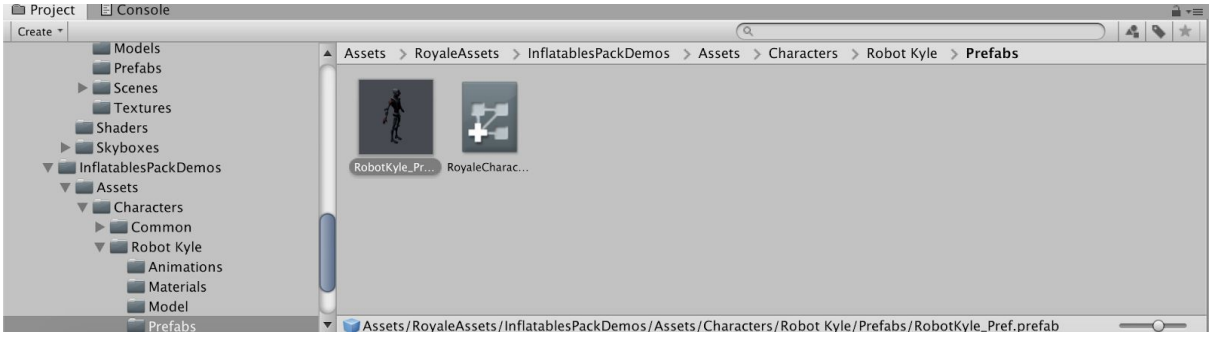
# Table of content:



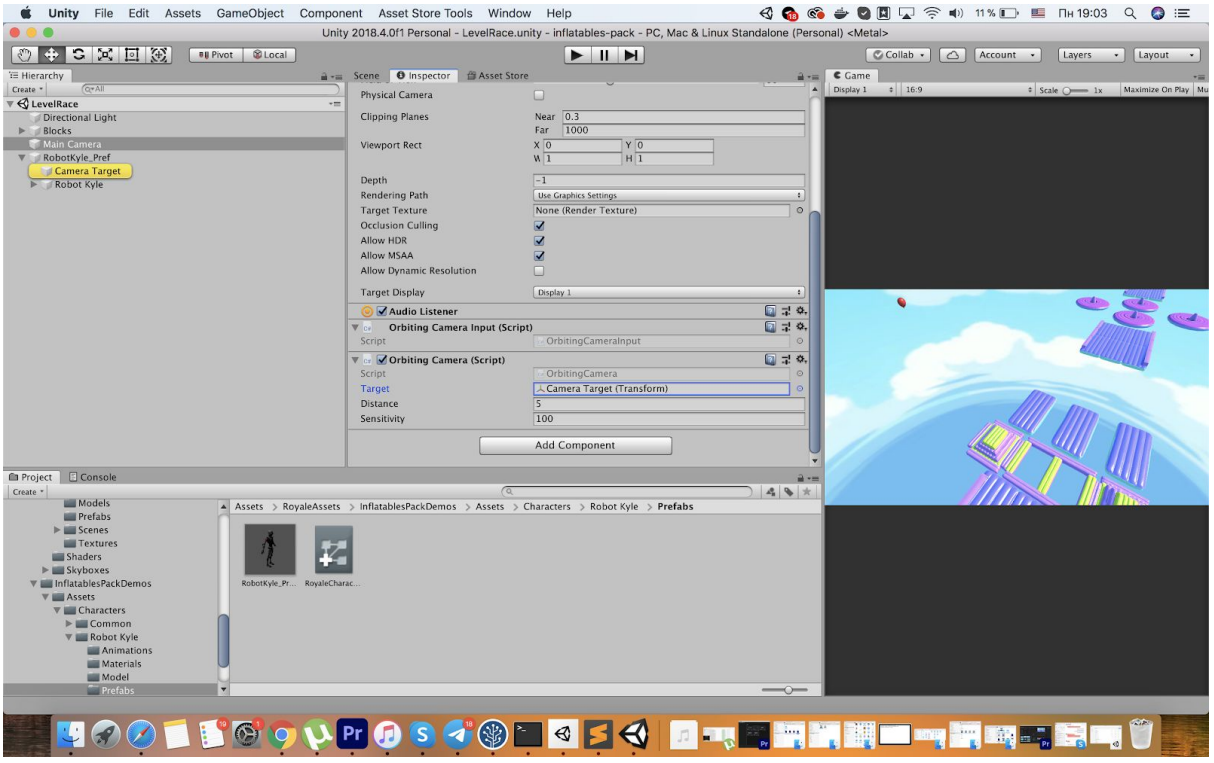
# Getting started

In order to start, create a new scene with a Camera object and place **RobotKyle_Pref.prefab** on scene.



