

***SHADOW TRANSFORM***

*Small asset with a big power!*

**Contents**

[What is…? 3](#_Toc502515148)

[How to install? 4](#_Toc502515149)

[How to use? 5](#_Toc502515150)

[Applications of asset 9](#_Toc502515151)

[Limitations 10](#_Toc502515152)

[License and agreement 10](#_Toc502515153)

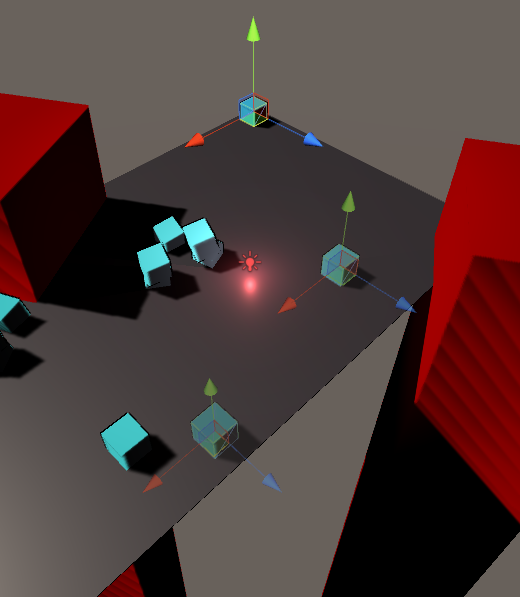
[Special thanks 11](#_Toc502515154)

[Contacts 12](#_Toc502515155)

# What is…?

When you make a game, it’s always necessary to make some tweaks on your levels. Let’s move that rock 3 units left and watch how gameplay has changed.

Later, after a week of tests, you’ve decided that it’s too bad.   
So, let’s return it back to a previous position.



*...does anybody remember where* ***exactly*** *that rock was?*

**ShadowTransform** is a tool to make process of creation and tweaking your levels more comfortable. It will ***remember previous positions*** for any of your objects and let you switch between them in one click.

Also, that’s a great tool for gameplay testing, temporary level re-planning or a massive A/B testing.

# How to install?

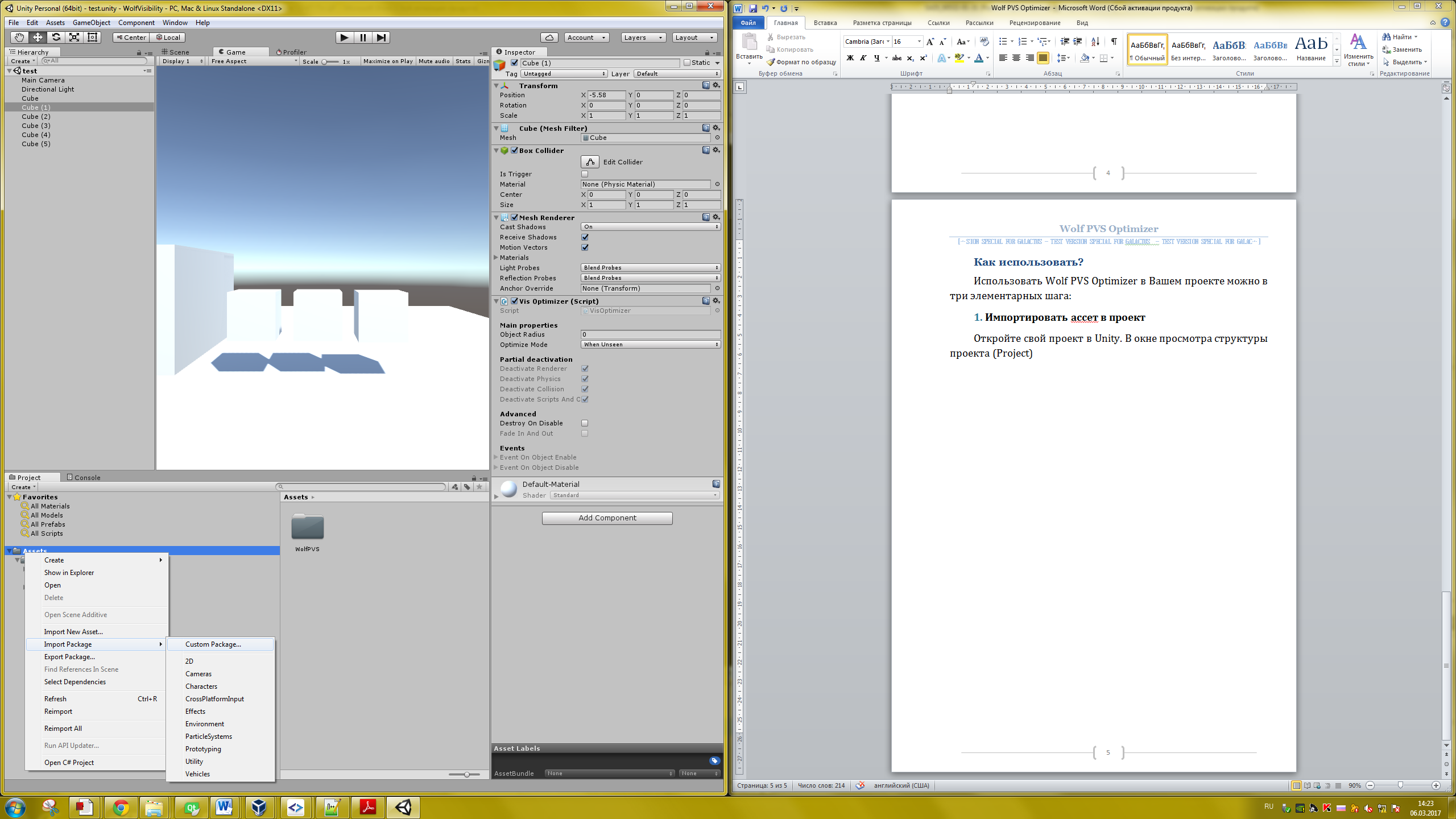
Installation process is kinda trivial:

