Eadon Survival for Invector TPC

Version 2.1

Table of Contents

Introduction

Eadon Survival is an add-on for Invector's Third Person Controller asset for the Unity game engine. The goal of this asset is to provide a framework for implementing survival games in Unity.
The asset does not make any assumptions on the type of survival games, allowing for a wide range of game styles.
The following functionalities are implemented:

- Hunger
- Thirst
- Temperature handling (via Enviro integration)
- Night and day management (via Enviro integration) with different hunger/thirst profiles
- Sicknesses
- Specific cures for different sicknesses
- Injuries (from criticall attacks or being damaged when below a threshold)
- Custom debilitating effects for sicknesses
- Custom debilitating effects for injuries, including preventing equipping items on broken arms
- Bonfire system

This add-on is currently compatible with Invector TPC version 2.5.7 and higher (Melee and Shooter) and tested on Unity 2019.4 and higher.

Changelog

V 1.0	Initial release
V 1.1	Added support for external protection providers (required by Eadon RPG for Invector integration) Added vItemAttributes for protections
V 1.2	Bonfire system
V 1.3	Bug fixes and improved installation process
V 1.4	Enviro 3 support
V 1.5	Bonfire custom animations (light, extinguish, add fuel) Resting bonfire custom animations (start rest, rest loop, end rest)
V 1.6	Update to Invector 2.6.4
V2.0	New project structure Support for networking (Mirror, Fish-Net and Unity Networking for Game Objects, separate assets) Bug fixes
V2.1	Added events on lit, extinguished and fuel added to bonfires Bug fixes

Credits

Some models used for the bonfire and the stick in the demo scene are free models licensed under Creative Common lAttribution license. Please respect the copyright if you use them in your game and display proper attribution:

Bonfire	"Bonfire-Lowpoly" (https://skfb.ly/6yvnq) by Christian Gentry is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).
Stick	"Wooden Stick" (https://skfb.ly/6RsL6) by Thunder is licensed under Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

Prerequisites

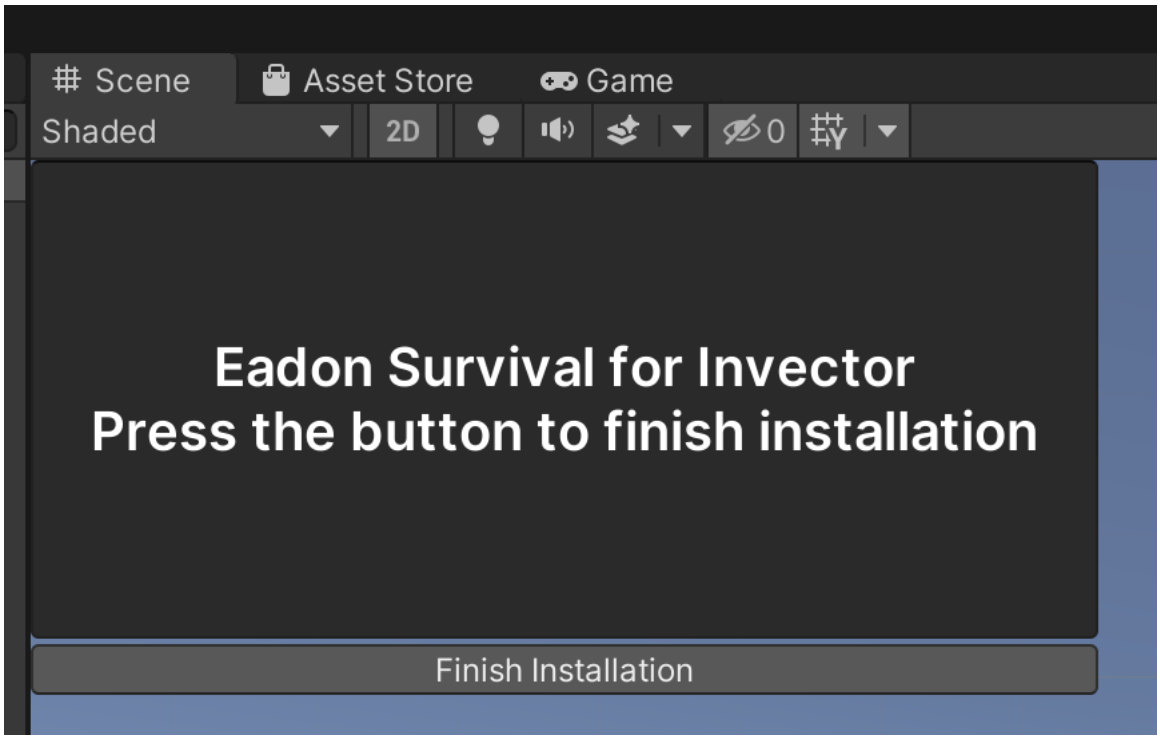
Eadon Survival contains support for the following package:

- Hendrik Haupt's Enviro - Sky and Weather (https://assetstore.unity.com/packages/tools/particles-effects/enviro-sky-and-weather-33963)

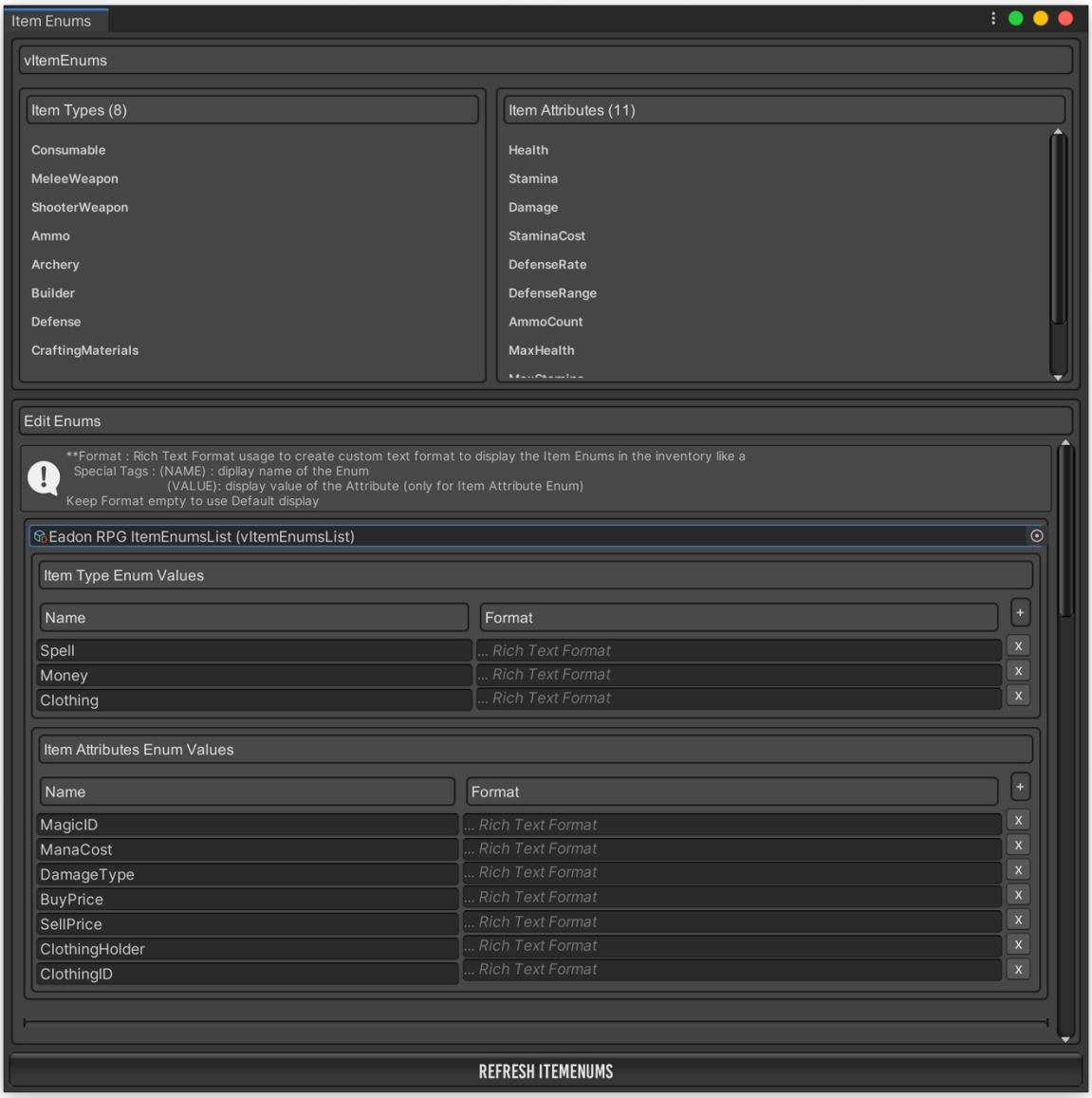
This asset is used for temperature effect, night and day management and managing effects that happen every game hour or game day.

The use of Enviro is completely optional and support is enabled automatically if Enviro is installed I your project.

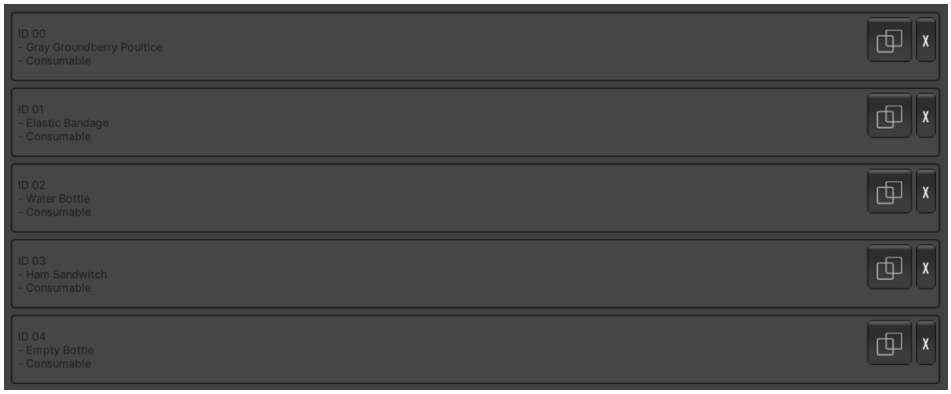
Starting with version 2.1, when you import the project you will see this in the scene view:



Press the button to complete the installation, which will automatically rebuild your **vItemEnums** and set a scripting define.
If you copy the scripts manually to another project, you might encounter several errors which will appear as soon as the add-on is imported because the code relies on the presence of three vItemType and seven vItemAttributes. In order to fix this error, you need to go to Invector/Inventory/Item Enums/Open ItemEnums Editor and click on "REFRESH ITEMENUMS" at the bottom and add the EADON_SURVIVAL_INVECTOR to the scripting defines in the Player section of the Project Settings.



