using-pre-built-logic-components

Scene Management

Overview

Scene Management Logic Template components are pre-built components aimed to help game developers readily build logic that affect the entire scene.

Logic Template	Description
Checkpoint	Restarts the game from a specific point if you fail a challenge or lose a life
Update Timer	Updates the timer to a new specified value
Reset Timer	Resets the timer to zero
Load Scene	Loads a New Scene
Random Level Selector	Loads a random new scene on game start instead of the default scene

All Overall Game Logic Template Components

Checkpoint

A checkpoint is a designated spot where player progress is saved automatically or manually. These markers allow you to restart the game from that specific point if you fail a challenge, lose a life, or need to pause the game.

To add the checkpoint logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Checkpoint under the header "Game".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
`Play SFX`	Choose a short chime to play when you arrive at checkpoint
`Play VFX`	Choose a visual effect to play when you arrive at the checkpoint
`BroadcastData`	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the you arrive at the checkpoint.

You can further customize the Checkpoint logic template in T# by accessing its T# Wrapper - CheckpointTemplate.

Update Timer

In the game, significant events such as completing tasks or defeating enemies can adjust the timer, adding bonus time or starting time-limited challenges. Conversely, mistakes or failures can reduce the time. The Update Timer template is used to handle these changes.

To add the Update Timer logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Update Timer under the header "Game".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
`Update When`	Select a trigger from the dropdown to activate the logic template: • **Player Touches**: Resets the timer when the player touches the selected object. • **Other Object Touches**: Resets the timer when another object touches the selected object. • **Clicked**: Resets the timer when you click the selected object. • **Broadcast Listened**: Resets the timer when it receives a broadcast
`Operation`	Define the operator that will modify the timer. Four operators are allowed - Add, Subtract, Multiply and Divide
`Update By`	The quantity specified in this context will determine the extent to which the timer is modified.
`Sound Effect on Start`	Choose a sound effect to play when the timer value is updated.
`Visual Effect on Start`	Choose a visual effect to play when the timer value is updated.
`Broadcast on Update`	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the timer is updated.
`Execute always`	This toggle, when activated, will always execute this. When off, it will execute it only once.

You can further customize the Checkpoint logic template in T# by accessing its T# Wrapper - UpdateTimerTemplate.

Reset Timer

The Reset Timer behavior resets the game timer to its initial value without affecting the game experience. If it's a countdown timer, the time is reset to the starting value. If it's a count-up timer, the time is reset to 0 seconds.

You can customize the template parameters according to the game requirements:

To add the Reset Timer logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Reset Timer under the header "Game".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
`Reset When`	Select a trigger from the dropdown to activate the logic template: • **Player Touches**: Resets the timer when the player touches the selected object. • **Other Object Touches**: Resets the timer when another object touches the selected object. • **Clicked**: Resets the timer when you click the selected object. • **Broadcast Listened**: Resets the timer when it receives a broadcast
`Broadcast`	Define a broadcast to be generated that can trigger other actions. The broadcast is sent when the timer is reset.

While there is no pre-built T# Wrapper available to customize the Reset Timer Logic Template you can write your own code in T# to implement this logic from scratch.

Load Scene

The Load Scene logic template enables you to transition from one game environment (scene) to another, such as progressing from one level to the next or exploring a new area.

To add the Load Scene logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Load Scene under the header "Game".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
`Load Scene When`	Choose from this dropdown when to transition to a different scene: • **Broadcast Listened**: After the object receives a broadcast message. • **Player Touch**: When the player touches the object. • **Other Object Touch**: When another object touches the object. • **Clicked**: When you click on the object.
`Scenes to Load`	Click the + button to choose the next scene to load when the trigger condition is met. It displays a list of all available scenes in the game.
`Can Repeat Previous Level`	Toggle that specifies whether the player can repeat the previous level

While there is no pre-built T# Wrapper available to customize the Load Scene Logic Template you can write your own code in T# to implement this logic from scratch.

Random Level Selector

The Random Level Selector Logic Template allows you to randomly load a scene from a list of predefined scenes. It is different from the Load Scene Logic Template because here the scene is chosen randomly from a list of multiple scenes

To add the Random Level Selector logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Random Level Selector.
3. Drag and drop it onto the desired asset.

You can then tailor the following parameters in Advanced Mode to match your needs:

Parameter	Description
`When`	Select When to Load a Random Level from a dropdown**:** • On Game Start • On Broadcast Listened
`Scenes`	Add a selection of scenes to randomly choose from for loading.

While there is no pre-built T# Wrapper available to customize the Load Scene Logic Template you can write your own code in T# to implement this logic from scratch.

Mechanics

Overview

This table outlines various mechanics that are commonly used in games and enable different interactions and behaviors for game objects, such as being collected by the player, creating enhancements, applying forces, enabling rotations, and more.

Logic Template	Description
Collectable	Enables an object to be collected by the player and update the game score. Used in Power-ups.
Teleport Player	Instantly spawns the Player in a new specified position
Jump Pad	Creates a jump enhancement for the player upon contact
Carryable	Enables an Asset to be carried by the Player. The Asset will now move with the Player
Deposit	Enables the Player to transfer the Carriable Asset and deposit it to a new Asset which is a storage
Modify Carryable	Modifies the number of carryables you have
Kill Player	Respawns the player to the start of the level
Hinge Joint	Enables assets to rotate about a defined hinge like a dore
Explosive Force	Applies a force / impulse on a radius
Add Force	Applies a force on an object and allows it to follow physics
Treadmill	Enables treadmill-like motion on contact
Multipoint Move	Shifts the Asset from its starting spot through a path of straight or curved points as needed.
Attach Object	Parents an object to another object

List of all Mechanics Logic Template Components

Collectable

A Collectible is a logic template added to an asset in a game that can be collected to increase score or to achieve another goal.

