



v1.0.0

Notice

This document is an extract from the online documentation available at:

<https://jnamobile.zendesk.com/hc/en-gb>

It is recommended you use the online documentation where possible.

Support

To get support visit: <https://jnamobile.zendesk.com/>

Or send an email to: support@jnamobile.com

Remember to include your Asset Store Order number in all support requests.

Idle Clicker Kit

Getting Started

— The Basics

Idle Clicker Kit uses a simple approach befitting for a simple genre. Almost all new behaviours are added to your game by dropping prefabs on to the UI Canvas.

Minimalist Idle Game

Steps to build the simplest possible idle game: a click display and a clicker.

Create a New Scene

1. Create the scene

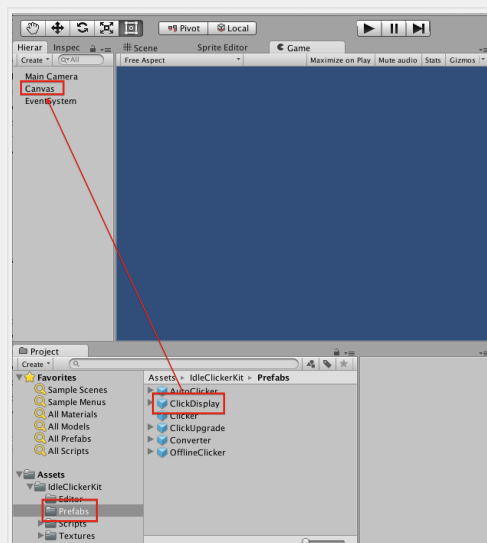
File -> New Scene

2. Add a UI Canvas

GameObject->UI->Canvas

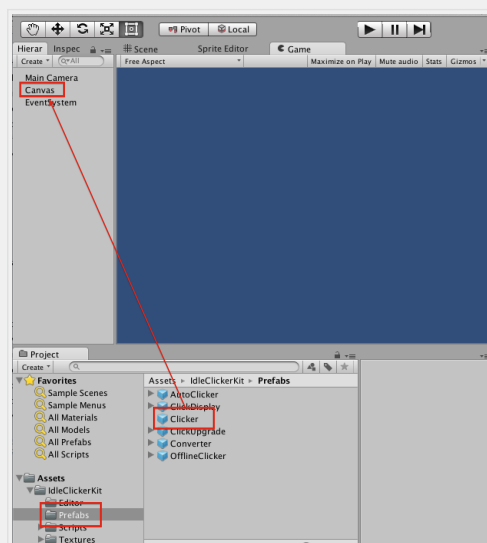
Add a Click Display

1. Drag the *ClickDisplay* prefab to the Canvas GameObject:



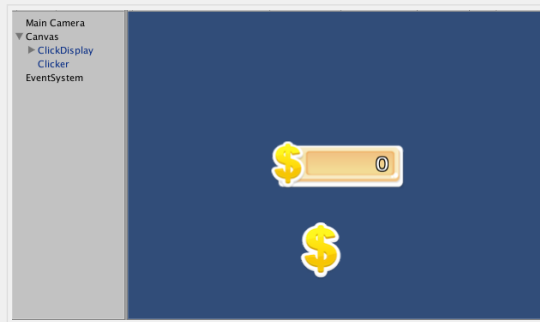
Add a Clicker

1. Drag the *Clicker* prefab to the Canvas GameObject:



Test it Out

1. Press the **Play** button and enjoy your game.

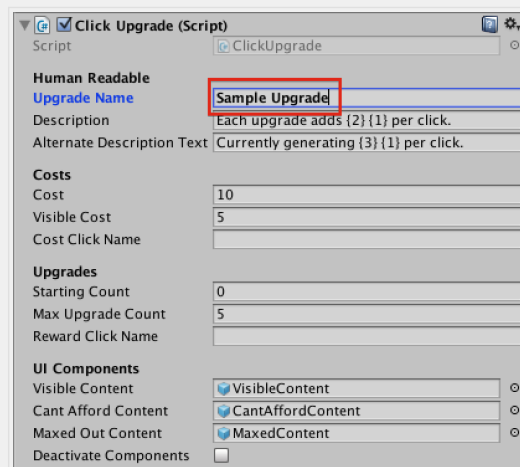


Adding Upgrades

Lets add some upgrades to enhance our game.

