

Escape Development Kit



version 1.0

Contents

Introduction	3
Escape Player	4
Inventory System	5
Doors (room, furniture, safe, etc.)	6
Items settings for moving and rotation	8
Inventory items (elements)	9
On/Off for objects and animations	10
Hits settings	11
Passwords settings:	12
Password-image	12
Password-Padlock	13
Password-panel	14
Libra Example	15
Object replacement	16
List of Events	17
Tips	18
Switchover to next scene	20
Start Scene	21
Contact	22

Introduction

Escape Development Kit – This is a simple solution for creating a game in the genre Escape-Room.

Escape Development Kit – a simple to use a set Unity3d packages to create Escape3D games in Unity3D Framework.

If you really think about the beginning of creating your Escape game and do not know how to start, this module will help you create Room Escape game from scratch.

With this module, you may not even know the programming, but in a short time make a full game and publish it in stores such as the App Store, Play Market, Windows store, etc.

Includes all the necessary features to create a game:

- User Friendly
- Documented Asset
- Mobile Friendly
- Contains a large number of ready to use examples
- 40+ C# scripts
- All prefabs from Folder Prefabs are ready to use without modifications

Escape Player

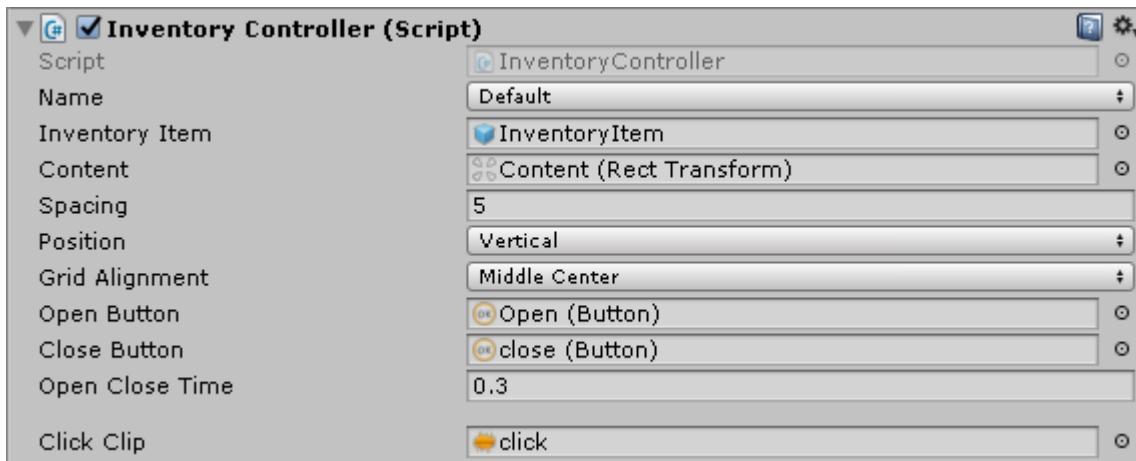
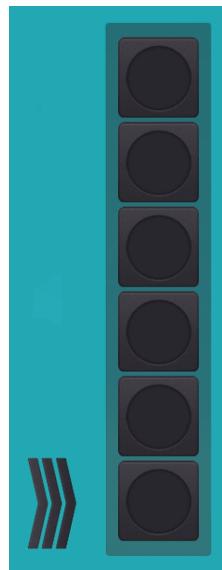
At the beginning, you have to add EscapePlayer prefab to the scene. It allows control and interact with game objects in the scene. Escape player contains Camera, so if you have already any camera in the scene just delete that camera.

EscapePlayer allows to move between set points using WalkPosition prefab.

Inventory System

Inventory System is created for saving game items using native Unity3D UI. Add Canvas before use Inventory System in the scene. (more about using Canvas is here <https://docs.unity3d.com/Manual/UICanvas.html>)

Then just add Inventory prefab in the Canvas. If needed you can add up to 3 Inventory Systems.



Settings:

- **Name** – just ignore it if you use one Inventory in the scene. Choose one name to work with. Each Inventory should have unique name.
- **Position** – Inventory could be vertical (left/right) or horizontal (top/bottom).
- Change width and height of the Inventory just expand InventoryItem.
- Number of visible items will be set automatically.

Doors (room, furniture, safe, etc.)

Using Door Element you can setup any type of doors in the scene. For example: room doors, cabinet doors, safe door, sewer hatch, etc. Also there could be added any type lockers like key, card, code, etc.

