First Chest Documentation

Welcome to First Chest documentation.

This manual contains a quick-start walkthrough, Component details and script reference.

Introduction

First Chest is a set of 3D models, Particles, Textures, Scripts which helps you quickly add and setup gifts, presents and treasure chests system for your project. It is easy to use and modify. A beginner can easily understand and use it.

Features:

- 68 different style of Chests.
- 42 Shuriken particles including curse, gas, fire, firework, smoke, magic, glow, ray, sparkles, etc.
- 36 sound effects including open, close, lock, open fail, gas, ghost, electric zap, angelic, success, etc.
- 17 PSD templates with color palettes and layers. Modify them as you wish.
- 9 props such as key, heart, sword, shield, gold coins.
- 9 scripts written in c# (full source code and documentation).
- 1 demo scene.
- Scripts and components are compatible with custom 3D Chest models.
- Smooth animation with tweeners. iTween, LeanTween, HOTween are supported.
- Use Preprocessor Directives to switch between tweeners; Let you work with your favorite one or you can easily expand/edit scripts to match your other tweener.
- Supports all player platforms.
- Works on Unity and Unity Pro.

More informations, try First Chest webpage.

http://www.ge-team.com/firstchest/

Table of Contents

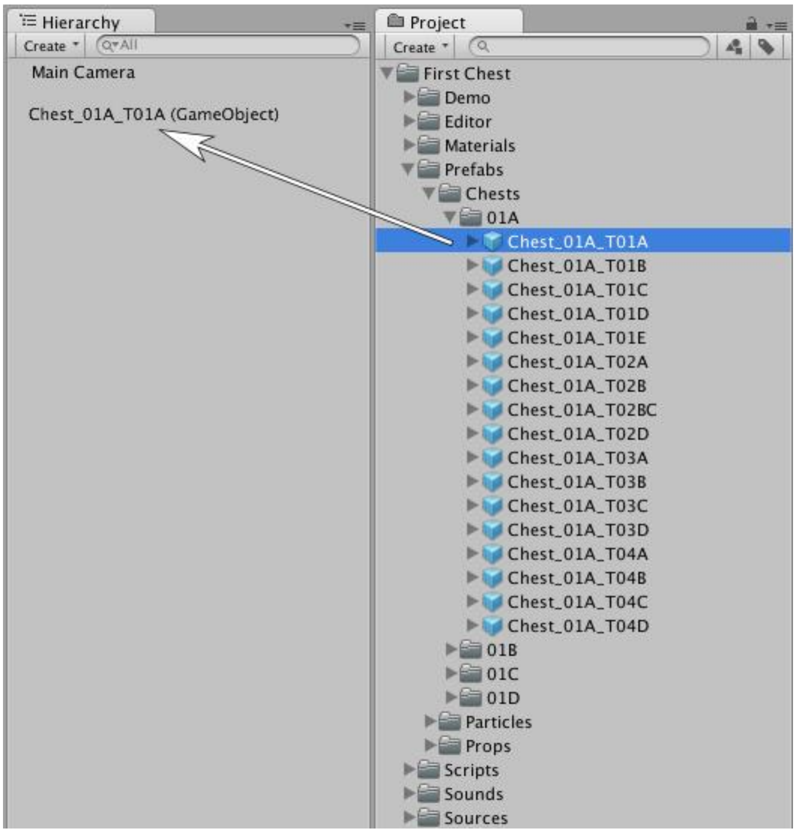
1. Quick-Start walkthrough

Follow these steps to figure out, or check out our video tutorial (http://youtu.be/bmtCBh5NrjQ).

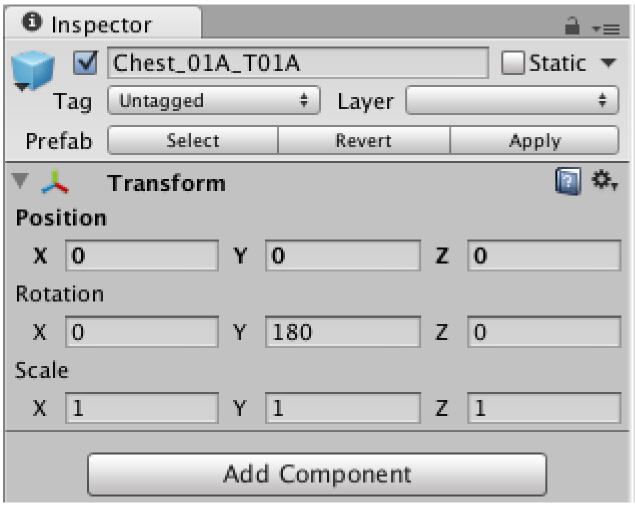
1. Import First Chest package.