Add a Click Upgrade

1. Drag the *ClickUpgrade* prefab to the Canvas.
2. Select the newly Created ClickUpgrade gameObject in the Scene Hierarchy.
3. Name your click upgrade, we called ours 'Sample Upgrade':



4. Press the **Play** button and enjoy your upgrade.

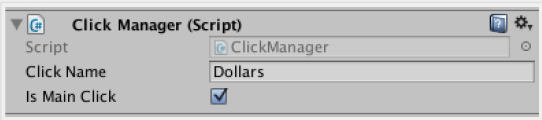
Note: By Default it will cost 10 clicks to buy the upgrade and the upgrade will boost clicks by 3. You can change these values in the inspector for the *ClickUpgrade* component.

Managers

Click Manager

The ClickManager component manages the basic loading and saving of click data.

You should add a click manager for each type of click in your game. One of these click managers should be marked as the main (i.e. default) clicks.



Note: Clicks will still work without a ClickManager in the scene, but they will not have proper naming.

Customisable Fields

Click Name	Mandatory	The name of the click being managed. Should be unique between ClickManagers.
Is Main Click	Optional	If ticked this click type will be the main (or default) click.

[View page](#)

Component Library - Interactive Components

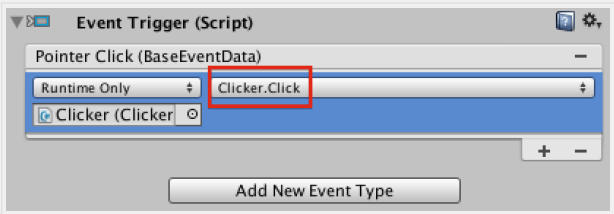
Clicker

The clicker is an interactive component that users click to generate clicks (or dollars).

The Click Method

The main interaction with the clicker is the public method Click().

Usually you would call this method through a UI Event or Button:



Customisable Fields

Click Name	Optional	The name of the click to generate. Blank means generate the default click type.
------------	----------	--

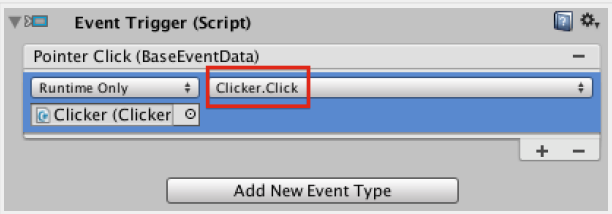
Cooldown Clicker

The Cooldown Clicker is an interactive component that users click to generate clicks. Unlike the base Clicker it has a cool down period after a click in which it cannot be clicked.

The Click Method

The main interaction with the clicker is the public method Click().

Usually you would call this method through a UI Event or Button:



Customisable Fields

Click Name	Optional	The name of the click to generate. Blank means generate the default click type.
Cool Down Time	Mandatory	The cool down time between clicks.
Image	Optional	An image with the <i>ImageType</i> set to <i>Filled</i> . This will be used to show the cool down progress.

Base Upgrade

The base upgrade is an abstract class that most upgrades inherit from.

Base Upgrade Fields

All upgrades have the following fields:

Upgrade Name	Optional	Name of the upgrade, usually displayed in the UI.
Description Text	Optional	Description of the upgrade which can include variables.