**If you’ve got an asset from Unity Store:**

* **Import Unity’s «Standard Assets»** **package first!**You’ll need ***«Cameras», «Characters», «Effects»*** *and* ***«Particle Systems»*** parts. If you don’t have one, get it from [Asset Store](https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-32351).
* Proceed latter steps as for site-downloaded asset.

**If you’ve downloaded a package from site:**

* **Right-click** at any place of your project’s tree and **select *«Import Package\Custom Package»***.



* Select a ***«ShadowTransform.unitypackage»*** file.
* Confirm extraction for all files.

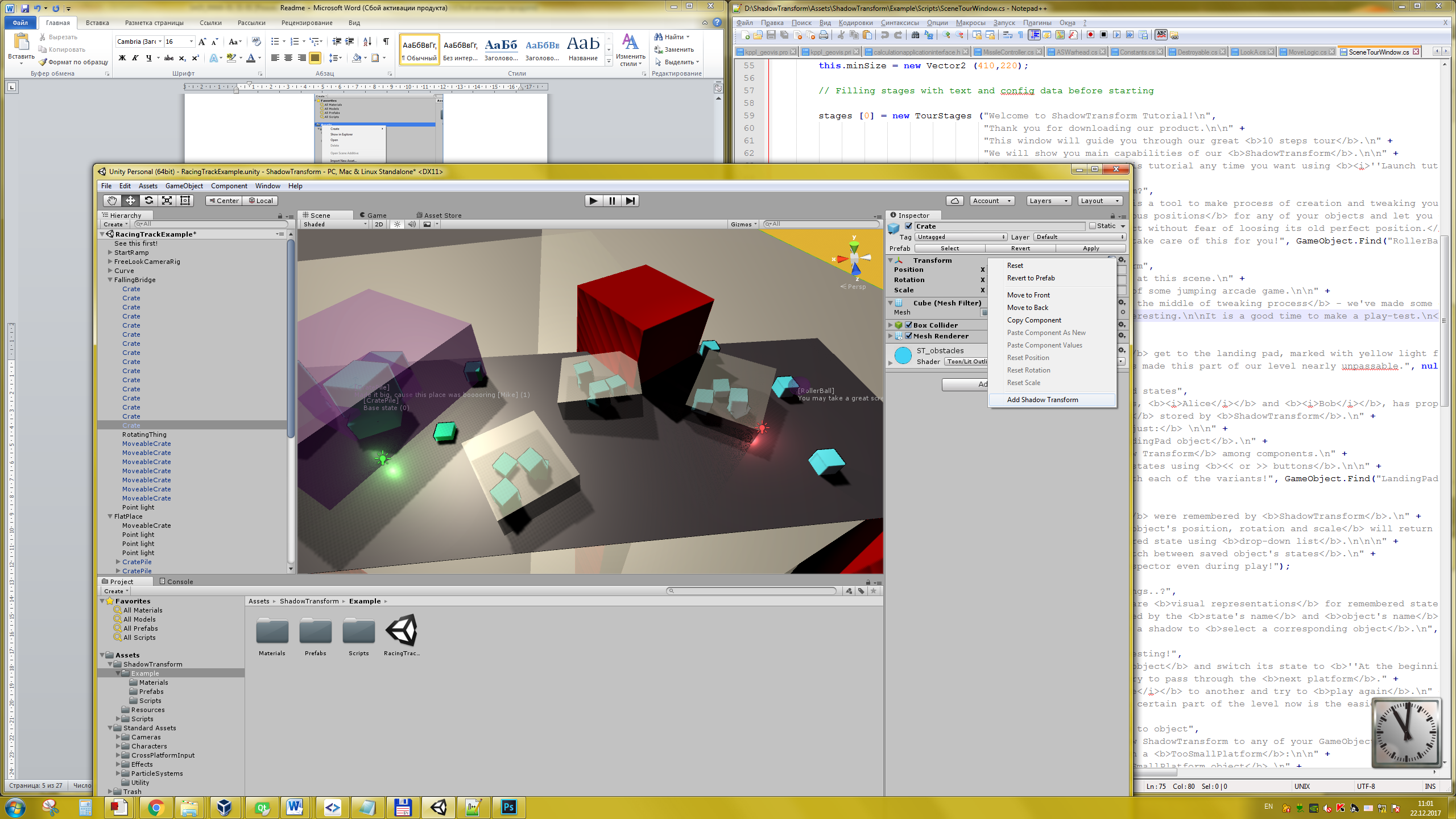
# 

# How to use?

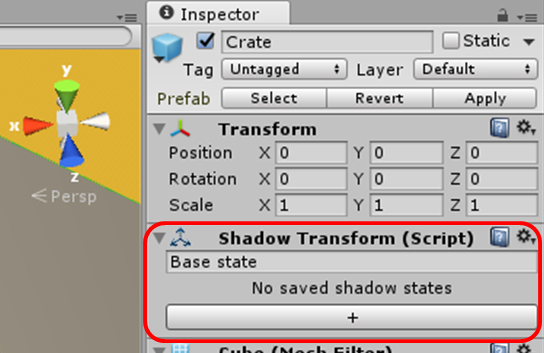
|  |
| --- |
| ***Note:***  **Want a quick start with ShadowTransform?**  ***Open «Example/RacingTrackExample.scene»***  ***and proceed with our in-editor tutorial.*** |

* **1)** **Add a ShadowTransform to object**

To remember states for some object, you must add a ***new******ShadowTransform******component*** to this object. Make a **right-click** at object’s Transform, and select **«Add Shadow Transform»** in menu.



You will see a new component appears below Transform:



New ShadowTransform comes empty, without saved states.

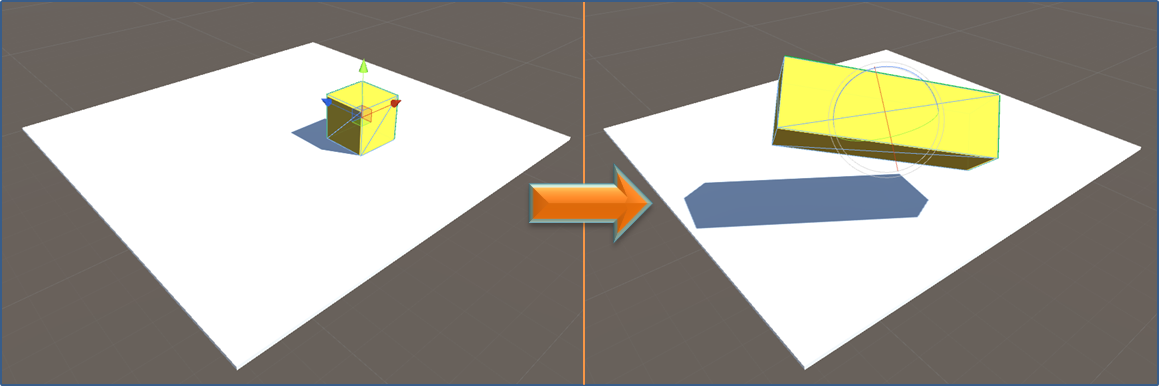
* **2)** **Saving object positions (states)**

**State** is a combination of object’s ***position & rotation & scale***. You may preserve ***up to 256 states*** in a single ShadowTransform.

