



V 1.1b

1. THE CONTENTS OF FOLDERS
2. THE MAIN (ONLINE) SCENE'S CONTENTS
3. CREATING A SCENE
4. CHARACTER CUSTOMIZATION
5. CREATING AN OBJECT THAT CAN BE MINED
6. CREATING AN OBJECT THAT YOU CAN PICK UP
7. RECIPES MANAGEMENT
8. ITEMS MANAGEMENT
9. CREATING AN ITEM PREFAB
 - a. MELEE WEAPON
 - b. RANGED WEAPON (BOW)
 - c. FOOD AND DRINK
10. CREATING AN ARMOR
11. CREATING A PLACEABLE PREFAB
 - a. SETTING THE ATTACHMENT POINT
12. CREATING A CONTAINER
 - a. CREATING A CONTAINER PANEL
13. CREATING A WORKSTATION
 - a. ADDING NEW WORKSTATION TYPES
14. CREATING A DOOR
15. EDITING INVENTORY PANELS
 - a. EDITING CHARACTER PANEL
 - b. CREATING A NEW TABS FOR CRAFTING PANEL
16. CREATING AN AGGRESSIVE MOB
17. CREATING A PASSIVE MOB
18. CREATING A SPAWN POINT (BED)
19. CREATING A GROWING OBJECT (TREE SAPLING OR OTHER PLANT)
20. DAY AND NIGHT CYCLE
21. CHANGELOG

The contents of folders

Audio – all sounds.

Materials – some materials needed for the template.

Models – in the folders inside are models, textures and materials.

Plugins – LitJson plugin for a database of items.

Prefabs – prefabs that are not related to inventory items.

Creatures – friendly and hostile mobs.

Environment – environment objects to spawn on the Terrain.

UI – user interface elements.

Container Panels – various panels for containers (chest and box).

Resources – resources for loading during the game.

Armor Prefabs – armor prefabs.

Item Placeable Prefabs – prefabs for construction items.

Item Prefabs – prefabs of items for hands.

Sprites – various sprites.

Items – items icons.

UI – sprites for interface elements.

Scene – different objects related to the scenes, including the scenes itself.

Scene Prefabs – a database of items and recipes, inventory panels and other things necessary for the template.

Scripts – all scripts.

Skybox – skybox textures and materials.

Streaming Assets – database of items and the readme for it. (In v.1.1a it was removed, database was moved in Resources folder)

Terrain Textures – additional textures for terrain.

The Main (Online) Scene's Contents

Level – the scene's contents.

Wind Zone – wind for trees and bushes.

Water Collider – the plane on the water level.

Cube (1,2,3,4) – invisible walls on the borders of terrain.

Terrain – terrain.

WaterProDaytime – water.

SunLight – the main source of lighting and parent for the system of day/night cycle.

StartPos – start position for players.

Audio – the sound of singing birds.

Other – the other objects in the scene, such as the house and the chest.

MenuCanvas – canvas for the in-game menu.

Return – return button.

Quit – quit button.

MainMenu – main menu button.

Cheat – button, which adds all of the items in your inventory.

Game Canvas – UI elements canvas.

Crosshair – crosshair image.

Sliders – panels of health, hunger, thirst and stamina.

Death Text – the text that appears at the death.

CraftItem – image of crafting item.

Inventory – Inventory canvas.

Inventory Panel – main inventory panel.

Quick Panel – the quick inventory panel.

Character Panel – character's armor panel.

Craft Panel – crafting panels.

Tooltip – item's tooltip panel.

DraggingItem Panel – panel for a dragging item.

GameTooltip – game hints canvas.

GameTooltipText – game tooltip text (when you hover the sight on the target).

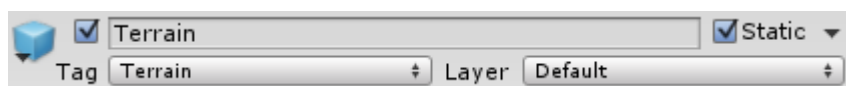
TempText – the tooltip text which fades with time.

Database – the database of items and recipes.

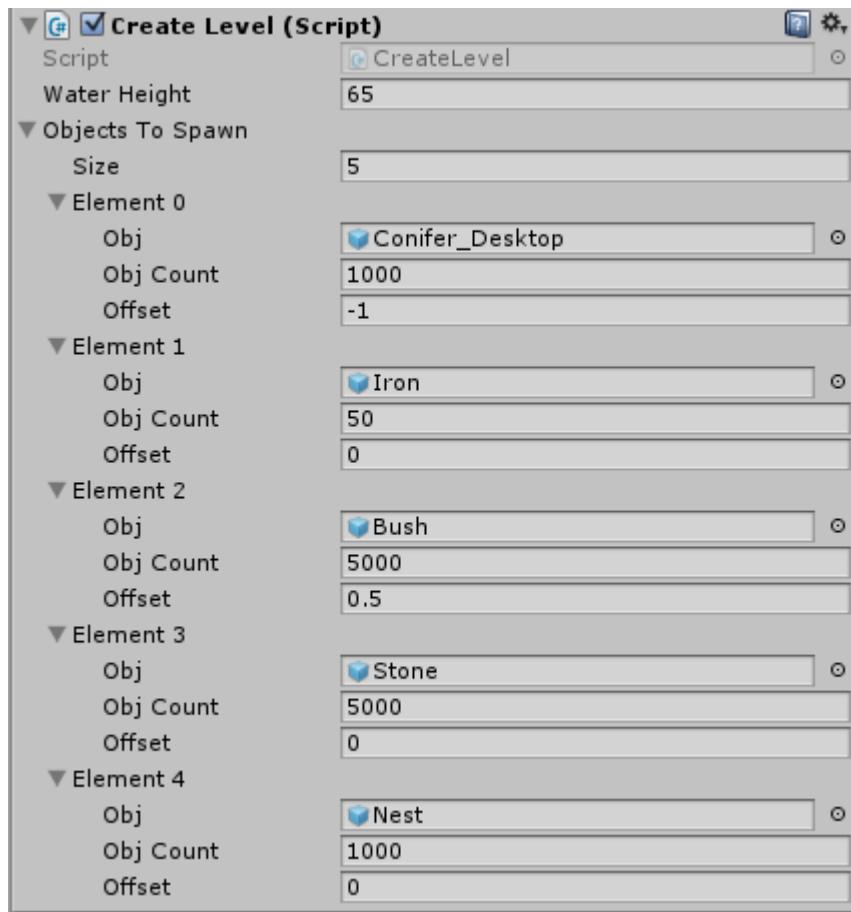
Creating a Scene

Terrain:

Create a terrain or use any surface as the scene geometry. Assign tag «Terrain» to the created terrain.



Add the **Create Level** script and **Network Identity** to it.



Create Level – script for spawn objects on the terrain.

