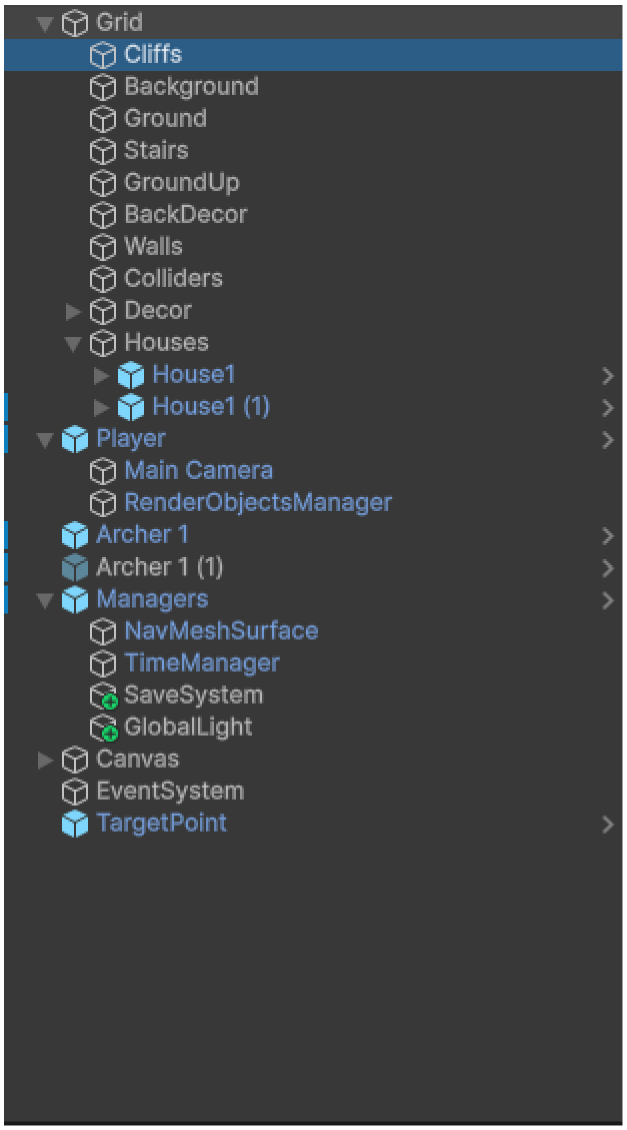
Map and Houses

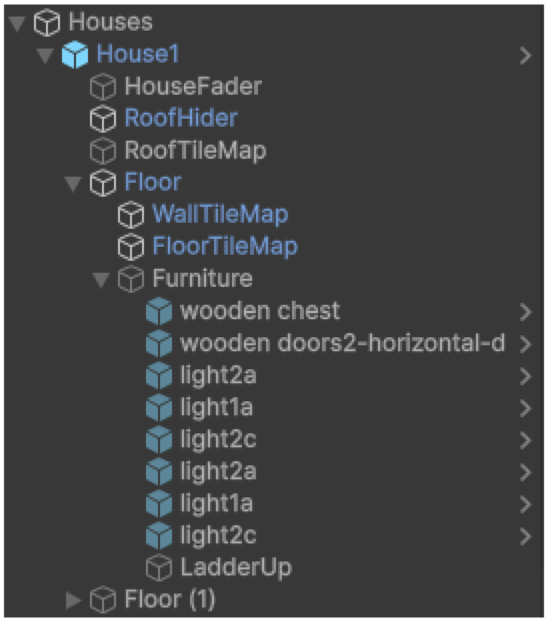


```
Grid
    Cliffs
    Background
    Ground
    Stairs
    GroundUp
    BackDecor
    Walls
    Colliders
    Decor
    Houses
        House1
        House1 (1)
Player
    Main Camera
    RenderObjectsManager
Archer 1
Archer 1 (1)
Managers
    NavMeshSurface
    TimeManager
    SaveSystem
    GlobalLight
Canvas
EventSystem
TargetPoint
```

1) **Cliffs** – Used to draw cliffs that act as the boundary of the playable area (i.e., the edge of the world).
2) **Background** – Used for drawing water or other tiles that appear *below* the main ground level, creating a depth effect.
3) **Ground** – Used for regular ground tiles.
4) **Stairs** – Used for placing stairs that the player can automatically ascend.
5) **GroundUp** – Similar logic to **Background**, but used for elevation effects above ground.
6) **BackDecor** – Decorative objects that are rendered *behind* the player.
7) **Walls** – Impassable wall tiles.
8) **Colliders** – Invisible walls used to block movement.
9) **Decor** – Decorative elements rendered *above* the player.

a) You can also add animated prefab child objects.

10) **Houses** – Acts as a container for buildings.



```
▼ Houses
  ▼ House1                              >
      HouseFader
      RoofHider
      RoofTileMap
    ▼ Floor
        WallTileMap
        FloorTileMap
      ▼ Furniture
          wooden chest                  >
          wooden doors2-horizontal-d    >
          light2a                       >
          light1a                       >
          light2c                       >
          light2a                       >
          light1a                       >
          light2c                       >
          LadderUp
    ▶ Floor (1)
```

a) **RoofTileMap** – Represents the house roof, which hides automatically when the player enters the house.

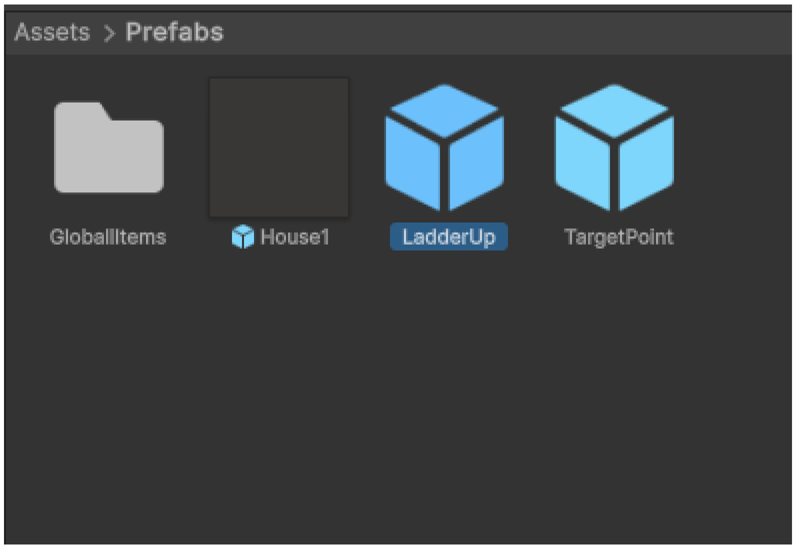
b) **Floor** – A container for each floor of the house, containing:

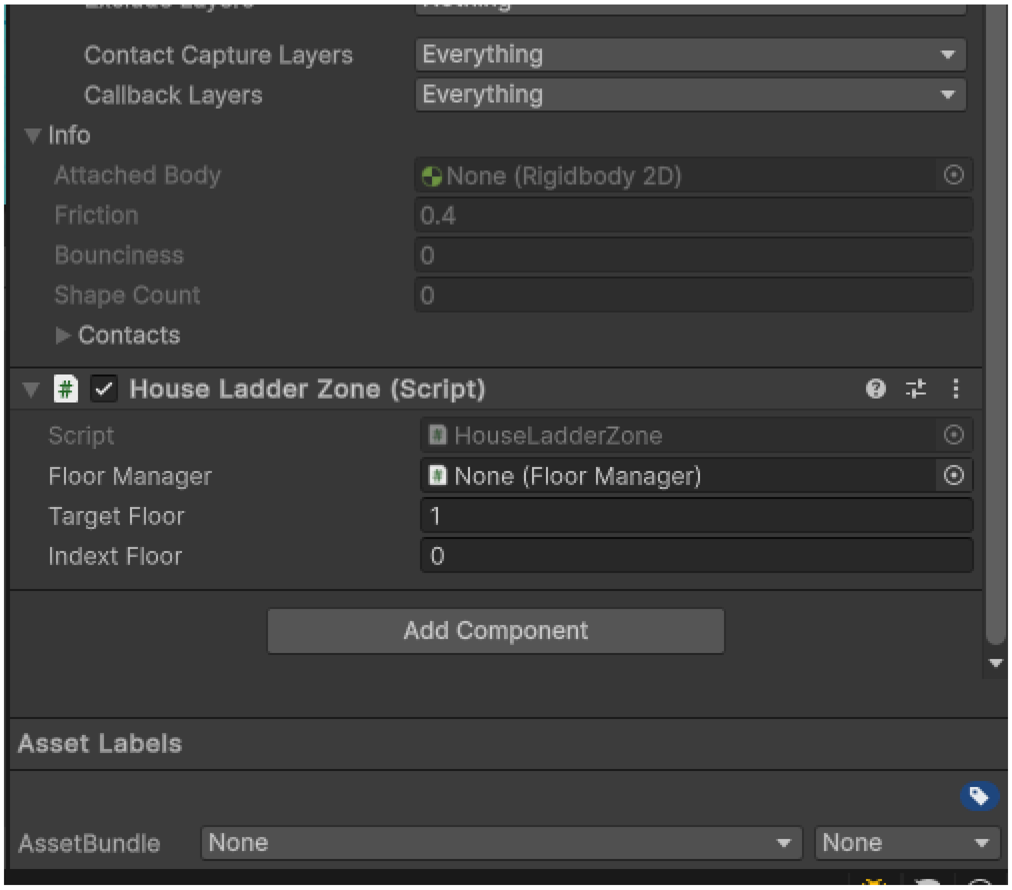
i)**WallTileMap** – House walls.

ii)**FloorTileMap** – House floor.

iii)**Furniture** – For decorations (you can also add animated prefab children).

iv)For multi-story buildings, each floor must include a **Ladder** to allow the player to move between floors.



