

Upgrades Tree PRO

Documentation

1.0.1

14-01-2026

Holender Games

Overview	2
Getting Started	2
Your Upgrades	3
Upgrade Object	4
How To Build You Upgrades Tree	5
Prefabs	6
Tooltip	6
UpgradeButton	6
Helper Prefabs	6

Overview

Upgrades Tree is a simple and efficient tool for creating a vast upgrade tree for your game. With some important functionality built it such as:

Scrollable view, Currency events, multiple levels per upgrade, upgrades unlock system, reflective UI to show which upgrade is affordable at any moment and more.

PRO VERSION:

Includes a fully working game example to show all systems and interactions.

Shows how your upgrades can affect real game stats and be applied to any game you wish to create.

Systems in the PRO version include: **Stats System, Upgrade-Effects System and a TextStyle System (BONUS: Sounds System:** fully featured system supporting pooling of audio sources, assigning sound effects using enums and defining random pitch/volume)

All PRO systems are documented at the end of this document.

Demo Scenes

Check out the various demo scenes in the asset to get familiar with the asset capabilities.

- **Demo_Wizards** - the main demo, showing a full implementation of the upgrade tree into a training grounds for wizards.
- **Demo_Examples** - featuring few examples of different ways to customize your trees
 - **Example_LockedAndVisible:** shows how to use the Locked but visible upgrade feature. This allows for skill trees and other ways to allow the player to plan forward.
 - **Example_Spiral** - A spiral shaped tree with different sized and visuals for upgrades.
 - **Example_Skills** - A tree shaped skill-tree similar to popular ARPGS. Featuring different kinds of upgrades visuals, connectors and orientations.

Getting Started

The best way to learn how to work with the asset is investigating the Demo scene included.

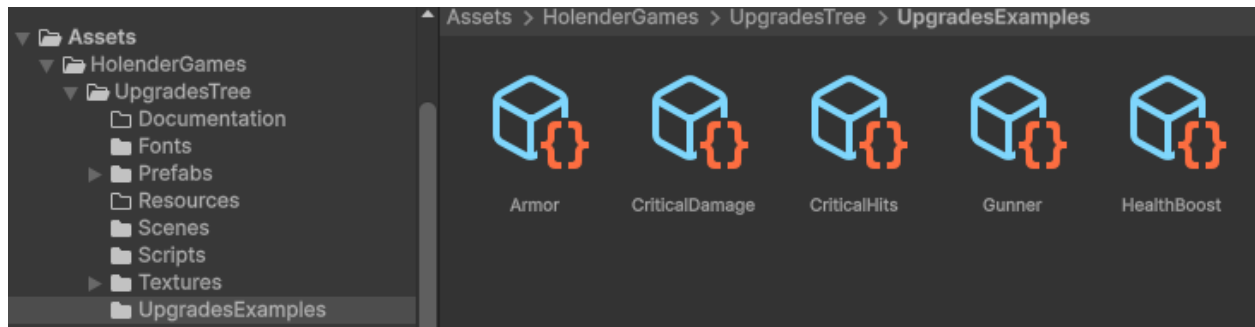


The main components and concepts of Upgrades Tree are:

- **UpgradeData**- the main scriptable object for storing the upgrade data for each upgrade in your game (name, descriptions, costs, effects, parent upgrade, icon, etc)
- **UpgradeView**- handles the upgrade view in the tree
- **UpgradeTooltip**: The UI element that is responsible for viewing the tooltip of your upgrade while hovering on it.
- **UpgradesTreeManager**- managing unlocks of new upgrades and handling purchased upgrades according to your own game.
- **CurrencyManager**: a simple manager for the game's currency as an example. Can be extended to your own game multiple currencies
- **CurrencyViewer**: a simple script to show the currency to the player.

Your Upgrades


Create upgrades by creating or duplicating the example upgrades and edit them as you wish.