We have prepared a list of ready to use prefabs to speed up development process:

List of Door prefabs:

- Door_Rotate
- Door_Up
- Door_Down
- Door_Slide
- Door_Open
- Door_Close
- Door_Double
- Door_Move_Up

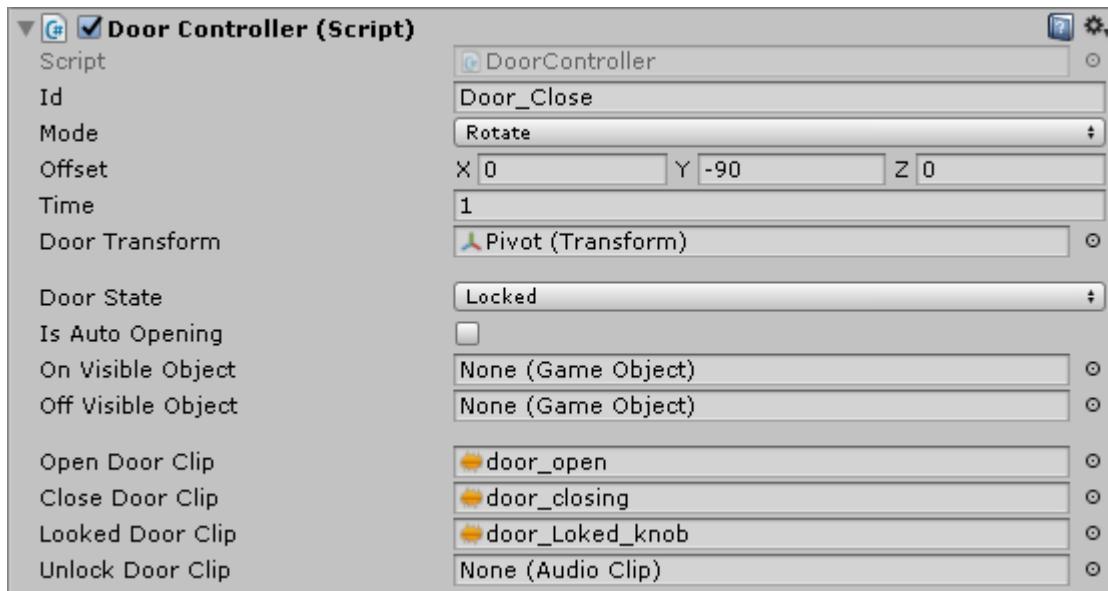
List of Lockers:

- Door_Card
- Door_Key
- Door_Password
- Door_Chain

List of different type of safes:

- Safe A
- Safe C
- Safe D
- Safe B
- Safe E

Choose prefab, set unique ID and that is it. Using these list of ready to use door prefabs you can modify or create much more different type of doors.



Settings:

Mode – have to parameters Rotate i Move:

- **Rotate** – rotate an object around pivot
- **Move** – move object

Offset – vector parameters

Time – time on action

Door Transform – object transform

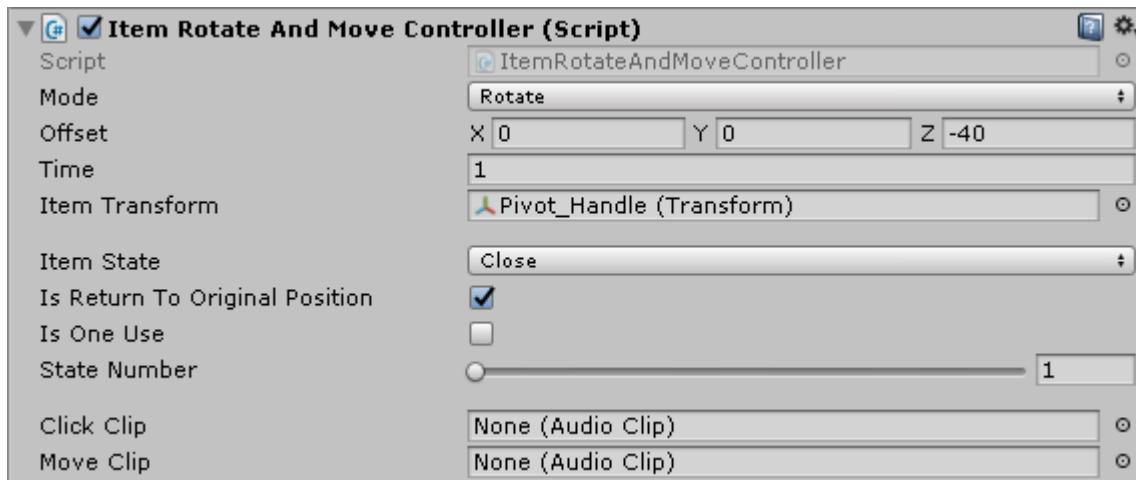
Is Auto Opening – automatic door opening after lock is on.