2. Go to **Top menu->File->New Scene** to create new scene.

3. Drag a **Chest prefab** from **"First Chest/Prefabs/Chests/"** folder into **Hierarchy tab** or **Scene tab**.



4. In Hierarchy tab, double click on the Chest. Change transform value; position=(0, 0, 0) and Rotation=(0,180,0).



Inspector

☑ Chest_01A_T01A □ Static

Tag Untagged Layer

Prefab Select Revert Apply

Transform

Position

X 0 Y 0 Z 0

Rotation

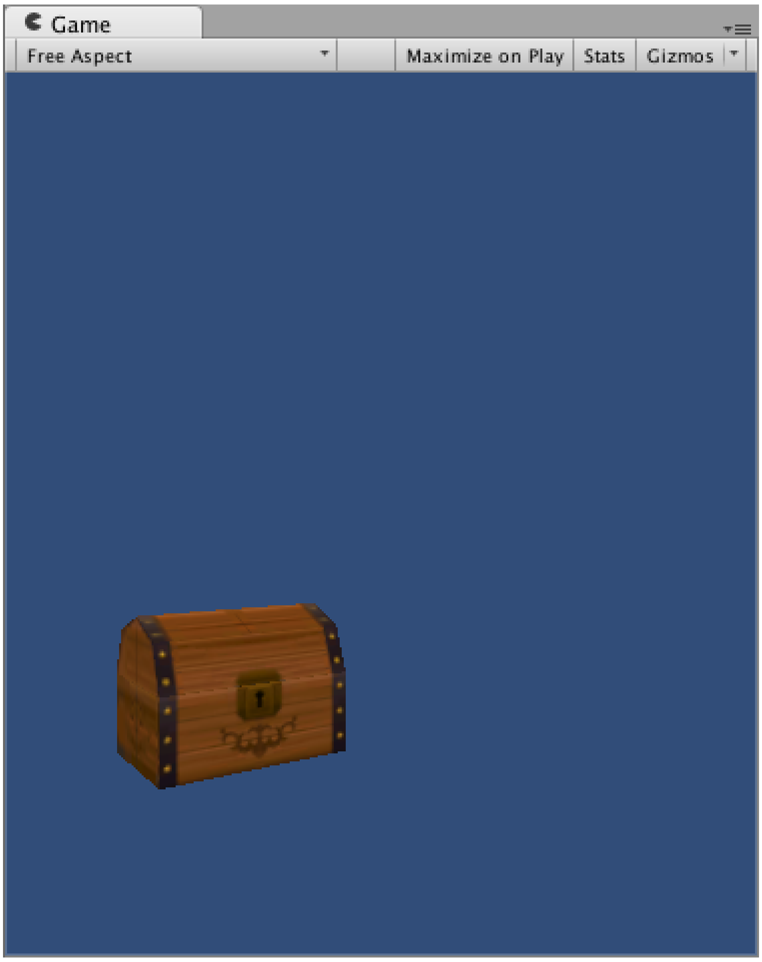
X 0 Y 180 Z 0

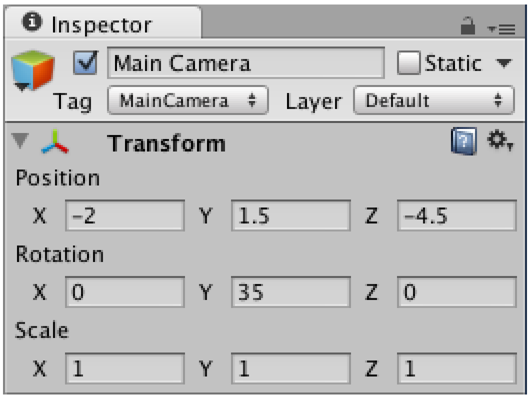
Scale

X 1 Y 1 Z 1

Add Component

5. In Scene tab, Select Main Camera and move it around to make sure you see front side of the Chest.





☑ Main Camera □ Static

Tag MainCamera Layer Default

Transform

Position

X -2 Y 1.5 Z -4.5

Rotation

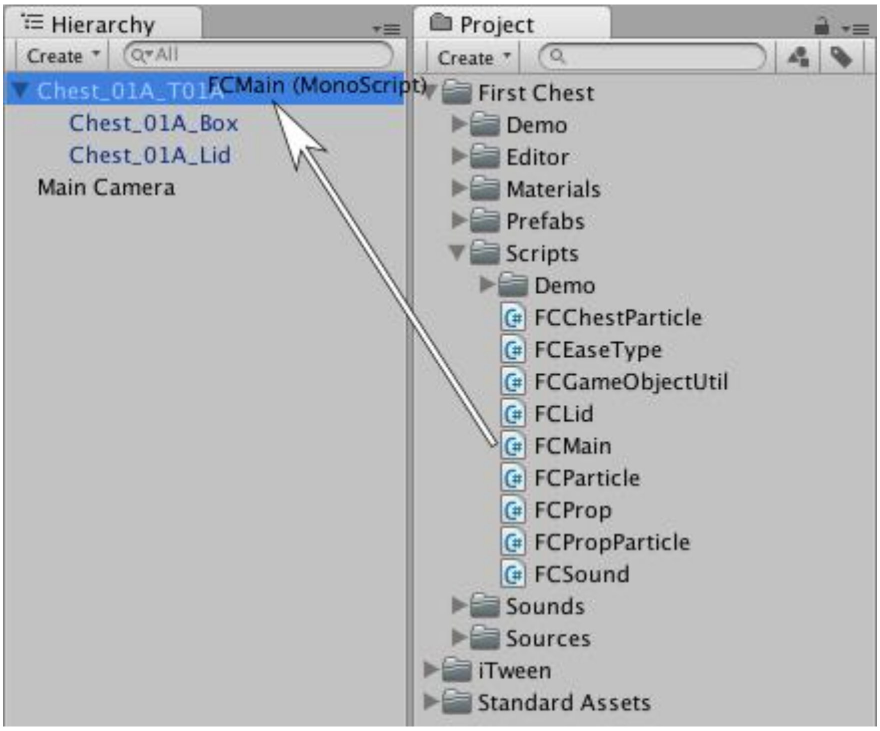
X 0 Y 35 Z 0

Scale

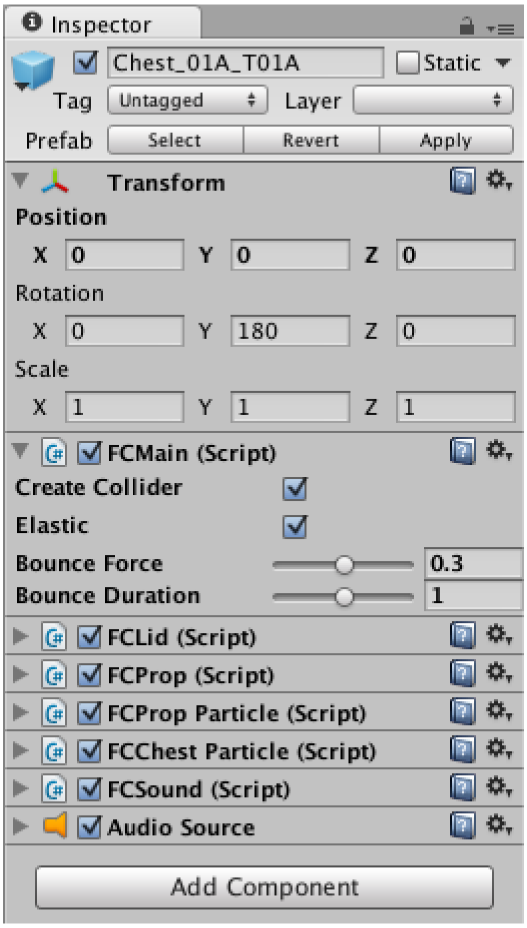
X 1 Y 1 Z 1

Main Camera Transformation.

6. Select the Chest in Hierarchy tab, drag **"FCMain"** script from **"First Chest/Scripts"** folder and drop it into the Chest.



7. FCMain and other First Chest components are added to the Chest automatically.



☑ Chest_01A_T01A □ Static

Tag Untagged Layer

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 180 Z 0

Scale X 1 Y 1 Z 1

☑ FCMain (Script)

Create Collider ☑

Elastic ☑

Bounce Force 0.3

Bounce Duration 1

☑ FCLid (Script)

☑ FCProp (Script)

☑ FCProp Particle (Script)

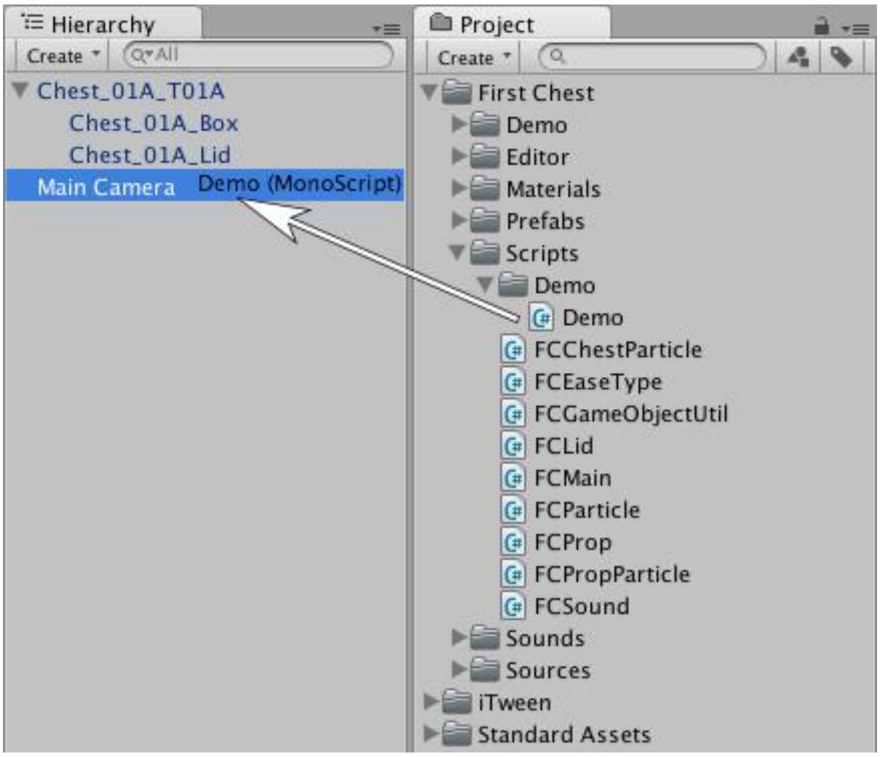
☑ FCChest Particle (Script)

☑ FCSound (Script)

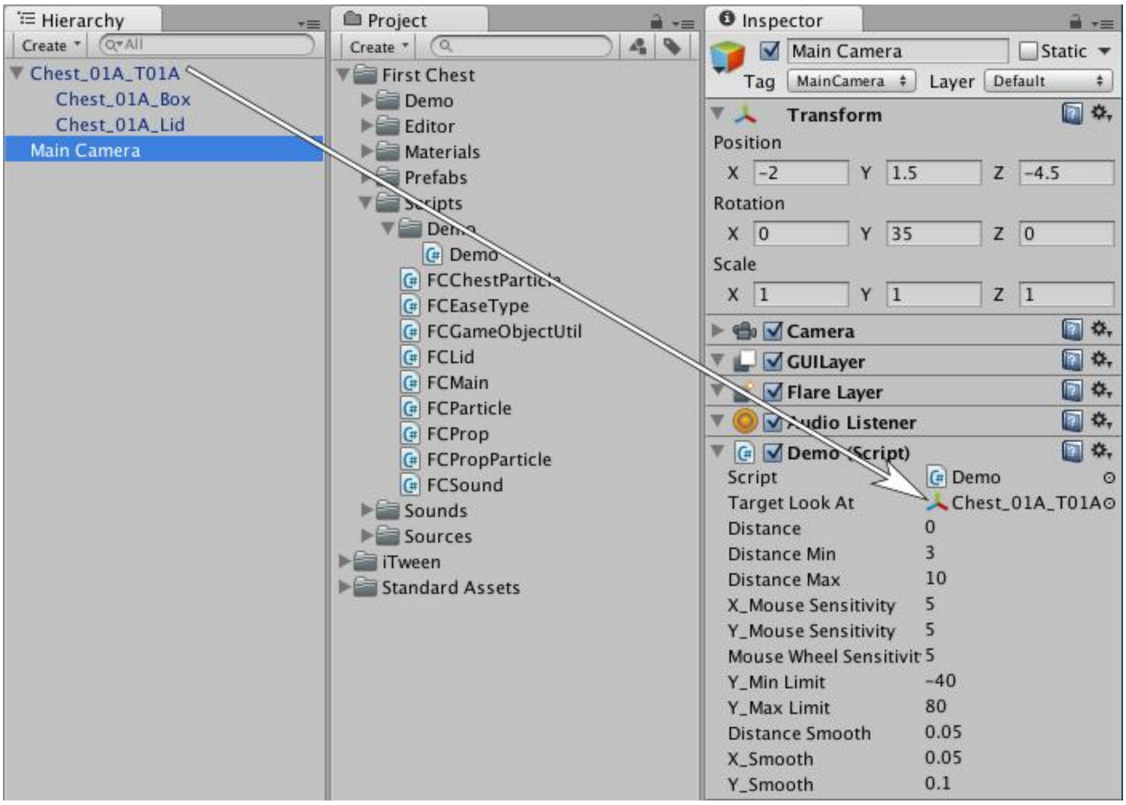
☑ Audio Source

8. Select Main Camera in Hierarchy tab.

9. Drag **"Demo"** script from **"First Chest/Scripts/Demo"** folder and drop it into the Main Camera.



10. Main Camera is selected; drag the Chest in Hierarchy tab to **“Target Look At”** property in Demo component.



☑ Main Camera ☐ Static
Tag MainCamera Layer Default

Transform
Position X -2 Y 1.5 Z -4.5
Rotation X 0 Y 35 Z 0
Scale X 1 Y 1 Z 1

☑ Camera
☑ GUILayer
☑ Flare Layer
☑ Audio Listener
☑ Demo (Script)

Script	Demo
Target Look At	Chest_01A_T01A
Distance	0
Distance Min	3
Distance Max	10
X_Mouse Sensitivity	5
Y_Mouse Sensitivity	5
Mouse Wheel Sensitivit	5
Y_Min Limit	-40
Y_Max Limit	80
Distance Smooth	0.05
X_Smooth	0.05
Y_Smooth	0.1

11. Press Play.

12. In Game tab, drag mouse around to orbit the camera. Left button to toggle open/close the Chest, Right button to toggle lock/unlock and use mouse-scroll to zoom in/out.