In order to run the demo scene correctly, please make sure the following items in the Eadon Survival Demo vItemList have the correct item type as in the picture below:



Check also all the items in the demo vItemList as some might have ended up with different item types.

All this is due to the fact that if you have any add-ons or different versions of Invector, the order of item types might change.

Configuration

Before Eadon Survival can be used, it needs to be configured. You need to create all the sicknesses, cures, injuries and remedies you plan to use. This is an optional step, in the sense that if you only want to use hunger and thirst (and temperature if using Enviro) you can skip this step.

Eadon Survival Data

You need one instance of this ScriptableObject in your project. Its role is to collect all the elements available in your game. To create one, right click in your project window and select Create/Eadon Survival/New Survival Manager Data. A new Eadon Survival Manager Data object will be created in that location. It looks like this:

The fields are as follows:

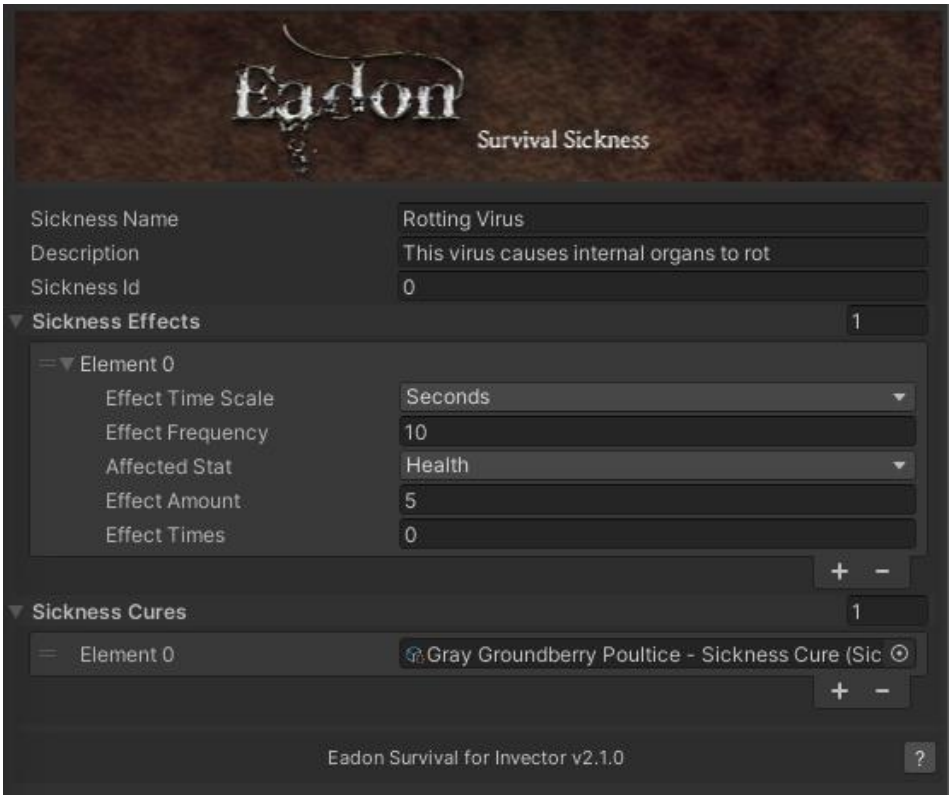
Field	Use
Hunger Recovery Attribute	The vItemAttribute name for the hunger recovery
Hunger Grace Attribute	The vItemAttribute name for the hunger grace period
Thirst Recovery Attribute	The vItemAttribute name for the thirst recovery
Thirst Grace Attribute	The vItemAttribute name for the thirst grace period
All Sicknesses	A list of all the Sickness ScriptableObjects
All Sickness Cures	A list of all the SicknessCure ScriptableObjects
All Injuries	A list of all the Injury ScriptableObjects
All Injury Remedies	A list of all the InjuryRemedy ScriptableObjects

For an explanation of the vItemAttributes, see the chapter on Hunger and Thirst.

When you create a new element, you need to add it to the relevant section.

Sickness

A Sickness ScriptableObject contains the definition of a sickness or infection available in game. To create one, right click in your project window and select **Create/Eadon Survival/New Sickness**. A new Eadon Survival Sickness object will be created in that location. It looks like this:



The fields are as follows:

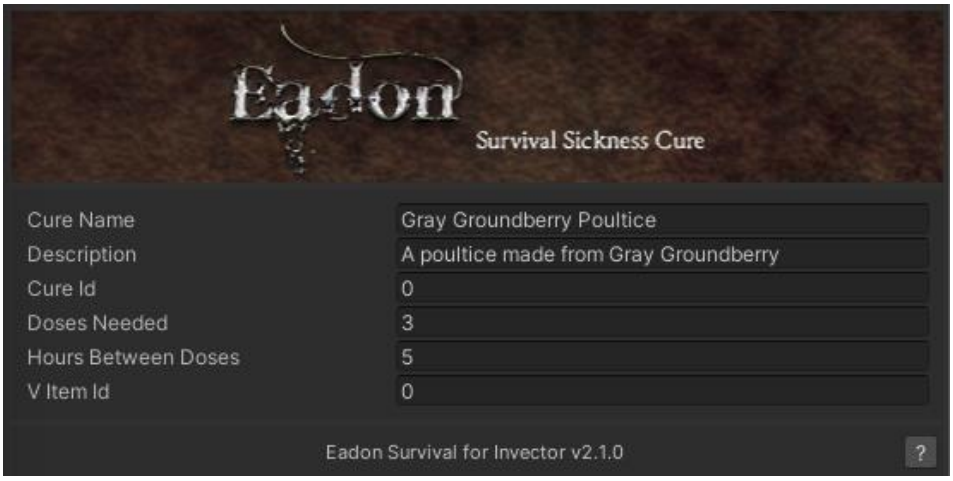
Field	Use
Sickness Name	The name of the sickness
Description	The description of the sickness
Sickness ID	The id of the sickness (needs to be unique across all sicknesses)
Sickness Effects	A list of all the effects caused by this sickness (see below for a description of the fields)
Sickness Cures	A list of all the cures that need to be applied to get rid of the sickness

The sickness effects fields are as follows:

Field	Use
Effect Time Scale	The time scale of the effect, choice of seconds and minutes (real time). If Enviro is installed in the project, game hour and game day are also available
Effect Frequency	How often (based on the time scale) the effect takes place
Affected Stat	Which stat is affected, choice of Health, Stamina, Hunger or Thirst
Effect Amount	The amount of the affected stat lost every time the effect takes place
Effect Times	How many times the effect takes place (if you leave it at 0 or less, it will happen until cured)

Sickness Cure

A SicknessCure ScriptableObject contains the definition of a sickness or infection cure available in game. To create one, right click in your project window and select **Create/Eadon Survival/New Sickness**. A new Eadon Survival Sickness object will be created in that location. It looks like this:



The fields are as follows:

Field	Use
Cure Name	The name of the cure
Description	The description of the cure
Cure ID	The id of the cure (needs to be unique across all cures)
Doses Needed	The number of doses needed
Hours Between Doses	The time interval between doses
vItem ID	The id of the vItem for this cure

Injury

A Injury ScriptableObject contains the definition of an injury available in game. To create one, right click in your project window and select **Create/Eadon Survival/New Sickness**. A new Eadon Survival Sickness object will be created in that location. It looks like this:

The fields are as follows:

Field	Use
Injury Name	The name of the injury
Description	The description of the injury
Injury ID	The id of the injury (needs to be unique across all injuries)
Speed Reduction	How much the speed of the character is reduced while injured (range of 0 to 1, 0 means no reduction, 1 means total reduction)
Health Reduction	How much the health of the character is reduced while injured (range of 0 to 1, 0 means no reduction, 1 means total reduction)
Stamina Reduction	How much the stamina of the character is reduced while injured (range of 0 to 1, 0 means no reduction, 1 means total reduction)
Damage Output Reduction	How much the damage output of the character is reduced while injured (range of 0 to 1, 0 means no reduction, 1 means total reduction)
Prevents Right Hand Items	A flag to prevent equipping items on the right hand

Prevents Left Hand Items	A flag to prevent equipping items on the left hand
Injury Remedies	A list of all the remedies for this injury

Injury Remedy

A InjuryRemedy ScriptableObject contains the definition of an injury remedy available in game. To create one, right click in your project window and select **Create/Eadon Survival/New Sickness**. A new Eadon Survival Sickness object will be created in that location. It looks like this:

The fields are as follows:

Field	Use
Injury Remedy Name	The name of the injury remedy
Description	The description of the sickness
Injury Remedy ID	The id of the injury remedy (needs to be unique across all injury remedies)
vItem ID	The id of the vItem for this cure

Character Creation

In order to enable the survival system on your character, you need to have an already created and configured Invector character. Once your character is in the scene, select it and click on **Invector/Eadon Survival/Convert to Eadon Survival** from the menu bar. This will perform the following changes on your character:

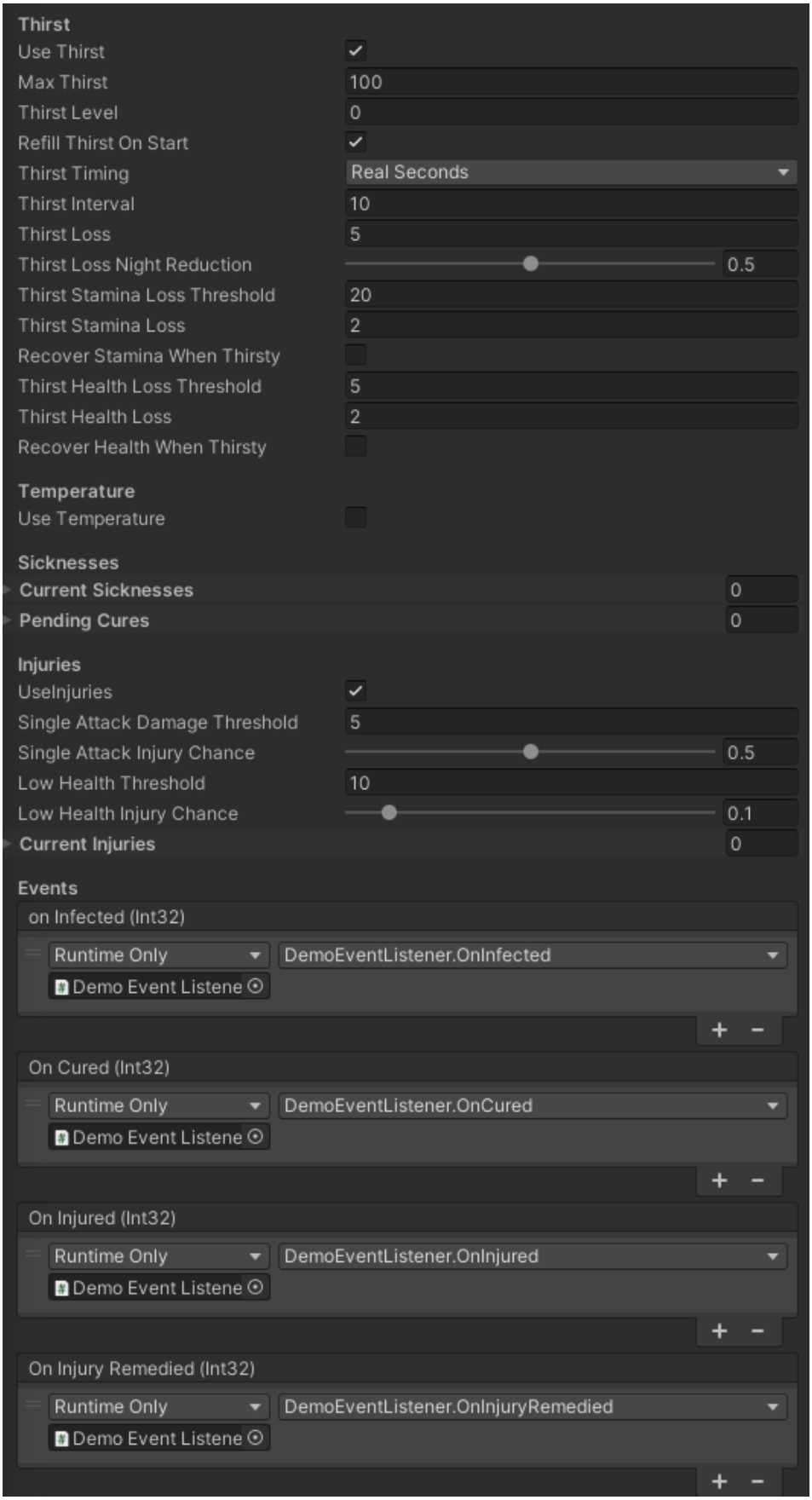
- If the character already has a vItemManager, it will be converted to EadonSurvivalItemManager
- If the character does NOT have a vItemManager, a new EadonSurvivalItemManager will be added
- A EadonSurvivalManager component will be added to the character

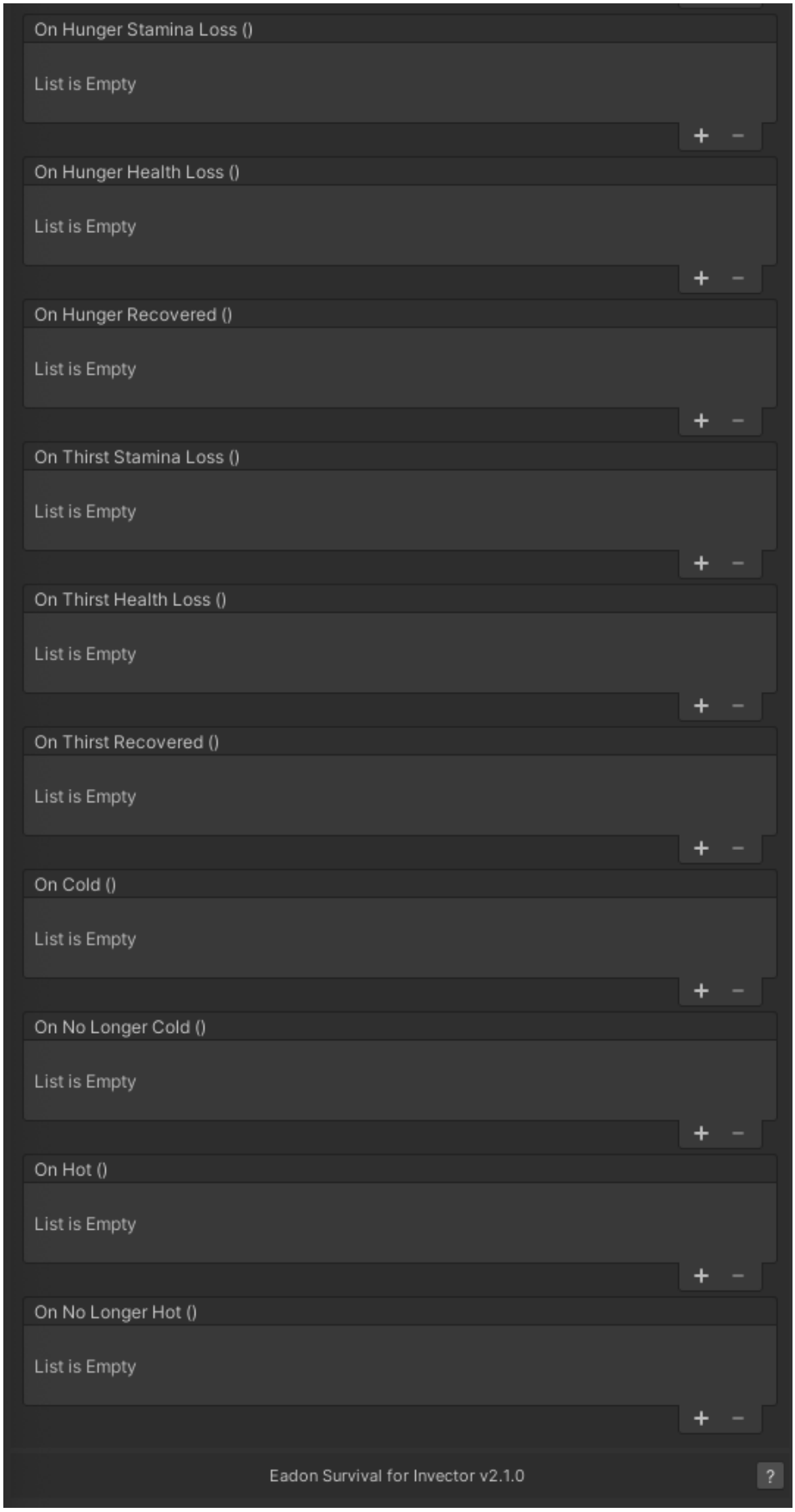
Eadon Survival Item Manager

This component looks and behaves exactly like the default vItemManager from Invector, but with some overridden methods needed to support the survival system.

Eadon Survival Manager

This is the main component for the survival system. It defines the behaviour of the system for the character and tracks what happens to the character.
This is character specific, because you could have different characters in your game with different characteristics, such as an alien from an icy world being adapted to cold temperature. It looks like this:


Thirst
Use Thirst
Max Thirst
100
Thirst Level
0
Refill Thirst On Start
Thirst Timing
Real Seconds
Thirst Interval
10
Thirst Loss
5
Thirst Loss Night Reduction
0.5
Thirst Stamina Loss Threshold
20
Thirst Stamina Loss
2
Recover Stamina When Thirsty
Thirst Health Loss Threshold
5
Thirst Health Loss
2
Recover Health When Thirsty
Temperature
Use Temperature
Sicknesses
Current Sicknesses
0
Pending Cures
0
Injuries
UseInjuries
Single Attack Damage Threshold
5
Single Attack Injury Chance
0.5
Low Health Threshold
10
Low Health Injury Chance
0.1
Current Injuries
0
Events
on Infected (Int32)
Runtime Only
DemoEventListener.OnInfected
Demo Event Listene
On Cured (Int32)
Runtime Only
DemoEventListener.OnCured
Demo Event Listene
On Injured (Int32)
Runtime Only
DemoEventListener.OnInjured
Demo Event Listene
On Injury Remedied (Int32)
Runtime Only
DemoEventListener.OnInjuryRemedied
Demo Event Listene


On Hunger Stamina Loss ()
List is Empty
On Hunger Health Loss ()
List is Empty
On Hunger Recovered ()
List is Empty
On Thirst Stamina Loss ()
List is Empty
On Thirst Health Loss ()
List is Empty
On Thirst Recovered ()
List is Empty
On Cold ()
List is Empty
On No Longer Cold ()
List is Empty
On Hot ()
List is Empty
On No Longer Hot ()
List is Empty
Eadon Survival for Invector v2.1.0
?