On Visible Object/Off Visible Object – visible or invisible object after unlock (use null if not needed)

Setup Objects for moving or rotation

Use ItemRotateAndMoveController to move and rotate objects in scene.

Examples: Water_Closet, Towel, Rug, Bucket_A, Bucket_B, door handles, ect.



Settings:

Mode – two option - Rotate i Move:

- Rotate – rotate object
- Move – move object

Offset – vector value of rotation/move

Time – time on action

Item Transform – object transform

Is Return To Original Position – set object in on/off position, for example for door handle.

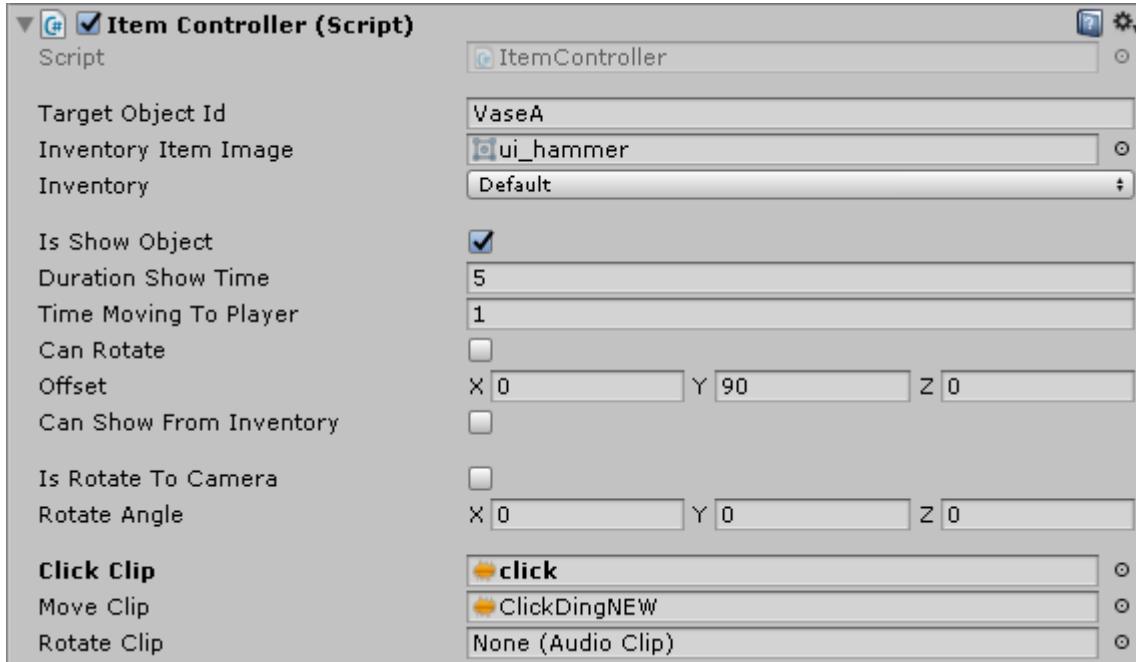
State Number – number of object states for move/rotate. Example prefab Combination.

Inventory Items (elements)

Inventory Items – this is objects in scene which could be placed in Inventory system to use later in quest.

Examples:

Door_Chain, Safe_B (Key), Door_Key (Key_Pivot), 1kg, 2kg, 5kg, TV (Pivot)



Settings:

Target Object Id – element ID which should be used to interact with this item.

Item Image – icon of the item for Inventory.

Inventory – name of Inventory which will be used to place items (Default if one Inventory system).

Is Show Object – use it if need to show item in camera before use it in the game.

Can Rotate – use it to allow rotate visible 3D.

Offset – vector to rotate item.

Can Show From Inventory – if need to shot the item again in Inventory.

On/Off Objects and Animations

Objects which could be On/Off for animation.

Examples: Washbasin, Lamps(folder), Animation(folder)