13. Stop and try adjusting the properties of First Chest components in Inspector tab.

2. Tweeners

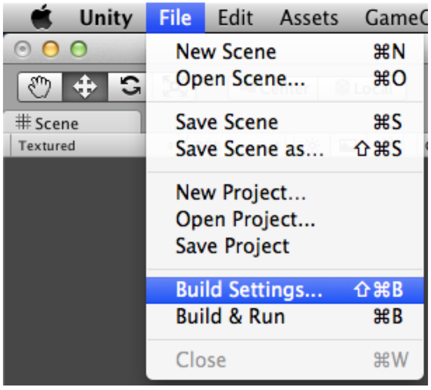
Tweener is the interpolation system that takes one value and animates it to another over a given amount of time. Three Tweeners are supported by First Chest;

- iTween: https://www.assetstore.unity3d.com/#/content/84
- HOTween https://www.assetstore.unity3d.com/#/content/3311
- LeanTween. https://www.assetstore.unity3d.com/#/content/3595

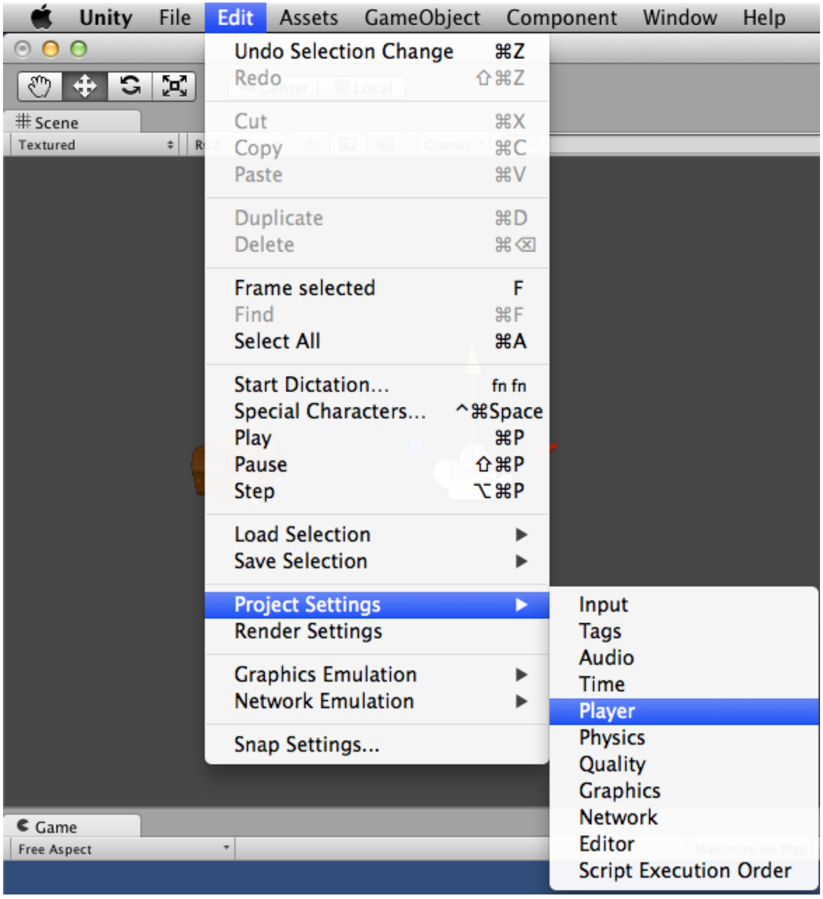
The default tweener is iTween, you can change the default to another by using preprocessor directives in **Scripting Define Symbols** in Unity **Player Configuration**.

1. Go to **Top menu->Files->Build Settings** press **Player Settings** button or Go to **Top menu->Edit->Project Settings->Player**.

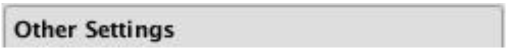
Top menu->Files->Build Settings



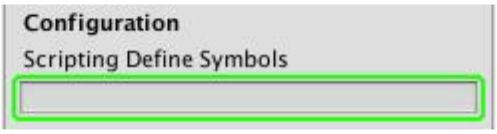
Top menu->Edit->Project Settings->Player



2. In **Inspector tab** expand **Other Settings**.



3. Look for **Configuration** and **Scripting Define Symbols** TextField.



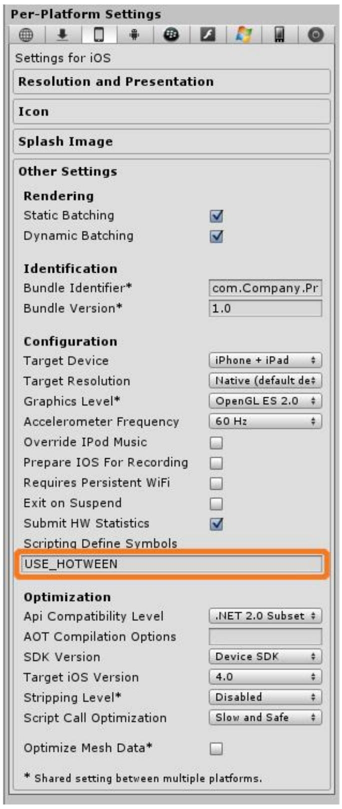
4. To use Hotween type **USE_HOTWEEN**.

5. To use LeanTween type **USE_LEANTWEEN**.

6. To use iTween, leave it blank.

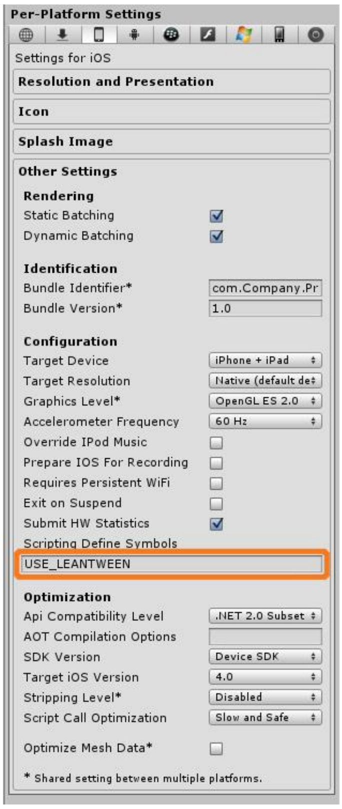
7. Refresh your Script Editor tool such as MonoDevelop, Microsoft Visual Studio if it opens your project.

HOTween



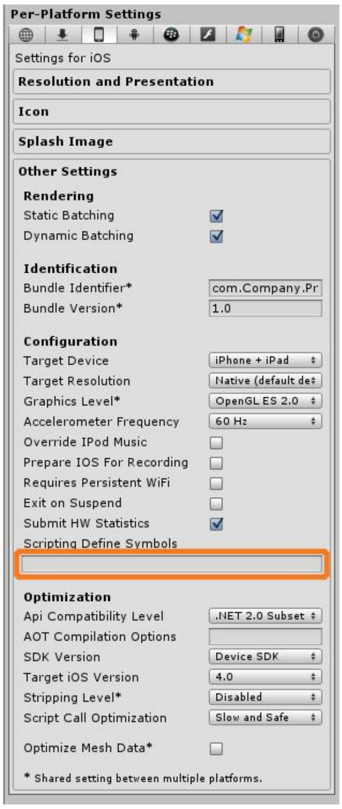
Per-Platform Settings	
Settings for iOS	
Resolution and Presentation	
Icon	
Splash Image	
Other Settings	
Rendering	
Static Batching	☑
Dynamic Batching	☑
Identification	
Bundle Identifier*	com.Company.Pr
Bundle Version*	1.0
Configuration	
Target Device	iPhone + iPad
Target Resolution	Native (default de
Graphics Level*	OpenGL ES 2.0
Accelerometer Frequency	60 Hz
Override IPod Music	☐
Prepare IOS For Recording	☐
Requires Persistent WiFi	☐
Exit on Suspend	☐
Submit HW Statistics	☑
Scripting Define Symbols	
USE_HOTWEEN	
Optimization	
Api Compatibility Level	.NET 2.0 Subset
AOT Compilation Options	
SDK Version	Device SDK
Target iOS Version	4.0
Stripping Level*	Disabled
Script Call Optimization	Slow and Safe
Optimize Mesh Data*	☐
* Shared setting between multiple platforms.	