One or multiple triggers can be created, resulting in one or multiple outcomes.

To add the Collectable logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Collectable under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
Collect When	Choose when the item is “collected”: • when player touches • when clicked on screen • when in a magnet range • When the player has to stay near it for a specific time
Sound Effect on Start	Choose a short chime to play when item is collected
Visual Effect on Start	Choose a small visual effect to play when item is collected
Score Group	The point of the Collectable will be contributed to the score group. You can either add it to Main Score group or make your own custom group
Update Score By	Enter a numerical score value to update when collected. ✨ Note: to reduce a score when collected, enter a negative value!
IsMultiLevel	Enabling this parameter can upgrade to a higher value on level up

Parameter	Description
Broadcast On Collection	Choose to enter a broadcast that can be used as a trigger for any other behaviour. The broadcast is sent when the item is collected

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - CollectableTemplate

Teleport Player

The Teleport Player logic template can be used to teleport the player from one location to another in response to a specified trigger.

To add the Teleport Player logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Teleport Player under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
Teleport When	You can choose the trigger to activate the behaviour - When the game starts - After a broadcast message has been received by the object - When the player touches the object - When a different object touches the object - When you click on the object
Teleport	You can choose the coordinates where you want the player to be teleported.
Loop-able	This allows you to loop the movement of the object. It appears as if it is oscillating between 2 different points.
Interval	Intervals add a delay between the back-and-forth movement of the object during the loop.
Move By	You can define how many units and in what axis the object will move
Sound Effect on Start	Choose a sound effect to play when the object starts to move
Visual Effect on Start	Choose a visual effect to play when the object starts to move
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the object stops moving..

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - TeleportTemplate

Jump Pad

The Jump Pad boosts the player's jump height when attached to an asset. Upon touching an object with the Jump Pad logic template, the player's jump is elevated. After the jump action, the jump height is restored to the initial value.

To add the Jump Pad logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Jump Pad under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameters	Description
Play SFX	Choose a sound effect to play when the player jumps
Play VFX	Choose a visual effect to play when the player jumps
Jump Force	Define the multiplier by which the existing jump height will be multiplied for that instance
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. `The broadcast is sent when the player jumps at an increased height.`

{% hint style="info" %} Your Jump Height should ideally be greater than 2 for the Jump Pad to work. Small jump height values will not lead to an increased jump height. You can find Jump Height in the Player Controller Drawer. Eg: A jump Height of 0.1 with a Jump Force of 10 will change the total jump height to 0.01 which is lower than the initial height. {% endhint %}

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - JumpPadTemplate

Carryable

A carriable is a logic template that, when attached to a game asset, enables the player to carry it.

To add the Carryable logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Carryable under the header "Mechanics".
3. Drag and drop it onto the desired asset.

To add the Carryable behavior to an asset, follow these steps:

1. Select the asset you wish to apply the Carriable behavior to.
2. In the Inspector panel, click on **Add Behavior**.
3. From the list of behaviors, choose Carriable.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
Group	You can group the carryables in different categories using this dropdown.
Carry on	Choose when the item is “carried”, using the dropdown: • when player touches • when clicked on screen • when in a magnet range • When the player has to stay near it for a specific time
Play SFX	Choose a short chime to play when item is collected
Play VFX	Choose a small visual effect to play when item is collected
Size of carriable	
Score Group	The point of the Collectable will be contributed to the score group. You can either add it to Main Score group or make your own custom group
Lerp	
Lerp Time	
Score	Enter a numerical score value to update when collected.
IsMultiLevel	Enabling this parameter can upgrade to a higher value on level up

Parameter	Description
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behaviour. The broadcast is sent when the item is collected

Configure the player setting for the carryable behaviour:

1. Navigate to the essentials tab from the builder menu.
2. Select the PlayerControllerDrawer from the builder panel. this would open the inspector panel
3. Navigate to carryable properties section in the inspector pannel.

Parameter	Description
Locator for cariable	set the position of the carryable on the player using the record button
Limit	the number of carryables can be limited using this field
Stack offset	Using this option you can set the position of the carrible around the player

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Deposit

The deposit logic template when applied to an asset, allows it to collect a specified carryable. The deposit logic template cannot work independently when there is no carryable in the scene

To add the Deposit logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Deposit under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
Deposit when	Choose when the item is “Deposited”, using the dropdown: • when player touches • when other object touches • when clicked on screen • when in a magnet range • When the player has to stay near it for a specific time
Take Resource	Selects the asset group which serves as the currency
Persistent	Checking this box
Play SFX	Choose a short chime to play when item is collected
Play VFX	Choose a small visual effect to play when item is collected
Lerp	
Lerp Time	
Cost type	
Size of carriable	
Score Group	The point of the Collectable will be contributed to the score group. You can either add it to Main Score group or make your own custom group
Deposit rate	You can set the rate at which the carriable will get deposited.
Of Amount	

Parameter	Description
Score	Enter a numerical score value to update when collected.
IsMultiLevel	Enabling this parameter can upgrade to a higher value on level up
limit	
Show Progress	
Is Ascending	
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behaviour. The broadcast is sent when the item is collected
Broadcast stack empty	

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Kill Player

When the player contacts an object with the Kill Player behavior, the player is killed and respawns at the last checkpoint. This is useful when the player enters a danger zone or interacts with a hazardous object.

To add the Kill Player logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Kill Player under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
`Play SFX`	Choose a short chime to play when the player is killed
`Play VFX`	Choose a small visual effect to play on the object when the player is killed
`Broadcast On Respawn`	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the player is killed.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - KillPlayerTemplate

Multi-Point Move

Multi-Point Move is a logic template that allows you to instruct an Asset to follow a path made up of several points once its Start Event occurs. This path can be straight lines or curves between the points, giving you many choices for how the Asset moves.

