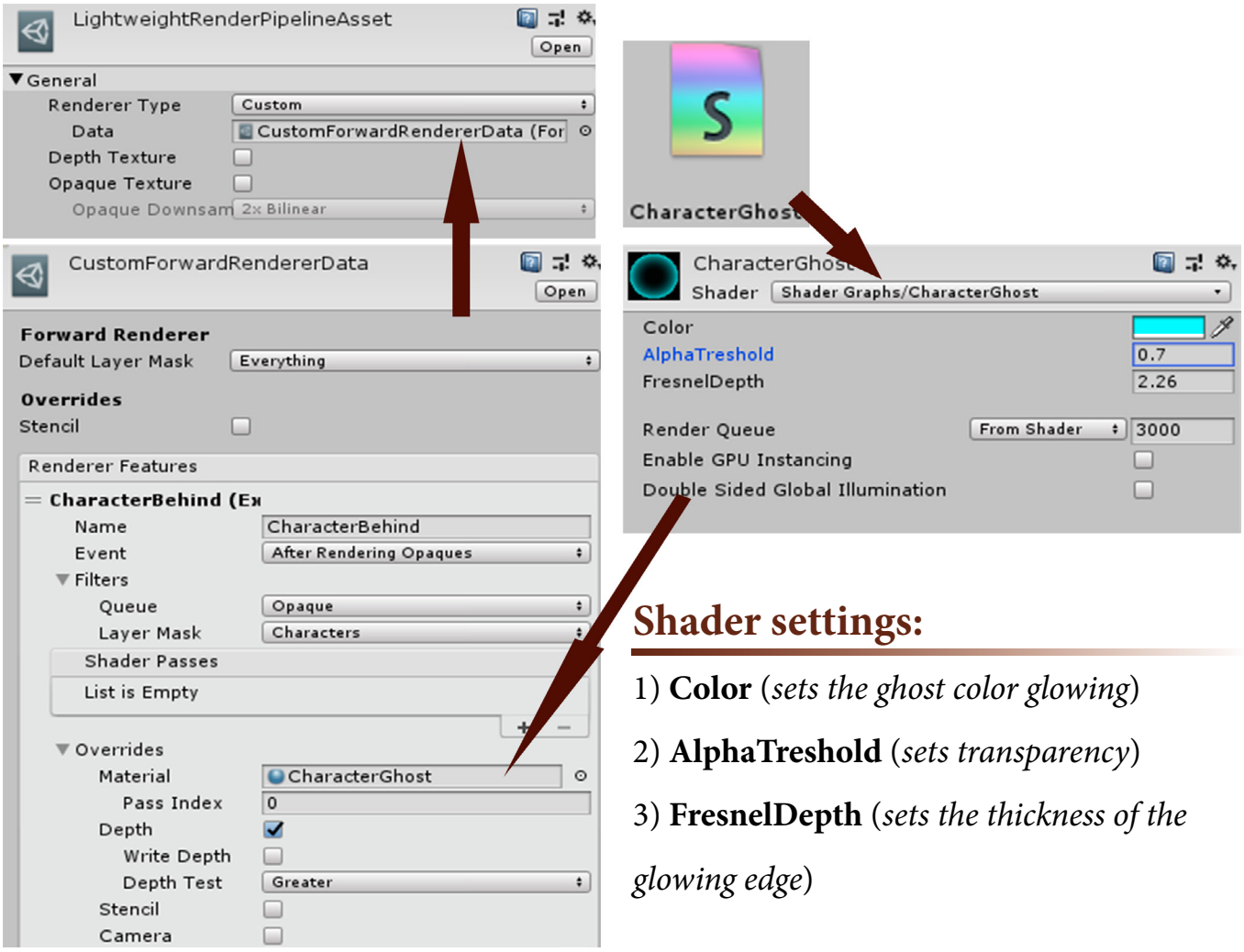
- **birds (**birds singing and chirping**)**
- **crash-wood** (a wooden object breaks down)
- **fire** (fire is burning)
- **water** (river flows)
- **waterfall** (flowing waterfall)
- **wind** (the wind is blowing in the gorge)
- **wood_01** (blow to the tree)
- **wood_02** (blow to the tree)
- **wood_03** (blow to the tree)

III. Custom Shader Settings

Shader «CharacterGhost»:

A ghost shader to highlight a character if it is blocked by another object.