LeanTween



Per-Platform Settings	
Settings for iOS	
Resolution and Presentation	
Icon	
Splash Image	
Other Settings	
Rendering	
Static Batching	☑
Dynamic Batching	☑
Identification	
Bundle Identifier*	com.Company.Pr
Bundle Version*	1.0
Configuration	
Target Device	iPhone + iPad
Target Resolution	Native (default de
Graphics Level*	OpenGL ES 2.0
Accelerometer Frequency	60 Hz
Override IPod Music	☐
Prepare IOS For Recording	☐
Requires Persistent WiFi	☐
Exit on Suspend	☐
Submit HW Statistics	☑
Scripting Define Symbols	
USE_LEANTWEEN	
Optimization	
Api Compatibility Level	.NET 2.0 Subset
AOT Compilation Options	
SDK Version	Device SDK
Target iOS Version	4.0
Stripping Level*	Disabled
Script Call Optimization	Slow and Safe
Optimize Mesh Data*	☐
* Shared setting between multiple platforms.	

iTween



Per-Platform Settings	
Settings for iOS	
Resolution and Presentation	
Icon	
Splash Image	
Other Settings	
Rendering	
Static Batching	☑
Dynamic Batching	☑
Identification	
Bundle Identifier*	com.Company.Pr
Bundle Version*	1.0
Configuration	
Target Device	iPhone + iPad
Target Resolution	Native (default de
Graphics Level*	OpenGL ES 2.0
Accelerometer Frequency	60 Hz
Override IPod Music	☐
Prepare IOS For Recording	☐
Requires Persistent WiFi	☐
Exit on Suspend	☐
Submit HW Statistics	☑
Scripting Define Symbols	
Optimization	
Api Compatibility Level	.NET 2.0 Subset
AOT Compilation Options	
SDK Version	Device SDK
Target iOS Version	4.0
Stripping Level*	Disabled
Script Call Optimization	Slow and Safe
Optimize Mesh Data*	☐
* Shared setting between multiple platforms.	

Learn more about tweeners.

Tweeners	URLs
HOTween	Asset Store: https://www.assetstore.unity3d.com/#/content/3311 Documentation: http://www.holoville.com/hotween/documentation.html
LeanTween	Asset Store: https://www.assetstore.unity3d.com/#/content/3595 Documentation: http://dentedpixel.com/LeanTweenDocumentation/classes/LeanTween.html
iTween	Asset Store: https://www.assetstore.unity3d.com/#/content/84 Documentation: http://itween.pixelplacement.com/documentation.php

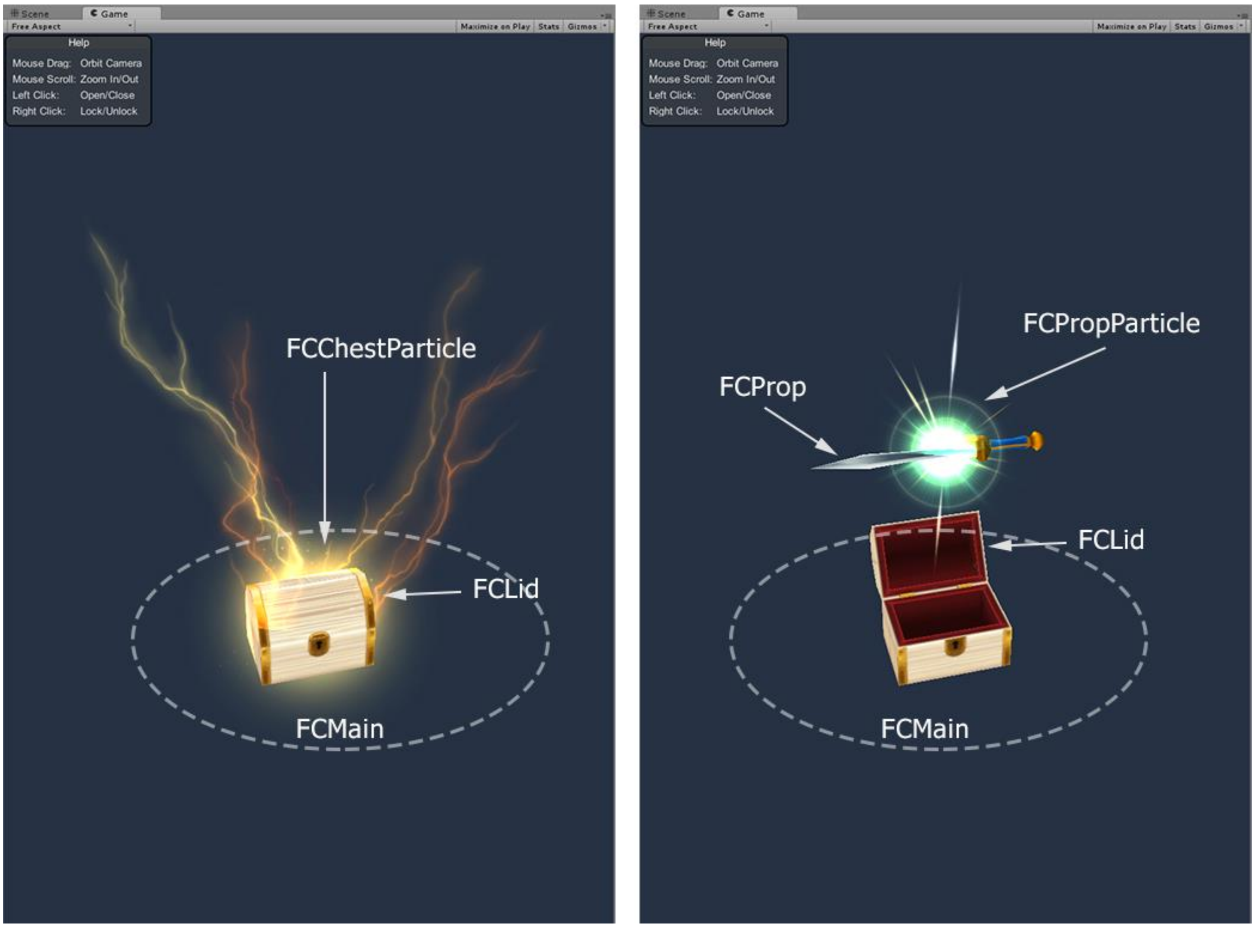
A good reference for what you can expect from each ease type.

http://www.robertpenner.com/easing/easing_demo.html

3. Components

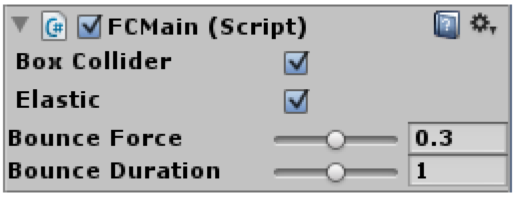
First Chest components control the behaviors of the Chest gameObject in the scene. There are many components work together for a Chest; FCLid controls the Lid, FCChestParticle controls the Particle of the Chest, FCProp controls the gameObject in the Chest, FCPropParticle controls the Particle of the gameObject in the Chest, FCSound controls Unity's Audio Source component. FCMain is main component to control others. You can adjust their variables or disable any component if you don't want it such as disable FCSound if you don't want the Chest to play sound.

This picture describe how each component handles the Chest.



3.1 FCMain (First Chest Main Component)

FCMain is the main component of First Chest. Drop FCMain script on any gameObject to make it a First Chest gameObject. Others components will be added after FCMain, automatically.



☑ FCMain (Script)

Box Collider ☑

Elastic ☑

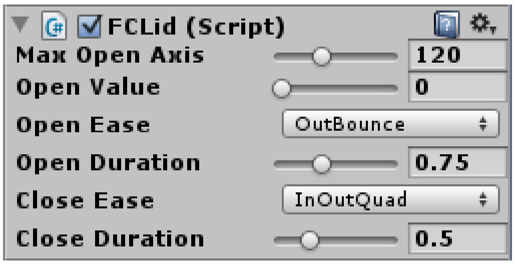
Bounce Force 0.3

Bounce Duration 1

Properties	Details
Box Collider	If enabled, the Chest will create a box collider. This makes Unity can cast a ray from camera against the Chest. This must be checked if you want the Chest can respond user input from Mouse and Touch screen.
Elastic	Enable Chest bounce animation when it turns to open/close or lock/unlock.
Bounce Force	Amount of the bouncing force.
Bounce Duration	The time in seconds the bounce animation will take to complete.

3.2 FCLid (First Chest Lid Component)

FCLid is the component that handles the Lid of the Chest. This component is added automatically when you drop FCMain component to a gameObject. You can disable it if you don't want to Open/Close and Lock/Unlock the Chest.