The Asset can automatically turn to make sure a specific side always points towards the path, just like how our face turns to the direction we are moving.

To add the Multi-Point Move logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Multi-Point Move under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
Move On	

Parameter	Description
	This is a dropdown from where you need select any one of the following Start Events for Interpolate Points to begin executing: 1. When the behavior when the game starts: Select "Game Start" 2. When any other Asset touches the currently selected Asset: Select "Other Object Touch" 3. When a particular broadcast is generated in the game: Select "Broadcast Listened" and specify the name of the signal to listen to 4. When the player touches the currently selected Asset: Select "Player Touchers" 5. When the Asset is clicked: Select "On Click"
Points	This helps you specify the coordinates of the multiple points through which the Asset will move in a path. Click the + button to add points and - button to remove an existing point
Speed	This is an input field where you can enter a number that represents the speed of Asset movement
Turn To Points	This toggle button ensures that only one side of the Asset always faces forward. When activated, it adjusts the Asset's orientation to maintain this specific direction during movement.
Delay at Point	This field lets you set a delay in seconds. During this delay, the Asset will not move. After the time passes, it will start moving again.
Loop	This toggle button, when activated, enables the movement to repeat continuously
Is Curve	This toggle button alters the Asset's trajectory between points to follow a curved path instead of a linear one.
Interpolate Types	This dropdown lets you pick how the object moves: • For forward movement only, select **One Direction** • For back-and-forth movement, select **Ping Pong**
	The Broadcast Signal option allows you to create a game signal that other can act as the Start Event for other behavior blocks to execute. You can choose "Game Win", "Game Lose", or create your own custom signal. For a custom signal, you must select "Custom" from the dropdown and enter a name in the input field.
Broadcast Type & Broadcast Signal	The Broadcast Type dropdown lets you specify when the broadcast signal will be sent. There are three options to choose from: • If you want no broadcast to be sent, Select Never • If you want to send a broadcast after finishing one whole movement from start to end, select End • If you want to send the broadcast signal every time the object pauses , select At Every Pause

If you want to further customize this logic template, you can do so by accessing its T# Wrapper- MoveBetweenPointsTemplate

Modify Carryable

The Modify Carryable Logic Template allows you to modify the number of carryables carried.

To add the Modify Carryable logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Modify Carryable under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can customize the below-mentioned parameters according to your requirements:

Parameter	Description
`Modify When`	This is a dropdown from where you need select any one of the following Start Events for the template

Parameter	Description
	1. When any other Asset touches the currently selected Asset: Select "Other Object Touch" 2. When a particular broadcast is generated in the game: Select "Broadcast Listened" and specify the name of the signal to listen to 3. When the player touches the currently selected Asset: Select "Player Touchers" 4. When the Asset is clicked: Select "On Click"
`Play VFX`	Choose a small visual effect to play on the execution
`Play SFX`	Choose a sound effect to play on the execution
`Haptics`	Select Haptics from a dropdown
`Modifier Group`	Select a which carryable needs to be modified
`Execute Always`	Toggle to Specify if this always needs to be executed
`Modifier`	Select a modifier to the carriable - Add, Subtract, Multiply
`Modify By`	Amount / Value to be modified by

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Hinge Joint

The Hinge Joint Logic Template allows you to make an asset rotate around a hinge joint to mimic the movement of a door.

To add the Hinge Joint logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Hinge Joint under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
`Axis`	Helps specify the axis of rotation
`Anchor`	Specify the anchor point around which hinge movement must happen
`Can Spin Back`	Toggle to specify if the object can spin back to original position after the rotation.

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Explosive Force

The Explosive Force Logic Template allows the attached object to exert a sudden explosive force or impulse on nearby objects within a certain radius.

To add the Explosive Force logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Explosive Force under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
`Explode When`	

Parameter	Description
	This is a dropdown from where you need select any one of the following Start Events for the template 1. When any other Asset touches the currently selected Asset: Select "Other Object Touch" 2. When a particular broadcast is generated in the game: Select "Broadcast Listened" and specify the name of the signal to listen to 3. When the player touches the currently selected Asset: Select "Player Touchers" 4. When the Asset is clicked: Select "On Click"
`Force`	Specify the value of the force to be applied
`Radius`	Specify the radius where the force is felt
`Explode SFX`	Choose a short chime to play on execution
`Explode VFX`	Choose a small visual effect to play on execution
`Broadcast`	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent after execution

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Add Force

The Add Force Logic Template allows you to apply a force on an asset and enable it to act according to the laws of physics.

To add the Add Force logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Add Force under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
	This is a dropdown from where you need select any one of the following Start Events for the template
`Add Force When`	1. When any other Asset touches the currently selected Asset: Select "Other Object Touch" 2. When a particular broadcast is generated in the game: Select "Broadcast Listened" and specify the name of the signal to listen to 3. When the player touches the currently selected Asset: Select "Player Touchers" 4. When the Asset is clicked: Select "On Click"
`Force`	Specify the value of the force to be applied
`Repeat Mode`	Dropdown to select the nature of repetition if any - Single, Repetitive, Periodic
`Period`	If periodic, specify the period
`Play SFX`	Choose a short chime to play on execution
`Play VFX`	Choose a small visual effect to play on execution
`Broadcast`	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent after execution

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Treadmill

The Treadmill Logic Template simulates treadmill movement for objects or the Player upon touch.