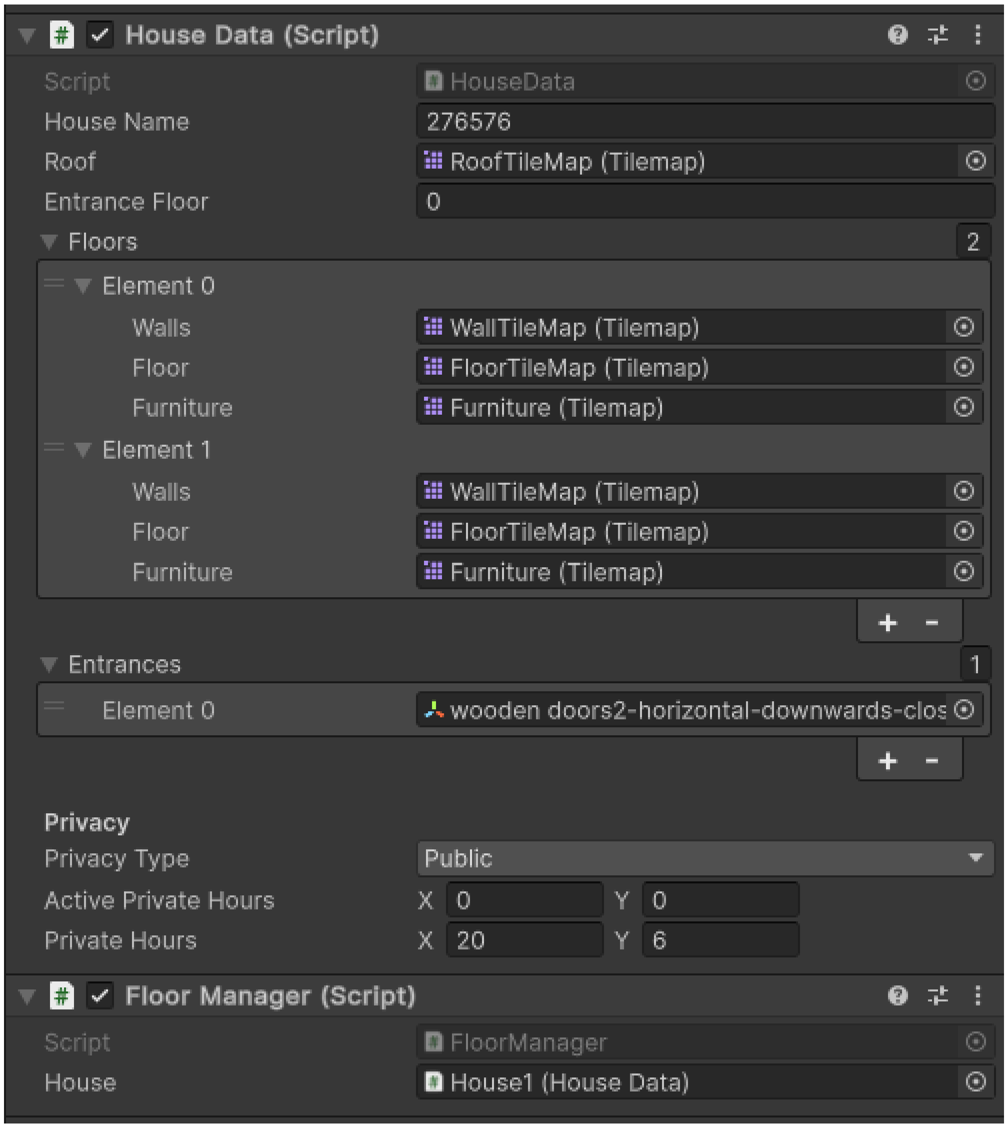

Contact Capture Layers: Everything
Callback Layers: Everything
Info
Attached Body: None (Rigidbody 2D)
Friction: 0.4
Bounciness: 0
Shape Count: 0
Contacts
House Ladder Zone (Script)
Script: HouseLadderZone
Floor Manager: None (Floor Manager)
Target Floor: 1
Indext Floor: 0

AssetBundle: None, None

Each **Ladder** must have a `HouseLadderZone` component with the following settings:

- **FloorManager** – The house this ladder belongs to.

- **TargetFloor** – The floor this ladder leads to.
- **IndexFloor** – The current floor the ladder is on (starting from 0).



House Data (Script)

Script	HouseData
House Name	276576
Roof	RoofTileMap (Tilemap)
Entrance Floor	0
Floors	2
Element 0	
Walls	WallTileMap (Tilemap)
Floor	FloorTileMap (Tilemap)
Furniture	Furniture (Tilemap)
Element 1	
Walls	WallTileMap (Tilemap)
Floor	FloorTileMap (Tilemap)
Furniture	Furniture (Tilemap)
Entrances	1
Element 0	wooden doors2-horizontal-downwards-clos
Privacy	
Privacy Type	Public
Active Private Hours	X 0 Y 0
Private Hours	X 20 Y 6

Floor Manager (Script)

Script	FloorManager
House	House1 (House Data)

Each house has a **HouseData** object containing:

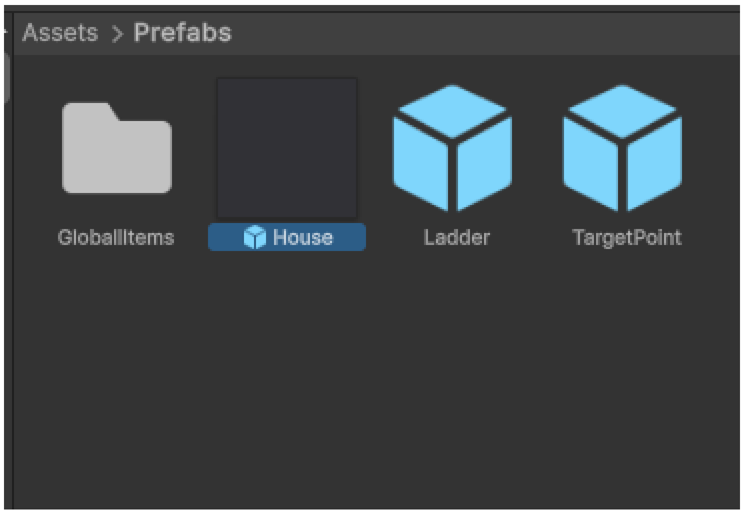
- **Roof** – The roof object.
- **EntranceFloor** – The index of the main entrance floor (e.g., set to 1 if the ground floor is considered a basement).

- **Floors** – A collection of all the house's floors.
- **Entrances** – Entry points into the house.

Privacy Settings

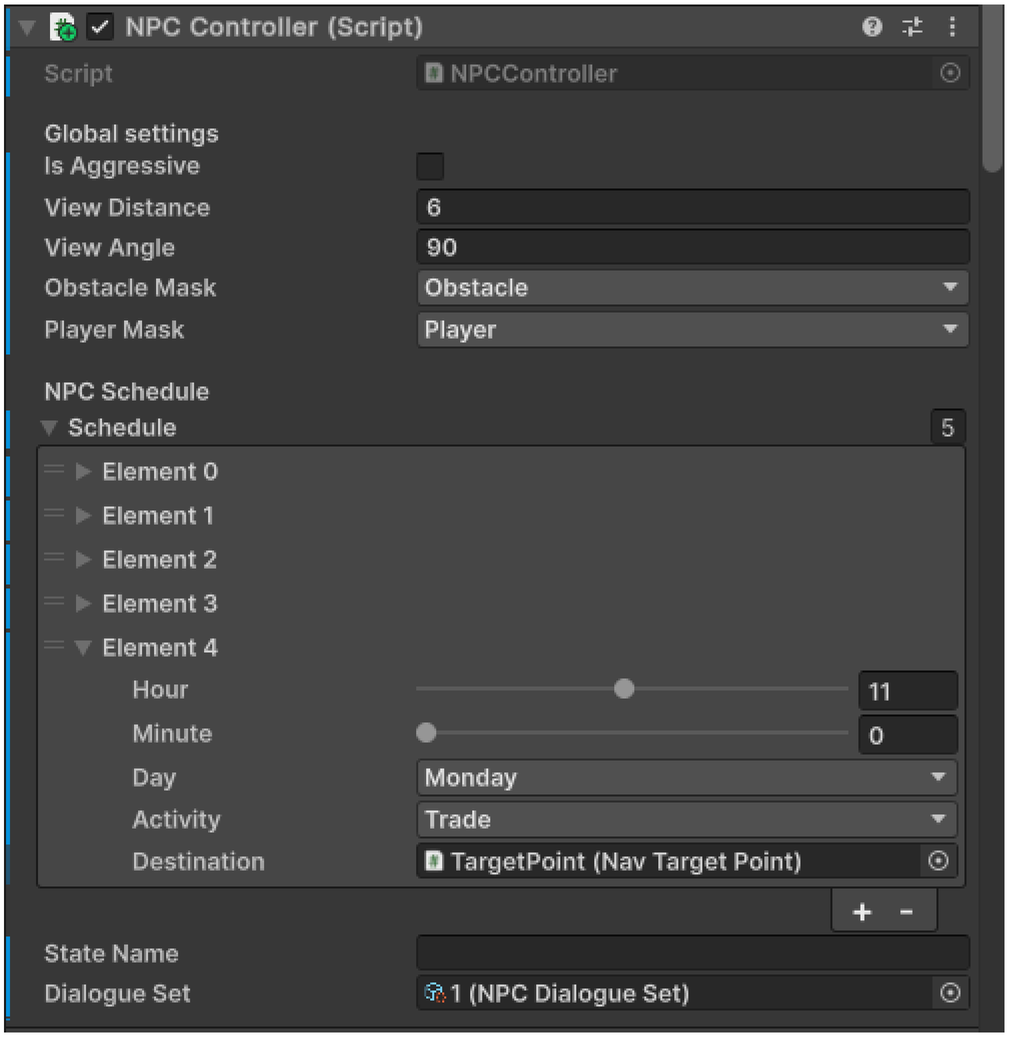
- Used to configure whether a house is public or private.
- `PrivacyType`: `Public`, `Private`, or `PrivateScheduled`.
- `PrivateHours`: The time range during which the house is considered private (used only with `PrivateScheduled`).

You can use prefabs from the **Prefabs** folder, where most settings are already configured.



NPCs and Schedules

NPC behavior is controlled by the `NPCController` script.


NPC Controller (Script)

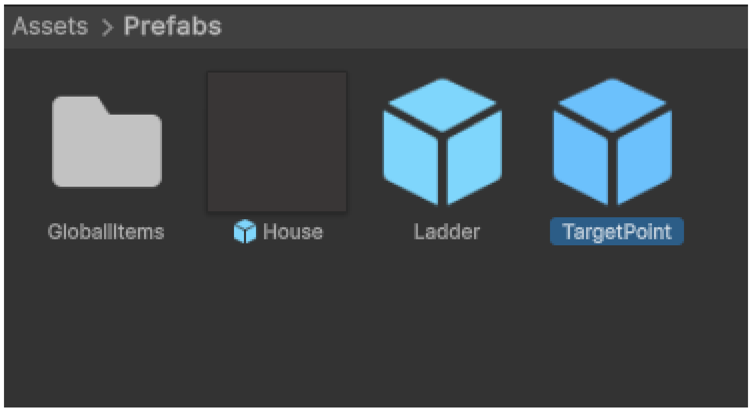
Script	NPCController
Global settings	
Is Aggressive	
View Distance	6
View Angle	90
Obstacle Mask	Obstacle
Player Mask	Player
NPC Schedule	
Schedule	5
Element 0	
Element 1	
Element 2	
Element 3	
Element 4	
Hour	11
Minute	0
Day	Monday
Activity	Trade
Destination	TargetPoint (Nav Target Point)
State Name	
Dialogue Set	1 (NPC Dialogue Set)

a)
b) **isAggressive** – If enabled, the NPC will chase the player as soon as they are spotted. (This will likely be reworked in the “Factions & Reputation” phase.)
c) **View Distance** – NPC vision radius.
d) **View Angle** – NPC field of view angle.
e) **Obstacle & Player Mask** – Generally do not need to be changed.

