

Skills for Game Creator Documentation



Support: support@mitchmr-studios.io

Table of Content

Introduction	4
Key Features	4
Online Documentation	4
Getting Started	5
Setup	5
Skills Overview	6
Skills Window	6
Preferences	6
Skills	7
General Settings:	8
Skill Settings:	8
Skill Events:	9
Types.....	10
Settings	11
User Interface	11
Behavior Configuration.....	11
Skills Component	12
Actions / Conditions / Triggers.....	13
Actions.....	13
Conditions	13
Triggers	14
Skill Tree	15
Automatic Skill Tree.....	17
Skill Tree Root.....	17
Skill Tree Content.....	18
Skill Element	19
Manual Skill Tree	21
Skill Tree Root.....	21
Skill Tree Content.....	21
Skill Element	22
Skill Bar	23
Skill Bar Skill Tree.....	23
Root	23
Skill Bar Skill Tree Element	24

Skill Bar	25
Root	25
Skill Bar Element.....	25
FAQ	27
Example Setup.....	30
Preferences Window.....	30
Scene	30

Introduction

Skills is a module for [Game Creator](#) and allows you to add skills to your game.

Key Features

- Create **skills** for gameobjects to unlock and execute
- Add **types** for automatically generating a **skill tree** (or make your own)
- Easily debug your **skills** directly on the gameobjects
- Define conditions before executing or unlocking a skill
- Add actions to every **stage of execution** of the skill (Activation, Casting, Execution, Finishing)
- Use a skill bar to quickly assign skills to buttons with visualization of the current state
- Let users have the possibility to **reset** the skills
- Tight integration with the stats module and the variable system for level and skill points

Online Documentation

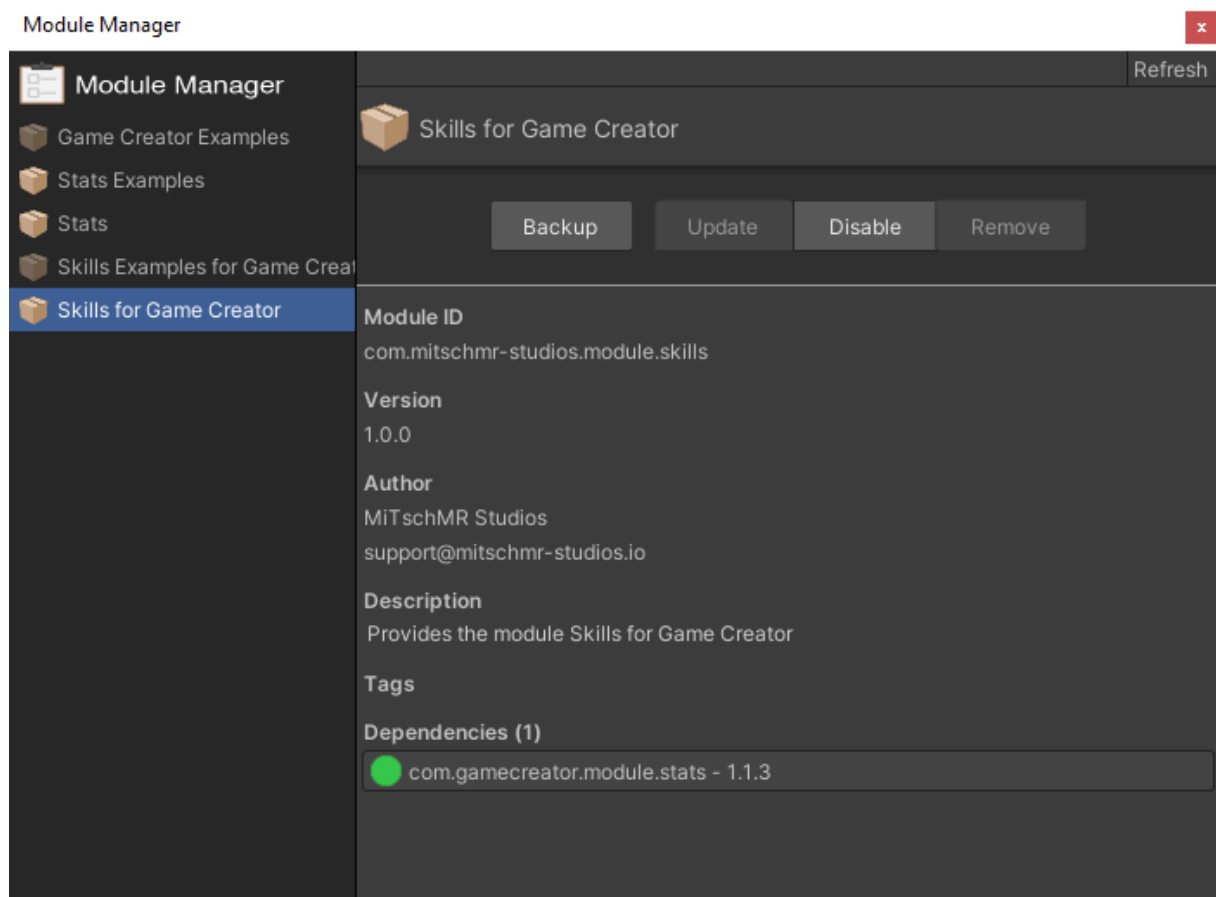
There is also an [online documentation](#) you can visit.

Getting Started

Setup

Download the package from the the **Unity Asset Store**. You need **Game Creator** and the **Stats** module installed first.

Then, bring up the *Modules Window* clicking on the Game Creator option in the toolbar. Click the Skills Module **Enable** button. It should then look like this after importing all of the content:



This asset requires **Game Creator** and the **Stats** module and won't work without it. Don't attempt to extract the package inside the Plugins/ folder as it will throw errors.

The demo scene of this asset requires the free mixamo assets [Magic Pack](#) and [Melee Axe Pack](#) and won't work properly without them.

Skills Overview

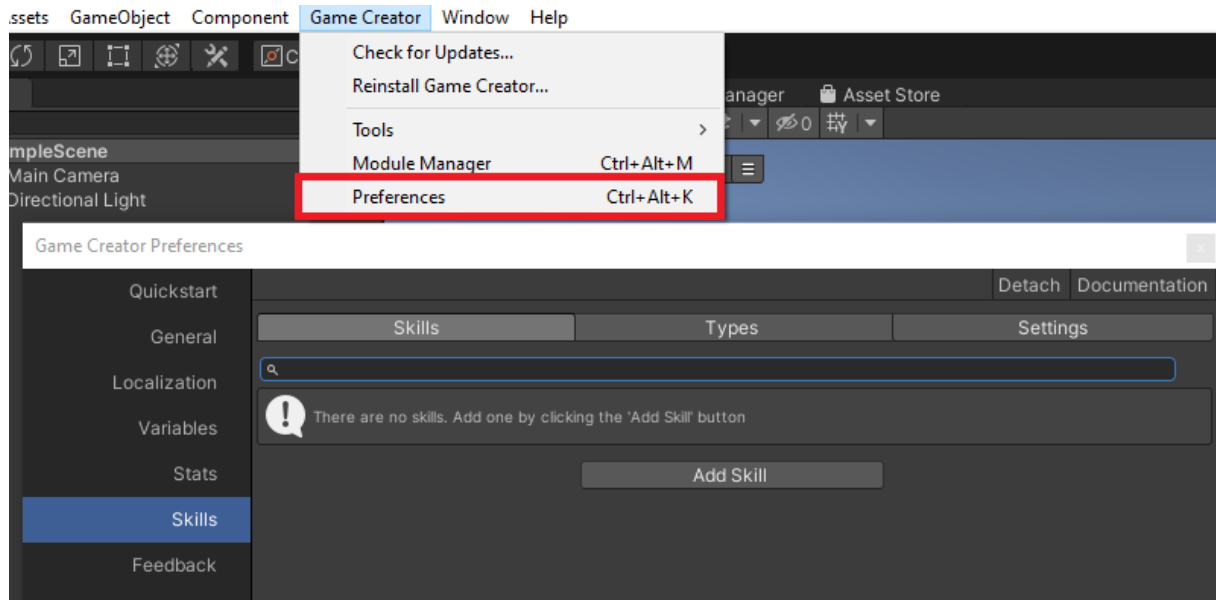
The skills module consists of four pieces, the **Skills Window**, the **Skills component**, the **Skill Tree** and the **Skill Bar**:

- **Skills tab in the Game Creator Preferences Window:** Manage all your skills, types and settings
- **Skills component on Game Objects:** This is where the skills data is stored and most logic is handled
- **Skill Tree:** Create a skill tree by choosing between two prefabs (automatic and manual)
- **Skill Bar:** Bring skills to life with the skill bar

Skills Window

Preferences

You must define skills before you can use them. In order to do this, open the Game Creator Preferences window and click on the skills tab:



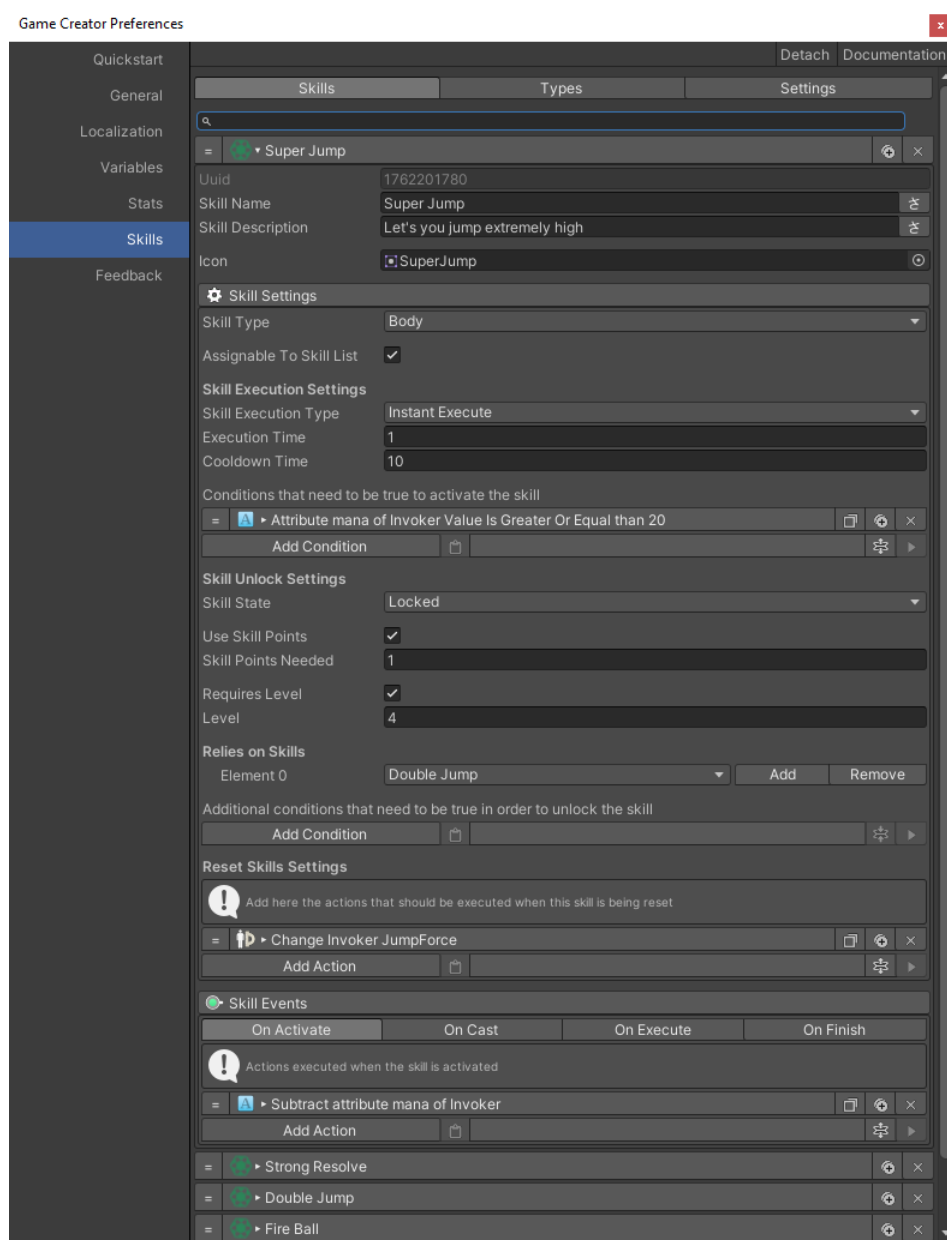
You can also open the skills window directly by pressing `CTRL + ALT + S` on Windows or `⌘ + ALT + S` on Mac.

Look below for more information on every tab in the window:

- [Skills](#): Here you can add new skills and define what they do and how they are connected between each of them
- [Types](#): Define the types of skill (i.e. Melee, Range, Warrior etc.)
- [Settings](#): Configure settings like what stat or global variable the level property should use

Skills

The skills tab allows you to add new skills and configure all settings about it:



As you can see, a skill has many settings.

General Settings:

- **Skill Name:** The name of the skill
- **Skill Description:** The description of the skill
- **Icon:** The icon of the image (used in the Skills component and skill tree)

Skill Settings:

- **Skill Type:** What type of skill is this one (you can only choose one type per skill). This type is also used to procedurally generate a skill tree
- **Assignable to Skill Bar:** Can you add the skill to a skill bar (like a hotbar with skills to execute)?
- **Skill Execution Type:** What execution type is the skill? There are three types of skill:
 - **Instant Execute** → Instantly executes a skill (skips the casting phase)
 - **Cast Execute** → This skill needs a cast time before it executes
 - **Permanent Execute** → Constantly executing this skill, like having an HP buff (only executes the execution phase, skips every other)
 - **Stored** → "Reloads" a skill after executing, can store up to a certain amount of skill executions. The "reload" will be reset when you execute the skill again. Only a full "reload" gives you an execution back

You can specify the time for the cast and execution phase before the next phase starts.