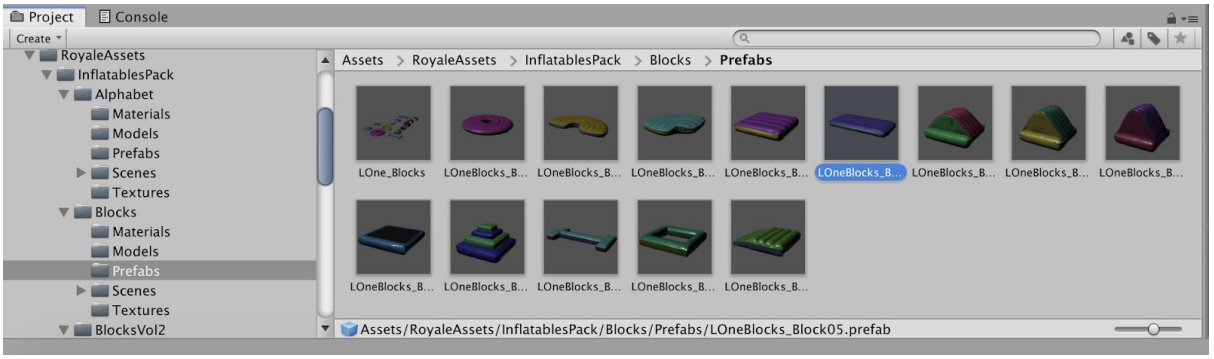

Add **OrbitingCamera** component on your Camera object and fill the

target field with “Camera target” from RobotKyle.




You don’t need to change any parameters on **RobotKyle** attached scripts, because default setup works well in most cases.
Well, the character controller is setted up. Now you can start to build your level.

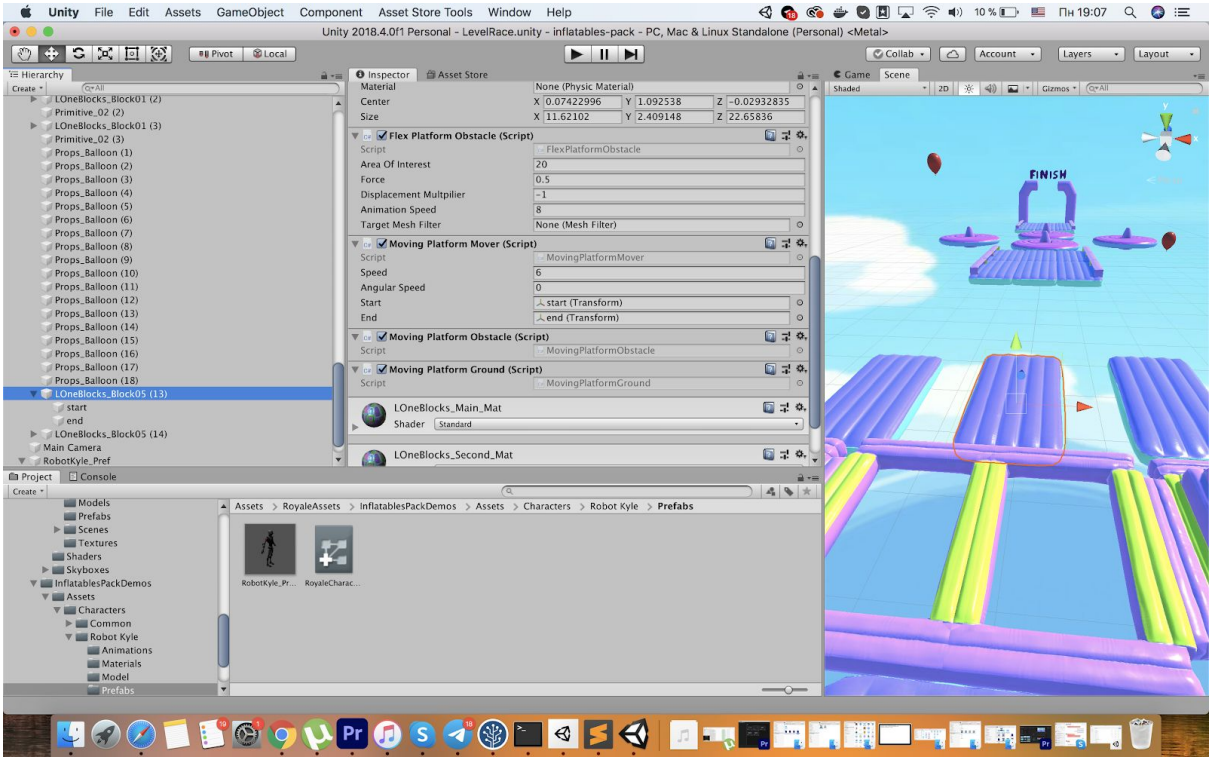
There are lots of building blocks in the Inflatables pack. You can explore them in the **RoyalAssets/InflatablesPack**. Each type of block has prefabs with setted colliders, raw meshes and textures. Let's place one of them on scene.