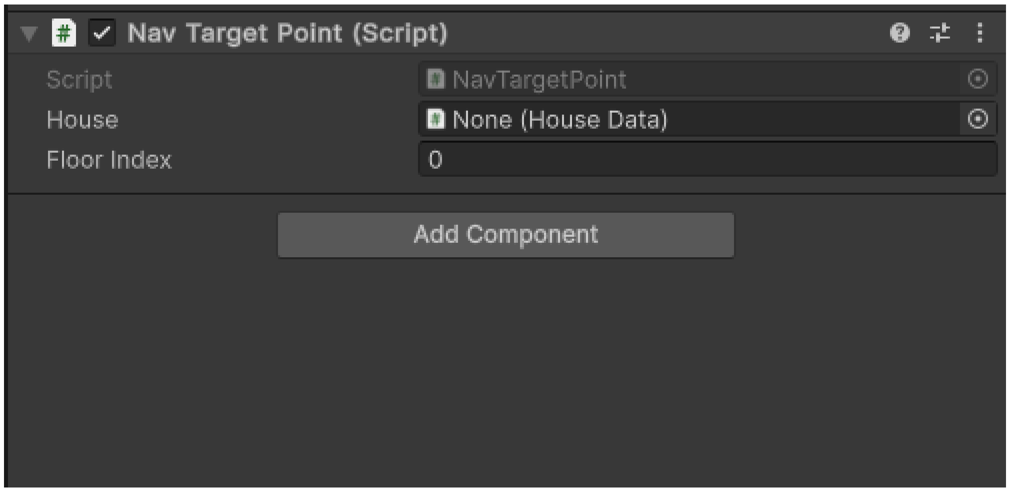
Schedule – A list of the NPC’s daily activities:

- **Hour/Minute** – When the event starts.
- **Day** – Day of the week. If events repeat daily, duplicate them for each day.
- **Activity** – The type of action performed at that time:
  - `Sleep` – NPC is sleeping.
  - `Work` – NPC is working.
  - `Idle` – NPC stands in place.
  - `Wander` – NPC roams randomly.

    - `Trade` – NPC is available for trading.
    - `Guard` – NPC guards a location (must define the position).
    - `Patrol` – Like Guard, but moves around a route.
    - `Hide` – NPC is hidden and cannot be interacted with.
    - `Hunt` – NPC roams and will chase the player upon spotting them.
    - `Chill` – NPC casually wanders near a point within a small radius.
- **TargetPoint** – An invisible dummy point used for pathfinding (created as a prefab for easier reuse).



Each point has the following additional data if placed inside a house:


Nav Target Point (Script)

Script: NavTargetPoint
House: None (House Data)
Floor Index: 0

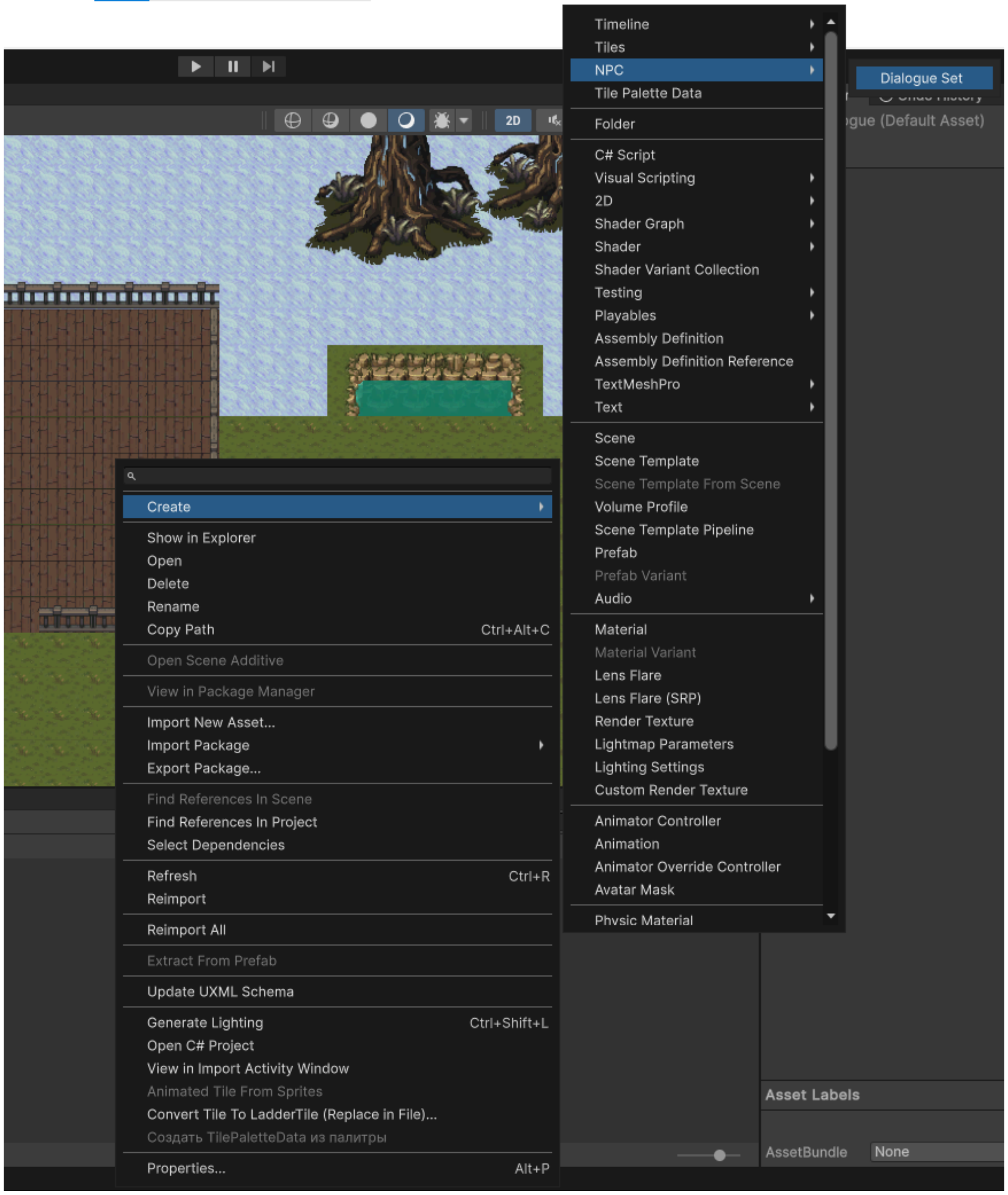
House – The house it belongs to.

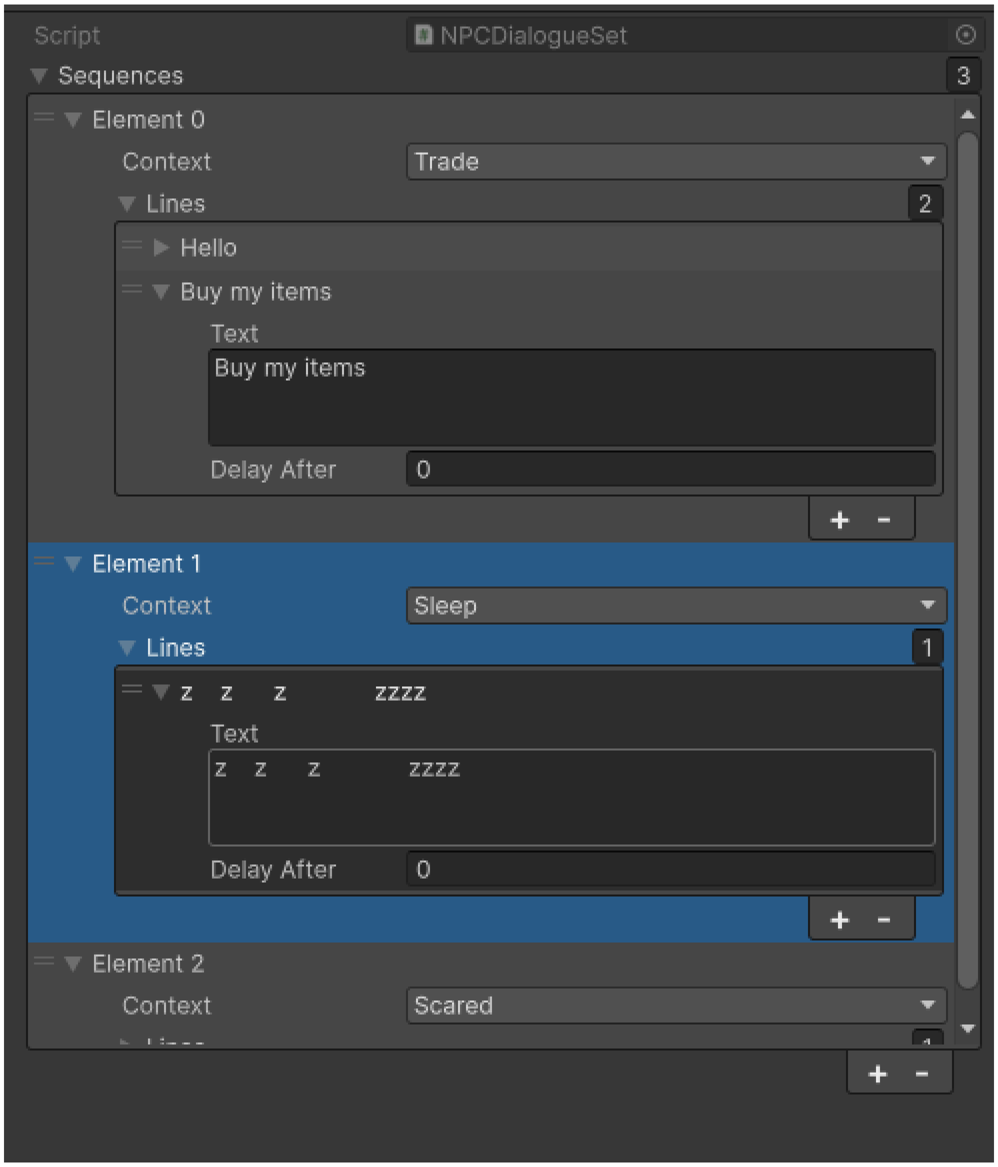
FloorIndex – Which floor the point is on.
(If the point is outside, these fields can be ignored.)

StateName – Used for debugging.

DialogueSet – A set of phrases that the NPC says periodically depending on their current state. For example, if trading, they might say "Come buy something!"

To create a dialogue set:

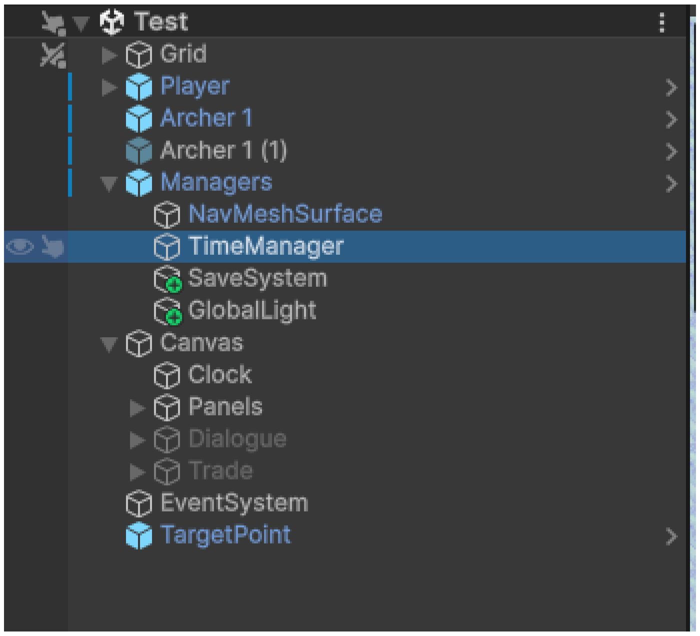



Script: NPCDialogueSet
Sequences: 3
Element 0
Context: Trade
Lines: 2
Hello
Buy my items
Text: Buy my items
Delay After: 0
Element 1
Context: Sleep
Lines: 1
z z   z   zzzz
Text: z z   z   zzzz
Delay After: 0
Element 2
Context: Scared