☑ FCLid (Script)

Max Open Axis 120

Open Value 0

Open Ease OutBounce

Open Duration 0.75

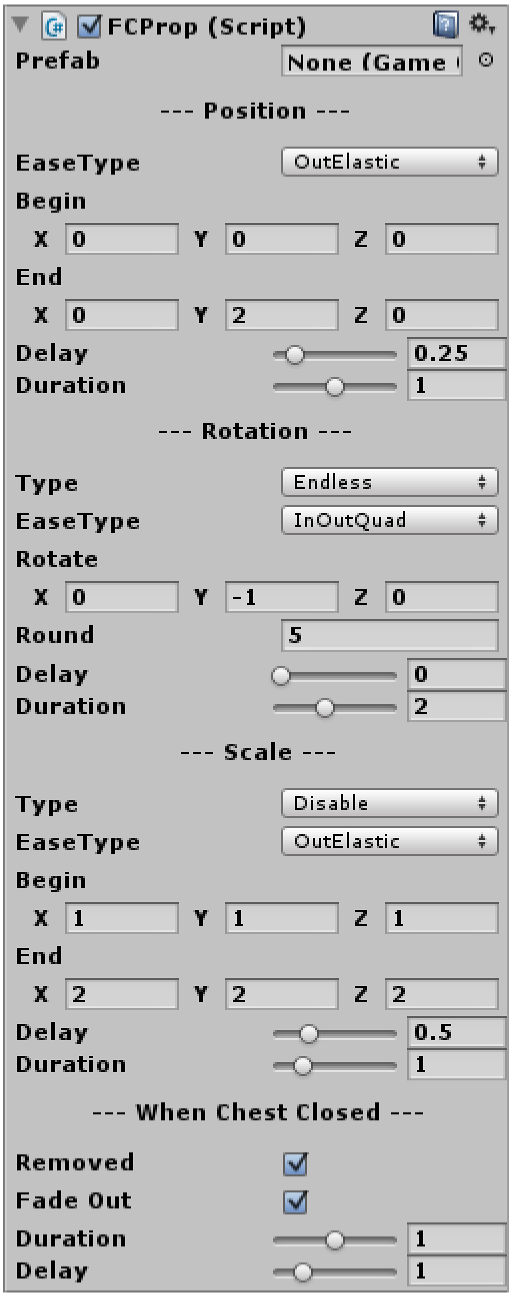
Close Ease InOutQuad

Close Duration 0.5

Properties	Details
Max Open Angle	Amount of maximum Axis in Euler Degrees when the Lid open.
Value	Amount between 0 and 1. If this value is 1 the Lid is opening in maximum angle. 0 means the Lid is closed.
Open Ease	The shape of the easing curve applied to the opening animation.
Open Duration	The time in seconds for the open animation will take to complete.
Close Ease	The shape of the easing curve applied to the closing animation.
Close Duration	The time in seconds the close animation will take to complete.

3.3 FCProp (First Chest Prop Component)

FCProp controls gameObject inside the Chest. This component is added automatically when you drop FCMain component to a gameObject. You can disable it if you want the Chest is empty and shows nothing when it opened.



☑ FCProp (Script)

Prefab: None (Game (

--- Position ---

EaseType: OutElastic

Begin: X 0, Y 0, Z 0

End: X 0, Y 2, Z 0

Delay: 0.25

Duration: 1

--- Rotation ---

Type: Endless

EaseType: InOutQuad

Rotate: X 0, Y -1, Z 0

Round: 5

Delay: 0

Duration: 2

--- Scale ---

Type: Disable

EaseType: OutElastic

Begin: X 1, Y 1, Z 1

End: X 2, Y 2, Z 2

Delay: 0.5

Duration: 1

--- When Chest Closed ---

Removed: ☑

Fade Out: ☑

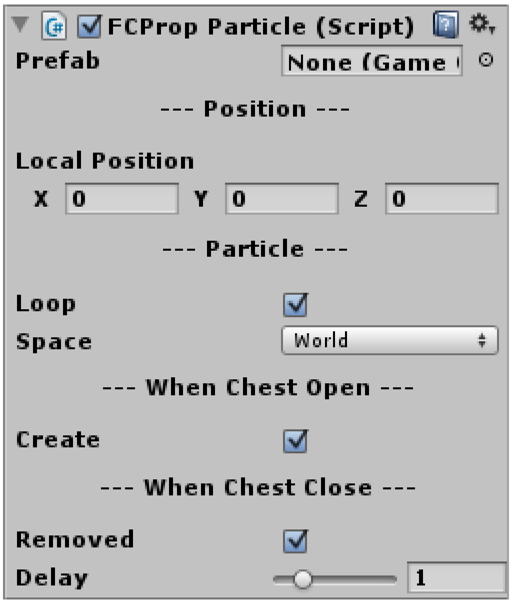
Duration: 1

Delay: 1

Properties	Details
Prefab	Reference to the prefab of the Prop to show when the Chest is opened.
Position	
- EaseType	The shape of the easing curve applied to the moving animation.
- Begin	Local position when initiated.
- End	Local position the Prop will move to.
- Delay	The time in seconds the animation will wait before begin moving.
- Duration	The time in seconds the move animation will take to complete.
Rotation	
- Type	Type of the rotation; Disable, Endless, Limited.
- EaseType	The shape of the easing curve applied to the rotation animation.
- Rotation	Direction of the rotation.
- Round	Max round when Type is set to Limited.
- Delay	The time in seconds the animation will wait before begin rotation.
- Duration/Round	The time in seconds the rotation animation will take to complete a round.
Scale	
- Type	Type of the scaling; Disable, Enable.
- EaseType	The shape of the easing curve applied to the scaling animation.
- Begin	Local scale to initiate the Prop.
- End	Local scale the Prop will scale to.
- Delay	The time in seconds the animation will wait before begin scaling.
- Duration	The time in seconds the scaling animation will take to complete.
When Chest Closed	
- Removed	Remove Prop when user closes the Lid.
- Fade out	Enable Prop fade out when user closes the Lid. Fade out needs shader which supports _Color with alpha channel.
- Duration	The time in seconds the fading will take to complete.
- Delay	The time in seconds to wait before remove the Prop.

3.4 FCPropParticle (First Chest Prop Particle Component)

FCPropParticle controls PaticleSystem of the Prop. This component is added automatically when you drop FCMain component to a gameObject. Disable this component when you don't want Particle of the Prop.



☑ FCProp Particle (Script)

Prefab: None (Game

--- Position ---

Local Position X 0 Y 0 Z 0

--- Particle ---

Loop ☑

Space: World

--- When Chest Open ---

Create ☑

--- When Chest Close ---

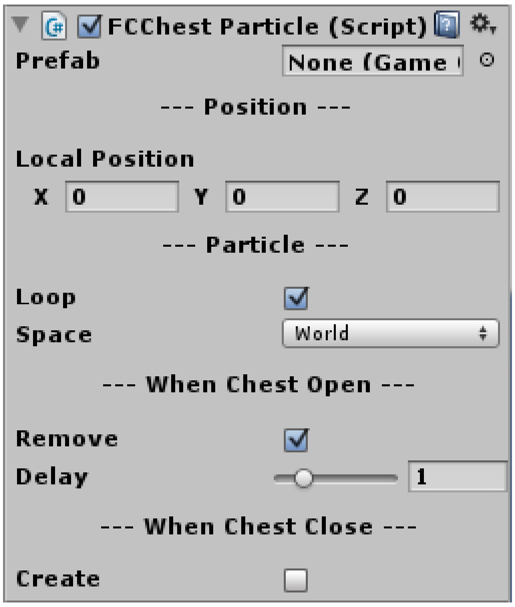
Removed ☑

Delay: 1

Properties	Details
Prefab	Reference to the prefab of the Particle.
Position	
- Local Position	Local position when initiated.
Particle	
- Loop	Enable this to make the Particle loop when it reaches the end.
- Space	Type of coordinate space in which to operate.
When Chest Open	
- Create	Initiate Particle when user opens the Chest.
When Chest Close	
- Remove	Remove Particle when user closes the Chest.
- Delay	The time in seconds to wait before remove the Particle.

3.5 First Chest Chest Particle (FCChestParticle)

FCChestParticle controls PaticleSystem of the Chest. This component is added automatically when you drop FCMain component to a gameObject. Disable this component when you don't want Particle of the Chest.