To add the Treadmill logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Treadmill under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
`Treadmill When`	This is a dropdown from where you need select any one of the following Start Events for the template 1. When the behavior when the game starts: Select "Game Start" 2. When any other Asset touches the currently selected Asset: Select "Other Object Touch" 3. When a particular broadcast is generated in the game: Select "Broadcast Listened" and specify the name of the signal to listen to 4. When the player touches the currently selected Asset: Select "Player Touchers" 5. When the Asset is clicked: Select "On Click"
`Play SFX`	Choose a short chime to play when the player starts the treadmill
`Play VFX`	Choose a small visual effect to play on the object when the player starts the treadmill
`Treading Speed`	Specify the speed of the treadmill
`Treading Direction`	Specify the direction of movement of the treadmill
`broadcastData`	Specify a broadcast signal to be generated after execution

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Attach Object

The Attach Object Logic Template allows you to attach one object to another, making the new object a child of the original object.

To add the Attach Object logic template, follow these steps:

1. Go to the Logic Tab.
2. Select Attach Object under the header "Mechanics".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
`AttachOn`	This is a dropdown from where you need select any one of the following Start Events for the template 1. When the behavior when the game starts: Select "Game Start" 2. When any other Asset touches the currently selected Asset: Select "Other Object Touch" 3. When a particular broadcast is generated in the game: Select "Broadcast Listened" and specify the name of the signal to listen to 4. When the player touches the currently selected Asset: Select "Player Touchers" 5. When the Asset is clicked: Select "On Click"
`Attach_To` `Attach To`	Select from the dropdown whether you want to attach the object to the Player or A Game Object.

Parameter	Description
	If you chose GameObject, drag and drop the game object references from the layers here. This will be set to None if you selected Player on Attach_To
`KeepWorldPos`	Toggle to indicate if you want to keep the world position of the object
`Offset`	Indicate the offset between the parent and the child

Currently, there's no T# Wrapper available to customize this logic template beyond the scene editor's capabilities. However, you can write your own code in T# to implement this logic from scratch.

Actions

Overview

The table below provides an overview of various action logic templates and their descriptions for handling assets in a scene.

Logic Template	Description
Destroy	Destroys the Asset from the scene
Set Position	Changes the Asset's position
Advance Instantiate	Spawns an instance of the player (with advanced settings)
Grow / Shrink	Increases or decreases the size of the Asset
Move	Moves the Asset in a straight line path to a specified new position from its starting point.
Rotate	Rotates the Asset about a chosen axis
MoveTo Player	Moves the Asset to the Player
Rotate Oscillate	Oscilates the Asset about a specified axis within a specified rotation about the initial position
Basic Instantiate	Spawns an instance of the player
Bump	Bounce back when you run into it

List of all Action Logic Components

Destroy

The Destroy logic template is used when you want to remove or destroy assets from the game scene on a certain trigger. The asset with this logic template are destroyed without any trace of them being there.

This logic template is very similar to the Collectable behaviour, the only difference being that the Collectable contributes to a score group and this does not. In both logic templates, the asset disappears after the trigger.

To add the Destroy logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Destroy under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Destroy When	You can choose the trigger when to destroy the asset. - When the game starts

Parameters	Description
	- When a different asset touches the asset having the behaviour - After a broadcast message has been received by the asset. - When the player touches the asset. - When the object is clicked.
Play SFX	Choose a sound effect to play when the asset is destroyed.
Play VFX	Choose a visual effect to play when the asset is destroyed.
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the asset is destroyed.
Destroy After	The time in seconds after which the asset disappears from the scene.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - DestroyTemplate

Set Position

The Set Position logic template enables you to specify the position of an asset.

To add the Set Position logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Set Position under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Set Position on	- When the game starts - When a different asset touches the asset having the behaviour - After a broadcast message has been received by the asset. - When the player touches the asset. - When the object is clicked.
Target	Specify the target destination position either by recording or entering the target coordinates in the fields X, Y and Z.
Play SFX	Choose a sound effect to play on execution
Play VFX	Choose a visual effect to play on execution
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the asset is destroyed.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - SetPositionTemplate

Advance Instantiate

The advanced instantiation logic template allows you to spawn asset instances in the game scene during runtime.

To add the Advance Instantiate logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Advance Instantiate under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Destroy When	You can choose the trigger when to destroy the asset. - When the game starts - When a different asset touches the asset having the behaviour - After a broadcast message has been received by the asset. - When the player touches the asset. - When the object is clicked.
Play SFX	Choose a sound effect to play when the asset is destroyed.
Play VFX	Choose a visual effect to play when the asset is destroyed.
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the asset is destroyed.
Destroy After	The time in seconds after which the asset disappears from the scene.

There are currently no available T# Wrappers for this template.

Grow / Shrink

The Grow or Shrink behavior can be used to change the size of an object. Various types of triggers can be used to grow or shrink the object.

To add the Grow/Shrink logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Grow/Shrink under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Grow When	You can choose the trigger on which the object will start growing. - It can be at the start of the game - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.
Scale by	You can specify the scale by which an object's size will grow or shrink
Speed	You can define the speed by which object will grow or shrink
Repeat	You can define the number of time you want the behaviour to be executed
Repeat forever	You can set the change in size behaviour to forever by checking this checkbox.
Pause for	The duration of the pause between each cycle of behaviour execution can be adjusted here
Repeat type	You can use this option to select the type of motion while change There are two types motion the behaviour supports at the moment 1. Ping Pong 2. Same Direction
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the object stops moving.
Sound Effect on Start	Choose a sound effect to play when object starts moving
Visual Effect on Start	Choose a visual effect to play when object starts moving

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - GrowTemplate

Move

The Move behavior lets objects travel in a straight line between two points. They can move in any direction, switch between two points, and trigger other actions. For nonlinear paths with multiple points, use the MultiPoint Move template instead of move.