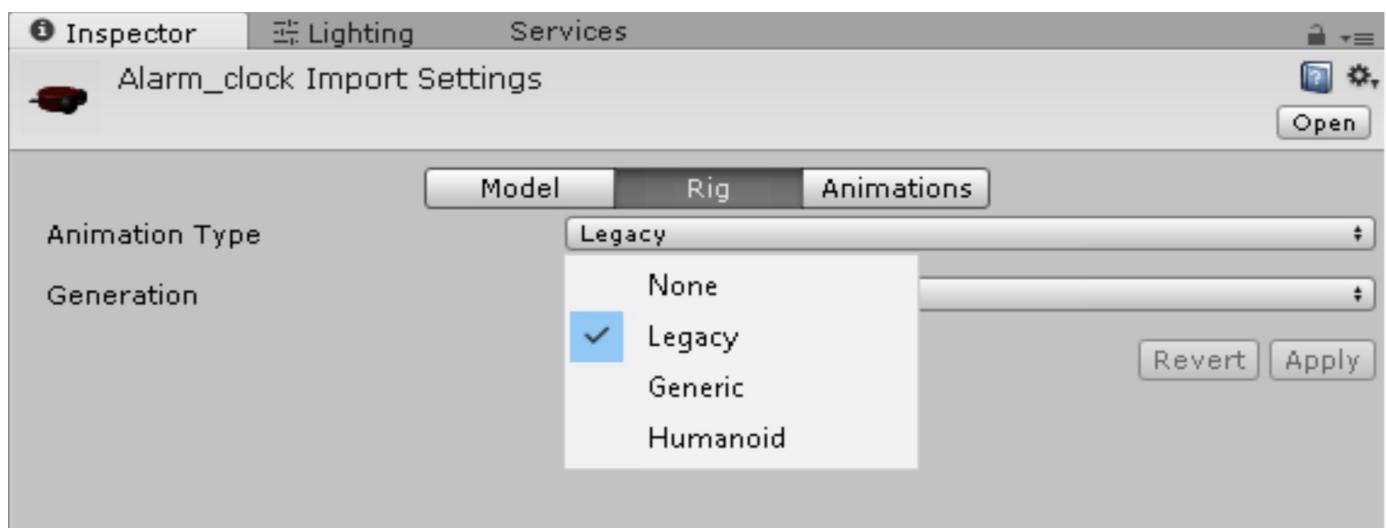


Settings:

On Off Game Object – object for On/Off could be null.

Animation – animation element could be null.

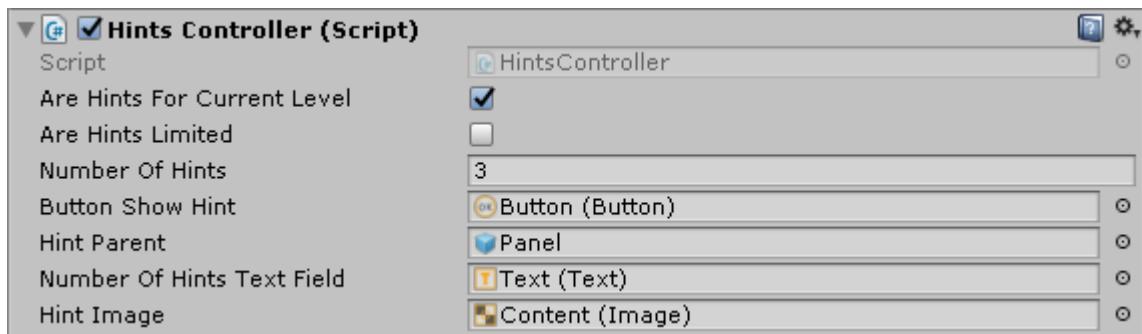
Using FBX files with animation, you should change Animation Type on Legasy.



Hits Settings

To setup hints in scene:

1. Add HintsObject prefab in the.
2. Add an image to Resource Folder. Name of the image should be the same as the scene name (or name of level if the hints are only for one level).



Settings:

Is Hints For Current Level – when hints are only for this.

Are Hints Limited – on/off hints limits.

