Constructor example

By Matt Hergaarden - Last edit: 06-10-2009

About the author
The author, Matt Hergaarden, is the artist of the two man game studio "**M2H**" (http://www.M2H.nl). This example project is based on the game Domino Creator. Matt started creating this simple game early 2009 to get some experience with the coding side of unity.

If you made something cool with the help of this example, we'd like to know! Email support@m2h.nl or PM us(Leepo and Zylex) on IRC or the forums.

What is this example about?
This example is not very extensive, however, it showcases a few unique features pretty well:

- Raycasting for object placement & deletion.
- Saving hundreds of objects via name/rotation/position in PlayerPrefs.
- A constructor/simulator: Easily start/stop/pause the physics.

How to use this example?
This document describes the example very basically. After finishing this document check out yourself the code and experiment with it. Feel free to use any code for your own projects, but be warned; don't take over my bad coding habits ;).

About the scripts
CameraControl.js
This script is attached to the cameras parent gameobject. It allows the user to use the arrow keys to move the camera around, furthermore zooming is possible by using the mouse scroll wheel or the + and – keys.

CollisionSound.js
This script is attached to domino objects. It will play a sound when there's a collision with enough force.

SimulationObject.js
This script is attached to all the dynamic objects in the scene. It enabled the simulation mode: Starting, stopping and pausing the physics and resetting to the right position and rotation. The script also features a function to change the objects color.

MainGameScript.js
As the title suggests; this is the main script in the example. This scripts content should have been divided over a few scripts since it features too much: The whole games GUI, input processing and even the actual saving/loading code. Saving and loading is done via raw data strings that are parsed with a custom made parser. The data needs to be divided over a few playerprefs slots due to string size limits.
All code for object placement via raycasting can be found in the Update function. The code for the simulation is very limited here, the only thing thats done in this script is defining the game state (editor/playing) and when switching the game state the SimulationObject script functions are called on all objects.