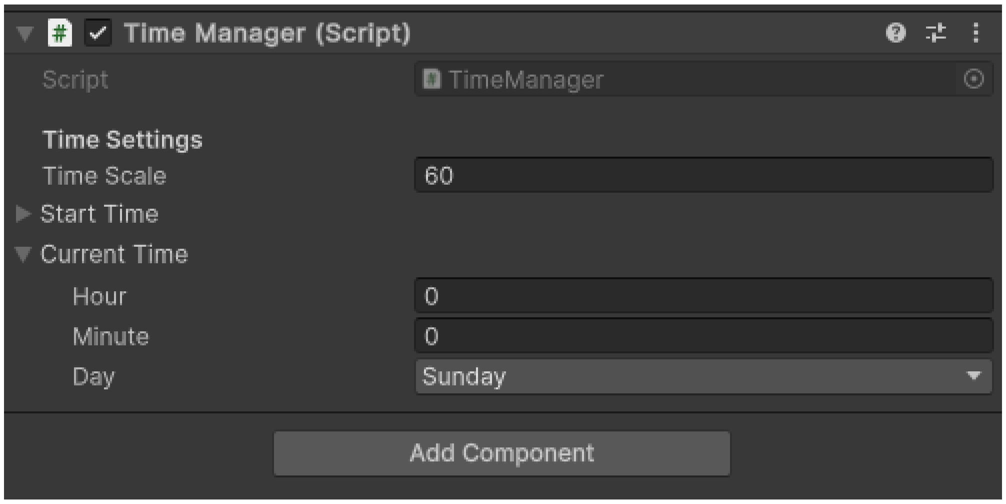
- **Context** – The state in which the lines are used. Matches the NPC activity types:
    - `Idle` – Also used for `Wander` and `Chill`.
    - `Work`
    - `Trade`
    - `Guard` – Also used for `Patrol`.
    - `Sleep`
    - `Hunt`
    - `Scared`
- **Lines** – The lines of dialogue spoken in that state.

This field is optional – you can leave it empty or reuse sets between NPCs.

Global settings


Test
Grid
Player
Archer 1
Archer 1 (1)
Managers
NavMeshSurface
TimeManager
SaveSystem
GlobalLight
Canvas
Clock
Panels
Dialogue
Trade
EventSystem
TargetPoint

Under **Managers**, the `TimerManager` controls in-game time.

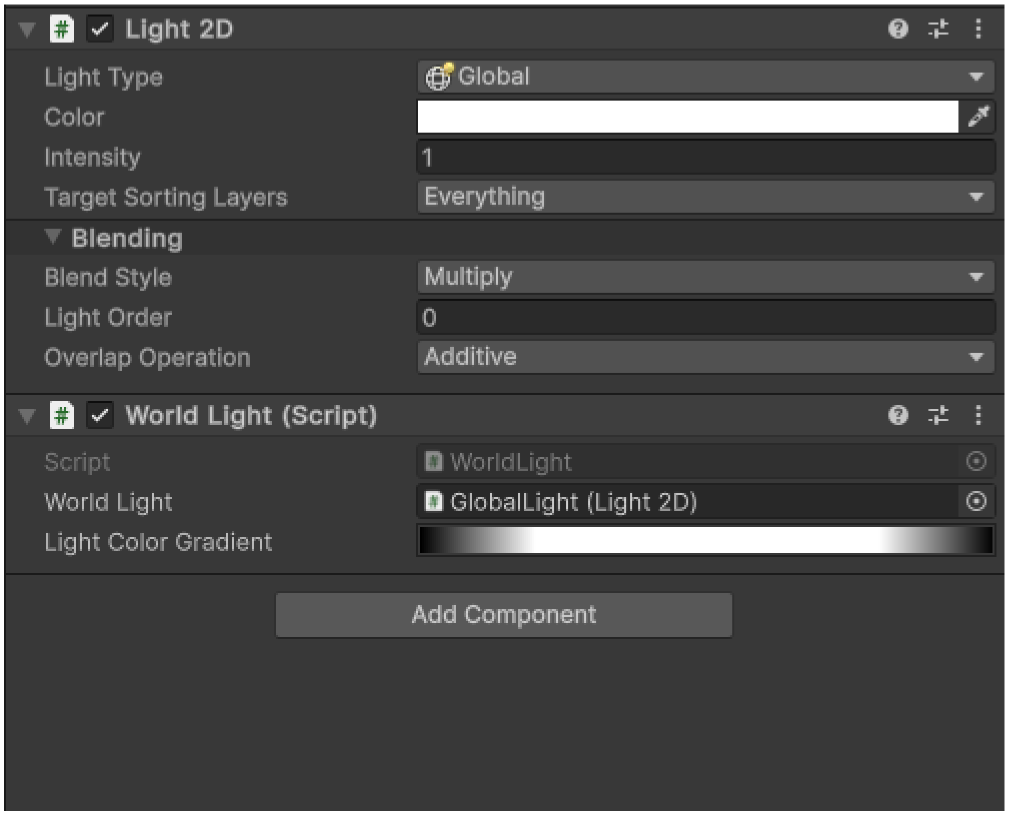

Time Manager (Script)

Script: TimeManager

Time Settings
Time Scale: 60
Start Time
Current Time
Hour: 0
Minute: 0
Day: Sunday

Time Scale – Determines how fast in-game time passes compared to real time (e.g., how many real seconds equal one in-game second).

The **GlobalLight** object controls lighting.


Light 2D

Light Type	Global
Color	
Intensity	1
Target Sorting Layers	Everything

Blending

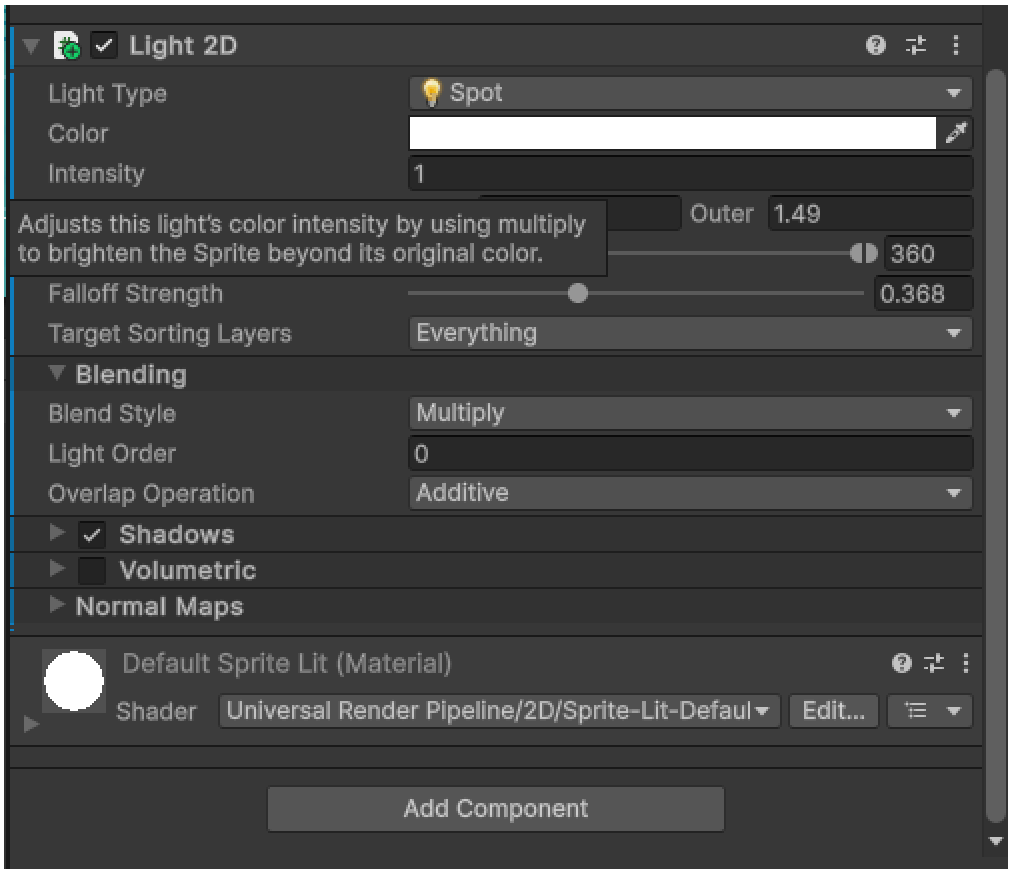
Blend Style	Multiply
Light Order	0
Overlap Operation	Additive

World Light (Script)

Script	WorldLight
World Light	GlobalLight (Light 2D)
Light Color Gradient	

Light Color Gradient – A gradient for smooth lighting transitions over time (0% = 00:00, 50% = 12:00, 100% = 24:00).

To set up light sources (lamps, etc.), add a `Light2D` component and configure its glow.



Light 2D

Light Type: Spot
Color
Intensity: 1
Outer: 1.49
Adjusts this light's color intensity by using multiply to brighten the Sprite beyond its original color.
360
Falloff Strength: 0.368
Target Sorting Layers: Everything

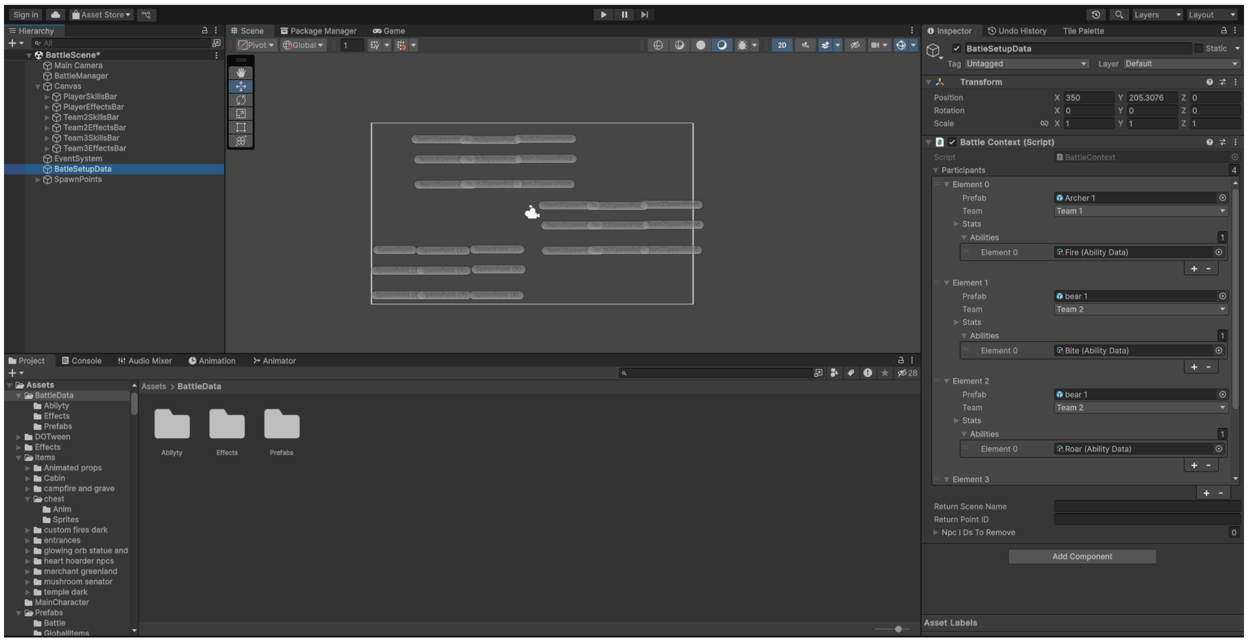
Blending
Blend Style: Multiply
Light Order: 0
Overlap Operation: Additive