- **Cast time:** Time for casting a skill (i.e. a magic spell with a preparation time)
- **Execution time:** Time for a skill to execute
- **Cooldown time:** Time until a skill can be used again. Does not apply to Permanent Execute skills
- **Cooldown time between:** Time to wait for a stored skill to wait until it can be used again
- **Max amount:** Maximum amount of executions of a stored skill
- **Start amount:** The starting amount of executions of a stored skill
- **Conditions for activating a skill:** These conditions must be true to activate the skill
- **Skill State:** The state a skill has at the start of the game. Ideal for specifying beginner skills that are active from the start

- **Use Skill Points:** Use the skill points property in the settings to unlock this skill
- **Skill Points Needed:** The amount of skill points to use to unlock the skill
- **Requires Level:** Use the level property in the settings as a requirement to unlock the skill
- **Level:** The level the character needs to have to unlock the skill
- **Relies On Skills:** Rely on these skills (they need to be unlocked)
- **Additional conditions to unlock the skill:** Use these additional conditions to check if someone can unlock the skill
- **Reset Skills Settings:** These actions will get executed when the skill is being reset (usually when doing a reset of all skills)

Skill Events:

- **On Activate:** Execute these actions when the skill gets activated
- **On Cast:** Execute these actions when the skill is in the casting phase
- **On Execute:** Execute these actions when executing the skill
- **On Finish:** Execute these actions when the skill reaches the finishing phase

Types

In this list you can define the type of skills you want to have in your game. A skill can only be assigned to one type. Based on this type and the "Relies On Skills" property the skills are grouped together in the skill tree.

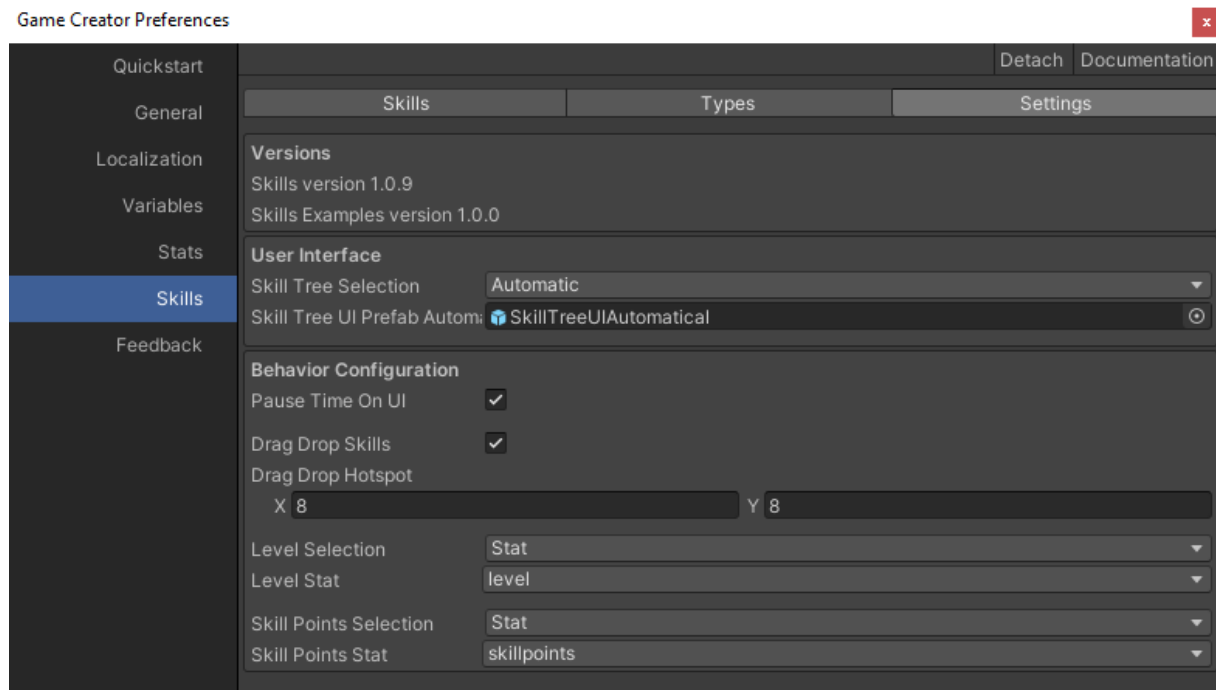
Game Creator Preferences ×

		Detach	Documentation
General	Skills	Types	Settings
Quickstart	ID: 1	General	
Localization	ID: 2	Warrior	
Variables	ID: 3	Magic	
Dialogue	ID: 4	Range	
Inventory	ID: 5		
Quests	ID: 6		
	ID: 7		
	ID: 8		

You are limited to 32 types of skills.

Settings

On this page you define the settings for the module:



User Interface

- **Skill Tree Selection:** Choose between Automatical and Manual skill tree
- **Skill Tree UI Prefab:** If none is set it takes the default one for the type selected

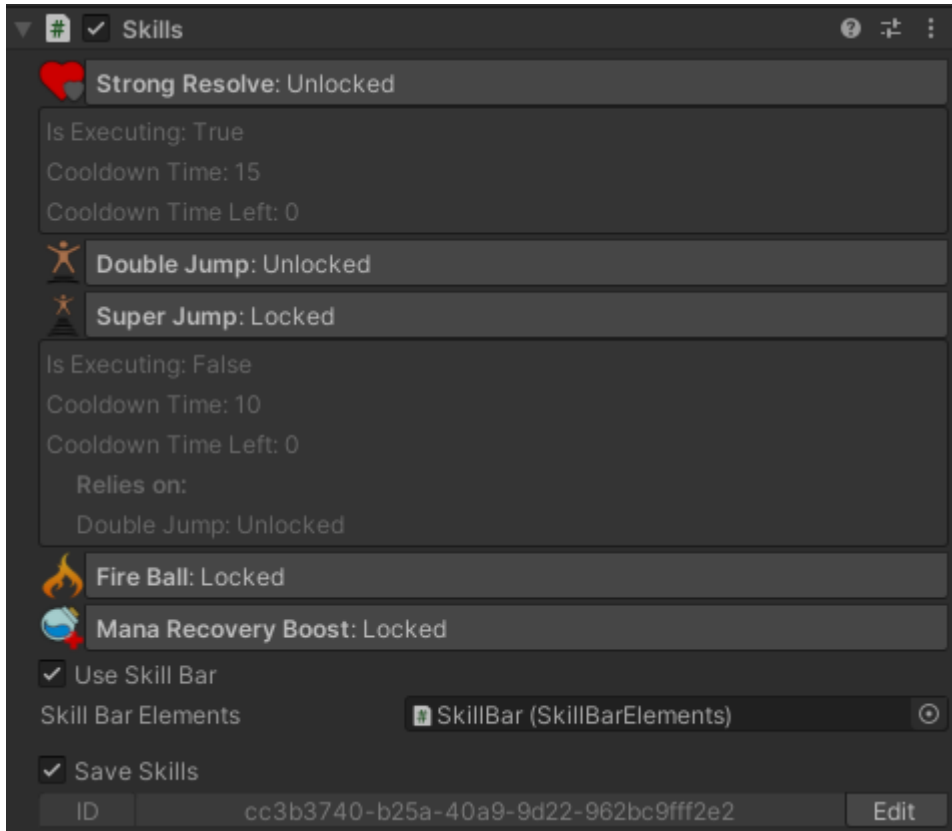
Behavior Configuration

- **Pause Time On UI:** Should the time be paused when you open the skill tree
- **Drag Drop Skills:** Enable the drag and drop skills feature for assigning skills to the skill bar
- **Drag Drop Hotspot:** How many pixels is the difference from the center of the image to the cursor
- **Level Selection:** Choose between a stat and a global number variable for the level
- **Level Variable:** Select a global number variable for the level
- **Level Stat:** Select a stat for the level
- **Skill Points Selection:** Choose between a stat and a global number variable for the skill points
- **Skill Points Variable:** Select a global number variable for the skill points

- **Skill Points Stat:** Select a stat for the skill points

Skills Component

The skills component stores the data for each skill and handles operations like executing a skill or unlocking one:



It has a UI which only works in runtime and allows you to see the current state of every skill (unlocked or not), if it is executing or not, the cooldown time left and finally on which skills it relies on and their states.