To add the Move logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Move under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
Move When	You can choose the trigger to activate the behaviour - When the game starts - After a broadcast message has been received by the object - When the player touches the object - When a different object touches the object - When you click on the object
Speed	You can define the speed of the object
Loop-able	This allows you to loop the movement of the object. It appears as if it is oscillating between 2 different points.
Interval	Intervals add a delay between the back-and-forth movement of the object during the loop.
Move By	You can define how many units and in what axis the object will move
Sound Effect on Start	Choose a sound effect to play when the object starts to move
Visual Effect on Start	Choose a visual effect to play when the object starts to move
Broadcast on End	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the object stops moving.
Stop When	You can choose the trigger to stop the movement - After a broadcast message has been received by the object - When the player touches the object - When a different object touches the object - When you click on the object
Resume When	You can choose the trigger to resume the movement - After a broadcast message has been received by the object - When the player touches the object - When a different object touches the object - When you click on the object.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - MoveTemplate

Rotate

When the Rotate logic template is applied to objects, they begin to rotate about their axis upon a specific trigger.

To add the Rotate logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Rotate under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Rotate When	You can choose the trigger on which the object will start rotating. - It can be at the start of the game - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.
Rotation Axis	You can choose the axis in which the object will rotate (X,Y,Z axis)
Speed	You can define the rotation speed of the object
Direction	You can define the direction of motion either clockwise or anticlockwise
Sound Effect on Start	Choose a sound effect to play when object starts moving
Visual Effect on Start	Choose a visual effect to play when object starts moving
Stop When	You can choose which trigger will stop the movement. Those can be: - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.
Restart When	You can choose which trigger will restart the movement. Those can be: - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - RotateTemplate

MoveToPlayer

Objects with the MoveToPlayer behavior will leave their designated location and move towards the player. They track the player's movement and follow in the same direction. Multiple triggers can activate this behavior as needed.

To add the Move To Player logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Move To Player under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
MoveToPlayerWhen	You can choose the trigger to activate the behaviour - After a broadcast message has been received by the object - When the player touches the object - When a different object touches the object - When you click on the object
MoveSpeed	You can fix the speed for object movement
Offset	Specifying offset value ensures that the object will move a specific number of units in the given axis. It won't follow you but will relocate to a different location on triggering.
Play SFX	Choose a sound effect to play when you collide with the object.
Play VFX	Choose a sound effect to play when you collide with the object.
Cancel On	You can choose when to stop the movement of the object. - After a broadcast message has been received by the object.

Parameter	Description
	- When the player touches the object. - When a different object touches the object.
Cancel Type	You can choose what happens to the object once it stops moving. - It can relocate back to its original position. - It can stop at the current position.
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the object comes in contact with you.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - MoveToPlayerTemplate

Rotate Oscillate

When the Rotate Oscillate logic template is applied to objects, they begin to oscillate around their axis at a defined angle and speed upon a specific trigger.

To add the Rotate Oscillate logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Rotate Oscillate under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Rotate On	You can choose the trigger on which the object will start rotating. - It can be at the start of the game - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.
Axis	You can choose the axis about which the object will oscillate (X,Y,Z axis)
Speed	You can define the rotation speed of the object
Degrees	define the angle at which the object would oscillate
Direction	You can define the direction of motion either clockwise or anticlockwise
Repeat	You can define the number of time you want the oscillation to occur
Repeat Forever	You can set the oscillation to forever by checking this checkbox.
Sound Effect on Start	Choose a sound effect to play when object starts moving
Visual Effect on Start	Choose a visual effect to play when object starts moving
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the object stops moving.
Stop On	You can choose which trigger will stop the movement. Those can be: - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.
Restart On	You can choose which trigger will restart the movement. Those can be: - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - RotateOscillateTemplate

Basic Instantiate

The Basic Instantiate behavior facilitates the dynamic spawning of identical objects in various locations during gameplay. It allows for controlled timing and quantity of object spawns.

You can determine where objects appear in the game scene by specifying a range of X, Y, and Z coordinates or by randomizing their location within a designated area. With the first method, you have precise control over the specific combinations of coordinates where objects will spawn. Alternatively, using the random allocation option, you can define an area using the record feature, allowing objects to appear randomly within that area during gameplay.

To add the Basic Instantiate logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Basic Instantiate under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Instantiate On	You can choose the start event from the dropdown to activate the behaviour - After a broadcast message has been received by the asset - When the game starts
Repeat On Event	If you check this box, the asset will appear only once in the scene
No of instance	You can set how many assets will respawn at a time
Position	You can choose where you want the assets to respawn using the dropdown. You can choose from: - Locator: Objects will spawn at specific coordinates. You can set the coordinates by clicking the "+" sign below the location positions. - Random in area: Objects will spawn at random coordinates. You can define the area using record feature.
Randomise	This parameter is specifically for locators. It causes objects to spawn at the specified coordinates, but the order of the coordinates will be randomized, disregarding the order in which they were defined.
Play SFX	Choose a sound effect to play every time the asset spawns
Play VFX	Choose a visual effect to play every time the asset spawns
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the assets spawn.

There are currently no available T# wrappers to access this template.

Bump

The Bump logic template 'bumps' and pushes any asset that contacts the asset it is applied to.

To add the Bump logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Bump under the header "Action".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Force	Enter the force to be applied
Play SFX	Choose a sound effect to play when the asset is destroyed.
Play VFX	Choose a visual effect to play when the asset is destroyed.
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the asset is destroyed.
Type	Choose from a dropdown whether the bump needs to be a Reflect or a Deflect

There are currently no available T# Wrappers for this template.

Conditionals

Overview

The table below shows a list of logic template components that can execute conditionals

Logic Template	Description
Switch	Helps activate or deactivate behaviors depending on the triggers associated with each action.
OR Gate	Acts as a gate that sends out a broadcast signal only after any one of the required conditions are met. These conditions are broadcast signals from various sources.
AND Gate	Acts as a gate that sends out a broadcast signal only after all required conditions are met. These conditions are broadcast signals from various sources. It won't activate until every condition is satisfied.
Tick	Generates a broadcast at a pre-defined time or time-intervals

List of all Conditional Logic Template Components

Switch

The Switch logic template enables you to regulate the behaviors of an asset, whether it's the one you're interacting with or a different one. You can transmit broadcasts to activate or deactivate behaviors depending on the triggers associated with each action.