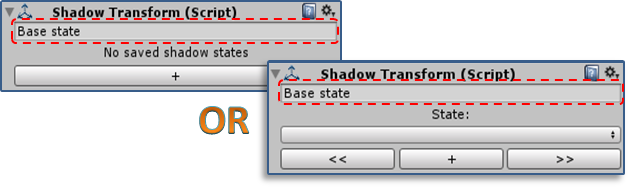
That’s a main function of ShadowTransform – to remember, where your object exactly was. So, you someday may return it back, or switch between its last positions.

**To remember state for an object:**

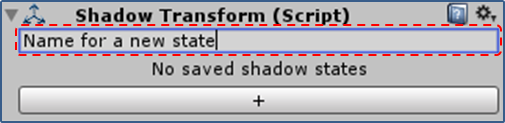
1. Move your object to desired position and rotate and scale it as you wish. *Note, that non-uniform scaling with rotation may be handled incorrectly (Unity3D does not like it anyway).*



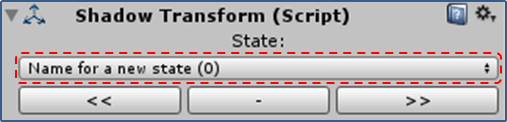
1. An **input line** below component’s caption will become visible.   
   ***It’s visible only when object is not in one of saved states.***