Upgrade Object

The basic upgrade object contains the following for you to edit:

- Upgrade Name
- Icon
- Currency Type (**PRO**)
- Parent Upgrade: the upgrade that required to unlock this upgrade. (once the parent upgrade is being maxed, this upgrade will show up)
- Levels Data (**PRO**)
 - Description to describe each level of this upgrade
 - Cost to define the cost of each level of this upgrade
 - Effects List to define which stats this upgrade changes and how

 Wizard_Damage (Upgrade Data) Open

Script

UpgradeData

Upgrade Name

Strong Fireball

Icon

icons_28

Currency Type

Gold

Unlock When Parent Maxed

☒

Unlock At Parent Level

1

Parent Upgrade

None (Upgrade Data)

Levels Data

3

▼ Increase the Wizard's Fireball <red>Damage</red> by 20%.

Description

Increase the Wizard's Fireball <red>Damage</red> by 20%.

Cost

50

▼ Effects

1

▼ Element 0

Stat Type

Wizard_Damage

Operation

Multiply

Value

1.2

Show On Tooltip

☒

Tooltip Label

Damage

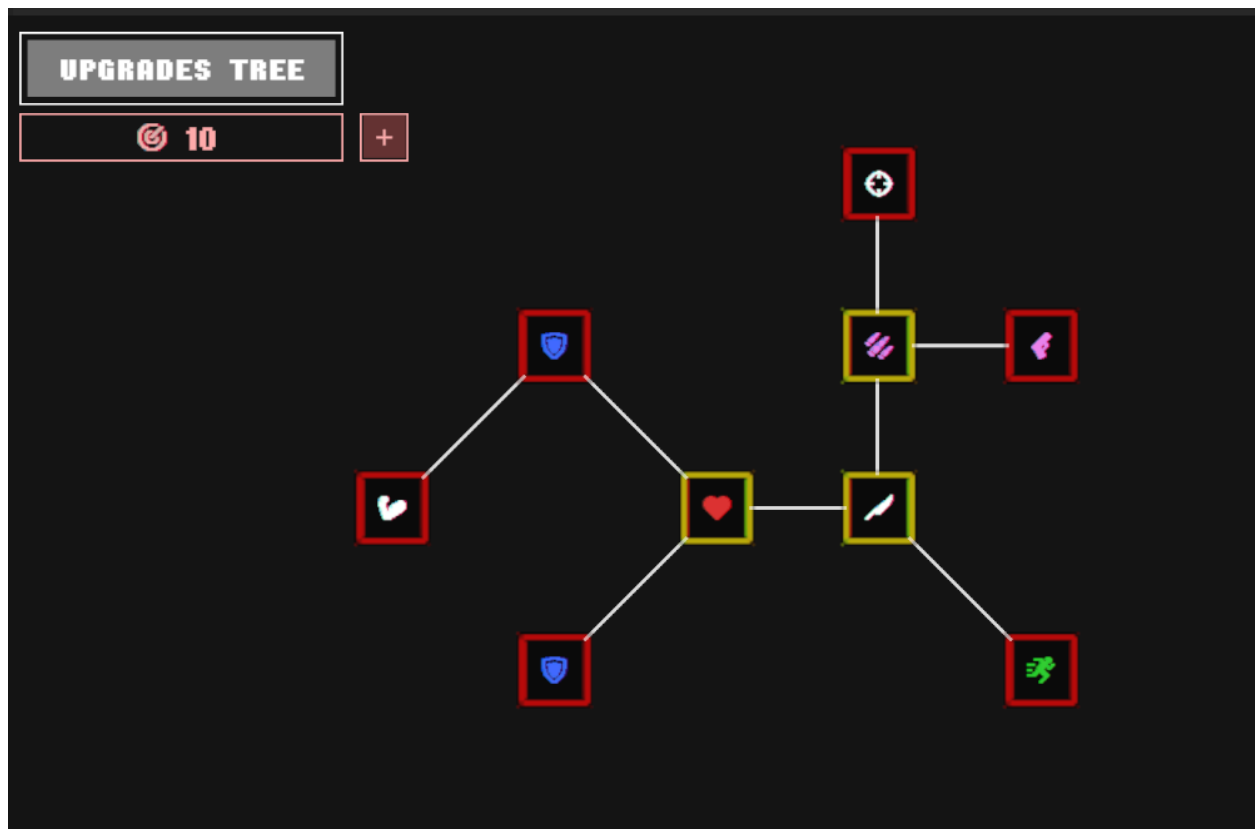
+

-

How To Build You Upgrades Tree

1. Create upgrades scriptable object as shown above
2. Create the tree layout using the helper Prefabs
3. Assign each prefab its relevant upgrade scriptable object.
4. **TIP: the space between the upgrade is set to exactly 100pt so you can easily position them (on both X and Y axis)**

That's All !