Water Height – height below which objects will not spawn.

Objects To Spawn – list of objects that will spawn.

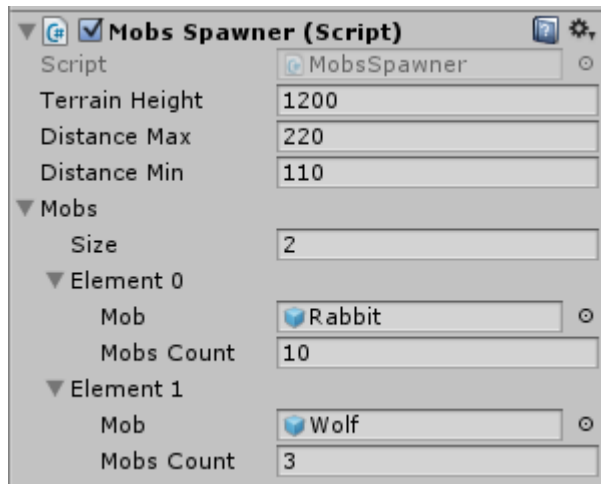
Obj – object's prefab.

Obj Count – the number of objects.

Offset – the offset on the y axis.

Add to the scene the prefabs from the folder «Scene \ ScenePrefabs» (interface and database of items).

Character Customization



Mobs Spawner – script to spawn mobs on the terrain:

Terrain Height – maximum height for spawn mobs.

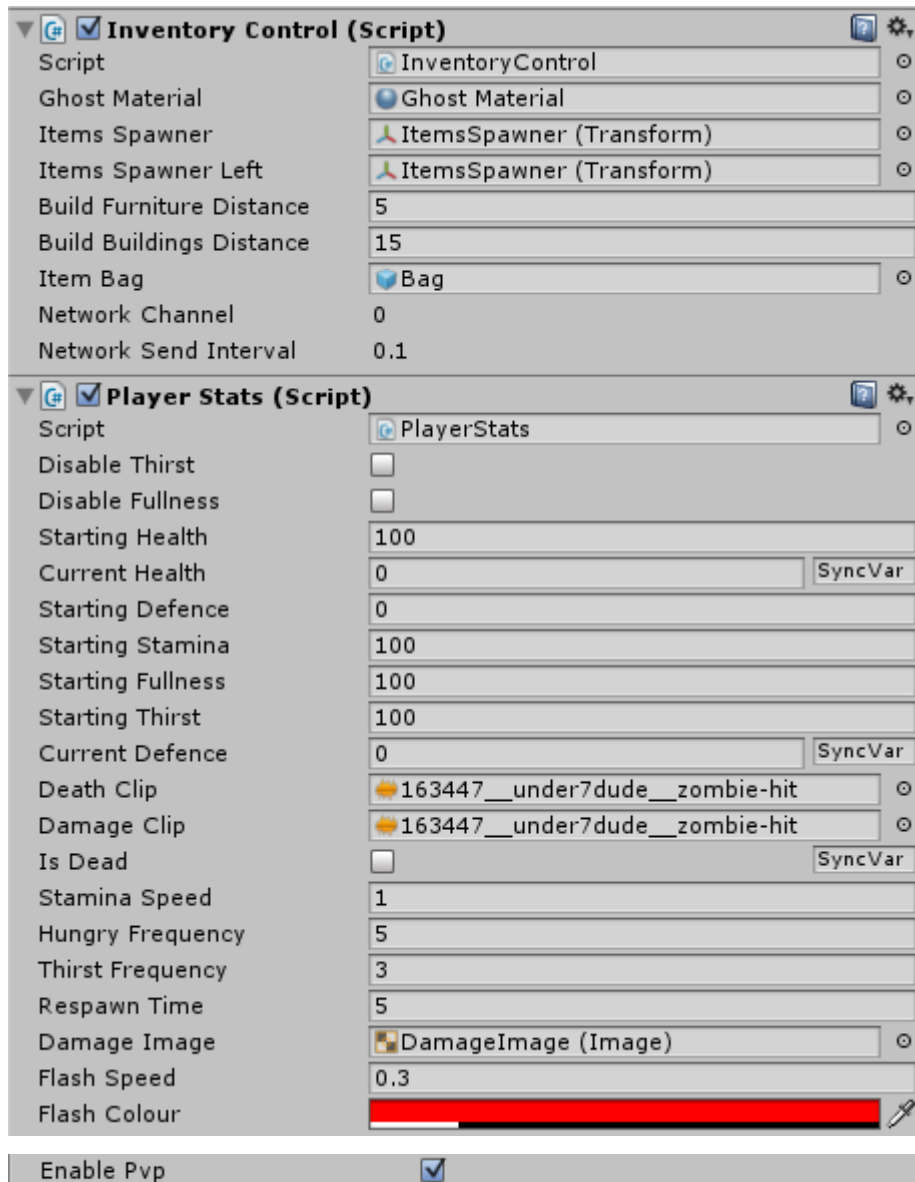
Distance Max – maximum distance for spawn mobs.

Distance Min – minimum distance for spawn mobs.

Mobs – list of mobs that will spawn.

Mob – mob's prefab.

Mobs Count – the maximum number of simultaneously existing in the world mobs of this type.



Inventory Control – script to manage inventory:

Ghost Material – the material that will be imposed on the "ghost" of Item Placeable Prefab. In our case the material is translucent, and code in script Player Controller controls its color.

Items Spawner – the point at which appears Item Prefabs selected by the player from Quick Panel, such as weapons. In our case it is special bone in player hands model.

Items Spawner Left – items spawner for left hand.

Build Furniture Distance – the maximum range of the construction of objects (such as chests or workbench).

Build Buildings Distance – the maximum range of the construction of buildings.

Item Bag – prefab of bag for items that drop into the world.

Player Stats – script to manage player stats:

Disable Thirst – disable thirst mechanic.

Disable Fullness – disable hunger mechanic.

Starting Health – the initial and maximum health of the character.

Starting Defence – the initial protection of the character.

Starting Stamina – the initial and maximum stamina of the character.

Starting Fullness – the initial and maximum fullness of the character.

Starting Thirst – the initial and maximum thirst of the character.

Death Clip – audio clip for character's death.

Damage Clip – audio clip for taking damage by character.

Stamina Speed – the speed multiplier that affects the rate of fall and recovery of stamina.

Hungry Frequency – the frequency of reducing the level of satiety (and cause damage to the character in the hunger).

Thirst Frequency – the frequency of reducing the level of thirst (and cause damage to the character in the thirst).

Respawn Time – the time that the character respawns.

Damage Image – the image that appears when the player taking damage.

Flash Speed – the rate of disappearance of Damage Image.