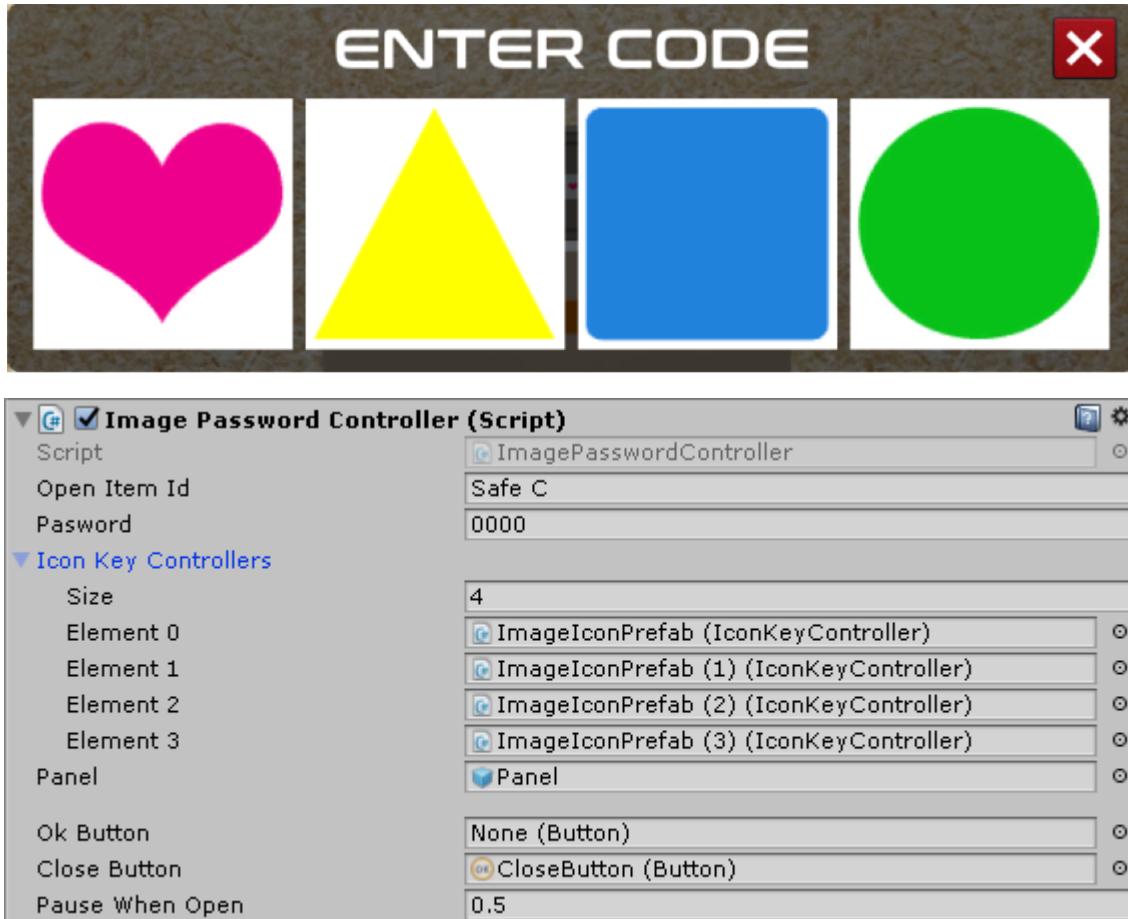
Number Of Hints – number of hints.

Passwords settings

Unlock systems: Image system, Padlock, code panel.

Password-Image

ImagePassword works with safe type Safe C



Settings:

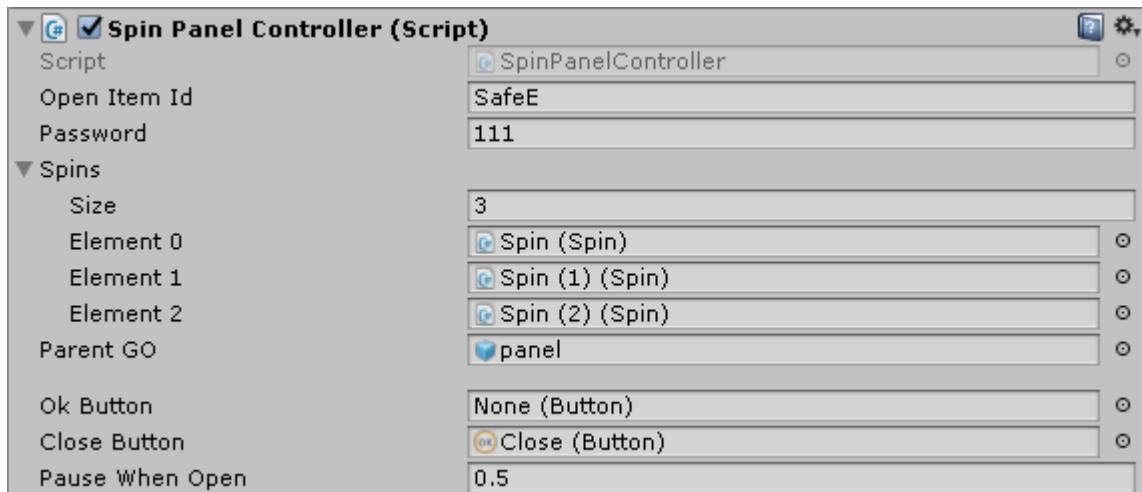
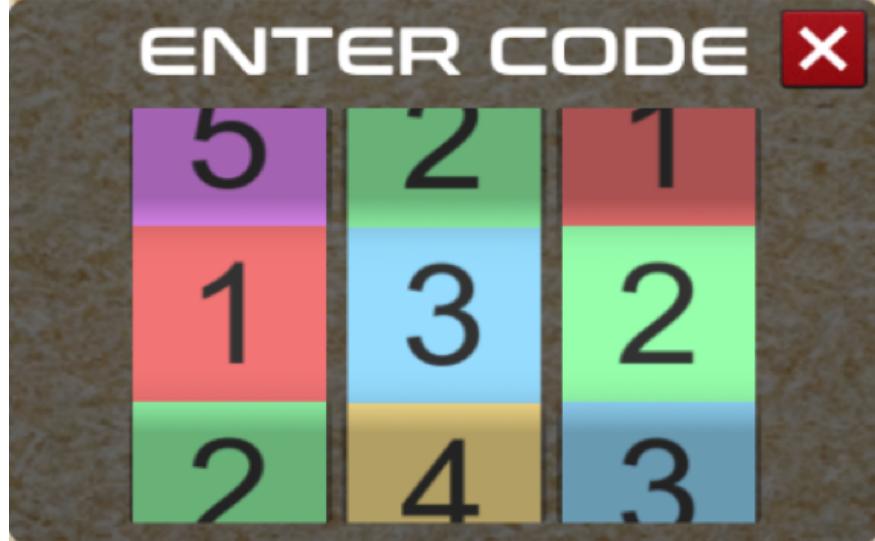
Open Item Id – ID of the element which should be opened.