Alternate Description Text	Optional	Alternate description of the upgrade which can include variables. Usually used by the tooltip when the upgrade has been purchased.
Cost	Mandatory	How much the upgrade costs.
Visible Cost	Mandatory	The number of clicks required before the upgrade switches from hidden to visible. Usually half or quarter the cost.
Cost Click Name	Optional	The name of the resource (click) that is used to purchase this upgrade. By default it is blank which refers to the default click.
Starting Count	Mandatory	Number of upgrades the user starts the game with. Typically 0.
Max Upgrade Count	Mandatory	Maximum number of upgrade allowed. Use 0 for unlimited.
Reward Click Name	Optional	What type of click does this upgrade apply to. Blank means upgrade the default click.
Visible Content	Mandatory	Reference to a GameObject which is shown (set active) when the upgrade is visible.
Cant Afford Content	Mandatory	Reference to a GameObject which is shown (set active) when the upgrade is visible but the user doesn't have enough resources (clicks) to pay for it.
Maxed Out Content	Optional (Mandatory if max is specified)	Reference to a GameObject which is shown (set active) when the upgrade is visible but the user has purchased the maximum amount allowable.
Deactivate Components	Optional	If <i>true</i> activating one of the visible components will deactivate the others. If <i>false</i> the components will be laid one on top of the other. In this mode you must ensure the components are in the appropriate order (Visible then Can't Afford then Maxed Out).
Pips	Optional	Optional array of GameObjects that will be activated sequentially with each upgrade. Use 'pips' to show the user how many upgrades they have purchased.

[View page](#)

[Click Upgrade](#)

Upgrade which adds to the amount of clicks generated per user click.

Click Upgrade Fields

Clicks Added	Mandatory	The number of clicks added to the base clicks when the upgrade is purchased.
--------------	-----------	--

Base Upgrade Fields

See: [Base Upgrade](#)

[View page](#)

— Auto Clicker

The auto clicker upgrade generates clicks without requiring the user to click.

Auto Clicker Fields

Click Amount	Mandatory	The number of clicks added each time the auto clicker fires.
Interval	Mandatory	How often the auto clicker generated clicks.

Base Upgrade Fields

See: [Base Upgrade](#)

[View page](#)

— Offline Clicker

The offline clicker generates clicks even when the player has the game closed. Clicks are generated in to 'storage' and there are various options for how clicks can be collected.

Offline are sometimes called 'advisors' who generate revenue/clicks/etc for you.

Offline Clicker Fields

Click Amount	Mandatory	The number of clicks added each time the auto clicker fires.
Time Interval	Mandatory	How often the offline clicker generates clicks.
Storage	Optional	How much can be stored in this offline clicker. 0 means unlimited storage.
		How are the clicks collected from storage. There are three options: AUTO_WHEN_FULL - The clicks will automatically be collected when the storage is filled. MANUAL WHEN FULL - The user must click
Collection Type	Mandatory	

the storage to collect and can only do so when the storage is full.

MANUAL_ANY_TIME - The user can click storage at any time to gather what clicks have been stored.

Storage Increases ...	Optional	If checked then storage will be multiplied by upgrade level. Else storage remains constant.
-----------------------	----------	---

Base Upgrade Fields

See: **Base Upgrade**

[View page](#)

- Click Converter

The click converter changes clicks of one type in to clicks of another type. For example if you have clicks for iron and gold you could use a *Click Converter* to turn iron in to gold.

Click Converter Fields

Cost Click Name	Mandatory	The name of the click we are converting from.
Cost Click Amount	Mandatory	The amount of clicks it costs.
Reward Click Name	Mandatory	The name of the click we are converting to.
Reward Click Amount	Mandatory	The number of reward clicks that are added.

[View page](#)

- Show Object Upgrade

The *Show Object Upgrade* simply activates a game object when an upgrade has been purchased. You can use it to (for example) enable new pieces of interface when the user unlocks a new item.

The show object upgrade has no unique fields, it uses the 'pips' field of *Base Upgrade* to define the GameObject to be activated.

Base Upgrade Fields

See: **Base Upgrade**

[View page](#)

Component Library - Display Components

- Upgrade Details Text

The *Upgrade Details Text* allows you to easily construct information about any given upgrade. The text is derived from the upgrades themselves so you can easily use it in prefabs, etc. The *Upgrade Details Text* component should be on the same Game Object as a **Text** component.