For instance, suppose a cube possesses a switch logic template. In that case, when the player collides with the cube, an 'on-broadcast' signal is dispatched to the laser, prompting its rotation. Conversely, clicking on the cube sends an 'off-broadcast' signal to the laser, ceasing its rotation.

To add the Switch logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Switch under the header "Conditionals".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Switch On	You can choose the trigger that will turn "on" the switch: - When a different object touches the object - After a broadcast message has been received by the object -When the player exits. - When the player collides with the object. - When the object is clicked
Sound Effect When On	Choose a sound effect to play when the switch is turned "on"
Visual Effect When On	Choose a visual effect to play when the switch is turned "on"
Broadcast After On	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when switch is turned "on"
Switch Off	You can choose the trigger that will turn "off" the switch: - When a different object touches the object - After a broadcast message has been received by the object -When the player exits. - When the player collides with the object - When the object is clicked
Sound Effect When Off	Choose a sound effect to play when the switch is turned "off"
Visual Effect When Off	Choose a visual effect to play when the switch is turned "off"
Broadcast After Off	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when switch is turned "off"

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - SwitchTemplate

OR Gate

The Or Operator is used when you want an event to occur when either one of the defined events has taken place. When broadcasts are received from any event indicated in the Or Operator behavior, the behaviour will perform a particular action.

Example: When the player completes either of the two checkpoints and the operator receives a broadcast message from either of them, the next event scheduled by the behavour is triggered.

To add the OR Gate logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select OR Gate under the header "Conditionals".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
Wait For (Listen for)	Choose the broadcasts which are prerequisite for the operator to do further tasks. You can choose broadcasts from the selection menu.
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the behaviour has received all broadcasts.

You can directly use conditionals in T# and do not need a wrapper to access this logic template

AND Gate

The **AND** operator is used when you want an event to occur only after a specific number of other events have taken place. When broadcasts are received from all the events specified in the **AND** operator behavior, another broadcast is triggered.

To add the AND Gate logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select AND Gate under the header "Conditionals".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
Wait For (Listen for)	Choose the broadcasts which are prerequisite for the operator to do further tasks. You can choose broadcasts from the selection menu.
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the behaviour has received all broadcasts.

You can directly use conditionals in T# and do not need a wrapper to access this logic template

Tick

Applying the Tick Logic template to an asset allows you to customize a timer. Unlike the default timer, which you can only start or stop, this template lets you control or modify the start and stop functions. You can also pause the timer or send specific broadcasts at determined intervals. Additionally, this logic template can act as a starting event for other logic templates.

To add the Tick logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Tick under the header "Conditionals".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:\

Parameters	Description
Tick When	You can choose the start event for the object's timer from this dropdown: • Game Start • Broadcast Received: When the object receives a broadcast message.
Stop When	You can choose which broadcast will stop the execution when listened to.
Resume When	You can choose which broadcast will resume the execution when listened to.
Special Broadcasts	You can specify time intervals for the template to run and generate a broadcast. The available fields are: • **When**: Select "At" or "Every". • **In**: Specify the time interval for the broadcast. • **Broadcast**: Define the broadcast to be generated.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - TickTemplate

Triggers

Overview

Triggers are logic templates that initiate various game interactions. The table below describes the three main trigger logic templates:

Logic Template	Description
Collide	Uses contact of collider of the player as a trigger and allows you to generate a broadcast
Click	Uses mouse click as a trigger and allows you to generate broadcast
Delay	Introduces a delay of a specified time

List of Trigger Logic Template Components

Collide

Any object with the Collide logic template attached sends a broadcast either when you collide with it or when a different object touches it.

To add the Collide logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Collide under the header "Triggers".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
Start On	Choose how to collide. You have OnPlayerCollide or OtherObjectTouches
Play SFX	Choose a sound effect to play when you collide with the object.
Play VFX	Choose a visual effect to play when you collide with the object.
BroadcastData	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when you collide with the object.

Click

The Click logic template enables you to send a broadcast when you click on an asset.

To add the Click logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Click under the header "Triggers".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameter	Description
Play SFX	Choose a sound effect to play when you click on the asset
Play VFX	Choose a visual effect to play when you click on the asset
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when you click on the asset.

Delay

The Delay logic template allows you to add a few seconds of delay in between the broadcasts which eventually creates a gap of a few seconds between two different events. Usually, events happen instantly as soon as they receive the broadcast but with the help of a delay behavior, you can add a delay between two events.

To add the Delay logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Delay under the header "Triggers".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor interface:

Parameters	Description
Listen To	Choose a broadcast from the drop drop menu. Once the object receives this broadcast, there will be a delay created for the next event.
Delay Time	Define the seconds by how long there will be a delay
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent after the defined delay time has completed.

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - DelayBroadcastTemplate

Effects

Overview

Logic Template	Description
Stop Rotate	Stops Rotation
ShowUI	Displays a UI on the screen
Stop Animation	Stops Animation
Play Player's Animation	Plays animation of the player

List of Effects Logic Components

Stop Rotate

Stop Rotate allows you to stop the rotation of any object by attaching the behaviour to it and defining any trigger to initiate the event.