Icon Key Controllers – list of objects which contain Password-Image.

Password – the numbers that correspond to the ordinal image element in IconKeyController

Password-Padlock

SpinPanel, works with safe type Safe E



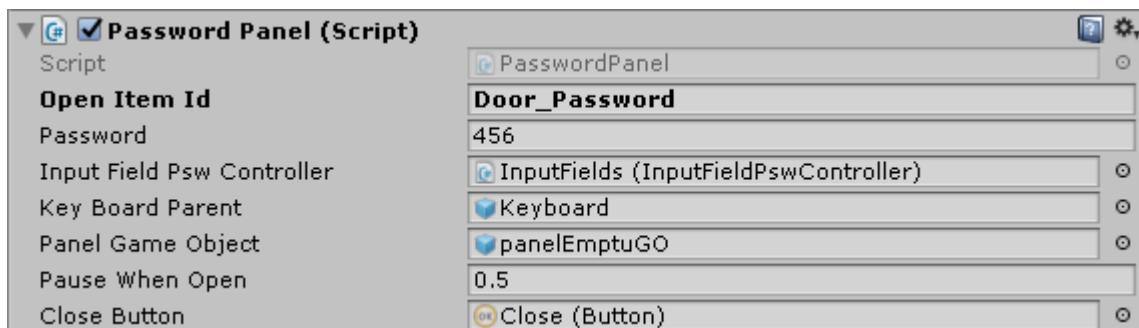
Settings:

Open Item Id – ID of the element which should be opened.

Password – set password

Password-Pannel

PasswordPanel, works with safe type Safe A



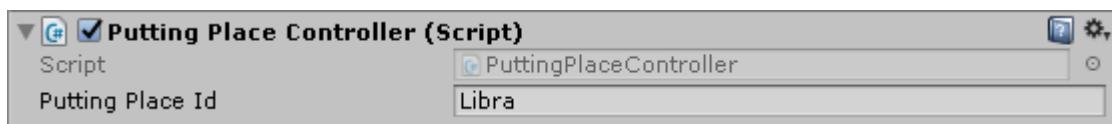
Settings:

Open Item Id – ID of the element which should be opened.

Password – set Password

Libra Example

Example how to use Libra with scale weight in 2 kg.

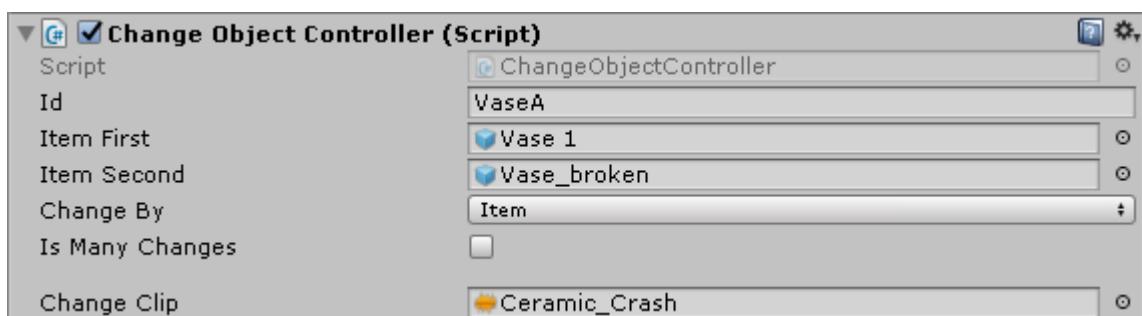


Settings:

Putting Place Id – ID of the element which should be opened.