The fields are as follows:

Field	Use
Data	The SurvivalManagerData object with the data for the game
Use Hunger	A flag to indicate if the hunger subsystem is in use
Max Hunger	The max hunger resistance level
Hunger Level	The current hunger resistance level
Refill Hunger On Start	A flag to indicate if the current level is set to the max level on start
Hunger Timing	The time scale of hunger loss (choice of Real Seconds or Enviro Hours, if Enviro is installed)
Hunger Interval	How often, based on the previous time scale, hunger loss happens
Hunger Loss	How much hunger is lost
Hunger Loss Night Reduction	How much is loss reduced at night (if Enviro is installed), 0 means no reduction and 1 is total reduction
Hunger Stamina Loss Threshold	The hunger threshold below which stamina loss happens
Hunger Stamina Loss	How much stamina is lost
Recover Stamina When Hungry	A flag to indicate if stamina recovery happens while below the threshold
Hunger Health Loss Threshold	The hunger threshold below which health loss happens
Hunger Health Loss	How much health is lost
Recover Health When Hungry	A flag to indicate if health recovery happens while below the threshold
Use Thirst	A flag to indicate if the thirst subsystem is in use
Max Thirst	The max thirst resistance level
Hunger Level	The current thirst resistance level
Refill Thirst On Start	A flag to indicate if the current level is set to the max level on start
Thirst Timing	The time scale of thirst loss (choice of Real Seconds or Enviro Hours, if Enviro is installed)
Thirst Interval	How often, based on the previous time scale, thirst loss happens
Thirst Loss	How much thirst is lost
Thirst Loss Night Reduction	How much is loss reduced at night (if Enviro is installed), 0 means no reduction and 1 is total reduction
Thirst Stamina Loss Threshold	The thirst threshold below which stamina loss happens

Thirst Stamina Loss	How much stamina is lost
Recover Stamina When Thirsty	A flag to indicate if stamina recovery happens while below the threshold
Hunger Thirst Loss Threshold	The thirst threshold below which health loss happens
Hunger Thirst Loss	How much health is lost
Recover Health When Thirsty	A flag to indicate if health recovery happens while below the threshold
Use Temperature	A flag to indicate if the temperature subsystem is in use (only available if Enviro is installed)
Temperature Check Interval	The interval in seconds between temperature checks
Cold Stamina Threshold	The cold temperature stamina threshold
Stamina Loss Per Interval Cold	How much stamina is lost per interval while below the threshold
Cold Health Threshold	The cold temperature health threshold
Health Loss Per Interval Cold	How much health is lost per interval while below the threshold
Hot Stamina Threshold	The hot temperature stamina threshold
Stamina Loss Per Interval Hot	How much stamina is lost per interval while above the threshold
Hot Health Threshold	The hot temperature health threshold
Health Loss Per Interval Hot	How much health is lost per interval while above the threshold
Current Sicknesses	The list of sicknesses currently affecting the character
Pending Cures	The list of cures the character is taking
Use Injuries	A flag to indicate if the injuries subsystem is in use
Single Attack Damage Threshold	The damage threshold above which a single attack can cause an injury
Single Attack Injury Chance	The chance of a single attack causing an injury with high damage
Low Health Threshold	The health threshold below which an attack can cause an injury
Low Health Injury Chance	The chance of an attack causing an injury when at low health
Current Injuries	The list of injuries the character is currently affected by
On Infected	An event triggered when the character is infected
On Cured	An event triggered when the player has cured an infection or sickness
On Injured	An event triggered when the character has sustained an injury

On Injury Remedied	An event triggered when the player has remedied an injury
On Hunger Stamina Loss	An event triggered when the player loses stamina due to hunger
On Hunger Health Loss	An event triggered when the player loses health due to hunger
On Hunger Recovered	An event triggered when the player recovers hunger
On Thirst Stamina Loss	An event triggered when the player loses stamina due to thirst
On Thirst Health Loss	An event triggered when the player loses health due to thirst
On Thirst Recovered	An event triggered when the player recovers thirst
On Cold	An event triggered when the player is cold
On No Longer Cold	An event triggered when the player is no longer cold
On Hot	An event triggered when the player is hot
On No Longer Hot	An event triggered when the player is no longer hot

For more information, see the following chapters on the various subsystems.

Hunger and Thirst

The hunger and thirst subsystem work in the same way. Both have a stat (hunger or thirst) which depletes over time and needs to be replenished. The two system are autonomous, and the loss interval for both are independent.
In order to replenish the relevant stat, you need to use a consumable item with a predefined vItemAttribute. You need to have the four vItemAttributes in your project for the hunger and thirst systems to work. You can either use the predefined attributes (by refreshing the vItemEnums and injecting the provided ones) or you can define your own. In any case, the SurvivalManagerData object needs to have the names of the vItemAttributes you decide to use. The attributes are:

Attribute	Default name	Role
Hunger Recovery	Hunger	Works like the Health attribute for health potions, the value is the amount of hunger restored
Hunger Grace	HungerGrace	The value is the grace period (in seconds) after consuming the item during which the system is paused
Thirst Recovery	Thirst	Works like the Health attribute for health potions, the value is the amount of thirst restored
Thirst Grace	ThirstGrace	The value is the grace period (in seconds) after consuming the item during which the system is paused

The use of the grace attributes is optional, but these attributes are checked only after restoring the stat, i.e. you cannot have a “grace only” item.

Both subsystems can drain stamina and health, with different threshold.
If Enviro is installed in your project, time of day can be used to reduce hunger and thirst loss at night.

Both subsystems can prevent health or stamina recovery while hungry/thirsty.

Temperature

The temperature subsystem is only available if Enviro is installed, as it relies on the temperature provided by this asset.

The subsystem defines cold and hot thresholds for stamina and health and handles loss if the character is below the cold thresholds or above the hot thresholds. The actual temperature compared to the thresholds is adjusted by a localised temperature change which can be manipulated by a trigger collider on a game object with the TemperatureZoneTrigger component. It looks like this:

And it changes the perceived temperature by that amount while the character is within the volume of the attached trigger collider.

This can be used with a small area for the effect of a campfire or with bigger areas for greater effects.

Sicknesses

A character can get sick or infected in two cases only. Either he's hit by an attack from a NPC which is an infector carrier or he spends time in an infectious zone. In both cases, players can be infected by a disease only once at the same time. Getting infected by a sickness you're already suffering from has no change in your situation.

Infected NPCs

If you attach the EadonInfectionCarrier component to an NPC, every time the character is damaged by an attack there's a chance of getting infected. The component looks like this:

The fields indicate the actual sickness transmitted and the chance of an attack infecting the player.

Infected Areas

You can create a game object with a trigger collider and attach to it the InffectionZoneTrigger component. This component tracks if the player is within the trigger volume. While inside, the component will check every x seconds for the chance to infect the character. It looks like this:

Sickness Cures

In order to get rid of a sickness or infection, the player needs to take the appropriate cure. Every sickness can be cured in multiple ways, but you need at least one cure. The cure determines how many times it must be taken and how often.

Injuries

Injuries are “extra effects” which happen to the player when he’s injured by an attack or by any other effect which deals damage (including fall damage).
The injury subsystem has two ways of dealing injuries to the player:

- By defining a damage threshold, every time the player receives damage equal or greater than that amount there’s a chance he’ll receive an injury
- By defining a low health threshold, whenever the player is below that level of health, every time he gets damage there’s a chance he’ll sustain an injury

If you enable the injuries system, you need to have some injuries in the SurvivalManagerData object. Whenever an injury is sustained (as determined by the two criteria above), a random injury is picked from the list of defined injuries and applied to the player. You can get a specific injury only once at the time.

Managing Sicknesses

Eadon Survival for Invector does not have a predefined UI for telling the player he's sick or injured. There are four events in the EadonSurvivalManager component which are triggered when the player is infected, cured, injured or remedies an injury.
All these events have an int parameter which is the id of the sickness that has infected the player or has been cured or of the injury sustained or remedied.
To display this information, you need a reference to the SurvivalManagerData and some methods to listen to the events.
There is a sample component, DemoEventListener, which writes messages to the console and which you can use as a template.

vItemAttributes

Eadon Survival for Invector defines 5 vItemAttributes which can be attached to vItems of type Consumables:

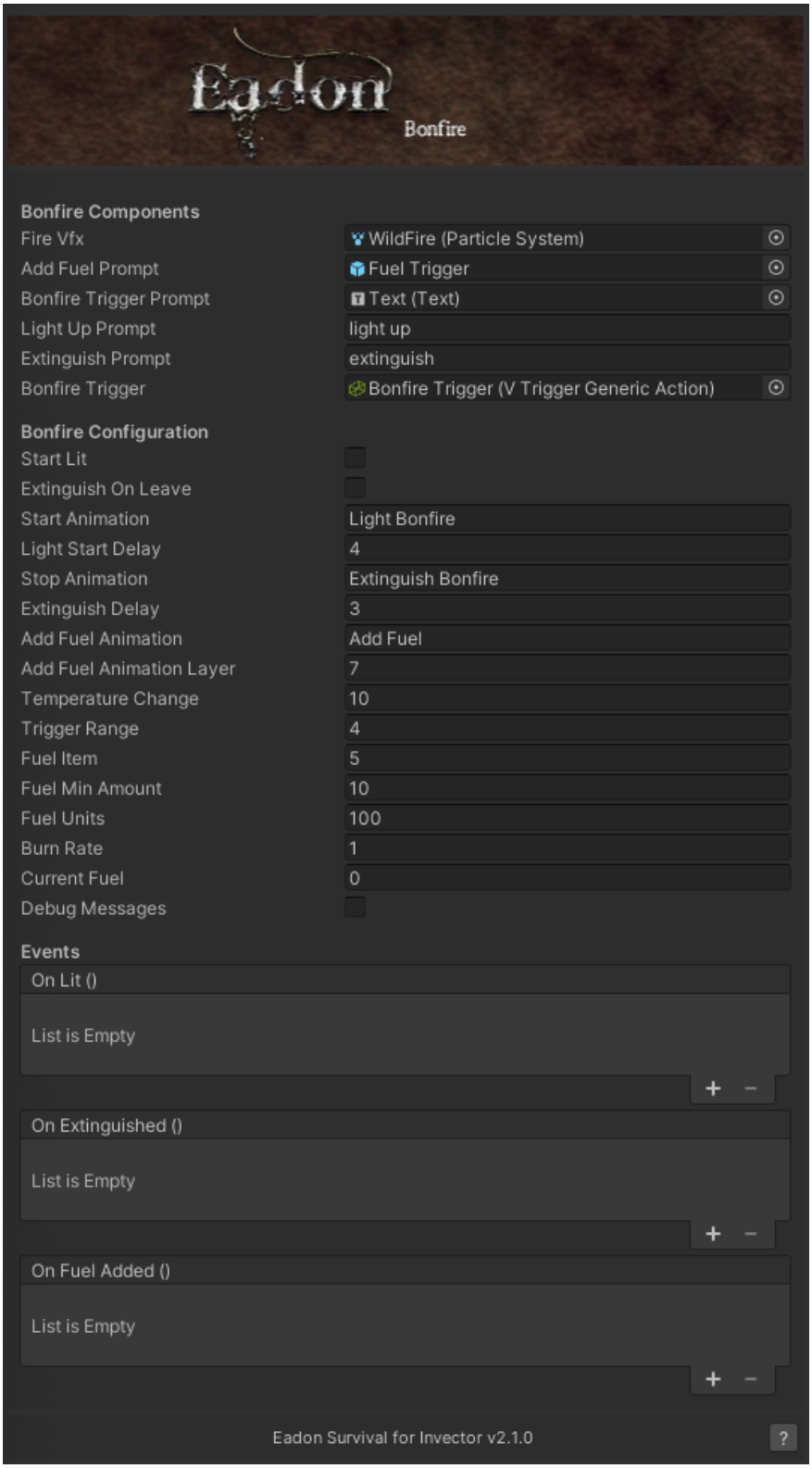
Attribute	Use
Hunger	Amount of hunger recovered when the item is consumed
HungerGrace	Grace period, in seconds, during which hunger checks are paused
Thirst	Amount of thirst recovered when the item is consumed
ThirstGrace	Grace period, in seconds, during which thirst checks are paused
EmptyContainer	The vItem id of an empty container which is added to the inventory when the item is consumed, used to “convert”, for example, a full bottle into an empty bottle so that it can be crafted again into a full bottle