This prefab doesn't have any physics mechanic. In the next paragraph I will tell you how to add them.

# Movable platform

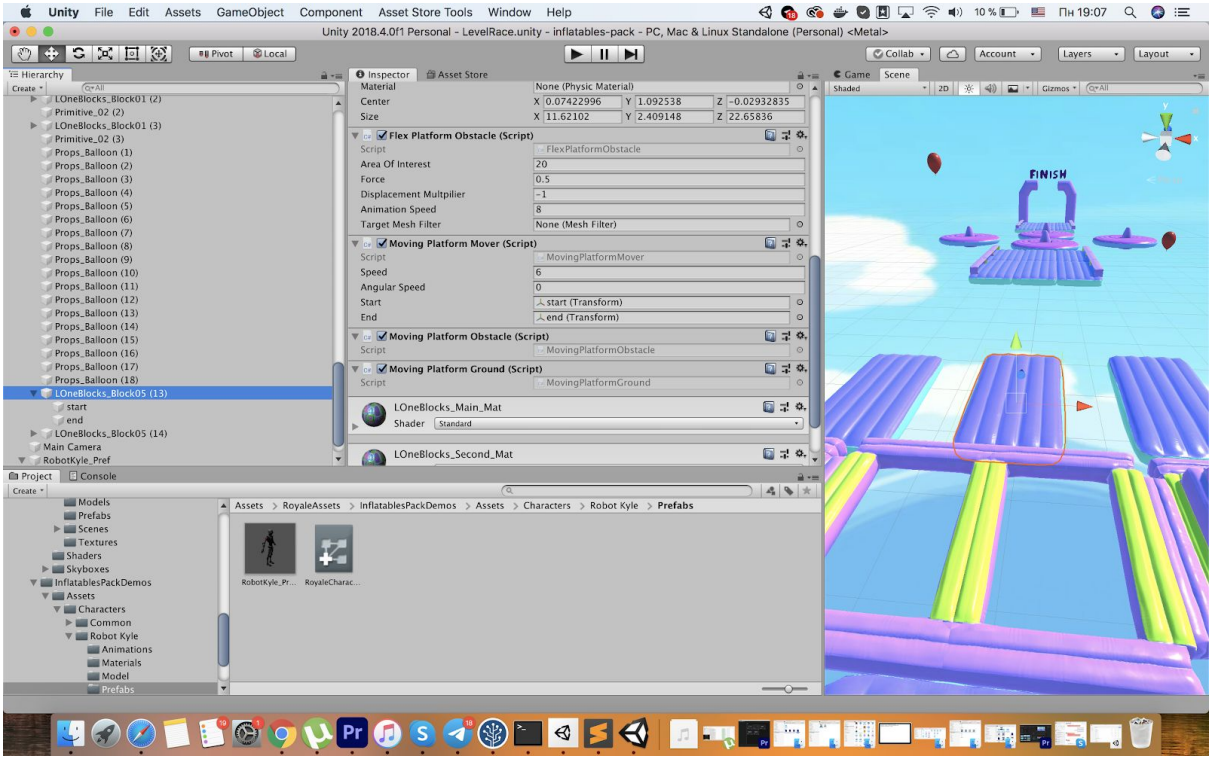
This mechanic moves the platform from point start to point end and rotates it with specified angular velocity. If you don’t need rotation just set this field in zero.