If you decide to use a skill bar, you need to set a reference to it here. Each skills component also has an ID in order to save the state of your skills and restore them.

Actions / Conditions / Triggers

This page provides an overview of every action, condition and trigger that comes with this module.

Actions

Name	Description
Assign Skill	Assigns a skill to the skill bar at a certain index on the selected target
Cancel All Skills	Cancels all executing skills on the selected target
Cancel Skill	Cancels an executing skill on the selected target
Change Skill Execution Settings	Changes the cooldown, execution and cast time of a skill on the selected target
Change Skill Execution Settings Variables	Same as "Change Skill Execution Settings", but supports Game Creator variables
Change Skill Icon	Changes the icon of a skill on the selected target
Change Skill Type	Changes the type of a skill on the selected target
Debug Skill	Prints out the name and state of the selected skill (unlocked or locked) on the selected target
Execute Skill	Executes the selected skill on the selected target
Reset Skill	Resets the skill on the selected target
Reset All Skills	Resets all skills on the selected target
Set Skill State	Sets the selected skill on the selected target to unlocked or locked
Skill Tree UI	Opens / Closes the skill tree UI for the selected target
Unassign Skill	Unassigns the assigned skill on the skill bar at a certain index on the selected target

Conditions

Name	Description
Skill State	Is the selected skill on the selected target unlocked or locked?

Is Skill Executing	Is the selected skill on the selected target executing or not?
Is Skill in Cooldown	Is the selected skill on the selected target in cooldown or not?
Skill Tree Element Skill Type	Is the selected skill on the selected target of the selected type or not?
Skill Tree UI State	Is the skill tree UI opened or not?

Triggers

Name	Description
On Skill Activation Start	Fires when the selected skill on the selected target starts the activation phase
On Skill Activation End	Fires when the selected skill on the selected target ends the activation phase
On Skill Casting Start	Fires when the selected skill on the selected target starts the casting phase
On Skill Casting End	Fires when the selected skill on the selected target ends the casting phase
On Skill Execution Start	Fires when the selected skill on the selected target starts the execution phase
On Skill Execution End	Fires when the selected skill on the selected target ends the execution phase
On Skill Finishing Start	Fires when the selected skill on the selected target starts the finishing phase
On Skill Finishing End	Fires when the selected skill on the selected target ends the finishing phase
On Skills Reset	Fires when skills reset button is pressed for the selected target
On Skill State Change	Fires when the selected skill on the selected target changes to the selected

Skill Tree

Skills for Game Creator allows you to add a skill tree system to your game.



There are two skill tree prefabs, one for the **automatic** generation of a skill tree based on the types and skills one relies on, and a **manual** one. Both feature the following functionality:

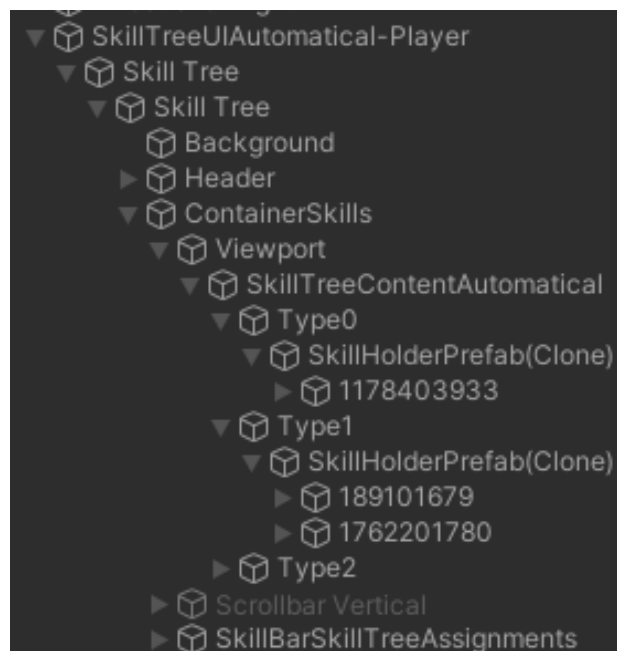
- **Context Menu support:** Unlock a skill, add or remove it to/from the skill bar
- **Hover over informations:** Hover over a skill to see informations about it like the required level
- **Skill Bar Assignments** via keyboard

You can find the prefabs under
Plugins/MiTschMRStudios/Skills/Resources/SkillTree

A skill tree is built like this:

- Root
 - Header
 - ContainerSkills
 - SkillTreeContentAutomatical
 - SkillType
 - SkillHolder
 - Skill1
 - Skill2
 - Skill...
 - Skill Bar

A skill type contains so called skill holders, which contain all skills that directly rely on each other and are displayed horizontally in one row. Let's see it in action:



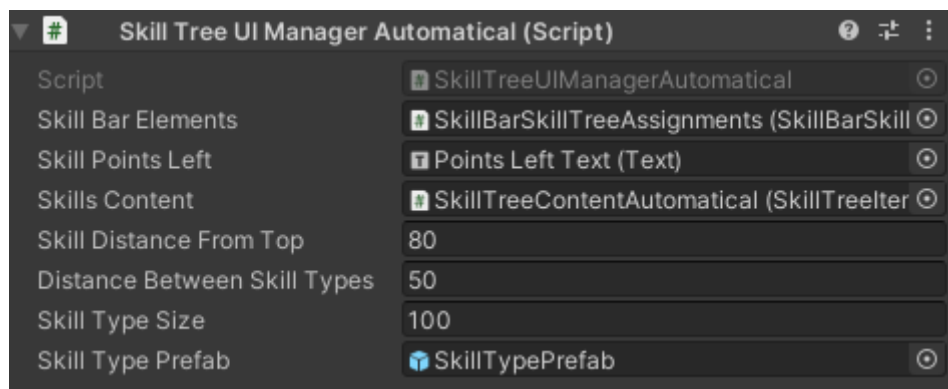
When you compare the hierarchy with the skill tree in the above picture, you will immediately notice what belongs to what.

Automatic Skill Tree

The automatic skill tree, as already mentioned, uses the type of the skill and the ones it relies on to set it up. The prefab consists of three parts, the root gameobject with the *SkillTreeUIManagerAutomatical* component on it and the *SkillTreeContentAutomatical* prefab with the *SkillTreeItemsListAutomatical* component on it.