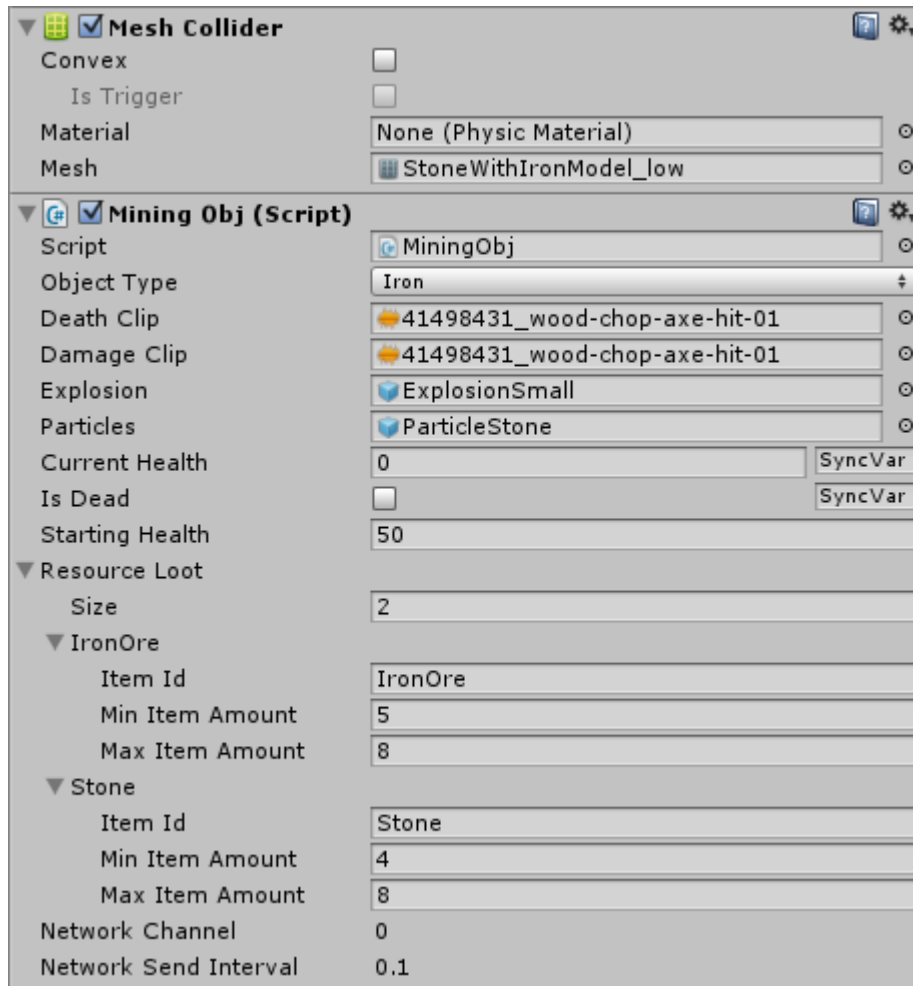
Flash Colour – color of Damage Image.

Enable Pvp – enable or disable PvP (Player vs Player) in multiplayer.

Player Controller:

This greatly rewritten First Person Character Controller script, there is code that is responsible for the construction, opening and use of the items in the world, and fatigue.

Creating an Object that can be mined



Add to your prefab **Collider**, **Network Identity**, **Audio Source** and **Mining Obj** script.

Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.

Mining Obj – script to determine the properties of the mining object.

Object Type – type of the mining object. None – can be mined by any weapon. Iron or Wood – can be mined only by a weapon with the type of mining Iron or Wood.

Death Clip – audio clip for object’s destruction.

Damage Clip – audio clip for taking damage by object.

Explosion – explosion’s prefab that appears when the object is destroyed.

Particles – particles prefab that appears when the object takes damage.

Starting Health – the health of the object.

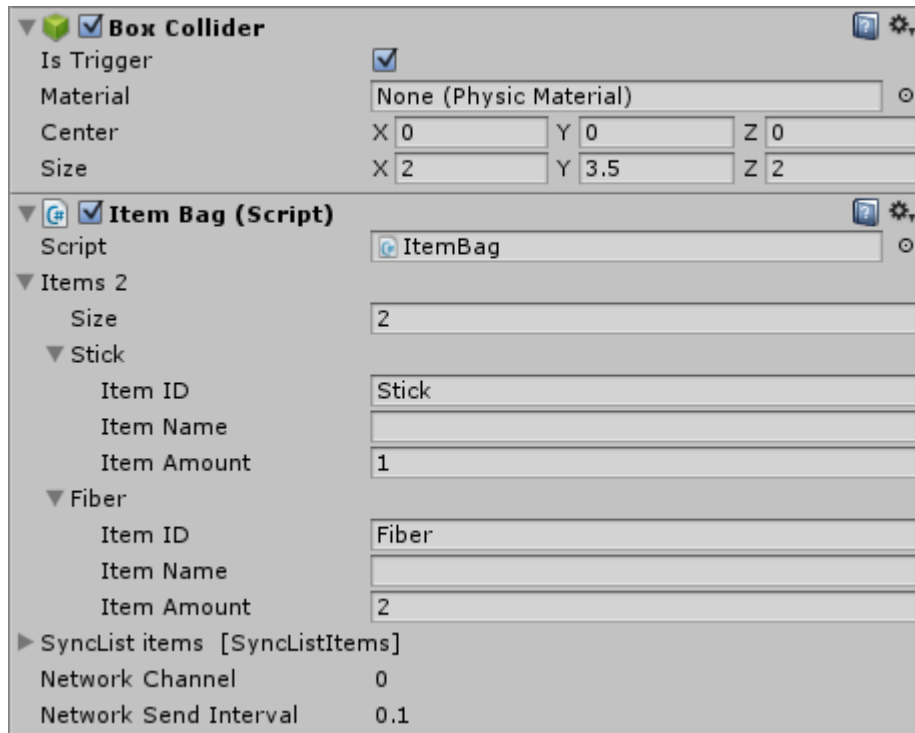
Resource Loot – a list of items received by the player when the object is destroyed.

Item ID – ID of received item.

Min Item Amount – minimum amount of received items.

Max Item Amount – maximum amount of received items.

Creating an Object that you can pick up



Add to your prefab **Collider, Network Identity, Item Bag** and **Game Tooltip** scripts.

Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.

Game Tooltip – the script that shows the hint on the screen when you move the sight to an object.

Tooltip – leave this field blank, then will be displayed a standard hint. Otherwise the tooltip will be the text from this field.

Item Bag – script to determine the properties of the picked object.

Items 2 – list of items received by the player when picking up the object.

Item ID – ID of received item.

Item Name – with this line you can change the name of the item displayed by the Game Tooltip script. Leave blank if you do not need it.

Item Amount – the number of received items.

Recipes Management

To manage crafting recipes select the Prefab "Database" in the scene.



Craft Database – script that allows you to add or remove crafting recipes.

Craft Database – a list of crafting recipes.