Object Replacement

Libra Example: Using the object (for example scale weight 2 kg) numbers on Libra are changing and appears scale weight in 2 kg.



Settings:

Item First – switched on object.

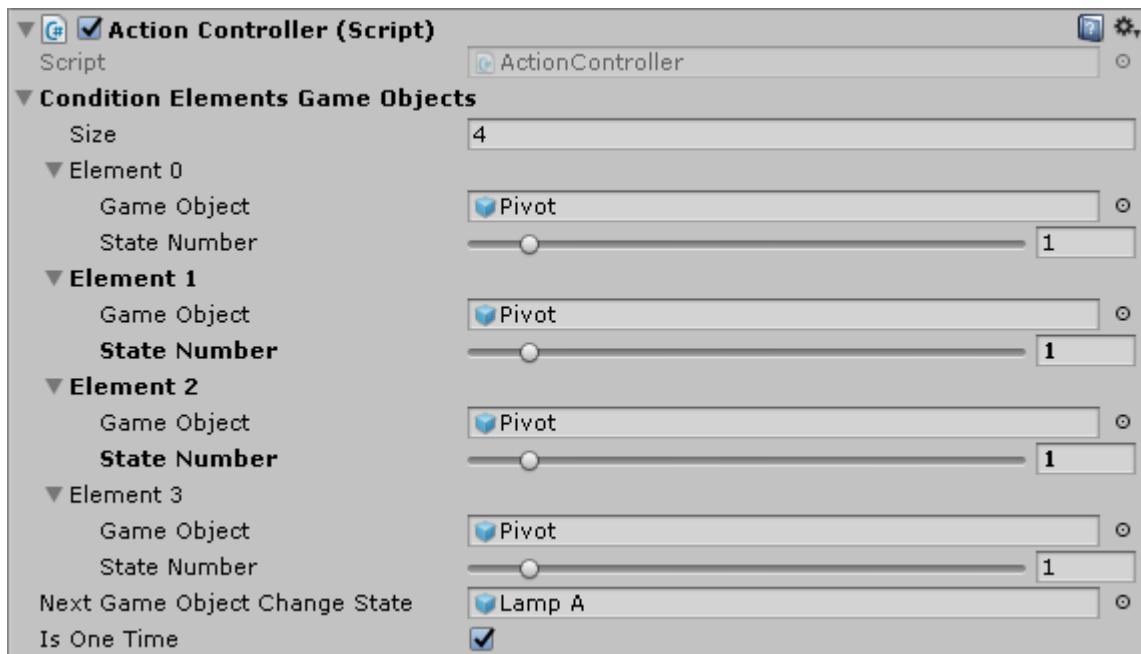
Item Second – the object that should be switched On.

List of Events

Use the script ActionController to build event threads.

For instance, put all switchers to upper position to turn on the light.

Example: Electrical_Switchboard ta Combination



Settings:

Condition Elements Game Objects – list of objects that are involved in the action.

Game Object – an object with the script that is responsible for its

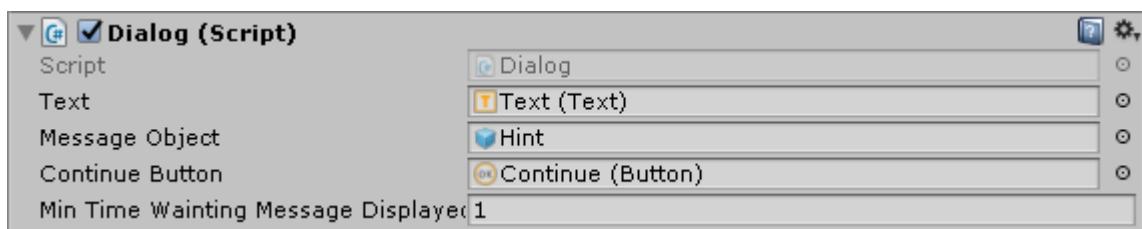
State – If the object has only two states, open / closed, then 0 is closed, and 1 is open.
If it has intermediate states, then the last state is completely open.

Next Game Object Change State – An object that must be executed after all the conditions have been fulfilled with the corresponding implementation script.

