Welcome to sportbike starter kit ver 1.3 !

First of all let me **thank you** for buying this starter kit.

So, there I try to explain **how it works**:

First you need to know It's not real physics simulator. There is many reasons why. One of them is my first attempt to make anything like real world and absolutely not stable motorcycle which falls every time as result. Yes, it has really round wheel and ruled by principles of countersteering but no fun at all. Why ? Because of physX and wheelCollider isn't represents real world physics. It's a code, and only a fake of some principles.

Do you need physically correct bike which makes you angry ? I don't think so - that's why you have bought that kit. Because demo you saw is fine, smooth and fun.

So, main idea is make looking good motorcycle with fancy controls.

And here we are.

How it works?

With that starter kit you've downloaded:

1. **Motorcycle model** contains few parts: body, wheelbar with upper front forge, front forge lower part, rear pendulum and two wheels
2. **Ground** with some obstacles
3. **new_bike5.js**
4. **new_bike5_mobile.js**
5. **mobileControls.cs**
6. **camBehaviour.cs**

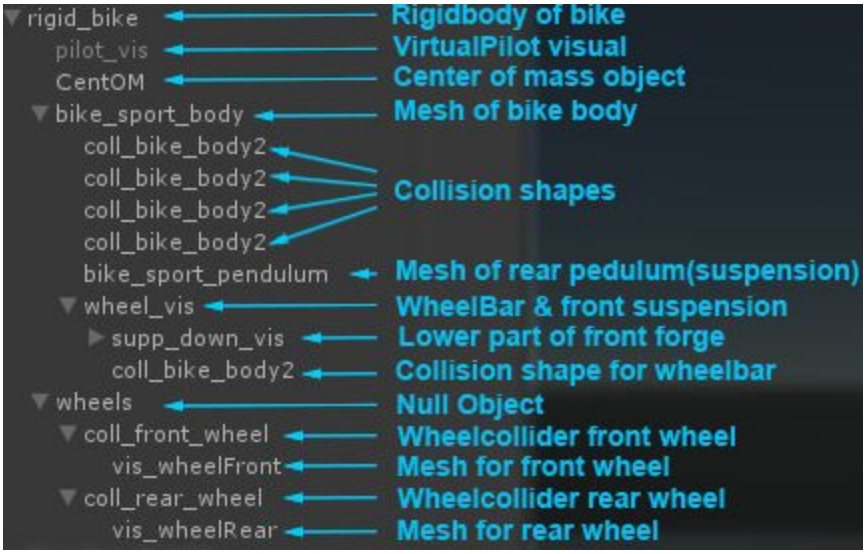
The model is just a **example** how your motorcycle should be **divide** to parts for best visual. So, it means there is only **graphics**. No matter what model and size of parts there is. When you want to export your own bike from 3d modeling package just **follow** the those **principles of cutting** model: **body, two parts of front forge, pendulum, two wheels.** Take care how front forge is made - it's exported NOT ANGLED but strongly vertical and it's already contain lower part(with brake).

Also, take a look to some coll_bike_body2 objects in model. There is gameObjects made in unity just to make collider boundaries. You might make it anything you like. It's for collision and funny rotation bike after fall.

The **ground** - is only for testing and debug. There is useless part of this package.

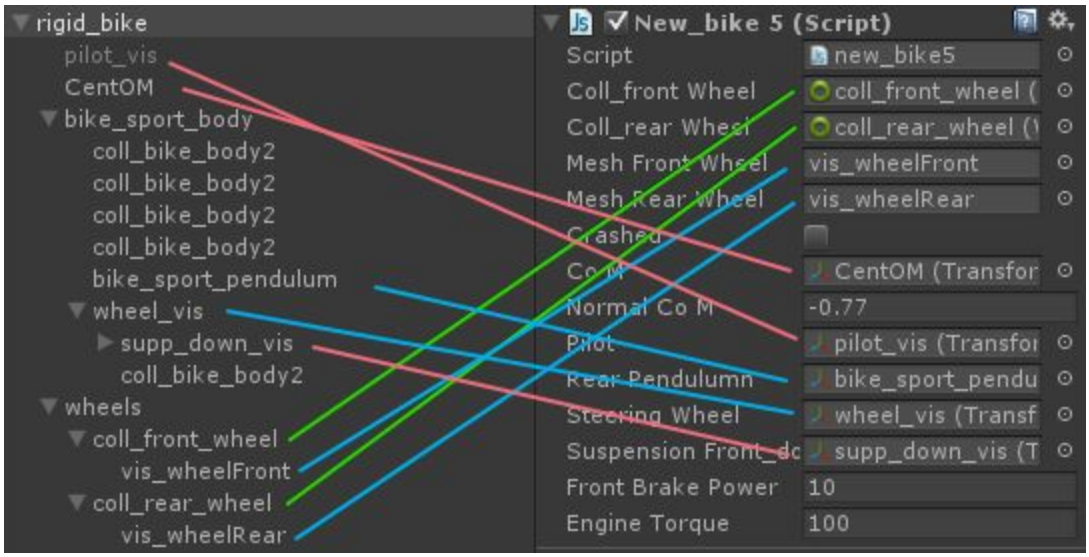
The **code**. This is last thing to explain. Every variable have a nice name which explains point of this variable or comment which explains how this variable is used.