First of all you have to add **MovingPlatformObstacle**, **MovingPlatformGround** and **MovingPlatformMover** script. MovingPlatformMover has dependency on start and end point. You have to create them as children of the current block and set the fields. Also, you should verify that the root GameObject has a **DynamicCollision** layer, otherwise RobotKyle will not interact with it.

# Flex platform

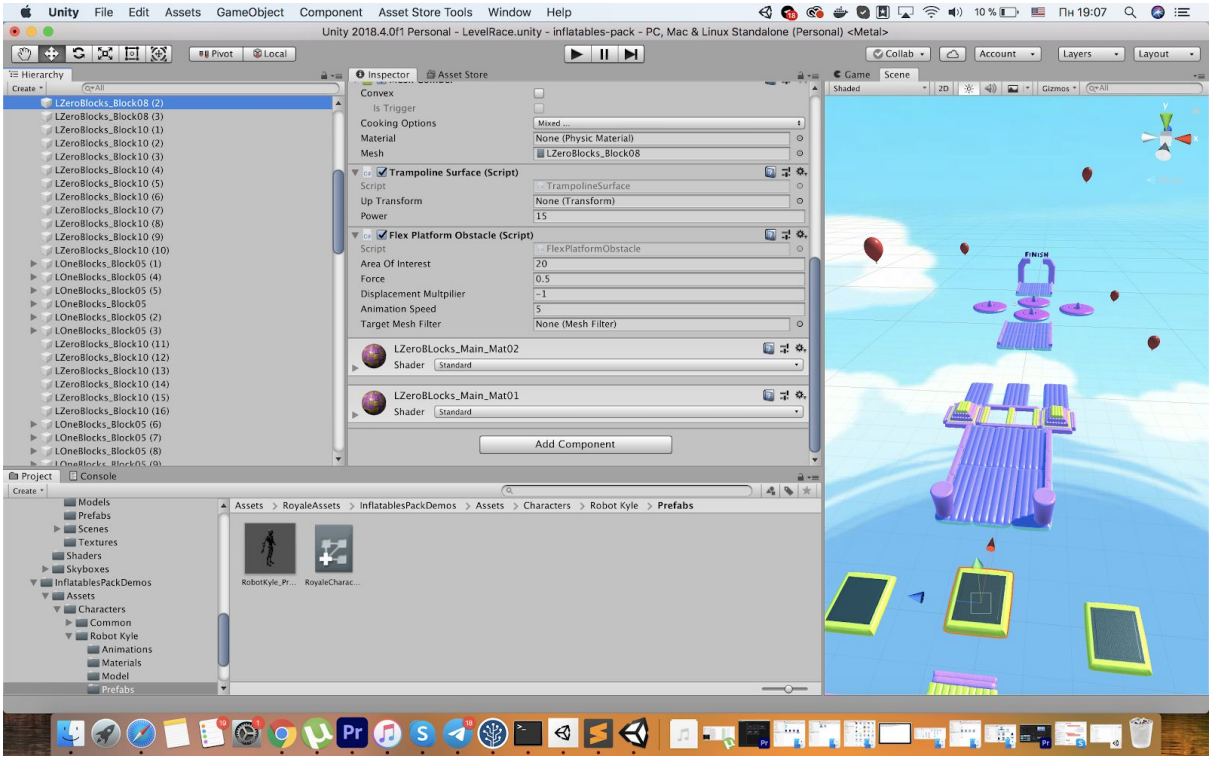
This script makes you mesh rubbery with help of mesh deformation. Before you hang up the script, check that the import settings for the mesh have “Read\write” enabled.