Item ID – ID of the created item.

Item Amount – the number of created items at a time.

Item Craft Time – the time required to create an item.

Subsection – subsection, under which there is a recipe in the crafting window.

Work Station – workstation required to create an item (Hands does not require a workstation).

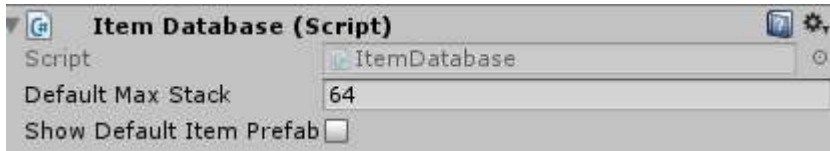
Required Items – list of items needed for crafting.

Item ID – ID of the required item.

Item Amount – the number of required items.

Items Management

Some properties of Item Database can be assigned through the Item Database script in the prefab "Database" in the scene.



Default Max Stack – the maximum number of items in single stack, determined for items without a configured parameter "MaxStack".

Show Default Item Prefab – show default prefab for all items that the player holds in hands and which do not have an Item Prefab.

To add or remove items, open the «Resources \ Items.json» file.

Add new items by adding in the file this structure:

```
{
  "ID": "Item ID",
  "Title": "Item Name",
  "Description": "Item Description",
  "ItemType": 0
},
```

Description of the parameters can be found in «Resources \ Readme.txt» file.

Notice that the last item should NOT have a comma after the closing brace, otherwise it will be a critical error:

```
,,
{
  "ID" : "Fat",
  "Title" : "Fat",
  "Description" : "Animal fat",
  "Stackable" : true,
  "ItemType": 7
},
{
  "ID" : "Bed",
  "Title" : "Bed",
  "Description" : "Warm and soft bed",
  "Stackable" : false,
  "ItemType": 7,
  "Placeable" : true
}
```

Creating an Item Prefab

Assign to your prefab the "Weapons" layer:



Add to your prefab **Network Identity** and **Item In Hands** script.



Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.

Item In Hands – script that makes the prefab a Item in hands.

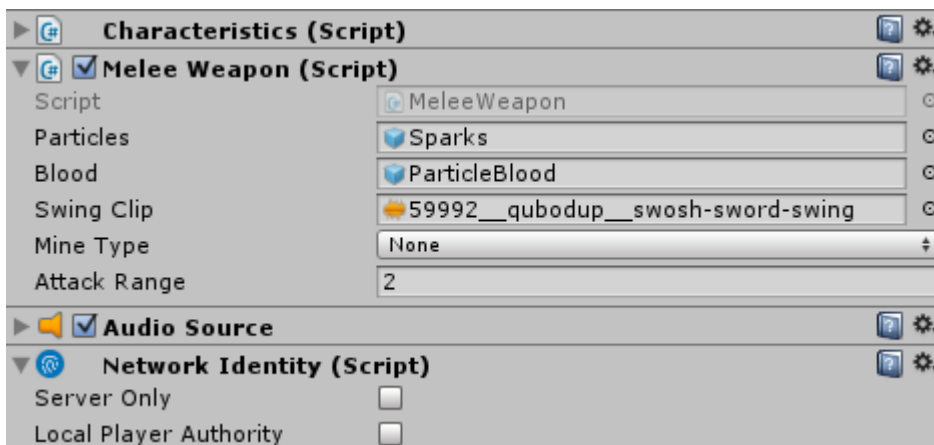
Animation Index – Index of animation that will be played with this item in hands (1 – bow, 2 – vertical weapon, 3 – horizontal weapon, 4 – food, 5 - gun).

Right Hand – this item will appear in the right hand.

Save Prefab in «Resources \ Item Prefabs». Prefab name must match the ID or Slug of item.

Melee Weapon:

Add to your prefab **Audio Source** and **Characteristics** and **Melee Weapon** scripts.



Configure Audio Source in the likeness of already present weapons.

Melee Weapon – script that makes the prefab a melee weapon.

Particles – prefab of particles that appears when you attack something.

Blood – prefab of blood that appears when you attack animal.

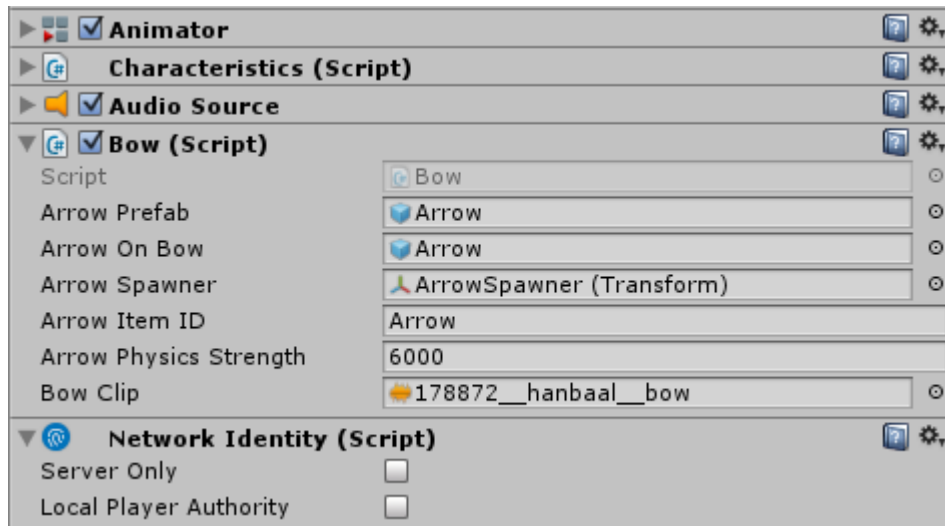
Swing Clip – the audio clip for the weapon swing.

Mine Type – the type of mining objects that you can destroy by this weapon.

Attack Range – attack range.

Ranged Weapon (Bow):

Add to your prefab **Audio Source**, **Animator** and **Characteristics** and **Bow** scripts.



Configure Animator and Audio Source in the likeness of already present bow.

Bow – script that makes the prefab a bow.

Arrow Prefab – prefab of arrow (explained later).

Arrow On Bow – an arrow on the bow (explained later).

Arrow Spawner – arrow spawner (explained later).

Arrow Item ID – ID of arrow, which is required for the shot and which will be consumed when fired.

Arrow Physics Strength – arrow's speed flight.

Bow Clip – Audio clip for shot from a bow.