☑ FCChest Particle (Script)

Prefab: None (Game

--- Position ---

Local Position

X 0 Y 0 Z 0

--- Particle ---

Loop ☑

Space: World

--- When Chest Open ---

Remove ☑

Delay: 1

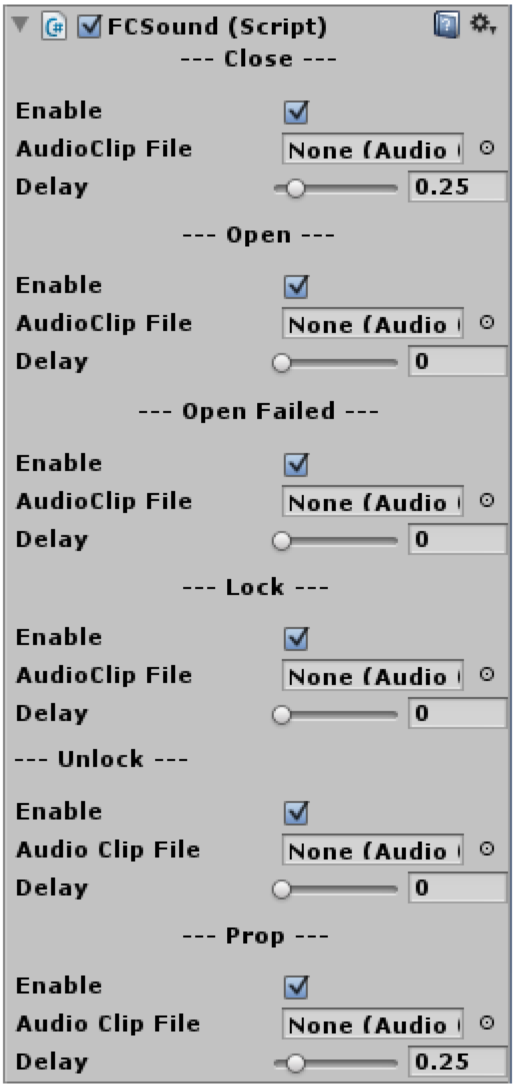
--- When Chest Close ---

Create □

Properties	Details
Prefab	Reference to the prefab of the Particle.
Position	
- Local Position	Local position when initiated.
Particle	
- Loop	Enable this to make the Particle loop when it reaches the end.
- Space	Type of coordinate space in which to operate.
When Chest Open	
- Remove	Remove Particle when user opens the Chest.
- Delay	The time in seconds to wait before remove the Particle.
When Chest Close	
- Create	Initiate Particle when user closes the Chest.

3.6 FCSound (First Chest Sound Component)

FCSound controls Audio Source. This component is added automatically when you drop FCMain component to a gameObject. You can disable it if you don't want to play any sound.



☑ FCSound (Script)

--- Close ---

Enable ☑
AudioClip File: None (Audio ⊙
Delay: 0.25

--- Open ---

Enable ☑
AudioClip File: None (Audio ⊙
Delay: 0

--- Open Failed ---

Enable ☑
AudioClip File: None (Audio ⊙
Delay: 0

--- Lock ---

Enable ☑
AudioClip File: None (Audio ⊙
Delay: 0

--- Unlock ---

Enable ☑
Audio Clip File: None (Audio ⊙
Delay: 0

--- Prop ---

Enable ☑
Audio Clip File: None (Audio ⊙
Delay: 0.25

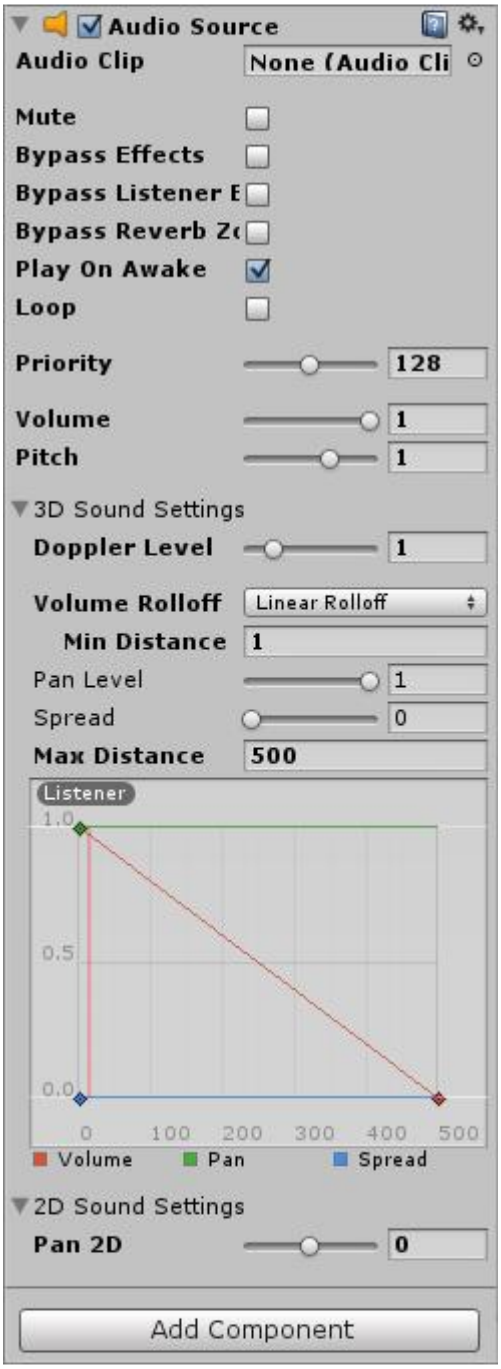
Properties	Details
Close	
- Enable	Enable this to make Close sound play.
- AudioClip	Reference to the Audio Clip.
- Delay	The time in seconds to delay before play sound.
Open	
- Enable	Enable this to make Open sound play.
- AudioClip	Reference to the Audio Clip.
- Delay	The time in seconds to delay before play sound.
Open Failed	
- Enable	Enable this to make Open Failed sound play.
- AudioClip	Reference to the Audio Clip.
- Delay	The time in seconds to delay before play sound.
Lock	
- Enable	Enable this to make Lock sound play.
- AudioClip	Reference to the Audio Clip.
- Delay	The time in seconds to delay before play sound.
Unlock	
- Enable	Enable this to make Unlock sound play.
- AudioClip	Reference to the Audio Clip.
- Delay	The time in seconds to delay before play sound.
Prop	
- Enable	Enable this to make Prop sound play.
- AudioClip	Reference to the Audio Clip.
- Delay	The time in seconds to delay before play sound.

3.7 Audio Source

First Chest uses Unity Audio Source to play sounds. This component is added automatically when you drop FCMain component to a gameObject.

(Unity Audio Source: http://docs.unity3d.com/Documentation/Components/class-AudioSource.html)

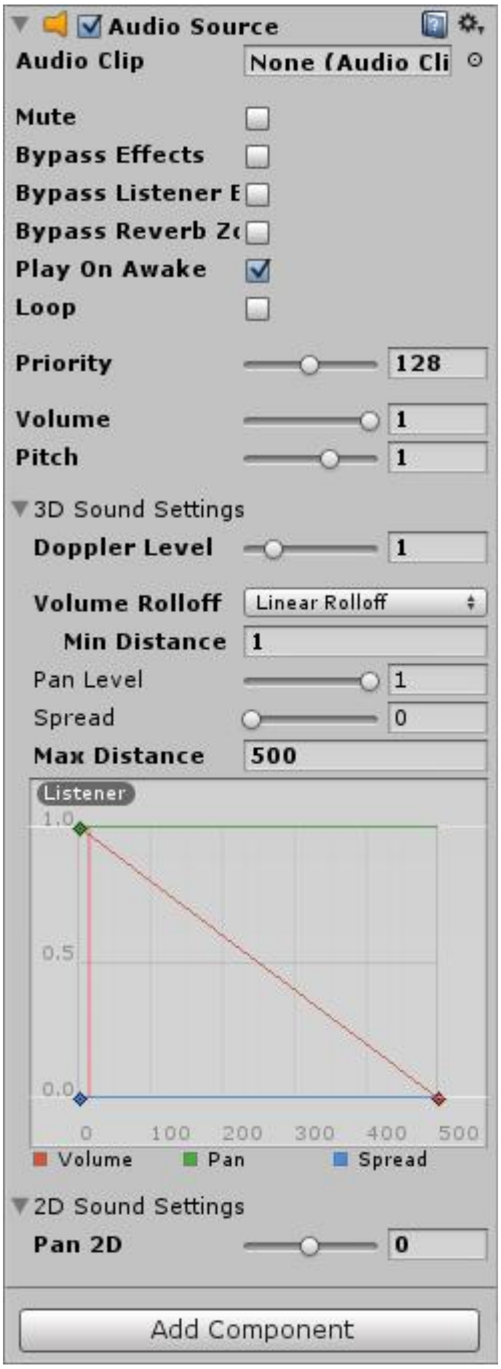
The Audio Source plays back an Audio Clip in the scene. If the Audio Clip is a 3D clip, the source is played back at a given position and will attenuate over distance. The audio can be spread out between speakers (stereo to 7.1) (Spread) and morphed between 3D and 2D (PanLevel). This can be controlled over distance with falloff curves. Also, if the listener is within one or multiple Reverb Zones, reverberations is applied to the source. (PRO only) Individual filters can be applied to each audio source for an even richer audio experience. See Audio Effects for more details.



