PUZZLE SYSTEM

BY ROMAN SIROKOV

http://puzzlesystem.gitbook.io

1. GET STARTED

In this chapter you will learn the basis of the framework; about the structure that each puzzle must stick to, as well as how to get started and what folder structure the framework has.

1.1. PUZZLE STRUCTURE

Before anything else, it is critical to understand how the puzzles will be structured in your project.

There are 3 types of elements in every puzzle:

- **Puzzle Handler** -- responsible for handling situations when the puzzle has been solved or failed. In other words, defines what actions will be taken on these critical events (e.g., solved puzzle -> open gates for the player).

- **Puzzle Logic** -- responsible for defining what kind of logic the puzzle is following; what player must accomplish in order to solve the puzzle, and what player's actions will lead to the failed condition (e.g., task: activate all triggers).

- **Puzzle Trigger** -- objects that the player will interact with in order to proceed with the puzzle (e.g., press a certain keyboard key in a dedicated zone). It should not contain any logic for puzzle solving. It can, however, contain the information for the Puzzle Logic, where it will be analyzed and the decision will then be given on whether or not this trigger helps to solve the puzzle.

As what up to the in-scene setup, it is required to have the same game object in the scene for both Puzzle Handler and Logic. The triggers will be then stored as children to it. It will ensure clean and intuitive structure in the project hierarchy.

The framework provides a set of default components, so it is possible to create common, but yet customisable, puzzles (almost) without any need to code.

It is also possible to extend the core classes in order to accomplish more complex and thoughtful puzzle experiences.

1.2. INSTALL

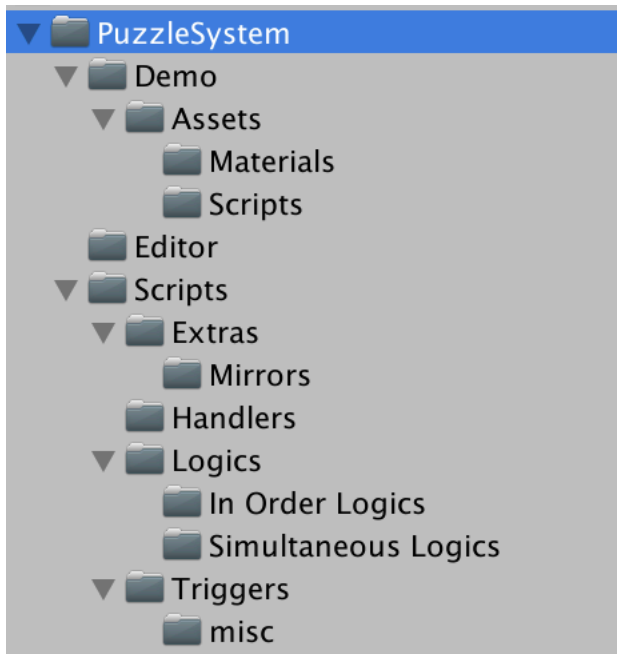
Setting up the framework is a fairly straight forward process.

It can be purchased on Unity's Asset (http://u3d.as/1rMy) Store for 14.99$ and then imported into the project as any other asset in the store.

No further modifications or authorisations are needed.

1.3. FRAMEWORK'S HIERARCHY

After you have imported the package into the project, the "PuzzleSystem" folder will appear in your hierarchy.



Inside you can find:

- **Demo** -- the levels for demonstration purposes including all needed components and assets for their functioning.

- **Scripts**
  - **Handlers** -- basic handler component.
  - **Logics** -- a set of various commonly used puzzle logics.
  - **Triggers** -- a set of various commonly used triggers.
  - **Extras** -- a set of additional assets for commonly used puzzles; less customizable, more specific solutions.

- **Editor** -- the folder that contains scripts that define the editor utility. You won't need these scripts in use of the system. The utility itself is accessible through the upper menu (Tools -> Puzzle System).