The material with the shader ghost character is used in the layers of user render settings LWRP:



Shader settings:

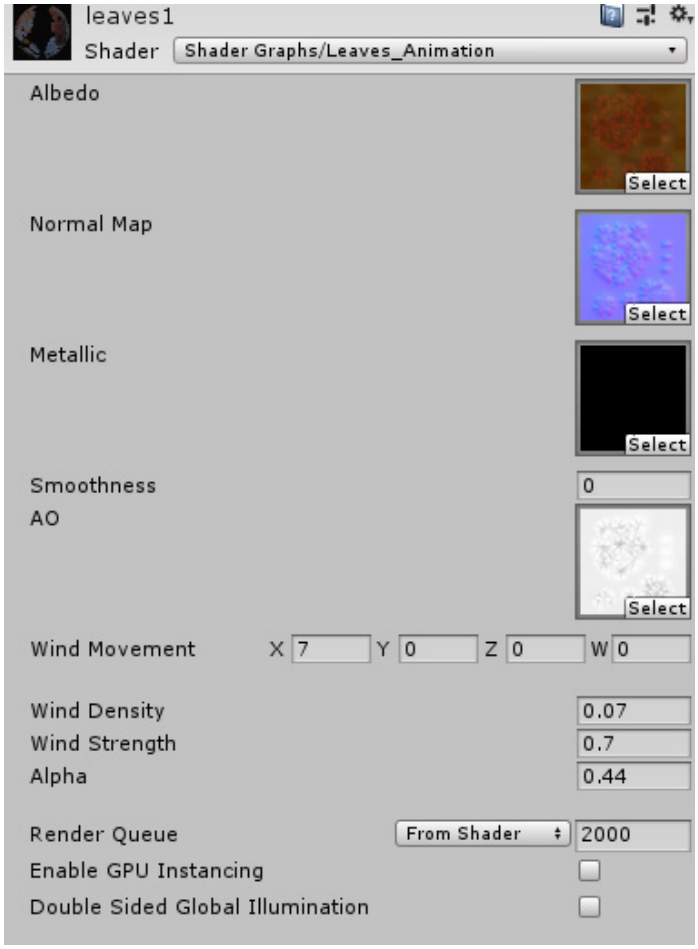
1) **Color** (*sets the ghost color glowing*)

2) **AlphaTreshold** (*sets transparency*)

3) **FresnelDepth** (*sets the thickness of the glowing edge*)

Shader «Leaves_Animation»:

Shader animation foliage and plants imitating swing in a gust of wind.

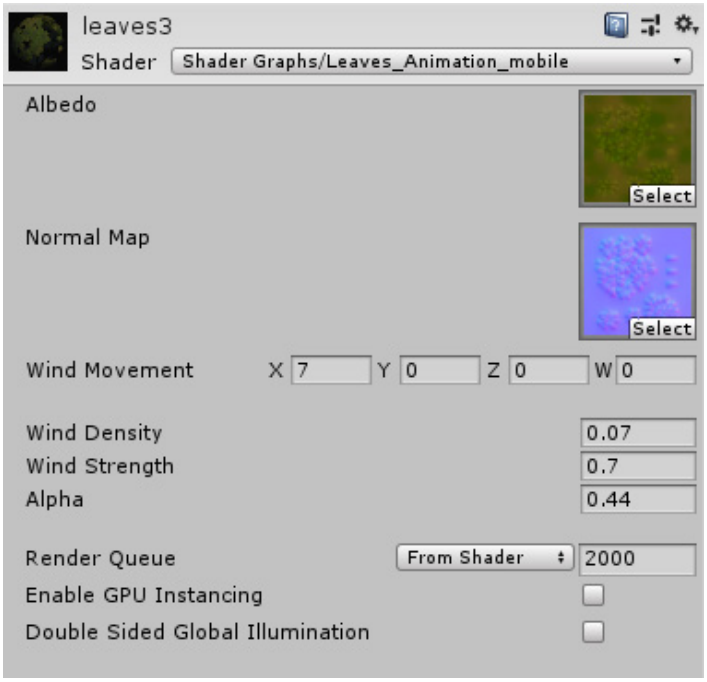


Shader settings:

- **Albedo, Normal Map, Metallic, AO** (*custom textures*)
- **Wind Movement** (*set the direction of the wind*)
- **Wind Density** (*set wind density*)
- **Wind Strength** (*set wind strength*)
- **Alpha** (*alpha channel impact on transparency*)

Shader «Leaves_Animation_mobile»:

Shader animation foliage and plants imitating swing in a gust of wind. Optimized for mobile platforms.

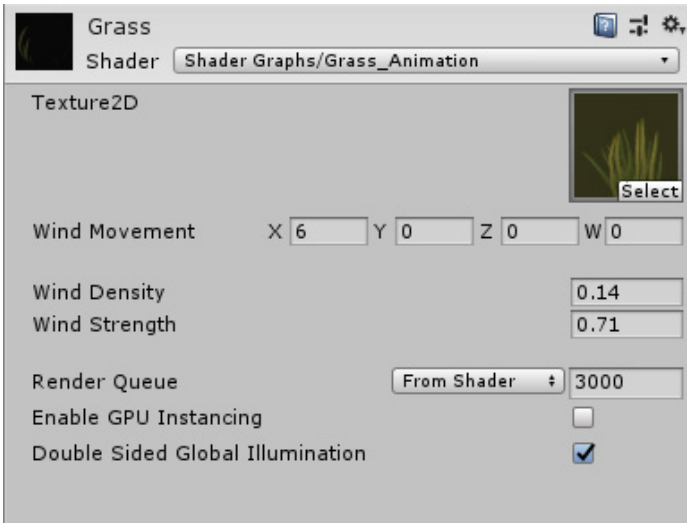


Shader settings:

- **Albedo, Normal Map** (*custom textures*)
- **Wind Movement** (*set the direction of the wind*)
- **Wind Density** (*set wind density*)
- **Wind Strength** (*set wind strength*)
- **Alpha** (*alpha channel impact on transparency*)

Shader «Grass_Animation»:

Shader animation grass imitating swing in a gust of wind.

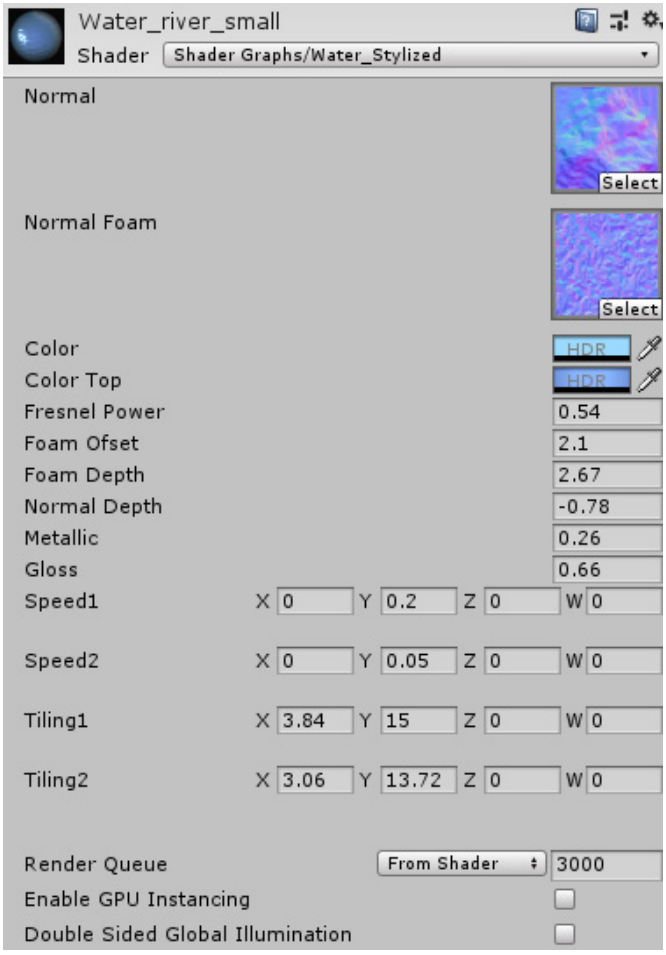


Shader settings:

- **Texture 2d** (*custom textures*)
- **Wind Movement** (*set the direction of the wind*)
- **Wind Density** (*set wind density*)
- **Wind Strength** (*set wind strength*)

Shader «Water_Stylized»:

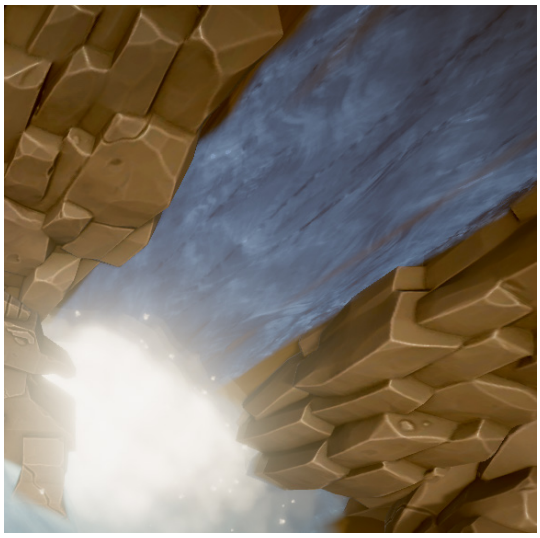
The water animation shader has a large number of settings and can be used to create rivers, lakes, seas and more.



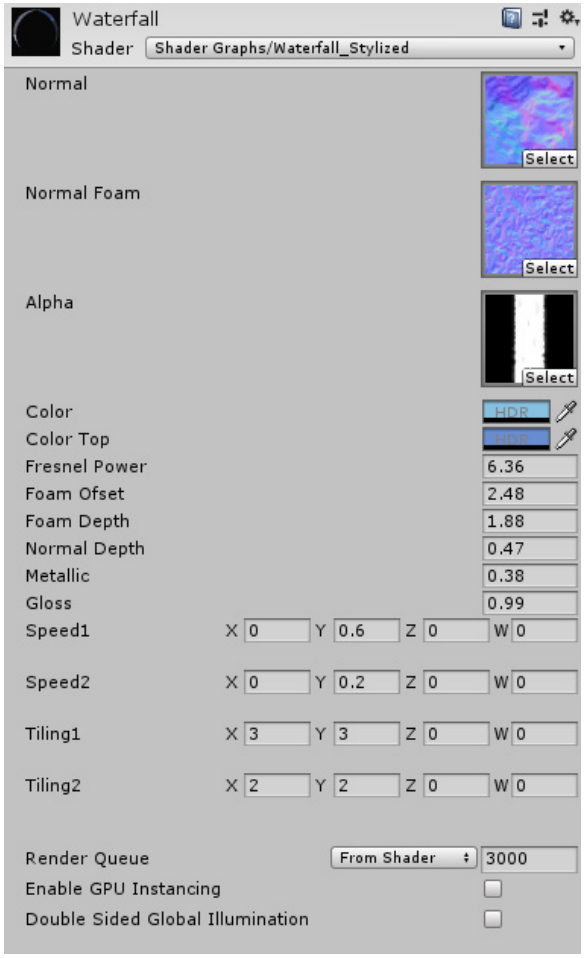
Shader settings:

- **Normal** (*Normal map to simulate waves on the water surface*)
- **Normal Foam** (*Normal map to simulate foam around objects placed in the water and along the shore*)
- **Color** (*The color of the water surface*)
- **Color Top** (*Color at depth*)
- **Fresnel Power** (*Water transparency power*)
- **Foam Ofset** (*Foam size*)
- **Foam Depth** (*Foam depth*)
- **Normal Depth** (*Wave height on the surface of the water*)
- **Metallic** (*Water reflection*)
- **Gloss** (*Water gloss*)
- **Speed1** (*The speed of the first water layer*)
- **Speed2** (*The speed of the second layer of water*)
- **Tiling1** (*The tiling of the first water layer*)
- **Tiling2** (*The tiling of the second water layer*)

Shader «Waterfall_Stylized»:



Waterfall animation shader.