Bonfire system

Eadon Survival for Invector comes with a bonfire system. There are three different types of bonfires: normal bonfires, healing bonfires and resting bonfires. The temperature effects require Enviro (like all other temperature effects).

Normal Bonfires

A normal bonfire is used to raise the temperature in an area around it. It looks like this:



The fields are:

Field	Use
Fire Vfx	The particle system to use for the fire

Add Fuel Prompt	The game object for the add fuel action prompt
Bonfire Trigger Prompt	The text field for the prompt (to change from “light up” to “extinguish” and vice versa
Light Up Prompt	The prompt message for the light up action
Extinguish Prompt	The prompt message for the extinguish action
Bonfire Trigger	A reference to the vTriggerGenericAction for the bonfire
Start Lit	A flag to indicate if the bonfire should start lit on scene load
Extinguish on leave	A flag to indicate if the bonfire should auto extinguish when the player leaves or not
Start Animation	The animation to play when the bonfire is lit (IMPORTANT: it needs to be the same value as the animation set in the bonfire vTriggerGenericAction animation field)
Light Start Delay	The delay from the start of the animation for the bonfire to light up (ignored if not using an animation)
Stop Animation	The animation to play when the bonfire is extinguished
Extinguish Delay	The delay from the start of the animation for the bonfire to extinguish (ignored if not using an animation)
Add Fuel Animation	The animation to play when fuel is added to the bonfire
Add Fuel Animation Layer	The animator layer that contains the add fuel animation
Temperature Change	How many degrees should the temperature raise
Trigger range	The range of the temperature change
Fuel Item	The id of the vItem to use as fuel
Fuel Min Amount	How many units of fuel are needed to replenish the bonfire
Fuel Units	How many units of fuel the bonfire starts with
Burn Rate	How many seconds it takes to burn one unit of fuel
Current Fuel	A counter for the current (runtime) amount of fuel in the bonfire. Values in this field will be ignored
Debug Messages	A flag to indicate if messages should be logged to the console when events happen
On Lit	An event raised when the bonfire is lit

On Extinguished	An event raised when the bonfire is extinguished
On Fuel Added	An event raised when fuel is added

Lighting a bonfire will change the action prompt from the light up prompt to the extinguish prompt, so that you can save fuel for later. When the fuel can be replenished (i.e. when the current fuel is lower than (fuel units – fuel min amount) a second action prompt appears to let you add fuel to the bonfire:

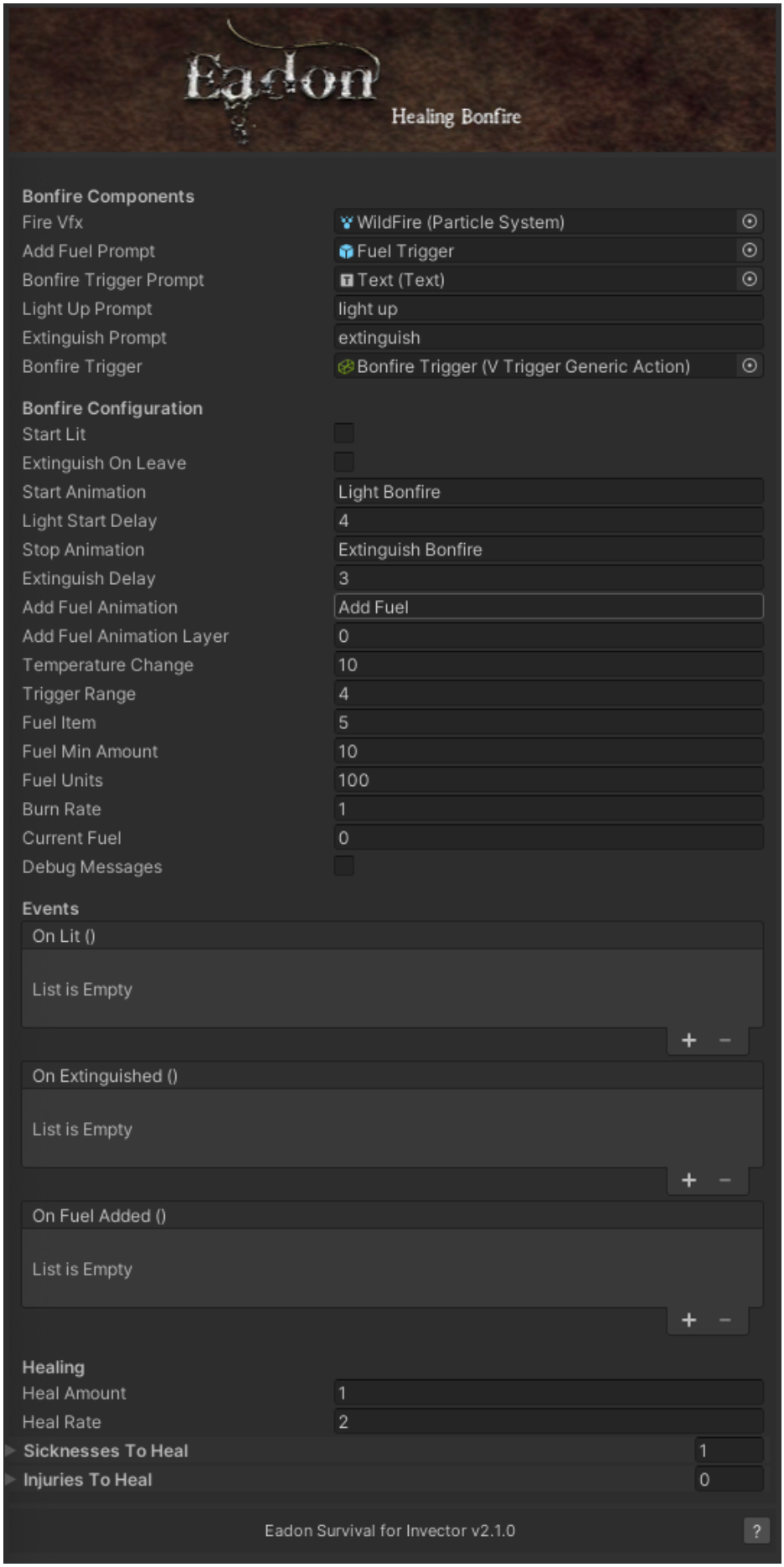
If you want to play animations when your character interacts with the bonfire, follow these steps:

1) Set the light bonfire animator in the bonfire vTriggerGenericAction inspector
2) Put the same value in the Start Animation field of the Bonfire component inspectoe
3) Set the other animations in the other fields of the Bonfire component inspector

All these values are the names of the animation states in the animator. Start and stop use the animation layer set in the vTriggerGenericAction component, for the add fule animation you need to set the layer in the Bonfire component inspector.