******

1. Enter a name for a new state:



1. Press **«+»** button – new state would be added. Now you may see it in **combo-box**:

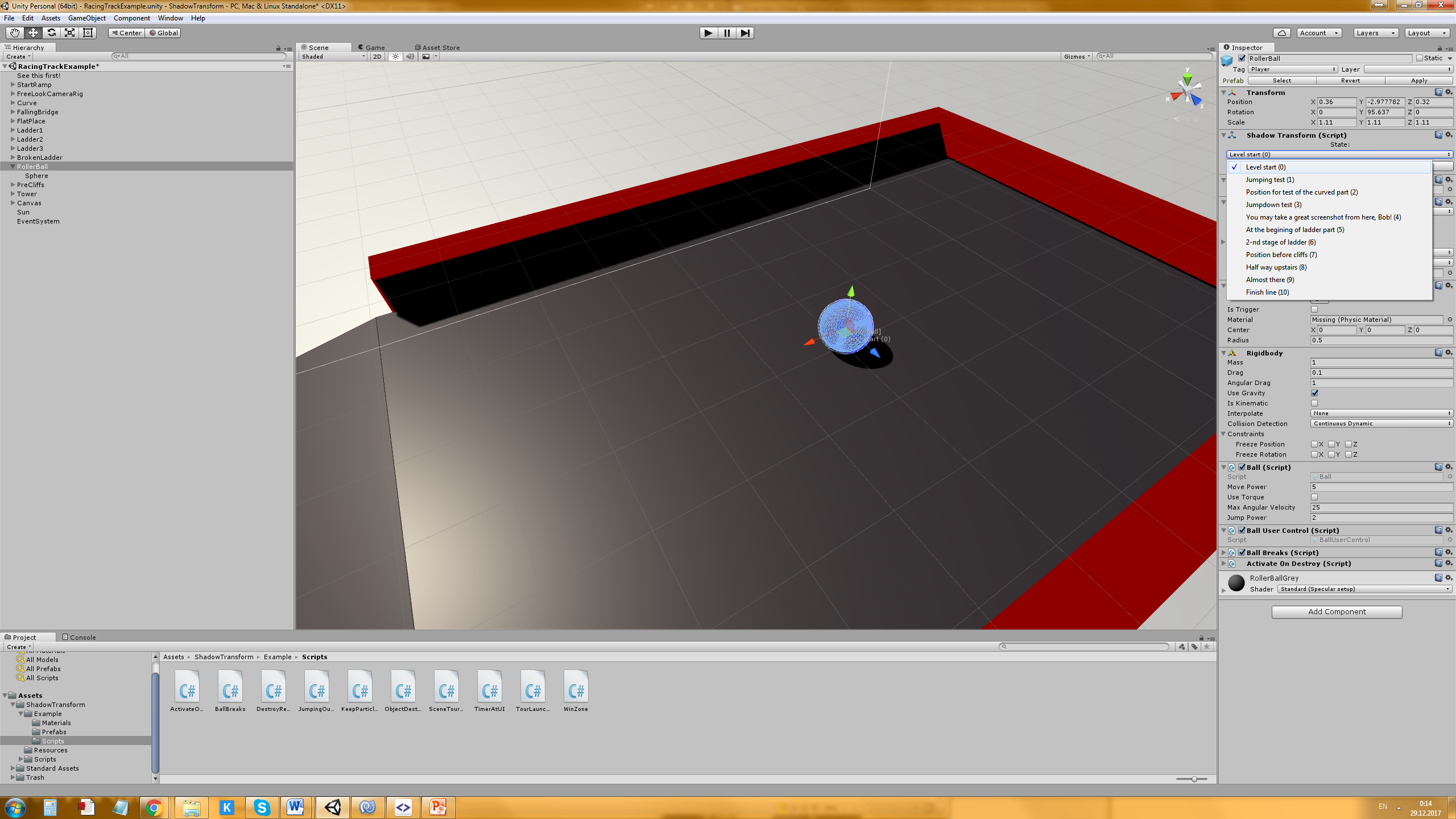


Now your state is saved. You may do anything with your object – scale, move, rotate, parent or unparent it – but you’ll always have a way to return it **where** it was before.

|  |
| --- |
| ***Note:*** You may add a new state **in a** **play mode**.  When you get back to editor mode, all of your added states  **would** **remain in list**.  It’s kinda useful for **playmode-to-editor** transferring and  ***tuning level using playmode***. |