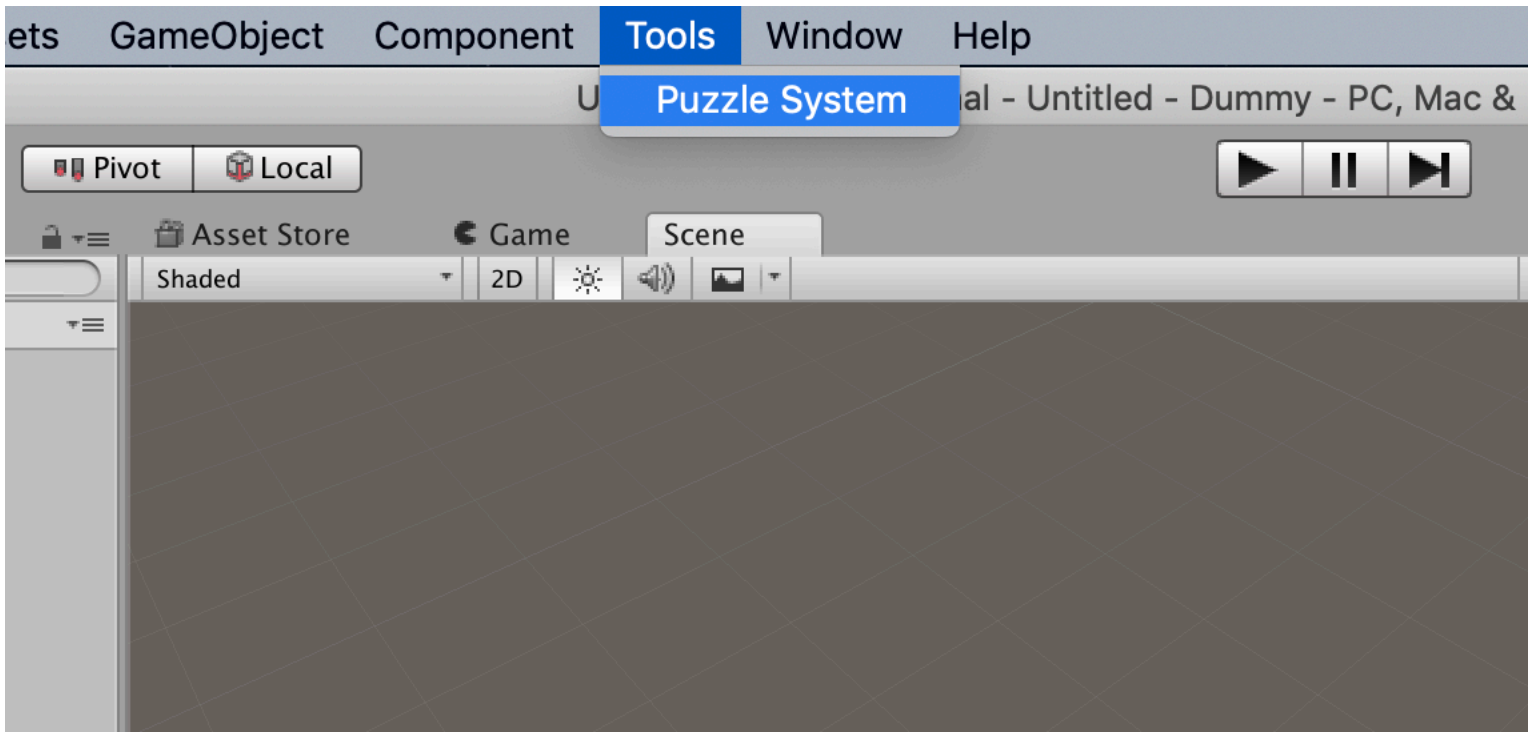
2. PUZZLESYSTEM UTILITY

2.1. GENERAL

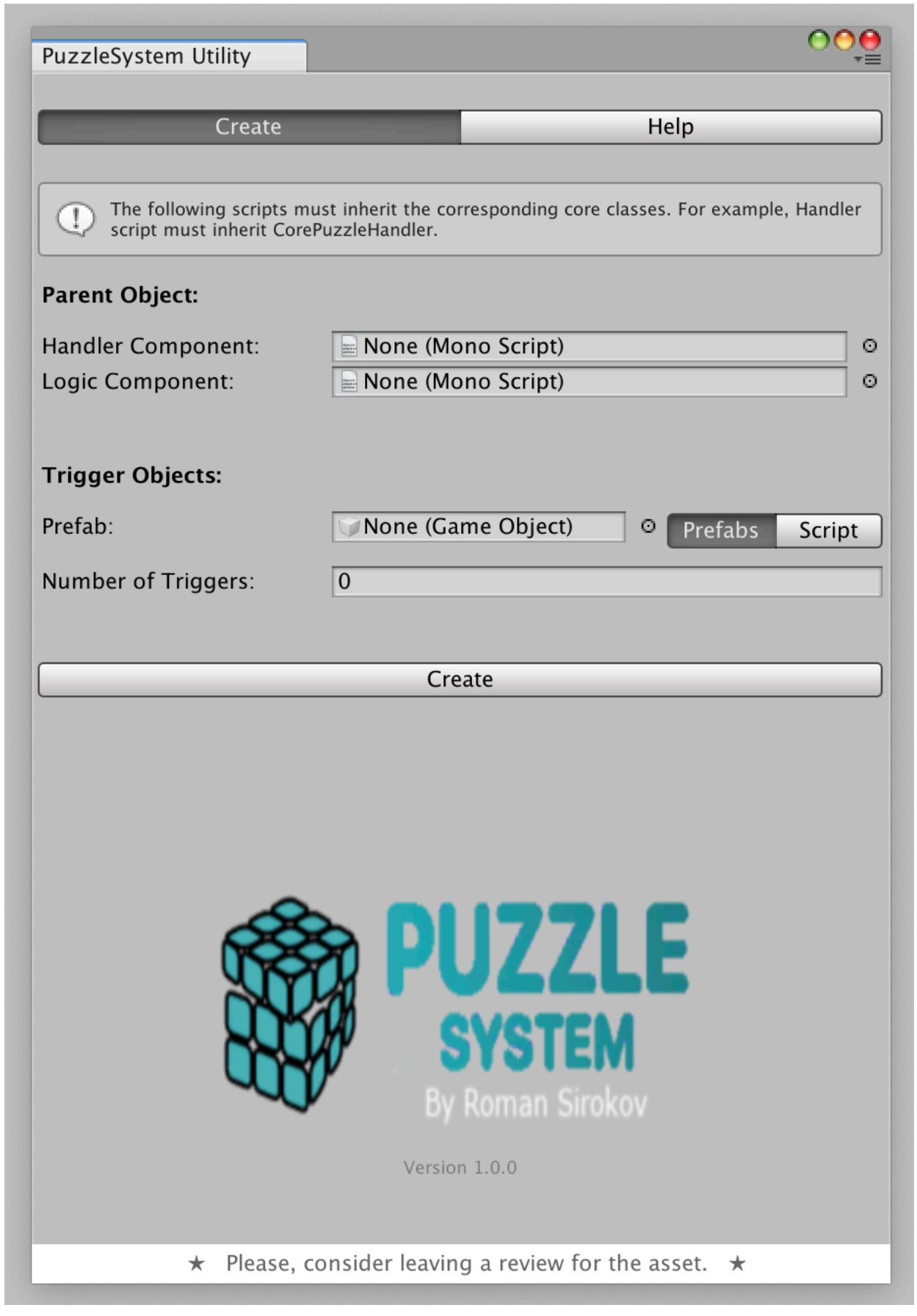
PuzzleSystem Utility is a custom tool integrated into the Unity Editor that will help you to create systems in just a few clicks.

To open it,

<Upper Menu> -> Tools -> Puzzle System



2.2. CREATE



In this tab you can easily create puzzles. In order to do so, you will need to first specify a few parameters.

- Parent Object:

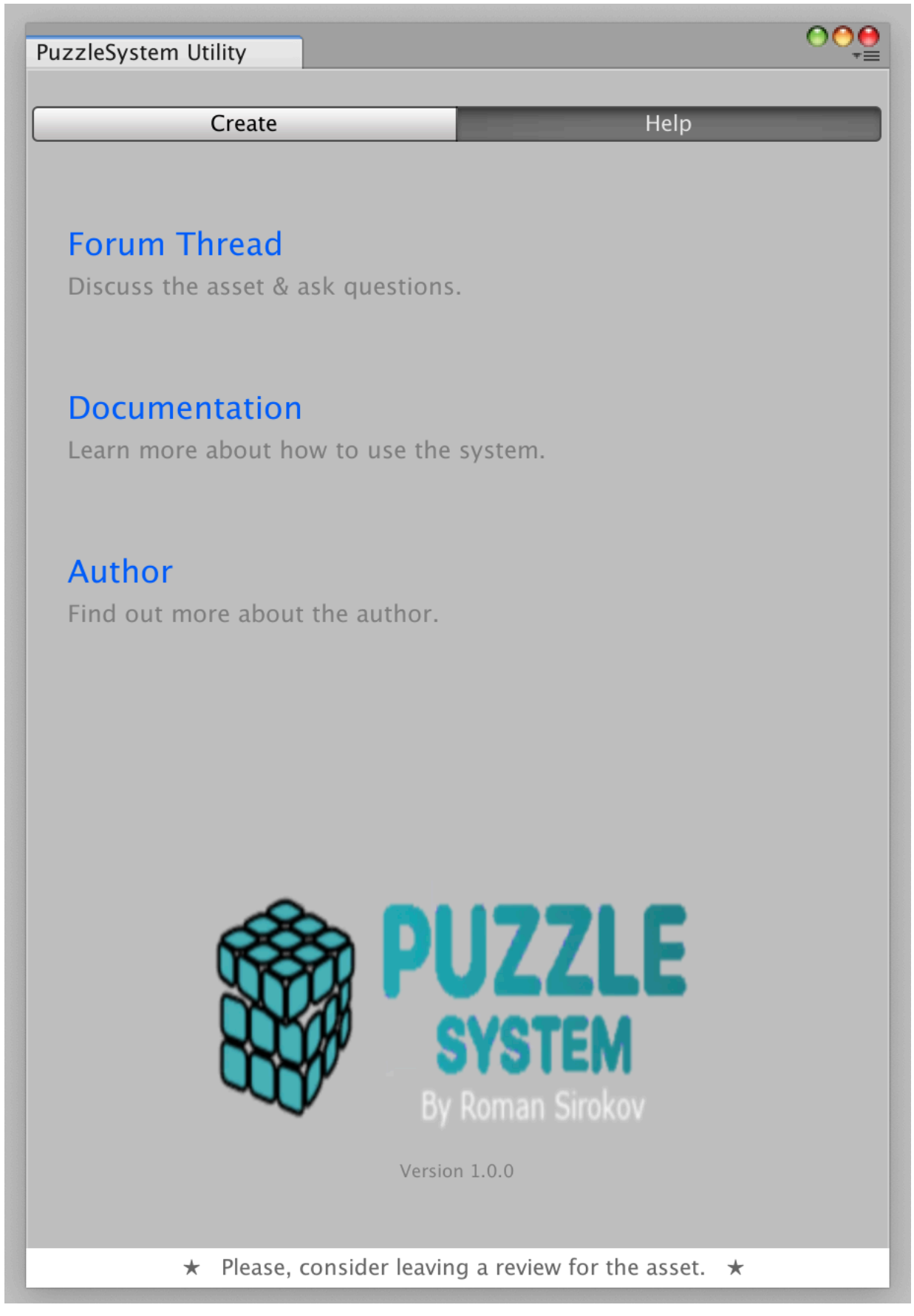
    - Handler Component
    - Logic Component
- Trigger Objects:
    - Prefab or Component
    - Number of Triggers

Make sure to choose the components that inherit from their corresponding core classes. For example, the component you put into Handler field must be a subclass of CorePuzzleHandler.

As soon as you click "Create" button, the puzzle of the specified components will appear in your scene hierarchy. Note that all triggers are already linked to the Logic component. And if you selected one of the Collider based triggers, then BoxColliders will be automatically attached and changed to be triggers.