Skill Tree Root

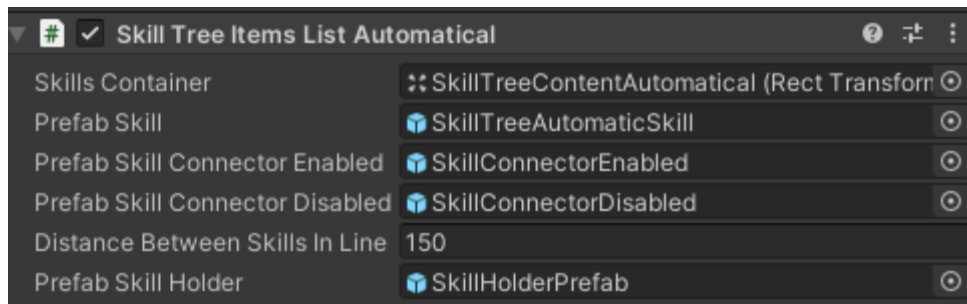
The *SkillTreeUIManagerAutomatical* component has seven settings and references you can set:



Setting	Description
Skill Bar Elements	References the Skill Bar where you can assign skills in the UI via keys
Skill Points Left	Gameobject with a text component on it that displays the amount of skill points left
Skills Content	Gameobject that will have the skills structure below it
Skill Distance From Top	Vertical distance from the top (bottom of the header)
Distance Between Skill Types	Vertical distance between each type section of skills
Skill Type Size	Defines how big the initial cell for a skill type section is
Skill Type Prefab	Prefab that should be spawned as a holder of all skills of this type. Has a vertical layout component on it for structuring purposes

Skill Tree Content

This gameobject has the component *SkillTreeItemsListAutomatical* on it, which has the following settings and references:

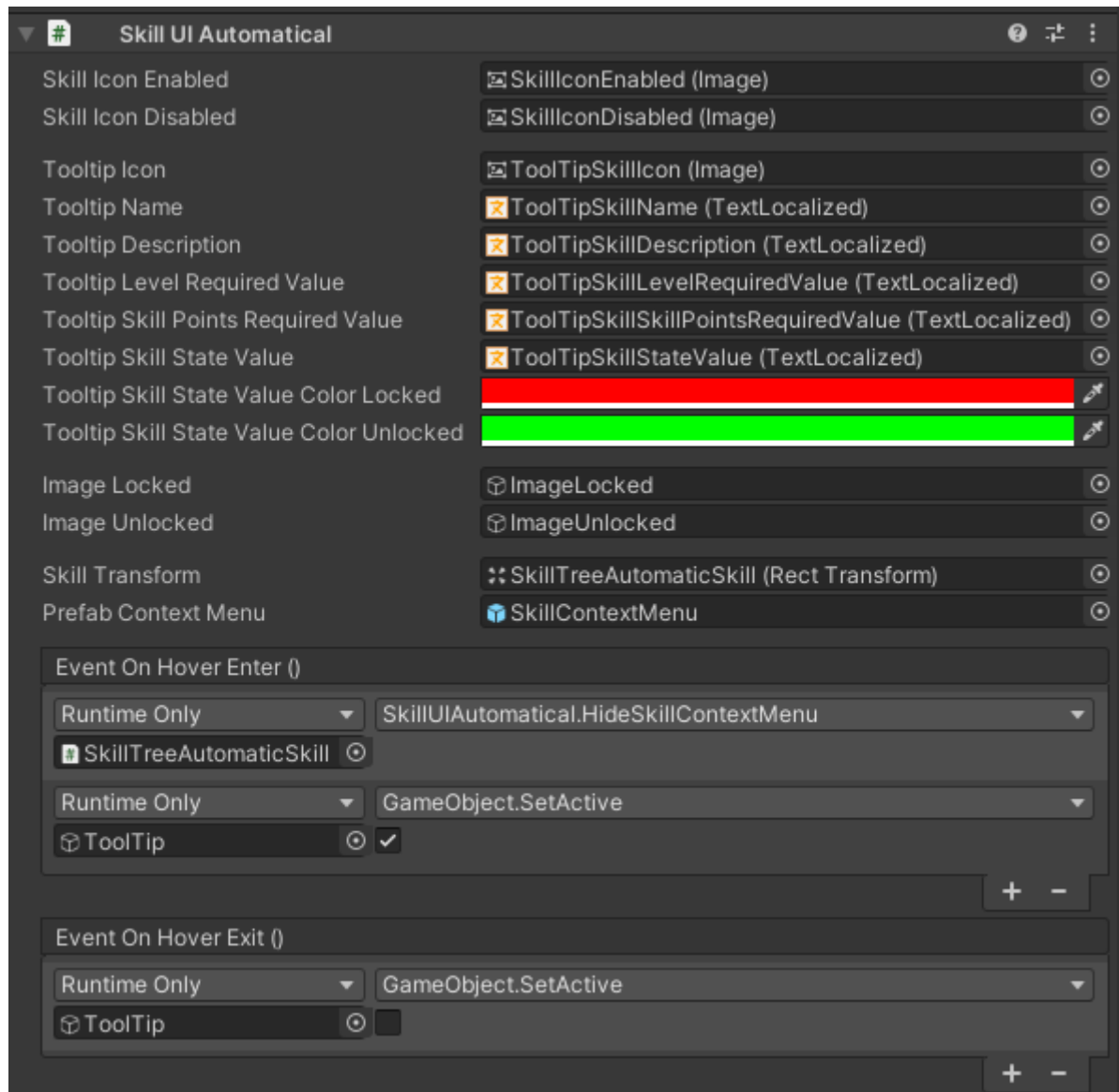


Setting	Description
Skills Container	The Rect Transform component of this gameobject. Will spawn the skill holders below it.
Prefab Skill	Prefab of the skill gameobject
Prefab Skill Connector Enabled	Prefab for the enabled skill connector
Prefab Skill Connector Disabled	Prefab for the disabled skill connector
Distance Between Skills In Line	Distance between the skills in one row, meaning the ones that rely on each other. In the middle of them the skill connectors will be spawned.
Prefab Skill Holder	Prefab for the skill holder

I recommend to play with the settings so you get a feeling for it.

Skill Element

A skill is the heart component of a skill tree. It has many settings and references:



Setting	Description
Skill Icon Enabled	References an image component used for the icon when the skill is unlocked
Skill Icon Disabled	References an image component used for the icon when the skill is locked
Tooltip Icon	References an image component used for the skill icon in the tooltip
Tooltip Name	References a text component where the name of the skill will be shown

Tooltip Description	References a text component where the description of the skill will be shown
Tooltip Level Required Value	References a text component where the required level for the skill will be shown
Tooltip Skill Points Required Value	References a text component where the required skill points for the skill will be shown
Tooltip Skill State Value	References a text component where the current state of the skill will be (locked / unlocked)
Tooltip Skill State Value Color Locked	The color in which the referenced text component of the tooltip skill state value is painted when the skill is locked
Tooltip Skill State Value Color Unlocked	The color in which the referenced text component of the tooltip skill state value is painted when the skill is unlocked
Image Locked	References an image component that shows if a skill is locked
Image Unlocked	References an image component that shows if a skill is unlocked
Skill Transform	The Rect Transform component of this gameobject
Prefab Context Menu	Prefab of the context menu
Event On Hover Enter	Defines the logic that will be executed when you start hovering over the skill
Event On Hover Exit	Defines the logic that will be executed when you exit hovering over the skill

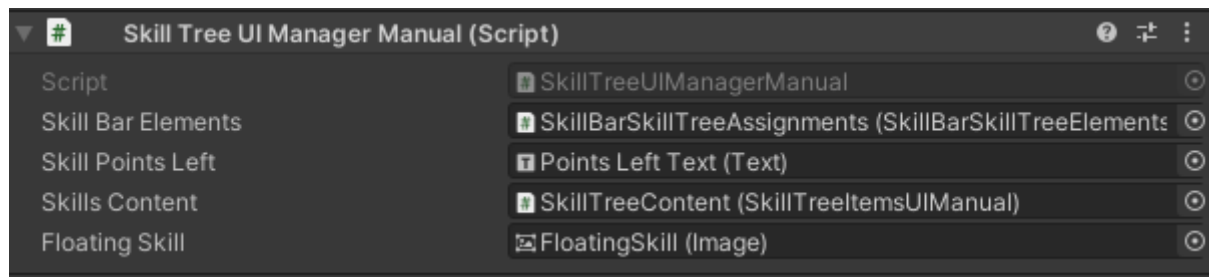
The automatic skill tree has the limitation that it displays skills always in one row, even if you have multiple skills one relies on. It takes the first one and attaches it to this one. If you want skills to rely on multiple skills, you better use the manual skill tree described below.