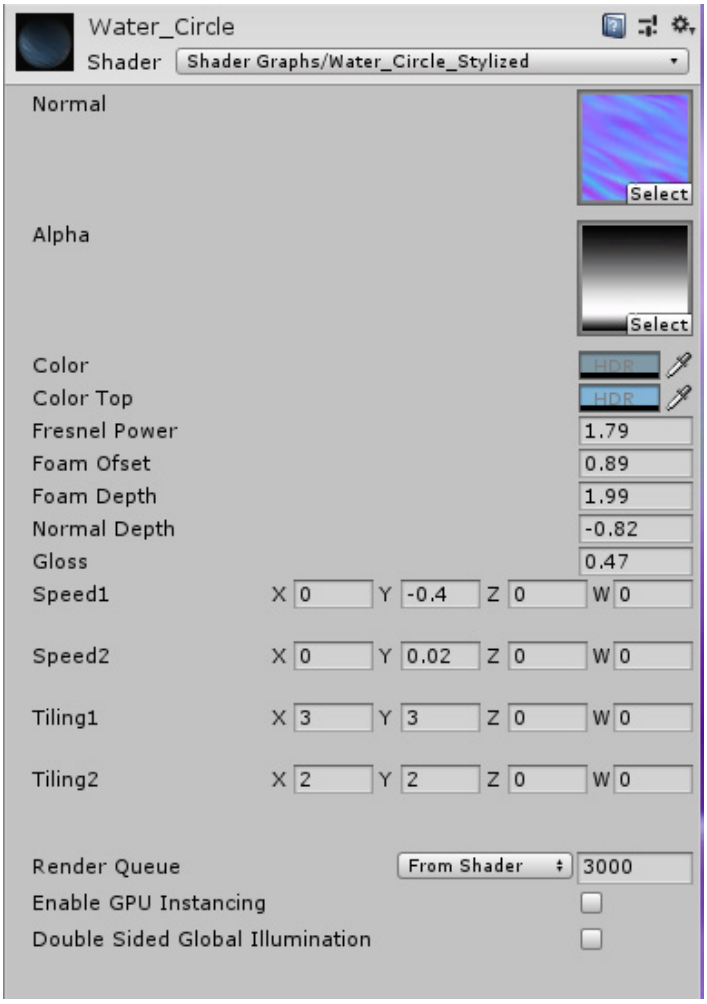


Shader settings:

- **Normal** (*Normal map to simulate waves on the water surface*)
- **Normal Foam** (*Normal map to simulate foam around objects placed in the water and along the shore*)
- **Alpha** (*Transparency mask to create inhomogeneity of falling water*)
- **Color** (*The color of the water surface*)
- **Color Top** (*Color at depth*)
- **Fresnel Power** (*Water transparency power*)
- **Foam Ofset** (*Foam size*)
- **Foam Depth** (*Foam depth*)
- **Normal Depth** (*Wave height on the surface of the water*)
- **Metallic** (*Water reflection*)
- **Gloss** (*Water gloss*)
- **Speed1** (*The speed of the first water layer*)
- **Speed2** (*The speed of the second layer of water*)
- **Tiling1** (*The tiling of the first water layer*)
- **Tiling2** (*The tiling of the second water layer*)

Shader «Water_Circle_Stylized»:

Shader animation of stains on the water.

