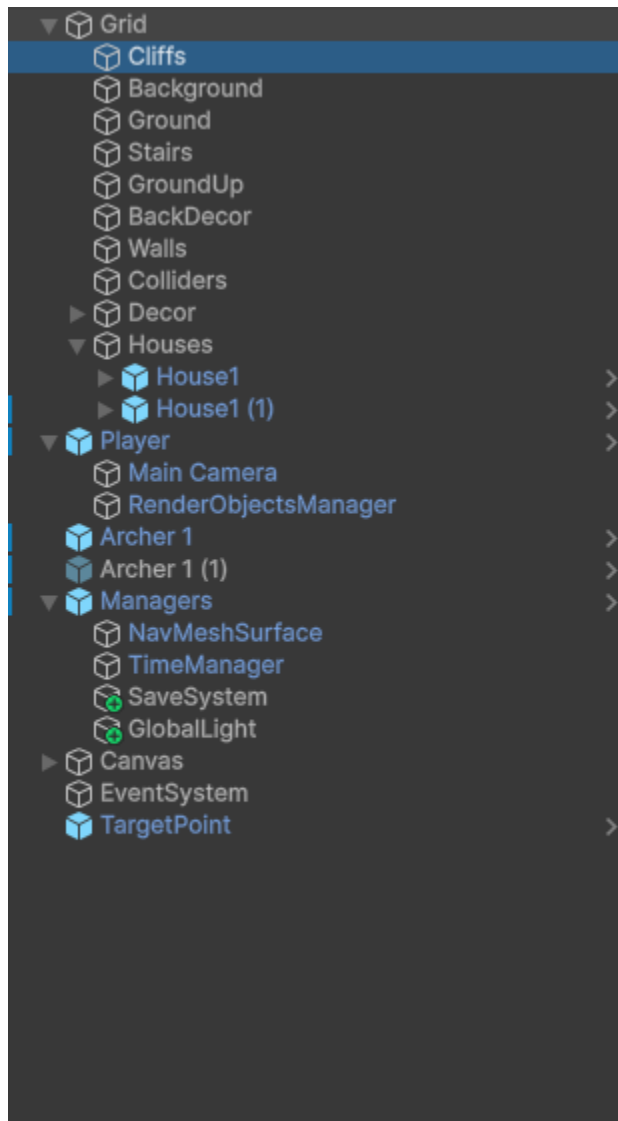
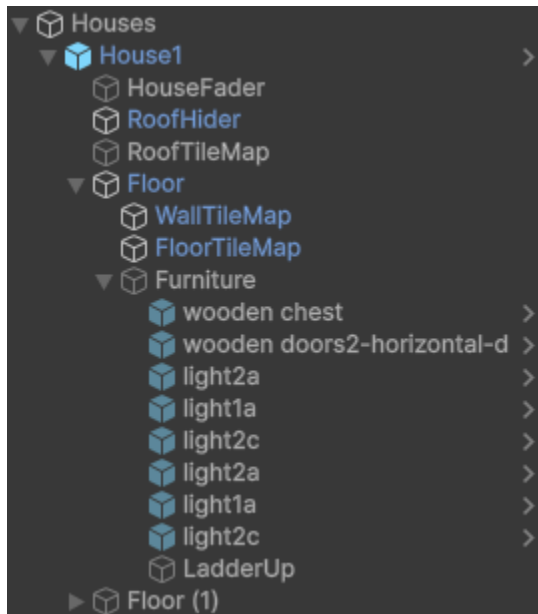


Map and Houses

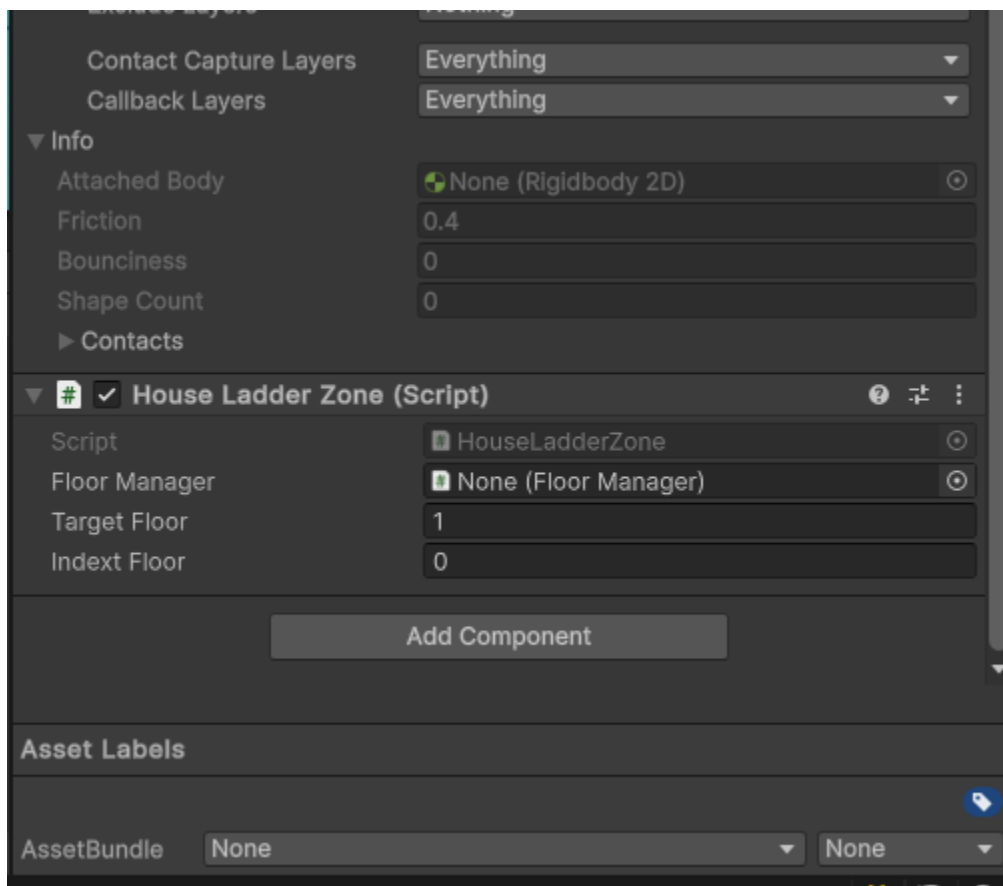
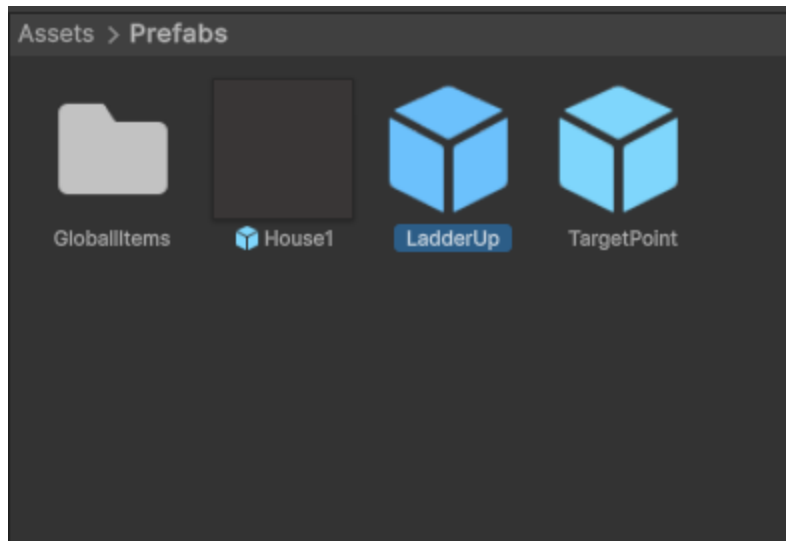