✓ Shadows
Volumetric
Normal Maps

Default Sprite Lit (Material)
Shader: Universal Render Pipeline/2D/Sprite-Lit-Defaul

Battle System

To prepare the battlefield, it is enough to select the character that will be created at the beginning of the battle from the BattleData/Prefabs folder, select which team he will be on and add the abilities he will have

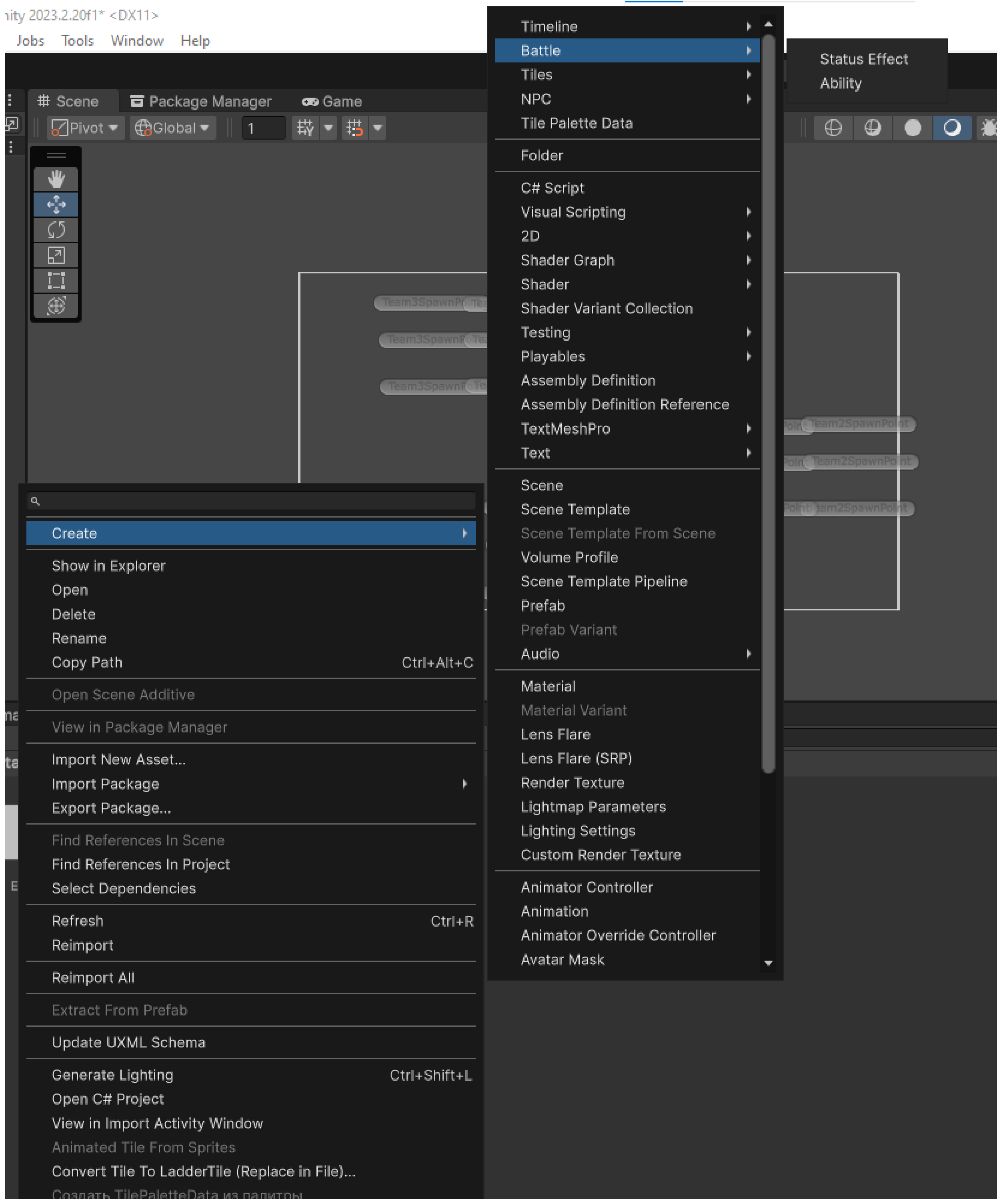


BatleSetupData
Tag Untagged Layer Default

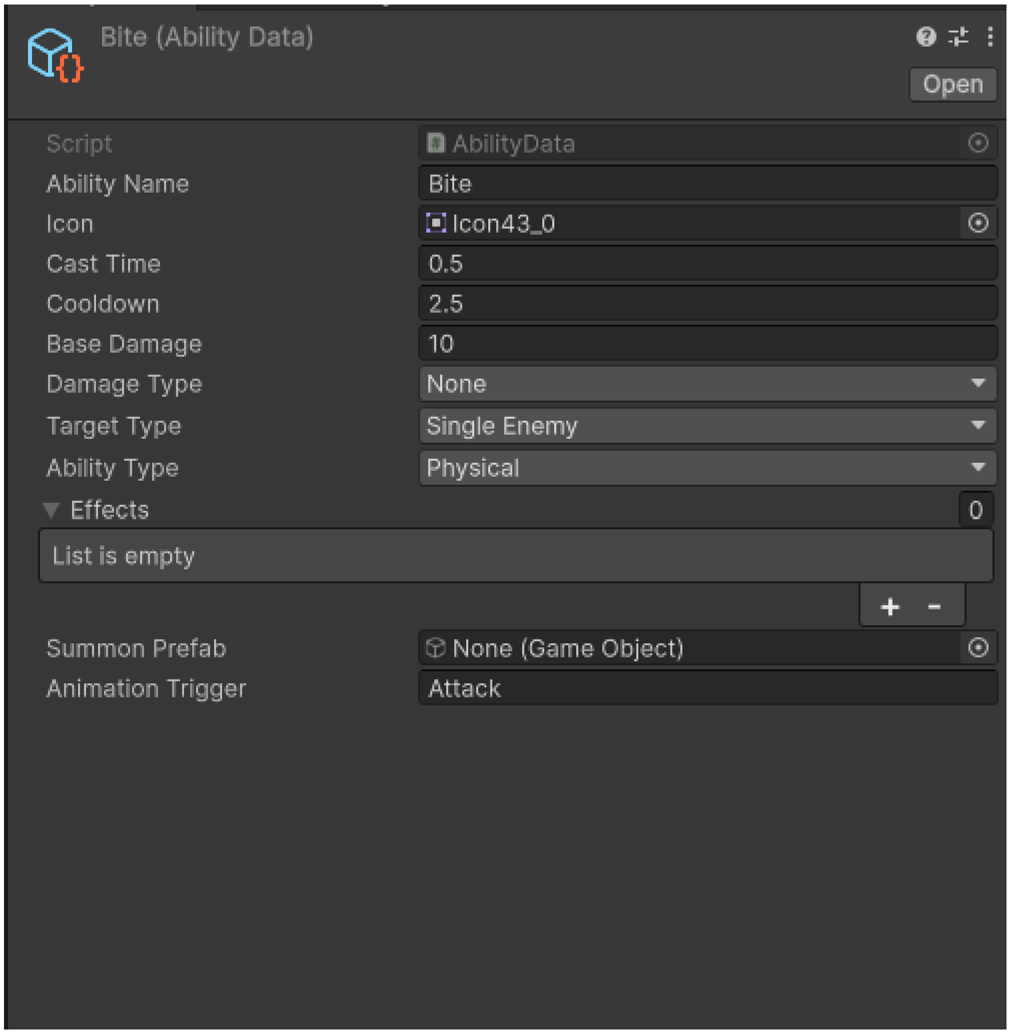
Transform
Position X 350 Y 205.3076 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Battle Context (Script)
Script BattleContext
Participants 4
Element 0
Prefab Archer 1
Team Team 1
Stats
Abilities 1
Element 0 Fire (Ability Data)
Element 1
Prefab bear 1
Team Team 2
Stats
Abilities 1
Element 0 Bite (Ability Data)
Element 2
Prefab bear 1
Team Team 2
Stats
Abilities 1
Element 0 Roar (Ability Data)
Element 3
Return Scene Name
Return Point ID
Npc I Ds To Remove 0

To create abilities or effects you need to select the appropriate menu:

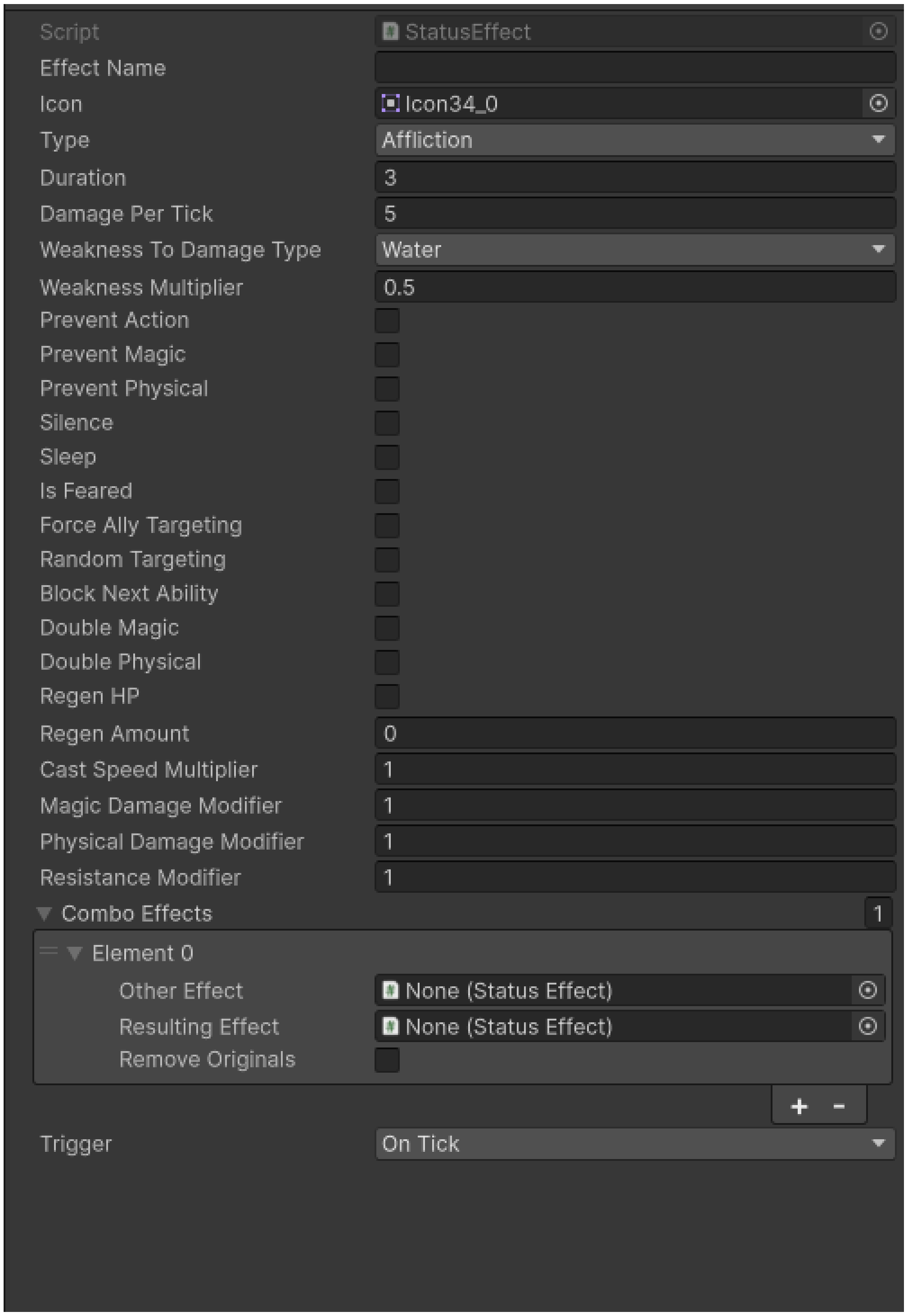


each ability will have its own parameters:


Bite (Ability Data)

Script	AbilityData
Ability Name	Bite
Icon	Icon43_0
Cast Time	0.5
Cooldown	2.5
Base Damage	10
Damage Type	None
Target Type	Single Enemy
Ability Type	Physical
Effects	0
List is empty	
Summon Prefab	None (Game Object)
Animation Trigger	Attack

icon - the ability icon that will be displayed on the UI panel
In the Effects tab, you can add effects that will be superimposed by this ability.
If the ability in the Summon Prefab field is specified from the character's BattleData/Prefabs folder, then when using the ability, that character will be summoned (if there is free space for it)
Animation Triger If the ability for characters has a special animation, then the animation name must be specified differently (see below for how to set up animations correctly)

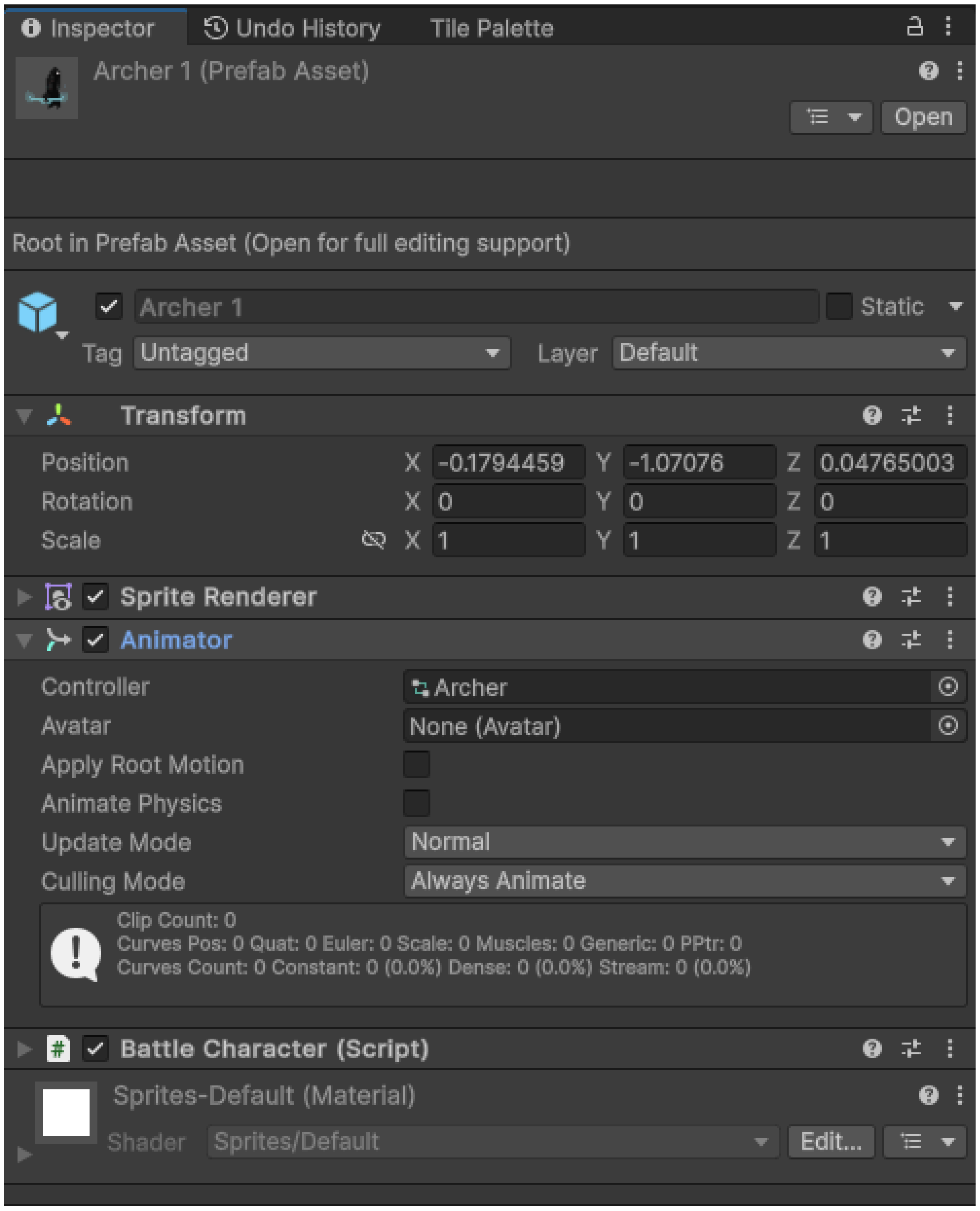


Script	StatusEffect
Effect Name	
Icon	Icon34_0
Type	Affliction
Duration	3
Damage Per Tick	5
Weakness To Damage Type	Water
Weakness Multiplier	0.5
Prevent Action	☐
Prevent Magic	☐
Prevent Physical	☐
Silence	☐
Sleep	☐
Is Feared	☐
Force Ally Targeting	☐
Random Targeting	☐
Block Next Ability	☐
Double Magic	☐
Double Physical	☐
Regen HP	☐
Regen Amount	0
Cast Speed Multiplier	1
Magic Damage Modifier	1
Physical Damage Modifier	1
Resistance Modifier	1
Combo Effects	1
Element 0	
Other Effect	None (Status Effect)
Resulting Effect	None (Status Effect)
Remove Originals	☐
Trigger	On Tick

combo effects is used for combinations of effects, in the Other field it is indicated which effect is combined with and in the Resulting field the effect that we get at the output is indicated
Trigger - remote field

Animations

each prefab has an Animator



Archer 1 (Prefab Asset)

Root in Prefab Asset (Open for full editing support)

Archer 1 ✓ Static

Tag Untagged Layer Default

Transform

Position X -0.1794459 Y -1.07076 Z 0.04765003

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

✓ Sprite Renderer

✓ Animator

Controller: Archer

Avatar: None (Avatar)

Apply Root Motion: ☐

Animate Physics: ☐

Update Mode: Normal

Culling Mode: Always Animate

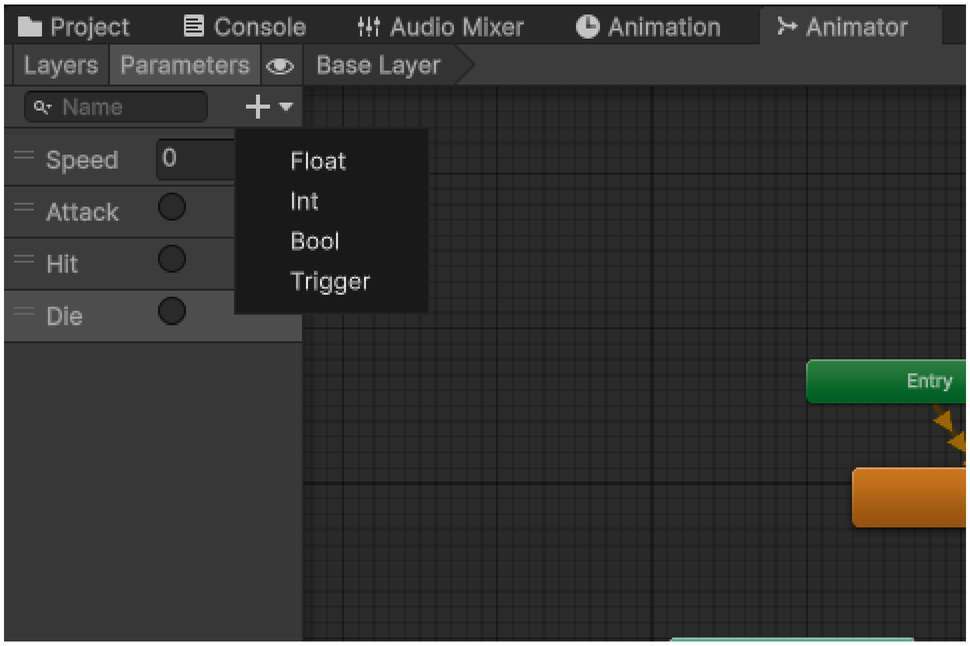
```
Clip Count: 0
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 0 (0.0%)
```

✓ Battle Character (Script)

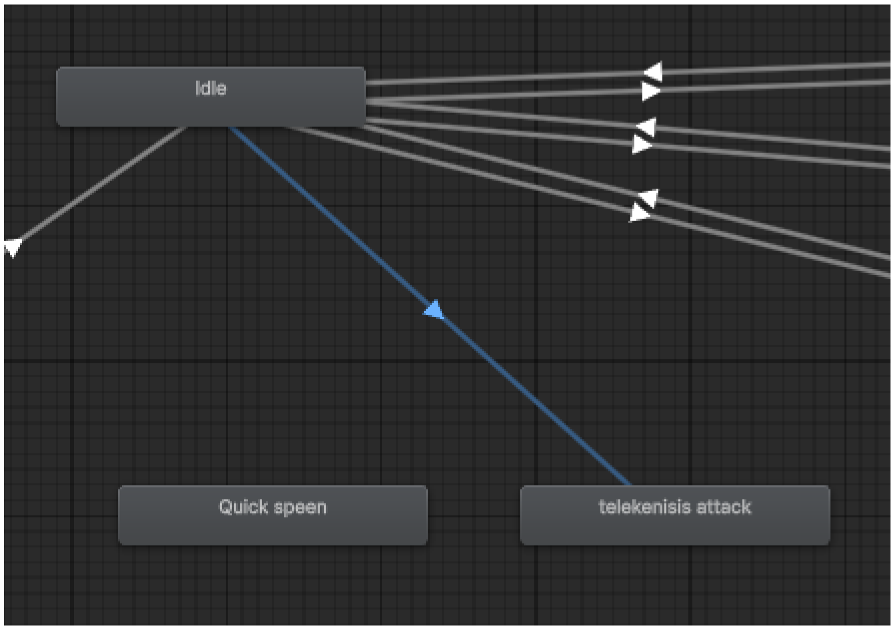
Sprites-Default (Material)