Now you can add a **FlexPlatformObstacle** component on GameObject. This script works well with any other obstacle from this package without additional actions.

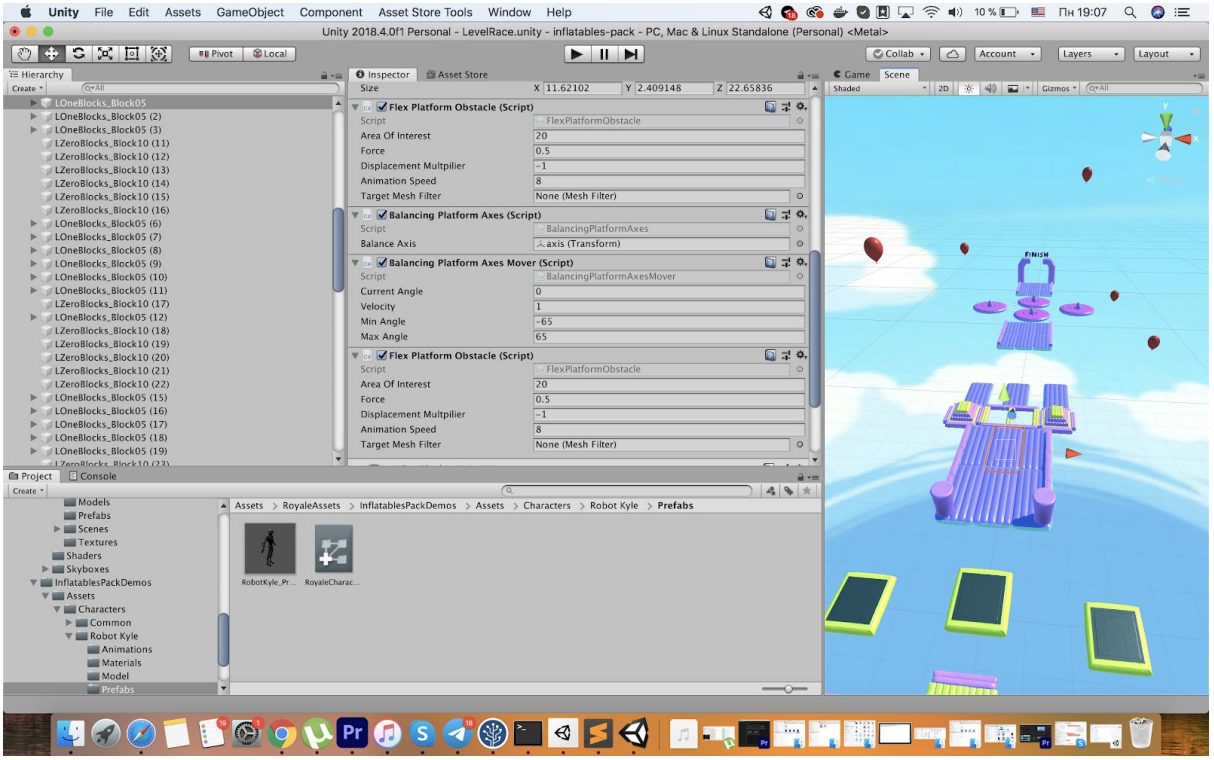
# Trampoline platform

This script transforms your block on trampoline. Each time when controllers collide with it, it changes vertical velocity.




# Balancing platform

When this script is attached, the object tilts under the influence of gravity when the character is standing on it. All you need is add BalancingPlatformAxes and BalancingPlatformAxisMover, create a center of mass (axis) and fill appropriate field.


Unity File Edit Assets GameObject Component Asset Store Tools Window Help
Unity 2018.4.0f1 Personal - LevelRace.unity - inflatables-pack - PC, Mac & Linux Standalone (Personal) <Metal>
Hierarchy
Inspector
Asset Store
Flex Platform Obstacle (Script)
Balancing Platform Axes (Script)
Balancing Platform Axes Mover (Script)
Game
Scene
FINISH
Project
Console
Assets > RoyaleAssets > InflatablesPackDemos > Assets > Characters > Robot Kyle > Prefabs