To make work your bike with that code you take **your model** and construct the **same tree** as on the picture:

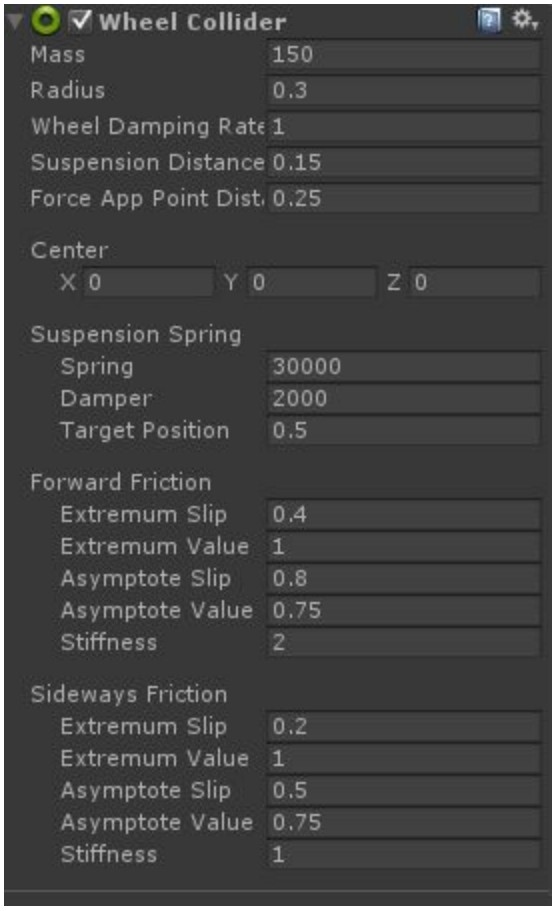


Make sure to **change weight** of rigid body to real weight of motorcycle multiply by 2(new Physics 3.3 “requiments”). In case of this streetfighter its 400kg(200 * 2).
Everything else(drag and angular drag) is doesn’t matter because it’s controlled by script.

And link parts of tree to new_bike.js:



Coll_front Wheel - this variable you should link to front wheel collider gameobject
This is good **wheel collider settings** for front wheel of sportbike(play with settings for chopper or so):



Mass 150kg for a wheel is another new Physics 3.3 “requiments”

Coll_rear Wheel - this you should link to rear wheel collider gameobject
The **settings** of wheel collider is same except **Suspension Distatance - 0.2**(for this streetfighter) and **Force App Point Distance**. It’s **0(zero)** for rear wheel.
There is good place to play if you want chopper.

Mesh Front Wheel - link it with front wheel mesh(FBX)
Mesh Rear Wheel - link it with rear wheel mesh(FBX)
Crashed - used by advanced kit so don’t pay attention to this
Co M - link it to CentOM gameobject. After created you created your own bike tree, place Center of Mass empty gameobject just few centimeters below the line where wheel touches ground.
Pilot - link to pilot_vis. You need that for manual move pilot mass along and across bike. It can be useful for wheelie, stoppie controls and some other stunt things.
Rear Pendulumn - link to bike_sport_pendulum mesh. This pendulum looks at rear wheel.
Steering Wheel - link to wheel_vis(FBX). So, it’s wheelbar of front forge.

Suspension Front_down - link to supp_down_vis. It's lowest part of front forge which looks at front wheel mesh and falls down when it's a gap below wheel.

As for mobile version: there is scene **sportbike_starter_kit_mobile.unit**y with mobile controls(**mobileControls.cs**) and mobile shaders.
It's ready to build for mobile devices.
Notice the path to **mobileControls.cs** - it's placed in folder "**Plugins**".
We need that to compile before java scripts compile.
We "call" this script from **new_bike5_mobile.js**
If you don't' need mobile controls, you may delete everything with word "mobile" in name.

And last thing is **camBehaviour.cs** - script for fancy chase camera.

That's it.
You may change graphics of this motorcyle, change some variables to make it more sporty or chopper like.

version changes:
1.0 - starter version
1.a-c - some minor bugs fixes
1.2 - Unity5.3 support(new ideology of inertiaTensor decoupled from center of mass)
1.3 - Mobile controls added

Feel free to ask me anything about that by e-mail: **smokerr@mail.ru**
Boris Chuprin 2016