Manual Skill Tree

For more control over the look, use the manual skill tree. Let's see which settings they have.

Skill Tree Root

The root gameobject has only three references to set:



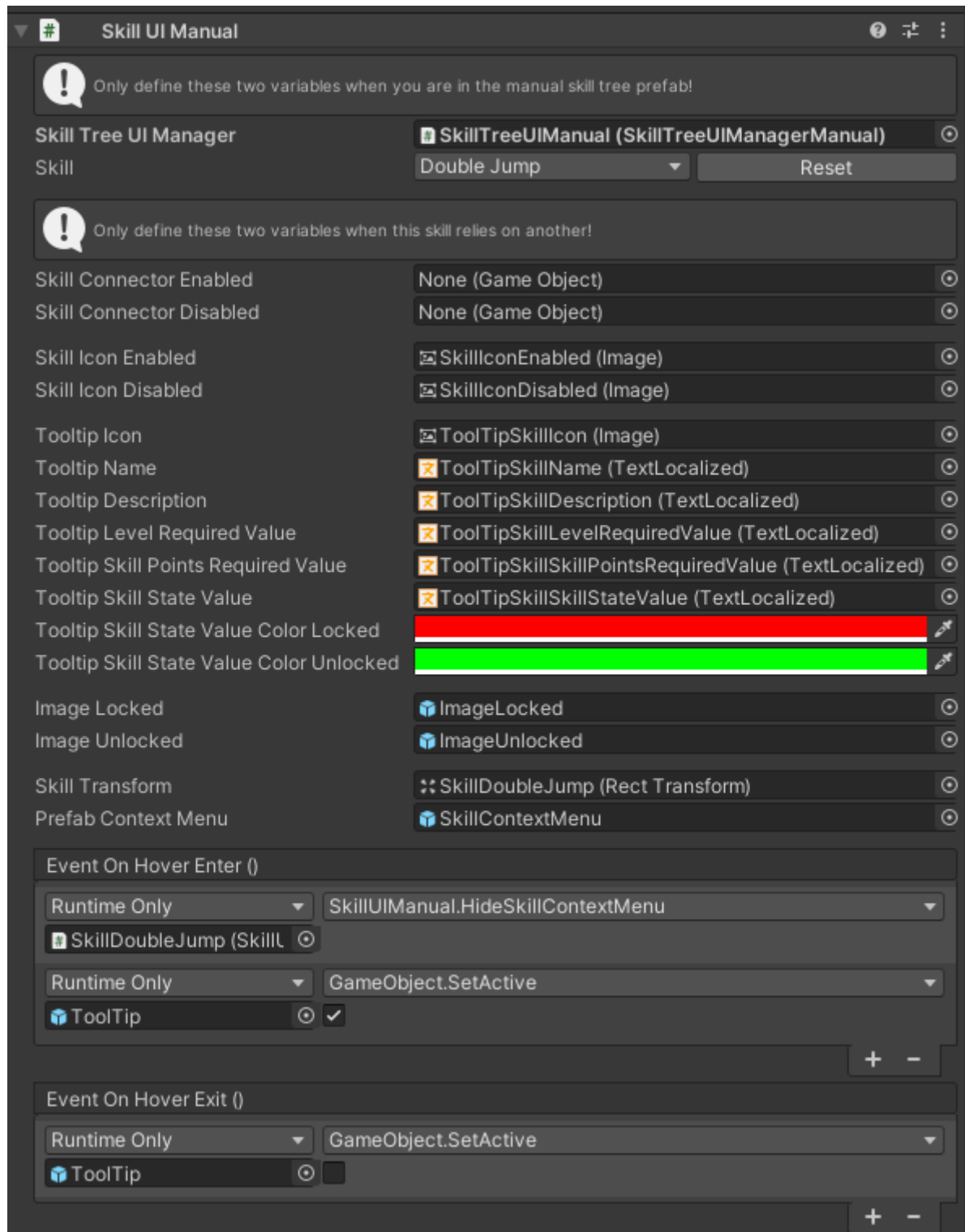
Setting	Description
Skill Bar Elements	References the Skill Bar where you can assign skills in the UI via keys
Skill Points Left	Gameobject with a text component on it that displays the amount of skill points left
Skills Content	Gameobject that will have the skills structure below it
Floating Skill	Image component of the gameobject used for the drag/drop functionality

Skill Tree Content

This gameobject only needs the *SkillTreeItemsListManual* component on it and is then already set up. Every skill that you add below is detected automatically.

Skill Element

The manual skill element has four settings more than the automatic one:



Setting	Description
Skill Tree UI Manager	The root object of the manual skill tree (automatically set in the automatic one)
Skill	The skill it represents
Skill Connector Enabled	References the enabled skill connector used between two skills
Skill Connector Disabled	References the disabled skill connector used between two skills

Skill Bar

Skill Bar Skill Tree

The skill bar in the skill tree UI is used to assign skills to the skill bar of the gameobject that opened the skill tree UI. It looks like the following:

- Root
 - Skill Bar Element 1
 - Skill Bar Element 2
 - Skill Bar Element x
 - ...