* **3)** **Switching between saved states**

All saved states are listed into **combo-box**:



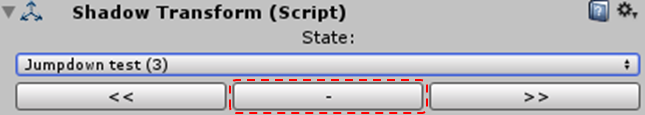
You may switch between states using two ways:

* select state by name using combo-box;
* move to previous or next state using **<<** or **>>** buttons.

When you switch to some state, your current position, rotation and scale would be lost!

* **4)** **Deleting saved states**

To delete some state, you must select it, and then press **–** button.



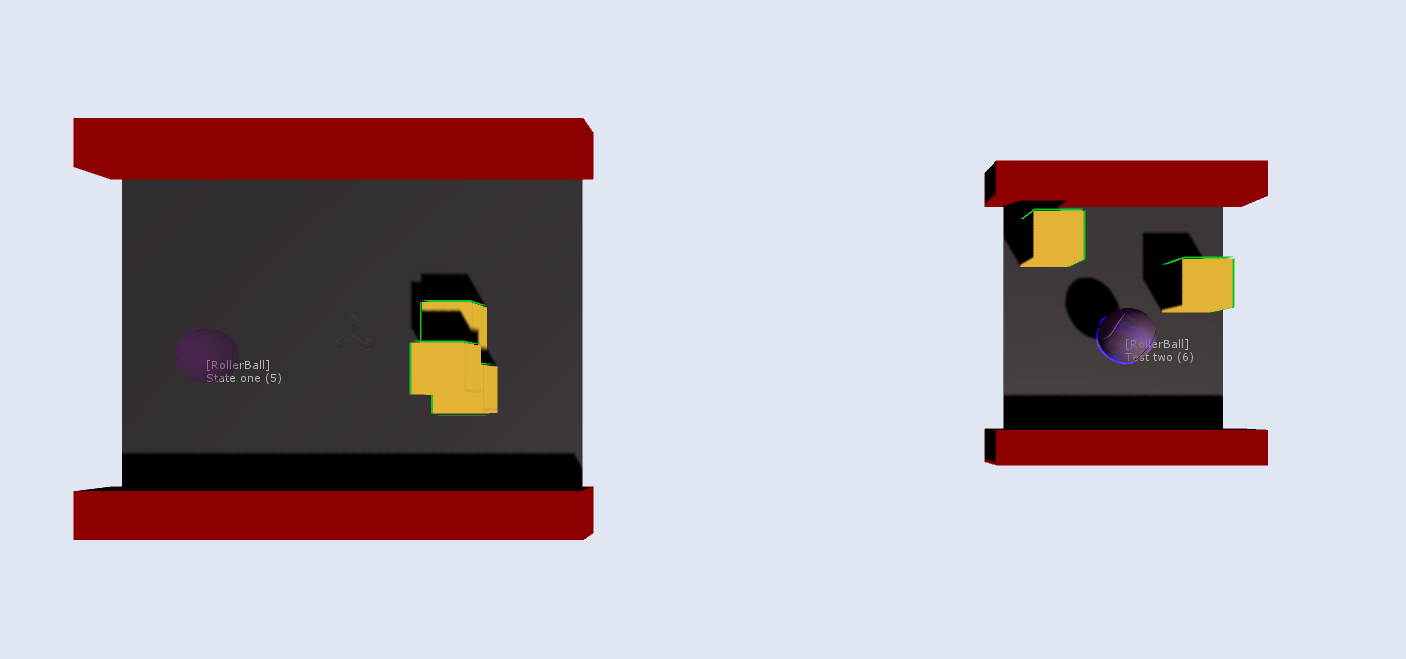
Note that your object would switch to a previous state loosing current position, rotation and scale.