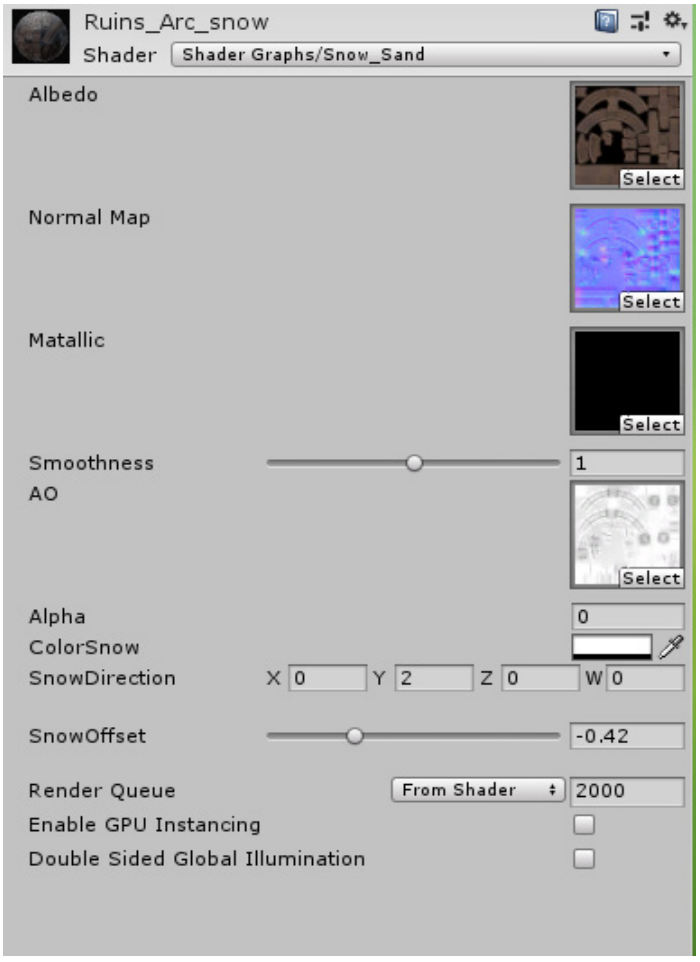


Shader settings:

- **Normal** (*Normal map to simulate waves on the water surface*)
- **Alpha** (*Transparency mask for fade out*)
- **Color** (*The color of the water surface*)
- **Color Top** (*Color at depth*)
- **Fresnel Power** (*Water transparency power*)
- **Foam Ofset** (*Foam size*)
- **Foam Depth** (*Foam depth*)
- **Normal Depth** (*Wave height on the surface of the water*)
- **Metallic** (*Water reflection*)
- **Gloss** (*Water gloss*)
- **Speed1** (*The speed of the first water layer*)
- **Speed2** (*The speed of the second layer of water*)
- **Tiling1** (*The tiling of the first water layer*)
- **Tiling2** (*The tiling of the second water layer*)

Shader «Snow_Sand»:

The shader is designed for applying snow cover or sand on objects. Regardless of the rotation or position of the object, the snow cover will always be on top.

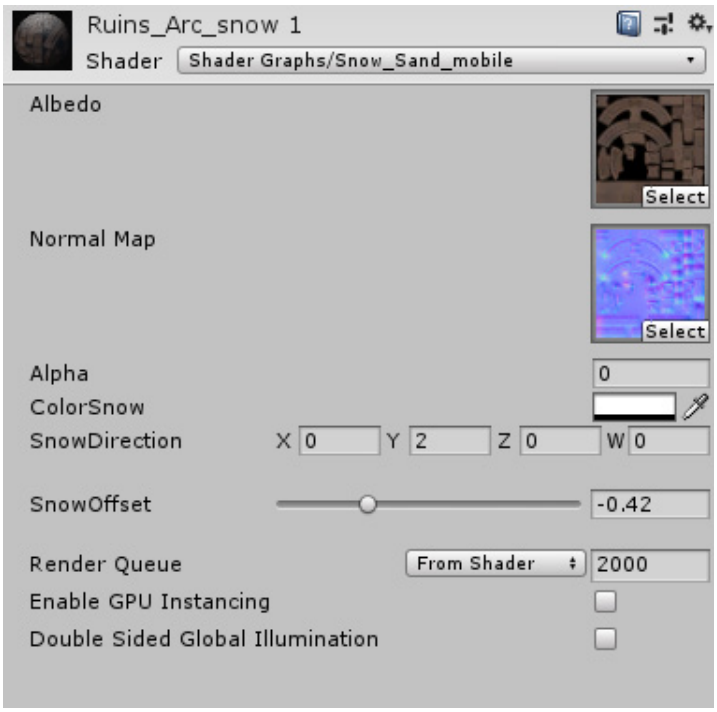


Shader settings:

- **Albedo, Normal Map, Metallic, AO** (*custom textures*)
- **ColorSnow** (*sets the color of snow or sand*)
- **SnowDirection** (*sets the position of the snow on the object*)
- **SnowOffset** (*sets the height of the snow cover*)

Shader «Snow_Sand_mobile»:

The shader is designed for applying snow cover or sand on objects. Regardless of the rotation or position of the object, the snow cover will always be on top.
Mobile version.



Shader settings:

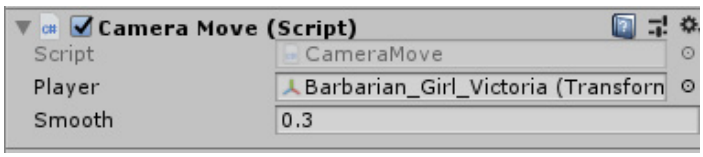
- **Albedo, Normal Map** (*custom textures*)
- **ColorSnow** (*sets the color of snow or sand*)
- **SnowDirection** (*sets the position of the snow on the object*)
- **SnowOffset** (*sets the height of the snow cover*)

Video review on shader settings:
https://youtu.be/iJVHFpbht-M

IV. Script Settings

CameraMove:

This script is assigned to an object that moves smoothly behind the character. This can be a camera, a game object in which the camera is located, or any other game object.



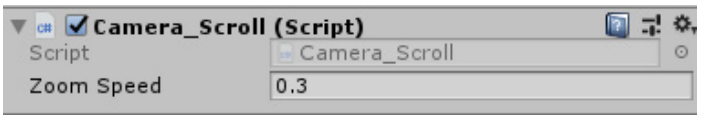
Script settings:

- **Player** (*Transform the object for which you want to move*)

- **Smooth** (*Motion smoothing*)

Camera_Scroll:

The script is assigned to the camera and is designed to approximate or remove the view from the camera.

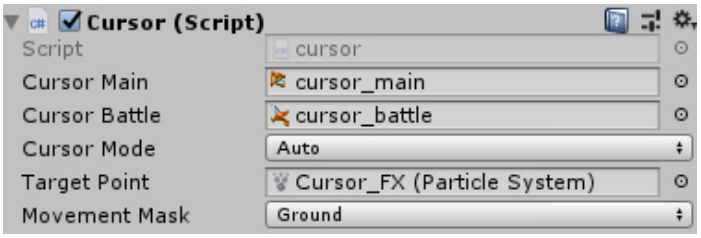


Script settings:

- **Zoom Speed** (set *zoom speed*)

Cursor:

The script is assigned to the camera and is designed to control the appearance of the cursor and the appearance of the effect of specifying the point of movement of the character.



Script settings:

- **Cursor Main** (*Cursor in a calm state.*)

- **Cursor Battle** (*Cursor when pointing at the enemy*)

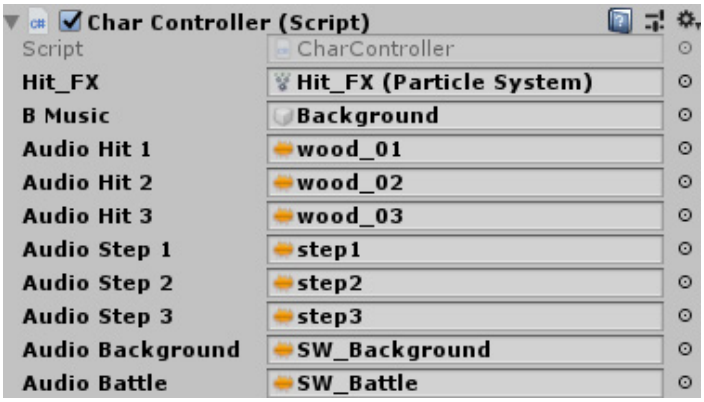
- **Cursor Mode** (***Auto****-the cursor scales to 32x32px,* ***Force Software*** *- the cursor is displayed with the size of the original cursor image.*)

- **Target Point** (*Choosing a particle system effect specifying the character's waypoint*)

- **Movement Mask** (*Selecting a layer on which the effect of specifying the character's waypoint is allowed to appear.*)

CharController:

The script is assigned to the main character and controls the movement and attack of the character, the sound effects of the character, the background music of the game, the effect of striking the enemy.