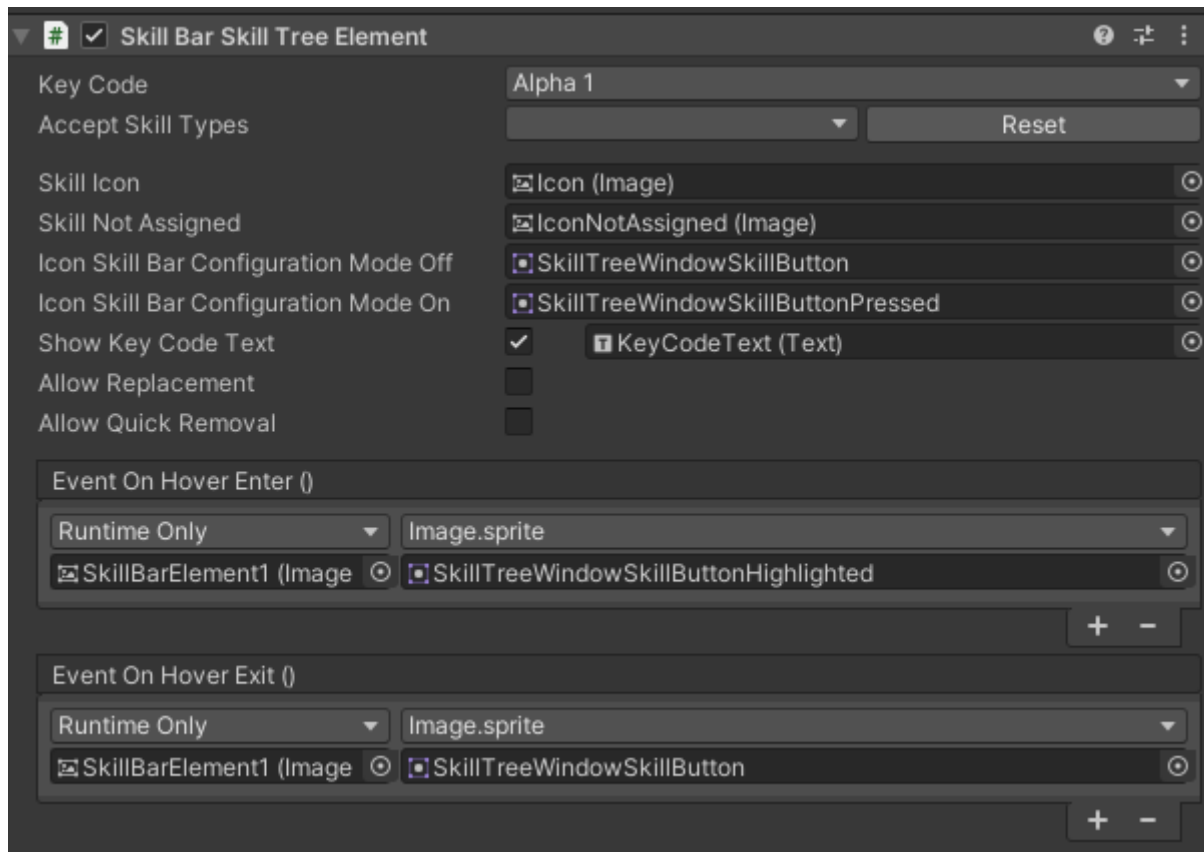
This feature is not a requirement for the skill tree UI, it can be used without it.

Root

The root gameobject has one component called *SkillBarSkillTreeElements* which automatically searches all *SkillBarSkillTreeElement* components that are used on the elements.

Skill Bar Skill Tree Element

The skill bar skill tree element has the following settings and references:



Setting	Description
Key Code	The key code used to assign a skill to this element
Accept Skill Types	Defines whether only a certain type of skill may be assigned
Skill Icon	References an image component that shows the icon of a skill when assigned
Skill Not Assigned	References an image component that is shown when no skill is assigned
Icon Skill Bar Configuration Mode Off	References an image that is shown when the skill bar configuration mode is turned off
Icon Skill Bar Configuration Mode On	References an image that is shown when the skill bar configuration mode is turned on
Show Key Code Text	If activated, the referenced text component is activated when the skill bar configuration mode is activated

Allow Replacement	Checkbox to determine whether to allow replacement on the skill bar
Allow Quick Removal	Checkbox to determine whether to allow the quick removal feature on the skill bar
Event On Hover Enter	Defines the logic that will be executed when you start hovering over the element
Event On Hover Exit	Defines the logic that will be executed when you stop hovering over the element

Skill Bar

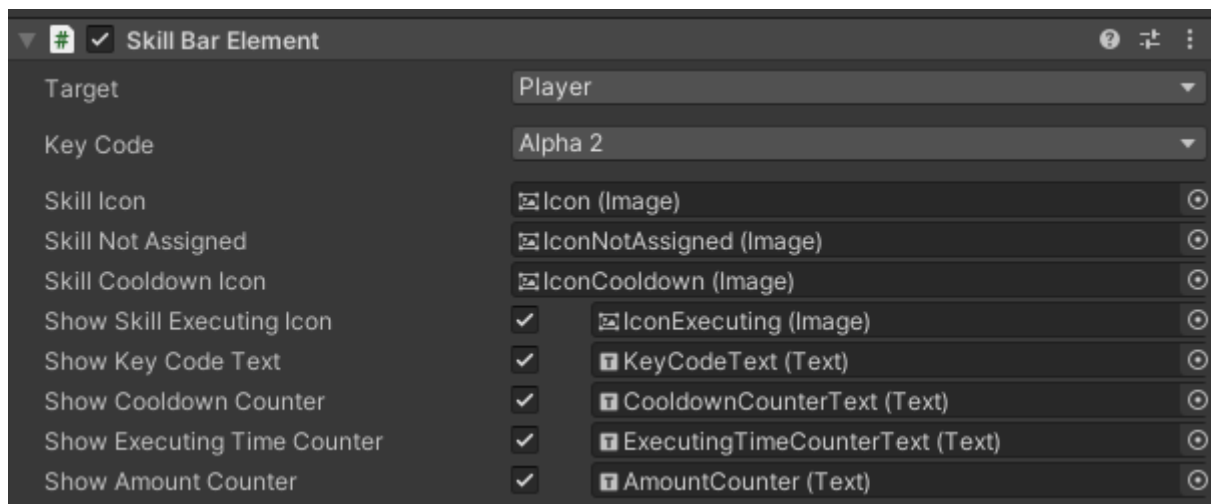
The [Skills component](#) has a toggle to turn on or off the use of this feature. It also consists of a root and slightly different skill bar elements. This one needs to be in the scene, referenced in the skills component.

Root

The root gameobject has one component called *SkillBarElements* which automatically searches all *SkillBarElement* components that are used on the elements.

Skill Bar Element

The skill bar element has the following settings and references:



I only list the differences between the above one and this one here:

Setting	Description
Target	The target on which the skill bar settings are applied to
Key Code	The key used to execute the skill
Skill Cooldown Icon	The Image component shown when the skill is in cooldown

Show Skill Executing Icon	If activated, this shows the currently executing skill
Show Key Code Text	If activated, this shows the referenced text component which is used to display the key to press to execute a skill
Show Cooldown Counter	Checkbox with Text component reference that shows the remaining cooldown time
Show Executing Time Counter	Checkbox with Text component reference that shows the remaining execution time
Show Amount Counter	Checkbox with Text component reference that shows the remaining skill executions of a stored skill