- 1) **Cliffs** – Used to draw cliffs that act as the boundary of the playable area (i.e., the edge of the world).
- 2) **Background** – Used for drawing water or other tiles that appear *below* the main ground level, creating a depth effect.
- 3) **Ground** – Used for regular ground tiles.
- 4) **Stairs** – 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.



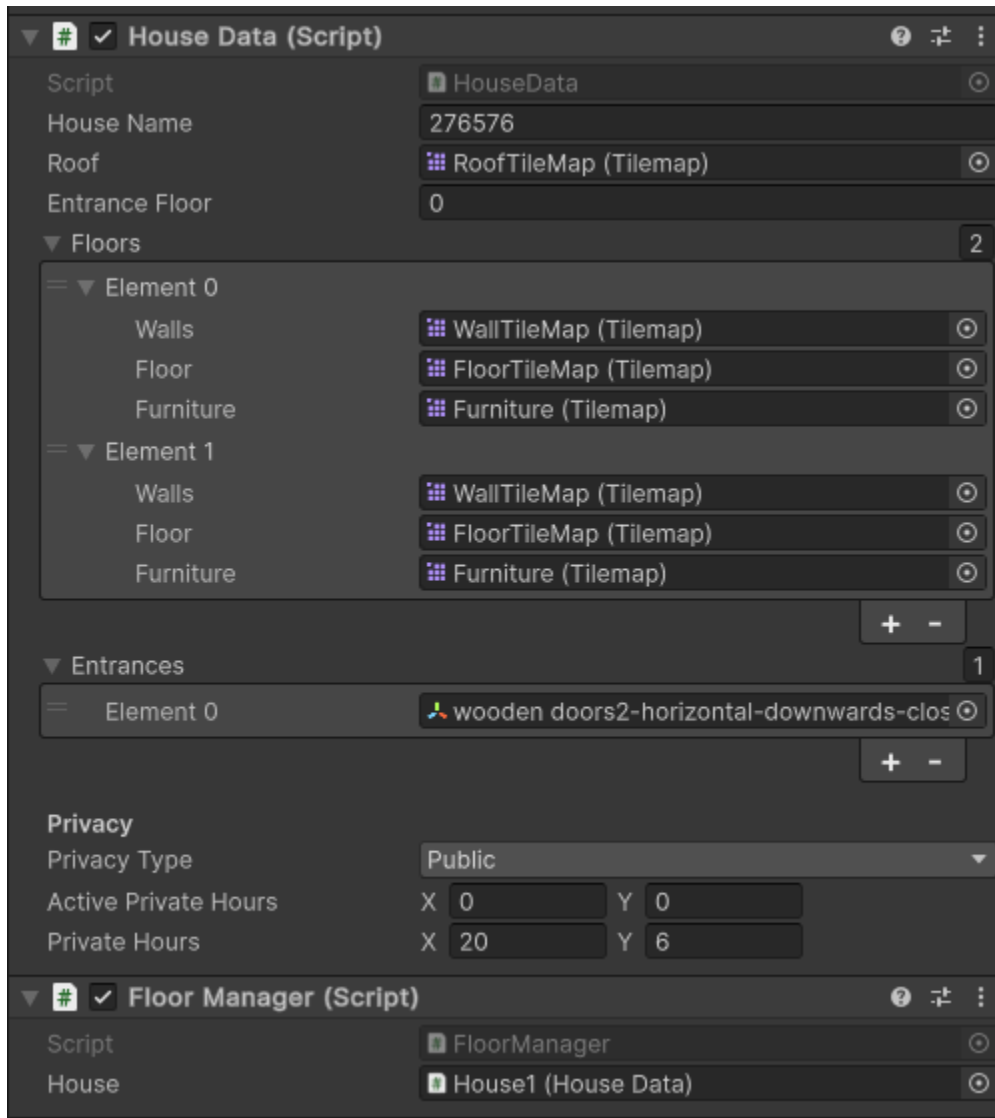
- a) **RoofTileMap** – Represents the house roof, which hides automatically when the player enters the house.
- b) **Floor** – A container for each floor of the house, containing:
- i) **WallTileMap** – House walls.
 - ii) **FloorTileMap** – House floor.
 - iii) **Furniture** – For decorations (you can also add animated prefab children).
 - iv) For multi-story buildings, each floor must include a **Ladder** to allow the player to move between floors.



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).



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

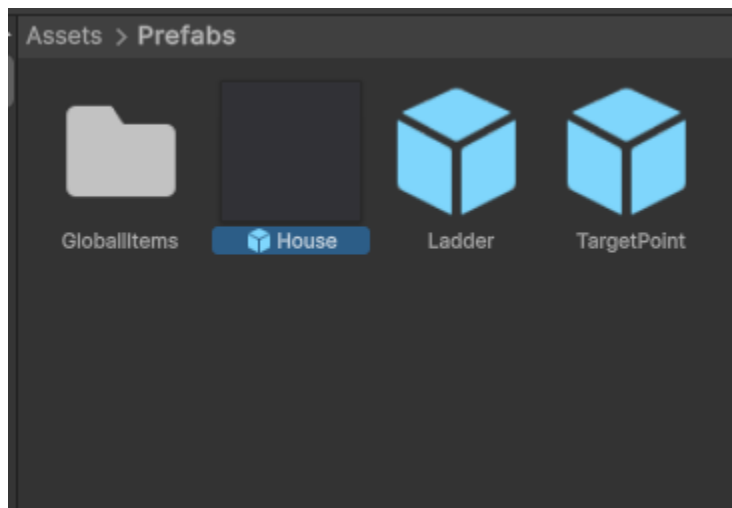
- **Roof** – The roof object.
- **EntranceFloor** – The index of the main entrance floor (e.g., set to 1 if the ground floor is considered a basement).