Audio Source

Audio Clip	None (Audio Cli
Mute	□
Bypass Effects	□
Bypass Listener E	□
Bypass Reverb Z(	□
Play On Awake	☑
Loop	□
Priority	128
Volume	1
Pitch	1
3D Sound Settings	
Doppler Level	1
Volume Rolloff	Linear Rolloff
Min Distance	1
Pan Level	1
Spread	0
Max Distance	500
2D Sound Settings	
Pan 2D	0

Properties	*Details*
Audio Clip	*Reference to the sound clip file that will be played.*
Mute	*If enabled the sound will be playing but muted.*
Bypass Effects	*This Is to quickly "by-pass" filter effects applied to the audio source. An easy way to turn all effects on/off.*
Play On Awake	*If enabled, the sound will start playing the moment the scene launches. If disabled, you need to start it using the **Play()** command from scripting.*
Loop	*Enable this to make the **Audio Clip** loop when it reaches the end.*
Priority	*Determines the priority of this audio source among all the ones that coexist in the scene. (Priority: 0 = most important. 256 = least important. Default = 128.). Use 0 for music tracks to avoid it getting occasionally swapped out.*
Volume	*How loud the sound is at a distance of one world unit (one meter) from the **Audio Listener**.*
Pitch	*Amount of change in pitch due to slowdown/speed up of the **Audio Clip**. Value 1 is normal playback speed.*
3D Sound Settings	*Settings that are applied to the audio source if the Audio Clip is a 3D Sound.*
Pan Level	*Sets how much the 3d engine has an effect on the audio source.*
Spread	*Sets the spread angle to 3d stereo or multichannel sound in speaker space.*
Doppler Level	*Determines how much doppler effect will be applied to this audio source (if is set to 0, then no effect is applied).*



Audio Source

Audio Clip None (Audio Cli

Mute □

Bypass Effects □

Bypass Listener E □

Bypass Reverb Z(□

Play On Awake ☑

Loop □

Priority 128

Volume 1

Pitch 1

3D Sound Settings

Doppler Level 1

Volume Rolloff Linear Rolloff

Min Distance 1

Pan Level 1

Spread 0

Max Distance 500

2D Sound Settings

Pan 2D 0

Min Distance	*Within the MinDistance, the sound will stay at loudest possible. Outside MinDistance it will begin to attenuate. Increase the MinDistance of a sound to make it 'louder' in a 3d world, and decrease it to make it 'quieter' in a 3d world.*
Max Distance	*The distance where the sound stops attenuating at. Beyond this point it will stay at the volume it would be at MaxDistance units from the listener and will not attenuate any more.*
Rolloff Mode	*How fast the sound fades. The higher the value, the closer the Listener has to be before hearing the sound.(This is determined by a Graph).*
Logarithmic Rolloff	*The sound is loud when you are close to the audio source, but when you get away from the object it decreases significantly fast.*
Linear Rolloff	*The further away from the audio source you go, the less you can hear it.*
Custom Rolloff	*The sound from the audio source behaves accordingly to how you set the graph of roll offs.*
2D Sound Settings	*Settings that are applied to the audio source if the Audio clip is a 2D Sound.*
Pan 2D	*Sets how much the engine has an effect on the audio source.*

4. Script Reference

This section of the documentation contains details of the scripting API that First Chest provides. All scripts are written in c#.

4.1 FCMain Class Reference

Description

Script interface for main component of First Chest gameObject. Call FCMain public functions to control the Chest.

Public Variables

public bool	**m_CreateBoxCollider** = true If enabled, the Chest will create a box collider. This makes Unity can cast a ray from camera against the Chest. It must be true if you want the Chest can respond user input from Mouse and Touch screen.
public bool	**m_Elastic** = true If true, Chest will play bounce animation when it turns to open, close, lock and unlock.
public float	**m_BounceForce** = 0.3f Amount of the bouncing force.
public float	**m_BounceDuration** = 1.0f The time in seconds the bounce animation will take to complete.

Public Functions

public void	**ToggleOpen**() Toggle Open/Close the Lid. Open failed if the Lid is locked.
public void	**ToggleLock**() Toggle Lock/Unlock the Lid.
public void	**Open**() Open the Chest. Open is failed if the Lid is locked.
public void	**Close**() Close the Chest.
public bool	**IsLocked**() Returns true, if Lid is locked.
public void	**Lock**() Lock the Chest.
public void	**Unlock**() Unlock the Chest.
public FCLid	**getLid**() Returns FCLid, null if FCLid is disable.
public FCProp	**getProp**() Returns FCProp, null if FCProp is disable.
public FCPropParticle	**getPropParticle**() Returns FCPropParticle, null if FCPropParticle is disable.

public FCChestParticle	**getChestParticle**() Returns FCChestParticle, null if FCChestParticle is disable.
public FCSound	**getSound**() Returns FCSound, null if FCSound is disable.
public bool	**IsOpened**() Is the Chest opened?
public bool	**IsClosed**() Is the Chest closed?
public FCLid.eState	**LidStatus**() Returns state of FCLid. {FCLid.eState.Open, FCLid.eState.Close}

4.2 FCLid Class Reference

Description

Script interface for the Lid gameObject.

Public Types

public enum eState
{
Open,
Close
};
	Enumeration of the type of Lid state.
	Open = Lid is open.
	Close = Lid is close.

public enum eLockStatus
{
Locked,
Free
};
	Enumeration of the type of Lock state.
	Locked = Chest is locked.
	Free = Chest is free and can be opened.

Public Variables

public float	**m_MaxOpenAngle** = 120.0f Amount of maximum Axis in Euler Degrees when the Lid open.
public float	**m_OpenValue** Amount between 0 and 1. If this value is 1 the Lid is opening in maximum angle. 0 means the Lid is closed.
public FCEaseType.eEaseType	**m_OpenEaseType** = FCEaseType.eEaseType.OutBounce The shape of the easing curve applied to the opening animation.
public float	**m_OpenDuration** = 0.75f The time in seconds for the open animation will take to complete.
public FCEaseType.eEaseType	**m_CloseEaseType** = FCEaseType.eEaseType.InOutQuad The shape of the easing curve applied to the closing animation.
public float	**m_CloseDuration** = 0.5f The time in seconds the close animation will take to complete.

Public Functions

public void	**Open**() Open the Lid.
public void	**Open**(float time) Open the Lid with a specified time duration in seconds.
public void	**Close**() Close the Lid.
public void	**Close**(float time) Close the Lid with a time specified in seconds.
public void	**Close**(eLockStatus LockStatus) Close the Lid and change Lock status.
public void	**Close**(float time, eLockStatus LockStatus) Close the Lid with a specified time duration in seconds and update Lock status.
public void	**Lock**() Lock the Lid.
public void	**Unlock**() Unlock the Lid.
public void	**SetLock**(eLockStatus LockStatus) Change Lock status of the Lid. {eLockStatus.Lock, eLockStatus .Free}
public bool	**IsOpened**() Is the Lid opened?
public bool	**IsOpeningOrClosing**() Returns true if Lid is playing Open/Close animation.
public eState	**getLidStatus**() Returns state of FCLid. {eState.Open, eState.Close}
public bool	**IsLocked**() Returns true, if Lid is locked.
public void	**FindLid**() Seek in children objects for a gameObject which its name contains string in m_LidString and keep it in m_Lid variable. (The Lid gameObject’s name has to contain string in m_LidString.)

4.3 FCProp Class Reference

Description

Script interface for a gameObject inside the Chest. This gameObject will be shown when the Chest is opened.

Public Types

public enum eTransformState
{
Begin,
Changing,
End
};
	Enumeration of the type of transform state to use with Prop.
	Begin = Prop begins transform.
	Changing = Prop position/rotation/scale are transforming.
	End = Prop transforms are finished.

public enum eRotationType
{
Disable,
Endless,
LimitedRound
};
	Enumeration of the type of rotation to use with Prop rotation.
	Disable = Disable rotation.
	Endless = Rotates Prop around itself, forever.
	LimitedRound = Rotate Prop until round count is equal to **m_MaxRotationRound.**

public enum eScaleType
{
Disable,
Enable
};
	Enumeration of the type of scale to use with Prop scaling.
	Disable = disable Prop scaling.
	Enable = enable Prop scaling.