Shader Sprites/Default

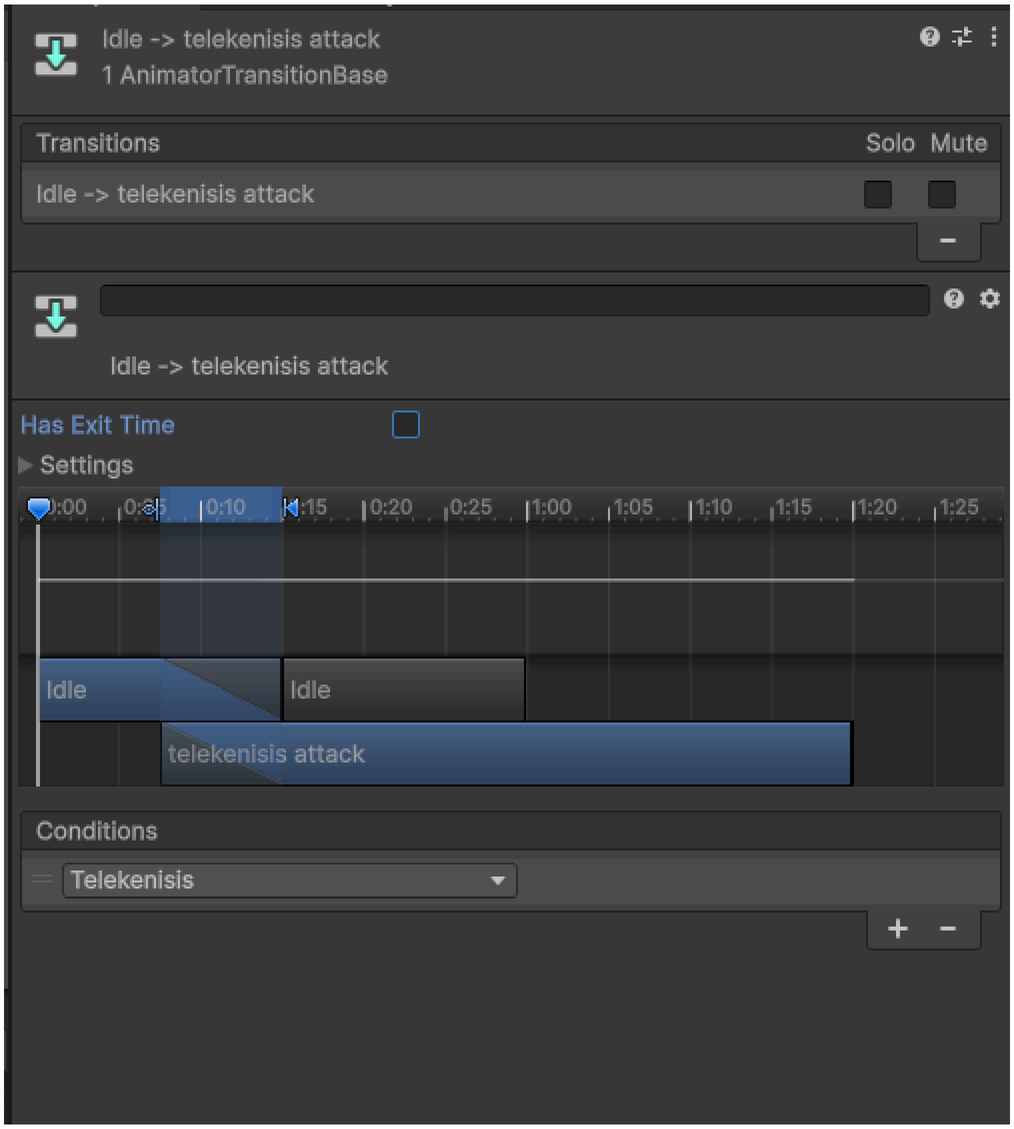
Simply go to the Animatior window or double-click in the Controller field.



create a new trigger and name it whatever you want the transition to animation to be called, for example “Telekenisis”



create a transition from the idle animation to the desired animation,
click on the arrow, remove Has Exit Time
and in Conditions specify the name of the newly created trigger



Idle -> telekenisis attack
1 AnimatorTransitionBase

Transitions | Solo | Mute
Idle -> telekenisis attack

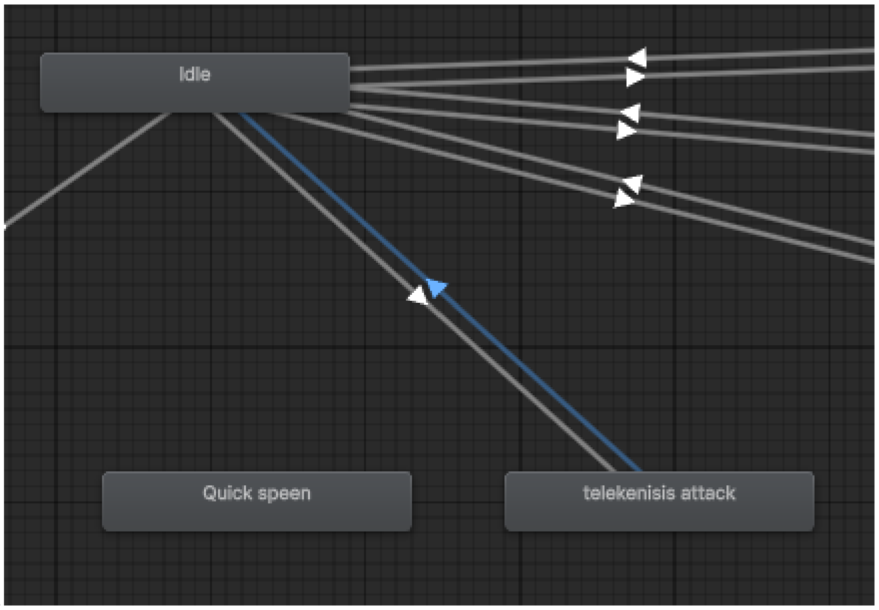
Idle -> telekenisis attack

Has Exit Time ☐

Settings

Conditions
Telekenisis

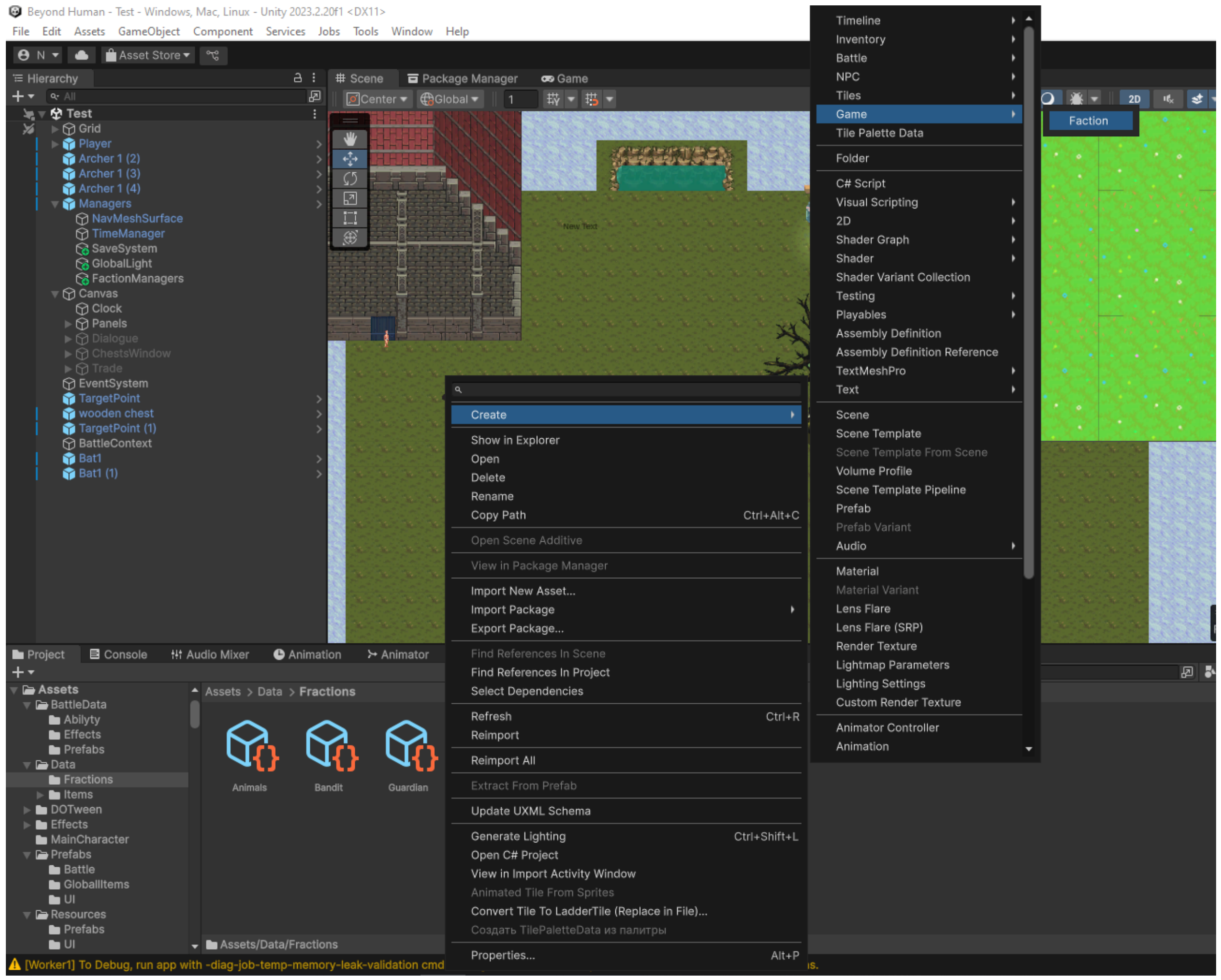
and you definitely need to remember to make a return arrow, but you don't need to do anything



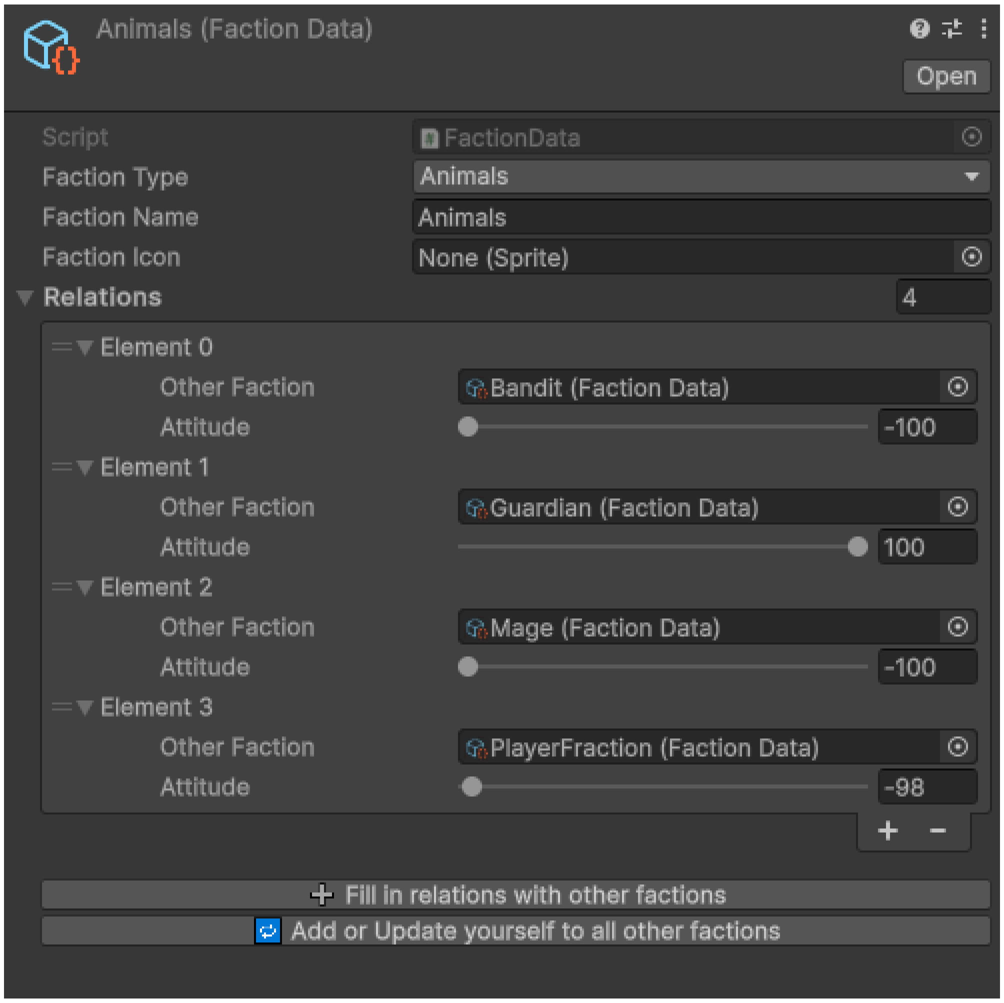
with it

Factions and battle start

Create Factions



To create a new faction, I recommend doing it inside the folder:
Assets/Data/Factions
Choose: **Create** → **Game** → **Faction**