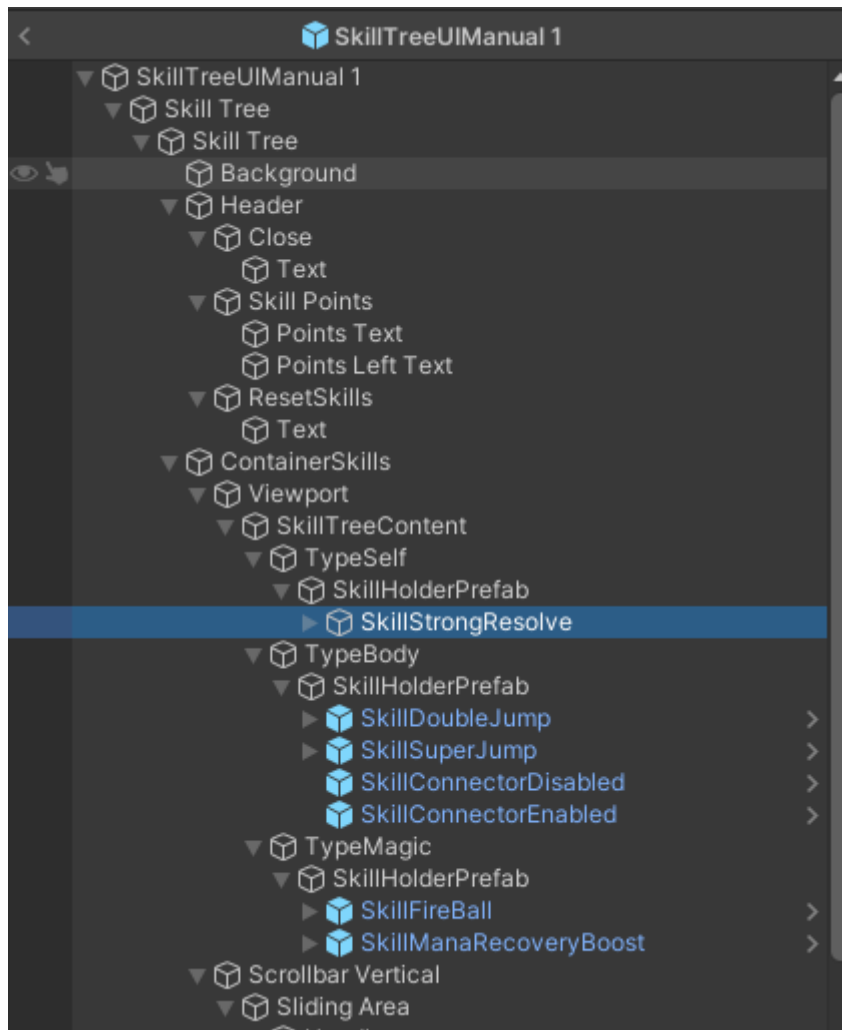
FAQ

Problem: You get a `NullReferenceException` when opening the skill tree and no skills are visible.

`NullReferenceException: Object reference not set to an instance of an object`
`MiTschMR.Skills.SkillTreeItemsUI.UpdateSkills (...)` (at ...)



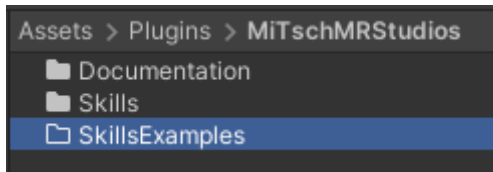
Solution: Unpack all prefabs in the skill tree by right clicking on the blue gameobjects and selecting "Unpack" --> "Unpack Completely" (may differ depending on your version of Unity). Screenshot shows the skill tree prefab with prefabs in it.



Problem: When disabling the modules, a warning is thrown and the “Enable” button is still disabled.

A meta data file (.meta) exists but its folder 'Assets/Plugins/MiTschMRStudiosData/SkillsExamples' can't be found, and has been created. Empty directories cannot be stored in version control, so it's assumed that the meta data file is for an empty directory in version control. When moving or deleting folders outside of Unity, please ensure that the corresponding .meta file is moved or deleted along with it.

Solution: Locate the folder and delete it.

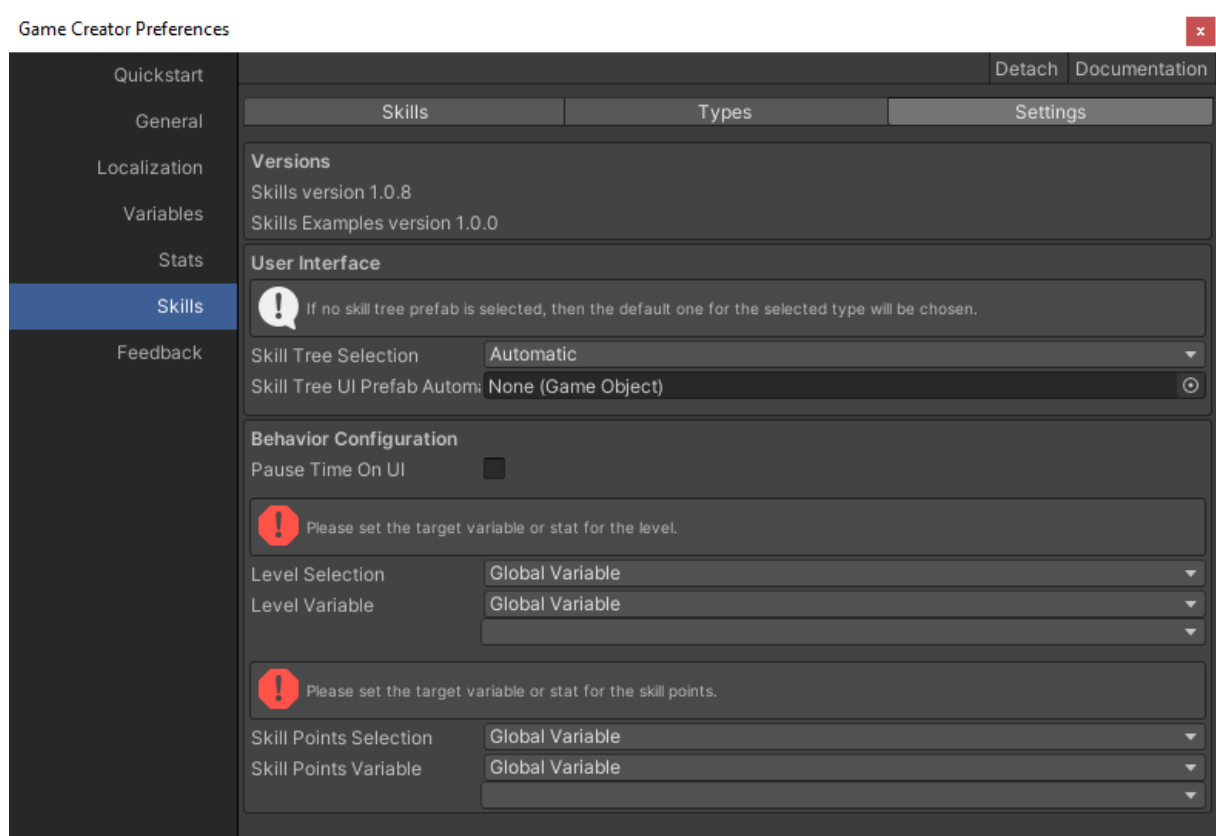


An empty folder is shown with an empty folder icon.

Example Setup

Preferences Window

The first thing you should do is set up a global variable or a stat for the HP and XP. This can be done in the Skills Preferences Window:



For more information see [Settings](#).

Scene

Now you are prepared for the scene. Either make a new or an existing one, where you feel comfortable. In order to make a gameobject be able to use skills, it needs the [Skills Component](#) attached to it.

On the Skills component you can choose to use a [Skill Bar](#) or not. If you decide to use one, you need to drag and drop the Skill Bar prefab into the scene and assign this gameobject to the variable on the gameobject with the Skills component. The prefab can be found under *Assets/MiTschMRStudios/Skills/Prefabs/SkillBar*.

If you want to use a Skill Bar you also need to add the SkillBarSillTreeAssignments prefab to the SkillTreeUIManual or SkillTreeUIAutomatic prefabs (done by default). If you decide not to use this feature, you should delete it from the SkillTreeUI prefabs. These can be found under *Assets/MiTschMRStudios/Skills/Resources/SkillTree*.

Refer to the Skills Examples for a setup.