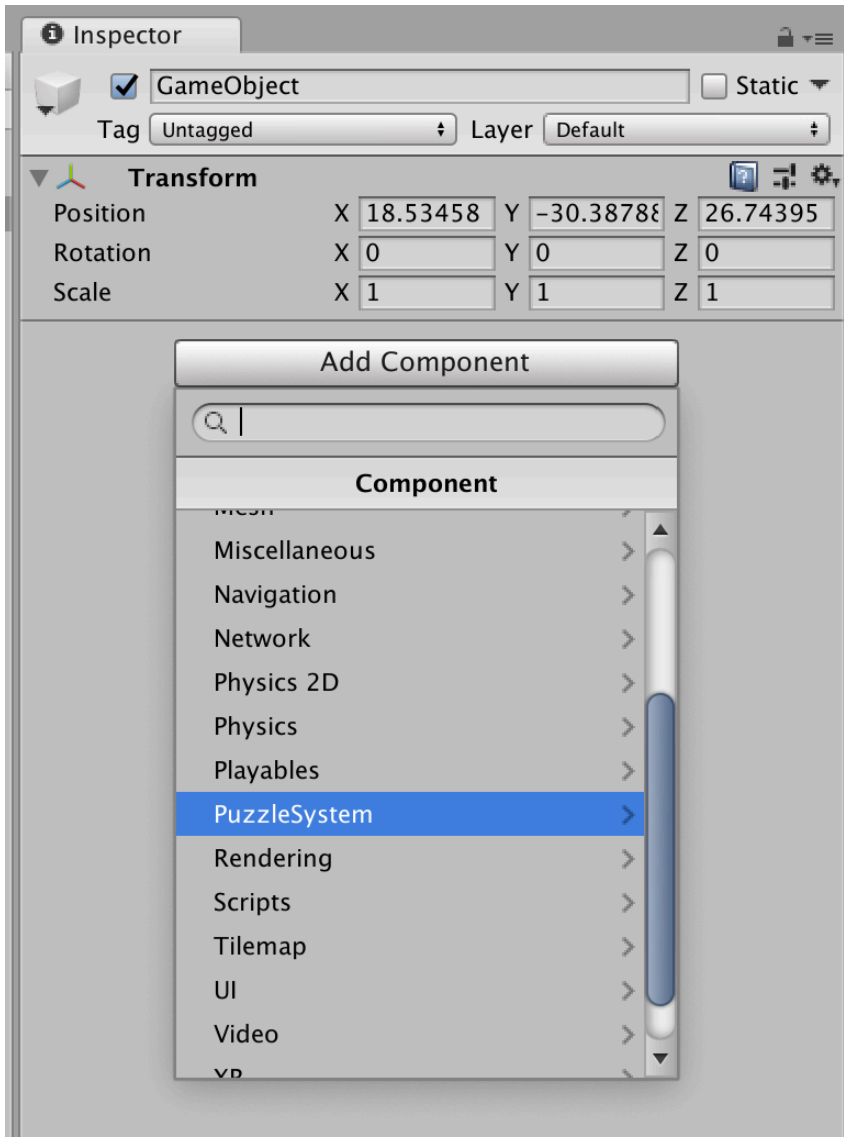
2.3. HELP

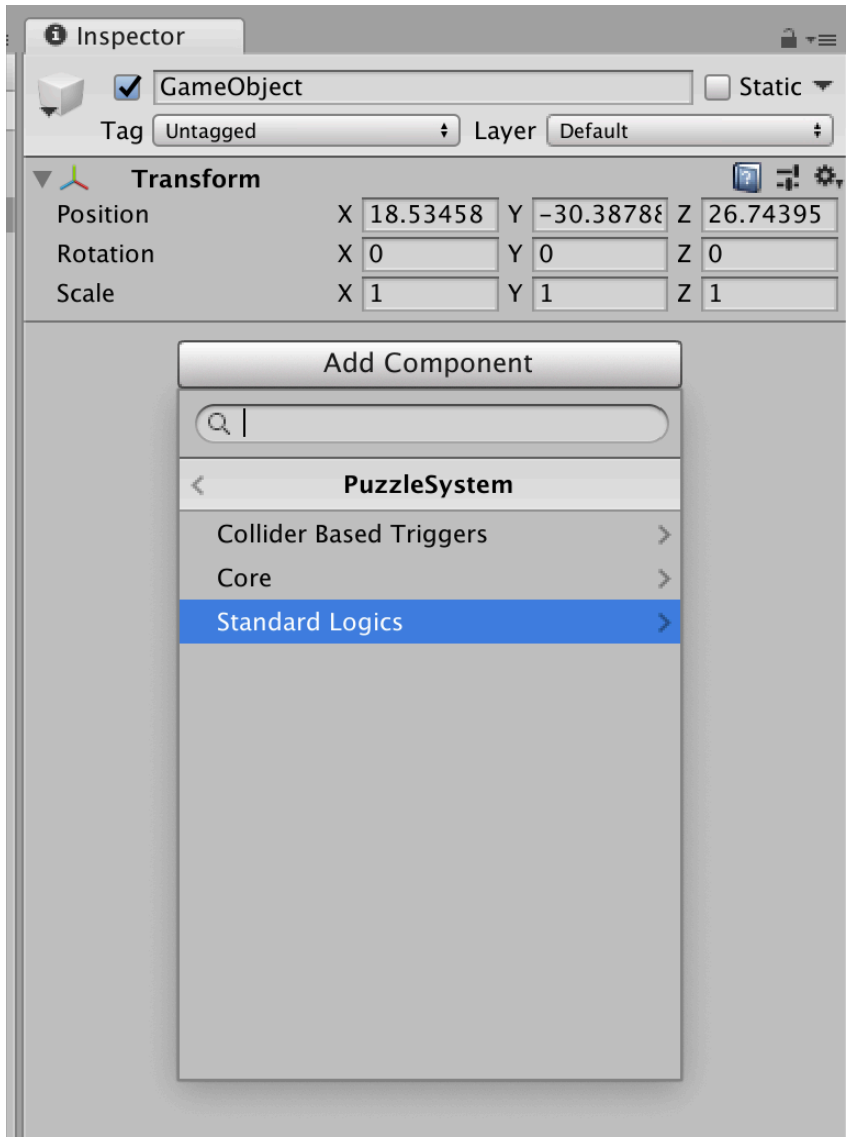
The tool also contains some basic links to this documentation page and to the author's contact details.



2.4. COMPONENT MENU ITEMS

All most important for the puzzles scripts, that are included in the framework, are available for addition in the component menu. You can add them to the objects on the scene quickly and easily.





3. EXTENDED DOCUMENTATION

This documentation is a limited version and lacks a lot of points. For the full documentation, access the website at https://puzzlesystem.gitbook.io.