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

Privacy Settings

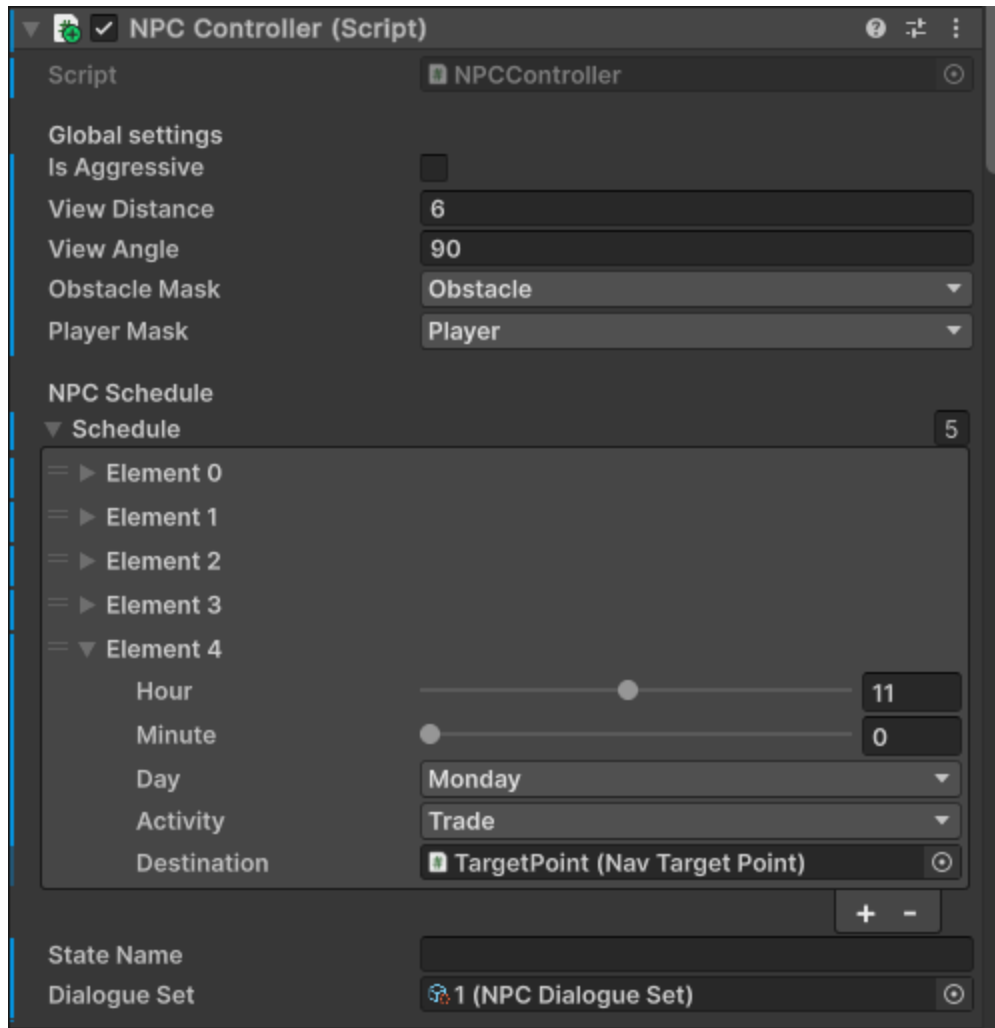
- Used to configure whether a house is public or private.
- **PrivacyType**: **Public**, **Private**, or **PrivateScheduled**.
- **PrivateHours**: The time range during which the house is considered private (used only with **PrivateScheduled**).

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.

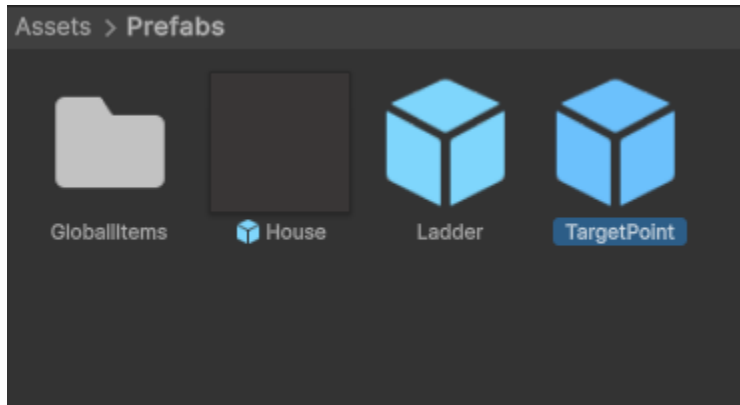


- a)
- b) **isAggressive** – If enabled, the NPC will chase the player as soon as they are spotted. (This will likely be reworked in the “Factions & Reputation” phase.)
- c) **View Distance** – NPC vision radius.
- d) **View Angle** – NPC field of view angle.
- e) **Obstacle & Player Mask** – Generally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

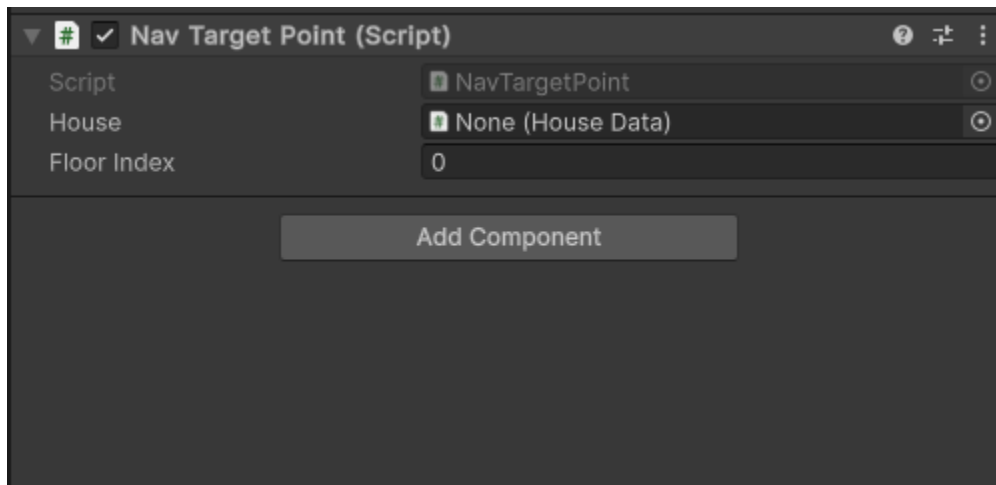
Schedule – A list of the NPC’s daily activities:

- **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:



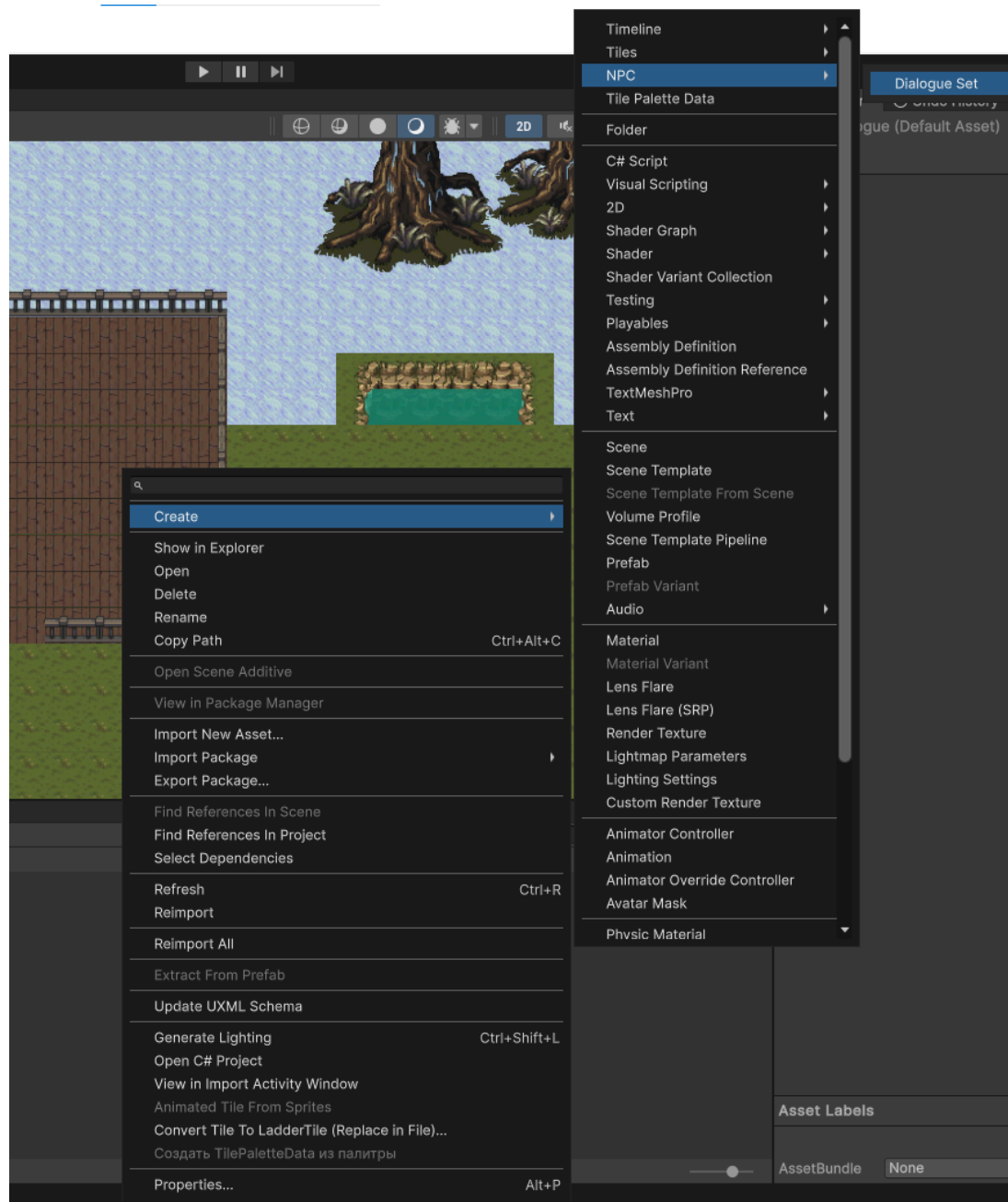
House – The house it belongs to.

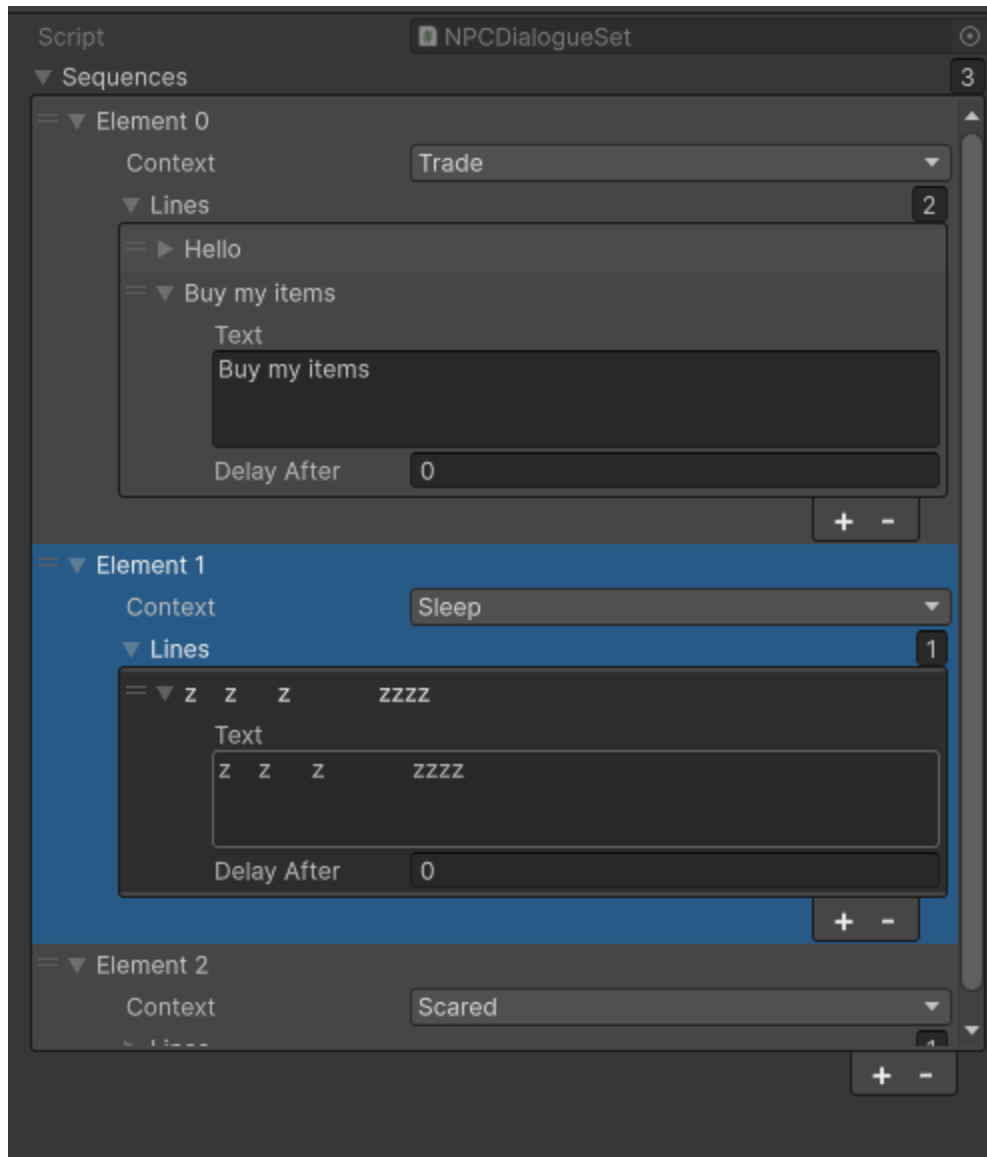
FloorIndex – Which floor the point is on.
(If the point is outside, these fields can be ignored.)