|  |
| --- |
| ***Note:*** You may delete states **in a** **play mode**. |

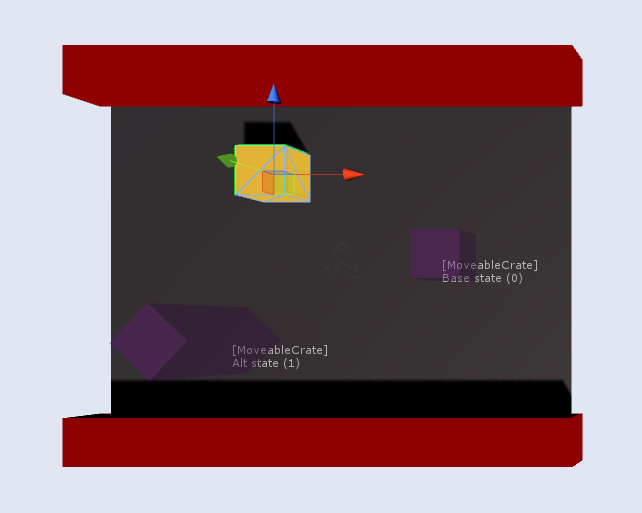
# Applications of asset

When ShadowTransform may come in handy many ways, our asset way made with that in mind:

* ***Play-testing for particular places of the map*** – just make a state before each of part of the level.



* ***Saving object’s states before changes*** – make an experiment, then switch to old state, then return to new and compare.



* ***A/B testing*** – hold all variants in one scene, and just switch between them.

# Limitations

ShadowTranform **may not** work correctly when you try to save state of:

* ***Object with non-uniform scaling & rotation*** *–* Unity hates non-uniform scaling together with rotation. Any object distorts and became an ugly mess. Collider goes insane. So, try not to use it at all, but if you need this badly – just remember, ShadowTransform may not work great with them.
* ***Very large and very far objects*** *–* if you decide to make something at the limit of floating point precision, ShadowTransform may not work properly.

# License and agreement

ShadowTranform comes under glorious **The 3-Clause BSD License** *– do anything you want with ше, just don’t remove my copyrights from sources. It’s not necessary to write about using it anywhere in your game.* ***Full license text included to asset***.



Asset is free for any legal usage, commercial and non-commercial. But, ***if you like it a lot***, please list it somewhere in your game’s credits – *and mail me.* That would be a great news for me! :)

***This asset is distributed «AS IS» and WITHOUT ANY WARRANTY.***

*Some license conditions may vary in future.*

# Special thanks

* Thanks to my mom (**Клёнова Тамара**) and dad (**Ястребов Леонид**) for all than unspeakable love and caring about me.
* To my beloved bride (**Малкова Маргарита**) for being together with me in all of my hard times, and bearing with my heavy and disgusting character ☺   
  My dear Rita, I love you so much!
* To my best friend (**Александр Романов** *aka* ***DanceCommander69***). *You may choose another side, but you’re still my bro.*
* Users **MadDocPrime**, **Samana**, **Lawsonilka** for advices and testing my asset.
* **Unity Technologies** for their Standard Assets *(used in demo resources)*.
* All of my friends and those who are dear to me.

# Contacts

******

***ShadowTransform*** developed by **Ivan Klenov** (aka Wolf4D).

Madness Studio, 2018 г.

All rights reserved (C).

If you need any help, wanna make a proposal, need some advice or want to employ me, feel free to e-mail me:

[**Wolf4D@list.ru**](mailto:Wolf4D@list.ru)

# *Thanks for using ShadowTransform!*