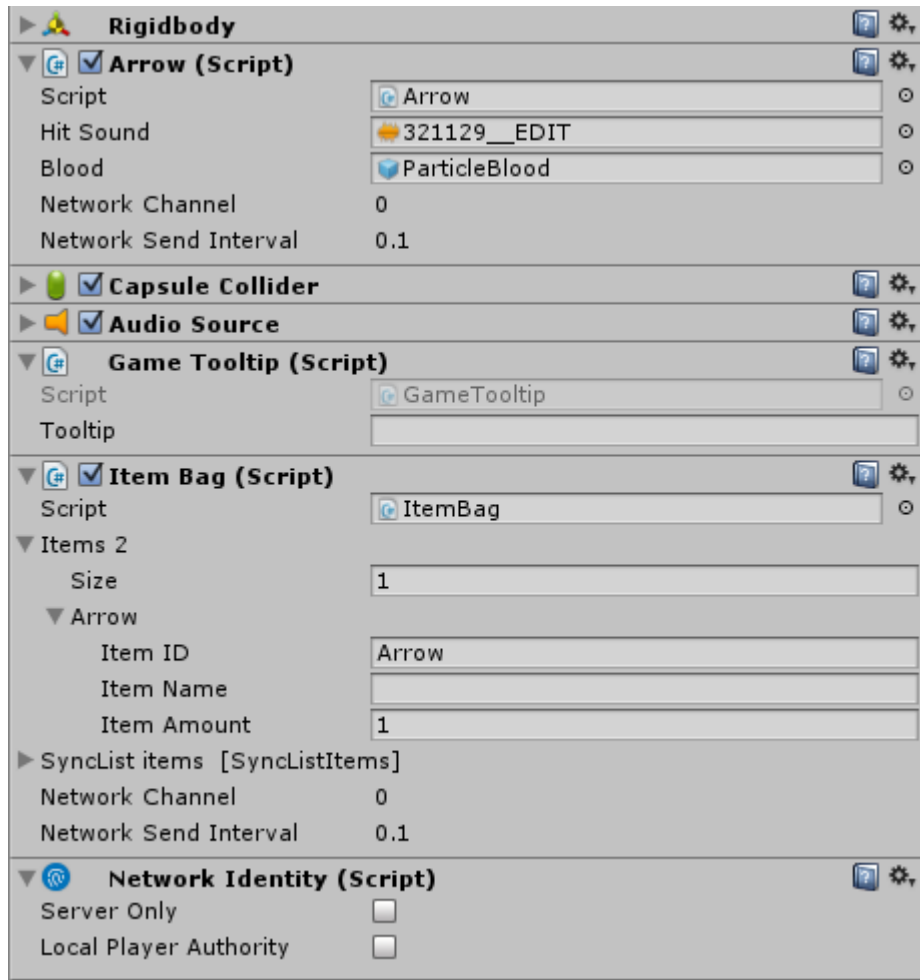
Creating an Arrow Prefab:

Assign to your prefab "Arrow" layer:



Add to it **Capsule Collider**, **Rigidbody**, **Audio Source**, **Network Identity**, **Arrow** script and, optionally, **Item Bag** + **Game Tooltip**.

Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.



Configure Rigidbody, Capsule Collider, Audio Source and Item Bag in the likeness of already present arrow.

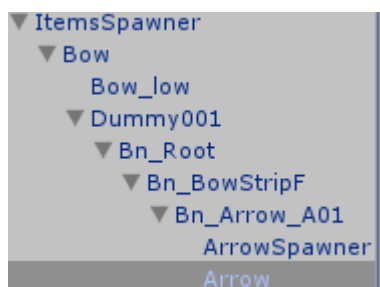
Arrow – script that makes the prefab an arrow.

Hit Sound – audio clip for arrow's hit.

Blood – prefab of blood that appears when arrow hit animal.

Creating an "Arrow On Bow":

Attach the model of arrow to the special bone of your bow:



Adjust its position:

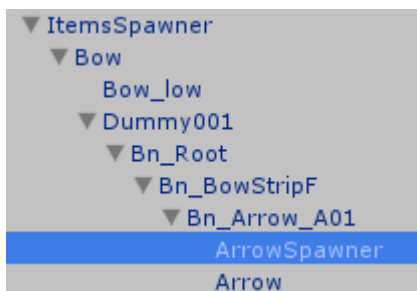


Disable Mesh Renderer:



Creating an "Arrow Spawner":

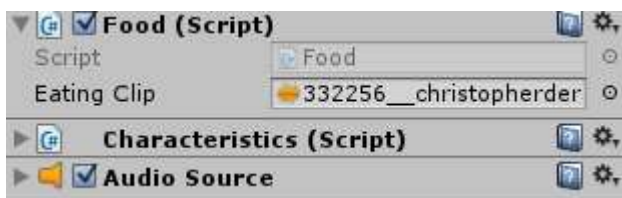
Create Empty and attach it to special bone of your bow:



Adjust the rotation and position of the subject as you need. In this place will appear Arrow Prefab.

Food and Drink:

Add to your prefab **Audio Source** and **Characteristics** and **Food** or **Drink** scripts.



Creating an Armor

Create model in the likeness of already present armors.

If your prefab is gloves, assign to it the "Weapons" layer:



Add to your prefab **Network Identity** and **Armor** script.



Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.

Save Prefab in «Resources \ Armor Prefabs». Prefab name must match the ID or Slug of item.

Creating a Placeable Prefab



Add to your prefab **Collider**, **Rigidbody**, **Network Identity** and **Build Collision Properties** script.

Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.

Optionally, add Mining Obj + Audio Source, or Item Bag + Game Tooltip. You can don't add, but then the object will not be able to destroy or take.

Save the Prefab in «Resources \ Item Placeable Prefabs». Prefab name must match the ID or PlaceableSlug of item.

Build Collision Properties – script to determine the properties of the object, which can be constructed.

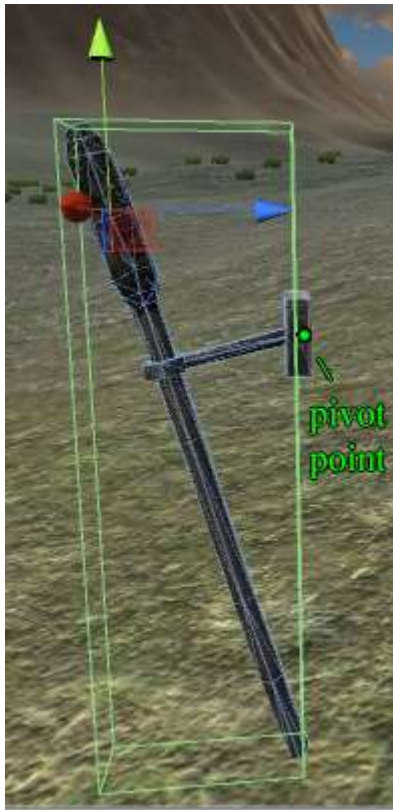
Can Objects Be Placed On This – check the box if other objects can be built inside it or on it.

Is Building – check the box if the object is a building (increases the range of construction of this object).