Public Variables

public GameObject	**m_Prefab** Reference to the prefab of the Prop to show when the Chest is opened.
public FCEaseType.eEaseType	**m_PosEaseType** = FCEaseType.eEaseType.OutElastic The shape of the easing curve applied to the moving animation.
public Vector3	**m_PosBegin** = new Vector3(0, 0, 0) Local position when initiated.
public Vector3	**m_PosEnd** = new Vector3(0, 2.0f, 0) Local position the Prop will move to.
public float	**m_PosValue** = 0.0f Amount between 0 and 1. 0 = The Prop is at m_PosBegin position. 1 = The Prop is at m_PosEnd position.
public float	**m_PosDelay** = 0.25f The time in seconds the animation will wait before begin moving.
public float	**m_PosDuration** = 1.0f The time in seconds the move animation will take to complete.
public eRotationType	**m_RotationType** = eRotationType.Endless Type of the rotation; Disable, Endless, Limited.
public FCEaseType.eEaseType	**m_RotationEaseType** = FCEaseType.eEaseType.InOutQuad The shape of the easing curve applied to the rotation animation.
public Vector3	**m_Rotation** = new Vector3(0, -1, 0)

	Direction of the rotation.
public float	**m_RotationValue** = 0.0f Amount between 0 and 1. 1 is 360 Euler Degrees.
public int	**m_MaxRotationRound** = 5 Max round when Type is set to Limited.
public float	**m_RotationDelay** = 0 The time in seconds the animation will wait before begin rotation.
public float	**m_RotationDurationPerRound** = 2.0f The time in seconds the rotation animation will take to complete a round.
public eScaleType	**m_ScaleType** = eScaleType.Disable Type of the scaling; Disable or Enable.
public FCEaseType.eEaseType	**m_ScaleEaseType** = FCEaseType.eEaseType.OutElastic The shape of the easing curve applied to the scaling animation.
public Vector3	**m_ScaleBegin** = new Vector3(1, 1, 1) Local scale to initiate the Prop.
public Vector3	**m_ScaleEnd** = new Vector3(2, 2, 2) Local scale the Prop will scale to.
public float	**m_ScaleValue** = 0.0f Amount between 0 and 1. 0 = The Prop is same scale with **m_ScaleBegin**. 1 = The Prop is scale to **m_ScaleEnd**.
public float	**m_ScaleDelay** = 0.5f The time in seconds the animation will wait before begin scaling.
public float	**m_ScaleDuration** = 1.0f The time in seconds the scaling animation will take to complete.
public bool	**m_RemovedWhenChestClose** = true If true, Prop will be removed when user closes the Lid.
public bool	**m_FadeOut** = true If true, Prop will be fade when user closes the Lid. Fade out needs shader which supports _Color with alpha channel.
public float	**m_FadeOutDuration** = 1.0f The time in seconds the fading will take to complete.
public float	**m_RemoveDelay** = 1.0f The time in seconds to wait before remove the Prop.

Public Functions

public void	**CreateProp**() Create a new Prop.
public void	**Remove**() Remove the Prop.
public void	**Remove**(float Delay) Remove the Prop with a delay specified in seconds.
public void	**Show**() Show the Prop. This function works only when Chest is opened.
public void	**Show**(float PosDelay, float RotationDelay, float ScaleDelay) Show the Prop with delays specified in seconds of Position, Rotation and Scaling.
public bool	**isShowing**() Returns true, if Prop is showing.
public GameObject	**getPrefab**() Return **m_Prefab**.
public GameObject	**getPropGameObject**() Return a gameObject reference to the Prop.

4.4 FCParticle Class Reference

Description

Script interface for First Chest Particle. FCParticle is base class of FCPropParticle and FCChestParticle.

Public Variables

public GameObject	**m_Prefab** Reference to the prefab of the Particle.
public Vector3	**m_OffSetLocalPosition** Local position when initiated.
public bool	**m_isLoop = true** True to loop Particle when it reaches the end.
public ParticleSystemSimulationSpace	**m_SimulationSpace = ParticleSystemSimulationSpace.World** Type of coordinate space in which to operate.
public float	**m_RemoveDelay = 1.0f** The time in seconds to wait before remove the Particle.

Public Functions

public void	**CreateParticle**() Create new Particle from **m_Prefab**.
public void	**Remove**() Remove the Particle.
public void	**Remove**(float Delay) Remove the Particle with a delay specified in seconds.
public void	**UpdateParticlePosition**() Move Particle follow to the parent’s position.
public GameObject	**getPrefab**() Returns **m_Prefab** of Particle.
public GameObject	**getParticleGameObject**() Returns Particle gameObject.
public FCMain	**getMain**() Returns **FCMain**.
public int	**getCounter**() Returns a number for checking how many times Particle was created.
public void	**setRemoveDelay** (float RemoveDelay) Set delay specified in seconds for using when remove this Particle; this Function is called by **Remove**().

4.5 FCPropParticle Class Reference

Description

Script interface for Prop Particle gameObject.

Public Variables

public bool	**m_CreateWhenChestOpen = true** If true, Particle will be initiated when user opens the Chest.
public bool	**m_RemovedWhenChestClose = true** If true, Particle will be removed when user closes the Chest.

4.6 FCChestParticle Class Reference

Description

Script interface for Chest Particle gameObject.

Public Variables

public bool	**m_RemoveWhenChestOpen = true** If true, Particle will be removed when user opens the Chest.
public bool	**m_CreateWhenChestClose = false** If true, Particle will be created when user closes the Chest.

4.7 FCSound Class Reference

Description

Script interface for First Chest Sound. This class needs AudioSource to work properly.

Public Variables

public bool	**m_EnableSoundClose** = true If true, Close sound is enabled.
public AudioClip	**m_SoundClose** Reference to the Audio Clip of Close sound.
public float	**m_SoundCloseDelay** = 0.25f The time in seconds to delay before play Close sound.
public bool	**m_EnableSoundOpen** = true If true, Open sound is enabled.
public AudioClip	**m_SoundOpen** Reference to an Audio Clip of Open sound.
public float	**m_SoundOpenDelay** = 0.0f The time in seconds to delay before play Open sound.
public bool	**m_EnableSoundOpenFailed** = true If true, Open Failed sound is enabled.
public AudioClip	**m_SoundOpenFailed** Reference to the Audio Clip of OpenFailed sound.
public float	**m_SoundOpenFailedDelay** = 0.0f The time in seconds to delay before play OpenFailed sound.
public bool	**m_EnableSoundSetLock** = true If true, Lock sound is enabled.
public AudioClip	**m_SoundSetLock** Reference to the Audio Clip of Lock sound.
public float	**m_SoundSetLockDelay** = 0.0f The time in seconds to delay before play Lock sound.
public bool	**m_EnableSoundSetUnlock** = true If true, Unlock sound is enabled.
public AudioClip	**m_SoundSetUnlock** Reference to the Audio Clip of Unlock sound.
public float	**m_SoundSetUnlockDelay** = 0.0f The time in seconds to delay before play Unlock sound.
public bool	**m_EnableSoundProp** = true If true, Prop sound is enabled.
public AudioClip	**m_SoundProp** Reference to the Audio Clip of Prop sound.
public float	**m_SoundPropDelay** = 0.25f The time in seconds to delay before play Prop sound.

Public Functions

public void	**CreateAudioSource**() Create Unity AudioSounce component.
public void	**PlaySoundClose**() Play Close sound.
public void	**PlaySoundOpen**() Play Open sound.
public void	**PlaySoundOpenFailed**() Play OpenFailed sound.
public void	**PlaySoundSetLock**() Play Lock sound.
public void	**PlaySoundSetUnlock**() Play Unlock sound.
public void	**PlaySoundProp**() Play PropShow sound.
public void	**StopSoundProp**() Stop Prop sound.
public void	**StopSoundProp**(float Delay) Stop Prop sound with a delay specified in seconds. Used in PlaySoundClose() function.

4.8 FCGameObjectUtil Class Reference

Description

Script interface for a First Chest GameObjectUtil. This class contains utilities for using in other First Chest classes.

Public Variables

public Vector3	**m_rotation** The rotation of gameObject.
public bool	**m_isRotate** = false If true, gameObject is rotating.
public float	**m_RemoveDelay** = 1.0f The time in seconds to wait before remove gameObject.
public bool	**m_RemoveFadeout** = false If true, the gameObject will be fade out before it is removed.
public float	**m_DurationCount** = 1.0f The count down time in seconds the fading will take to complete.
public float	**m_FadeOutDuration** = 1.0f The time in seconds the fading will take to complete.
public float	**m_AlphaFadeValue** = 1.0f Amount of alpha color.

Public Functions

public void	**InitRotation**(Vector3 rotation) Sets the rotation of gameObject.
public void	**StartRotation**()

	Start rotation.
public void	**StopRotation**() Stop rotation.
public void	**SelfRemoveGameObject**(float RemoveDelay, bool RemoveFadeout, float FadeOutDuration) Remove gameObject with a delay specified in seconds. If RemoveFadeout is true, Fade out gameObject with time specified in seconds while delay is counting. Fade out needs shader which supports _Color with alpha channel.
public void	**SelfRemoveParticle**(float RemoveDelay, float ParticleLifeTime) Remove ParticleSystem with a delay specified in seconds. Specified delay will be added with Particle life time.

4.9 FCEaseType Class Reference

Description

Script interface for the easetype to use with HOTween, LeanTween and iTween.

Public Types

public enum **eEaseType**{
InQuad, OutQuad, InOutQuad,
InCubic, OutCubic, InOutCubic,
InQuart, OutQuart, InOutQuart,
InQuint, OutQuint, InOutQuint,
InSine, OutSine, InOutSine,
InExpo, OutExpo, InOutExpo,
InCirc, OutCirc, InOutCirc,
linear,
spring,
InBounce, OutBounce, InOutBounce,
InBack, OutBack, InOutBack,
InElastic, OutElastic, InOutElastic
}
	An enumeration of the type of easing to use with tweeners.
	Reference for each ease type:
	http://www.robertpenner.com/easing/easing_demo.html

Static public Functions

public static Holoville.HOTween.EaseType	**EaseTypeConvert(eEaseType easeType**) Convert eEaseType to HOTween EaseType.
public static LeanTweenType	**EaseTypeConvert(eEaseType easeType**) Convert eEaseType to LeanTween EaseType.
public static string	**EaseTypeConvert(eEaseType easeType**) Convert eEaseType to iTween EaseType string.