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:

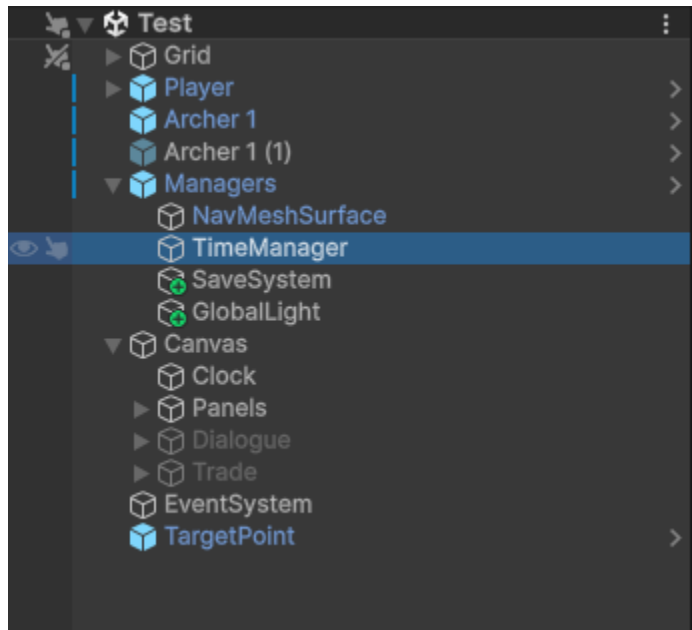




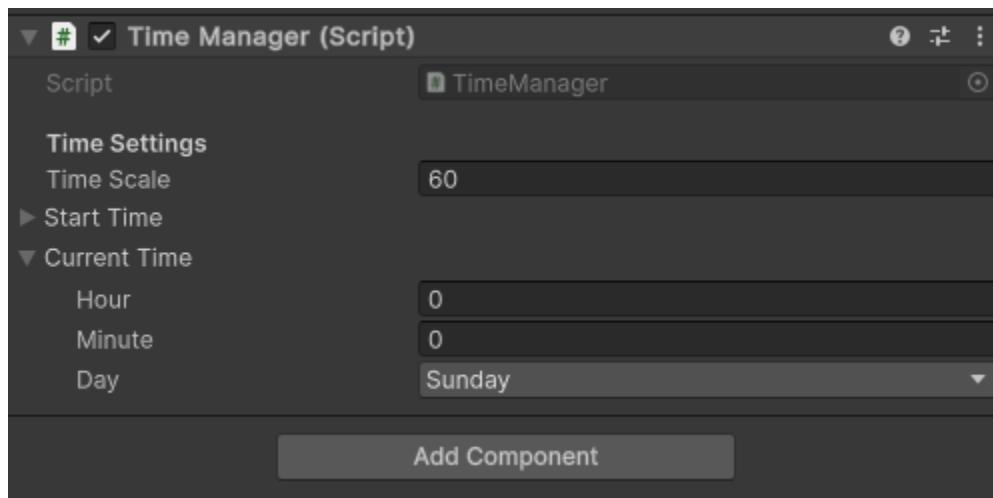
- **Context** – The state in which the lines are used. Matches the NPC activity types:
 - Idle – Also used for Wander and Chill.
 - Work
 - Trade
 - Guard – Also used for Patrol.
 - Sleep
 - Hunt
 - Scared
- **Lines** – The lines of dialogue spoken in that state.

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

Global settings

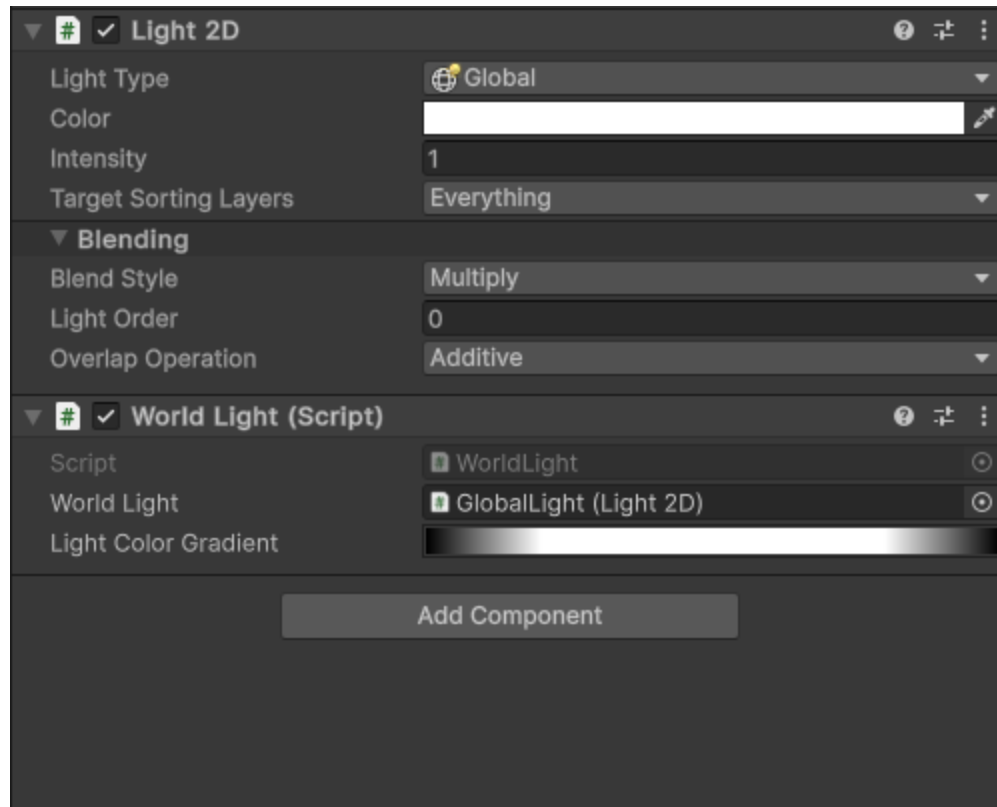


Under **Managers**, the **TimeManager** controls in-game time.