Setting the attachment point:

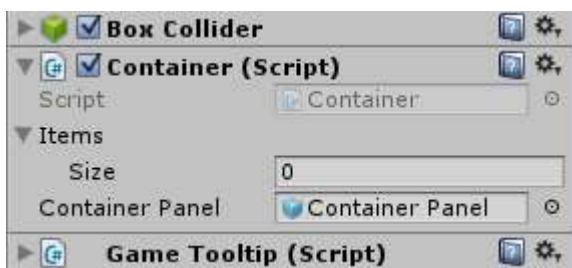
To create an attachment point for your model, put there a Pivot Point. Collider should not intersect the attachment point.

For wall torch it will look like this:



Or create an Empty GameObject, put your prefab and set it the desired position. (For example, if you want to put on the floor a cube with the pivot point in the center of cube, place it in Empty and move at half its height upwards).

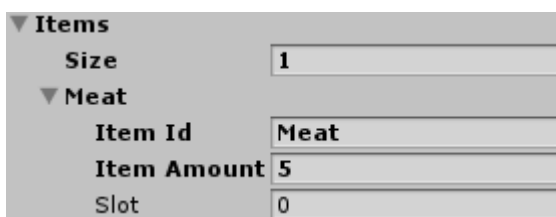
Creating a Container



Add to your prefab **Collider**, **Network Identity** and **Container** and **Game Tooltip** scripts.

Container – script that makes the prefab a container.

Items – if the container is in the scene, you can add items:



Slot – the number of slot that will contain the item.

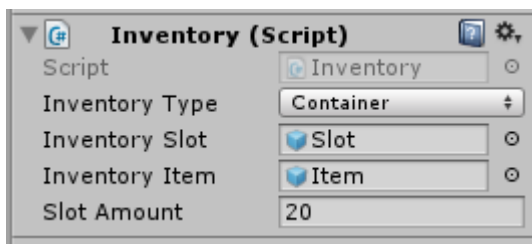
Container Panel – GameObject – container panel.

Creating a Container Panel:

Go to «*Prefabs \ UI \ Container Panels*», copy one of the panels. Drag it onto the scene, set the width and height as you like. Change the text in the Title Panel and an external view of the panel, if needed.

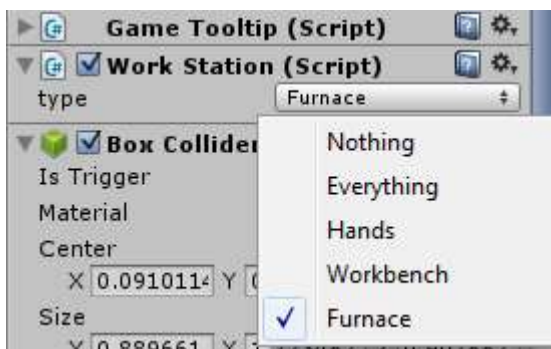


In the **Inventory** script set the number of slots as you like:



Save the panel in «*Prefabs \ UI \ Container Panels*» under any name and use in a script Container of your container.

Creating a Workstation



Add to your prefab **Collider** and **Work Station** and **Game Tooltip** scripts.

Work Station – script that makes the prefab a workstation.

type – Workstation type. Select one or more options (for workbench it will be a tick on Hands and Workbench).

Adding new Workstation types:

Open «*Scripts \ UI \ Inventory \ CraftPanel.cs*».

Find the lines:

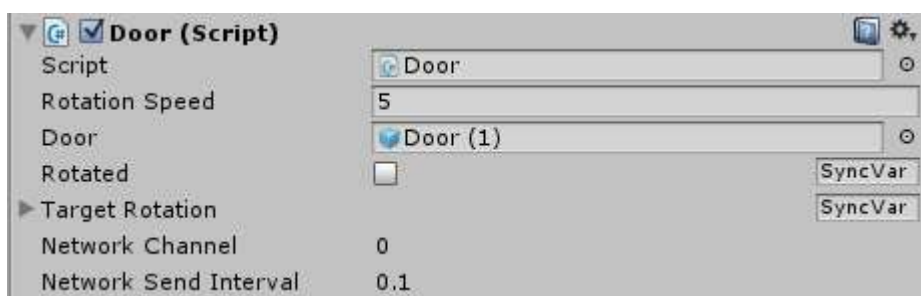
```
[System.Flags]
9 references
public enum workstations : byte
{
    Hands = 1,
    Workbench = 2,
    Furnace = 4
}
```

Add new workstations as follows:

```
public enum workstations : byte
{
    Hands = 1,
    Workbench = 2,
    Furnace = 4,
    Anvil = 8,
    MagicWorkbench = 16
}
```

Creating a door

Add to your door **Collider** and **Game Tooltip** script and to door's parent **Network Identity** and **Door** script.

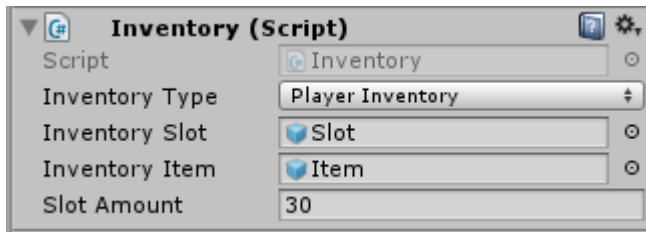


Door – script that makes the prefab a door's parent.

Rotation Speed – the speed at which the door opens and closes.

Door – door's prefab.

Editing Inventory Panels



Inventory – script to determine the properties of the inventory panel.

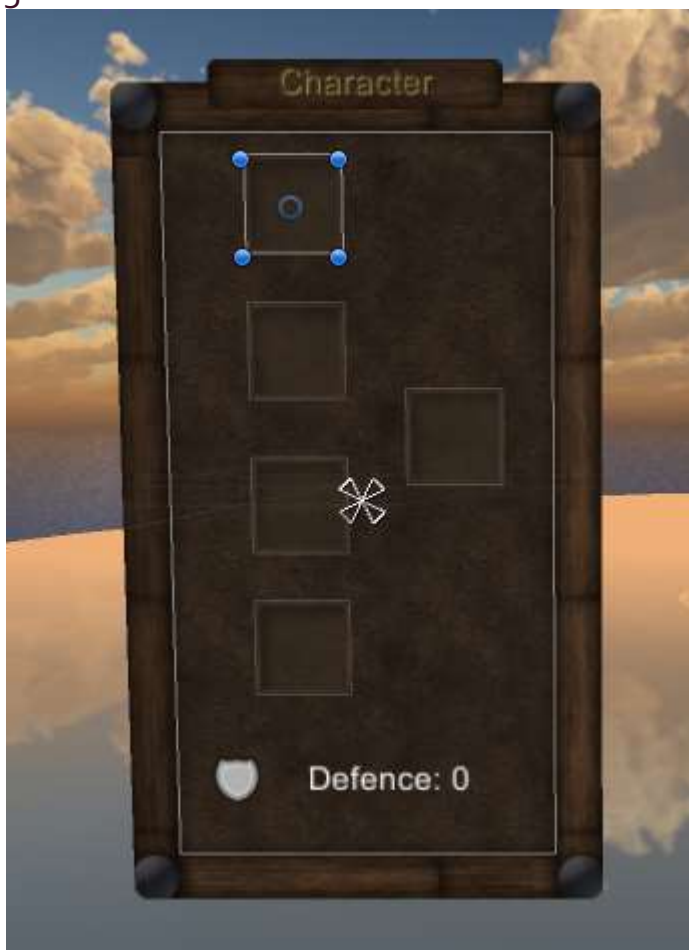
Inventory Type – the type of inventory (Player Inventory – the main inventory, Quick Panel - the quick access toolbar, Container – the container's inventory, Character Panel – panel of character's armor).