Prefabs

Tooltip

The tooltip visuals. Can be modified as you wish according to your game.

UpgradeButton

The base upgrade button includes the line connector to its parent upgrade.

Just adjust the line to connect visually to the upgrade you want.

Or use the help prefabs.

Helper Prefabs

These prefabs are just variants of the UpgradeButton with the connector line adjusted to each of the directions:

Up, Down, Left, Right, Up-Right, Up-Left, Down-Right, Down-Left

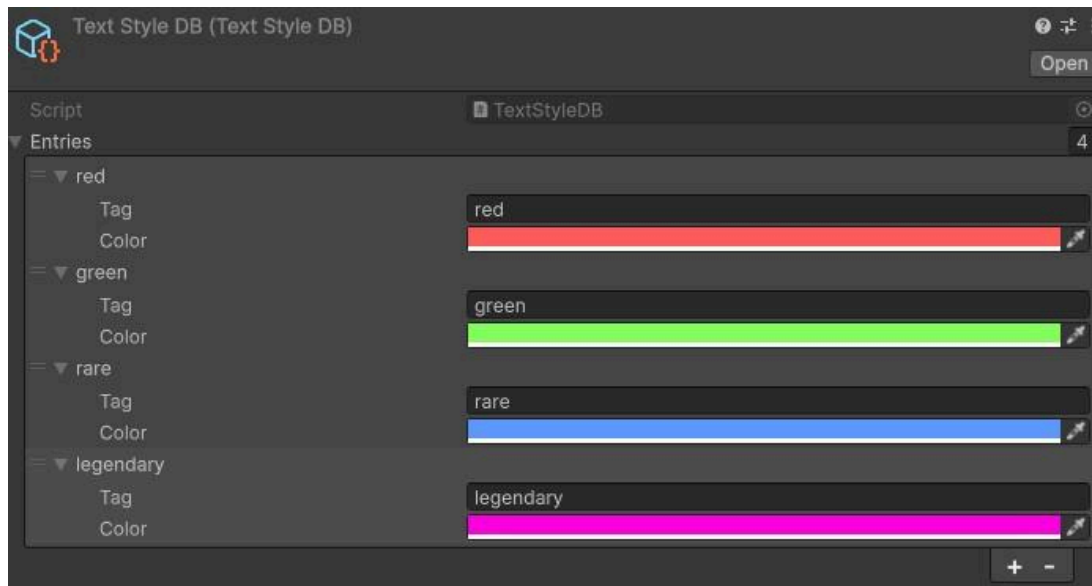
For your convenience!

PRO Systems

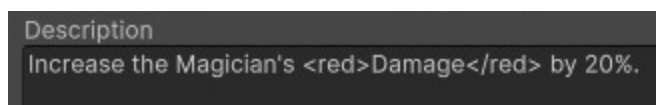
Style Text - Color tags

You can define any amount of tags and colors to use to mark and colorize your upgrades descriptions for better visuality and nicer UI

DEFINE TAGS:



WRITE YOUR DESCRIPTION:

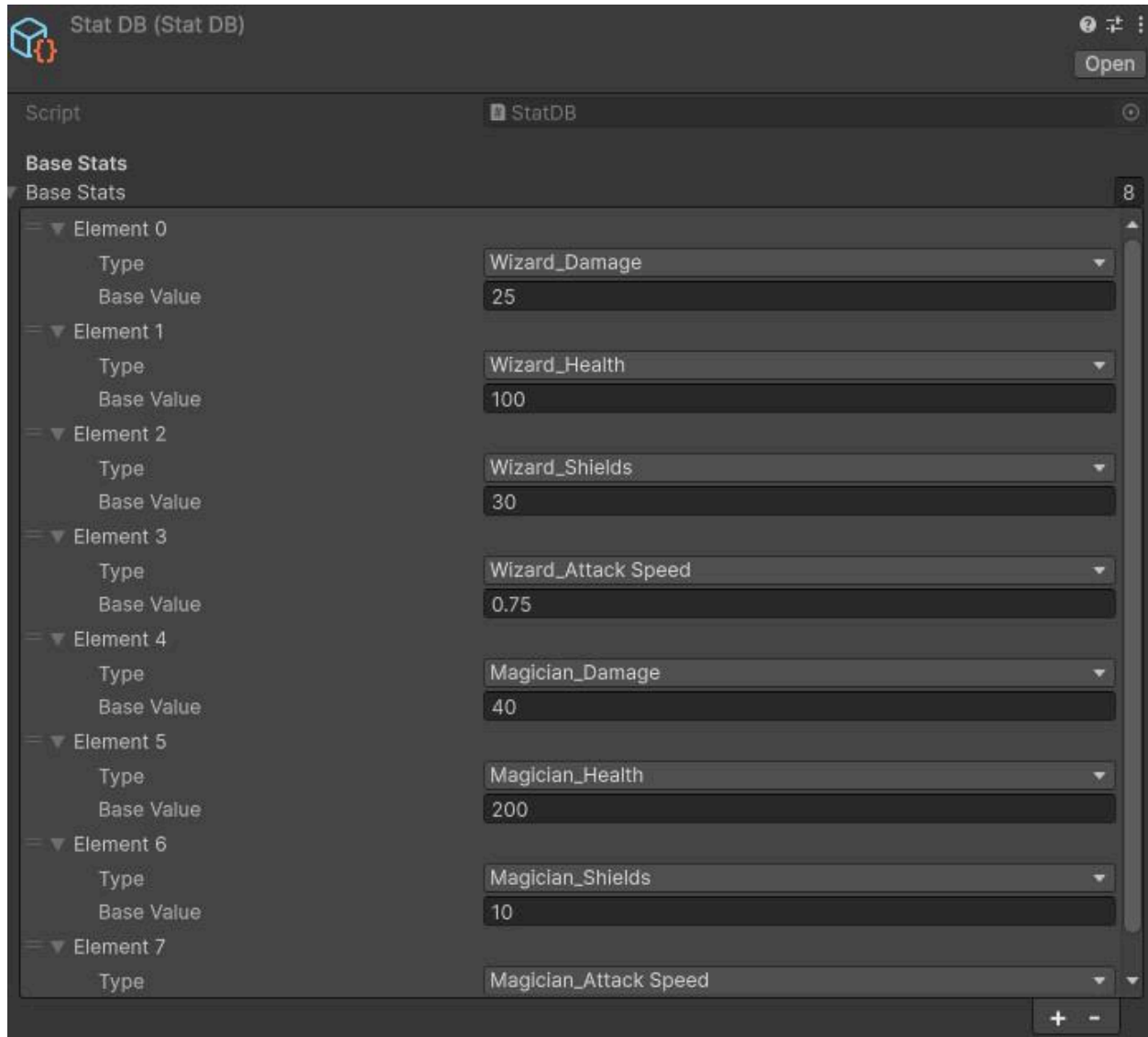


RESULT:



STATS System:

Define in a scriptable object DB all the upgradable stats of the game and unlockables, Such as **DAMAGE, SPEED, UNLOCK_WEAPON1, UNLOCK_WEAPON2**



These stats will be easily accessed by your game at any point and will be accessible to your upgrade system to manipulate with UpgradeEffects (see next)

EFFECTS System:

Effects are the mechanisms that upgrades use to manipulate STATS in your game.

Stat Type: which stat this effect modifies or unlocks.

Operation: what type of stat modification

They come in 3 types: **ADDITION, MULTIPLY, UNLOCK.**

examples:

ADDITION: Increase fireball damage by **2**. (damage + 2)

MULTIPLY: Increase Attack Speed by **30%**. (attack speed * 1.3)

UNLOCK: Unlock double fireball (doubleFireball = true)

Value: the addition/multiply value. For example 1.2 to increase a stat by 20%

Show On Tooltip (PRO): allows showing this effect preview on the tooltip.

Tooltip Label: the label that will show up for evaluation the effect.

For example, the tooltip will show: "Damage 25 -> 30".



You can add multiple effects to 1 upgrade (add damage but reduce hp)

Example using: effects, Color Coding, multi-level upgrade