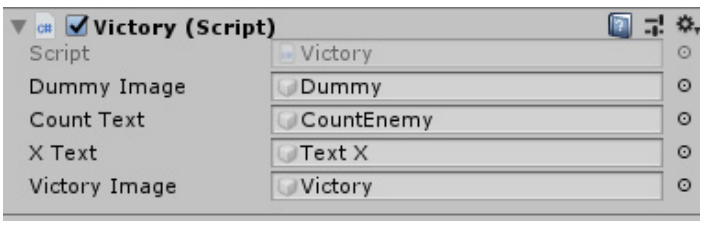


Script settings:

- **Hit_FX** (*Particle system of the blow effect on the enemy.*)
- **B Music** (*Set game object with audio source for background music.*)
- **Audio_Hit1** (*Set sound effect first strike.*)
- **Audio_Hit2** (*Set sound effect second strike.*)
- **Audio_Hit3** (*Set sound effect third strike.*)
- **Audio_Step1** (*Set the sound effect of the first step.*)
- **Audio_Step2** (*Set the sound effect of the second step.*)
- **Audio_Step3** (*Set the sound effect of the third step.*)
- **Audio_Background** (*Set background music.*)
- **Audio_Battle** (*Set battle music.*)

Victory:

Victory Script is located on the Canvas object. This script starts when the enemies counter is zero, while hiding all the UI elements of counting the number of remaining enemies and the inscription Victory appears.

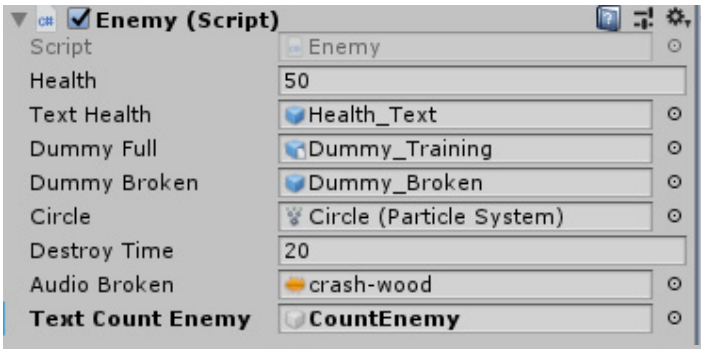


Script settings:

- **Dummy Image** (*Set UI Dummy*)
- **Count Text** (*Set UI CountEnemy*)
- **X Text** (*Set UI Text X*)
- **Victory Image** (*Set UI* Victory)

Enemy:

The script controls the number and display of the health of the enemy, sound effects and time of destruction after death.



Script settings:

- **Health** (*The initial health of the enemy.*)
- **Text Health** (*Set the remaining health display object.*)
- **Dummy Full** (*Set game object Dummy Full. Deactivated after reaching health «0»*)
- **Dummy Broken** (*Set game object Dummy Broken. Activated after reaching health «0».*)
- **Circle** (*Set particle system of active enemy release.*)
- **Destroy Time** (*Set time to remove dummy broken.*)
- **Audio Broken** (*Set sound effect destruction of the enemy.*)
- **Text Count Enemy** (*Set Text Object to Count the Number of Remaining Enemies.*)

TextHealh:

The script is applied to the game object Health_Text. It contains a void that starts after playing an animation of the game object Health_Text and hides this object.

Resolution:

The script is bound to the GameManager_MOBILE game object. This object must be activated to optimize the project for mobile platforms. The script sets the resolution to 1024x576.

For more details on project optimization and launch on a mobile platform, see the video: https://youtu.be/KZy9kLeWKVE

Support

Web Site: http://polynext.dx.am

Email: polynext.3d@gmail.com