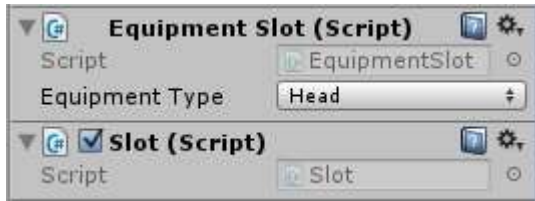
Inventory Slot – slot's prefab.

Inventory Item – item's prefab.

Slot Amount – the number of slots.

Editing Character Panel:





Create new slots for equipment in the likeness of the existing ones. Assign them desired position. In the Equipment Slot script put the right type of equipment for the slot.

Add new types of equipment in the script «*Scripts \ UI \ Inventory \ ItemDatabase.cs*».

```
public enum ItemTypeEnum {
    Torso,
    Head,
    Legs,
    Feet,
    Gloves,
    Weapon,
    Consumable,
    Other
}
```

For example, like this:

```
public enum ItemTypeEnum {
    Torso,
    Head,
    Legs,
    Feet,
    Gloves,
    Weapon,
    Consumable,
    Other,
    Cloak,
    Ring,
    Neck
}
```

Creating a new tabs for Crafting Panel:

Open «*Scripts \ UI \ Inventory \ CraftPanel.cs*».

Find the lines:

```
-----
public enum subsections
{
    Armor,
    Weapon,
    Tools,
    Materials,
    Containers,
    Food,
    Furniture
}
```

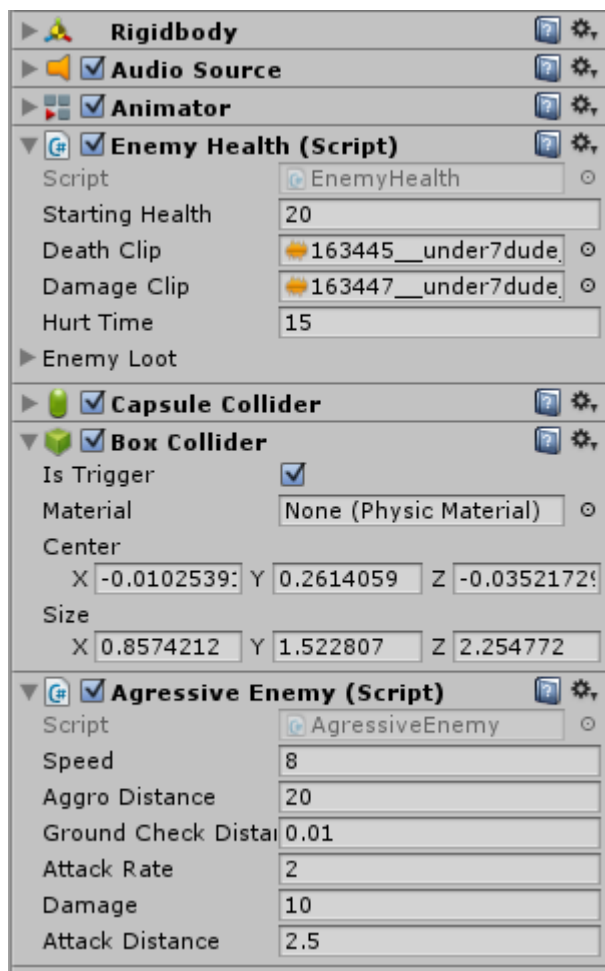
Add new tabs as follows:

```

public enum subsections
{
    Armor,
    Weapon,
    Tools,
    Materials,
    Containers,
    Food,
    Furniture,
    Potions,
    Magic
}

```

Creating an Agressive Mob



Add to your prefab **Rigidbody**, **Audio Source**, **Animator**, **Capsule Collider**, **Network Identity**, **Network Transform**, Collider-trigger for determining hits, **Enemy Health** and **Agressive Enemy** scripts.

Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.

Configure the animations and animator as well as an existing wolf.

Assign it the "Enemy" tag:



Enemy Health – script to determine the properties of the mob's health.

Starting Health – the initial and maximum health of the mob.

Death Clip – audio clip for mob's death.

Damage Clip – audio clip for taking damage by mob.

Hurt Time – the time that mob will be chasing the player (or running away) after receiving damage.

▼ Enemy Loot	
Size	2
▼ Leather	
Item Id	Leather
Min Item Amou	3
Max Item Amou	6
▼ RawMeat	
Item Id	RawMeat
Min Item Amou	2
Max Item Amou	4

Enemy Loot – list of items that drop from the mob after his death.

Aggressive Enemy – script to determine the properties of the aggressive mob.

Speed – mob's movement speed.

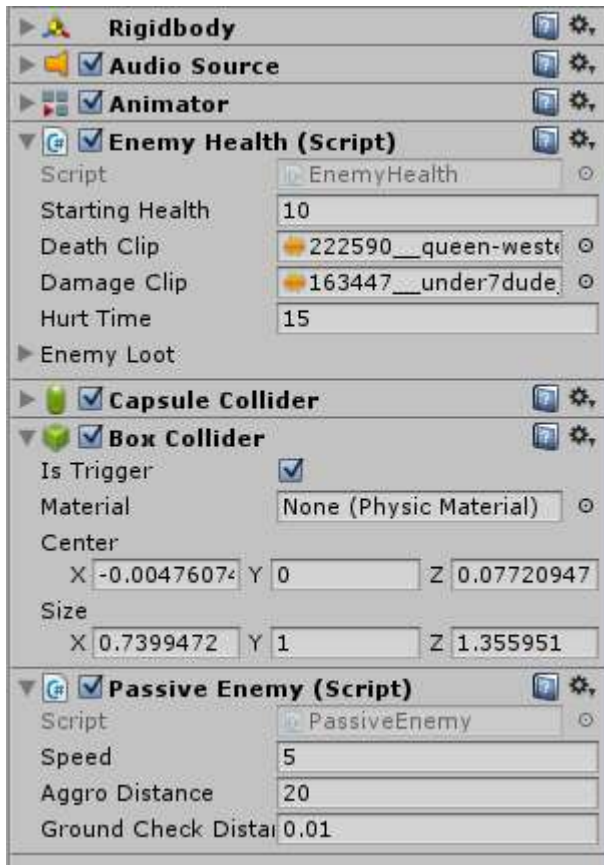
Aggro Distance – the distance at which the mob sees the player.