Animals (Faction Data)

Script	FactionData
Faction Type	Animals
Faction Name	Animals
Faction Icon	None (Sprite)
Relations	4
Element 0	
Other Faction	Bandit (Faction Data)
Attitude	-100
Element 1	
Other Faction	Guardian (Faction Data)
Attitude	100
Element 2	
Other Faction	Mage (Faction Data)
Attitude	-100
Element 3	
Other Faction	PlayerFraction (Faction Data)
Attitude	-98

Open
Fill in relations with other factions
Add or Update yourself to all other factions

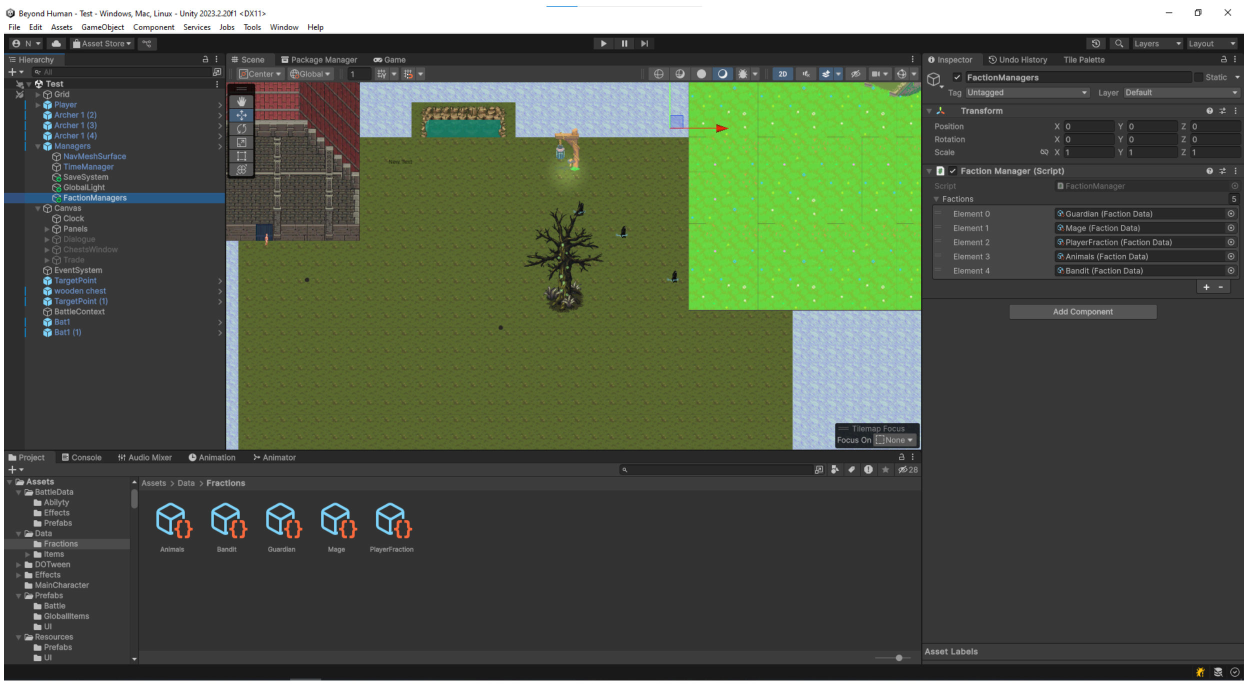
This is what the faction creation window looks like — here you define the type, name, icon, and relationships with other factions.
 Each faction must be manually added and assigned a relationship (e.g., Friendly, Neutral, Hostile) with the others.

IMPORTANT: The relationship values must be mirrored — if Faction A is Friendly to Faction B, then Faction B must also be Friendly to Faction A.

To make this easier, I've added two buttons:

- The first button goes through all factions and sets the current faction's relationships based on how others relate to it.

- The second button adds this faction to all others (if it's not already listed), letting you quickly populate missing entries.



Assets > Data > Fractions

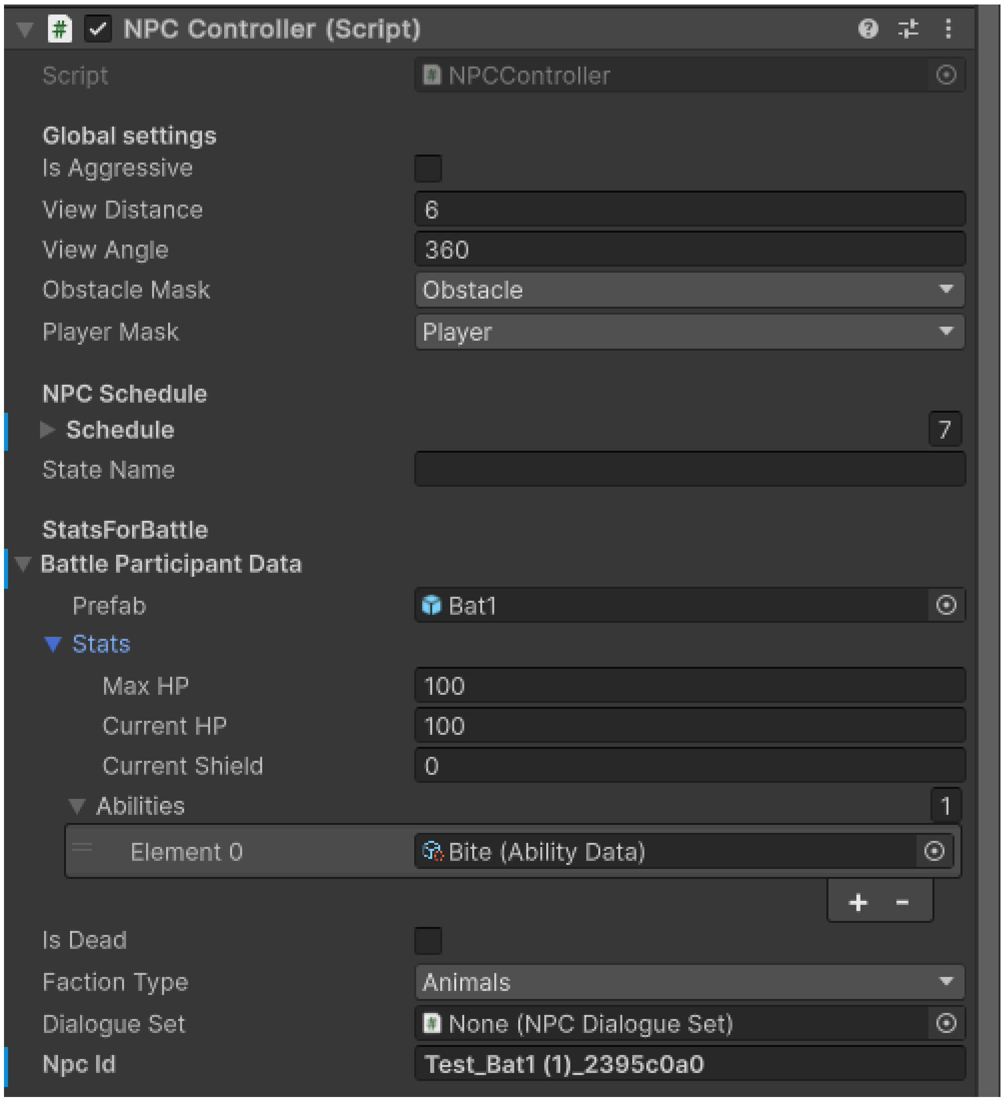
FactionManagers
Tag Untagged Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Faction Manager (Script)
Script FactionManager
Factions 5
Element 0 Guardian (Faction Data)
Element 1 Mage (Faction Data)
Element 2 PlayerFraction (Faction Data)
Element 3 Animals (Faction Data)
Element 4 Bandit (Faction Data)

To ensure factions work properly in the game, make sure all created factions are registered in the **FactionManager**.
This is required for proper battle preparation and faction logic.

settings to prepare for battle

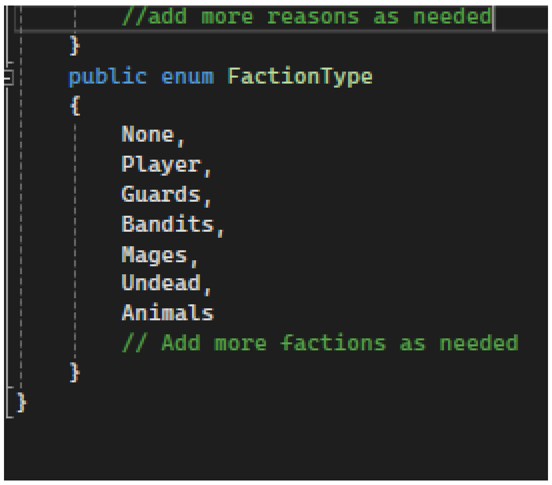


```
NPC Controller (Script)
Script                    NPCController
Global settings
Is Aggressive             [ ]
View Distance             6
View Angle                360
Obstacle Mask             Obstacle
Player Mask               Player
NPC Schedule
▶ Schedule                                7
State Name
StatsForBattle
▼ Battle Participant Data
  Prefab                  Bat1
  ▼ Stats
    Max HP                100
    Current HP            100
    Current Shield        0
  ▼ Abilities                             1
      Element 0           Bite (Ability Data)
                                        + -
Is Dead                   [ ]
Faction Type              Animals
Dialogue Set              None (NPC Dialogue Set)
Npc Id                    Test_Bat1 (1)_2395c0a0
```

Here's how the updated **NPCController** looks now:

- **Prefab** – a reference to the prefab used during battle scenes
- **Stats** – starting stats such as HP at the beginning of the battle
- **Abilities** – the list of skills/abilities this NPC will use during combat
- **IsDead (checkbox)** – indicates whether the NPC is dead; if true, the NPC will not appear in the scene on the next load (not recommended to modify manually)
- **Faction Type** – assigns the NPC to a specific faction

- **NPC ID** – a unique identifier automatically assigned when creating an NPC; used for saving/loading data



```
        //add more reasons as needed
    }
    public enum FactionType
    {
        None,
        Player,
        Guards,
        Bandits,
        Mages,
        Undead,
        Animals
        // Add more factions as needed
    }
}
```

o add or modify the list of factions, open the **NPCEnums** script located at:
Assets/Scripts/Abstracts/NPCEnums.cs
Find the `FactionType` enum and add the new faction names to that list.