Healing Bonfires

Healing bonfires are bonfires that in addition to warming you up can heal you. They look like this:

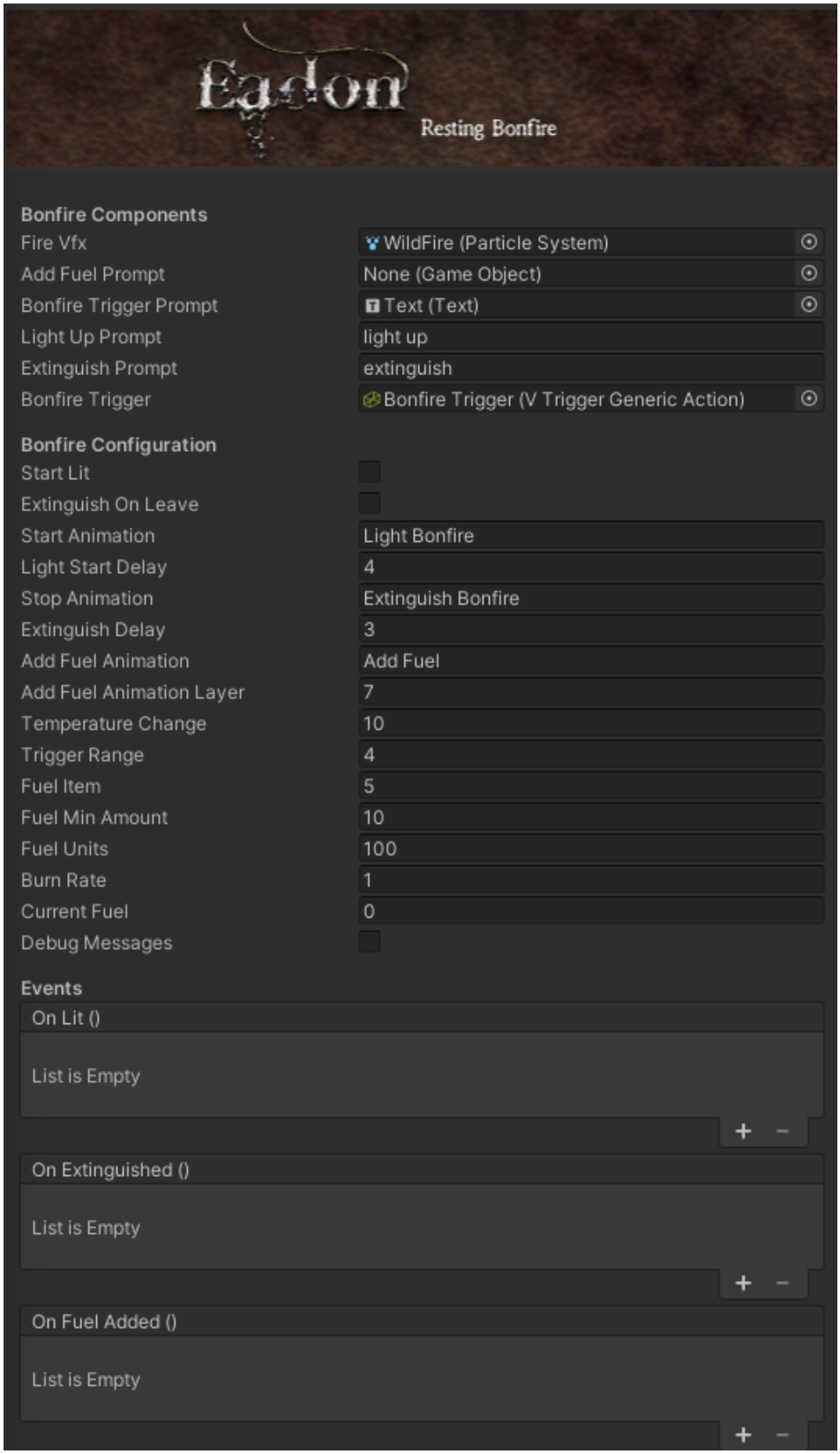

Eadon
Healing Bonfire
Bonfire Components
Fire Vfx
WildFire (Particle System)
Add Fuel Prompt
Fuel Trigger
Bonfire Trigger Prompt
Text (Text)
Light Up Prompt
light up
Extinguish Prompt
extinguish
Bonfire Trigger
Bonfire Trigger (V Trigger Generic Action)
Bonfire Configuration
Start Lit
Extinguish On Leave
Start Animation
Light Bonfire
Light Start Delay
4
Stop Animation
Extinguish Bonfire
Extinguish Delay
3
Add Fuel Animation
Add Fuel
Add Fuel Animation Layer
0
Temperature Change
10
Trigger Range
4
Fuel Item
5
Fuel Min Amount
10
Fuel Units
100
Burn Rate
1
Current Fuel
0
Debug Messages
Events
On Lit ()
List is Empty
On Extinguished ()
List is Empty
On Fuel Added ()
List is Empty
Healing
Heal Amount
1
Heal Rate
2
Sicknesses To Heal
1
Injuries To Heal
0
Eadon Survival for Invector v2.1.0
?

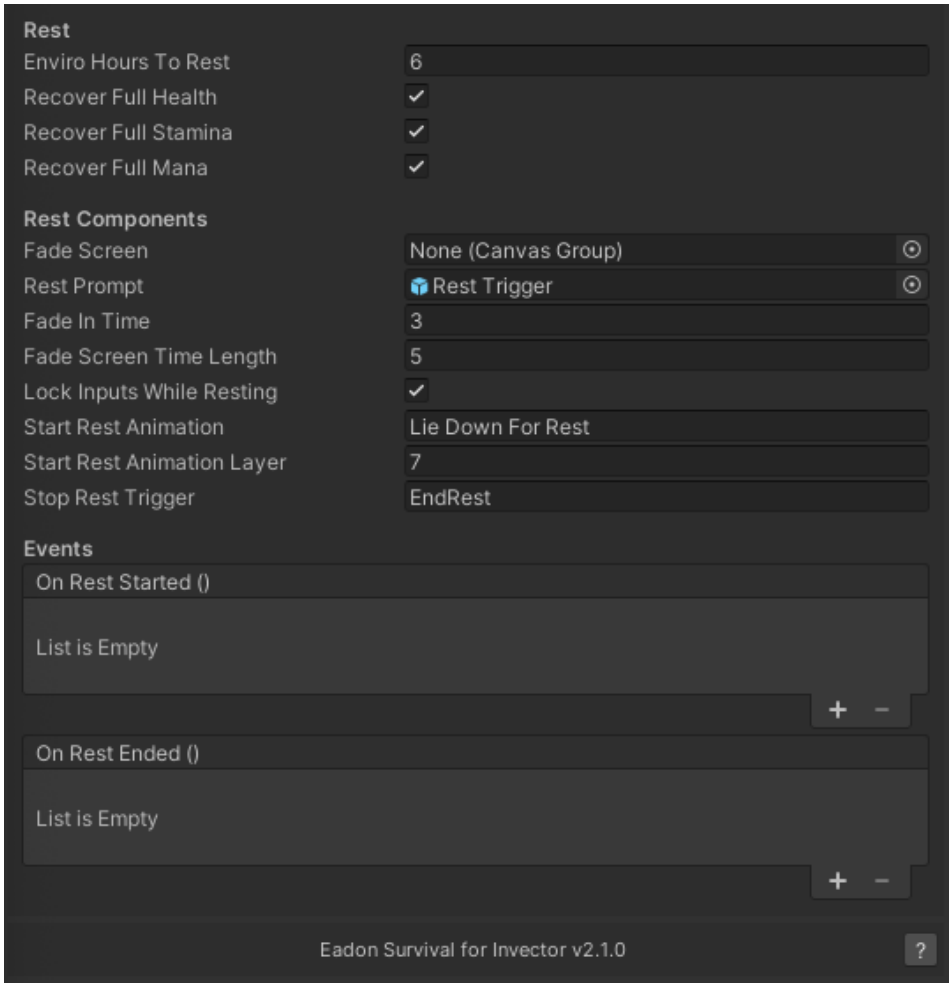
The extra fields are:

Field	Use
Heal Amount	The amount of health to recover in an interval of time (see next field)
Heal Rate	How often does the character recover health, in seconds
Sicknesses To Heal	A list of sicknesses that can be healed by standing in front of the bonfire
Injuries To Heal	A list of injuries that can be healed by standing in front of the bonfire

Resting Bonfires

A resting bonfire, when lit, allows you to rest for a specified period and recover full health and stamina. If used in conjunction with Eadon RPG for Invector, full mana is also recovered. Resting will fade the screen to a rest screen and fade it back to normal after a configurable amount of time. If Enviro is in the project, in game time will also progress for a specified amount of hours while the player is resting. Resting bonfire look like this:


Eadon
Resting Bonfire
Bonfire Components
Fire Vfx
WildFire (Particle System)
Add Fuel Prompt
None (Game Object)
Bonfire Trigger Prompt
Text (Text)
Light Up Prompt
light up
Extinguish Prompt
extinguish
Bonfire Trigger
Bonfire Trigger (V Trigger Generic Action)
Bonfire Configuration
Start Lit
Extinguish On Leave
Start Animation
Light Bonfire
Light Start Delay
4
Stop Animation
Extinguish Bonfire
Extinguish Delay
3
Add Fuel Animation
Add Fuel
Add Fuel Animation Layer
7
Temperature Change
10
Trigger Range
4
Fuel Item
5
Fuel Min Amount
10
Fuel Units
100
Burn Rate
1
Current Fuel
0
Debug Messages
Events
On Lit ()
List is Empty
On Extinguished ()
List is Empty
On Fuel Added ()
List is Empty

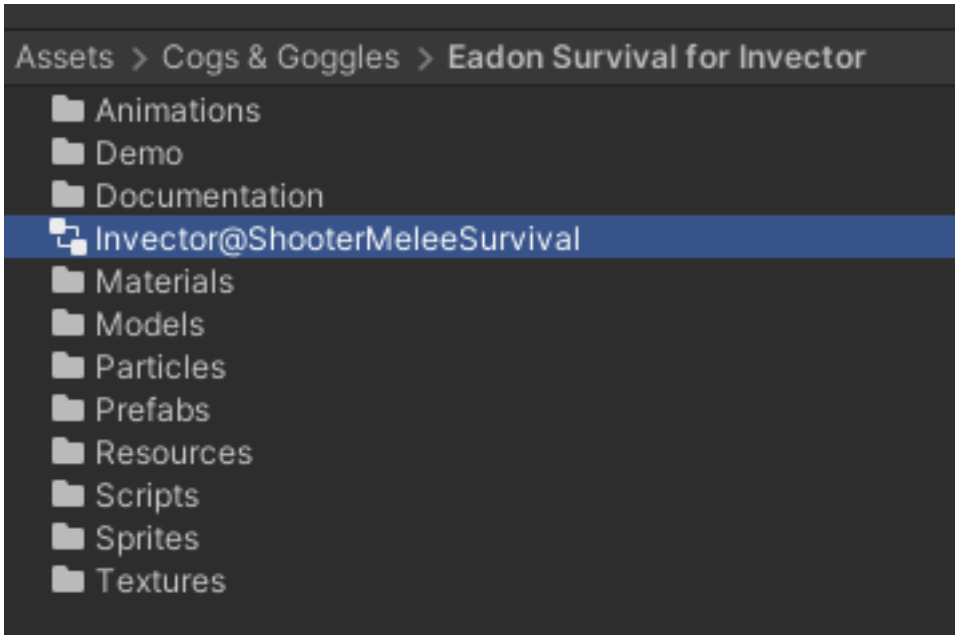


The extra fields are:

Field	Use
Enviro Hours To Rest	How many hours will Enviro advance while the character is resting (this option is enabled only if Enviro is I the project)
Recover Full Health	A flag to indicate if full health is restored
Recover Full Stamina	A flag to indicate if full stamina is restored
Recover Full Mana	A flag to indicate if full health is restored (this option is enabled only if Eadon RPG for Invector is in the project)
Fade Screen	A Canvas Group for the rest screen (used to fade in and out
Rest Prompt	A reference to the rest action trigger
Fade In Time	How long, in seconds, it takes to fade the rest screen
Fade Screen Time Length	How long the rest screen stays active (fade in/out time is in addition)
Lock Inputs While Resting	A flag to determine if player inputs are locked while resting

Start Rest Animation	The animation to play when rest starts
Start Rest Animation Layer	The animator layer that contains the start rest animation
Stop Rest Trigger	The trigger to end the rest animation
On Rest Started	An event triggered when rest starts
On Rest Ended	An event triggered when rest ends

For the setup of animations, see the notes on the normal bonfire. This asset contains simple start/loop/rest animations, and you can see an example of how to set them up in your custom animator by checking the Fullbody layer of this animator:

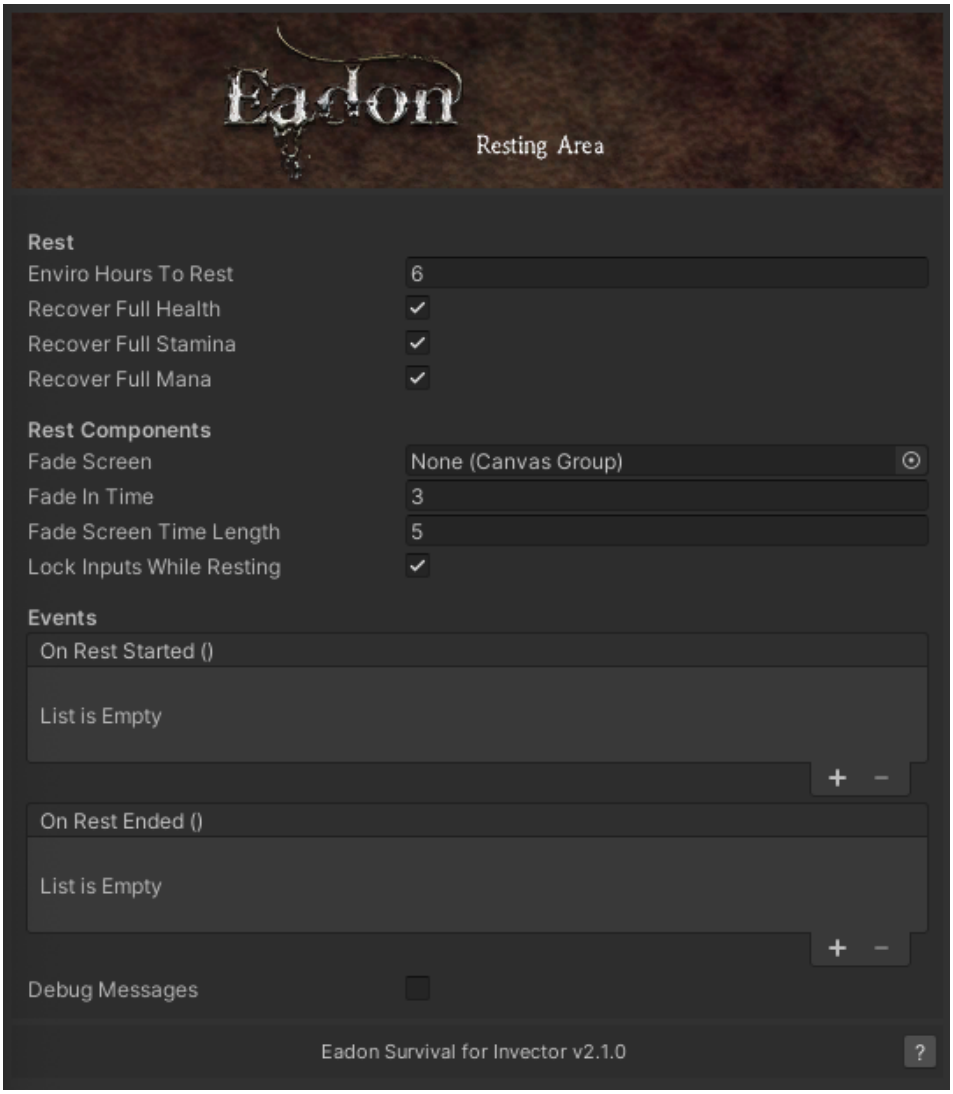


These two last events can be used, for example, to hide UI elements and prevent them from being on screen during rest.

The Enviro time progression feature requires an EadonEnviroManager component on an object in your scene (in addition to Enviro Sky Manager). See the chapter on Bonus Content for more information on this component.

Resting Area

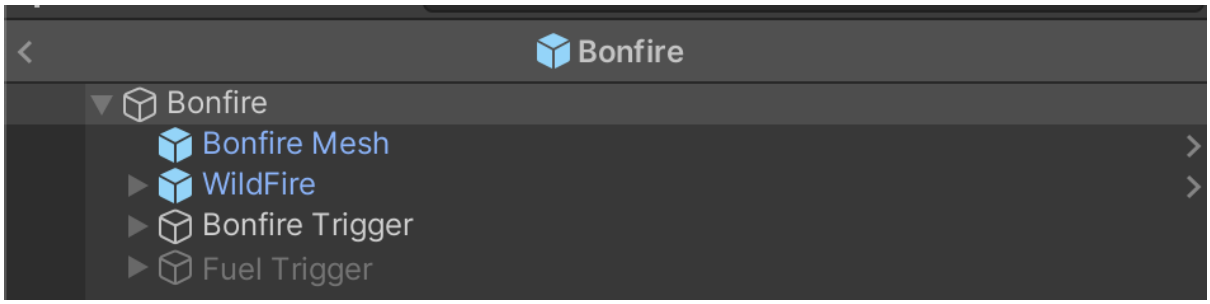
A resting area lets you use the rest functionality without a bonfire. It looks like this:



The fields are the same as for a resting bonfire.

Customizing a bonfire

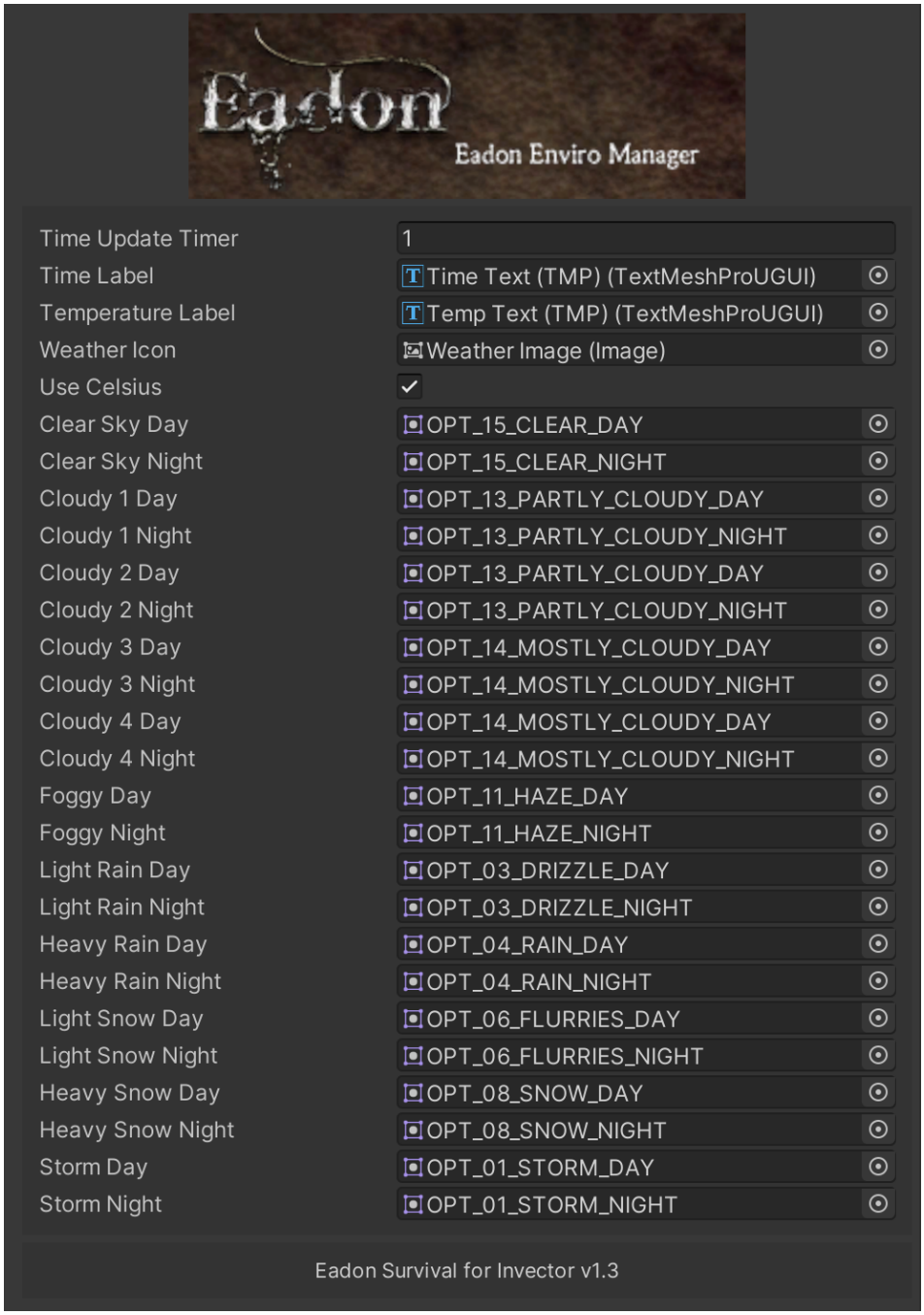
Creating your own bonfires is very simple. The prefab for a bonfire looks like this:



You can replace the bonfire mesh with any other mesh. If you replace the fire VFX prefab, just assign the reference to the new object in the Fire Vfx field in the inspector.