Parameters	Type
Start Event	other object touches, mouse clicked, broadcast listened
Effects	Change in an Asset's Orientation, Generate a Broadcast Signal, Enable an SFX or Particle
Type	dependent

To add the Stop Rotate behavior to an asset, follow these steps:

1. Select the asset you wish to apply the Stop Rotate behavior to.
2. In the Inspector panel, click on **Add Behavior**.
3. From the list of behaviors, choose **Stop Rotate.**

You can customize the below-mentioned parameters according to your requirements:

Parameters	Description
StopWhen	

Parameters	Description
	You can choose the trigger on which the object will stop rotating. These are: - After a broadcast message has been received by the object. - When a different object touches the object - When you click on the object.
Broadcast Data	Define a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when object stops rotating
Play SFX	Choose a sound effect to play when the object stops rotating.
Play VFX	Choose a visual effect to play when the object stops rotating.

There are currently no available T# Wrappers for this template but you can write your own custom T# code from scratch.

Stop Animation

The Stop Animation logic template is used for animated objects. When executed, it halts the current animation of the object.

To add the Stop Animation logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Stop Animation under the header "Effects".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
Stop When	Choose from the following options in the dropdown menu for when the animation should be stopped: 1. On Player Touch 2. On Other Object Touch 3. On Click 4. On Broadcast: If selected, specify the broadcast to listen to.
Broadcast	Enter a broadcast to be generated at the end of execution. This broadcast can be used as a trigger for other behaviors.

There are currently no available T# Wrappers for this template.

Play Player's Animation

The Play Player's animation logic template activates a predefined player animation when the trigger condition is met.

To add the Play Player's Animation logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Play Player's Animation under the header "Effects".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
Play On	Choose from the following options in the dropdown menu for when the Player animation should Start: 1. On Game Start 2. On Player Touch 3. On Other Object Touch 4. On Click 5. On Broadcast: If selected, specify the broadcast to listen to.

Parameters	Description
Broadcast	Enter a broadcast to be generated at the end of execution. This broadcast can be used as a trigger for other behaviors.
Animation	Select an animation for the player to execute from the dropdown list
Reset Automatically	Toggle button that specifies whether the player animation must reset

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - PlayPlayerAnimationTemplate. You may also find the wrapper - PlayerAnimationControlTemplate useful

ShowUI

The ShowUI behavior allows the creator to display UI elements on the screen in response to player interactions with game objects.

To add the Show UI logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Show UI Animation under the header "Effects".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
Show On	Choose from the following options in the dropdown menu for when the UI Needs to be displayed: 1. On Game Start 2. On Player Touch 3. On Other Object Touch 4. On Click 5. On Broadcast: If selected, specify the broadcast to listen to.
Animation	CHoose the animation for the UI element
Screen Position	Choose from the dropdown the position of the UI element
UI Template	Select from a list of pre-available UI Templates
Animation Duration	Specify the duration of Animation in seconds
Show for	Specify the duration for which the UI must be displayed
Broadcast Data	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the UI is displayed

Terra Studio has pre-defined UI Templates as shown in `ShowUI Prefabs`.

ShowUI offers a wide selection of pre-defined UI layouts that can be used to display a UI in your game. You can choose any UI from the dropdown menu under "UI To Show." The name and layout of each UI are provided in the "Available UI Prefabs" table below. Each layout contains two types of elements:

- **Editable Text**: You can include up to three text elements—Text 1, Text 2, and Text 3. You can directly input the desired text in these fields.
- **Icons**: You can use up to two icons—Icon 1 and Icon 2. To access or modify an icon, simply input the icon's name from the list of available Show UI Icons

PlayerStats

Overview

Logic Template	Description
Update Score	Updates a specific score group to a new specified value
Reset Score	Resets the specified score group to zero
Increase Player HP	Increases the player Health value by the specific amount
Decrease Player HP	Decreases the player Health value by a specific amount
Reset Player Health	Resets the player Health value to zero
Level Up	Guides the Level Mapper on how to increase a property's level to the next tier.
Update Magnet	Changes the magnet range for the player's collection
Stop Player Movement	Stops or starts the player movements
Change Player Speed	Changes the speed of movement of the player

List of PlayerStats Logic Template Components

Update Score

You may want to enhance the gameplay experience by adjusting the game score when specific conditions are met. For instance:

- Update the score when the player collects items such as coins, gems, or power-ups. This promotes exploration and rewards the player's curiosity.
- Increase the score upon completing a level or reaching a milestone, such as finishing a race track or solving a puzzle.
- Award points for unique or challenging achievements outside the normal gameplay loop, such as discovering hidden areas.

You do this using the Update Score logic template component. To add the Update Score logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Update Score under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameter	Description
Update When	Allows you to define the trigger for updating the score. There are four triggers allowed: Player Touches- Updates the score when the player touches the selected object Other Object Touches - Updates the score when another object touches the selected object Clicked - Updates the score when you click the selected object Broadcast Listened - Updates the score when it listens to a broadcast from another object
Score Group	Define which score group needs to be updated based on the trigger event happening
Operator	Define the operator to multiply by. Four operators are allowed - Add, Subtract, Multiply and Divide
Update By	Define an integer (positive or negative) to change the score by. For instance, if the operator is Add and Update By is given the value 10, then the score is increased by 10 every time the trigger event defined in Update When happens
Broadcast	Define any broadcast for another object to listen to

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - UpdateScoreTemplate

Reset Score

You can reset a player's score in Terra Creator Studio:

To reset the player score using logic templates, you need to add the Reset Score logic template using these steps:

1. Go to the Logic Tab.
2. Select Reset Score under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameter	Description
Update When	Allows you to define the trigger for resetting the score. There are four triggers allowed: Player Touches- Resets the score when the player touches the selected object Other Object Touches - Resets the score when another object touches the selected object Clicked - Resets the score when you click the selected object Broadcast Listened - Resets the score when it listens to a broadcast from another object
Score Group	Define which score group needs to be reset based on the trigger event happening
Broadcast	Define any broadcast for another object to listen to

You can choose not to use any logic template and directly use the T# wrapper for resetting score - the ResetScoreTemplate

Increase Player HP

When triggered, the Increase HP (HP stands for Health Points) logic template boosts the player's health. The behavior can be activated in response to various events, such as the player collecting coins or other items that improve health.