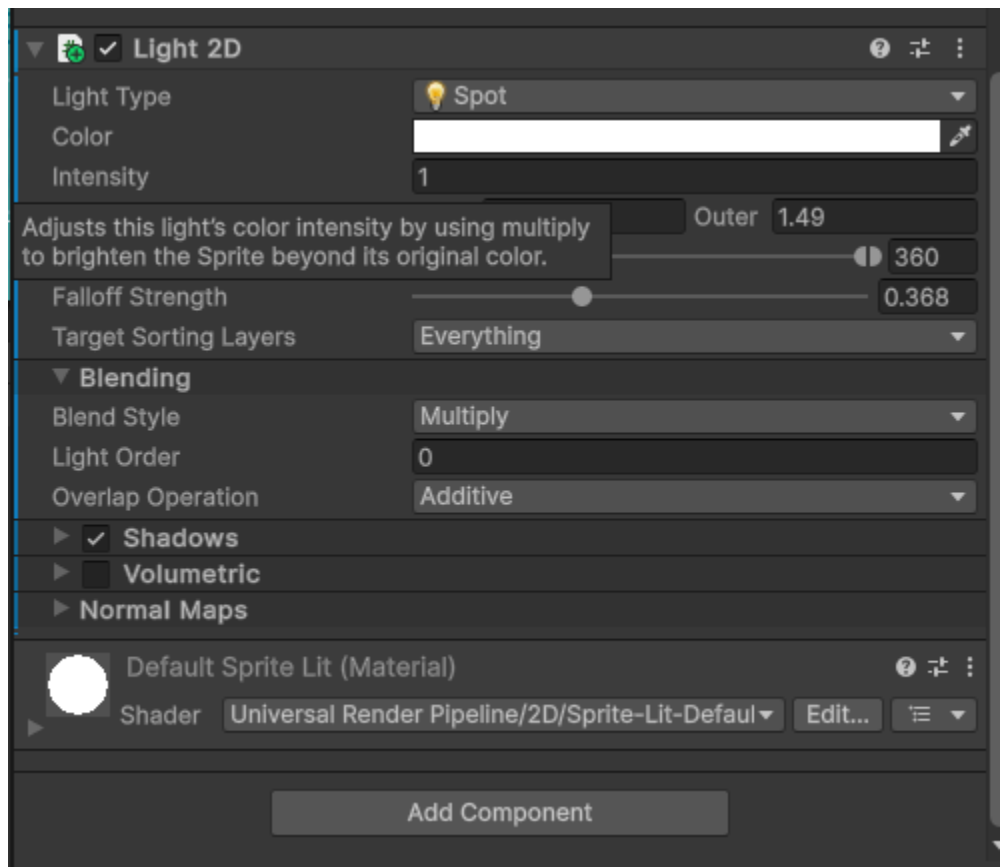
Time Scale – Determines how fast in-game time passes compared to real time (e.g., how many real seconds equal one in-game second).

The **GlobalLight** object controls lighting.



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

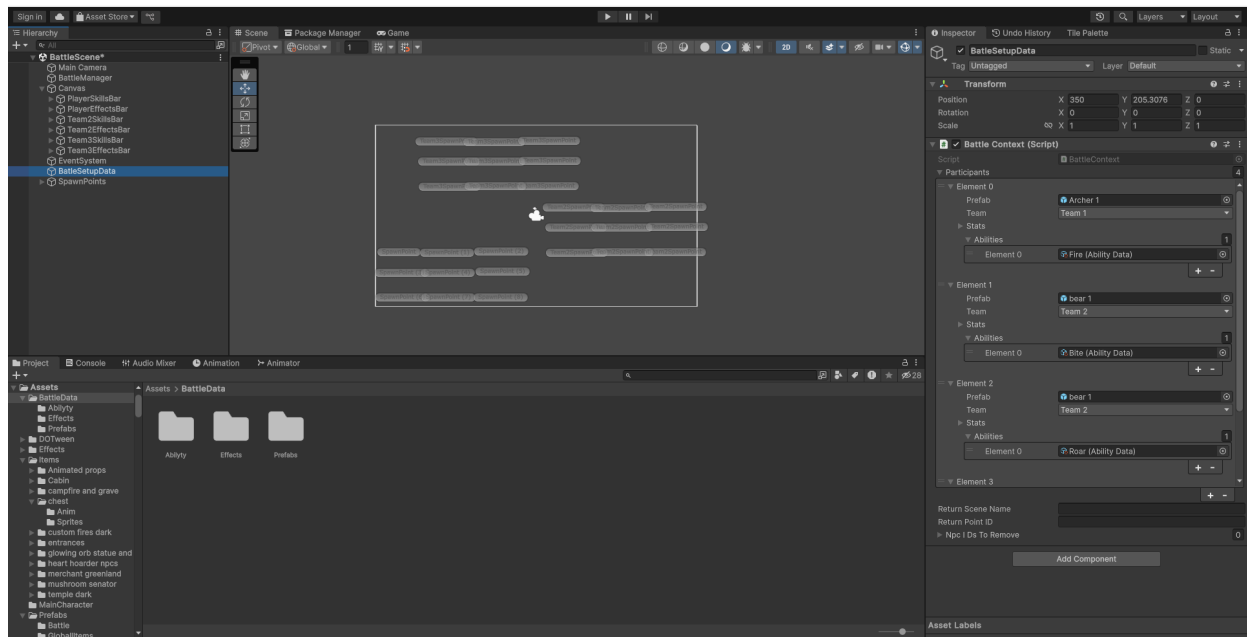
To set up light sources (lamps, etc.), add a **Light2D** component and configure its glow.