Attack Rate – attack rate in seconds.

Damage – the amount of damage per hit.

Attack Distance – attack range.

Creating a Passive Mob



Add to your prefab **Rigidbody**, **Audio Source**, **Animator**, **Capsule Collider**, **Network Identity**, **Network Transform**, Collider-trigger for determining hits, **Enemy Health** and **Passive Enemy** scripts.

Register prefab as Spawnable Prefab in Offline Scene (Menu) -> Network -> Network Manager Changed -> Spawn Info -> Registered Spawnable Prefabs.

Configure the animations and animator as well as an existing rabbit.

Assign it the "Enemy" tag:

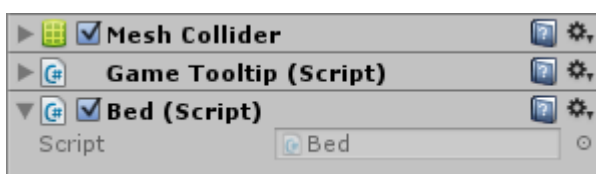


Passive Enemy – script to determine the properties of the passive mob.

Speed – mob's movement speed.

Aggro Distance – the distance at which the mob sees the player.

Creating a Spawn Point (Bed)



Add to your prefab **Collider** and **Bed** and **Game Tooltip** scripts.

Creating a Growing Object (Tree sapling or other plant)

Add to your prefab **Grow** script.



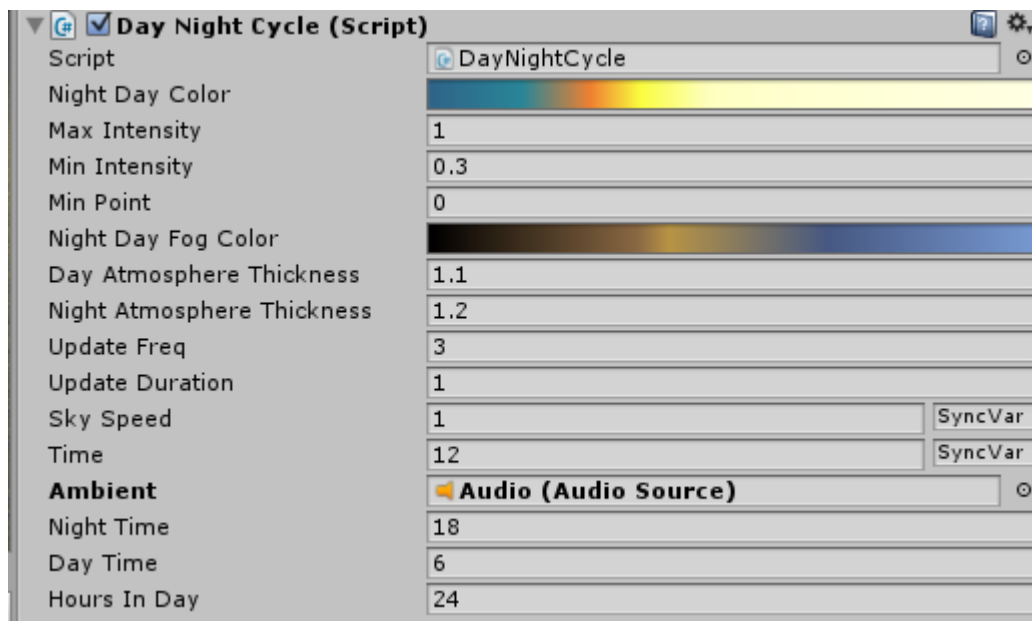
Grow – script for growing object.

Time – time in seconds after which object will grow.

Grown Obj – previous object will be replaced by this prefab.

Day and Night Cycle

Everything associated with Day and Night Cycle is located in "SunLight" prefab in the "Main" scene.



Day Night Cycle – script for Day and Night Cycle.

Night Day Color – gradient of color of light during the day.

Max Intensity – the maximum intensity of light during the day.

Min Intensity – the minimum intensity of light during the night.

Min Point – height of sunrise and sunset.

Night Day Fog Color – gradient of color of fog during the day.

Day Atmosphere Thickness – atmosphere thickness during the day.

Night Atmosphere Thickness – atmosphere thickness during the night.

Update Freq – rotation frequency of the sky in seconds.

Update Duration – rotation duration of the sky in seconds.

Sky Speed – speed of sky rotation.

Time – current time of day.

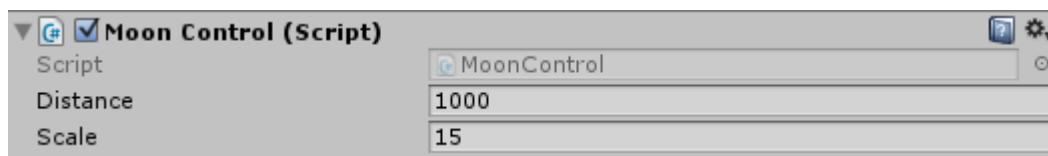
Ambient – Audio Source with sound of singing birds..

Night Time – a time when the night comes.

Day Time – a time when the day comes.

Hours In Day – the number of hours in a day.

Script, which controls the size of moon are in "Moon" prefab.



Moon Control – script, which controls the size and distance of moon.

Changelog

V 1.1b:

- Added Day and Night Cycle.
- Added PvP.
- Changed TempText.cs
- Added pop-up messages when picked up item.
- Added script for trees growing and "Sapling" item.
- Fixed some bugs in singleplayer and multiplayer.

V 1.1a:

- Main Menu scene modified
- Changed water can description.
- Items.json file is moved to the Resources folder, StreamingAssets folder is deleted, ItemDatabase.cs script changed. All of this made for easier transfer template on mobile platforms.
- In both scenes added "Audio" Prefab (A copy is located in "Scene \ ScenePrefabs") with environment sounds.
- Added aiming for gun - changed Gun prefab, script Firearm.cs, and in Player prefab removed "Use Fov Kick" in script "Player Controller" in order to avoid bugs with camera.

V 1.1:

New:

- Main menu.
- Multiplayer (Beta)
- Character hands in first person mode.
- Character model in multiplayer.
- Armor displaying.

- Item - water and thirst mechanic.
- Rifle and shooting mechanic.
- Blood and hit particles.
- Hit sounds.
- Craft timer.
- Craft time in recipes.
- And other less important features.

Changed:

- Almost all of the scripts.
- All prefabs.

**Gnome's
Artworks**