Upgrade Details Text Fields

Label Type	Mandatory	The type of details to show. The optional are:
		NAME - Show the upgrade name.
		DESCRIPTION - Show the upgrade description, substituting variables where applicable.
		ALT_DESCRIPTION - Show the upgrade alternative description, substituting variables where applicable.
		MIXED_DESCRIPTION - Show the alternate description if the upgrade count is bigger than 0 (i.e. at least one upgrade has been purchased). Otherwise show the base description.
		COST - Show the upgrade cost.
Upgrade	Optional	MAXED_NAME - Show the name with " (MAXED)" after it.
		Reference to the upgrade to show details for. If blank the component will search for a BaseUpgrade (or subclass) in its parents.

[View page](#)

Click Total

Click total simple shows the total number of clicks for a given click type. It should be added to a GameObject that has a **Text** component.

It has various options controlling how the clicks are shown and incremented.

Click Total Fields

Click Name	Optional	Name of the click to show, or blank for default click type.
Use Count Up	Mandatory	Should we count up to the new value. If false we instantly change.
Use K Symbol	Mandatory	Use a K symbol when the click count is larger than 100,000.
Use M Symbol	Mandatory	Use an M symbol when the click count is larger than 100,000,000.
Use B Symbol	Mandatory	Use a Bsymbol when the click count is larger than 100,000,000,000.
Use Commas	Mandatory	If true use commas to separate thousands. Else don't

[View page](#)

Click Label

[View page](#)

— Per Click Text

The Per Click Text shows how many clicks are generated with each click of a clicker. It should be added to a `GameObject` that has a **Text** component.

Click Total Fields

Click Name	Optional	Name of the click to show, or blank for default click type.
Text String	Mandatory	Defines a formatting string for the per click text. Variable {0} is the number of clicks and variable {1} is the name of the click.

[View page](#)

— Offline Clicker Percentage

Uses an image fill to show how full the [Offline Clicker](#) storage is (0 empty to 1 full). Should be attached to a `gameObject` with an **Image** component and an image type of **Filled**.

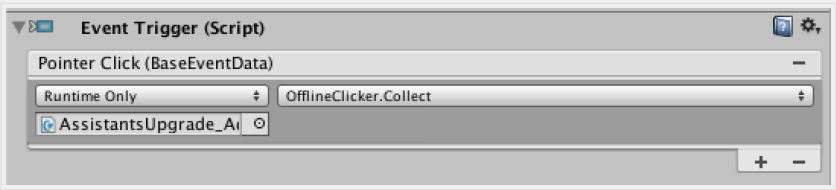
Offline Clicker Percentage Fields

Clicker	Optional	Reference to an Offline Clicker. If not assigned the component will try to find an Offline Clicker in its parent <code>GameObject</code> .
---------	----------	--

Event Trigger

You will frequently want to add an *Event Trigger* component to the same `GameObject` as the *Offline Clicker Percentage* component which will trigger the collect action when the image is clicked.

For example:



[View page](#)

— Tool Tip

A simple tooltip implementation which will pop up UI components if the user leaves the mouse stationary for a period of time. This component should be added to any UI object for which you want to show a tool tip (multiple Tool Tips may point to the same UI component).

You often use it in conjunction with [Upgrade Details Text](#) components in order to easily show details about the current state of an upgrade.

Tool Tip Fields

Tool Tip Component	Mandatory	Link to the <code>GameObject</code> to activate when the
--------------------	-----------	--

Tool Tip Component	Mandatory	
		tool tip is shown. (i.e. the UI component).
Move Tip to Mouse Position	Mandatory	<p>If true we will move the tip position to match the mouse position.</p> <p>If false tip will not be moved (for example you may want to display the tip in a static 'help' area).</p>
View page		

— Converter Details

Allows you to display values pulled a [Click Converter](#) object.

Converter Details Field

Label Type	Mandatory	Which type of value to show..
Coconverter	Optional	The converter to show values for. If not assigned the component will try to find a Click Converter in its parent GameObject.

[View page](#)

RECENT ACTIVITY (COMMUNITY)

No recent activities yet.