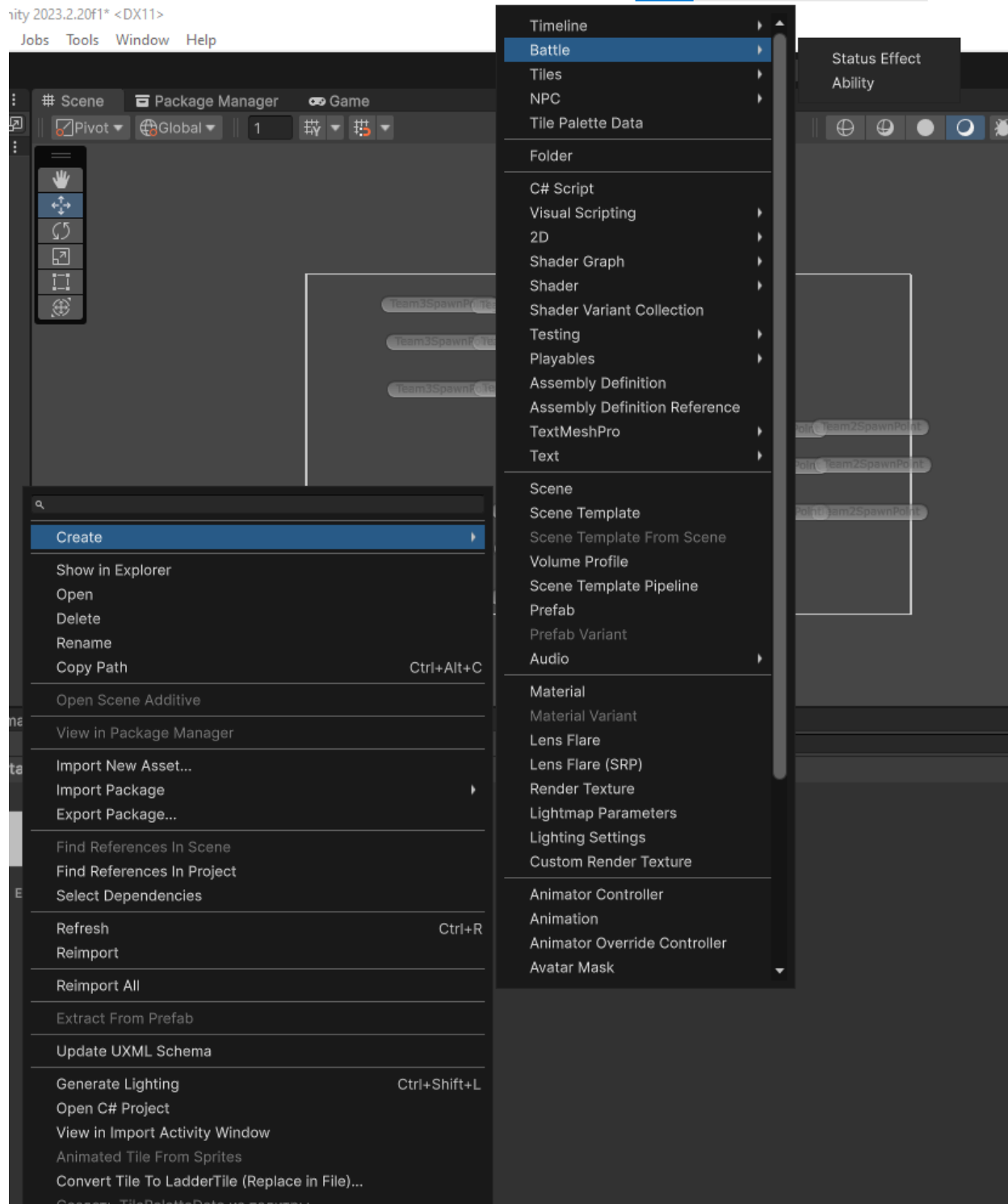
Battle System

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



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



each ability will have its own parameters:

Parameter	Value
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
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)

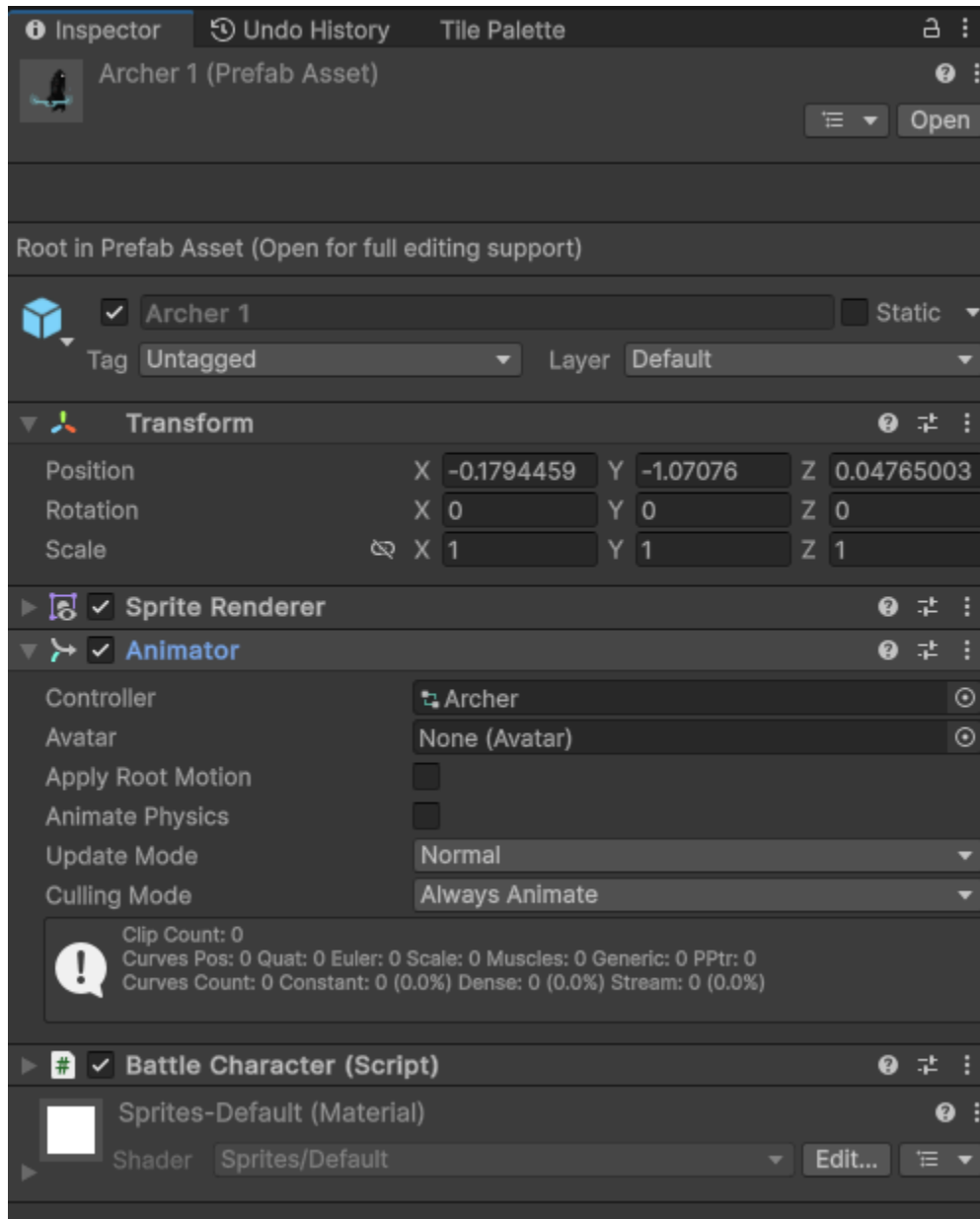
Animation Trigger If the ability for characters has a special animation, then the animation name must be specified differently (see below for how to set up animations correctly)

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	<input type="checkbox"/>
Prevent Magic	<input type="checkbox"/>
Prevent Physical	<input type="checkbox"/>
Silence	<input type="checkbox"/>
Sleep	<input type="checkbox"/>
Is Feared	<input type="checkbox"/>
Force Ally Targeting	<input type="checkbox"/>
Random Targeting	<input type="checkbox"/>
Block Next Ability	<input type="checkbox"/>
Double Magic	<input type="checkbox"/>
Double Physical	<input type="checkbox"/>
Regen HP	<input type="checkbox"/>
Regen Amount	0
Cast Speed Multiplier	1
Magic Damage Modifier	1
Physical Damage Modifier	1
Resistance Modifier	1
▼ Combo Effects	1
<div> <div>▼ Element 0</div> <div> <div>Other Effect</div> <div>None (Status Effect)</div> </div> <div> <div>Resulting Effect</div> <div>None (Status Effect)</div> </div> <div> <div>Remove Originals</div> <div><input type="checkbox"/></div> </div> </div>	
<div> <div>+</div> <div>-</div> </div>	
Trigger	On Tick