To add the Increase HP logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Increase Player HP under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
When	You can choose the trigger to activate the behaviour - After a broadcast message has been received by the object - When the player touches the object
By Point	The player's health will increase by the defined amount when the behaviour is triggered.
Play SFX	Choose a sound effect to play when the health increases.
Play VFX	Choose a visual effect to play when the health increases.
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent everytime the player health increases.

There are currently no available T# Wrappers for this template.

Decrease Player HP

In games, lowering the player's health as a result of certain interactions is a common mechanic. The Decrease Player HP Logic template reduces the player's health when specific triggers are activated.

To add the Decrease Player HP logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Decrease Player HP under the header "PlayerStats".

3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
When	You can choose the trigger to activate the behaviour. - After a broadcast message has been received by the object. - When the player touches the object.
By Point	The player's health will decrease by the defined amount when the behaviour is triggered.
Play SFX	Choose a sound effect to play when the health decreases.
Play VFX	Choose a visual effect to play when the health decreases.
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent everytime the player health decreases.

There are currently no available T# Wrappers for this template.

Reset Player Health

When a player interacts with an object tagged with the Reset Player Health logic template, the player's health is fully restored. This can be useful when a player respawns after dying, enters a safe zone, collects healing items, or faces similar situations.

To add the Reset Player Health logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Reset Player Health under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
When	You can choose the trigger to activate the behaviour. - After a broadcast message has been received by the object. - When the player touches the object. - When the object is clicked.
Play SFX	Choose a sound effect to play when the health is reset
Play VFX	Choose a visual effect to play when the health is reset
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when health is restored.

There are currently no available T# Wrappers for this template.

Update Magnet

The Update Magnet behavior increases the player's magnetic range, allowing them to collect items from a larger radius compared to the initial specified range.

To add the Update Magnet logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Update Magnet under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameter	Description
Change Magnet When	

Parameter	Description
	Define the action that will trigger a change in the magnetic field. The actions could be: - After a broadcast message received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.
Radius	The amount defined will act as the updated radius
Play SFX	Choose a sound effect to play when the magnet range is updated
Play VFX	Choose a visual effect to play when the magnet range is updated
Broadcast Data	Define a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the magnetic range is updated

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - ChangeMagnetTemplate

Change Player Speed

Applying the Change Player Speed template changes the speed of the player to the desired value.

To add the Change Player Speed logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Change Player Speed under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameter	Description
Change On When	Define the trigger for changing the speed. The four available triggers are: • **Player Touches** • **Other Object Touches** • **Clicked** • **Broadcast Listened**
Modifier	Define the modifier: 1. Set - Set to a defined value 2. Multiply - Multiply by the given value 3. Add - Increase by the given value 4. Subtract - Subtract the given value 5. Divide - Divide by the given value
Speed	Specify the value to apply the modifier to the current speed
SFX / VFX	Specify the SFX and VFX to be played upon execution
Broadcast	Define any broadcast for another object to listen to

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - ChangePlayerSpeedTemplate

Level Up

Level up behaviour is used to upgrade different in-game assets based on the the upgrade paths of objects defined using the Level Mapper game system.

To add the Level Mapper logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Level Mapper under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameter	Description
Level up when	You can choose the trigger to activate the behaviour - After a broadcast message has been received by the object - When the player touches the object - When a different object touches the object - When you click on the object
Sound Effect on Start	Choose a sound effect to play on execution
Visual Effect on Start	Choose a visual effect to play on execution
Manager group	Helps to choose the score group based on which the Level up will be decided
Broadcast Success	Define a broadcast to be generated when the level up is successful
Broadcast Fails	Define the broadcast to be generated when the level up fails
Execute Times	Number of times behaviour need to be excuted

There are currently no available T# Wrappers for this template.

Stop Player Template

When a player comes into contact with an asset that has the StopPlayerMovement logic template applied, the player's motion in the game environment is halted. This logic template locks the player in their current position and deactivates any movement through the controller.

To add the Stop Player Movement logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Stop Player Movement under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameters	Description
Start When	You can choose the trigger to activate the behaviour - When the game starts - After a broadcast message has been received by the object. - When the player touches the object. - When a different object touches the object. - When you click on the object.
Play SFX	Choose a sound effect to play when the player stops moving.
Play VFX	Choose a visual effect to play when the player stops moving.
Broadcast	Choose to enter a broadcast that can be used as a trigger for any other behavior. The broadcast is sent when the player stops moving again.

There are currently no available T# Wrappers for this template.

Change Player Speed

Applying the Change Player Speed template changes the speed of the player to the desired value.

To add the Change Player Speed logic template to an asset, follow these steps:

1. Go to the Logic Tab.
2. Select Change Player Speed under the header "PlayerStats".
3. Drag and drop it onto the desired asset.

You can edit the following parameters of this template directly through the scene editor:

Parameter	Description
Change On When	Define the trigger for changing the speed. The four available triggers are: • **Player Touches** • **Other Object Touches** • **Clicked** • **Broadcast Listened**
Modifier	Define the modifier: 1. Set - Set to a defined value 2. Multiply - Multiply by the given value 3. Add - Increase by the given value 4. Subtract - Subtract the given value 5. Divide - Divide by the given value
Speed	Specify the value to apply the modifier to the current speed
SFX / VFX	Specify the SFX and VFX to be played upon execution
Broadcast	Define any broadcast for another object to listen to

If you want to further customize this logic template, you can do so by accessing its T# Wrapper - ChangePlayerSpeedTemplate