Tips

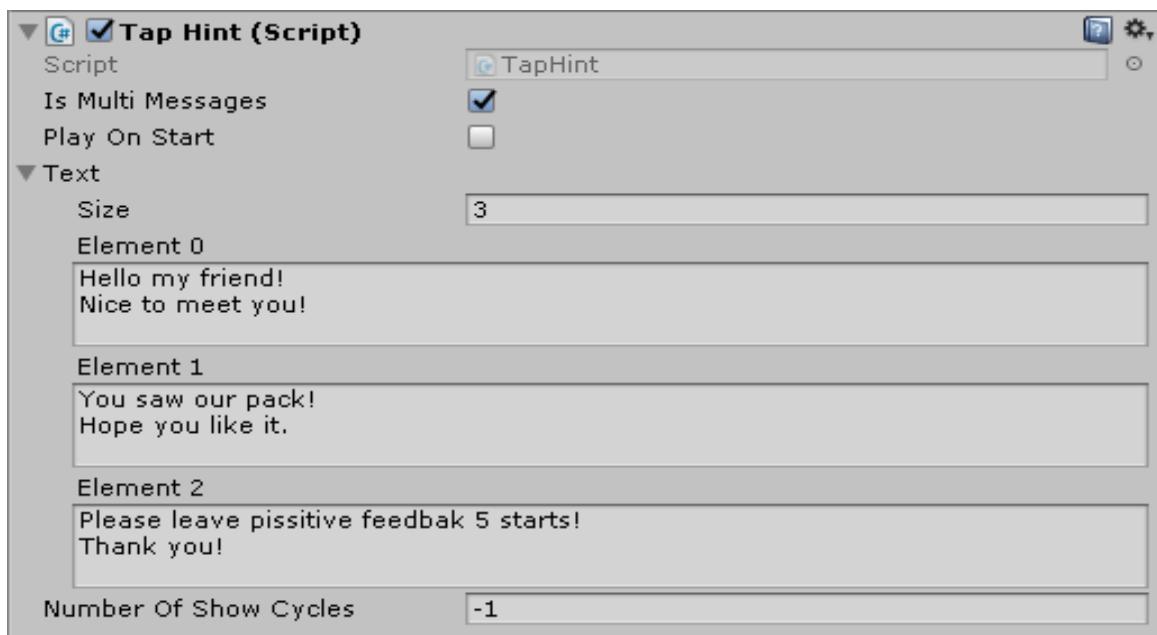


To setup a tip, you need to place the TapHintDialog prefab in the canvas.



And then to display the tips for the desired clickable element (element with collider) place the TapHint component.

Example of how to use Tips is in prefabs: Door_Close, ShowUI.



Settings:

Is Multi Message – If enabled, then texts from the array

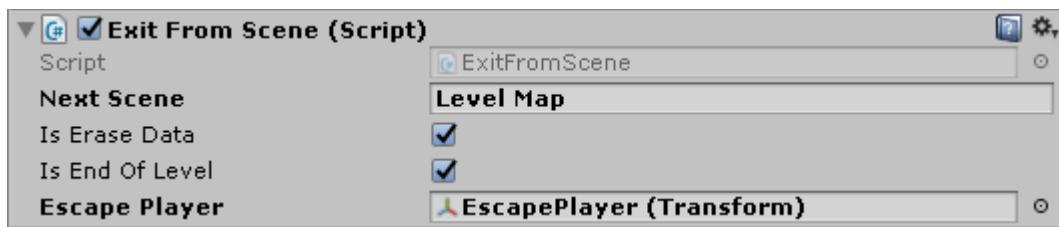
Text will be displayed sequentially, according to the click on the Continue button.
After completing the texts, the window closes.

Number Of Show Cycles – number of cycles to display. If -1, it will show a set of times. If the locked door has this component installed, then the door is unlocked.

Play On Start – shows a hint at the start.

Switchover to next scene

If you use multiple scenes for a single game level or automatically go to the next level, use the ExitFromScene prefab.



Settings:

Next Scene – The name of the scene you want to go to.

Is Erase Data – Deletes current passes. Mostly used at the end of the level.

Is End Of Level – if the end of the level, then maintains the passed level.

Escape Player - looks for and connects the player on the title if it is not stretched in this field

Level Map

Use LevelMenu prefab to build Level Map. Don't forget to put LevelMenu prefab in Canvas.



Each Level has own settings:

Id – unique integer value for this menu

Scene Name – start scene name of the Level

Faced a problem? No worries – our customer support is always there to help you.

Support requests are being processed on business days from 8:00 to 17:00 (GMT + 02.00) [normally] within 24h in the order they were received.

Contact

Email: toda@todagroup.com

Website: www.todagroup.com

Facebook: www.facebook.com/Todagroupcom-702756889837989

YouTube: www.youtube.com/channel/UCWDLrd2woWkJj_BAfYPH89w

**THANKS
TO BE CONTINUE...**