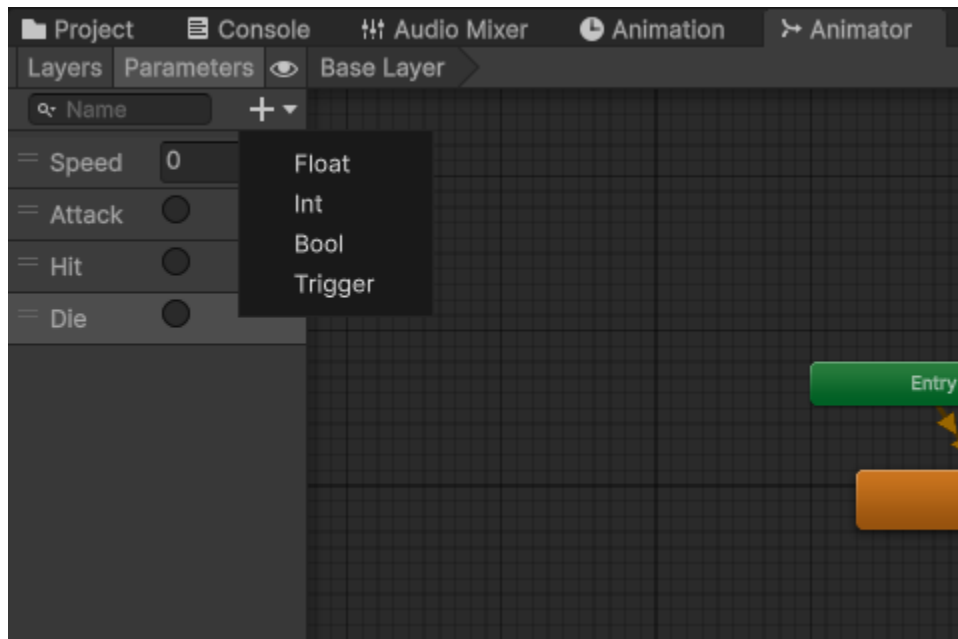
combo effects is used for combinations of effects, in the Other field it is indicated which effect is combined with and in the Resulting field the effect that we get at the output is indicated
Trigger - remote field

Animations

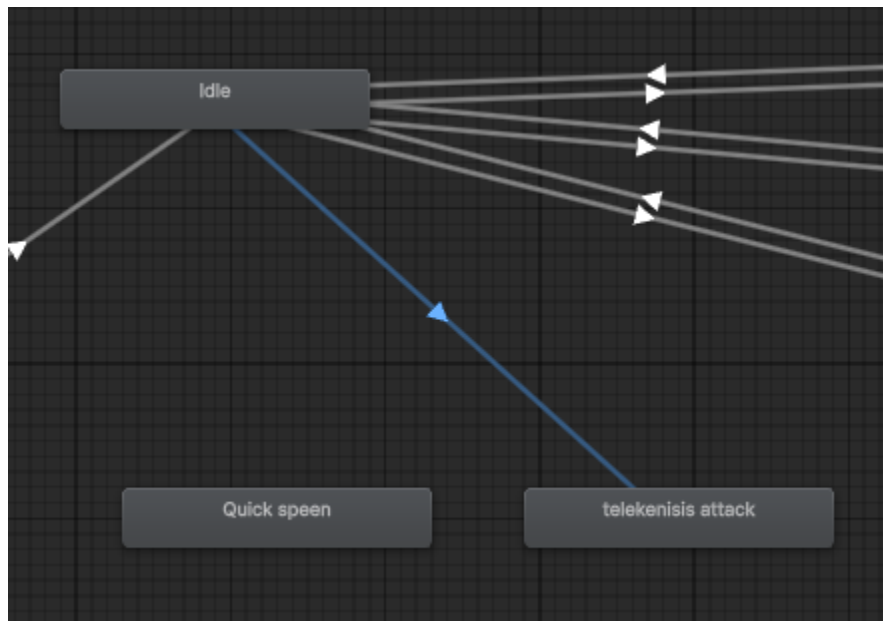
each prefab has an Animator



Simply go to the Animator 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

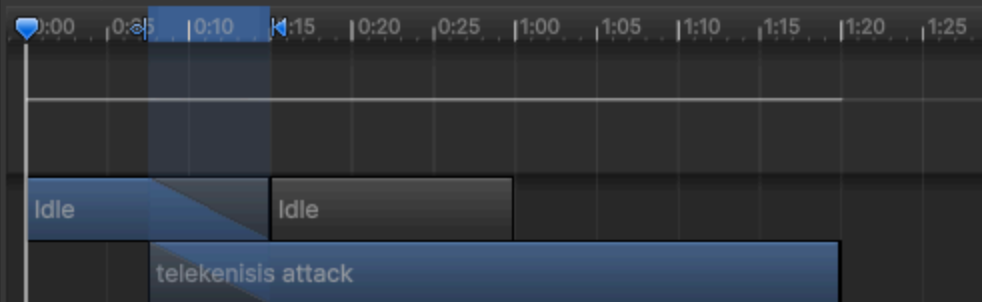


Idle -> telekenisis attack

Has Exit Time



Settings

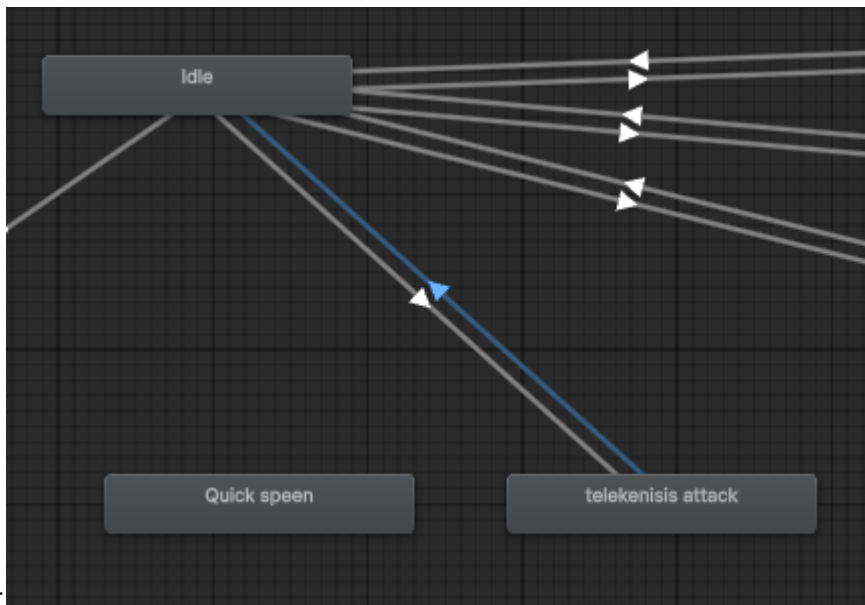


Conditions

Telekenisis



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



with it