Bonus Content

In the Integrations/Enviro folder you can find a prefab that you can put in your scene which will show the current time, temperature and weather from Enviro. It uses a component called EadonEnviroManager, which looks like this:

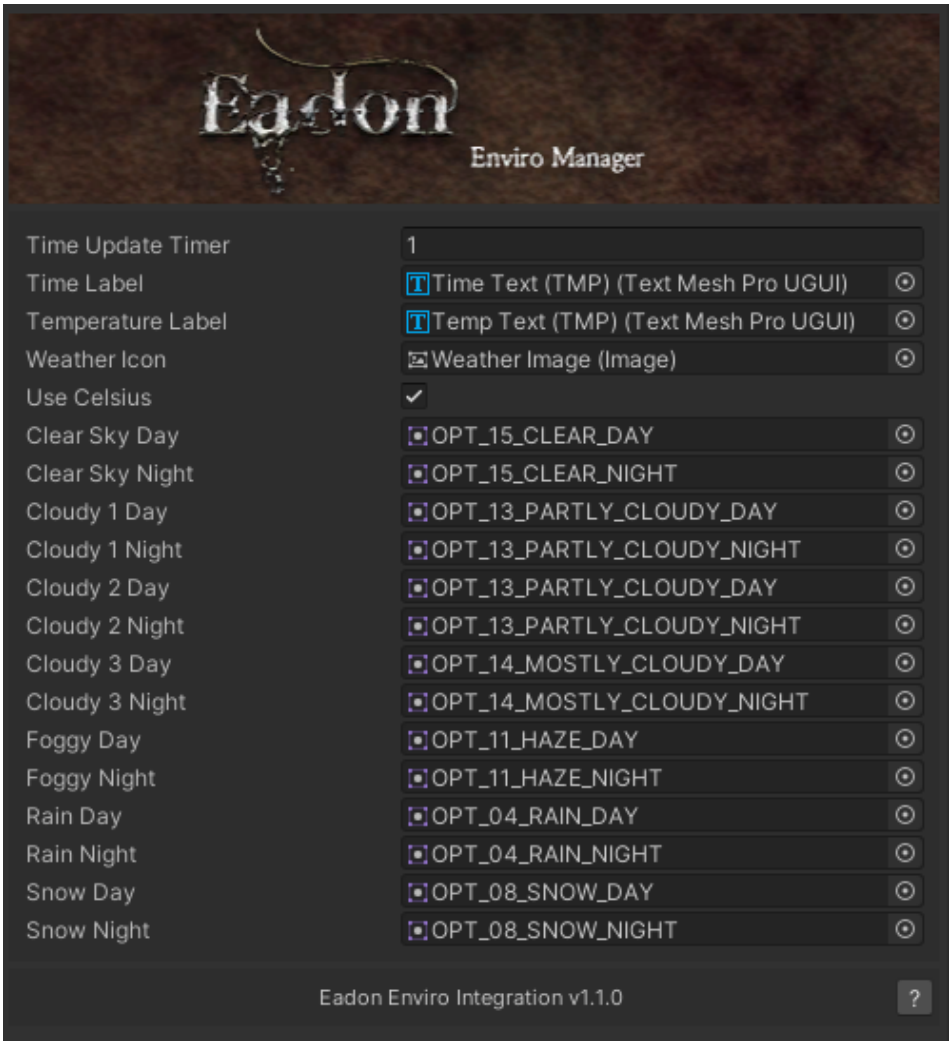

Eadon
Eadon Enviro Manager
Time Update Timer
1
Time Label
Time Text (TMP) (TextMeshProUGUI)
Temperature Label
Temp Text (TMP) (TextMeshProUGUI)
Weather Icon
Weather Image (Image)
Use Celsius
Clear Sky Day
OPT_15_CLEAR_DAY
Clear Sky Night
OPT_15_CLEAR_NIGHT
Cloudy 1 Day
OPT_13_PARTLY_CLOUDY_DAY
Cloudy 1 Night
OPT_13_PARTLY_CLOUDY_NIGHT
Cloudy 2 Day
OPT_13_PARTLY_CLOUDY_DAY
Cloudy 2 Night
OPT_13_PARTLY_CLOUDY_NIGHT
Cloudy 3 Day
OPT_14_MOSTLY_CLOUDY_DAY
Cloudy 3 Night
OPT_14_MOSTLY_CLOUDY_NIGHT
Cloudy 4 Day
OPT_14_MOSTLY_CLOUDY_DAY
Cloudy 4 Night
OPT_14_MOSTLY_CLOUDY_NIGHT
Foggy Day
OPT_11_HAZE_DAY
Foggy Night
OPT_11_HAZE_NIGHT
Light Rain Day
OPT_03_DRIZZLE_DAY
Light Rain Night
OPT_03_DRIZZLE_NIGHT
Heavy Rain Day
OPT_04_RAIN_DAY
Heavy Rain Night
OPT_04_RAIN_NIGHT
Light Snow Day
OPT_06_FLURRIES_DAY
Light Snow Night
OPT_06_FLURRIES_NIGHT
Heavy Snow Day
OPT_08_SNOW_DAY
Heavy Snow Night
OPT_08_SNOW_NIGHT
Storm Day
OPT_01_STORM_DAY
Storm Night
OPT_01_STORM_NIGHT
Eadon Survival for Invector v1.3

The fields are:

Field	Use
Time Update Timer	Frequency, in seconds, for checking time and weather in Enviro
Time Label	A TMP_Text component in which to display the time
Temperature Label	A TMP_Text component in which to display the temperature
Weather Icon	An Image in which to display the current weather
Use Celsius	Untick if you want temperature displayed in Fahrenheit

The remaining fields are icons to be used with every weather preset.

The Enviro 3 version looks like this:


Eadon
Enviro Manager
Time Update Timer
1
Time Label
Time Text (TMP) (Text Mesh Pro UGUI)
Temperature Label
Temp Text (TMP) (Text Mesh Pro UGUI)
Weather Icon
Weather Image (Image)
Use Celsius
Clear Sky Day
OPT_15_CLEAR_DAY
Clear Sky Night
OPT_15_CLEAR_NIGHT
Cloudy 1 Day
OPT_13_PARTLY_CLOUDY_DAY
Cloudy 1 Night
OPT_13_PARTLY_CLOUDY_NIGHT
Cloudy 2 Day
OPT_13_PARTLY_CLOUDY_DAY
Cloudy 2 Night
OPT_13_PARTLY_CLOUDY_NIGHT
Cloudy 3 Day
OPT_14_MOSTLY_CLOUDY_DAY
Cloudy 3 Night
OPT_14_MOSTLY_CLOUDY_NIGHT
Foggy Day
OPT_11_HAZE_DAY
Foggy Night
OPT_11_HAZE_NIGHT
Rain Day
OPT_04_RAIN_DAY
Rain Night
OPT_04_RAIN_NIGHT
Snow Day
OPT_08_SNOW_DAY
Snow Night
OPT_08_SNOW_NIGHT
Eadon Enviro Integration v1.1.0
?

License

You agree that Cogs & Goggles own all right, title and interest in this Asset, including without limitation all applicable Intellectual Property Rights. "Intellectual Property Rights" means any and all intellectual property rights wherever in the world and whenever arising (and including any application), including patent laws, copyright, trade secrets, know-how, confidential information, business names and domain names, computer programs, trademark laws, service marks, trade names, utility models, design rights, semi-conductor topography rights, database rights, goodwill or rights to sue for passing off, and any and all other proprietary rights worldwide. You agree that you will not, and will not allow any third party to,

(i) copy, sell, license, distribute, transfer, modify, adapt, translate, prepare derivative works from, decompile, reverse engineer, disassemble or otherwise attempt to derive source code from the Asset, unless otherwise permitted,

(ii) take any action to circumvent or defeat the security or content usage rules provided, deployed or enforced by any functionality (including without limitation digital rights management or forward-lock functionality) in the Asset,

(iv) remove, obscure, or alter Cogs & Goggles' or any third party's copyright notices, trademarks, or other proprietary rights notices affixed to or contained within the Unity Asset Store or Assets.

YOU EXPRESSLY UNDERSTAND AND AGREE THAT YOUR USE OF THE ASSET IS AT YOUR SOLE RISK AND THAT THE ASSET IS PROVIDED “AS IS” AND “AS AVAILABLE” WITHOUT WARRANTY OF ANY KIND, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW. IN PARTICULAR, COGS & GOGGLES, ITS SUBSIDIARIES, HOLDING COMPANIES AND AFFILIATES, AND ITS LICENSORS DO NOT REPRESENT OR WARRANT TO YOU THAT:

(A) YOUR USE OF THE ASSETS WILL MEET YOUR REQUIREMENTS,

(B) YOUR USE OF THE ASSETS WILL BE UNINTERRUPTED, TIMELY, SECURE OR FREE FROM ERROR,

(C) ANY INFORMATION OBTAINED BY YOU AS A RESULT OF YOUR USE OF THE ASSETS WILL BE ACCURATE OR RELIABLE, AND

(D) THAT DEFECTS IN THE OPERATION OR FUNCTIONALITY OF ANY SOFTWARE PROVIDED TO YOU AS PART OF THE ASSETS WILL BE CORRECTED.

YOUR USE OF THE ASSET IS AT YOUR OWN DISCRETION AND RISK AND YOU ARE SOLELY RESPONSIBLE FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM, OR OTHER DEVICE, OR LOSS OF DATA THAT RESULTS FROM SUCH USE.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, COGS & GOGGLES FURTHER EXPRESSLY DISCLAIMS ALL WARRANTIES TERMS OR CONDITIONS OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO ANY IMPLIED WARRANTIES TERMS AND CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.